

**UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION**

TQ DELTA, LLC,

Plaintiff,

v.

**COMMSCOPE HOLDING COMPANY, INC.,
COMMSCOPE INC., ARRIS US HOLDINGS,
INC., ARRIS SOLUTIONS, INC., ARRIS
TECHNOLOGY, INC., and ARRIS
ENTERPRISES, LLC**

Defendants.

CIV. A. NO. 2:21-CV-310-JRG
(Lead Case)

TQ DELTA, LLC,

Plaintiff,

v.

**NOKIA CORP., NOKIA SOLUTIONS AND
NETWORKS OY, and NOKIA OF AMERICA
CORP.,**

Defendants.

CIV. A. NO. 2:21-CV-309-JRG
(Member Case)

**OPENING EXPERT REPORT OF DR. RICHARD WESEL ON THE INVALIDITY OF
THE ASSERTED CLAIMS OF THE FAMILY 9 PATENTS
(U.S. PATENT NOS. 8,468,411; 9,094,348; 10,833,809)**

TQ Delta Exhibit 2020
COMMSCOPE, INC. v. TQ DELTA, LLC
IPR2022-00833

TABLE OF CONTENTS

I.	INTRODUCTION	1
II.	BACKGROUND AND QUALIFICATIONS.....	1
III.	MATERIALS AND OTHER INFORMATION CONSIDERED	3
IV.	PRIOR TESTIMONY.....	3
V.	COMPENSATION.....	4
VI.	LEGAL PRINCIPLES.....	4
	A. Priority Date	4
	B. Prior Art.....	5
	C. Anticipation Under 35 U.S.C. § 102	6
	D. Obviousness Under 35 U.S.C. § 103.....	6
VII.	LEVEL OF SKILL IN THE ART	9
VIII.	BACKGROUND OF THE TECHNOLOGY.....	10
	A. Description and Timeline of Technologies in Patents at Issue	10
	1. Multicarrier Transmission	10
	2. Reed-Solomon Coding	21
	3. Interleaving.....	28
	4. Retransmission	34
	a. TCP/IP Retransmission	37
	b. Cellular Standard Uses of Retransmission	39
	5. Memory Allocation Using a Common Memory Resource	40
	6. Configuration Messages for Inter-Operability	75
	B. The 802.16 IEEE Air Interface for Fixed Broadband Wireless Access Systems.....	85
	C. ADSL2 Standard	99
	D. VDSL1 Standard	114
	E. VDSL2 Standard	126
	1. O-PMS Message.....	139
IX.	THE FAMILY 9 PATENTS.....	141
	A. Overview of the Family 9 Patents and Asserted Claims	141
	B. File History.....	144
	1. The '411 Patent	144

2.	The '348 Patent	146
3.	The '809 Patent	148
C.	Claim Construction.....	150
X.	OVERVIEW OF THE PRIOR ART	151
A.	ITU-T Standard G.993.1 (“VDSL1 Standard”) (NOK00078763)	151
B.	ITU-T Standard G.992.3 (“ADSL2 Standard”) (NOK00078327)	160
C.	“Interleaved FEC/ARQ coding for QoS multicast over the Internet” by Chen et al. (“Chen”) (NOK00075559).....	160
D.	3GPP TS 25.401 v6.0.0 (“TS 25.401”) (NOK00075136).....	165
E.	3GPP TR 25.896 v6.0.0 (“TR 25.896”) (NOK00074926).....	171
F.	U.S. Patent No. 6,865,233 issued to Eriksson, et al. (“Eriksson”) (NOK00083199).....	179
G.	PCT Publication WO2006/030478 / U.S. Patent No. 8,775,890 to Yap et al. (“Yap”) (NOK00084269).....	182
XI.	PRIOR ART ANALYSIS	184
A.	Motivations to Combine.....	184
1.	Chen in View of ADSL2 Standard.....	184
2.	Yap in View of VDSL1	186
3.	VDSL1 in View of ADSL2	187
4.	3GPP TR 25.896 v6.0.0 and TS 25.401 v6.0.0 in View of Eriksson	190
B.	The Asserted Claims of the '411 Patent Are Obvious	193
1.	Claim 10	193
2.	Claim 18	285
C.	The Asserted Claims of the '348 Patent Are Obvious	294
1.	Claim 1	294
D.	The Asserted Claims of the '809 Patent Are Obvious	360
1.	Claim 4	360
2.	Claim 6	365
XII.	SECONDARY CONSIDERATIONS OF NON-OBVIOUSNESS	367
A.	Commercial Success.....	368
B.	Licensing	368
C.	Standardization	369
XIII.	CONCLUSION	369

I. INTRODUCTION

1. My name is Dr. Richard Wesel, and I have been retained as a technical expert by Defendants Nokia of America Corporation, Nokia Corporation, and Nokia Solutions and Networks, Oy (collectively, “Nokia”) to address certain issues regarding U.S. Patent Nos. 8,468,411 (“the ’411 Patent”); 9,094,348 (“the ’348 Patent”); 10,833,809 (“the ’809 Patent”); 9,485,055 (“the ’055 Patent”). Specifically, I have been asked to opine on, and if necessary, give expert testimony on, the validity of claims 10 and 18 of the ’411 Patent, claim 1 of the ’348 Patent, and claims 4 and 6 of the ’809 Patent.¹ I understand that Plaintiff TQ Delta, LLC (“TQ Delta”) has alleged that certain Nokia products infringe these claims. I further understand that these patents have collectively been referred to as the Family 9 Patents.

2. This report addresses the validity of the claims currently asserted by TQ Delta. Should TQ Delta be permitted to assert additional claims against Nokia, I reserve the right to amend this report accordingly.

II. BACKGROUND AND QUALIFICATIONS

3. Below is a summary of my educational, professional, and technical background as it pertains to the subject matter of the Family 9 Patents. A true and correct copy of my curriculum vitae is attached as Appendix A.

4. I have over 30 years of experience in communications and signal processing. I am a professor in the Electrical and Computer Engineering Department of UCLA as well as the Associate Dean for Academic and Student Affairs of the Henry Samueli School of Engineering and Applied Science. After receiving bachelors and master’s degrees in Electrical Engineering from MIT in 1989, I worked at AT&T Bell Laboratories from 1989 to 1991 as a Member of Technical Staff

¹ I understand Dr. Bruce McNair will opine on the validity of claim 17 of the ’055 Patent. I reserve all available rights to also provide testimony on this patent, to be set forth in future reports.

performing research and development in telecommunications. I attended Stanford University from 1991 to 1996, receiving my Ph.D. in Electrical Engineering in 1996. Upon receiving my Ph.D., I joined the faculty of the UCLA Electrical Engineering Department, where I have been teaching and doing research on communications and signal processing for the last 24 years.

5. I have extensive experience researching and teaching communications techniques including multi-carrier modulation, interleaving, and error control coding in general as well as Reed-Solomon coding in particular. My publications and patents are listed in my CV. I have published over 200 conference and journal publications, and I am the inventor on nine patents. I have received the National Science Foundation (NSF) CAREER Award and an Okawa Foundation award for research in information theory and telecommunications. I am a Fellow of the Institute of Electrical and Electronics Engineers (IEEE). I previously served as an Associate Editor for Coding and Coded Modulation for the *IEEE Transactions on Communications*. I currently serve as an Associate Editor for Coding and Decoding for the *IEEE Transactions on Information Theory*.

6. I authored an Asilomar conference paper in 1995 entitled “Fundamentals of Coding for Broadcast OFDM” that discusses techniques for designing coding and modulation for multicarrier transmission with interleaving. My 1996 Ph.D. dissertation included the design of trellis codes for multicarrier transmission with interleaving. In a 1999 Communications Letter, I presented related results on trellis codes for multicarrier transmission with interleaving. In a 2000 *IEEE Transactions on Communications* paper, I compared the performance of these new codes to Reed-Solomon codes used on multicarrier transmission with interleaving. I also authored several conference papers in the late 1990’s related to coding for multicarrier transmission with interleaving including “Joint Interleaver and Trellis Code Design,” “Periodic Symbol Puncturing of Trellis Codes,” and “Trellis Codes for Compound Periodic Gaussian Channels.”

7. I authored the chapter entitled “Error Control” in the book Wireless Multimedia Communications: Networking Video, Voice, and Data in 1997, which discussed the concepts of Reed-Solomon codes and interleaving. This book also discusses multicarrier modulation in Section 3.4 and Section 5.2.

8. I am an inventor on U.S. Patent No. 6,125,150, filed in October 1995 and entitled “Transmission System using Code Designed for Transmission with Periodic Interleaving,” which discloses, among other things, error control techniques for multicarrier transmission with interleaving. I am also an inventor on U.S. Patent No. 6,158,041, filed in October 1998 and entitled “System and Method for I/Q Trellis Coded Modulation” which also describes, among other things, error correction coding techniques for multicarrier transmissions using interleaving.

III. MATERIALS AND OTHER INFORMATION CONSIDERED

9. My opinions are based on my years of education, research, and experience, as well as my study of the relevant materials. A list of materials I considered is included as Appendix B.

10. I reserve all available rights to supplement this report if further information becomes available, or if I am asked to consider additional information. This report represents only those opinions I have formed to date.

11. Further, I reserve all available rights to consider and comment on any additional expert statements or testimony of TQ Delta’s expert(s) in this matter.

IV. PRIOR TESTIMONY

12. The cases in which I have testified as an expert at trial or by deposition within the preceding four years are listed in Appendix C.

V. COMPENSATION

13. I am being compensated at my usual consulting rate of \$600 per hour for my time spent working on issues in this matter. My compensation does not depend on the outcome of this matter or the opinions I express.

VI. LEGAL PRINCIPLES

14. In arriving at the opinions set forth in this report, I have applied the following legal principles provided to me by counsel.

A. Priority Date

15. I understand that the “priority date” (also called the “earliest effective filing date”) of a patent is the date on which it was filed. If the patentee claims priority to an earlier-filed U.S. or international patent application, the priority date of the patent is the priority date of the patent is the filing date of the earlier-filed U.S. or international patent application.

16. I understand that absent evidence of an invention date prior to the filing date of a patent, the invention date of the patent is presumed to be its filing date. A prior invention requires a complete conception of the invention and a reduction to practice of that invention. The patent owner has the burden of establishing by clear and convincing evidence a date of conception earlier than the filing date of the patent.

17. I understand TQ Delta stated in an interrogatory response that “[a]ny inventions in [the ’411 Patent, the ’809 Patent, and the ’055 Patent] that are the subject of claims asserted in this action have dates of conception that are at least on or before April 12, 2006, and dates of constructive reduction to practice that are on or before April 12, 2006” and “[a]ny inventions in [the ’348 Patent] that are the subject of claims asserted in this action have dates of conception that are at least on or before October 5, 2006, and dates of constructive reduction to practice that are on

or before October 5, 2006.” TQ Delta’s Response to Nokia’s Interrogatory No. 5. For the purpose of my analysis in this report, I applied these priority dates.

B. Prior Art

18. I understand that a patent or other publication must first qualify as prior art before it can be used to invalidate a patent claim.

19. I understand that a U.S. or foreign patent qualifies as prior art under 35 U.S.C. 102(a)² if the date of issuance of the U.S. or foreign patent is more than one year before the earliest effective filing date of the asserted patent. I understand that a printed publication (such as an article published in a magazine or trade publication) qualifies as prior art to an asserted patent (regardless of the date of invention of the asserted patent) if that printed publication published more than one year before the earliest effective filing date of the asserted patent.

20. I understand that a U.S. patent qualifies as prior art under 35 U.S.C. § 102(e) if the application for that patent was filed before the invention of the asserted patent. I understand that, for prior art patents under 35 U.S.C. § 102(e), a patentee may try to show that the claimed invention was conceived before the U.S. patent’s filing date. I have been informed and understand that, to do so, a patentee must prove (with corroborating evidence) that the named inventors conceived of the complete claimed invention before the prior art issued/was published and that the named inventors were diligent in reducing the claimed inventions to practice.

21. I understand that a prior art reference must contain an enabling disclosure to qualify as prior art. I have been informed and understand that an enabling disclosure is one that allows a person of ordinary skill in the art to practice the claims without undue experimentation. I have been informed and understand that U.S. patents are presumed to contain an enabling disclosure.

² Because the Family 9 patents were filed before the AIA passed in 2011, I cite to the pre-AIA version of 35 U.S.C.

22. I understand that documents, materials, and prior inventions that qualify as prior art can be used to invalidate a patent claim as anticipated under 35 U.S.C. § 102 or obvious under 35 U.S.C. § 103.

C. Anticipation Under 35 U.S.C. § 102

23. I understand that a patent claim is invalid under 35 U.S.C. § 102 if a single asserted prior art reference discloses each and every limitation of the claimed invention, either explicitly, implicitly, or inherently, to a person having ordinary skill in the art to which the subject matter pertains. I understand there must be no difference between the claimed invention and the disclosure of the alleged prior art reference as viewed from the person of ordinary skill in the art. I understand that the reference must describe the claimed subject matter with sufficient clarity to establish that the subject matter existed and that its existence was recognized by persons of skill in the art. Additionally, I understand that, to establish that a claim limitation is “inherent” in the disclosure of a prior art reference, extrinsic evidence must make clear that the missing limitation is necessarily found in the prior art and would be recognized by a person of ordinary skill in the art.

D. Obviousness Under 35 U.S.C. § 103

24. I understand that a patent claim is invalid under 35 U.S.C. § 103 if the differences between the invention and the prior art are such that the subject matter as a whole would have been obvious at the time of the invention to a person having ordinary skill in the art to which the subject matter pertains. Obviousness, I am informed, is determined based on the scope and content of the prior art, the differences between the prior art and the claim, the level of ordinary skill in the art, and secondary indications of non-obviousness to the extent they exist.

25. I understand that whether there are any relevant differences between the prior art and the claimed invention is to be analyzed from the view of a person of ordinary skill in the art at the time of the invention. A person of ordinary skill in the art is a hypothetical person who is

presumed to be aware of all of the relevant art at the time of the invention. The person of ordinary skill is not an automaton, and may be able to fit together the teachings of multiple patents (or other printed publications) by employing ordinary creativity and the common sense that familiar items may have obvious uses in another context or beyond their primary purposes.

26. In analyzing the relevance of the differences between the claimed invention and the prior art, I understand that I must consider the impact, if any, of such differences on the obviousness or non-obviousness of the invention as a whole, not merely some portion of it. The person of ordinary skill faced with a problem is able to apply his or her experience and ability to solve the problem and also look to any available prior art to help solve the problem.

27. I understand that an invention would have been obvious if a person of ordinary skill in the art, facing the wide range of needs created by developments in the field, would have seen an obvious benefit to the solutions tried by the patent applicant. When there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions, it would have been obvious to a person of ordinary skill to try the known options. If a technique has been used to improve one device, and a person of ordinary skill in the art would recognize that it would improve similar devices in the same way, using the technique would have been obvious.

28. I understand that I do not need to look for precise teachings in the prior art directed to the subject matter of the claimed invention. I understand that I may consider the inferences and creative steps that a person of ordinary skill in the art would have employed in reviewing the prior art at the time of the invention. For example, if the claimed invention combines elements that were known in the prior art and if the combination yields results that were predictable to a person of ordinary skill in the art at the time of the invention, then this would make it more likely that the claim was obvious. On the other hand, if the combination of known elements yields unexpected or

unpredictable results, or if the prior art teaches away from combining the known elements, then this would make it more likely that the claim that successfully combined those elements was not obvious. I understand that hindsight must not be used when comparing the prior art to the invention during an obviousness analysis.

29. I understand that obviousness may be shown by demonstrating that it would have been obvious to modify what is taught in a single piece of prior art to create the patented invention. Obviousness may also be shown by demonstrating that it would have been obvious to combine the teachings of more than one item of prior art. I understand that a claimed invention may be obvious if some teaching, suggestion, or motivation exists that would have led a person of ordinary skill in the art to combine the invalidating references. I also understand that this suggestion or motivation may come from sources such as explicit statements in the prior art, or from the knowledge of a person having ordinary skill in the art. Alternatively, any need or problem known in the field at the time and addressed by the patent may provide a reason for combining elements of the prior art. I also understand that when there is a design need or market pressure, and there are a finite number of predictable solutions, persons of ordinary skill may be motivated to apply both their skill and common sense in trying to combine the known options in order to solve the problem.

30. I understand that, when determining whether a claimed invention is obvious, any “secondary considerations” of non-obviousness identified by the patentee should also be considered. These secondary considerations can include:

- commercial success of the invention, causally related to the invention itself rather than to companion factors, such as advertising or attractive packaging;
- the invention taught away from the technical direction followed by those skilled in the art;
- a long-felt but unsatisfied need for the invention while the needed implementing arts and elements had long been available;

- the invention achieves results unexpected to those skilled in the art;
- copying of the invention by competitors as distinguished from their independent development;
- unsuccessful attempts by those skilled in the art to make the invention;
- acquiescence by the industry to the patent's validity by honoring the patent through taking licenses or not infringing the patent, or both; and
- skepticism, disbelief in or incredulity by those skilled in the art that the patentee's approach worked.

31. I understand that, for the above information to impact the obviousness of a patent claim, there must be a nexus between the alleged secondary considerations and the claims. Additionally, I understand that the burden of introducing evidence of secondary considerations generally is on the patent owner.

VII. LEVEL OF SKILL IN THE ART

32. I understand that invalidity is evaluated from the perspective of a person of ordinary skill in the art to which the patent pertains at the time of the invention. I have been asked to assume that the relevant time period for my analysis of the Family 9 Patents is April 12, 2006, unless stated otherwise herein.

33. I understand that the person of ordinary skill in the art is identified by considering factors such as the education level of those working in the field, the sophistication of the technology, the types of problems encountered in the art, the prior art solutions to those problems, and the speed at which innovations are made may help establish the level of skill in the art.

34. In my opinion, with regard to the Family 9 Patents, a person of ordinary skill in the art ("POSITA") would have had at least a bachelor's degree in electrical or computer engineering, or the equivalent, and at least 5-6 years of experience in telecommunications or a related field; a master's degree in electrical or computer engineering, or the equivalent, and at least 2-3 years of

experience in telecommunications or a related field; or a Ph.D. in electrical or computer engineering, or the equivalent, with at least 1-2 years of experience in telecommunications or a related field.

35. By at least April 12, 2006, I had qualifications that met or exceeded those of a POSITA.

36. Counsel for Nokia has informed me that TQ Delta has proposed the following definition for a POSITA: A person of ordinary skill in the art at or before the time of the inventions of the Patents-in-Suit addressed in this declaration “would have had a bachelor’s degree in electrical engineering with 2-3 years of experience in DSL communication systems. Cooklev Decl., dated March 14, 2022, ¶ 20. The opinions set forth in this report would not change under TQ Delta’s definition.

VIII. BACKGROUND OF THE TECHNOLOGY

A. Description and Timeline of Technologies in Patents at Issue

1. Multicarrier Transmission

37. Information (bits) are often communicated using sinusoids, waveforms that look like the sine and cosine functions. A sinusoid is sometimes categorized by its frequency expressed in units of Hertz (Hz), which indicates how many periods of the sinusoid are completed in one second. Figure 2 below shows a 1 Hz sine wave. Figure 3 below shows a 2 Hz sine wave, and Figure 4 shows a 10 Hz sine wave.

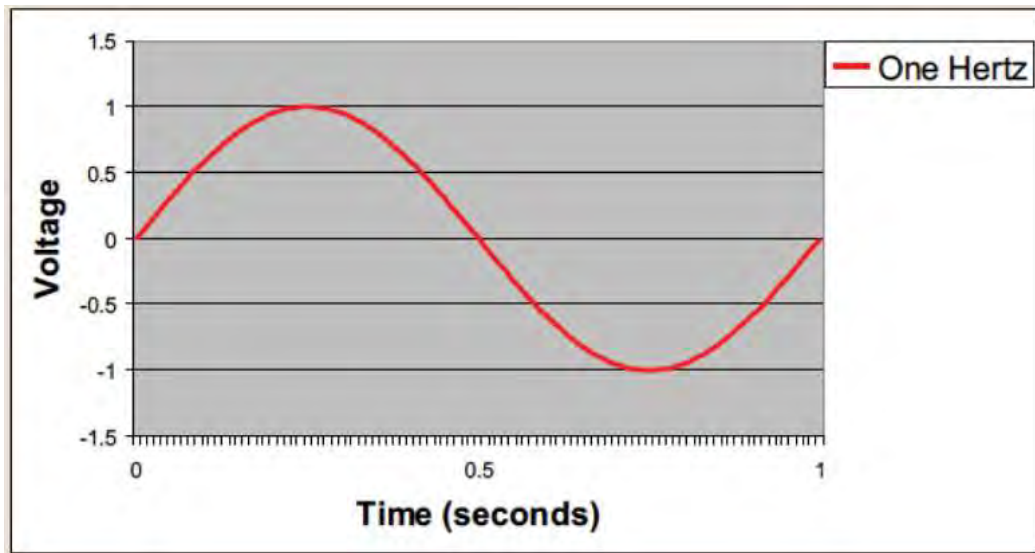


Figure 2: A one Hertz sine wave.

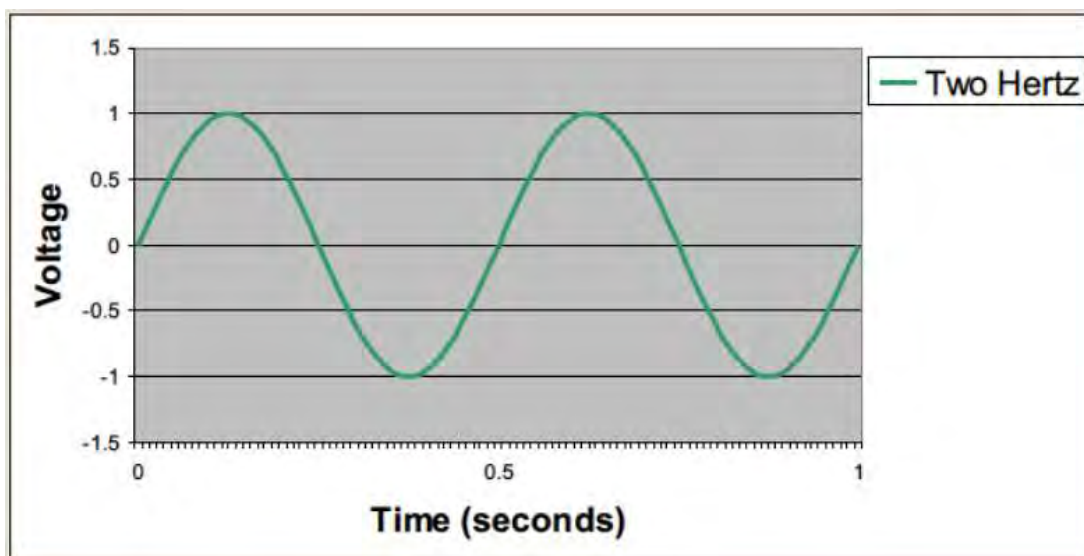


Figure 3: A two Hertz sine wave.

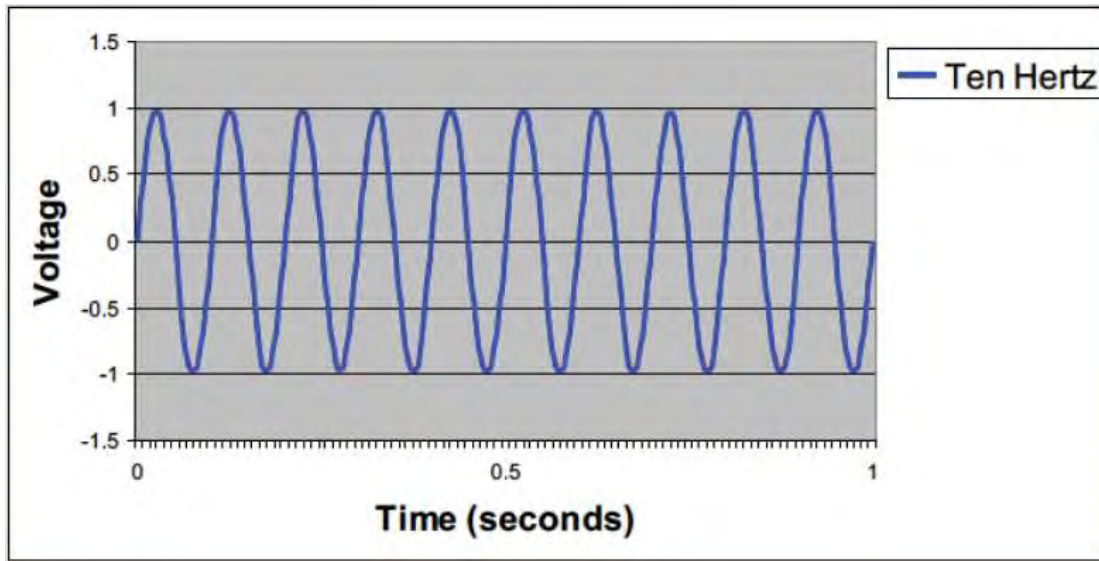


Figure 4: A ten Hertz sine wave.

38. Figure 5 shows one period of the periodic waveform produced by adding these three sinusoids. In 1807, Joseph Fourier proposed, and Dirichlet later proved rigorously, that an arbitrary periodic function can be represented by a sum of sinusoids. *See* the discussion on page 371 of *Circuits, Signals, and Systems* by William Siebert and the English translation of *Théorie Analytique de la Chaleur* by Joseph Fourier.

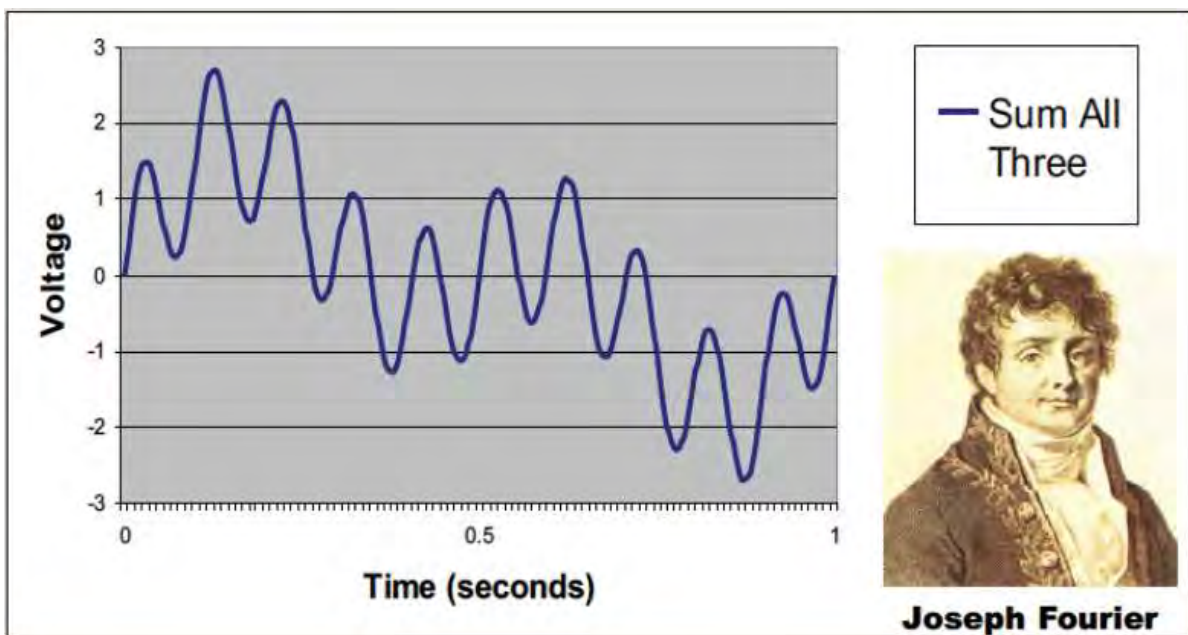


Figure 5: One period of the periodic waveform produced by adding a 1 Hz, 2Hz, and 10 Hz sine waves.

39. There are a variety of Fourier relationships or “Fourier Transforms” that convert a time domain signal to its representation as a sum of sinusoids and back. Of particular interest in this case is the discrete Fourier transform (DFT) where a periodic sequence that repeats every N samples is represented as the sum of N complex sinusoids $s_f(t) = \frac{1}{\sqrt{N}} e^{-\frac{j2\pi ft}{N}}$ where $f, t \in \{0, 1, \dots, N-1\}$. Note that $e^{-\frac{j2\pi ft}{N}} = \cos\left(\frac{2\pi ft}{N}\right) - j \sin\left(\frac{2\pi ft}{N}\right)$. Thus, the time signal $x(t)$ and its frequency representation $X(f)$ have the following discrete Fourier transform relationship:

$$X(f) = \frac{1}{\sqrt{N}} \sum_{t=0}^{N-1} x(t) e^{-\frac{j2\pi ft}{N}}$$

$$x(t) = \frac{1}{\sqrt{N}} \sum_{f=0}^{N-1} X(f) e^{\frac{j2\pi ft}{N}}$$

40. Waveforms that carry information are called “carriers.” Through the process of modulation, a sine wave that is a carrier can carry information through its phase and amplitude. The discrete phases and amplitudes that communicate each possible bit pattern are represented by a constellation. For example, an 8 pulse amplitude modulation (8-PAM) constellation is shown below in Figure 6 with a binary reflected Gray code labeling. Set-partitioning is another strategy for labeling a constellation.

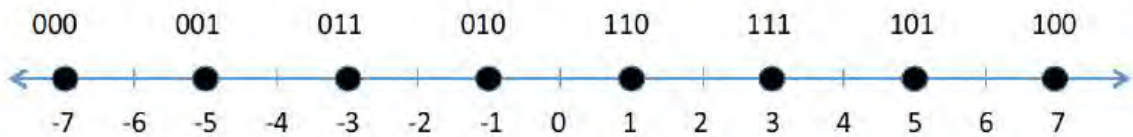


Figure 6: A constellation showing binary reflected Gray labeling for 8 pulse amplitude modulation (8-PAM).

41. Figure 7 below illustrates how a sinusoid can be used as the carrier wave to communicate information from the 8-PAM constellation of Figure 6. The top subfigure shows the carrier wave, which is a 16 Hz sine function. The middle subfigure shows the sequence of levels (-7, -3, 1, 5, -5, -1, 7, 3), which communicates the bits (000, 011, 110, 101, 001, 010, 100, 111) using the 8-PAM constellation with the labeling of Figure 6. The bottom subfigure shows the modulated single-carrier waveform produced by multiplying the waveforms in the top two subfigures.

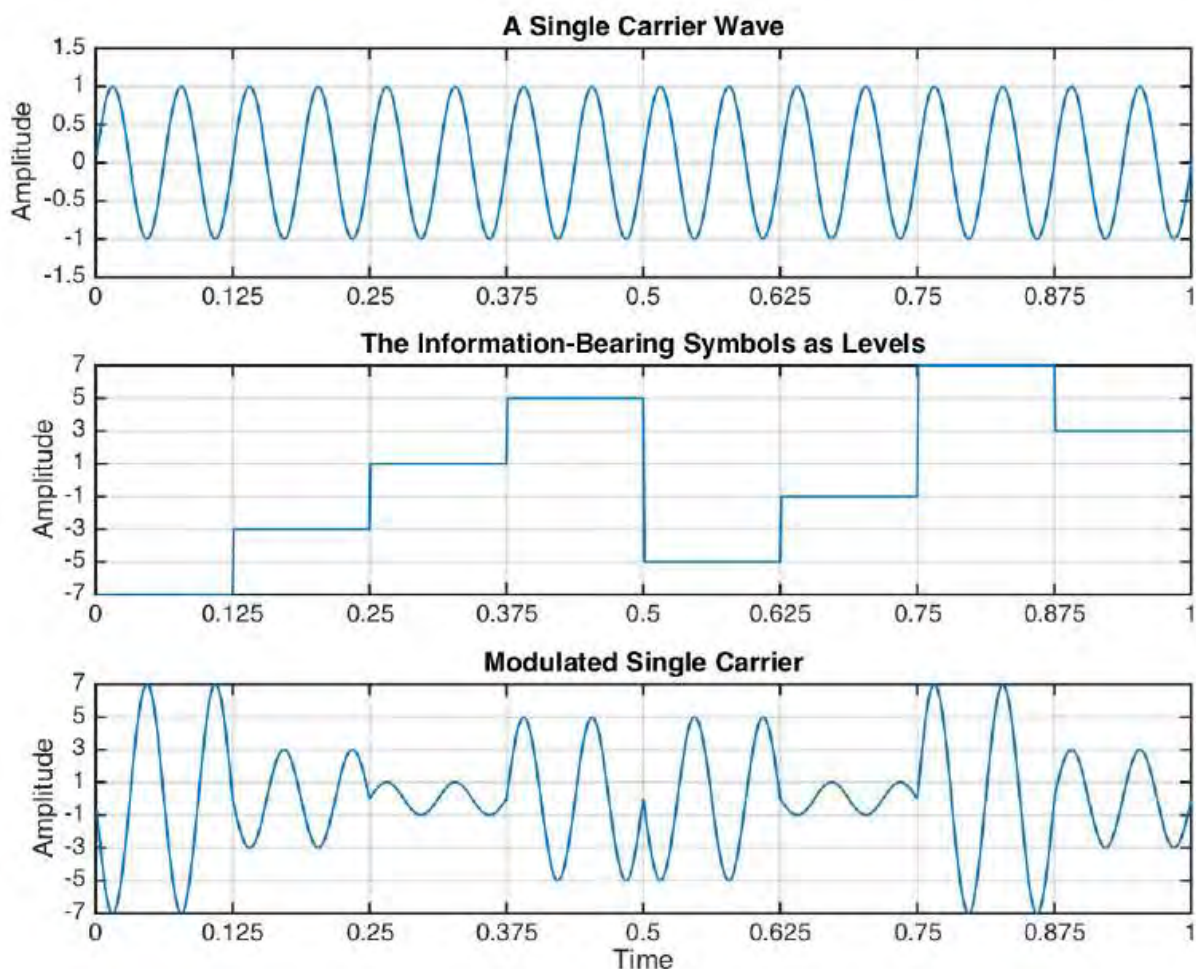


Figure 7: This figure illustrates single-carrier modulation. The top subfigure shows the carrier wave, which is a 16 Hz sine function. The middle subfigure shows the sequence of levels (-7, -3, 1, 5, -5, -1, 7, 3), which communicates (000, 011, 110, 101, 001, 010, 100, 111) using the 8-PAM constellation with the labeling of Figure 6. The bottom subfigure shows the modulated single-carrier waveform produced by multiplying the waveforms in the top two subfigures.

42. Notice that with single-carrier modulation each transmitted constellation point has its own symbol period where its associated level is the amplitude of the carrier wave. An alternative to single-carrier modulation is multicarrier modulation. Figure 8 below illustrates how multicarrier modulation can be used to communicate the same bit sequence as the example shown in Figure 7. The top eight subfigures each show a modulated sub-carrier sinusoid, modulated by the levels (-7, -3, 1, 5, -5, -1, 7, 3), which communicate (000, 011, 110, 101, 001, 010, 100, 111) respectively, using the 8-PAM constellation with the labeling of Figure 6. These eight sub-carriers have frequencies of 4 Hz, 8 Hz, 12 Hz, 16 Hz, 20 Hz, 24 Hz, 28 Hz, and 32 Hz. The bottom subfigure shows the modulated multicarrier waveform produced by adding the waveforms in the top eight subfigures.

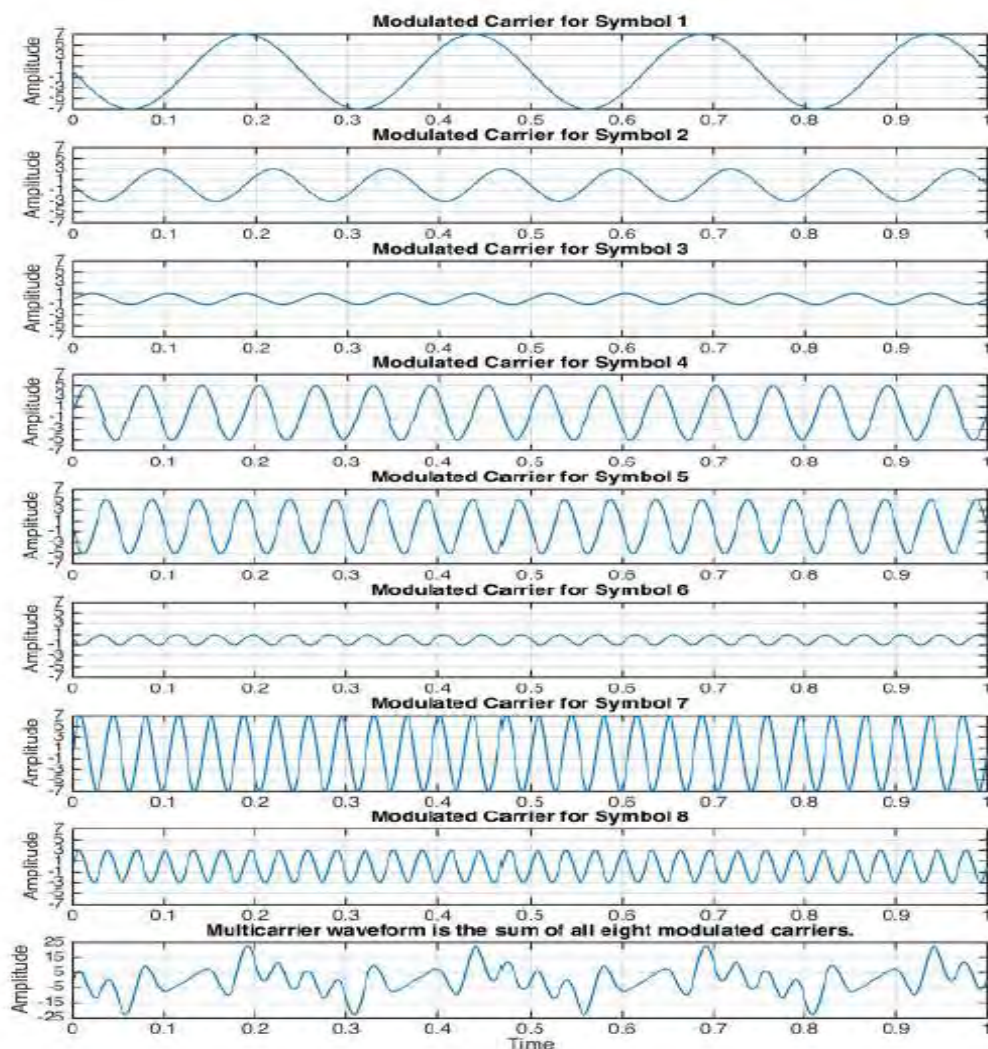


Figure 8: This figure illustrates multicarrier modulation. The top eight subfigures each show a modulated sub-carrier sinusoid, modulated by the levels $(-7, -3, 1, 5, -5, -1, 7, 3)$, which communicate $(000, 011, 110, 101, 001, 010, 100, 111)$ respectively, using the 8-PAM constellation with the labeling of Figure 6. These eight sub-carriers have frequencies of 4 Hz, 8 Hz, 12 Hz, 16 Hz, 20 Hz, 24 Hz, 28 Hz, and 32 Hz. The bottom subfigure shows the modulated multicarrier waveform produced by adding the waveforms in the top eight subfigures.

43. One way to think about the difference between multicarrier transmission and single-carrier transmission is that single-carrier transmission is analogous to a solo musician playing a melody on their instrument while multicarrier transmission would be analogous to an orchestra where the music from many individual instruments is transmitted simultaneously to the audience.

During a symphony performance one might hear multiple melodies simultaneously and the human brain can focus on one of the melodies and tune out the others. With multicarrier transmission, there may be 64, 256, or 1024 “melodies” and the discrete Fourier transform is able to isolate the information from each “melody,” which is a sinusoidal carrier. Unlike a human ear and brain, a multicarrier transceiver can receive and distinguish the information from each of these “melodies” at the same time.

44. Another way to think about the difference between multicarrier transmission and single-carrier transmission is to think of single carrier transmission as a single-lane road in which one information-carrying vehicle at a time can move from the transmitter to the receiver. In contrast, multicarrier transmission is like a multi-lane highway where each sinusoidal carrier provides a separate “lane” on which information can travel.

45. One can think of the bandwidth of a communication channel as the overall width of roadway that can be used for either one very wide “lane” or multiple more narrow “lanes”. In this way one can understand that with a fixed amount of bandwidth, the total amount of information throughput with single-carrier and multicarrier is actually very similar. With single-carrier, the entire bandwidth is devoted to that one carrier. It is as if an extremely wide vehicle is traveling down the road with its “wide load” of information. In contrast, with multicarrier there can be many more “lanes” but since the lanes are more narrow, the overall information throughput is the same since both transportation systems are limited by the overall width of the roadway.

46. A distortion phenomenon called inter-symbol interference (ISI) causes multiple copies of transmitted symbols to arrive at the receiver at different times. This causes the received signal to include components of multiple symbols added together. ISI occurs in both wireless and wired communication systems and in both point-to-point and multi-cast or broadcast applications.

ISI can be mitigated by equalization in the time domain, which seeks to recover the original transmitted symbols from the received combinations.

47. ISI may also be understood through its effects at different frequencies; in the frequency domain ISI induces a frequency-selective gain. Multicarrier transmission overcomes ISI by transmitting on multiple parallel sub-carriers with relatively narrow effective bandwidth. Each sub-carrier typically experiences a different gain because of frequency selectivity, but each sub-carrier experiences negligible or significantly reduced inter-symbol interference because the sub-carriers are narrowband relative to the overall bandwidth.

48. Multicarrier transmission was a standard tool in communications at the time of the alleged inventions. The Kineplex modem developed by the Collins Radio Company employed multicarrier transmission no later than 1957. As described on page 657 of the 1957 IRE journal paper “Binary Data Transmission Techniques for Linear Systems,” by Doelz, Heald, and Martin, “Data is transmitted by means of multiple tones in the voice band...” and specifically “[t]wenty tones, at different frequencies, are multiplexed to provide the 3000 bit per second capacity.”

49. Multicarrier transmission was studied in a variety of research publications in the 1960s including the following sampling of articles: The technical report “Digital communication over fixed time-continuous channels with memory, with special applications to telephone channels,” by Holsinger (1964) laid out the theoretical framework for multicarrier transmission as a capacity-achieving approach. The journal paper “The AN//GSVC-10 (KATHRYN) Variable Rate Data Modem for HF Radio” by Zimmerman and Kirsch (1967) described an implemented multicarrier system. The journal paper “Performance of an efficient, parallel data transmission system,” by Salzberg (1967), and his related U. S. Patent No. 3,511,936 entitled “Multiply Orthogonal System for Transmitting Data Signals Through Frequency Overlapping Channels,”

filed in May 1967, described a multicarrier transmission system. The journal paper “A theoretical study of performance of an orthogonal multiplexing data transmission scheme,” by Chang and Gibby (1968) presents a theoretical analysis of multicarrier transmission that investigates practical problems such as sampling time error, carrier phase offset, and non-ideal phase characteristics.

50. The idea of implementing multicarrier transmission using the discrete Fourier Transform (DFT) was understood in the literature by 1969 at the latest. Salz and Weinstein presented this idea in their 1969 conference paper at ACM entitled “Fourier Transform Communication System.” Subsequently, Weinstein and Ebert elaborated on the scheme in the 1971 *IEEE Transactions on Communication Technology* journal paper “Data Transmission by Frequency-Division Multiplexing Using the Discrete Fourier Transform.”

51. The benefit of converting an ISI channel into parallel, independent ISI-free channels using the DFT applies both to point-to-point scenarios and to multi-cast or broadcast scenarios. This would have been well understood by a POSITA long before the time of the alleged inventions. Although the fundamental transmitter and receiver functionality is very similar, these two use cases have different acronyms. The use of DFT-based multicarrier transmission for point-to-point transmission is sometimes called discrete multi-tone (DMT). The use of DFT-based multicarrier transmission for broadcast or multicast transmission is sometimes called Orthogonal Frequency-Division Multiplexing (OFDM).

52. In the years between 1987 and 1991 papers were written describing both OFDM and DMT systems. In 1987, Alard and Lasalle described an OFDM system in their EBU paper “Principles of modulation and channel coding for digital broadcasting for mobile receivers.” In 1989-1991 John Cioffi and his students at Stanford University published a series of papers described applying DFT-based multicarrier transmission to the point-to-point transmission problem

of transmitting data between a central office and a home over telephone lines called digital subscriber lines (DSLs). These include the conference papers “Crosstalk-limited performance of a computationally efficient multichannel transceiver for high rate digital subscriber lines” by Tu, Chow, Dudevoir, and Cioffi, presented at the 1989 IEEE Global Telecommunications Conference, and “A computationally efficient adaptive transceiver for high-speed digital subscriber lines” by Chow, Tu and Cioffi and presented at the 1990 IEEE International Conference on Communications.

53. The journal paper “Performance evaluation of a multichannel transceiver system for ADSL and VHDSL services” by Chow, Tu, and Cioffi in the *IEEE Journal on Selected Areas in Communications* (JSAC) in 1991 and “A Multicarrier Primer,” by John Cioffi in 1991 described the DMT approach for DSLs which eventually evolved into the ADSL and VDSL standards. The Chow et al. JSAC article reiterates the basic advantage of the multicarrier approach on page 913: “The fundamental concept of multicarrier modulation is the conversion of a data transmission channel with inter-symbol interference (ISI), and possibly crosstalk and/or colored noise, into a set of parallel, independent and ISI-free subchannels.”

54. Additional papers on DMT and OFDM published in 1995 will be of interest in the discussions below. In 1995, Sari, Karam, and Jeanclaude explored OFDM for television broadcast in “Transmission techniques for digital terrestrial TV broadcasting,” in the *IEEE Communications Magazine*. Also in 1995 I wrote a paper along with Cioffi entitled “Fundamentals of coding for broadcast OFDM” at the 1995 Asilomar conference which explored coding and interleaving (two topics discussed below) for OFDM. For DMT systems Zogakis and Aslanis published “A Coded and Shaped Discrete Multitone System” in the *IEEE Transactions on Communications* in December of 1995, which presents a DMT system that employs Reed-Solomon coding and interleaving.

2. Reed-Solomon Coding

55. Bits are often communicated over channels as the labels of constellation points in Euclidean space (e.g., the coordinates $x = 1, y = 1$). When such bits are transmitted over noisy channels, the constellation points can be distorted introducing bit errors as shown below in Figure 9. Error control coding uses extra bits or symbols (sometimes called redundancy) to correct errors that may occur during transmission. The addition of the extra redundancy symbols can lower the probability of a message an error.

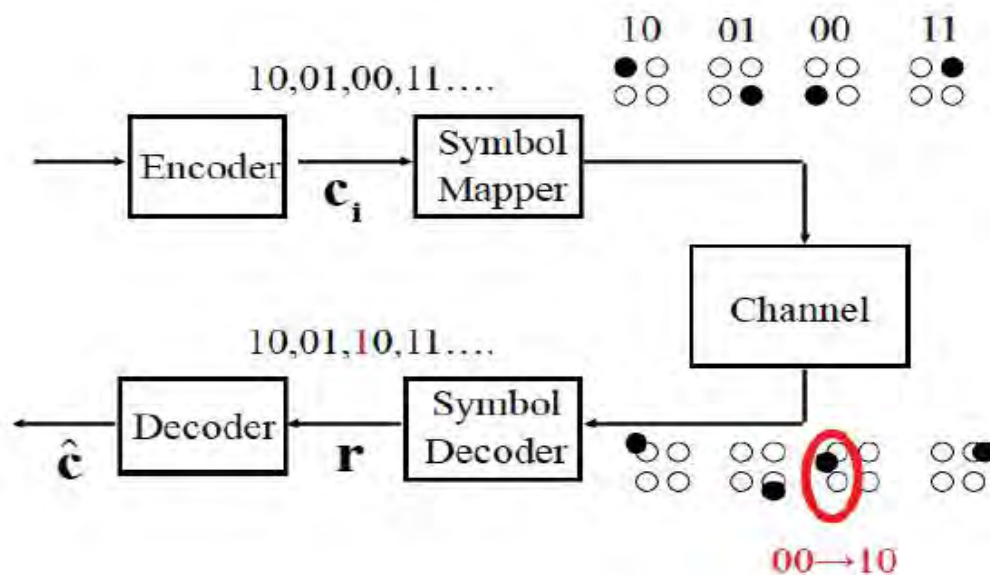


Figure 10: An error control encoder followed by a symbol mapper that maps bits to constellation points. Noise in the channel distorts the constellation points leading to possible bit errors.

56. A simple example of how this works is the three-bit repetition code. Consider a bit-error channel (often called the binary symmetric channel) that has a bit error probability of 10^{-2} (or one in one hundred). If the transmitter sends a one-bit message on this channel, the receiver will observe an incorrect message with a probability of 10^{-2} (that is, about one out of every hundred one-bit messages will be observed incorrectly). Suppose that instead of sending a single bit, the transmitter creates a codeword by repeating the message bit three times. When the receiver observes

the three bits of the codeword, the receiver can select the message corresponding to the majority vote of the three observed bits. The extra two bits of redundancy reduce the probability of a message error significantly, from 0.01 to about 0.0003 (a 1% chance versus an ~.03% chance of an error).

57. Reed-Solomon codes are a particularly powerful form of error control coding that is optimal in the sense that Reed-Solomon codes achieve the Singleton bound. The Singleton bound involves Hamming distance. The Hamming distance between two codewords is simply the number of symbol positions where the codewords have different symbols. The Singleton bound says that for a code where each codeword has N symbols of redundancy, the minimum Hamming distance between codewords cannot be more than $N + 1$. Our simple example of a 3-bit repetition code achieves the Singleton bound by achieving a Hamming distance of 3 with redundancy $N = 2$. All Reed-Solomon codes achieve the Singleton bound, as stated and proved on page 175 of the classic 1983 textbook *Theory and practice of Error Control Codes* by Richard E. Blahut. That is, every Reed-Solomon code achieves the Hamming distance of at least $r + 1$ between every two codewords.

58. Error control coding in general, and Reed-Solomon coding in particular, was a standard tool in communications at the time of the alleged invention. For example, in the 1967 article describing the AN/GSC-10 KATHRYN multicarrier transmission system, the authors state that “[t]he AN/GSC-10 is compatible with error-correcting coding” without specifying the specific error control code. The AN/GSC-10 (KATHRYN) Variable Rate Data Modem for HF Radio at page 198. Reed-Solomon codes would have been one choice that a POSITA would have considered even in 1967.

59. As noted in Blahut’s 1983 textbook on page 206:

NOTES

The BCH codes were discovered independently by Hocquenghem (1959) and by Bose and Ray-Chaudhuri (1960). It was noticed quickly by Gorenstein and Zierler (1961) that the codes discovered earlier by Reed and Solomon (1960) are a special case of nonbinary BCH codes.

60. The Gorenstein and Zierler 1961 article describes a decoder for BCH codes generally, and they also note that the codes discovered earlier by Reed and Solomon (1960) are a special case of non-binary BCH codes. An important improvement to the decoding procedure for BCH codes was identified for example by Berlekamp (1968) and re-interpreted by Massey (1969).

61. Thus Reed-Solomon codes were invented no later than 1960 (*see* I. S. Reed and G. Solomon, Polynomial Codes over certain Finite Fields, *J. Soc. Indust. Appl. Math*, 8, (1960): 300-304). Well before the alleged invention of the asserted patents, Reed-Solomon codes were in common use. Blahut referred to them in his 1983 textbook as “An important and popular subset of the set of BCH codes.” *See* Blahut 1983 textbook at page 174.

62. Reed-Solomon coding is a general purpose error control technique that can protect information in a wide variety of applications. Because they achieve the Singleton bound and had well-developed algebraic decoding algorithms, Reed-Solomon codes had already found many applications, including multicarrier transmission, by the time of the alleged invention.

63. As pointed out by Stephen B. Wicker in his 1995 book *Error Control Systems for Digital Communication and Storage* (*see* page 14), “The non-binary Reed-Solomon codes are, quite simply, ubiquitous. A shortened pair of cross-interleaved Reed-Solomon codes provides error control for the compact digital audio disc (CD). Another Reed-Solomon code was used for error

control during the Voyager exploration of the outer solar system.” Dr. Wicker goes on to say that “Reed-Solomon codes are easily tailored to meet the needs of a particular application.”

64. In the context of the state of the art as understood by a POSITA, the use of Reed-Solomon codes in various communication systems at the time of the alleged invention, including a multicarrier modulation system with interleaving and retransmission, would be an obvious choice to consider. In those cases where it had not already been explicitly done, it would certainly be applying a known improvement to achieve predictable results.

65. The use of Reed-Solomon coding in multicarrier transmission was certainly well-known by 1995. For a DMT, an example is illustrated in Figure 9 from page 2948 of the *IEEE Transactions on Communications* article, “A Coded and Shaped Discrete Multitone System” by Zogakis and Aslanis from December of 1995, which is shown in Figure 11 below. In such a system, transmitted packets would contain one or more Reed-Solomon codewords.

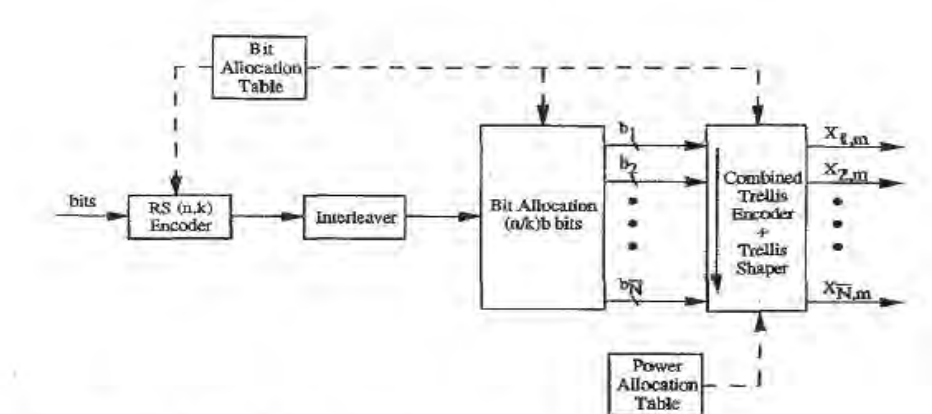


Fig. 9. Coded and shaped multicarrier encoder.

Figure 11: In this figure, which is Figure 9 from the December 1995 Zogakis journal paper, we see an example of a system that used Reed-Solomon coding, interleaving, and multicarrier transmission.

66. The use of Reed-Solomon coding in an OFDM, i.e., broadcast or multicast, scenario is illustrated in Figure 14 from page 180 of the EBU article, “Principles of modulation and channel coding for digital broadcasting for mobile receivers” by Alard and Lasalle from 1987, which is

shown in Figure 12 below. In this figure, the “Block Code” is described on pages 179-180 as “an external Reed-Solomon code, as shown in the schematic diagram of Fig. 14.” The article considers both the standard Reed-Solomon code and cyclotomically shortened Reed-Solomon (CSRS) codes. In either case, transmitted packets would include one or more Reed-Solomon codewords.

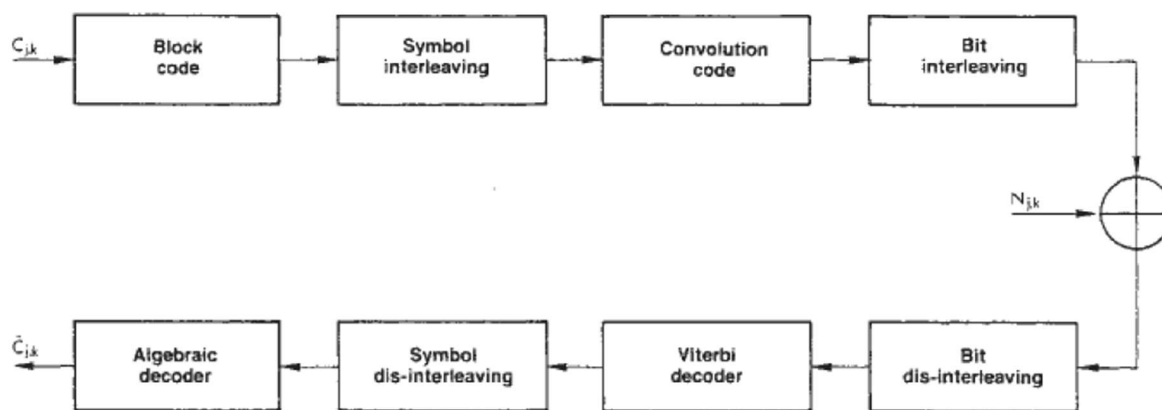


Figure 14
Principle of concatenated codes

Figure 12: In this figure, which is Figure 14 from the 1987 Alard and Lasalle journal paper, we see an example of a system that used Reed-Solomon coding, interleaving, and multicarrier transmission.

67. The use of Reed-Solomon coding in an OFDM, i.e., a broadcast or multicast, scenario was also considered in my 2000 journal paper with Liu and Shi entitled “Trellis Codes for Periodic Erasures,” where we designed new trellis codes for interleaved multicarrier transmissions and compared to a Reed-Solomon encoded, interleaved multicarrier transmission.

68. For the Reed-Solomon encoded case, transmitted packets would include one or more Reed-Solomon codewords.

69. While Reed-Solomon coding was, and is, a common technique for error control coding used in a multicarrier transmission system, there are numerous alternatives for providing effective error control for multicarrier transmission. Popular alternatives to Reed Solomon codes include low-density parity-check (LDPC) codes, turbo codes, polar codes, convolutional codes,

trellis codes, Reed-Muller codes, Golay codes, BCH codes that are not Reed-Solomon codes, and others. As discussed in the previous paragraph, I have personally designed trellis codes that, when utilizing soft Viterbi decoding, outperform Reed-Solomon codes in a multicarrier setting in certain scenarios.

70. Another example of alternative coding used with multi-carrier transmission is provided by the 1999 802.11a standard and specifically in the document *IEEE Standard 802.11a – 1999 (R2003), Part 11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications, High-speed Physical Layer in the 5 GHz Band*. This document is available online at <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=815305>. Figure 13 below shows the general block diagram of the transmitter and receiver for 802.11a.

17.3.8.1 Outline description

The general block diagram of the transmitter and receiver for the OFDM PHY is shown in Figure 118. Major specifications for the OFDM PHY are listed in Table 86.

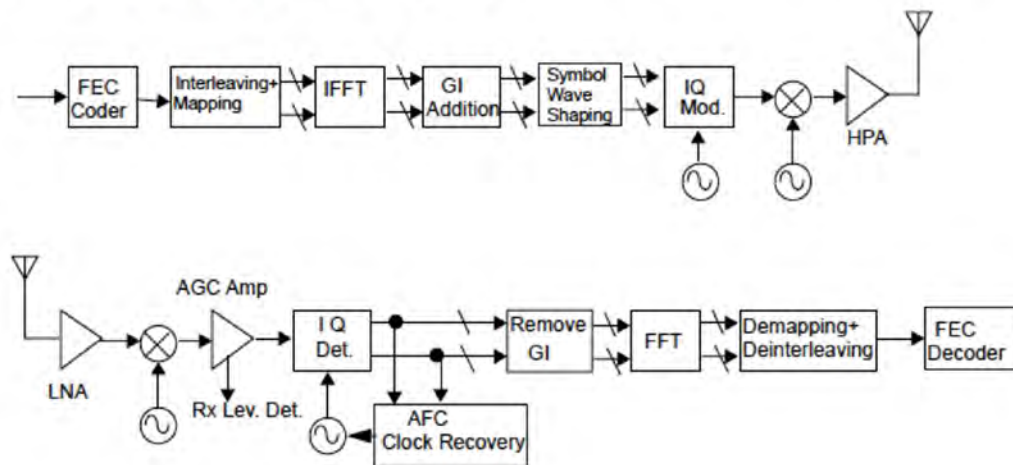


Figure 13: This figure, which is Figure 118 from the 1999 802.11a standard, provides an example of coding, interleaving, and multicarrier transmission where the forward error correction (FEC) code is a convolutional code rather than a Reed-Solomon code.

71. As we can see, 802.11a uses error control coding (the forward error correction (FEC) Coder transmitter module), interleaving (the Interleaving+Mapping transmitter module) and

multicarrier transmission (the IFFT transmitter module). The error control coding used in 802.11a employs convolutional codes rather than a Reed-Solomon code. This is shown in Table 1 below.

Table 86—Major parameters of the OFDM PHY

Information data rate	6, 9, 12, 18, 24, 36, 48 and 54 Mbit/s (6, 12 and 24 Mbit/s are mandatory)
Modulation	BPSK OFDM QPSK OFDM 16-QAM OFDM 64-QAM OFDM
Error correcting code	K = 7 (64 states) convolutional code
Coding rate	1/2, 2/3, 3/4
Number of subcarriers	52
OFDM symbol duration	4.0 μ s
Guard interval	0.8 μ s ^a (T_{GI})
Occupied bandwidth	16.6 MHz

^aRefer to 17.3.2.4.

Table 1: This table, which is Table 86 from the 1999 802.11a standard, provides information about the data rates, modulations, error correction codes, coding rates, and other parameters of the 802.11a physical layer.

72. Note that three convolutional code rates are possible, 1/2, 2/3, and 3/4. Also, note that four modulations are possible, BPSK, QPSK, 16-QAM, and 64-QAM. A configuration message is required so that the transmitter can communicate to the receiver which convolutional code rate and which modulation are being used for a particular transmission, as will be discussed further below.

73. It is a standard part of the design of a communications system to select the error control coding after evaluating some options. Table 7.5.2 from Section 7.5.2.2 of TR 25.896 lists some of the options for error control coding considered for uplink signaling. Note that both convolutional codes (CCs) and block codes are under consideration as well as repetition and hybrid techniques.

7.5.2.2 Coding, multiplexing and mapping options

Table 7.5.2 provides a list of parameters and possible options related to coding, multiplexing and mapping of the UL signalling. All of these options can be combined and the value of each combination could be evaluated. Some of the resulting combinations are described and evaluated in section 9.2.4

Table 7.5.2: UL signalling coding, multiplexing and mapping options

Parameter	Options	Comments
TTI	10 ms / 2 ms	2 ms is to illustrate a shorter TTI value
Timing relative to E-DCH transmission	Aligned / prior	Transmission prior to E-DCH transmission provides possibility for early decoding procedure, but implies longer HARQ delays or reduced processing time in the UE
CRC	Yes / no	
Coding	CC / Block / Repetition / Hybrid	Repetition coding allows for early UL signalling decoding and may reduce buffering requirement in the Node B
Mapping	(E-)DPDCH / DPCCH / E-DPCCH	(E-)DPDCH and DPCCH mapping are described in section 7.5.2.2.1 and 7.5.2.2.2 respectively. Mapping the control bits on a new code channel is described in section 7.5.2.2.3; using SF=256 allows mapping of UL signalling bits with either 2 ms or 10 ms TTI.

Table 2: This is Table 7.5.2 from 3GPP TR 25.896 at NOK00074958 provides a list of parameters and possible options related to coding, multiplexing, and mapping of the uplink (UL) signaling.

3. Interleaving

74. Interleaving re-orders (permutes) symbols so that impairments that impact consecutive transmitted symbols in time or frequency will be spread out as seen by a decoder such as a Reed-Solomon decoder.

75. Interleaving was another standard tool in communications at the time of the alleged invention. In his 1971 journal paper in the *IEEE Transactions on Communications*, entitled “Burst-Correcting Codes for the Classic Bursty Channel,” G. David Forney observes on page 772 that:

The use of interleaving to adapt random-error correcting codes to bursty channels is frequently proposed...

76. Forney later discusses block interleaving under the heading of “Archetypal Coding Schemes” on page 775 stating:

The usual type of interleaver is a block interleaver, in which, for example, bits are laid down in the rows of a $B \times N$ matrix, and read out from the columns. In this paper we shall use a somewhat simpler and more effective type of interleaver, which we have called [11] a periodic (or convolutional) interleaver. (Similar interleavers were independently proposed by Cowell and Burton [12] and Ramsey [13])

77. In the quotation above, reference [12] is the 1962 paper “Computer Simulation of the use of Group Codes with Retransmission on a Gilbert Burst Channel” by W. R. Cowell and H. O. Burton published in the journal *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics*. This paper shows that interleaving was used in conjunction with retransmission by 1962 at the latest.

78. For block interleaving the bits or symbols are written into a matrix (described as having B rows and N columns in the quote by Forney above. A typical implementation of such an interleaver would require enough memory to store the entire $B \times N$ matrix of bits or symbols. For convolutional interleaving, the two key parameters are the interleaver blocklength I and the interleaver depth (or delay increment) D . For the typical triangular implementation as shown below in Figure 8-2 from the VDSL1 standard with D chosen as a multiple of the interleaver blocklength I , i.e., $D - 1 = M \times I$, the equation $M \times I \times (I - 1)/2$ computes the interleaver/deinterleaver memory requirement.

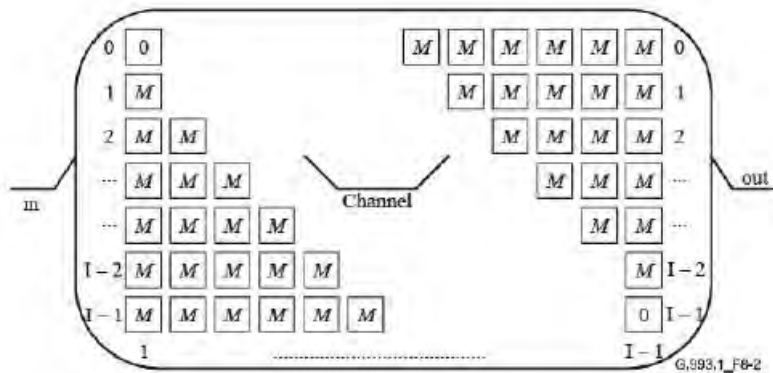


Figure 8-2/G.993.1 – Implementation example with $D - 1 = M \times I$ and $I = 7$

VDSL1 at NOK00078786, § 8.4.2.

79. As stated on the first page of the Cowell and Burton article (page 577), they briefly describe the principle of using interleaving in conjunction with error control coding:

Interleaving

If errors occur in bursts, codes may be designed that are particularly suited to detecting or correcting this type of error pattern or the code words may be transmitted in an interleaved fashion so that digits of a code word are separated by digits of other code words. Thus, the effect of a burst is spread over several words and is less likely to exceed the error detecting or correcting capabilities of any one word.

80. In the quotation above, reference [13] is the 1970 paper by J. Ramsey entitled “Realization of Optimum Interleavers,” which appeared in the *IEEE Transactions on Information Theory*. Ramsey gives an excellent general description of an interleaver in the first sentence of his article as follows: “An interleaver is a device that rearranges the ordering of a sequence of symbols

in some one-to-one deterministic manner.” Ramsey went on to describe how interleavers are often used in communication systems:

In many of the applications to communication technology, interleaving is used as an adjunct to coding for error correction. One technique, which is useful for some types of burst-error channels, is to insert an interleaver between the channel encoder and the channel. The interleaver redistributes the channel symbols so that the symbols from a codeword are mutually separated by somewhat more than the length of a “typical” burst of errors. Thus, interleaving effectively makes the channel appear like a random-error channel to the decoder. For some HF channels, this technique can improve the performance by one to three orders of magnitude [1].

81. The combination of multicarrier transmission, Reed-Solomon coding, and interleaving was well-known certainly by 1995. For a point-to-point example, an example is illustrated in Figure 9 from page 2948 of the *IEEE Transactions on Communications* article, “A Coded and Shaped Discrete Multitone System” by Zogakis and Aslanis from December of 1995. This figure is reproduced above as Figure 11. In this figure the interleaver is shown after the Reed-Solomon Encoder, and the symbols at the far right of the figure are passed on to a multicarrier transmitter.

82. The use of interleaving in an OFDM, i.e., broadcast or multicast, scenario is illustrated in Figure 14 from page 180 of the EBU article, “Principles of modulation and channel coding for digital broadcasting for mobile receivers,” by Alard and Lasalle from 1987, which was shown above as Figure 12. In this figure, interleaving and de-interleaving of the symbols is clearly indicated. The discussion on page 178 in the section 5.2 entitled “Use of an interleaving arrangement” describes both “frequency interleaving,” and “time interleaving” and notes that

frequency interleaving, i.e., the interleaving of multicarrier symbols, is “essential for fixed reception or “stationary mobile.”

- interleaving in time, which is very useful for mobile reception, even at slow speed. This interleaving can be very deep (several hundreds of milliseconds), because the constraints at the level of processing delay in the receiver are much less severe in digital sound broadcasting than in radiotelephony;
- frequency interleaving, which is essential for fixed reception or “stationary mobile”. Note that with OFDM the number of carriers per programme can be very large. In fact, the real independence of these carriers depends in particular on the product of the bandwidth used W and the standard deviation of the distribution of the delays. From this point of view, the existence of multiple paths can be considered as an advantage.

83. There is considerable discussion in the literature specifically describing interleaving in conjunction with multicarrier transmission and coding such as Reed-Solomon coding as an option for OFDM. In the Alard and Lasalle 1987 article, Alard and Lasalle state on page 178 that interleaving allows Reed-Solomon codes to simultaneously exploit the “two essential particularities” of the Rayleigh channel they are studying, “these two properties to be exploited simultaneously by associating an interleaving system matched to the length of the code symbols and a system for deleting the affected symbols in the case of a deep fade.” Ultimately, the article suggests using interleaved Reed-Solomon codes used in conjunction with convolutional codes.

84. The article by Sari et al. in 1995 states in its conclusions on page 108, “With coding, interleaving, and weighted decoding, OFDM signaling eventually surpasses the performance of single-carrier transmission...” While pointing out that OFDM has strong sensitivity to nonlinear

distortion and carrier synchronization difficulties, the Sari et al. article concludes that OFDM with coding and interleaving was a strong potential technique for digital terrestrial TV broadcasting. Indeed, from this article a POSITA would conclude that OFDM is only a strong contender when used with coding and interleaving. My own paper at Asilomar studying OFDM in 1995 focused on developing improved approaches for OFDM only in the context of coding and interleaving, never even considering OFDM without both coding and interleaving.

85. Thus certainly by 1995 it was well understood that a multi-cast system using multicarrier transmission would typically use coding and interleaving to perform well. I also pointed out the natural application of interleaving to multicarrier transmission in my 2000 journal paper with Liu and Shi entitled “Trellis Codes for Periodic Erasures,” where I began that paper with the following statement:

Partial-band interference often distorts frequency-hopped or multicarrier transmissions. This interference might be caused by a malicious jammer in a military communications scenario or by adjacent channel interference in a multicarrier digital audio broadcast scenario. In both cases, the partial-band interference is contiguous and may be transformed to a periodic erasure pattern by a block interleaver. This paper treats the design and analysis of trellis codes specifically for such periodic erasure scenarios.

86. While the focus of my 2000 paper was on trellis codes, the idea of using Reed-Solomon codes in conjunction with interleaving and multicarrier transmission was so well-fixed at that time that I used that configuration as a baseline for comparison. As I stated in the introduction, “Section II also includes a motivational comparison between a trellis code designed according to this paper and a Reed-Solomon (RS) code of comparable rate.” Thus, six years prior to the alleged inventions, the idea of using Reed-Solomon codes, interleaving, and multicarrier transmission together in a system was considered a standard approach. As we have seen, several such systems were discussed in the literature.

4. Retransmission

87. Retransmission is often referred to as an automatic-repeat-request (ARQ) protocol. Whenever retransmission is used, a message on the reverse channel must be sent to enable retransmission; any transmitter that uses retransmission must receive a message from the receiver informing it whether it successfully received the packet (an acknowledgement or ACK) or not (a negative acknowledgement or NACK) so that the transmitter knows whether or not retransmission is needed.

88. As pointed out by Dr. Wicker in Error Control Systems for Digital Communication and Storage, retransmission is perhaps the simplest form of error control for channels with feedback:

At its simplest, error control for channels with feedback takes the form of the pure automatic-repeat-request (ARQ) protocol. In these protocols the transmitted data is encoded for error detection; detected errors at the receiver result in the generation of a retransmission request.

Stephen Wicker, Error Control Systems for Digital Communication and Storage at page 393.

89. Retransmission is another standard tool in communications at the time of the alleged invention. The simplest form of ARQ was certainly in use before 1960. Dr. Wicker discusses ARQ history on page 393 of his textbook:

It is difficult to date the earliest introduction of ARQ protocols in digital communications. They were almost certainly an immediate consequence of the development of parity-check codes in the early days of digital computers. The next major development in the area came in 1960, when Wozencraft and Horstein [Woz3],[Woz4] described and analyzed a system that allowed for both error correction and error detection with retransmission requests. Their system, now known as a type-I hybrid-AREQ protocol, provided significantly improved performance over the pure ARQ protocols in many applications.

90. In the quotation above, [Woz3] is the 1961 MIT Technical Report by Wozencraft and Horstein entitled “Coding for Two-Way Channels,” which provides an exploration of retransmission. In the abstract of the article, they state “The system is somewhat similar to human

communication, in that typical errors are corrected, while grievous ones initiate a request for retransmission.”

91. In the quotation above, [Woz4] is the 1960 conference presentation by Wozencraft and Horstein on which the MIT Technical Report was based. It was entitled “Digitalized Communication over Two-Way Channels,” which they presented at The Fourth London Symposium on Information Theory.

92. The notion that retransmission in communication systems is similar to human conversation is also stated in the 1961 *IRE Transactions on Information Theory* journal article on retransmission by W. M. Cowell “The use of Group codes in Error Detection and Retransmission.” Cowell states on page 168:

Many communication links on which such codes would be used permit the transfer of information in both directions so that it is possible, just as in human conversation, for the receiver to request the retransmission of messages or parts of messages in which errors are detected.

93. Later on the same page, Cowell describes how retransmission would be used in the particular system he was studying in 1961, “If a is in S , y is decoded to x . If a is not in S , the transmitter is instructed, via a reverse channel, to retransmit the code word.”

94. Soon after his 1961 paper on retransmission, Cowell used retransmission with interleaving, in the 1962 paper “Computer Simulation of the use of Group Codes with Retransmission on a Gilbert Burst Channel” by W. R. Cowell and H. O. Burton published in the journal *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics*. This paper shows that retransmission was used with interleaving by 1962 at the latest. The brief summary they provide of their paper describes how they used both retransmission and interleaving to address bursts of errors on the channel:

Summary: A study of error control by coding was made by Monte-Carlo simulation of a burst-noise channel on an IBM (International Business Machines) 7090. Using short group codes, comparisons were made between correction and detection with retransmission. Also, the effect of interleaving the code words was studied. Error detection with retransmission showed a consistently better performance than error correction and the time division resulting from interleaving was effective in combating burst type error patterns.

95. Their simulations were performed on an IBM 7090 computer, which is shown below:



Figure 14: In this figure, which is Figure 14 from the 1987 Alard and Lasalle journal paper, we see an example of a system that used Reed-Solomon coding, interleaving, and multicarrier transmission.

a. TCP/IP Retransmission

96. In 1974, Vinton Cerf and Robert Kahn published a paper entitled “A Protocol for Packet Network Intercommunication” in the *IEEE Transactions on Communications* journal. This paper proposed what Cerf and Kahn termed a “transmission control program” (TCP) to handle transmission and reception of messages on behalf of various processes. Cerf & Kahn 1974 Paper, page 4, col. 1 ¶ 3. Each TCP would communicate with multiple processes. *Id.* Cerf and Kahn’s TCP would break messages into segments for transmission, multiplex segments among the processes that it communicated with into network packets, and provide those network packets made of segments for delivery to packet switches, which would in turn send those packets onto the network for transmission to their destination. *Id.* ¶¶ 3-4. The TCP would also receive network packets from its packet switches, demultiplex the segments for different processes, and reassemble the segmented data before passing it up to the correct process. *Id.* TCP is now one of the foundational technologies underlying the Internet—specifically, the Internet uses TCP combined with “internet protocol” (IP), more commonly called TCP/IP.

97. TCP uses source and destination port addressing to ensure that segments are routed correctly. Cerf & Kahn 1974 Paper, page 5, col. 1 ¶ 5-col. 2 ¶ 1. The source and destination port addresses are the first two fields of a TCP packet. *Id.* page 6, col. 1 ¶ 1. Each packet includes a sequence number to allow segments to be reassembled at the destination port. *Id.* ¶ 3. Cerf & Kahn provide an example of how sequence numbers help with packet segmentation:

The original message A in Fig. 9 is shown split into two segments A1 and A2 and formatted by the TCP into a pair of internetwork packets. Packets A1 and A2 have their ES bits set, and A2 has its EM bit set as well. When packet A1 passes through the GATEWAY, it is split into two pieces: packet A11 for which neither EM nor ES bits are set, and packet A12 whose ES bit is set. Similarly, packet A2 is split such that the first piece, packet A21, has neither bit set, but packet A22 has both bits set. The sequence number field (SEQ) and the byte count field (CT) of each packet is modified by the GATEWAY to properly identify the text bytes of each packet. The GATEWAY need only examine the internetwork header to do fragmentation.

Id. page 7, col. 1 ¶ 3.

98. TCP uses a timeout and positive acknowledgment system to signal when a packet should be retransmitted. Cerf & Kahn 1974 Paper, page 7, col. 1 ¶ 5. The system is described in detail:

No transmission can be 100 percent reliable. We propose a timeout and positive acknowledgement mechanism which will allow TCP's to recover from packet losses from one HOST to another. A TCP transmits packets and waits for replies (acknowledgements) that are carried in the reverse packet stream. If no acknowledgement for a particular packet is received, the TCP will retransmit. It is our expectation that the HOST level retransmission mechanism, which is described in the following paragraphs, will not be called upon very often in practice.

Id. (partial). Cerf and Kahn explain that the use of sequence numbers allows tracking of packets even when the retransmitted packets go through a different set of nodes (which might require different message segmentation) in the later transmission.

A 600-byte packet might pass through one GATEWAY and be broken into two 300-byte packets. On retransmission, the same packet might be broken into three 200-byte packets going through a different HOST. Since each byte has a sequence number, there is no confusion at the receiving TCP. We leave for later the issue of initially synchronizing the sender and receiver left window edges and the window size.

Id., page 8, col. 1 ¶ 6.

99. A POSITA would have been aware that TCP provided for retransmission. *See, e.g.*, W. Richard Stevens, "TCP/IP Illustrated Volume 1 The Protocols" Chapter 21 "TCP Timeout and Retransmission":

TCP provides a reliable transport layer. One of the way it provides reliability is for each end to acknowledge the data it receives from the other end. But data segments and acknowledgments can get lost. TCP handles this by setting a timeout when it sends data, and if the data isn't acknowledged when the timeout expires, it retransmits the data. A critical element of any implementation is the timeout and retransmission strategy. How is the timeout interval determined, and how frequently does a retransmission occur?

TCP/IP Illustrated, page 297.

100. In addition to the Cerf and Kahn paper and basic textbooks, of course, the Internet Engineering Task Force (IETF) publishes the official Internet standards (as well as proposed standards) online, free for the public to access. RFC 1122 and RFC 793 together define the communication layers that the Internet uses. The official text of RFC 793 and 1122 are located at the following URLs³: (RFC 793) <https://www.rfc-editor.org/rfc/rfc793.txt> (RFC 1122) <https://www.rfc-editor.org/rfc/rfc1122.txt>.

101. As suggested by the Cerf and Kahn paper, RFC 793 mandates that, for reliable communication, “When the TCP transmits a segment containing data, it puts a copy on a retransmission queue and starts a timer; when the acknowledgment for that data is received, the segment is deleted from the queue. If the acknowledgment is not received before the timer runs out, the segment is retransmitted.” RFC 793, page 10.

102. Again, as suggested by the Cerf and Kahn paper, each TCP segment has a sequence number, which is used to enable retransmission:

A fundamental notion in the design is that every octet of data sent over a TCP connection has a sequence number. Since every octet is sequenced, each of them can be acknowledged. The acknowledgment mechanism employed is cumulative so that an acknowledgment of sequence number X indicates that all octets up to but not including X have been received. This mechanism allows for straight-forward duplicate detection in the presence of retransmission. Numbering of octets within a segment is that the first data octet immediately following the header is the lowest numbered, and the following octets are numbered consecutively.

RFC 793, page 24.

b. Cellular Standard Uses of Retransmission

103. As described below in Section IX-F, retransmission for the uplink was proposed for Universal Terrestrial Radio Access Network (“UTRAN”) in TR 25.896 v6.0.0 (“TR 25.896”),

³ Note that the RFC designates the live URL as the canonical, definitive text.

which was publicly available at the 3GPP website in March of 2004. Downlink retransmission capabilities for UTRAN were already available at that time.

5. Memory Allocation Using a Common Memory Resource

104. Memory allocation using a common memory resource is the common practice of general purpose computers and specialized computers such as digital signal processing devices to have a common memory resource used for multiple applications. The computer allocates memory from the common memory resource to the various applications according to their needs. Memory allocation using a common memory resource is another standard tool in communications at the time of the alleged invention.

105. The idea that a computer will have a common memory resource that is shared by multiple functions is evident in the standard libraries defined for the C programming language as described in the 1988 Second Edition of the classic 1978 book “The C Programming Language” by B. W. Kernighan and D. M. Ritchie. On page 167, as shown below, this classic book on computer programming defines the functions malloc and calloc that obtain blocks of memory dynamically from the common memory resource.

7.8.5 Storage Management

The functions `malloc` and `calloc` obtain blocks of memory dynamically.

```
void *malloc(size_t n)
```

returns a pointer to `n` bytes of uninitialized storage, or `NULL` if the request cannot be satisfied.

```
void *calloc(size_t n, size_t size)
```

returns a pointer to enough space for an array of `n` objects of the specified size, or `NULL` if the request cannot be satisfied. The storage is initialized to zero.

The pointer returned by `malloc` or `calloc` has the proper alignment for the object in question, but it must be cast into the appropriate type, as in

```
int *ip;
```

```
ip = (int *) calloc(n, sizeof(int));
```

`free(p)` frees the space pointed to by `p`, where `p` was originally obtained by a call to `malloc` or `calloc`. There are no restrictions on the order in which space is freed, but it is a ghastly error to free something not obtained by calling `calloc` or `malloc`.

106. The function `free` shown above un-allocates the memory block that was allocated by `malloc` or `calloc`. As defined in this ubiquitous computer programming language, at least a portion of the memory may be allocated to any function at any one particular time. Indeed, which portion of the memory is used to supply the allocated memory is not even visible to the calling function, it only sees the pointer to the memory resource.

107. Many graduate students and working engineers in the 1980s and 1990s, including myself, implemented standard communications tools including multicarrier transmission, Reed-Solomon coding, interleaving, and retransmission using the C programming language and in particular the `malloc`, `calloc`, and `free` commands to manage memory usage carefully.

108. Variables and arrays can often be defined even without explicitly using these commands, but these variables and arrays nonetheless are allocated memory from the common

memory resource which is returned to the common memory resource when the program has completed its execution. Here again, which portion of the memory is used to supply the allocated memory is not even visible programmer, who only sees the variable or array names that indirectly refer to the memory resource. Here again, at least a portion of the memory may be allocated to any function at any one particular time.

109. After initial simulations that were often implemented using the C programming language, standard communications tools including multicarrier transmission, Reed-Solomon coding, interleaving, and retransmission were implemented on digital signal processors (DSPs) such as the AT&T DSP, the AT&T DSP 32, and the Texas Instruments TMS 320.

110. The AT&T DSP was described by J. R. Boddie in his 1981 article in the *Bell System Technical Journal*. As shown below as Figure 15 of this document, Figure 2 on page 1433 of Boddie's 1981 article shows that the DSP has a single common resource for data memory so that the various functions using that data memory had to share this common resource and at least a portion of the memory may be allocated to any function at any one particular time.

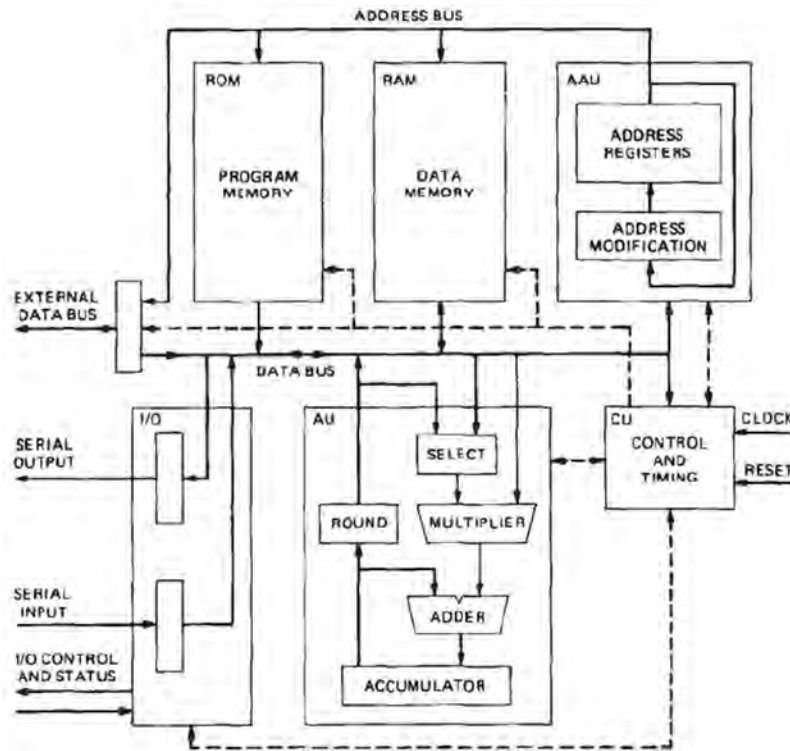


Fig. 2—Digital signal processor block diagram.

OVERVIEW 1433

Figure 15: This figure, which is Figure 2 from the 1981 Boddie journal paper, shows the AT&T DSP architecture which features a single common resource for data memory so that the various functions using that data memory had to share this common resource and at least a portion of the memory may be allocated to any function at any one particular time.

111. In his 1981 article and other articles in that issue of the Bell System Technical Journal, Boddie described how the original AT&T DSP could support a variety of applications, shown in Table 1 of that article:

Table I—Application complexity of DSP

Application	Complexity (No. of DSPs)
Second-order section	0.03
ADPCM* coder	0.25
Dual-tone receiver	0.90
Modem, 1200-baud	1.50
Transmultiplexer	6.00

* Adaptive differential pulse-code modulation.

112. Boddie revisited that historic device in his 2017 magazine article “A Brief History of AT&T’s First Digital Signal Processor.”

113. The original DSP used fixed point arithmetic. In 1986, Boddie described the AT&T DSP32, which was a digital signal processor that supported 32-bit floating-point arithmetic. As described by Boddie in his AT&T Technical Journal article, entitled “The DSP32 Digital Signal Processor and Its Application Development Tools”:

The WE® DSP32 digital signal processor is a high-speed, programmable, VLSI circuit with 32-bit floating-point arithmetic. The device can be used cost-effectively in a wide variety of complex digital signal processing applications, such as speech recognition, high-speed modems, low bit-rate voice coders, multi-channel signaling systems, and signal processing workstations. In this paper, we review the architecture of the DSP32 and present its instruction set, including some examples. We also discuss the software and hardware support tools that are available for developing DSP32 applications.

114. As with the DSP, the DSP32 used a common resource for data memory so that the various functions using that data memory had to share this common resource and at least a portion of the memory may be allocated to any function at any one particular time. Figure 1 from page 93 of Boddie’s 1986 article shows the DSP 32 architecture. This figure is reproduced as Figure 16 below.

115. As described in Boddie’s 1986 article:

The DSP32 can do the following, all with 32-bit, floating-point precision and dynamic range:

- a 1024-point, complex, Fast Fourier Transform (FFT) in 19.2 ms (including bit reversal)
- a finite impulse response filter in 250 ns per tap
- a second-order section (four multiply) for a recursive, infinite-impulse-response (IIR) filter in 1 μ s.

116. Thus, the basic building block of DMT or OFDM could be performed on a DSP32 in 1986 at a throughput of 1024 received symbols every 19.2 ms or 53,333 symbols per second.

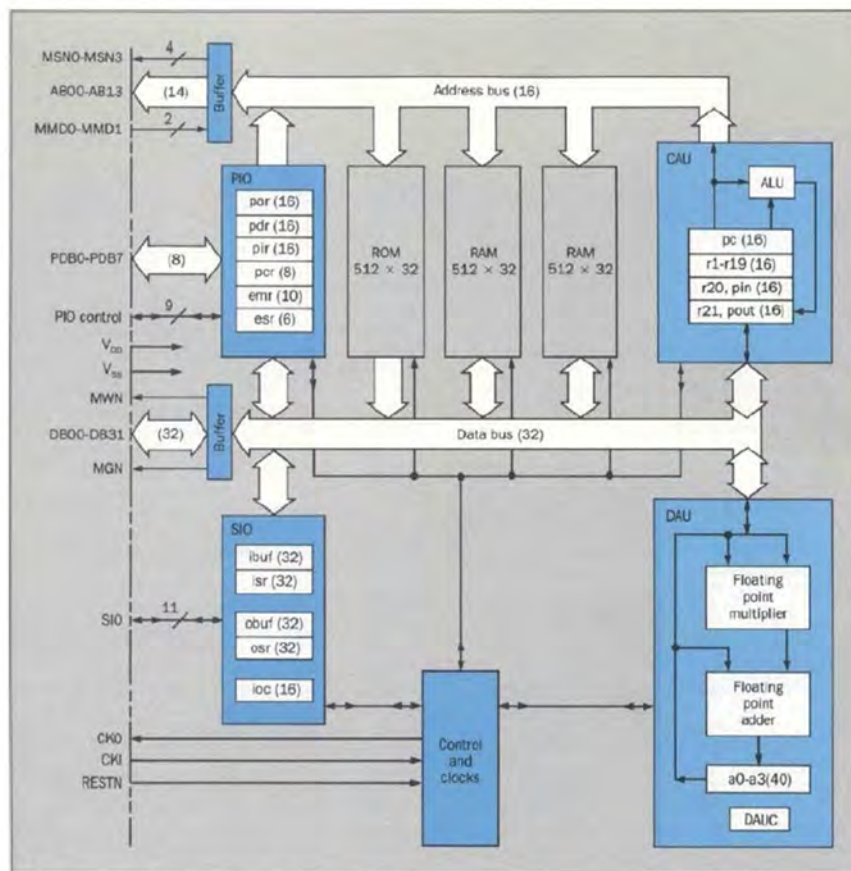


Figure 1. Block diagram of the DSP32 digital signal processor. The numbers in parentheses designate the size of a particular register or bus. The symbols at the left represent pins on the device package.

Figure 16: This figure, which is Figure 1 from the 1986 Boddie journal paper, shows the AT&T DSP32 architecture which features a common resource for data memory so that the various functions using that data memory had to share this common resource and at least a portion of the memory may be allocated to any function at any one particular time.

117. In 1995 R. E. Crochiere and J. R. Boddie published a journal paper in the *AT&T Technical Journal* entitled “Digital Signal Processors: Toward Vision 2001” that described the general trend towards the use of digital signal processors with a shared memory resource. In their words, “Today, the increasing sophistication of telephone equipment, modems, and computer multimedia is driving the need for high-volume implementation of application-specific digital signal processors (DSPs) having particular emphasis on low-voltage battery operation, moderate cost, faster time to market, software flexibility, and improved performance.”

118. Figure 17 below shows Figure 1 from the Crochiere and Boddie 1995 article, which shows the general architecture of DSPs. Note that the module in the diagram labeled Dual-port RAM is the shared memory resource. Any DSP that follows this general architecture will have the property that at least a portion of the memory may be allocated to any function at any one particular time.

119. Not only is the memory of the DSP shared, but it is also recognized by Crochiere and Boddie in 1995 as that DSPs would have only the minimum amount of this precious resource, so that it must be allocated carefully:

Memory. Two types of memory are typically incorporated into the design of DSPs: *read-only memory* (ROM) for non-volatile storage of program instructions, and *random-access memory* (RAM) for storage of algorithm-state variables, computed data, or control information. RAM is also used for applications in which programs are downloaded or cache loaded into the DSP.

For high-volume, cost-sensitive applications, DSPs are often customized. They are designed having only the minimum required RAM and ROM, as well as the necessary I/O interfaces for a given application. Such a design reduces both the DSP's silicon area and device cost. Memory sizes range from 512 to 8,192 words of RAM and from 4,096 to 32,767 words of ROM.

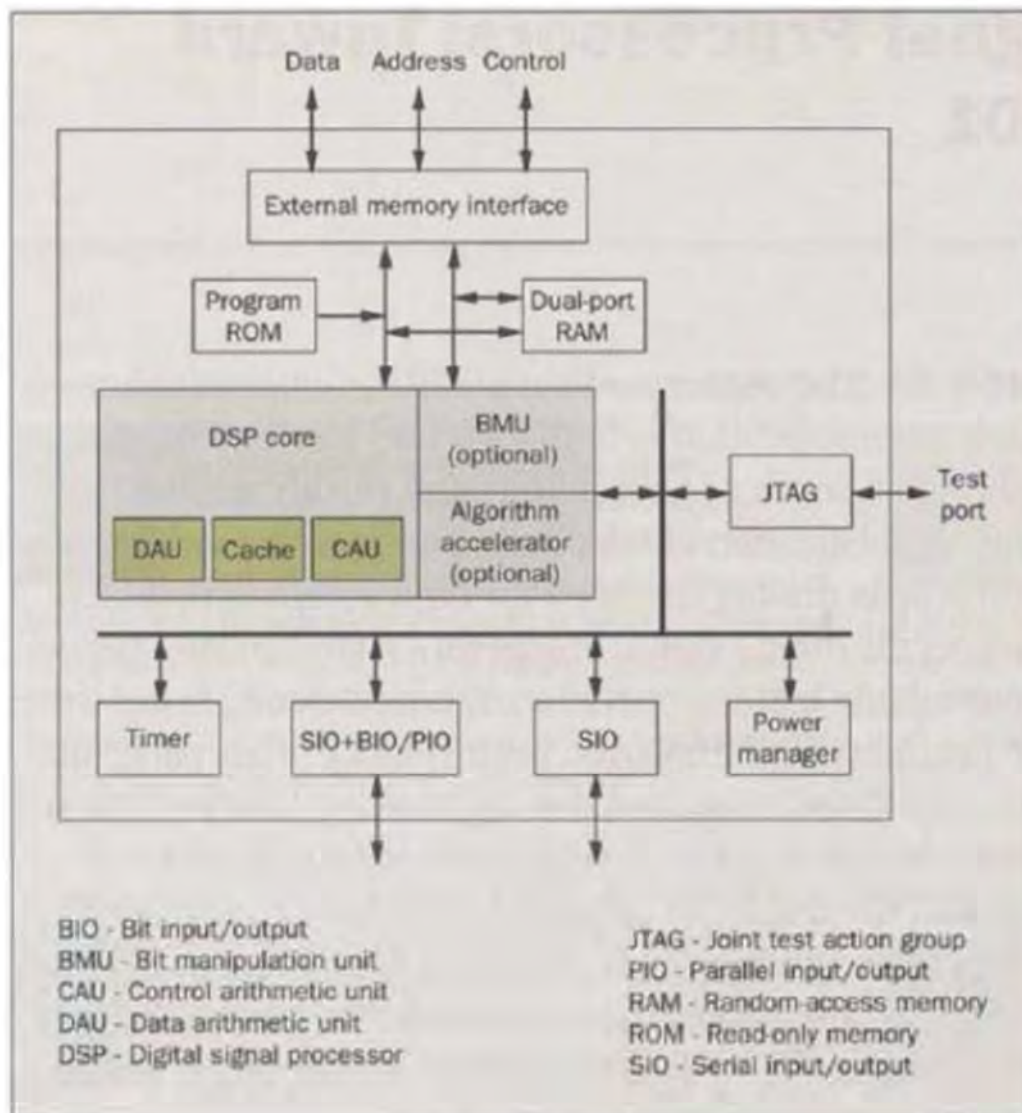


Figure 1. This illustration shows some of the basic features found in DSP architectures. DSPs are designed for high-speed, real-time numerical computation. They differ from conventional microprocessors or microcontrollers in that they are designed specifically for signal processing applications. They can process signals far more cost effectively—and with lower overhead and power consumption—than general-purpose processors. They also feature input/output (I/O) interfaces configured for such sampled data-processing systems as coders/decoders (codecs) and high-speed memory.

Figure 17: This figure, which is Figure 1 from the 1995 Crochiere and Boddie journal paper, shows the general architecture for DSPs. The module in the diagram labeled Dual-port RAM is the shared memory resource.

120. The Crochiere and Boddie 1995 article gives the example of a DSP 1618 performing the digital baseband functions for both the 13-kbps GSM speech coding and the channel processing. The channel processing functions include such operations as Viterbi processing, convolutional decoding, and burst interleaving. Figure 18 shows the overall GSM architecture using the DSP 1618. As with other DSPs, that at least a portion of the memory may be allocated to any of several functions at any one particular time.

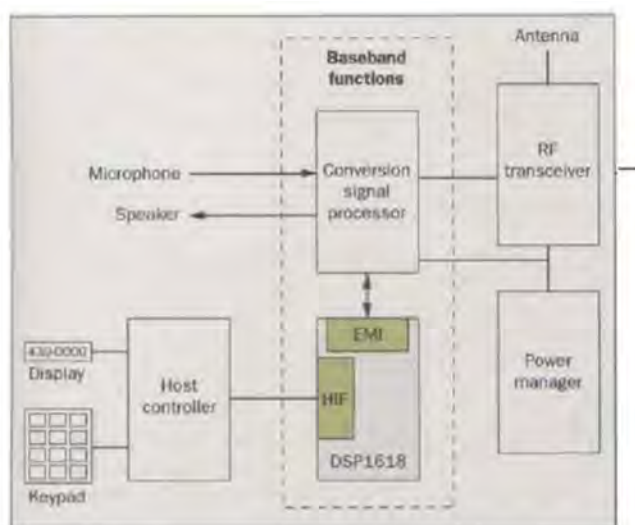
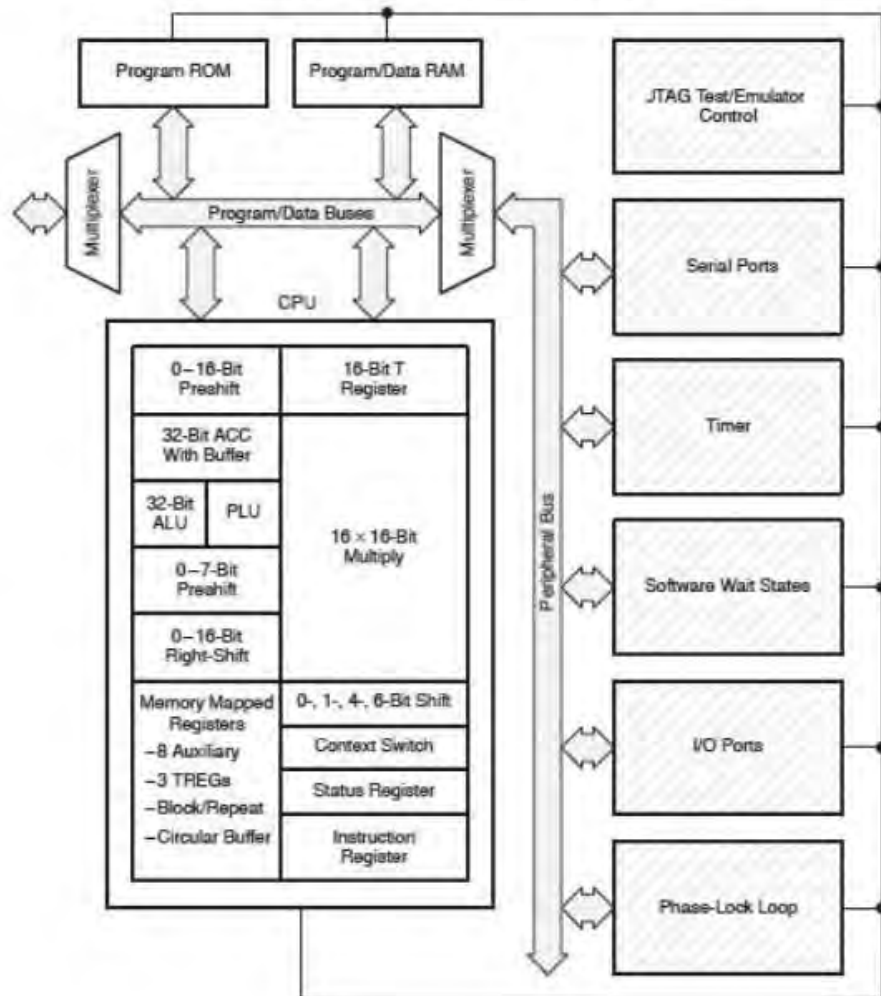


Figure 4. This drawing shows an example of a highly customized AT&T DSP1618 architecture for the European Global System for Mobile Communications (GSM) cellular terminal standard. The DSP communicates with a microcontroller that handles system-management functions. Additionally, the DSP communicates with a conversion processor chip that provides both the voice-band analog-to-digital (A/D) and digital-to-analog (D/A) conversions and analog connections to the speaker and microphone. The conversion processor also performs base-band A/D and D/A conversions, modulation functions, buffering, and communications to the radio interface. A transceiver provides the RF modulation and processing functions for the terminal.

Figure 18: This figure, which is Figure 4 from the 1995 Crochiere and Boddie journal paper, shows the architecture for a DSP1618. The DSP 1618 performs the digital baseband functions for both the 13-kbps GSM speech coding and the channel processing.

121. While AT&T was developing its line of DSPs, there was serious competition from Texas Instruments, who developed their own line of DSPs including the extremely successful TI DSP TMS320. In 1994, TI published an application book entitled “Telecommunications Applications With the TMS320C5x DSPs,” which described transceivers with shared memory. Figure 2 from the application book shows the TMS 320 C5x Architecture, which features the Program/Data RAM as the shared memory.

Figure 2. Key Features of the TMS320C5x Architecture

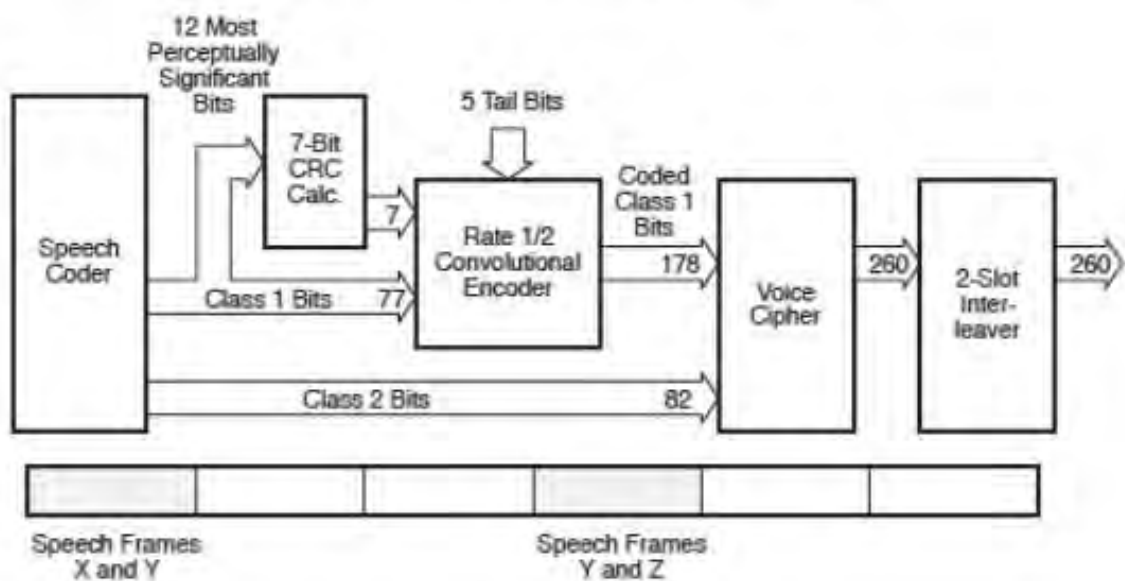


122. The application book describes an IS 54 transceiver, and also describes a C-language simulation of IS-54 that TI made publicly available. As discussed above, any C-language implementation of a transceiver would utilize shared memory. *See* the excerpt and figure shown below.

Introduction

This paper describes a C language simulation of both the transmit and receive baseband processing for a digital cellular telephone that meets the U.S. digital cellular standard (IS-54B). This simulation is needed for two reasons: first, to gain greater understanding of the IS-54 digital cellular standard and the associated digital signal processing required in a terminal that meets this standard with a vision toward efficient implementation on the TMS320 DSPs; second, to gain the capability to evaluate the effect of bit errors on the speech coder (vector sum excited linear prediction, or VSELP) and IS-54 control functions. This necessitated development of a simulation of the IS-54 processing and RF channel. See Figure 1.

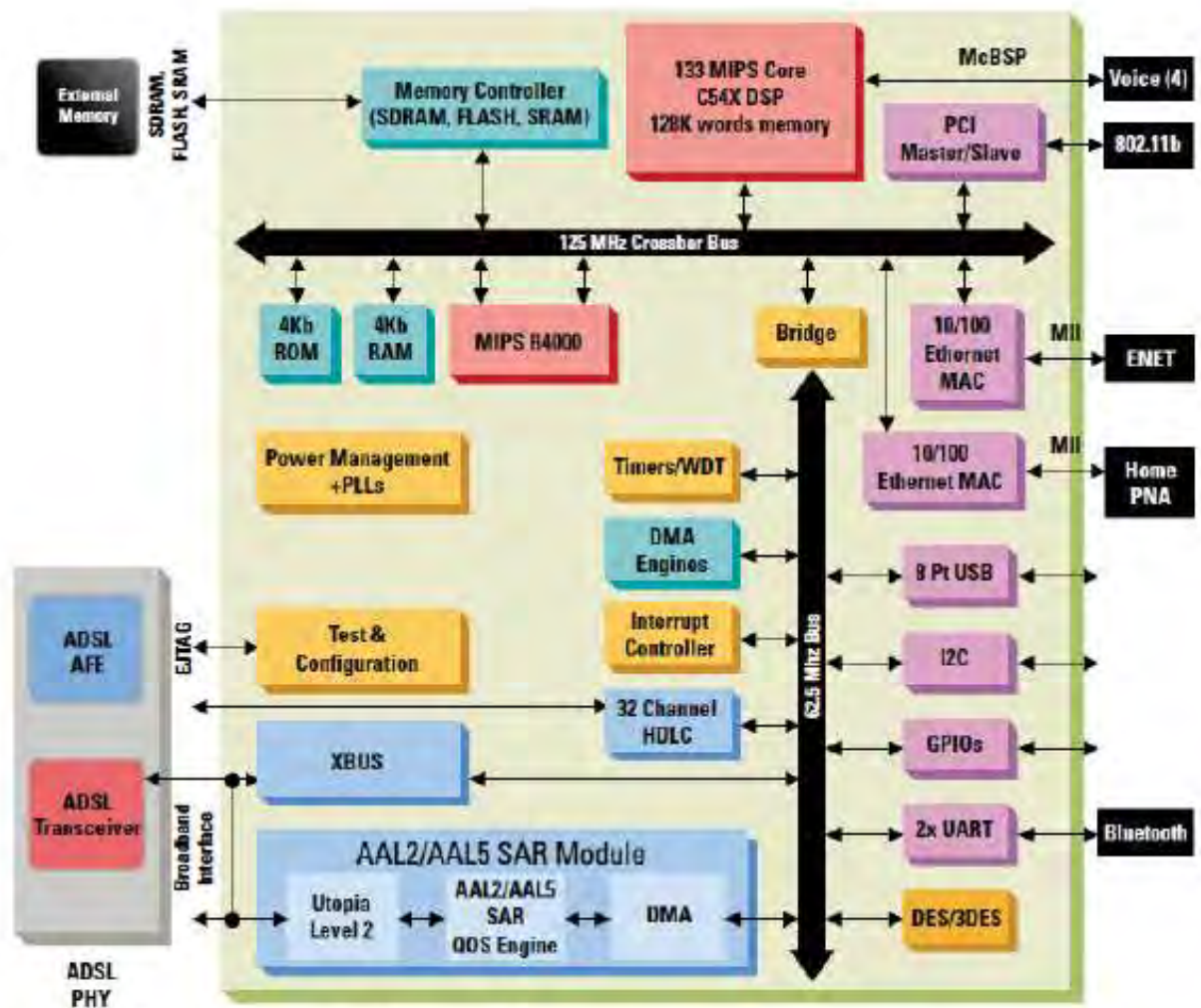
Figure 2. IS-54 Error Encoding and Interleaving



123. Additionally, the TMS 320 was specifically used to implement ADSL transceivers with shared memory long before the alleged inventions. For example, the TI AR5 Chipset Family described in the 2001 Product Bulletin entitled “AR5 Chipset Family: ADSL Router, Integrated Access Device and Residential Gateway Chipset Solutions” included an ADSL transceiver shown in the figure below:

AR5 Chipset Family

ADSL Router, Integrated Access Device and Residential Gateway Chipset Solutions



AR5V10 DSL Communications Processor

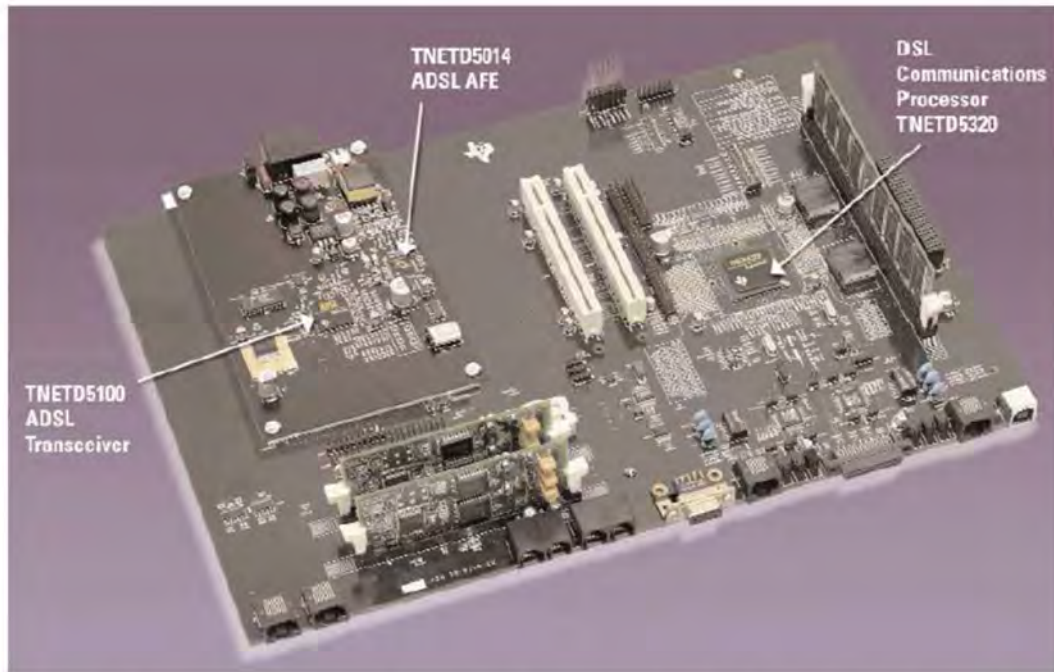
A superior architecture, including a 32-bit RISC processor and a high-performance DSP, provides outstanding throughput for best-in-class routing and low voice latency.

124. As described in the product bulletin as excerpted below, the ADSL transceiver (specifically the TNET5100) is implemented on a TMS 320 DSP, specifically the TMS320C6000, which is explicitly identified as a programmable DSP-based engine, which would use shared memory for the programmable functions.

The common two-chip AX5 PHY includes:

- The TNETD5100 ADSL Transceiver, a programmable TMS320C6000™ DSP-based engine dedicated to high-speed ADSL transmission and reception.
- The TNETD5014 ADSL Codec and Line Driver/Receiver, a highly integrated ADSL analog front end (AFE) with support for both full-rate G.dmt and G.lite.

125. Below is a photograph from the AR5 product bulleting showing the TNET 5100 ADSL Transceiver.



The AR5 platform for VoDSL offers up to eight channels of managed voice.

126. The AR5 product bulletin explicitly describes the shared memory used by the TNETD5100 ADSL Transceiver as a 1-Mbit on-chip instruction/data cache with enhanced DMA control for sustained high throughput, as shown below:

PHY Devices (common to all chipsets)

TNETD5100 ADSL Transceiver

- Programmable 1600-MIPS TMS320C6200™ DSP core with VelociTI™ very-long instruction word (VLIW) architecture performs up to eight 32-bit instructions simultaneously
- 1-Mbit on-chip instruction/data cache with enhanced DMA control for sustained high throughput
- Glueless 32-bit external memory interface to synchronous and asynchronous devices
- Glueless interface to TNETD5014 AFE

TNETD5014 ADSL Codec and Line Driver/Receiver

- Complete AFE integrates 14-bit ADC and DAC, line driver/receiver, TX/RX filter, programmable gain amplifiers and equalizer
- Supports both full-rate G.dmt (ITU G.992.1) Annex A, and G.lite (ITU G.992.2) ADSL transmission
- Coexists with Bluetooth™, Wireless Ethernet (802.11b) and HPNA devices
- Low power consumption

127. Separately, Intel developed implementations of multicarrier transceivers that used shared memories such as the Intel PRO/Wireless 5116 implementation of the WiMAX standard 802.16. As described below in Section VII.B, the 802.16 standard implemented multicarrier transmission, Reed-Solomon encoding, interleaving, and retransmission on the 5116, which uses shared memory. As described below in the Intel “Fact Sheet” dated February 11, 2008, by April 2005 Intel had announced volume shipments of the 5116. *See also* the Intel News Release dated April 18, 2005.

April 2005: Intel announces volume shipments of its first WiMAX product, the Intel® PRO/Wireless 5116. The 5116 targets interoperable, customer premise equipment used by telecommunications service providers to deploy fixed broadband wireless based upon the IEEE 802.16-2004 standard.

128. The typical implementation of cellular technology uses a baseband processor that employs shared memory. Examples include the Zyray spinner baseband processor that supported 3GPP WCDMA technology such as that disclosed in TR 25.896. A press release from April 2004 describes the WCDMA baseband processor as supporting 3GPP Release 99 and providing an upgrade path to release 4.

WCDMA Baseband Processor

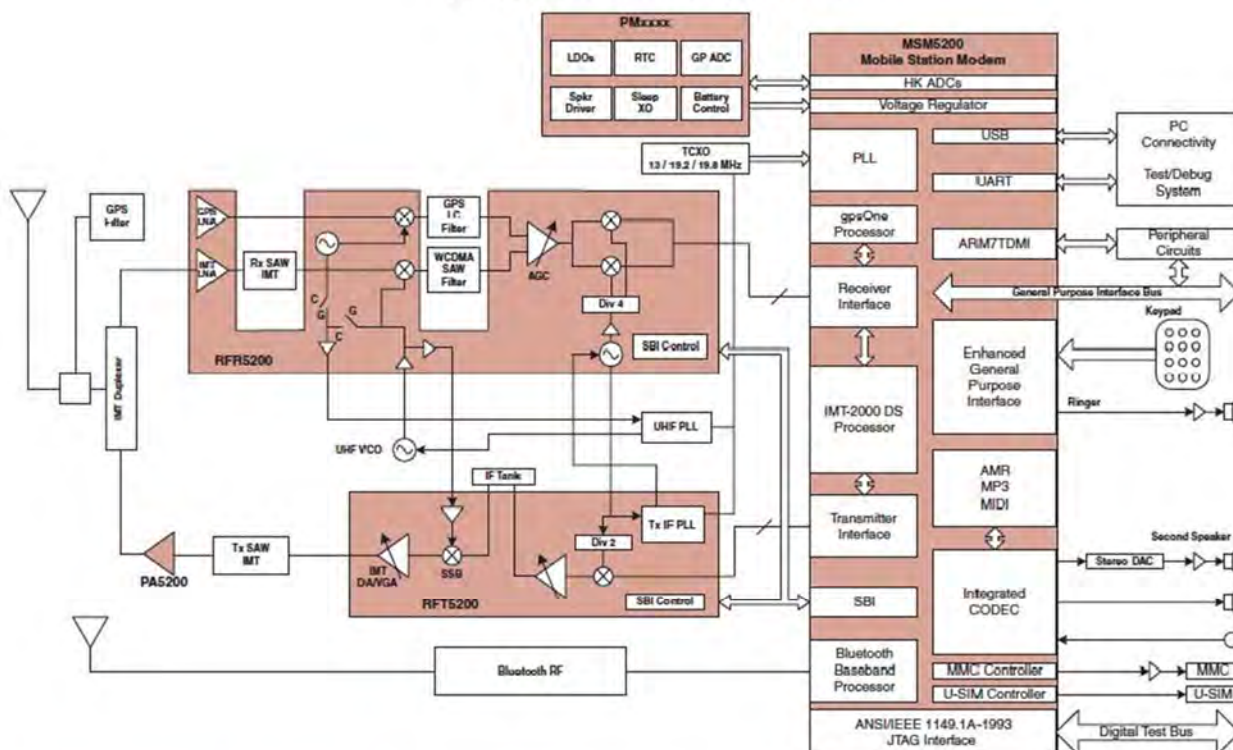
Wed, 04/07/2004 - 11:04am by

Zyray introduces the *SPINNERchip* 1.1. The product enables device manufacturers to develop single-mode WCDMA, dual-mode GSM/GPRS & WCDMA, and multi-mode GSM/GPRS, EDGE & WCDMA devices. *SPINNERchip* 1.1 is housed in a 10 × 10 mm BGA including integrated analog technology and is flexible enough to mate with various RF transceivers, baseband processors, and dual-mode protocol stacks. Other features include the following: 3GPP Rel 99 December 2003 compliant WCDMA FDD modem; firmware upgrade path via to 3GPP Rel 4; 384 kbps class modem operation in both uplink and downlink; supports Circuit-Switched Voice, Packet-Switched Data and Circuit-Switched Data; Generic Layer 1 Abstraction Layer allows for 3rd party protocol stack support; complete firmware suite including PHY layer and Layer 1; support for USIM interface (optional); on-chip, programmable power management; supply voltage: 1.8 V / 3.0 V.

Zyray Wireless

129. Another example is the Qualcomm Mobile Station Modem (MSM) 5200 baseband processor chip that was available by 2001 and was 3GPP Release 99 compliant. Shown below is an excerpt from Qualcomm's product literature about the MSM 5200:

QUALCOMM's MSM5200 Chipset Solution



MSM5200 DEVICE DESCRIPTION



The MSM5200 chipset and system software builds on the successful architecture of QCT's MSM3xxx and MSM5xxx families of CDMA devices, and is based on QCT's core CDMA systems expertise gained from years of field experience with this technology. The MSM5200 solution integrates both digital and analog

functions into a single chip.

The MSM5200 solution includes advanced technologies such as GPS based position location and Bluetooth, as well as multimedia features such as iTunes™ MP3 player software and Compact Media Extension™ (CMX™) MIDI-based multimedia software. Along with an

optimized software solution for the WCDMA modem, QCT offers system development software, verification, test, debug, calibration, manufacturing, and field test support using the WCDMA designer development tools, which reduce time-to-market for a complete WCDMA product.

MSM5200 General Device Enhancements

- Integrated audio codec
- Supports low-power, low-frequency crystal to enable TCXO shutoff
- Enhanced I/O support for faster RS-232
- 16-bit-wide memory support
- Embedded Bluetooth baseband processor
- Voice Recognition (optional) –Speaker-dependent, speaker-independent and voice-prompt support
- Integrated mass-storage device (MMC) controller
- Audio enhancement technology support (iTunes and CMX software)
- Integrated USB device controller for fast and simple PC interconnect
- Interfaces with the MSP™ family of applications processors

MSM5200 IMT-2000 DS Device Enhancements

- 3GPP Release 99 compliant
- Full 3GPP protocol stack including L1, L2:MAC/RLC/BMC/PDCP, L3:RRC, (G)MM, SS, CC, SMS, SM, RABM
- Data rates: 384 kbps on both the uplink and downlink
- Adaptive Multiple Rate (AMR) speech codec
- Support for convolutional and turbo coding
- USIM interface

130. As discussed above, evidence abounds that a POSITA understood that for each transceiver function, appropriate memory must be allocated to that function. The very nature of general purpose computers and digital signal processors demands this. Of interest is evidence that a POSITA would understand that this is specifically true for interleaving/de-interleaving and for retransmission.

131. As an example of such evidence regarding the necessity to allocate memory to interleaving, the VDSL2 standard G.993.2 from February 2006 includes the following specific statement in Section 6.2.8 entitled “Aggregate interleaver and de-interleaver delay” describing the amount of memory required to support the interleaving/de-interleaving for each latency path and overall:

Each interleaver and each de-interleaver for each latency path requires at least $\lceil \text{delay\_octet}_{x,p}/2 \rceil$ octets of memory to meet this delay. The actual amount of memory used is implementation specific.

Referring to Figure 6-1, the aggregate interleaver and de-interleaver delay is specified as the sum $\text{delay\_octet}_{DS,0} + \text{delay\_octet}_{DS,1} + \text{delay\_octet}_{US,0} + \text{delay\_octet}_{US,1}$,

which can be rewritten as:

$$\sum_p (I_{US,p} - 1) \cdot (D_{US,p} - 1) + (I_{DS,p} - 1) \cdot (D_{DS,p} - 1)$$

VDSL2 modems shall comply with the requirement

$$\sum_p (I_{US,p} - 1) \cdot (D_{US,p} - 1) + (I_{DS,p} - 1) \cdot (D_{DS,p} - 1) \leq \text{MAXDELAYOCTET},$$

where the summation is over all latency paths and MAXDELAYOCTET is the parameter “aggregate interleaver and de-interleaver delay”, in octets, specified in Table 6-1 for the profile.

The minimum amount of memory required in a transceiver (VTU-O or VTU-R) to meet this requirement is $\frac{\text{MAXDELAYOCTET}}{2}$ octets. The actual amount of memory used is implementation specific.

132. As an example of such evidence regarding the necessity to allocate memory to retransmission functions, TR 25.896 in Section 7.2 entitled “Hybrid ARQ” discloses “memory requirements” in both transceivers as the first on a list of aspects that must be taken into account in a hybrid ARQ scheme: shared memory.

The design of an uplink hybrid ARQ scheme should take the following aspects into account:

- Memory requirements, both in the UE and the Node B. Rapid retransmissions reduce the amount of buffer memory required in the Node B for buffering of soft bits when a retransmission has been requested.

TR25.986 at NOK00074952.

133. As a further example of sharing memory, consider U.S. Patent No. 7,027,782 to Moon et. al. (“Moon”), which illustrates the use of shared memory in a communications system, particularly a mobile communications system. Moon issued on April 11, 2006, from an application filed on October 18, 2002.

134. Moon is directed to a “method for retransmitting coded bits by a transmitter in response to a retransmission request from a receiver in a mobile communication system.” Moon at Abstract. As shown in Figure 7, below, Moon describes the use of multiple interleavers in its transmission path.

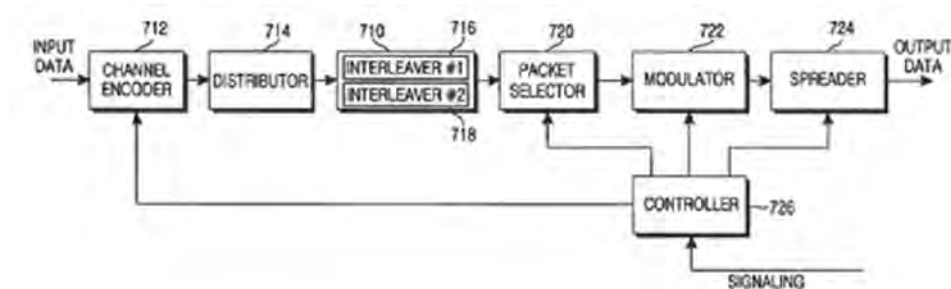


FIG.7

135. Moon notes that while “the first interleaver 716 and the second interleaver 718 are separated by hardware” in Figure 7, “the first interleaver 716 and the second interleaver 718 can be simply logically separated.” Moon at NOK00083404, 12:39-43. As described by Moon, the “logical separation means dividing one memory into a memory area for storing the systematic bits and another memory area for storing the parity bits” that are interleaved by the first and second

interleavers, respectively. Moon at NOK00083404, 12:43-45. In other words, Moon describes a common memory that is logically separated between two different functions, e.g.: the first and second interleavers.

136. Such logical separation of the memory clearly extends to logical separation of memory, i.e., sharing memory, with the retransmission function as well. Moon discusses retransmission at length, including in the title, which is “Transceiver Apparatus and method for efficient High-Speed Data Retransmission and Decoding in a CDMA Mobile Communication System.” Moon at NOK00083378.

137. As shown in Moon 2:24-65, Moon teaches retransmission using three distinct techniques: Chase combining, Full Incremental Redundancy (FIR), and Partial Incremental redundancy (PIR), each of which requires storing at the receiver every received packet for which a retransmission will be requested. Moon at NOK00083399, 2:24-65. Each method also requires storing portions of the transmitted packet at the transmitter so they are available if they are needed for retransmission. *Id.* In both cases, this information would, following the teaching of Moon, be stored using logical separation, i.e., a shared memory, in at least one implementation. *Id.*

The HARQ is a link control technique for correcting an
25 error by retransmitting the errored data upon an occurrence
of a packet error at an initial transmission. Generally, the
HARQ is classified into Chase Combining (CC), Full Incre-
mental Redundancy (FIR), and Partial Incremental Redun-
dancy (PIR).

30 CC is a technique for transmitting a packet such that the
whole packet transmitted at a retransmission is equal to the
packet transmitted at the initial transmission. In this tech-
nique, a receiver combines the retransmitted packet with the
initially transmitted packet that is previously stored in a
35 buffer thereof by a predetermined method. By doing so, it is
possible to increase reliability of coded bits input to a
decoder, thus resulting in an increase in the overall system
performance. Combining the two same packets is similar to
repeated coding in terms of the effects, so it is possible to
40 increase a performance gain by about 3 dB on average.

FIR is a technique for transmitting a packet comprised of
only redundant bits generated from the channel encoder
instead of the same packet, thus improving performance of
a decoder in the receiver. That is, the FIR uses the new
45 redundant bits as well as the initially transmitted information
during decoding, resulting in a decrease in the coding rate,
which thereby improves performance of the decoder. It is
well known in coding theory that a performance gain by a
low coding rate is higher than a performance gain by
50 repeated coding. Therefore, the FIR is superior to the CC in
terms of only the performance gain.

Moon at NOK00083399, 2:24-51.

138. Hence Moon taught sharing a memory between a retransmission function and an interleaving/deinterleaving function. I note that in rejecting claim 107 the examiner determined that Moon teaches sharing a memory between a retransmission function and an interleaving/deinterleaving function. This is shown below in an excerpt from the '411 Pros. Hist.

7. In regard to claim 107, Moon teaches:

- A method of packet retransmission comprising
- transmitting or receiving, using a transceiver, a plurality of packets;

(Figure 9A, ref. S1 & S2 & S3 & S4 & P1 & P2 & P3 & P4 and section [0086] in Moon)

- identifying at least one packet of the plurality of packets as a packet that should be retransmitted; and

(Packets S1 & S2 & S3 in figure 9A in Moon)

- sharing a memory between a retransmission function and an interleaving/deinterleaving function.

(Section [0059] in Moon)

'411 Pros. Hist.

139. Moon further describes how the relative sizes, i.e., allocations, of the logically separated memory, i.e., shared memory, may change during the operation of the transceivers as they adjust their operating modes. For example, depending on the coding rate applied, the relative inputs to the first and second interleaver memories can change. With a coding rate of $1/2$, the first and second interleaver memories will receive a symmetric input, i.e., for each bit that goes into the first interleaver, one bit will go into the second interleaver. Moon at NOK00083404, 12:12-14. But for a coding rate of $3/4$, for every three bits that go into the first interleaver, only one bit will go into the second interleaver. Moon at NOK00083404, 12:15-18.

140. Thus, the required relative sizes of the first interleaver and the second interleaver may change as the coding rate changes. With a coding rate of $1/2$, 50% of the common memory used for interleaving would be assigned to the first interleaver and 50% of the common memory

used for interleaving would be assigned to the second interleaver. With a coding rate of 3/4, 75% of the common memory used for interleaving would be assigned to the first interleaver and 25% of the common memory used for interleaving would be assigned to the second interleaver.

141. Moon also describes in 2:30-65 how the memory allocated to retransmission changes based on the mode of retransmission selected. Moon at NOK00083399, 2:30-65. The portions of packets that must be stored for future transmission are different for Chase (the original packet must be stored), FIR (incremental redundancy for the original packet must be stored) and PIR (Both the original packet and a portion of the incremental redundancy must be stored). *Id.* Moon describes in 2:30-65 how different retransmission approaches will have different shared memory requirements (buffer size) and that “the HARQ should be considered in the light of not only the performance but also the system complexity, such as a buffer size and signaling of the receiver.” *Id.* NOK00083399, 2:61-64.

142. Moon discusses the shared memory allocated to retransmission (the buffer size) in numerous places including 14:35-40 “The demodulator 814 may have a buffer with a specific size to perform symbol combining if a modulation technique used at initial transmission is identical to a modulation technique used at retransmission, thereby resulting in an improvement in reliability of the LLR value.” Moon at NOK00083405, 14:35-40. And at 14:55-63 “The selective packet combiner 816 is comprised of first a buffer for an S sub-packet comprised of systematic bits and a second buffer for a P sub-packet comprised of parity bits. The combining is separately performed on the same S or P sub-packet. For example, if only the S packet was transmitted during retransmission, the retransmitted S sub-packet is combined with data that was stored in the S sub-packet buffer during initial transmission.” *Id.* 14:55-63.

143. Thus, Moon represents a particular application of the well-known and widely employed principle of memory sharing described above.

144. U.S. Patent No. 7,269,208 to Mazzoni et al. (Mazzoni) also illustrates the use of shared memory in communications system. Mazzoni issued September 11, 2007, from a PCT application filed on July 11, 2001. The PCT application published January 24, 2002, and the corresponding U.S. application published on January 30, 2003.

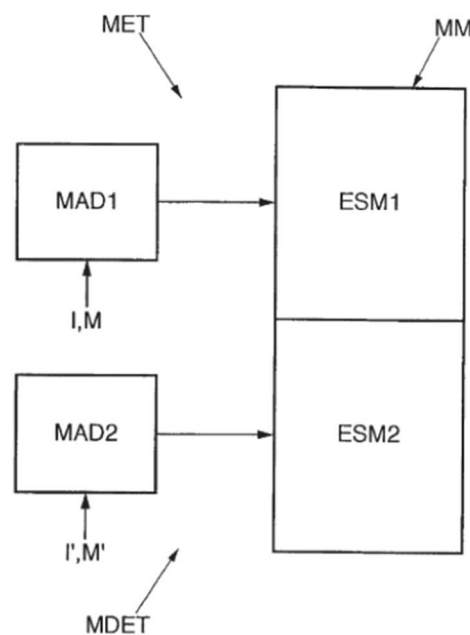
145. Mazzoni is directed to a “device for sending/receiving digital data” that “may include a channel coding/decoding stage including an interleaver, a deinterleaver, and a memory whose minimum size is fixed as a function of maximum bit rate of [a] group of predetermined bit rates.” Mazzoni at Abstract. Mazzoni’s invention “may be advantageously applied to a very high rate digital subscriber line (VDSL) environment. Mazzoni at NOK00083793, 1:19-21.

146. Mazzoni notes the well-known fact that the “process of interleaving and deinterleaving data sent and received by a modem necessitates the use of memories.” Mazzoni at NOK00083793, 1:39-40. Mazzoni sought to design a modem “which requires a reduced quantity of memory.” Mazzoni at NOK00083793, 1:47-48.

147. To do so, Mazzoni then explains how the memory can be shared between the interleaver and deinterleaver. Mazzoni describes the use of a “memory means” that “can be shared between the interleaving means and the deinterleaving means, and whose memory allocation can be reconfigured in accordance with the bit rate actually processed by the . . . [modem].” Mazzoni at NOK00083793, 1:59-65. “The memory . . . has a first memory space assigned to the interleaving means and a second memory space assigned to the deinterleaving means. The size of each of the two memory spaces is set as a function of the bit rate actually processed by the device.” Mazzoni at NOK00083793, 2:10-15.

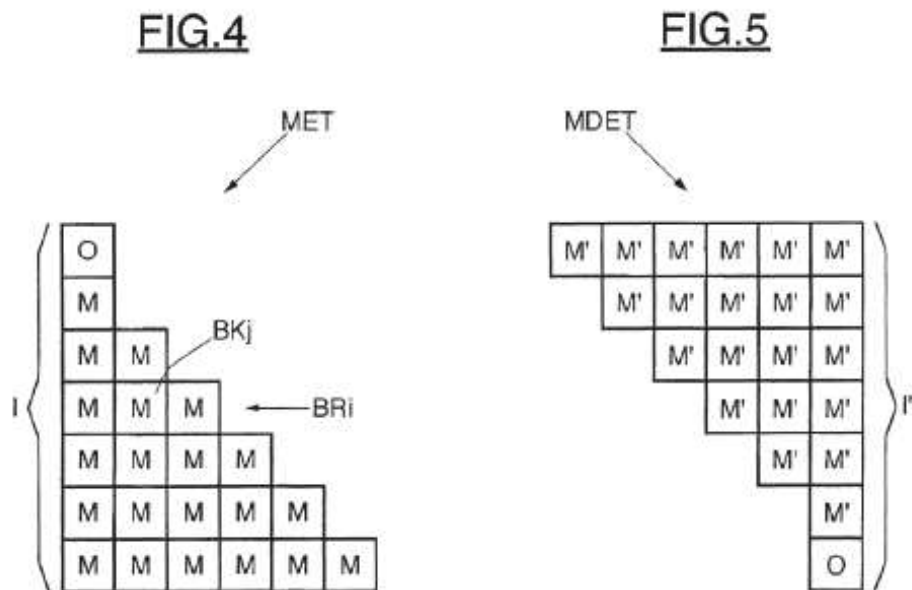
148. The common memory is depicted in Figure 6 of Mazzoni. Figure 6 shows that “the interleaving means and the deinterleaving means include common memory means MM, e.g., a dual-port random access memory. The memory space of the memory MM is then divided into a first memory space ESM1 assigned to the interleaving means MET, and a second memory space ESM2 is assigned to the deinterleaving means MDET.” Mazzoni at NOK00083795, 5:61-67.

FIG.6



149. Mazzoni also describes the parameters, I and M and I' and M' (associated with the interleaver and deinterleaver, respectively) that “define the sizes of the respective memory spaces assigned to the interleaving means and to the deinterleaving means. This is done according to the bit rate of the information sent by the terminal TO (parameters I and M) and the bit rate of the information received by the terminal TO (parameters I' and M').” Mazzoni at NOK00083795, 5:21-30.

150. As shown below in Figures 4 and 5 from Mazzoni, the typical triangular implementation of convolutional interleaving is used to compute how much memory will be allocated the interleaver and de-interleaver.



Mazzoni at NOK00083789, Figs. 4-5.

151. This is the implementation considered in VDSL1 as discussed in Section VII.D below. Mazzoni at NOK00083795, 6:37-50. Mazzoni discloses the same equation for computing the memory requirement of this triangular implementation as disclosed in VDSL1, specifically $M \times I \times (I - 1)/2$ bytes of memory are required to implement an interleaver or deinterleaver with parameters M and I . *Id.*

The parameters I , I' , M and M' can be determined from the above capacities. More particularly, the size of the first memory space needed to implement triangular convolutional interleaving with I branches of $i-1$ blocks of M bytes is equal to $I \times (I-1) \times M/2$. Similarly, the size of the second memory space ESM2 required to support the uplink bit rate is equal to $I' \times (I'-1) \times M'/2$. Also, I and I' must be sub-multiples of the size N of the Reed-Solomon code.

40 Since $I \times (I-1) \times M/2$ must be equal to 24,960, it is possible to choose $I=40$ and $M=32$. Similarly, because $I' \times (I'-1) \times M'/2$ must be at least equal to 1,920, it is possible to choose $I'=24$ and $M'=7$. This requires a slight increase in size to 1,932 to facilitate the implementation. The final size of the memory

45 MM is therefore equal to 26 892 bytes.

Mazzoni at NOK00083795, 6:37-50.

152. Thus, Mazzoni describes a common memory that can be allocated between two different functions: the interleaver and the deinterleaver. And the sizes of the interleaver and deinterleaver within that common memory can be reconfigured such that the relative sizes of the interleaver and deinterleaver vary.

153. A person of ordinary skill in the art would understand that Mazzoni's reconfigurable common memory can be shared amongst additional functions as well, including the retransmission. Thus, Mazzoni represents a particular application of the well-known and widely employed principle of memory sharing described above.

154. Another example of sharing memory between an interleaving function and a retransmission function is provided in Chen, which is described in detail in Section IX.C below. In Chen a general purpose central processing unit (CPU) in the server or in the client shares memory between interleaving and retransmission.

155. Chen performed experiments to validate the concepts presented in his paper, which combined interleaving, Reed-Solomon coding, and retransmission to provide improved internet communication. Chen at NOK00075561-562, page 3, col. 2 ¶ 3-page 4, col. 2 ¶ 1. In these

experiments, retransmission and interleaving were performed by general purpose CPUs that would, of course, used shared memory and therefore would have allocated shared memory between an interleaving function and a retransmission function. *Id.* Chen explicitly discusses how interleaving, FEC, and ARQ are performed in his experiments and expected to be performed in real systems by the server and client CPUs, as shown in the excerpt below:

mental results above show that the performance of our algorithms is sensitive mainly to packet loss and errors. Bandwidth expansion due to FEC/ARQ utilization may bring occasional congestion, packet loss and errors. However, test results indicate that the performance of our algorithms is not compromised as long as the CPU speeds of the server and clients are capable of handling the real-time interleaved FEC/ARQ coding/decoding algorithms therein.

IV. Theoretical analysis

Chen at NOK00075562, page 4, col. 2 ¶ 1 (partial).

156. This excerpt provides additional evidence that a general purpose CPU sharing memory between interleaving and retransmission functions would have been considered a typical practice by a POSITA before the alleged invention.

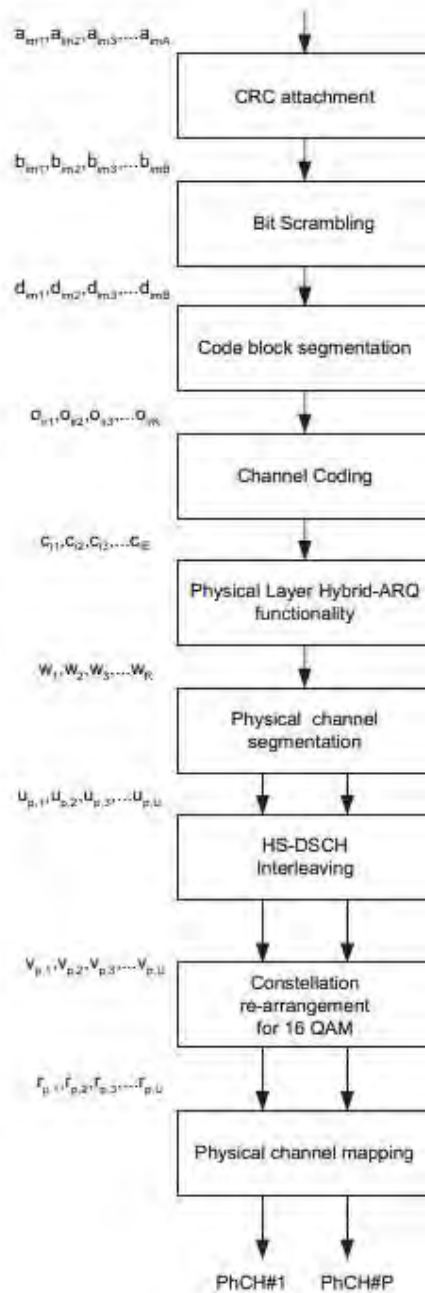
157. Another example of sharing memory between an interleaving function and a retransmission function is provided with respect to the High-Speed Downlink Shared Channel described in Section 4.5 of 3GPP TS 25.212 Version 6.0.0, released in December 2012. I note that TS 25.212 is specifically references in TR 25.896, for example in § 8.4.1 “Physical layer structures in time domain (TS25.212).” TR 25.896 at NOK00074965, § 8.4.1 “Physical layer structures in time domain (TS25.212).” The beginning of Section 4.5 is excerpted below.

4.5 Coding for HS-DSCH

Data arrives to the coding unit in form of a maximum of one transport block once every transmission time interval. The transmission time interval is 2 ms which is mapped to a radio sub-frame of 3 slots.

158. The following coding steps can be identified:

- add CRC to each transport block (*see* subclause 4.5.1);
- bit scrambling (*see* subclause 4.5.1a);
- code block segmentation (*see* subclause 4.5.2);
- channel coding (*see* subclause 4.5.3);
- hybrid ARQ (*see* subclause 4.5.4);
- physical channel segmentation (*see* subclause 4.5.5);
- interleaving for HS-DSCH (*see* subclause 4.5.6);
- constellation re-arrangement for 16 QAM (*see* subclause 4.5.7);
- mapping to physical channels (*see* subclause 4.5.8). The coding steps for HS-DSCH are shown in the figure below.



In the following the number of transport blocks and the number of transport channels is always one i.e. $m=1$, $i=1$. When referencing non HS-DSCH formulae which are used in correspondence with HS-DSCH formulae the convention is used that transport block subscripts may be omitted (e.g. X_1 may be written X).

TS 25.212, § 4.5 “Coding for HS-DSCH.”

159. Note that in Figure 16 above, the Physical Layer Hybrid ARQ functionality module occurs prior to the HS-DSCH interleaving. In Section 4.5.4 of 3GPP TS 25.212 Version 6.0.0, the retransmission capability is described. Note that there are at least two interleaving functions described below. One is the interleaving function that is part of the bit collection process described in Section 4.5.4.4. Another is the Interleaving for HS-DSCH described in Section 4.5.6.

160. A POSITA would naturally consider allocating memory from a shared memory to the retransmission function and each of these two interleaving functions. I note that, for example, reallocating memory used to interleave for bit collection in an initial transmission for later use in a possible retransmission as a so-called HARQ buffer, can be effective because this avoids needlessly copying those bits to a separate memory.

4.5.4 Hybrid ARQ for HS-DSCH

The hybrid ARQ functionality matches the number of bits at the output of the channel coder to the total number of bits of the HS-PDSCH set to which the HS-DSCH is mapped. The hybrid ARQ functionality is controlled by the redundancy version (RV) parameters. The exact set of bits at the output of the hybrid ARQ functionality depends on the number of input bits, the number of output bits, and the RV parameters.

The hybrid ARQ functionality consists of two rate-matching stages and a virtual buffer as shown in the figure below.

The first rate matching stage matches the number of input bits to the virtual IR buffer, information about which is provided by higher layers. Note that, if the number of input bits does not exceed the virtual IR buffering capability, the first rate-matching stage is transparent.

The second rate matching stage matches the number of bits after first rate matching stage to the number of physical channel bits available in the HS-PDSCH set in the TTI.

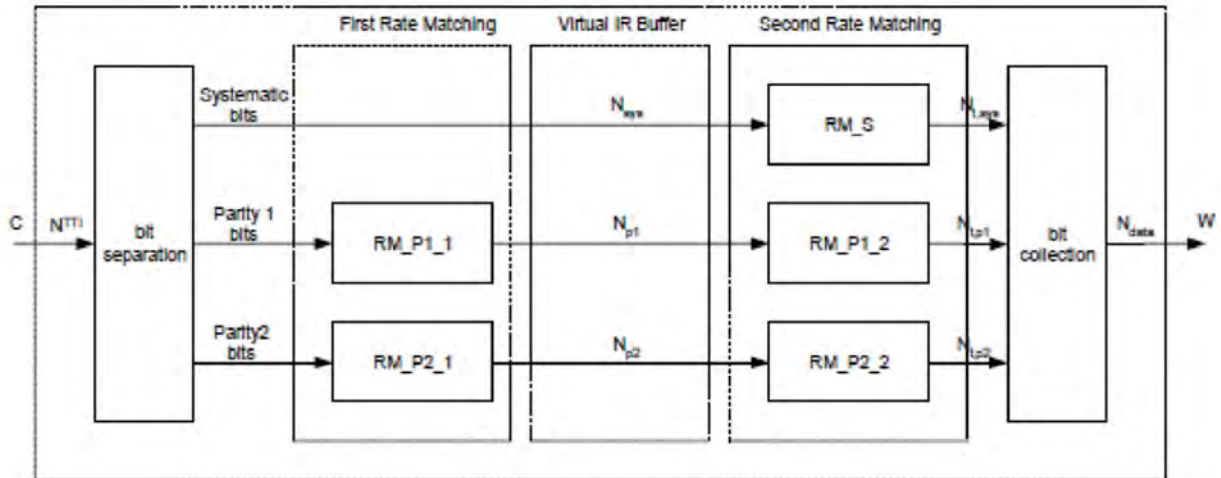


Figure 17: HS-DSCH hybrid ARQ functionality

4.5.4.1 HARQ bit separation

The HARQ bit separation function shall be performed in the same way as bit separation for turbo encoded TrCHs in 4.2.7.4 above.

4.5.4.2 HARQ First Rate Matching Stage

HARQ first stage rate matching for the HS-DSCH transport channel shall be done with the general method described in 4.2.7.2.2.3 above with the following specific parameters.

The maximum number of soft channel bits available in the virtual IR buffer is N_{IR} which is signalled from higher layers for each HARQ process. The number of coded bits in a TTI before rate matching is N^{TTI} this is deduced from information signalled from higher layers and parameters signalled on the HS-SCCH for each TTI. Note that HARQ processing and physical layer storage occurs independently for each HARQ process currently active.

If N_{IR} is greater than or equal to N^{TTI} (i.e. all coded bits of the corresponding TTI can be stored) the first rate matching stage shall be transparent. This can, for example, be achieved by setting $e_{rmms} = 0$. Note that no repetition is performed.

If N_{IR} is smaller than N^{TTI} the parity bit streams are punctured as in 4.2.7.2.2.3 above by setting the rate matching parameter $\Delta N_{il}^{TTI} = N_{IR} - N^{TTI}$ where the subscripts i and l refer to transport channel and transport format in the referenced sub-clause. Note the negative value is expected when the rate matching implements puncturing. Bits selected for puncturing which appear as δ in the algorithm in 4.2.7 above shall be discarded and not counted in the totals for the streams through the virtual IR buffer.

4.5.4.3 HARQ Second Rate Matching Stage

HARQ second stage rate matching for the HS-DSCH transport channel shall be done with the general method described in 4.2.7.5 above with the following specific parameters. Bits selected for puncturing which appear as δ in the algorithm in 4.2.7.5 above shall be discarded and are not counted in the streams towards the bit collection.

The parameters of the second rate matching stage depend on the value of the RV parameters s and r . The parameter s can take the value 0 or 1 to distinguish between transmissions that prioritise systematic bits ($s = 1$) and non systematic bits ($s = 0$). The parameter r (range 0 to $r_{max}-1$) changes the initial error variable e_{int} in the case of puncturing. In case of repetition both parameters r and s change the initial error variable e_{int} . The parameters X_i , e_{plus} and e_{minus} are calculated as per table 10 below.

Denote the number of bits before second rate matching as N_{sys} for the systematic bits, N_{p1} for the parity 1 bits, and N_{p2} for the parity 2 bits, respectively. Denote the number of physical channels used for the HS-DSCH by P . N_{data} is the number of bits available to the HS-DSCH in one TTI and defined as $N_{data} = P \times 3 \times N_{data1}$, where N_{data1} is defined in [2]. The rate matching parameters are determined as follows.

Denote the number of bits before second rate matching as N_{sys} for the systematic bits, N_{p1} for the parity 1 bits, and N_{p2} for the parity 2 bits, respectively. Denote the number of physical channels used for the HS-DSCH by P . N_{data} is the number of bits available to the HS-DSCH in one TTI and defined as $N_{data} = P \times 3 \times N_{data1}$, where N_{data1} is defined in [2]. The rate matching parameters are determined as follows.

For $N_{data} \leq N_{sys} + N_{p1} + N_{p2}$, puncturing is performed in the second rate matching stage. The number of transmitted systematic bits in a transmission is $N_{t,sys} = \min\{N_{sys}, N_{data}\}$ for a transmission that prioritises systematic bits and $N_{t,sys} = \max\{N_{data} - (N_{p1} + N_{p2}), 0\}$ for a transmission that prioritises non systematic bits.

For $N_{data} > N_{sys} + N_{p1} + N_{p2}$ repetition is performed in the second rate matching stage. A similar repetition rate in all bit streams is achieved by setting the number of transmitted systematic bits to $N_{t,sys} = \left\lfloor N_{sys} \cdot \frac{N_{data}}{N_{sys} + 2N_{p1}} \right\rfloor$.

The number of parity bits in a transmission is: $N_{t,p1} = \left\lfloor \frac{N_{data} - N_{t,sys}}{2} \right\rfloor$ and $N_{t,p2} = \left\lfloor \frac{N_{data} - N_{t,sys}}{2} \right\rfloor$ for the parity 1 and parity 2 bits, respectively.

Table 10 below summarizes the resulting parameter choice for the second rate matching stage.

Table 10: Parameters for HARQ second rate matching

	X_i	e_{plus}	e_{minus}
Systematic RM S	N_{sys}	N_{sys}	$ N_{sys} - N_{t,sys} $
Parity 1 RM P1_2	N_{p1}	$2 \cdot N_{p1}$	$2 \cdot N_{p1} - N_{t,p1} $
Parity 2 RM P2_2	N_{p2}	N_{p2}	$ N_{p2} - N_{t,p2} $

The rate matching parameter e_{int} is calculated for each bit stream according to the RV parameters r and s using

$e_{int}(r) = \left\{ \left(X_i - \left\lfloor r \cdot e_{plus} / r_{max} \right\rfloor - 1 \right) \bmod e_{plus} \right\} + 1$ in the case of puncturing, i.e., $N_{data} \leq N_{sys} + N_{p1} + N_{p2}$, and

$e_{int}(r) = \left\{ \left(X_i - \left\lfloor (s + 2 \cdot r) \cdot e_{plus} / (2 \cdot r_{max}) \right\rfloor - 1 \right) \bmod e_{plus} \right\} + 1$ for repetition, i.e.,

$N_{data} > N_{sys} + N_{p1} + N_{p2}$. Where $r \in \{0, 1, \dots, r_{max} - 1\}$ and r_{max} is the total number of redundancy versions allowed by varying r as defined in 4.6.2. Note that r_{max} varies depending on the modulation mode, i.e. for 16QAM $r_{max} = 2$ and for QPSK $r_{max} = 4$.

Note: For the modulo operation the following clarification is used: the value of $(x \bmod y)$ is strictly in the range of 0 to $y-1$ (i.e. $-1 \bmod 10 = 9$).

4.5.4.4 HARQ bit collection

The HARQ bit collection is achieved using a rectangular interleaver of size $N_{row} \times N_{col}$.

The number of rows and columns are determined from:

$$N_{row} = 4 \text{ for 16QAM and } N_{row} = 2 \text{ for QPSK}$$

$$N_{col} = N_{data} / N_{row}$$

where N_{data} is used as defined in 4.5.4.3.

Data is written into the interleaver column by column, and read out of the interleaver column by column starting from the first column.

N_{Lsys} is the number of transmitted systematic bits. Intermediate values N_s and N_c are calculated using:

$$N_s = \left\lfloor \frac{N_{Lsys}}{N_{col}} \right\rfloor \text{ and } N_c = N_{Lsys} - N_s \cdot N_{col}$$

If $N_c=0$ and $N_r > 0$, the systematic bits are written into rows $1 \dots N_r$.

Otherwise systematic bits are written into rows $1 \dots N_r - 1$ in the first N_c columns and, if $N_r > 0$, also into rows $1 \dots N_r$ in the remaining $N_{col} - N_c$ columns.

The remaining space is filled with parity bits. The parity bits are written column wise into the remaining rows of the respective columns. Parity 1 and 2 bits are written in alternating order, starting with a parity 2 bit in the first available column with the lowest index number.

In the case of 16QAM for each column the bits are read out of the interleaver in the order row 1, row 2, row 3, row 4. In the case of QPSK for each column the bits are read out of the interleaver in the order row1, row2.

4.5.5 Physical channel segmentation for HS-DSCH

When more than one HS-PDSCH is used, physical channel segmentation divides the bits among the different physical channels. The bits input to the physical channel segmentation are denoted by $w_1, w_2, w_3, \dots, w_R$, where R is the number of bits input to the physical channel segmentation block. The number of PhCHs is denoted by P .

The bits after physical channel segmentation are denoted $u_{p,1}, u_{p,2}, u_{p,3}, \dots, u_{p,U}$, where p is PhCH number and U is the

number of bits in one radio sub-frame for each HS-PDSCH, i.e. $U = \frac{R}{P}$. The relation between w_k and $u_{p,k}$ is given below.

For all modes, some bits of the input flow are mapped to each code until the number of bits on the code is U .

Bits on first PhCH after physical channel segmentation:

$$u_{1,k} = w_k \quad k = 1, 2, \dots, U$$

Bits on second PhCH after physical channel segmentation:

$$u_{2,k} = w_{k+U} \quad k = 1, 2, \dots, U$$

...

Bits on the P^{th} PhCH after physical channel segmentation:

$$u_{P,k} = w_{k+(P-1)U} \quad k = 1, 2, \dots, U$$

4.5.6 Interleaving for HS-DSCH

The interleaving for FDD is done as shown in figure 18 below, separately for each physical channel. The bits input to the block interleaver are denoted by $u_{p,1}, u_{p,2}, u_{p,3}, \dots, u_{p,U}$, where p is PhCH number and U is the number of bits in one TTI for one PhCH. For QPSK $U = 960$ and for 16QAM $U = 1920$. The basic interleaver is as the 2nd interleaver described in Section 4.2.11. The interleaver is of fixed size: $R2=32$ rows and $C2=30$ columns.

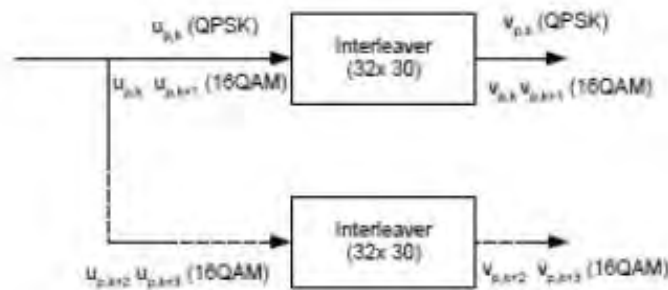


Figure 18: Interleaver structure for HS-DSCH

For 16QAM, there are two identical interleavers of the same fixed size $R2 \times C2 = 32 \times 30$. The output bits from the physical channel segmentation are divided two by two between the interleavers: bits $u_{p,k}$ and $u_{p,k+1}$ go to the first interleaver and bits $u_{p,k+2}$ and $u_{p,k+3}$ go to the second interleaver. Bits are collected two by two from the interleavers: bits $v_{p,k}$ and $v_{p,k+1}$ are obtained from the first interleaver and bits $v_{p,k+2}$ and $v_{p,k+3}$ are obtained from the second interleaver, where $k \bmod 4 = 1$.

TS 25.212 §§ 4.5.4-4.5.6 “Hybrid ARQ for HS-DSCH,” “Physical channel segmentation for HS-DSCH,” and “Interleaving for HS-DSCH.”

6. Configuration Messages for Inter-Operability

161. It is a basic principle well understood by a POSITA that whenever two transceivers have multiple communication configurations from which to choose, messages can be sent from one transceiver to another to establish which specific configuration will be employed. Indeed, such configuration messages are typically the first and only approach considered to solve the problem of ensuring that the transceivers agree on which configuration is being used. Such configuration messages had been employed in numerous systems well before the date of the alleged inventions.

162. As an example, consider U.S. Patent No. 4,924,456 entitled “High Speed Modem” filed August 29, 1988 and issued May 8, 1990. This patent describes a modem that uses configuration messages to change its speed (data rate) based on how much data awaits transmission based on the contents of a transmit data buffer and also based on the quality of the telephone line based on how many retransmissions are being requested. Below is the abstract for U.S. Patent No. 4,924,456:

[57]

ABSTRACT

A modem is disclosed having a data transmission protocol involving lower-speed, full-duplex operation during the connect sequence with a remote modem and an automatic switch to higher-speed, half-duplex operation for data transfer. Further, the modem data transmission involves transparently changing between lower-speed, interactive operation and higher-speed operation based upon data transmission demands. The operation is controlled by a processor monitoring the contents of a transmit data buffer and providing a mode control command to the modem transmitter. The modem also adapts its speed to the quality of the telephone line by fallback or fallforward to a different speed based upon predetermined data frame retransmission criteria.

163. U.S. Patent No. 4,924,456 explicitly describes the “handshake” process which includes the exchange of configuration messages that allow for “the communication of a number

of parameters” in 4:9-35 shown below. Note in particular the statement in 4:28-31 that “The various exchanges between an originating modem and an answering modem in the handshake sequence and in linking are well known to those of skill in the modem art.” This statement affirms that at least by 1988 the use of configuration messages was well known to a POSITA.

10 It is to be understood that the modem also places signals
on the telephone line that allow the answering modem
to link with the originating modem. This involves com-
munication of a number of parameters that allow the
receiver to establish carrier detection, adjust automatic
15 gain control circuits, establish timing synchronization,
converge and adapt the equalizer to the initial line con-
ditions, and synchronize the descrambler. Also, in the
initial linking procedure, referred to in the art as the
“handshake” sequence, tones will be placed on the tele-
20 phone line.
In addition to accepting digital data from the data
terminal for communication, i.e., TXD, and providing
received digital data to the data terminal, i.e., RXD,
there may be an exchange between the modem and the
25 data terminal of various control signals such as data
terminal ready (DTR), request to send (RTS), data set
ready (DSR), clear to send (CTS) and Carrier (CXR).
The various exchanges between an originating modem
and an answering modem in the handshake sequence
30 and in linking are well known to those of skill in the
modem art. Similarly, the various exchanges between a
modem and the data terminal to which it is coupled
including the above-mentioned DTR, RTS, DSR, CXR
and CTS signals are also well-known to those of skill in
35 the modem art.

Figure 10 from U.S. Patent No. 4,924,456 illustrates the connect sequence for the modem described in the patent. That figure is reproduced as Figure 19 below.

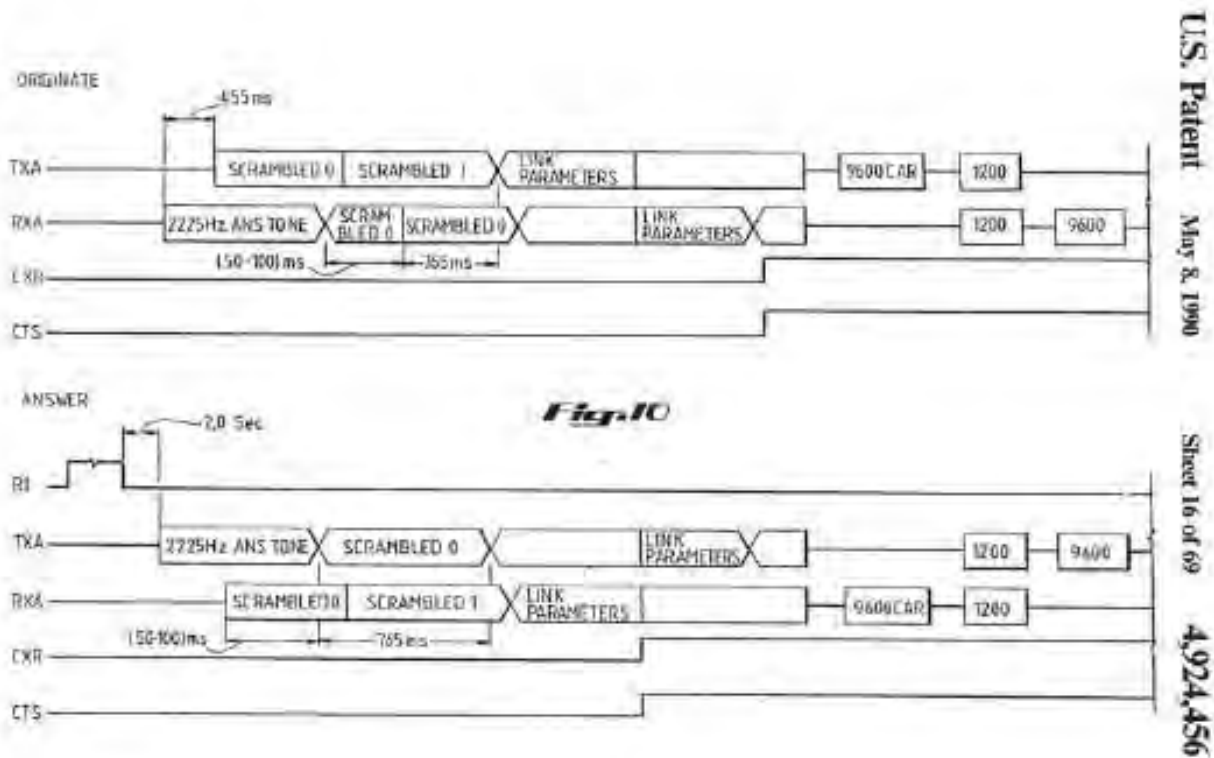


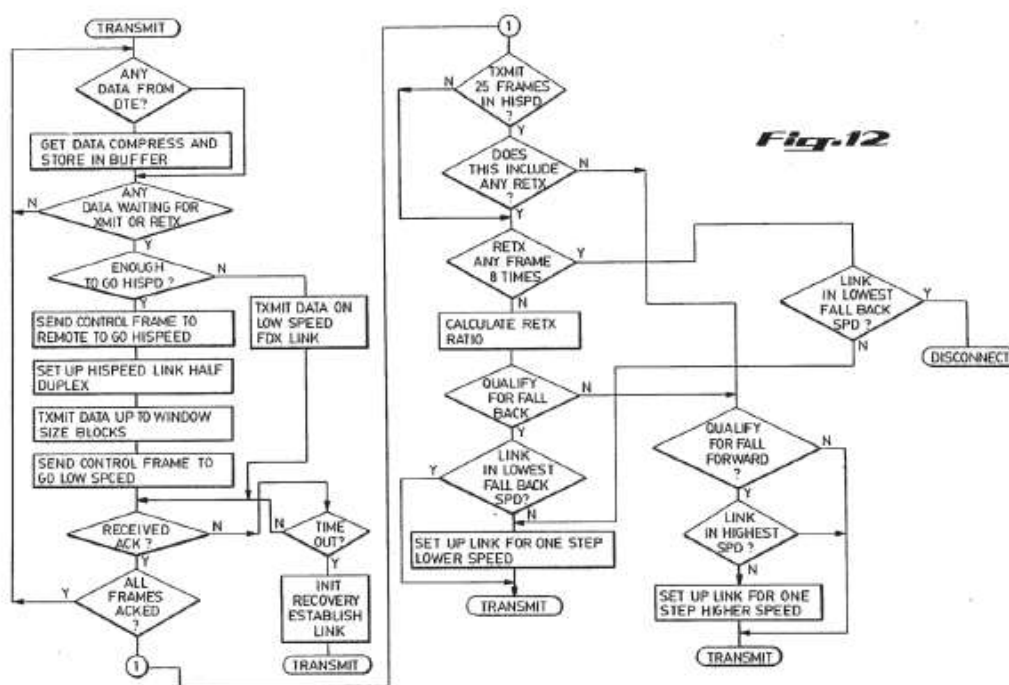
Figure 19: Figure 10 from U.S. Patent No. 4,924,456 shows the exchange of configuration messages (link parameters) during initialization of a modem.

164. The exchange of link parameters illustrated in Figure 19 is described in the patent at 18:33-40:

The modems then exchange link parameters for error control. The originate modem first sends approximately 23 bytes of link parameter information. The answer modem receives the link parameters and if any parameters need to be changed, it sends update link parameters to the originate modem. After the update link parameters are exchanged, the CRR and CTS lines are turned on at both modems.

165. Configuration messages are also understood by a POSITA to be useful for adapting the communications configuration to specific channel conditions and in consideration of available memory requirements. As described in U.S. Patent No. 4,924,456, configuration messages can be used after initialization to change the configuration adaptively. An example of this is provided by

Figure 12 of U.S. Patent No. 4,924,456, which is shown below as Figure 20. Note in the figure the blocks that indicate configuration messages: “SEND CONTROL FRAME TO REMOTE TO GO HISPEED,” “SEND CONTROL FRAME TO GO LOW SPEED,” “SET UP LINK FOR ONE STEP LOWER SPEED,” and “SET UP LINK FOR ONE STEP HIGHER SPEED.”



U.S. Patent May 8, 1990 Sheet 18 of 69 4,924,456

Figure 20: Figure 12 from U.S. Patent No. 4,924,456 shows the exchange of configuration messages (control frames) that change the configuration to high speed and then back to low speed based on the amount of user data stored in the buffer. The figure also shows adjusting the data rate dynamically based on the number of retransmissions and the retransmission ratio.

166. The use of configuration messages to adaptively change the configuration as illustrated in Figure 12 of U.S. Patent No. 4,924,456 “automatically changes between speeds of transmission based upon data transmission demand during a communication and upon the quality of the data transmissions.” See 19:31-34. As shown in the above figure, these automatic changes in transmission speed are initiated by configuration messages from the modem and the remote modem

to which it is transferring data. The above figure is described explicitly in lines 19:30-20:25 of U.S. Patent No. 4,924,456, which are excerpted below.

FIG. 12 flowcharts an operational sequence of the 30
modem involving data communication that automati-
cally changes between speeds of transmission based
upon data transmission demand during a communica-
tion and upon the quality of the data transmissions.
When the modem begins transmitting, it checks for data 35
from the data terminal. The data for transmission is
compressed and stored in the transmit data buffer. Next,
there is evaluation of whether any data is waiting for
retransmission. If so and there is enough to go higher-
speed, the modem sends control frame information to 40
the remote modem to go higher-speed. The higher-
speed, half-duplex link is set up and transmission of data
begins. At the conclusion of data transmission, a control
frame is sent requesting the remote modem to go into 45
the lower-speed mode. The modem then awaits recep-
tion of an acknowledgement. If the acknowledgement
does not come within a prescribed time out period, the
modem initiates recovery by establishing the link and
reentering the transmit sequence. If the request to go 50
lower-speed is acknowledged, the routine goes back to
the beginning of the transmit sequence.

If there is no acknowledgement that all data frames
were received properly, there is consideration of
whether a predetermined number of data frames have
been transmitted at higher-speed (e.g. 25 frames). If so, 55
there is a check to determine whether the transmitted
data frames included any retransmissions. If there was
retransmission of any data frame for predetermined
number of times (e.g. 8 times), there is a calculation of
the retransmit frame ratio. Based upon the calculated 60
ratio, a determination is made whether the quality of the
data transmission indicates qualification for a fallback in
speed. If there had been retransmission of any data
frame at least the predetermined number of times, and if
the modem is linked in the lowest fallback speed, then 65
the modem disconnects. If the link is not in the lowest
fallback speed, the modem sets up the link for one step
lower in speed and reenters the transmit sequence.

24,456

20

If the evaluation to determine whether a predeter-
mined number of preceding transmitted frames included
any retransmitted data indicates that none were retrans-
mitted frames, the routine makes an evaluation for qual-
ification to fallforward to a higher speed. If the qualifi-
cation is not met, the transmit sequence is reentered at
the same speed. But if the data transmission quality
qualifies for a fallforward in speed, and the link is not
already in the highest speed, the modem sets up the link
for a one step higher speed of transmission.

The routine in FIG. 12 illustrates the ability of the
modem to adapt its speed to the quality of the telephone
line. As indicated, the modem can either fallback to a
lower speed or fallforward to a higher speed. In gen-
eral, to determine the quality of the line, the processor
of the modem constantly monitors the number of errors
in data transmission as reflected by the number of re-
transmitted data frames. If the number of retransmis-
sions is high, indicating too many errors are encoun-
tered and line quality is poor, the modem drops down to
the next lower speed until an acceptable reduction in
errors in transmission is achieved. If the line quality
improves and the number of errors is reduced, the
modem will automatically fallforward to the next
higher speed.

167. Specific quotes from lines 19:30-20:25 of the '456 patent that describe configuration messages include: "[i]f so and there is enough to go higher-speed, the modem sends control frame information to the remote modem to go higher-speed," (19:39-41), "[a]t the conclusion of data transmission, a control frame is sent requesting the remote modem to go into the lower-speed mode," (19:43-45), "[i]f the link is not in the lowest fallback speed, the modem sets up the link for one step lower in speed and reenters the transmit sequence." (19:66-68), and "[b]ut if the data

transmission quality qualifies for a fallforward in speed, and the link is not already in the highest speed, the modem sets up the link for a one step higher speed of transmission (20:7-10).

168. Note that the cases described above in Figure 12 of the '456 patent and the accompanying text, these configuration changes are intimately related to how much memory is being utilized by certain buffers. The “data transmission demand” is determined by how much data is waiting in the transmit data buffer. The number of times a frame has been retransmitted and the retransmission ratio are both closely related to how much memory is required by the retransmission buffer. For example, in 8:37-39 of the '456 patent, the dependence of modem speed and memory pressure on the transmit data buffer is explicitly stated: “If the modem can keep its transmit data buffer from filling using the lower-speed mode, it will not go to high speed.” As another example, the choice of when to change speeds based on the retransmission behavior would be understood by s POSITA to be related to the number of unacknowledged packets (outstanding packets) that can be accommodated by the retransmission buffer. This memory allocation is discussed, for example, at 11:26-29: “The Error Control Mode provides for synchronous data packets operation and enables selective ARQ, dynamic fallback/fallforward, and data compression with an outstanding packet window of 12.”

169. Configuration messages in a multicarrier transmission system with multicast capability are found, for example, in 802.11a as described in the IEEE Std 802.11a-1999, which added OFDM capability to the 1997 802.11 standard. Sections 17.3.2 and 17.3.4, reproduced in part below, describe the physical layer convergence protocol (PLCP) frame format of the PLCP protocol data unit (PPDU), which includes configuration messages in the OFDM PLCP header also referred to in Figure 107 of 802.11a as the signal field (SIGNAL). The signal field provides

configuration information required for successful communication including the length of the transmission, the coding rate, and the type of modulation to be used in the rest of the packet.

17.3.2 PLCP frame format

Figure 107 shows the format for the PPDU including the OFDM PLCP preamble, OFDM PLCP header, PSDU, tail bits, and pad bits. The PLCP header contains the following fields: LENGTH, RATE, a reserved bit, an even parity bit, and the SERVICE field. In terms of modulation, the LENGTH, RATE, reserved bit, and parity bit (with 6 “zero” tail bits appended) constitute a separate single OFDM symbol, denoted SIGNAL, which is transmitted with the most robust combination of BPSK modulation and a coding rate of $R = 1/2$. The SERVICE field of the PLCP header and the PSDU (with 6 “zero” tail bits and pad bits appended), denoted as DATA, are transmitted at the data rate described in the RATE field and may constitute multiple OFDM symbols. The tail bits in the SIGNAL symbol enable decoding of the RATE and LENGTH fields immediately after the reception of the tail bits. The RATE and LENGTH are required for decoding the DATA part of the packet. In addition, the CCA mechanism can be augmented by predicting the duration of the packet from the contents of the RATE and LENGTH fields, even if the data rate is not supported by the station. Each of these fields is described in detail in 17.3.3, 17.3.4, and 17.3.5.

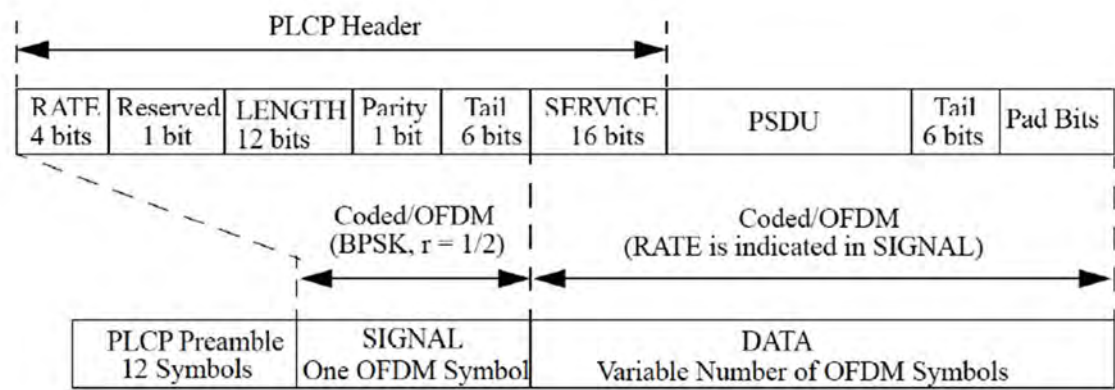


Figure 107—PPDU frame format

17.3.4 Signal field (SIGNAL)

The OFDM training symbols shall be followed by the SIGNAL field, which contains the RATE and the LENGTH fields of the TXVECTOR. The RATE field conveys information about the type of modulation and the coding rate as used in the rest of the packet. The encoding of the SIGNAL single OFDM symbol shall be performed with BPSK modulation of the subcarriers and using convolutional coding at $R = 1/2$. The encoding procedure, which includes convolutional encoding, interleaving, modulation mapping processes, pilot insertion, and OFDM modulation, follows the steps described in 17.3.5.5, 17.3.5.6, and 17.3.5.8, as used for transmission of data at a 6 Mbit/s rate. The contents of the SIGNAL field are not scrambled.

The SIGNAL field shall be composed of 24 bits, as illustrated in Figure 111. The four bits 0 to 3 shall encode the RATE. Bit 4 shall be reserved for future use. Bits 5–16 shall encode the LENGTH field of the TXVECTOR, with the least significant bit (LSB) being transmitted first.

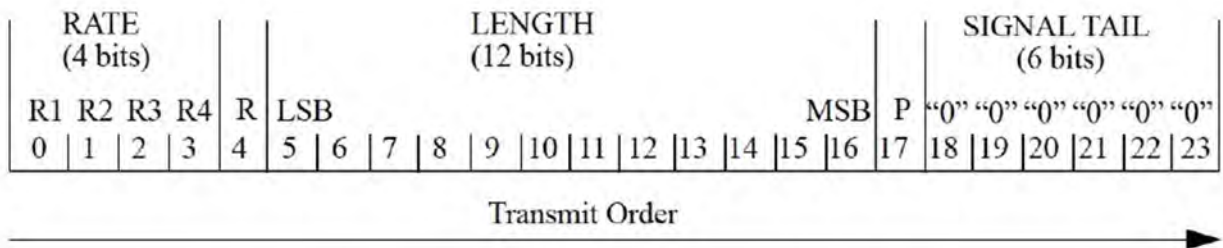


Figure 111 – SIGNAL field bit assignment

The process of generating the SIGNAL OFDM symbol is illustrated in Annex G (G.4).

17.3.4.1 Data rate (RATE)

The bits R1–R4 shall be set, dependent on RATE, according to the values in Table 80.

Table 80—Contents of the SIGNAL field

Rate (Mbits/s)	R1–R4
6	1101
9	1111
12	0101
18	0111
24	1001
36	1011
48	0001
54	0011

17.3.4.2 PLCP length field (LENGTH)

The PLCP length field shall be an unsigned 12-bit integer that indicates the number of octets in the PSDU that the MAC is currently requesting the PHY to transmit. This value is used by the PHY to determine the number of octet transfers that will occur between the MAC and the PHY after receiving a request to start transmission. The transmitted value shall be determined from the LENGTH parameter in the TXVECTOR issued with the PHY-TXSTART.request primitive described in 12.3.5.4 (IEEE Std 802.11, 1999 Edition). The LSB shall be transmitted first in time. This field shall be encoded by the convolutional encoder described in 17.3.5.5.

17.3.4.3 Parity (P), Reserved (R), and Signal tail (SIGNAL TAIL)

Bit 4 shall be reserved for future use. Bit 17 shall be a positive parity (even parity) bit for bits 0–16. The bits 18–23 constitute the SIGNAL TAIL field, and all 6 bits shall be set to zero.

170. Another example of configuration messages is provided in Moon 15:41-51. Moon at NOK00083406, 15:41-51. Moon teaches that the receiver has to understand each operation performed by the transmitter so that the receiver can perform the corresponding operation.

Meanwhile, the receiver should previously recognize information on the coding rate, the modulation technique, the orthogonal codes, and the number of orthogonal codes, used by the transmitter illustrated in FIG. 7, and the number of retransmissions, for demodulation and decoding. That is, the above information should be previously provided to the despreader 812, the demodulator 814, the selective packet combiner 816, and the decoder 824 so that the receiver can perform a corresponding operation of the transmitter. Therefore, the above information is provided from the transmitter to the receiver over a downlink control channel.

Moon at NOK00083406, 15:41-51.

171. Information that the transmitter and receiver exchange to agree on how they will communicate affects the allocation of shared memory between interleaving and retransmission in the system described by Moon. For example, depending on the coding rate applied, the relative inputs to the first and second interleaver memories can change. With a coding rate of 1/2, the first and second interleaver memories will receive a symmetric input, i.e., for each bit that goes into the

first interleaver, one bit will go into the second interleaver. Moon at NOK00083404, 12:12-14. But for a coding rate of 3/4, for every three bits that go into the first interleaver, only one bit will go into the second interleaver. Moon at NOK00083404, 12:15-18. Similarly, the choice of modulation technique will affect the amount of memory that must be allocated for retransmission.

172. Another example of configuration messages affecting how memory is allocated between functions is in Yap discussed further below in Section IX.I. Figures 5 and 6 from the patent are shown below, which describe two different approaches to retransmission that require different allocations of shared memory between transmission of new data and retransmission. As described in Yap, “FIG. 4 and FIG. 5 depict examples of the previous transmission status and the current retransmission arrangement for methods I and II.” Yap at NOK00084271, 5:33-35. Yap goes on to note, “For method II, attention has to be put to both transmission and receiving buffers due to the changed data setting. The transmitter needs to inform the receiver of the difference in arrangement between previous and current transmissions so that the buffers can be properly updated. This notification from the transmitter to the receiver is regulated by an upper layer.” Yap at NOK00084271, 5:61-6:1.

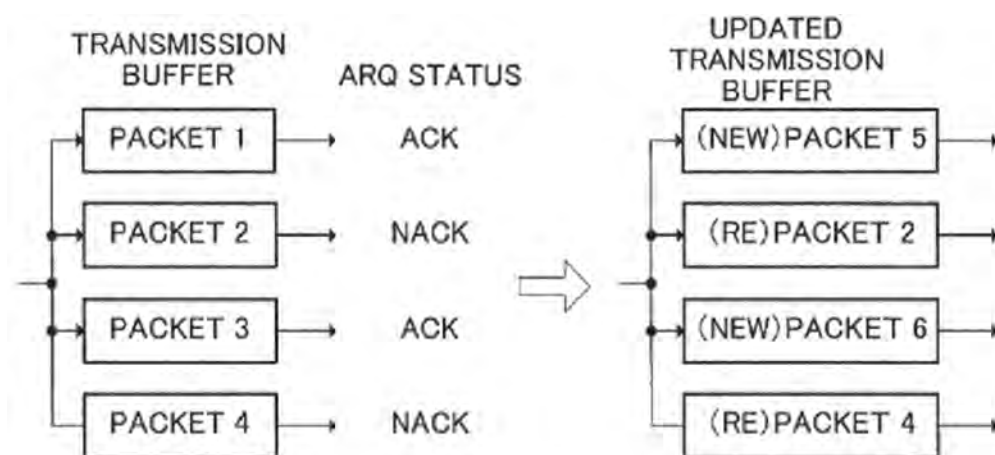


FIG.4

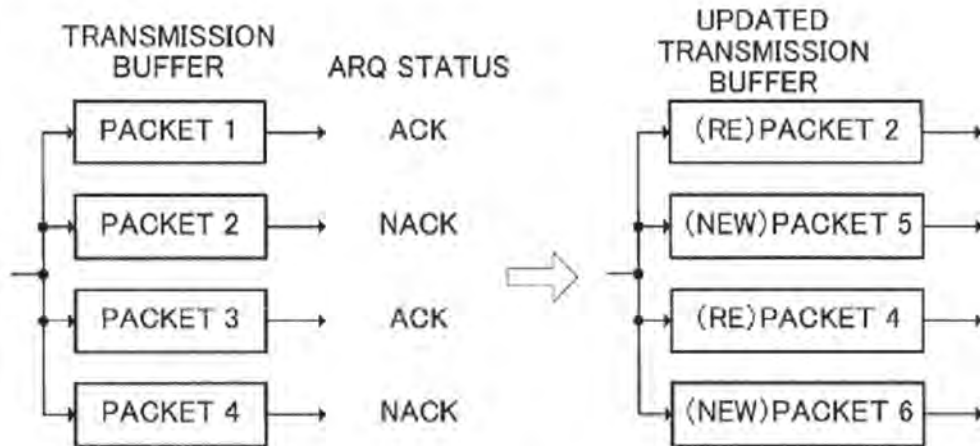


FIG.5

Yap at NOK00084267, Figs. 4-5.

173. Another example of configuration messages is the messages that would be used in the system disclosed by Mazzoni, messages between the two transceivers that would allow the system to adapt the sharing of memory between interleaving and deinterleaving. As disclosed in Mazzoni, “Still another object of the invention is to provide such an architecture which is adaptable, particularly in terms of the memory capacity of the interleaving and deinterleaving means, to suit a number of different bit rates selected from a predetermined group of bit rates.” Mazzoni at NOK00083793, 1:54-58.

174. Another example of a configuration message is the higher level signaling described in TS 25.212 that is used to make changes to rate matching.

4.2.7 Rate matching

Rate matching means that bits on a transport channel are repeated or punctured. Higher layers assign a rate-matching attribute for each transport channel. This attribute is semi-static and can only be changed through higher layer signalling. The rate-matching attribute is used when the number of bits to be repeated or punctured is calculated.

B. The 802.16 IEEE Air Interface for Fixed Broadband Wireless Access Systems

175. As discussed above, multicarrier transmission, Reed-Solomon encoding, interleaving, and retransmission were all well known by the time of the alleged invention. There

are examples of actual systems disclosed prior to the alleged invention where all four techniques are used together. An example of such a system is the 802.16 IEEE Air Interface for Fixed Broadband Wireless Access, as published on October 1, 2004.

176. Section 8.3 entitled “WirelessMAN-OFDM PHY” describes the use of multicarrier transmission, Reed-Solomon coding, and interleaving. Section 8.3.1.1 entitled “OFDM symbol description” describes how the inverse Fourier transform is used to create the OFDM waveform as excerpted below:

8.3.1.1 OFDM symbol description

8.3.1.1.1 Time domain

Inverse-Fourier-transforming creates the OFDM waveform; this time duration is referred to as the useful symbol time T_b . A copy of the last T_g of the useful symbol period, termed CP, is used to collect multipath, while maintaining the orthogonality of the tones. Figure 195 illustrates this structure.

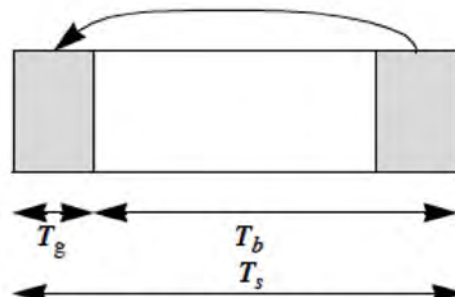


Figure 195—OFDM symbol time structure

177. Section 8.3.3 entitled “Channel coding” describes the use of Reed-Solomon coding and interleaving. Section 8.3.3.2 “FEC” includes a description of the Reed-Solomon code as excerpted below:

The Reed-Solomon encoding shall be derived from a systematic RS ($N = 255$, $K = 239$, $T = 8$) code using $GF(2^8)$,

where

N is the number of overall bytes after encoding,
 K is the number of data bytes before encoding,
 T is the number of data bytes which can be corrected.

The following polynomials are used for the systematic code:

Code Generator Polynomial: $g(x) = (x + \lambda^0)(x + \lambda^1)(x + \lambda^2) \dots (x + \lambda^{2T-1})$, $\lambda = 02_{HEX}$ (66)

Field Generator Polynomial: $p(x) = x^8 + x^4 + x^3 + x^2 + 1$ (67)

This code is shortened and punctured to enable variable block sizes and variable error-correction capability. When a block is shortened to K' data bytes, add $239 - K'$ zero bytes as a prefix. After encoding discard these $239 - K'$ zero bytes. When a codeword is punctured to permit T bytes to be corrected, only the first $2T$ of the total 16 parity bytes shall be employed. The bit/byte conversion shall be MSB first.

178. Section 8.3.3.3 entitled “Interleaving” discloses that all encoded data bits are interleaved by a block interleaver as excerpted below from page 441.

8.3.3.3 Interleaving

All encoded data bits shall be interleaved by a block interleaver with a block size corresponding to the number of coded bits per the allocated subchannels per OFDM symbol, N_{cbps} . The interleaver is defined by a two step permutation. The first ensures that adjacent coded bits are mapped onto nonadjacent subcarriers. The second permutation insures that adjacent coded bits are mapped alternately onto less or more significant bits of the constellation, thus avoiding long runs of lowly reliable bits.

Let N_{cpe} be the number of coded bits per subcarrier, i.e., 1, 2, 4 or 6 for BPSK, QPSK, 16-QAM, or 64-QAM, respectively. Let $s = \text{ceil}(N_{cpe}/2)$. Within a block of N_{cbps} bits at transmission, let k be the index of the coded bit before the first permutation; m_k be the index of that coded bit after the first and before the second permutation; and let j_k be the index after the second permutation, just prior to modulation mapping.

The first permutation is defined by Equation (71):

$$m_k = (N_{cbps}/12) \cdot k_{\text{mod}12} + \text{floor}(k/12) \quad k = 0, 1, \dots, N_{cbps} - 1 \quad (71)$$

179. Section 8.3.3.3 continues to describe the interleaving on page 442, which is excerpted below. As described specifically in Table 223 of the excerpt, there are a variety of interleavers that can be implemented depending on the number of subchannels (1, 2, 4, 8, or 16)

and the modulation (BPSK, QPSK, 16-QAM, or 64-QAM). As the table shows, there is a specific interleaver block size N_{cbps} that indicates the memory requirement associated with each interleaving configuration.

The second permutation is defined by Equation (72):

$$j_k = z \cdot \text{floor}(m_k/z) + (m_k + N_{cbps} - \text{floor}(12 \cdot m_k / N_{cbps})) \bmod(z) \quad k = 0, 1, \dots, N_{cbps} - 1 \quad (72)$$

The de-interleaver, which performs the inverse operation, is also defined by two permutations. Within a received block of N_{cbps} bits, let j be the index of a received bit before the first permutation; m_j be the index of that bit after the first and before the second permutation; and let k_j be the index of that bit after the second permutation, just prior to delivering the block to the decoder.

The first permutation is defined by Equation (73):

$$m_j = z \cdot \text{floor}(j/z) + (j + \text{floor}(12 \cdot j / N_{cbps})) \bmod(z) \quad j = 0, 1, \dots, N_{cbps} - 1 \quad (73)$$

The second permutation is defined by Equation (74):

$$k_j = 12 \cdot m_j - (N_{cbps} - 1) \cdot \text{floor}(12 \cdot m_j / N_{cbps}) \quad j = 0, 1, \dots, N_{cbps} - 1 \quad (74)$$

The first permutation in the de-interleaver is the inverse of the second permutation in the interleaver, and conversely.

Table 223 shows the bit interleaver sizes as a function of modulation and coding.

Table 223—Block sizes of the Bit Interleaver

	Default (16 subchannels)	8 subchannels	4 subchannels	2 subchannels	1 subchannel
	N_{cbps}				
BPSK	192	96	48	24	12
QPSK	384	192	96	48	24
16-QAM	768	384	192	96	48
64-QAM	1152	576	288	144	72

The first bit out of the interleaver shall map to the MSB in the constellation.

180. Section 8.3.3.4 is excerpted below and describes the four modulation types, which are BPSK, QPSK, 16-QAM, and 64-QAM.

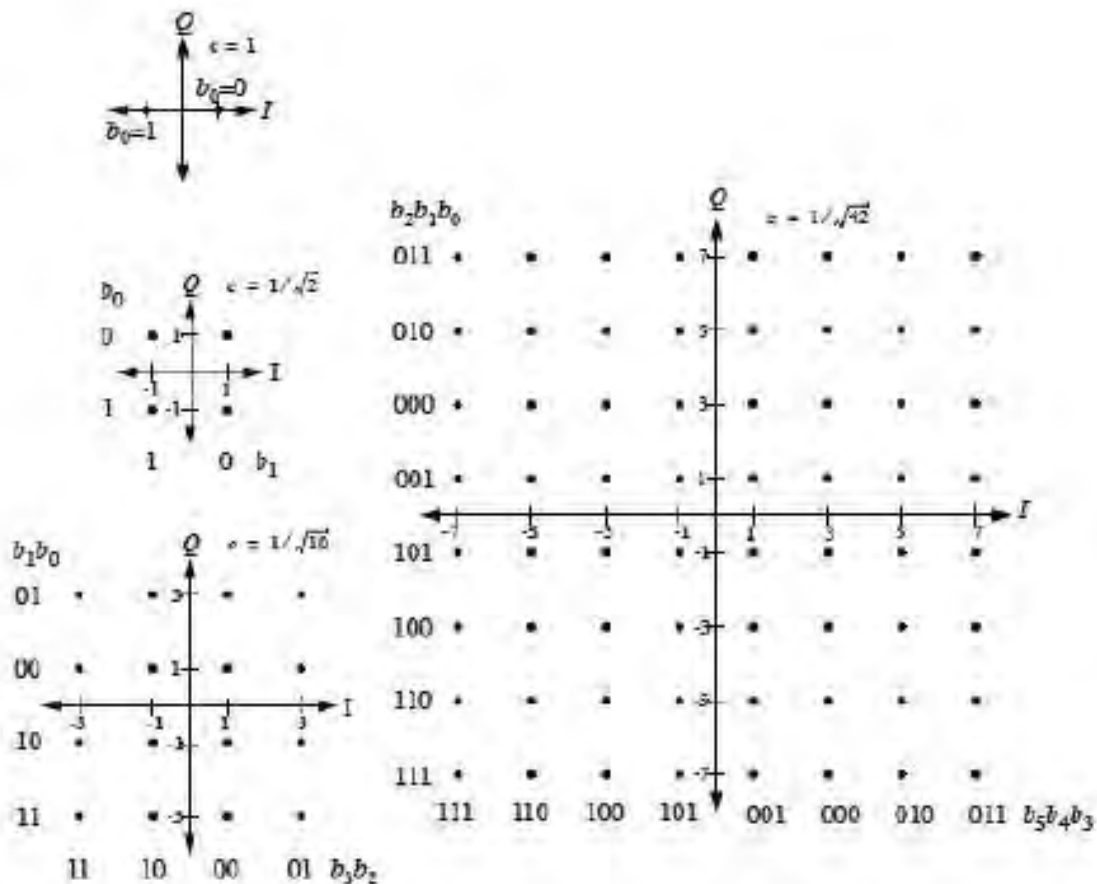


Figure 203—BPSK, QPSK, 16-QAM, and 64-QAM constellations

The constellation-mapped data shall be subsequently modulated onto all allocated data subcarriers in order of increasing frequency offset index. The first symbol out of the data constellation mapping shall be modulated onto the allocated subcarrier with the lowest frequency offset index.

181. Section 6.3.4 entitled “ARQ Mechanism” describes the retransmission protocol of 802.16. Below is an excerpt from the beginning of Section 6.3.4 on page 128.

6.3.4 ARQ mechanism

ARQ shall not be used with the PHY specification defined in 8.1.

The ARQ mechanism is a part of the MAC, which is optional for implementation. When implemented, ARQ may be enabled on a per-connection basis. The per-connection ARQ shall be specified and negotiated during connection creation. A connection cannot have a mixture of ARQ and non-ARQ traffic. Similar to other properties of the MAC protocol the scope of a specific instance of ARQ is limited to one unidirectional connection.

For ARQ-enabled connections, enabling of fragmentation is optional. When fragmentation is enabled, the transmitter may partition each SDU into fragments for separate transmission based on the value of the ARQ_BLOCK_SIZE parameter. When fragmentation is not enabled, the connection shall be managed as if fragmentation was enabled. In this case, regardless of the negotiated block size, each fragment formed for transmission shall contain all the blocks of data associated with the parent SDU.

The ARQ feedback information can be sent as a standalone MAC management message on the appropriate basic management connection, or piggybacked on an existing connection. ARQ feedback cannot be fragmented. The implementation of ARQ is optional.

182. Note that the standard states explicitly that ARQ shall not be used with the PHY specification defined in 8.1. The PHY specification defined in 8.1 entitled “WirelessMAN-SC PHY specification” is the single carrier PHY which predated the OFDM PHY described in Section 8.3. The ARQ mechanism of 6.3.4 is explicitly disclosed as applying to the PHY described in Section 8.3, which employs multicarrier transmission, Reed-Solomon coding, and interleaving in Table 1 of Section 1.3.4 “Air interface nomenclature and PHY compliance,” which is shown below:

Table 1 – Air Interface nomenclature

Designation	Applicability	PHY	Additional MAC requirements	Options	Duplexing alternative
WirelessMAN-SC™	10–66 GHz	8.1			TDD FDD
WirelessMAN-SCa™	Below 11 GHz licensed bands	8.2		AAS (6.3.7.6) ARQ (6.3.4) STC (8.2.1.4.3)	TDD FDD
WirelessMAN-OFDM™	Below 11 GHz licensed bands	8.3		AAS (6.3.7.6) ARQ (6.3.4) Mesh (6.3.6.6) STC (8.3.8)	TDD FDD
WirelessMAN-OFDMA	Below 11 GHz licensed bands	8.4		AAS (6.3.7.6) ARQ (6.3.4) STC (8.4.8)	TDD FDD
WirelessHUMAN™	Below 11 GHz license-exempt bands	[8.2, 8.3, or 8.4] and 8.5	DFS (6.3.15)	AAS (6.3.7.6) ARQ (6.3.4) Mesh (6.3.6.6) (with 8.3 only) STC (8.2.1.4.3/8.3.8/8.4.8)	TDD

183. Configuration messages are used to establish the specific 802.16 ARQ protocol. As stated in the excerpt of Section 6.3.4 below, “The per-connection ARQ shall be specified and negotiated during connection creation.”

184. The use of configuration messages to establish per-connection ARQ parameters is discussed further in Section 6.3.4.5 entitled “ARQ-enabled connection setup and negotiation,” which is excerpted below:

6.3.4.5 ARQ-enabled connection setup and negotiation

Connections are set up and defined dynamically through the DSA/DSC class of messages. CRC-32 shall be used for error detection of PDUs for all ARQ-enabled connections. All the ARQ parameters (see 6.3.4.3) shall be set when an ARQ-enabled connection is set up. The transmitter and receiver variables (defined in 6.3.4.4.1) shall be reset on connection setup.

185. As described in the excerpt from 6.3.4.5, all ARQ parameters described in Section 6.3.4.3 entitled “ARQ parameters” are set when an ARQ connection is set up. Below is an excerpt Section 6.3.4.3 from page 132. Note that these parameters fully describe how the ARQ will operate for this connection so that a transceiver implementing 802.16 on a device using shared memory would know how much memory to allocate to retransmission. Note also that this memory

requirement is not fixed and depends on the specific parameters exchanged by configuration messages during connection setup.

6.3.4.3 ARQ parameters

6.3.4.3.1 ARQ_BSN_MODULUS

ARQ_BSN_MODULUS is equal to the number of unique BSN values, i.e., 2^{11} .

6.3.4.3.2 ARQ_WINDOW_SIZE

ARQ_WINDOW_SIZE is the maximum number of unacknowledged ARQ blocks at any given time. An ARQ block is unacknowledged if it has been transmitted but no acknowledgment has been received.

ARQ_WINDOW_SIZE shall be less than or equal to half of the *ARQ_BSN_MODULUS*.

6.3.4.3.3 ARQ_BLOCK_LIFETIME

ARQ_BLOCK_LIFETIME is the maximum time interval an ARQ block shall be managed by the transmitter ARQ state machine, once initial transmission of the block has occurred. If transmission (or subsequent retransmission) of the block is not acknowledged by the receiver before the time limit is reached, the block is discarded.

6.3.4.3.4 ARQ_RETRY_TIMEOUT

ARQ_RETRY_TIMEOUT is the minimum time interval a transmitter shall wait before retransmission of an unacknowledged block for retransmission. The interval begins when the ARQ block was last transmitted.

6.3.4.3.5 ARQ_SYNC_LOSS_TIMEOUT

ARQ_SYNC_LOSS_TIMEOUT is the maximum time interval *ARQ_TX_WINDOW_START* or *ARQ_RX_WINDOW_START* shall be allowed to remain at the same value before declaring a loss of synchronization of the sender and receiver state machines when data transfer is known to be active. The ARQ receiver and transmitter state machines manage independent timers. Each has its own criteria for determining when data transfer is "active" (see 6.3.4.6.2 and 6.3.4.6.3).

6.3.4.3.6 ARQ_RX_PURGE_TIMEOUT

ARQ_RX_PURGE_TIMEOUT is the time interval the receiver shall wait after successful reception of a block that does not result in advancement of *ARQ_RX_WINDOW_START*, before advancing *ARQ_RX_WINDOW_START* (see 6.3.4.6.3).

6.3.4.3.7 ARQ_BLOCK_SIZE

ARQ_BLOCK_SIZE is the length used for partitioning an SDU into a sequence of ARQ blocks prior to transmission (see 6.3.4.1).

186. The basic connection that carries the ARQ Feedback message is defined in Section 4 entitled “Definitions” as excerpted below:

6.3.2.3.30 ARQ Feedback message

A system supporting ARQ shall be able to receive and process the ARQ Feedback message.

The ARQ Feedback message, as shown in Table 59, can be used to signal any combination of different ARQ ACKs (cumulative, selective, selective with cumulative). The message shall be sent on the appropriate basic management connection.

Table 59— ARQ Feedback message format

Syntax	Size	Notes
ARQ_Feedback_Message_Format() {		

Table 59— ARQ Feedback message format (continued)

Syntax	Size	Notes
Management Message Type =33	8 bits	
ARQ_Feedback_Payload	variable	See 6.3.3.4.3.
}		

ARQ_Feedback_Payload field shall be either sent using this ARQ Feedback message or by packing (“piggy-backing”) the ARQ_Feedback_Payload as described in 6.3.3.4.3.

187. The basic connection that carries the ARQ Feedback message is defined in Section 4 entitled “Definitions” as excerpted below:

3.6 basic connection: Connection that is established during subscriber station (SS) initial ranging and used to transport delay-intolerant medium access control (MAC) management messages.

188. Section 6.3.1.1, excerpted below from page 33 of 802.16, provides a further description of how MAC management messages are sent using the basic connection. In particular, it states that, “The basic connection is used by the BS MAC and SS MAC to exchange short, time-

urgent MAC management messages.” This is an example of the fact the 802.16 recognized the need for rapid conveyance of the feedback messages, the ACKs, used to control retransmission.

6.3 Data/Control plane

6.3.1 Addressing and connections

6.3.1.1 PMP

Each SS shall have a 48-bit universal MAC address, as defined in IEEE Std 802[®]-2001. This address uniquely defines the SS from within the set of all possible vendors and equipment types. It is used during the initial ranging process to establish the appropriate connections for an SS. It is also used as part of the authentication process by which the BS and SS each verify the identity of the other.

Connections are identified by a 16-bit CID. At SS initialization, two pairs of management connections (uplink and downlink) shall be established between the SS and the BS and a third pair of management connections may be optionally generated. The three pairs of connections reflect the fact that there are inherently three different levels of QoS for management traffic between an SS and the BS. The **basic connection** is used by the BS MAC and SS MAC to exchange short, time-urgent MAC management messages. The primary management connection is used by the BS MAC and SS MAC to exchange longer, more delay-tolerant MAC management messages. Table 14 specifies which MAC Management messages are transferred on which of these two connections. Finally, the Secondary Management Connection is used by the BS and SS to transfer delay tolerant, standards-based [Dynamic Host Configuration Protocol (DHCP), Trivial File Transfer Protocol (TFTP), SNMP, etc.] messages. These messages are carried in IP datagrams, as specified in 5.2.6. Messages carried on the Secondary Management Connection may be packed and/or fragmented. For the SCa, OFDM, and OFDMA PHY layers, management messages shall have CRC. Use of the secondary management connection is required only for managed SS.

189. A PDU is a protocol data unit. An SDU is a service data unit. SS is a subscriber station. TLV stands for type/length/value.

190. Based on the standard practices at the time, a POSITA would have known to implement a communications transceiver such as described by the 2004 IEEE 802.16 standard using processors with shared memory. Indeed, Intel devoted the entire August 20, 2004 issue of the Intel Technical Journal to WiMAX 802.16. The last article in the issue, entitled “Fully Integrated CMOS Radios from RF to Millimeter Wave Frequencies” discusses the concepts of System-on-a-Chip and System-on-a-Package which feature programmable processors with shared memory that implement the transceiver functions. Evolving standards were discussed as a reason that programmability is needed. Figures 24 and 25 of that article illustrate this use of processors with shared memory. Note

in particular the modules “Flex DSP” and “MEM” in Figure 25 referring to a digital signal processor and its shared memory.

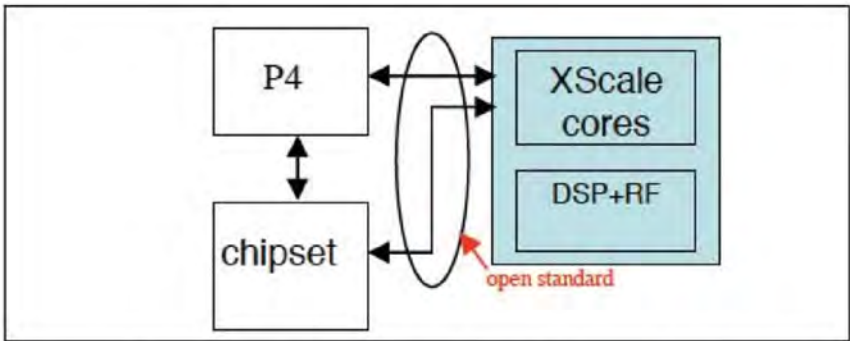


Figure 24: Flexible radios: a PC platform for merging computing and communications under always-evolving communication standards

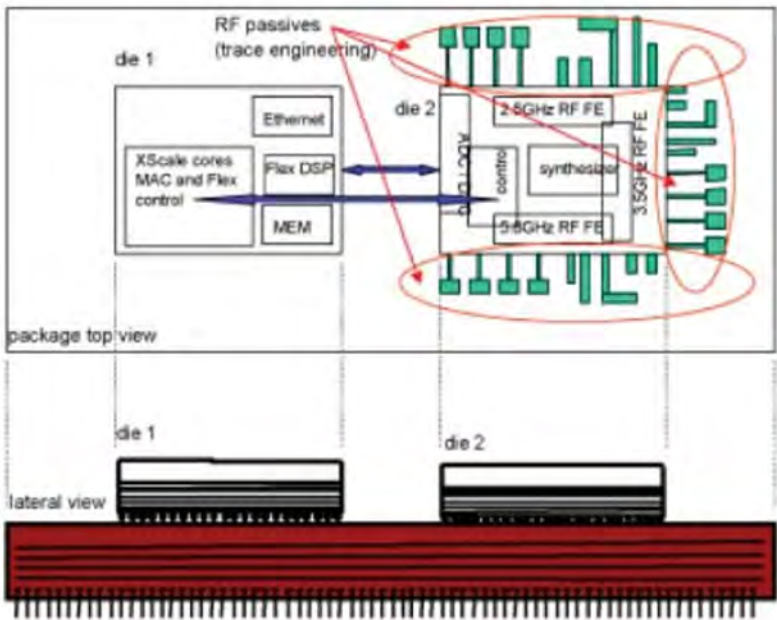


Figure 25: A flexible radio realization: top and lateral view of a realization with two dies and one package. More package surface for the RF passive network.

191. Below are excerpts from this article describing the concept of flexible, programmable computing that would necessarily feature use of a flexible shared memory.

Flexible Radios: a Practical Vision

The CMOS-based computer industry takes full advantage of CMOS scaling to produce always changing ever more powerful computing platforms. This was thought to be in fundamentally stark contrast to the standards-driven communications industry.

Despite that, the reality of recent years and plans for the foreseeable future appear to show a path for further integration of computing and communications. The apparent paradox disappears when one considers communication standards are always evolving documents.

The standards themselves generate new standards and addenda to standards are always being made. Standards serve as guidelines for products, and these products at a given time only attend to a limited subset of recommendations in the standards to warrant a compatibility stamp like “Wi-Fi” or “WiMAX.” Other parts of the standards are left for future implementation. In such a state of affairs, computing and communications do share the always-evolving aspects that are the spirit of the CMOS business model.

At the implementation level, multiband, multiprotocol radio for always-evolving communication standards is too complex a system to fit in a Pentium or chipset die. It would be probably better to start thinking that the PC motherboard will be a multiprocessor platform whose ecosystem will be populated by new chips besides the Pentium and chipset.

Figure 24 illustrates in a highly simplified abstraction the addition of a flexible radio to a multiprocessor PC platform. Note the suggestion that Intel® XScale® technology should become the general-purpose processor to handle all aspects of reprogrammability and hardware switchability of the flexible radio. XScale would be also in charge of a software-MAC (for easy reprogrammability) and all network layers of operation for the radio.

Figure 25 depicts a flexible radio concept and its realization as one package and two silicon chips. This concept exploits all the CMOS technology and package

advancements discussed in this paper. Note that the two chips on the same package is only a suggestion, not the only possible solution. Nevertheless, this particular configuration allows for more surfaces on the package onto which the RF passives can be realized.

Note the addition of an Ethernet interface in die 1 of this flexible radio realization (Figure 25) allows for easy access and reprogrammability/reconfigurability in case this flexible radio is taken out of the multi-processor PC platform and placed as a stand alone core component of radio base stations.

192. The 2004 802.16 IEEE standard was indeed implemented using the “system-on-a-chip” concept by Intel with the Intel 5116, which is a multicarrier communications transceiver using shared memory. The Intel 5116 was described as “the first 802.16-2004 system-on-a-chip optimized for cost-effective WiMAX modems and residential gateways” that “offers a programmable architecture that makes it easier for equipment manufacturers to add innovative, unique applications on top of the standards-based Intel solution.” This programmability would only be possible if the 5116 featured a programmable processor with shared memory.

193. The Intel 5116 implemented the 2004 IEEE 802.16 standard and thus implemented multicarrier transmission, Reed-Solomon encoding, interleaving, and retransmission. Multicarrier transmission, Reed-Solomon coding, and retransmission were described as “Key Feature

Highlights” of the modem implemented on the 5116 in the excerpt below from the 2005 Intel Product Brief entitled “Intel PRO/Wireless 5116 Broadband Interface” where “256 OFDM PHY” indicated multicarrier transmission, Reed-Solomon coding is explicitly mentioned, and “ARQ capable” refers to retransmission.

Key Feature Highlights

Modem

- Highly integrated SoC based on IEEE 802.16-2004 standard
- 256 OFDM PHY with support for channel bandwidths up to 10 MHz
- TDD and H/FDD duplexing modes
- Concatenated Reed-Solomon and Convolutional Encoding Forward Error Correction
- Adaptive modulation (BPSK, QPSK, QAM16, QAM64)
- Enhanced link budget support
 - Receive space time coding
 - Uplink sub-channelization
 - SNR, RSSI channel quality measurement
 - ARQ capable

194. The processing on the Intel 5116 is performed using Dual-core ARM 946E-S engines and a DSP engine which are programmable processors that use shared memories. This is described in the “Processing” section excerpted below from the 2005 Intel Product Brief entitled “Intel PRO/Wireless 5116 Broadband Interface.”

Processing

- Dual-core ARM® 946E-S engines for PHY, MAC, and application protocol processing
- DSP engine with three parallel ALUs allow three simultaneous complex multiply operations per cycle for OFDM processing
- In-line security processing using advanced encryption techniques (3DES, AES, and RC4)

Key Feature Highlights

Modem

- Highly integrated SoC based on IEEE 802.16-2004 standard
- 256 OFDM PHY with support for channel bandwidths up to 10 MHz
- TDD and H/FDD duplexing modes
- Concatenated Reed-Solomon and Convolutional Encoding Forward Error Correction
- Adaptive modulation (BPSK, QPSK, QAM16, QAM64)
- Enhanced link budget support
 - Receive space time coding
 - Uplink sub-channelization
 - SNR, RSSI channel quality measurement
 - ARQ capable

C. ADSL2 Standard

195. The ITU-T published the G.992.3 standard in January 2005 on their publicly available website. ADSL2, cover page. ADSL2 is the second version of the asymmetric digital subscriber line standard. *Id.* DSL transceivers (such as those described in the ADSL2 standard) specifically function over two-wire twisted metallic copper pairs—that is, over telephone lines. *Id.* § 1 “Scope.”

196. The ADSL2 standard uses multicarrier transceivers to transmit data in the form of discrete multitone (DMT) symbols.

8.8 Modulation

The modulator shall modulate a constellation encoder output data frame or sync frame (containing $NSC - 1$ complex values $Z_i, i = 1$ to $NSC - 1$) into a DMT symbol. The data frame can be taken from the data symbol constellation encoder (68 per superframe) as defined in 8.6. The sync frame can be taken from the synchronization symbol constellation encoder (1 per superframe) as defined in 8.7. For (short) initialization and diagnostics mode signals, the frame is defined in 8.13, 8.14 and 8.15.

8.8.1 Subcarriers

A DMT symbol consists of a set of subcarriers, with index $i = 0$ to NSC . The DMT subcarriers spacing Δf , shall be 4.3125 kHz, with a tolerance of ± 50 ppm. The subcarrier frequencies shall be $f_i = i \times \Delta f$, $i = 0$ to NSC .

ADSL2 at 89, §§ 8.8-8.8.1 “Modulation” and “Subcarriers.”

197. The ADSL2 standard breaks down the multicarrier transceiver into multiple logical layers that perform certain functions. The transceiver at the customer premises is termed the Customer Premises Equipment “CPE” or ADSL Transceiver Unit at remote terminal “ATU-R” and the transceiver at the central office is termed the “CO” or ATU-C. ADSL2 at 6-9, §§ 4, 5.1 “Abbreviations,” “ATU functional model.” The layers used in ADSL2 are shown generally below, with some of the relevant portions (such as the Physical Media-Specific Transmission Convergence “PMS-TC” and Physical Media Dependent “PMD” layers) highlighted.

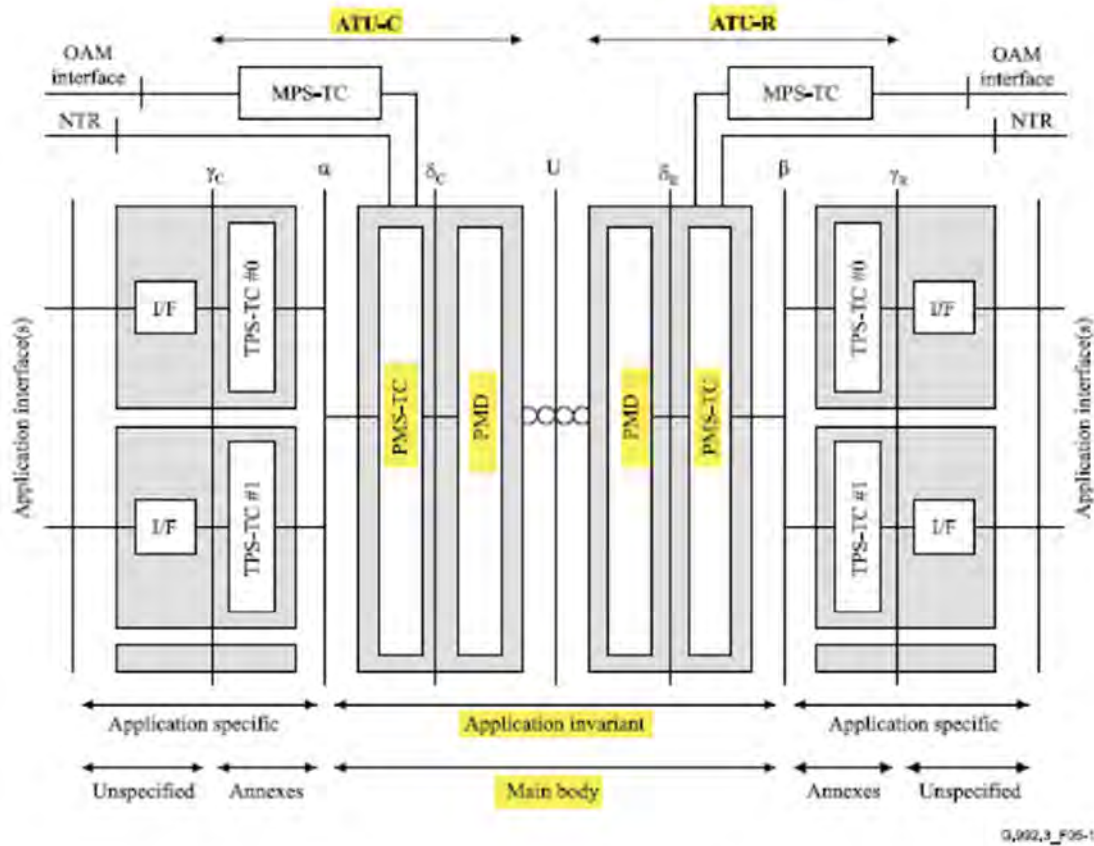


Figure 5-1/G.992.3 – ATU functional model

ADSL2 at 9, Figure 5-1.

198. The PMS-TC layer combines data from multiple frame bearers (such as different application streams, e.g. email and video streaming traffic) into latency paths, then performs operations on the data to protect it from errors, including: scrambling, CRC insertion, (Reed-Solomon) FEC, interleaving, and then combining the latency paths into a single stream provided to the PMD layer. The PMS-TC layer is shown below, with some relevant portions highlighted.

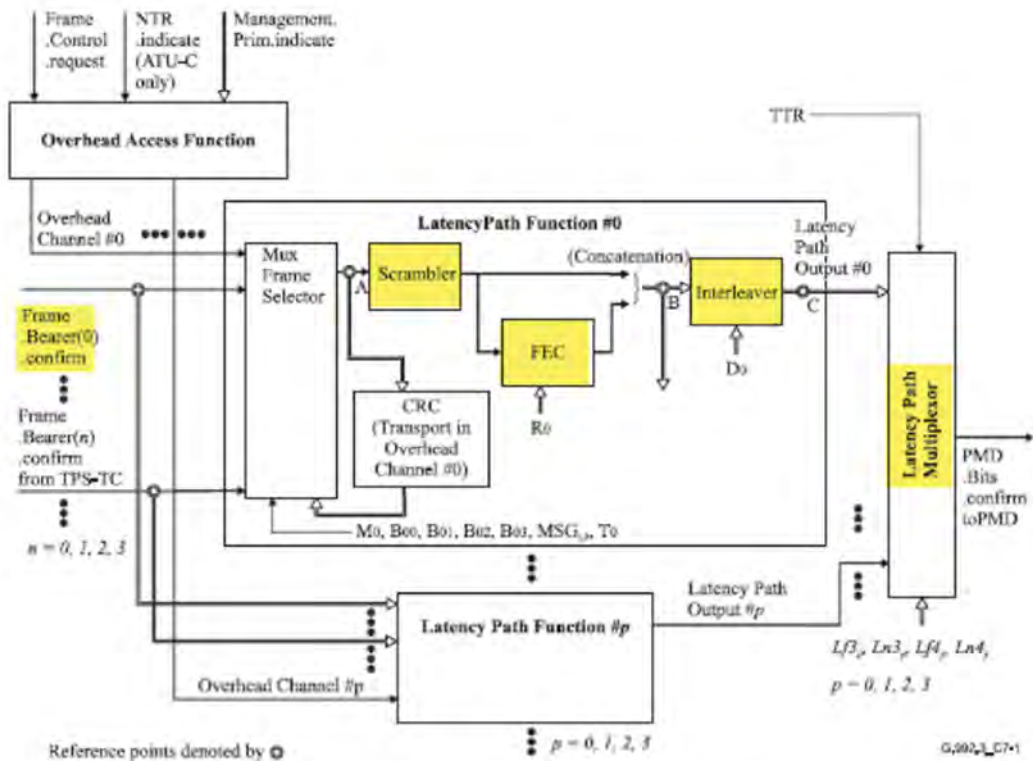


Figure C.7-1/G.992.3 – Block diagram of transmit PMS-TC function

199. ADSL2 at 27, Figure 7-6. The different latency paths allow for traffic to be treated differently—e.g., data that requires a strict limitation on tolerable delay (such as streaming video) will be given less interleaving protection because increased interleaving protection (i.e., increasing interleaver depth) introduces increased delay, but data that can tolerate more delay (such as an email) may be heavily interleaved, i.e., interleaved with a larger interleaver depth.

200. Two different examples of data progressing through the PMS-TC layer are provided in the ADSL2 standard. Figure 7-7 shows data from multiple bearers being formed into frames for a single latency path.

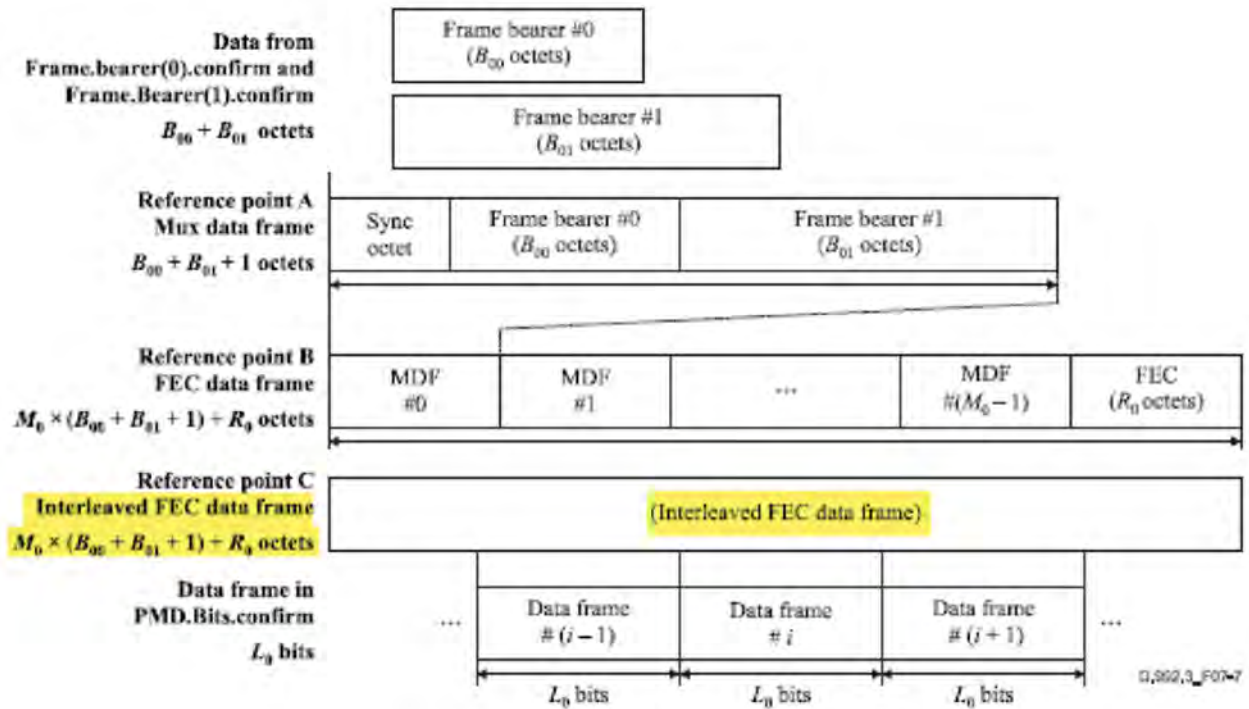


Figure 7-7/G.992.3 – Illustration of frame structure with single latency dual bearers and $T_p = 1$

ADLS2 at 29, Figure 7-7. MDF stands for Mux Data Frame.

201. Figure 7-8, on the other hand, shows data from two bearers being framed for transmission over two latency paths.

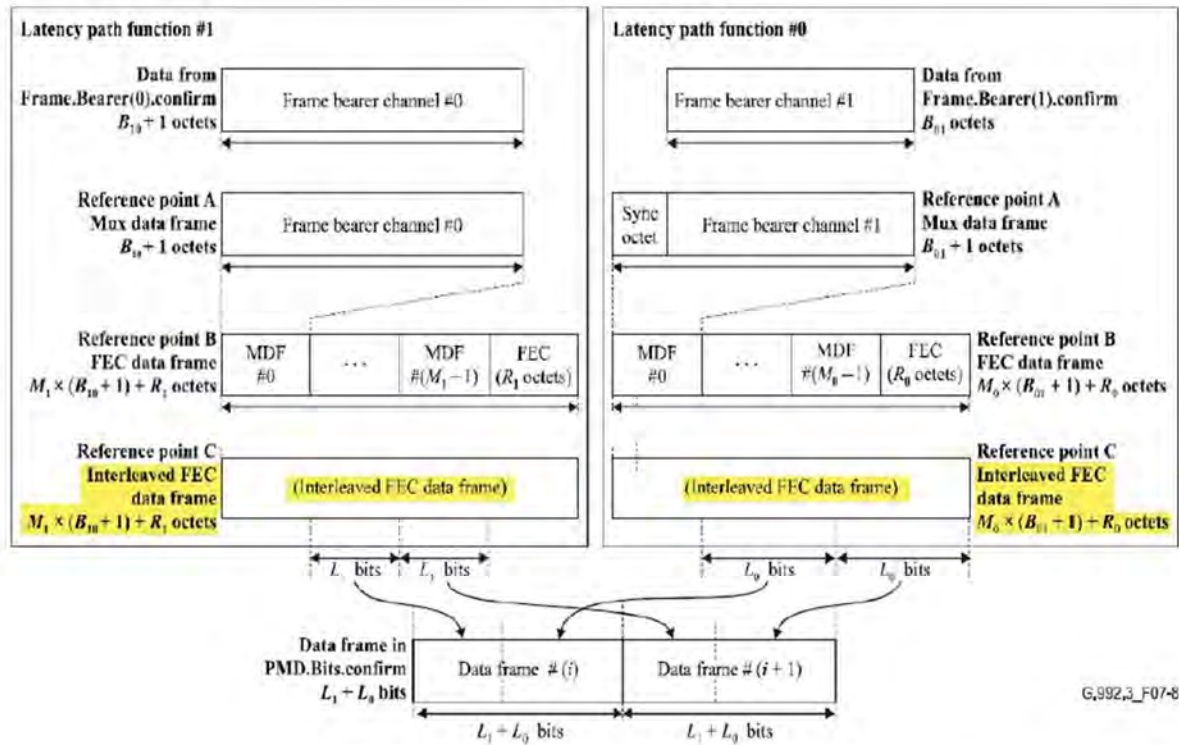


Figure 7-8/G.992.3 – Illustration of frame structure with dual latency and dual bearers

ADLS2 at 30, Figure 7-8.

202. The PMS-TC layer uses a Reed-Solomon Code. This is described in Section 7.7.1.4, which is excerpted below:

7.7.1.4 Forward error correction function

The FEC procedure inserts Reed-Solomon FEC redundancy octets to provide coding gain as an outer coding function to the PMD function. The FEC procedure of latency path function # p shall calculate R_p octets from $M_p \times K_p$ input octets. The octets are appended to the end of the input octets in the structure of FEC Output Data Frame at Reference Point B.

When $R_p = 0$, no redundancy octets are appended and the values in the FEC Output Data Frame are identical to the input values. For all other values of R_p , the following encoding procedure shall be used to create the R_p octets:

The FEC procedure shall take in M_p scrambled Mux Data Frames comprising message octets, $m_0, m_1, \dots, m_{M_p \times K_p - 2}, m_{M_p \times K_p - 1}$. The procedure shall produce R_p redundancy octets $c_0, c_1, \dots, c_{R_p - 2}, c_{R_p - 1}$. These two taken together comprise the FEC codeword of size $M_p \times K_p + R_p$ octets. The R_p redundancy octets shall be appended to the message octets to form the FEC Output Data Frame at Reference Point B.

ADSL2 at 37-38, § 7.7.1.4 “Forward error correction function.”

203. The PMS-TC layer uses an interleaver, specifically a convolutional interleaver. This is described in Section 7.7.1.5, which is excerpted below. Note that different interleaver depths D_p and different FEC Codeword sizes will require different minimum amounts of shared memory to implement.

7.7.1.5 Interleaver

To spread the Reed-Solomon codeword and, therefore, reduce the probability of failure of the FEC in the presence of impulse noise, the FEC Output Data Frames shall be convolutionally interleaved. The interleaver creates the Interleaved FEC Output Data Frames at Reference point C, at the output of the latency path function. This procedure is followed by the frame multiplexing procedure.

Convolutional interleaving is defined by the rule (using the currently defined values of the framing control parameters D_p and the derived parameter $N_{FEC,p}$):

Each of the $N_{FEC,p}$ octets $B_0, B_1, \dots, B_{N_{FEC,p}-1}$ in an FEC Output Data Frame is delayed by an amount that varies linearly with the octet index. More precisely, octet B_i (with index i) is delayed by $(D_p - 1) \times i$ octets, where D_p is the interleaver depth.

An example for $N_{FEC,p} = 5$, $D_p = 2$ is shown in Table 7-13, where B_i^j denotes the i -th octet of the j -th FEC Output Data Frame.

Table 7-13/G.992.3 – Convolutional interleaving example for $N_{FEC,p} = 5$, $D_p = 2$

Interleaver input	B_0^j	B_1^j	B_2^j	B_3^j	B_4^j	B_0^{j+1}	B_1^{j+1}	B_2^{j+1}	B_3^{j+1}	B_4^{j+1}
Interleaver output	B_0^j	B_3^{j-1}	B_1^j	B_4^{j-1}	B_2^j	B_0^{j+1}	B_3^j	B_1^{j+1}	B_4^j	B_2^{j+1}

With the above-defined rule, the output octets from the interleaver always occupy distinct time slots when $N_{FEC,p}$ is odd and D_p is a power of 2. When $N_{FEC,p}$ is even, a dummy octet shall be added at the beginning of the codeword at the input to the interleaver. The resultant odd-length codeword is then convolutionally interleaved, and the dummy octet shall then be removed from the output of the interleaver.

The interleaving procedure of the latency path function # p shall interleave a single FEC Output Data Frame, or $M_p \times K_p + R_p$ octets. This procedure is followed by the Frame Multiplexing procedure.

ADSL2 at 38, § 7.7.1.5 “Interleaver.”

204. The ADSL2 overhead channel (shown at the top left portion of Figure C.7-1) additionally uses retransmission to protect messages from errors.

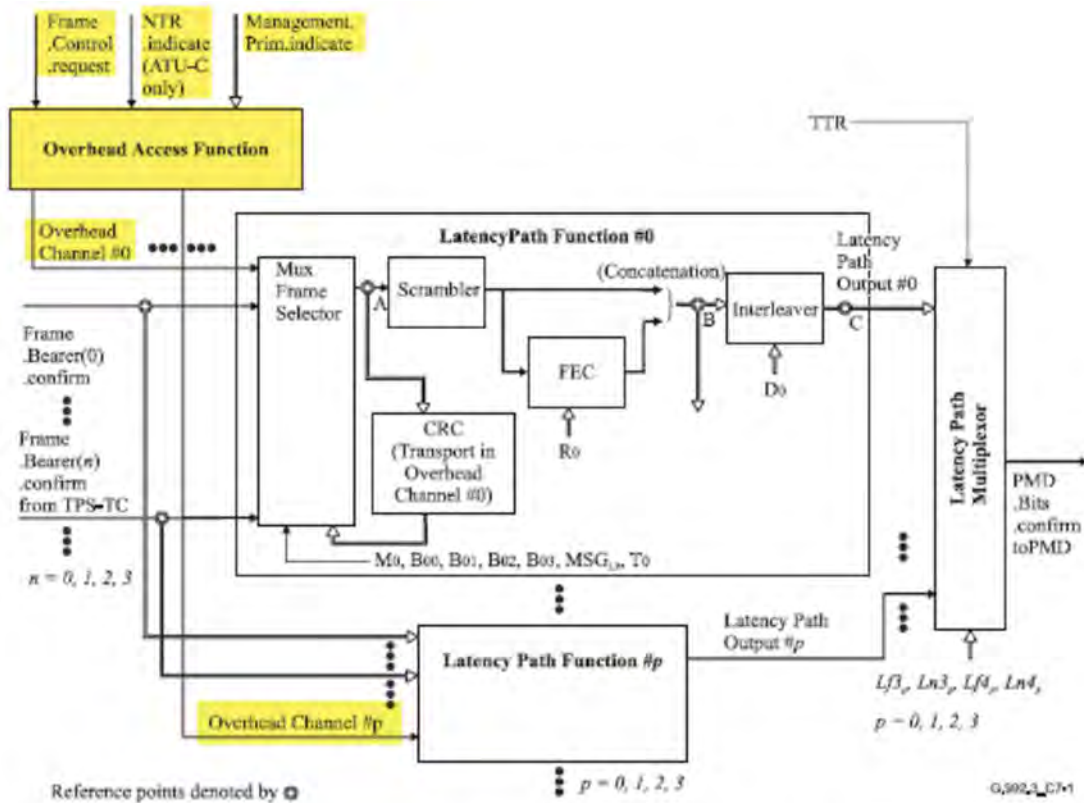


Figure C.7-1/G.992.3 – Block diagram of transmit PMS-TC function

ADSL2 at 27, Figure 7-6. Specifically, overhead messages are either individually acknowledged or broken into numbered segments with each segment except the last one in a message being acknowledged using the segment number. *See also:*

Each transmitted message segment shall be acknowledged by the far-end with a segment acknowledge message, except the last one. The last command message segment shall be acknowledged by the far-end with the appropriate response message (identical to the non-segmented case). The last response message segment shall not be acknowledged. Table 7-17a defines the segment acknowledge message. The following shall apply to all encapsulated segment acknowledge messages:

- HDLC frame shall contain 5 message octets ($P = 5$);
- The two least significant bits of the address field shall be identical to the corresponding bits in the address field of the acknowledged message segment. All other bits of the address field shall be set to 0₂;

- The second least significant bit of the control field shall either indicate a command message (request to continue the response, e.g., L2 grant) or a response message (request to continue the command, e.g., OLR);
- The least significant bit of the control field shall toggle for each acknowledgement message as it normally toggles for each command/response message (*see* 7.8.2.4.2);
- All other bits of the control field shall be set to 0₂.

Table 7-17a/G.992.3 – Segment acknowledge message

Message octet number	Message octet definition
Octet 1	Message designator 1111 0000 _b for acknowledgement of high priority message segment 1111 0001 _b for acknowledgement of normal priority message segment 1111 0010 _b for acknowledgement of low priority message segment
Octet 2	Message type of segment acknowledge message 01 ₁₆
Octet 3	Acknowledged message segment ID (in range 1 to 7)
Octet 4	Message Designator (first message octet of acknowledged message segment)
Octet 5	Message Type (second message octet of acknowledged message segment)

Every message segment shall be acknowledged by the far-end before the next message segment is transmitted. The segment acknowledge message shall not be segmented.

Timeouts shall be defined as follows:

- For a non-segmented message, timeout shall apply between the last octet of the command message transmitted and the first octet of the response message received.
- For a segmented response message, the timeout shall apply between the last octet of the command message transmitted and the first octet of the first message segment received.
- A timeout corresponding to the command priority shall apply between the last octet of the message segment transmitted and the first octet of the segment acknowledge message received.
- Another timeout corresponding to the command priority shall apply between the last octet of the segment acknowledge message transmitted and the first octet of the next message segment received.

If a timeout expires, the transmitter may repeat the last transmitted message. This repeated message can be an unsegmented command message, a command/response message segment (except the last segment of a response message), or a segment

acknowledge message. Alternately, the ATU may abandon the message after an implementation-specific number of retransmissions.

ADSL2 at pages 43-46, § 7.8.2.4.3 “Overhead message segmentation.”

205. ADSL2 requires that the CO (ATU-C) and CPE (ATU-R) exchange messages during the start-up phase, called “initialization.”

TPS-TC functions shall be fully configured prior to the initialization of the PMS-TC and PMD functions or be configured after initialization of the PMS-TC and PMS function in a manner that is outside the scope of this Recommendation. The configuration prior to initialization is performed via a G.994.1 MS message. Information may be exchanged prior to the mode select to ascertain capabilities using G.994.1 CL or CLR messages. Most of the information conveyed through G.994.1 messages is dependent on the TPS-TC type and is defined in Annex K.

ADSL2 at 19, § 6.6 “Initialization procedure.”

206. Initialization messages are protected from errors through retransmission, with the receiving transceiver sending an ACK (acknowledgment) or NACK (negative acknowledgment) to indicate if a message was correctly received. The receiver determines that an initialization message was correctly received by use of a CRC. Like the overhead messages, ADSL2 initialization messages can be broken into segments, each having a distinct segment number. If that is the case, then each segment is acknowledged with an ACK that includes the segment number of the segment being acknowledged.

At the start of the Exchange Phase, the ATU-C shall transition to the C-TREF1-LD state (in which C-TREF symbols shall be transmitted until the first R-MSGx-LD message is received).

If the ATU-C receives an R-MSGx-LD message, the ATU-C shall transition to the C-ACK or C-NACK state within 128 symbols from the end of the R-MSGx-LD state. If the R-MSGx-LD message is successfully received, the ATU-C shall transition to the C-ACK state (in which a positive acknowledgment C-ACK message shall be transmitted). Instead, if a decoding error occurs (i.e., the CRC locally computed at the ATU-C does not correspond to the CRC transmitted by the ATU-R), the ATU-C shall transition to the C-NACK state.

The C-ACK message shall be represented by the "01010101" octet and shall be transmitted over 64 symbol periods using the same modulation technique as the loop

diagnostics information bearing messages. No CRC shall be added to the C-ACK message. In the C-NACK state, the ATU-C shall transmit 64 C-TREF symbols. Note that from the ATU-R's perspective, this is equivalent to the ATU-C not responding to the R-MSGx-LD message.

At the end of the C-ACK or C-NACK state, the ATU-C shall transition to the C-TREF2-LD state (in which 256 C-TREF symbols shall be transmitted). During the C-TREF2-LD state, the ATU-R transitions to the R-QUIET2-LD state (because the R-ACK message is successfully received and no more R-MSGx-LD messages remain to be transmitted) or the ATU-R transitions to the R-SEGUE-LD state (because no, or a corrupted C-ACK message is received or more R-MSGx-LD messages remain to be transmitted). At the end of the C-TREF2-LD state, the ATU-C shall transition to the C-SEGUE-LD state (if the ATU-R has transitioned to the R-QUIET2-LD state) or shall return to the C-TREF1-LD state (if the ATU-R has returned to the R-SEGUE-LD state).

Note that, as a result of a corrupted C-ACK message, the ATU-C could successfully receive the same message twice. In this case, the ATU-C shall ignore the second identical (same Sequence Number) message.

ADSL2 at 158-159, § 8.15.5.1.2 “Message flow, acknowledgement and retransmission.”

207. The ADSL2 initialization messages include an exchange of some capabilities in the “G.994.1 capabilities message,” including maximal supported upstream (CPE to CO) and downstream (CO to CPE) data rates and data rates for each latency path in each direction. Additionally, the initialization messages include information about the maximum values for the number R_p of Reed-Solomon Code redundancy octets and the interleaver depth D_p for each latency path.

Table 7-18/G.992.3 – Format for PMS-TC capability list information

Npar(2) bit	Definition of Npar(2) bit
NTR	This bit is set to a one if the ATU has the capability to transport the NTR signal in the downstream direction.
Spar(2) bit	Definition of related Npar(3) octets
Downstream Overhead Data Rate	Parameter block of 2 octets that describes the minimum message based data rate that is needed by the ATU. The unsigned 6-bit value is the data rate divided by 1000 bits per second minus 1 (covering the range 1 to 64 kbit/s) – see Note.
Upstream Overhead Data Rate	Parameter block of 2 octets that describes the minimum message based data rate that is needed by the ATU. The unsigned 6-bit value is the data rate divided by 1000 bits per second minus 1 (covering the range 1 to 64 kbit/s) – see Note.
Downstream PMS-TC latency path #0 supported (always set to 1)	Parameter block of 2 octets that describes the maximum net_max downstream rate supported in the latency path #0. The unsigned 12-bit net_max value is the data rate divided by 4000. The net_max downstream rate shall be greater than or equal to the maximum required downstream data rate for each TPS-TC type that is supported by the ATU.
Upstream PMS-TC latency path #0 Supported (always set to 1)	Parameter block of 2 octets that describes the maximum net_max upstream rate supported in the latency path #0. The unsigned 12-bit net_max value is the data rate divided by 4000. The net_max upstream rate shall be greater than or equal to the maximum required upstream data rate for each TPS-TC type that is supported by the ATU.
Downstream PMS-TC latency path #1 Supported	Parameter block of 4 octets that describes the maximum net_max downstream rate, downstream $R_{1\text{max}}$, and downstream $D_{1\text{max}}$ supported in the latency path #1. The unsigned 12-bit net_max value is the data rate divided by 4000. $R_{1\text{max}}$ is an unsigned 4-bit value and shall be one of the valid R_p values divided by 2. $D_{1\text{max}}$ is an unsigned 3-bit value and shall be the logarithm base 2 of one of the valid D_p values.
Upstream PMS-TC latency path #1 Supported	Parameter block of 4 octets that describes the maximum net_max upstream rate, upstream $R_{1\text{max}}$, and upstream $D_{1\text{max}}$ supported in the latency path #1. The unsigned 12-bit net_max value is the data rate divided by 4000. $R_{1\text{max}}$ is an unsigned 4-bit value and shall be one of the valid R_p values divided by 2. $D_{1\text{max}}$ is an unsigned 3-bit value and shall be the logarithm base 2 of one of the valid D_p values.
Downstream PMS-TC latency path #2 Supported	Parameter block of 4 octets that describes the maximum net_max downstream rate, downstream $R_{2\text{max}}$, and downstream $D_{2\text{max}}$ supported in the latency path #2. The unsigned 12-bit net_max value is the data rate divided by 4000. $R_{2\text{max}}$ is an unsigned 4-bit value and shall be one of the valid R_p values divided by 2. $D_{2\text{max}}$ is an unsigned 3-bit value and shall be the logarithm base 2 of one of the valid D_p values.
Upstream PMS-TC latency path #2 Supported	Parameter block of 4 octets that describes the maximum net_max upstream rate, upstream $R_{2\text{max}}$, and upstream $D_{2\text{max}}$ supported in the latency path #2. The unsigned 12-bit net_max value is the data rate divided by 4000. $R_{2\text{max}}$ is an unsigned 4-bit value and shall be one of the valid R_p values divided by 2. $D_{2\text{max}}$ is an unsigned 3-bit value and shall be the logarithm base 2 of one of the valid D_p values.
Downstream PMS-TC latency path #3 Supported	Parameter block of 4 octets that describes the maximum net_max downstream rate, downstream $R_{3\text{max}}$, and downstream $D_{3\text{max}}$ supported in the latency path #3. The unsigned 12-bit net_max value is the data rate divided by 4000. $R_{3\text{max}}$ is an unsigned 4-bit value and shall be one of the valid R_p values divided by 2. $D_{3\text{max}}$ is an unsigned 3-bit value and shall be the logarithm base 2 of one of the valid D_p values.

Table 7-18/G.992.3 – Format for PMS-TC capability list information

Upstream PMS-TC latency path #3 Supported	Parameter block of 4 octets that describes the maximum net_max upstream rate, upstream $R_{3\text{max}}$, and upstream $D_{3\text{max}}$ supported in the latency path #3. The unsigned 12-bit net_max value is the data rate divided by 4000. $R_{3\text{max}}$ is an unsigned 4-bit value and shall be one of the value R_p values divided by 2. $D_{3\text{max}}$ is an unsigned 3-bit value and shall be the logarithm base 2 of one of the valid D_p values.
NOTE – By construction of the ADSL2 framing, the message-based overhead data rate is strictly smaller than 64 kbit/s. Hence, the G.994.1 phase of initialization should not request a minimum value of 64 kbit/s.	

ADSL2 at 49, Table 7-18.

208. During the Exchange phase, the ATU-C (CO) then transmits the C-PARAM message to the ATU-R (CPE) and the ATU-R (CPE) transmits the R-PARAM message to the ATU-C (CO).

The information in C-PARAM includes:

- The latency path MSGLP to carry the upstream message-oriented portion of the overhead channel.
- Assignment of upstream frame bearers to upstream latency paths.
- The number of message octets MSGc included in the upstream overhead structure.
- B_{pn} for each upstream latency path and frame bearer.
- M_p for each upstream latency path.
- R_p for each upstream latency path.
- D_p for each upstream latency path.
- TP for each upstream latency path.
- Lf3p, Ln3p, Lf4p, Ln4p corresponding to each upstream latency path.

The information in R-PARAM includes:

- The latency path MSGLP to carry the downstream message oriented portion of the overhead channel.
- Assignment of downstream frame bearers to downstream latency paths.
- The number of message octets MSGc included in the downstream overhead structure.
- B_{pn} for each downstream latency path and frame bearer.
- M_p for each downstream latency path.
- R_p for each downstream latency path.
- D_p for each downstream latency path.
- Tp for each downstream latency path.
- Lf3p, Ln3p, Lf4p, Ln4p corresponding to each downstream latency path.

This C-PARAMS and R-PARAMS information is represented as a parameter block as in Table C.7-2. The information is transmitted in the order shown during C-PARAM and R-PARAM as described in the PMD initialization procedure.

ADSL2 at 229-233, § C.7.10.3 “Exchange phase.” *See also id.* Table C.7-2 (excerpted).

Table C.7-2/G.992.3 – Format for PMS-TC PARAMS Information

Octet Number [i]	Format PMS-TC Bits $[8 \times i - 7 \text{ to } 8 \times i + 0]$	Description
Octet 0	[pfff 00bb] bit 7 to 0	The bits bb encode the value of MSG_{LP} . MSG_{LP} indicates the latency path in which the message-based overhead information is to be transmitted. The values 00, 01, 10, and 11 correspond to latency path #0, #1, #2, #3, respectively. The bits ffff encode the initialization success/failure code as defined in this clause. The bit p is the probing bit. A value 1 indicates that the current initialization is used for automode probing. A value 0 indicates that the current initialization is normal initialization.
Octet 1	[cccc dddd] bit 7 to 0	The bits cccc are set to 0000, 0001, 0010, or 0011 if the frame bearer #0 is to be carried in latency path #0, #1, #2, or #3 respectively. The bits dddd are set to 1111 if η_{p0} is zero (i.e., disabled frame bearer, see Table 6-1). The bits dddd describe where the frame bearer #1 is to be carried using the same encoding method as cccc.
Octet 2	[eeee ffff] bit 7 to 0	The bits eeee and ffff describe where the frame bearers #2 and #3, respectively, are to be carried using the same encoding method as cccc of octet 1.
Octet 3	[gggg gggg] bit 7 to 0	The bits gggggggg encode the value of MSG_C , the number of octets in the message-based portion of the overhead structure. The latency path $\#MSG_{LP}$ is used to transport the message based overhead information.
Octet 4	[hhhh hhhh] bit 7 to 0	The bits hhhhhhhh give the number of octets from bearer #0 per Mux Data Frame being transported. This value is zero or the non-zero value from the value of the set $\{B_{00}, B_{10}, B_{20}, B_{30}\}$.
Octet 5	[iiii iiii] bit 7 to 0	The bits iiiiii give the number of octets from bearer #1 per Mux Data Frame being transported. This value is zero or the non-zero value from the value of the set $\{B_{01}, B_{11}, B_{21}, B_{31}\}$.
Octet 6	[jjjj jjjj] bit 7 to 0	The bits jjjjjj give the number of octets from bearer #2 per Mux Data Frame being transported. This value is zero or the non-zero value from the value of the set $\{B_{02}, B_{12}, B_{22}, B_{32}\}$.
Octet 7	[kkkk kkkk] bit 7 to 0	The bits kkkkkkkk give the number of octets from bearer #3 per Mux Data Frame being transported. This value is zero or the non-zero value from the value of the set $\{B_{03}, B_{13}, B_{23}, B_{33}\}$.
Octet 8	[nnnnn mmmm] bit 7 to 0	The bits mmmmmmmm give the value of M_P for latency path #0. They are always present and set to zero if not used.
Octet 9	[ttt tttt] bit 7 to 0	The bits tttttt give the value of T_P for latency path #0. They are always present and set to zero if not used.
Octet 10	[rrrr 0DDD] bit 7 to 0	The bits rrrr0DDD give the value of R_P and D_P for latency path #0. The rrrr and DDD bits are coded as defined in Table 7-18. They are always present and set to zero if not used.
Octet 11	[llll llll] bit 7 to 0	The bits llllll give the LSB of the value of $Lf3_P$ for latency path #0. They are always present and set to zero if not used.
Octet 12	[llll llll] bit 15 to 8	The bits llllll give the MSB of the value of $Lf3_P$ for the latency path #0. These are always present and set to zero if not used.

209. The definitions of many of control parameters exchanged during the Exchange phase are provided Table 7-6 shown below:

Table 7-6/G.992.3 – Control parameters

Parameter	Definition
MSG_{min}	The minimum rate of the message-based overhead that shall be maintained by the ATU. MSG_{min} is expressed in bits per second.
MSG_{max}	The maximum rate of the message-based overhead that shall be allowed by the ATU. MSG_{max} is expressed in bits per second.
N_{BC}	See Table 6-1. This is a TPS-TC configuration parameter repeated here for clarity.
N_{LP}	The number of latency paths enabled to transport frame bearers and overhead. The latency path functions are labelled #0, #1, #2 and #3.
MSG_{LP}	The label of the latency path used to transport the message-based overhead information.
MSG_C	The number of octets in the message-based portion of the overhead structure.
$B_{p,n}$	The nominal number of octets from frame bearer # n per Mux Data Frame at Reference Point A in latency path function # p . When T_p is not set to 1 and n is the lowest index of the frame bearers assigned to latency path # p , the number of octets from the frame bearer # n in the latency path function # p varies between $B_{p,n}$ and $B_{p,n} + 1$.
M_p	The number of Mux Data Frames per FEC Data Frame in latency path function # p .
T_p	The ratio of the number of Mux Data Frames to the number of sync octets in the latency path function # p . A sync octet is inserted with every T_p -th Mux Data Frame. When T_p is not set to one, an extra frame bearer octet is carried whenever a sync octet is not inserted.
R_p	The number of RS redundancy octets per codeword in latency path function # p . This is also the number of redundancy octet per FEC Data Frame in the latency path function # p .
D_p	The interleaving depth in the latency path function # p .
L_p	The number of bits from the latency path function # p included per PMD.Bits.confirm primitive.

ADSL2 at 28, Table 7-6.

210. Related definitions are provided in Table 6-1, for which the values are set during the G.994.1 phase of initialization.

Table 6-1/G.992.3 – TPS-TC parameters

Parameter	Definition
N_{BC}	The number of enabled TPS-TC functions and the number of enabled frame bearers. The TPS-TC functions and frame bearers are labelled #0, #1, #2 and #3. N_{BC} is the number of non-zero values in the { $type_0$, $type_1$, $type_2$, $type_3$ } set. The value may be different for the ATU-C and the ATU-R.
$type_n$	The TPS-TC type mapped to frame bearer # n ($n = 0$ to 3). The TPS-TC type shall be set to a value described in Annex K. The $type_n$ value of zero shall be used to disable TPS-TC function # n and frame bearer # n .
$maxtype_n$	The maximum number of TPS-TC functions of type n supported.

ADSL2 at 18, Table 6-1.

211. The information exchanged in these messages including Bpn, the number of octets for each frame bearer and the number of redundancy octets Rp allows computation of the induced parameter of NFEC,p (as shown below in Table 7-7). Memory requirements (and the overall delay) associated with typical implementations of convolutional interleaving can be deduced from NFEC,p. and Dp. Since both sides must agree on all of the parameters related to convolutional interleaving for each latency path, it would have been obvious that these initialization messages could have included a computed value of memory requirements associated with a specific implementation of convolutional interleaving according to the parameters provided in the ADSL2 initialization messages. This would have had predictable results.

Table 7-7/G.992.3 – Derived characteristics of the ATU data frame

Symbols	Definition and value
K_p	Definition: The number of octets per Mux Data Frame in latency path function #p $K_p = \sum_{i=0}^{N_{BC}-1} B_{p,i} + 1$
$N_{FEC,p}$	Definition: The number of octets per FEC Data Frame and Interleaved FEC Data Frame in latency path function #p $N_{FEC,p} = M_p \times K_p + R_p$
S_p	Definition: The number of PMD.Bits.request primitives (and correspondingly the number of PMD symbols) over which the FEC Data Frame spans, not accounting for the $S_p = \frac{8 \times N_{FEC,p}}{L_p}$ The value of S_p may represent a non-integer value.

ADSL2 at 31, Table 7-7.

D. VDSL1 Standard

212. The ITU-T published the G.993.1 standard in June 2004 on their publicly available website. VDSL1 at NOK00078763, cover page. VDSL1 is the first version of the very high speed digital subscriber line standard. Id. DSL transceivers (such as those described in the VDSL1 standard) specifically function over the existing copper wires originally deployed to provide telephone service (sometimes called “POTS service”). Id. § 1 “Scope.”

213. Figure 5-1 shows the reference configuration used for the VDSL1 standard:

5.1 General reference models

Figure 5-1 shows the reference configuration used for G.993.1. It is essentially a Fibre to the Node architecture with an Optical Network Unit (ONU) sited in the existing metallic access network (or at the serving Local Exchange or Central Office). The first architectural model covers Fibre-to-the-cabinet (FTTCab) type of deployment; the second one is Fibre-to-the-exchange (FTTEx) type of deployment. Existing unscreened twisted metallic access wire-pairs are used to convey the signals to and from the customer's premises.

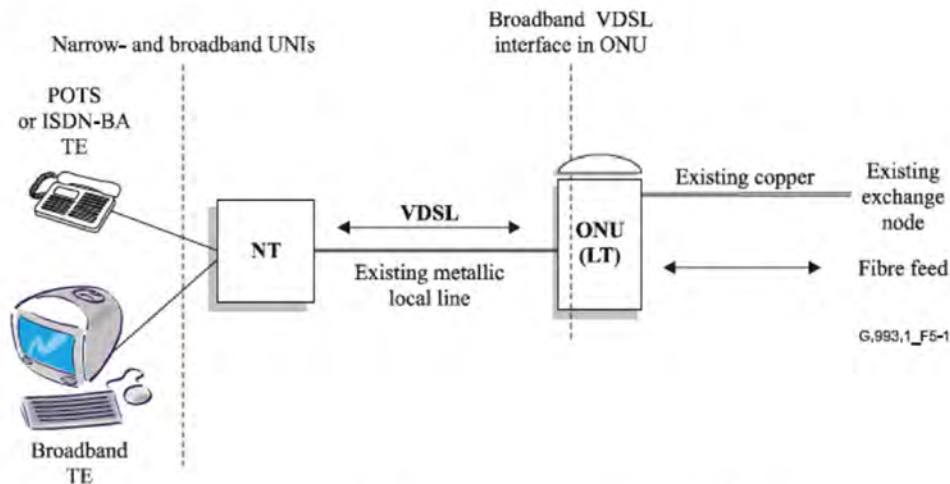


Figure 5-1/G.993.1 – General reference model

VDSL1 at NOK00078775, Section 5.1.

214. The VDSL1 standard uses multicarrier transceivers to transmit data in the form of discrete multitone (DMT) symbols.

9.2.1 Multi-carrier modulation

The modulation shall use a maximum number of subcarriers equal to $N_{SC} = 2^{n+8}$ where n can take the values 0, 1, 2, 3, 4. Disjoint subsets of the N_{SC} sub-carriers shall be defined for use in the downstream and upstream directions. These subsets are determined by the frequency plan (see 6.1). The exact subsets of sub-carriers used to modulate data in each direction shall be determined during initialization and shall be based on management system settings and the signal-to-noise ratios (SNRs) of the sub-channels. In many cases the number of sub-carriers used in a direction will be less than the maximum number allowed by the partitioning.

VDSL1 at NOK00078792, § 9.2.1 “Multi-carrier modulation.” *See also:*

G.993.1 transceivers shall use Frequency Division Duplexing (FDD) in separating upstream and downstream transmission.

G.993.1 systems use a four-band plan that starts at 138 kHz and extends up to 12 MHz. The four frequency bands denoted as DS1, US1, DS2, and US2, for the first downstream band, the first upstream band, the second downstream band, and the second upstream band, respectively, as shown in Figure 6-1, shall be allocated according to the band separating frequencies f_1 , f_2 , f_3 , f_4 , and f_5 .

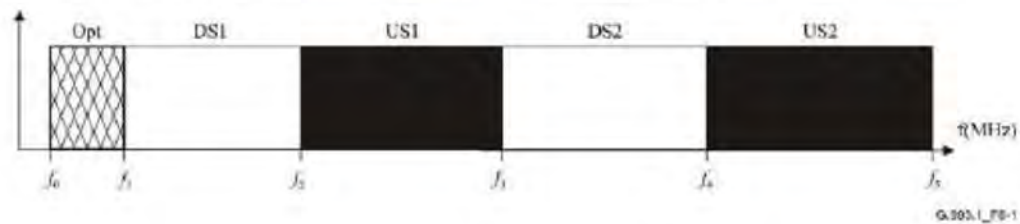


Figure 6-1/G.993.1 – G.993.1 band allocation

VDSL1 at NOK00078777, § 6.1 “Duplexing method.”

215. The VDSL1 standard breaks down the multicarrier transceiver into multiple logical layers that perform certain functions. The transceiver at the customer premise is termed the “CPE” or VDSL Transceiver at the Remote site “VTU-R” and the transceiver at the Optical Network Unit is termed the VTU-O. VDSL1 at NOK00078773-776, §§ 4, 5.2 “Abbreviations,” “Functional reference model.” The layers used in VDSL1 are shown generally below, with some of the relevant portions (such as the PMS-TC and PMD layers) highlighted.

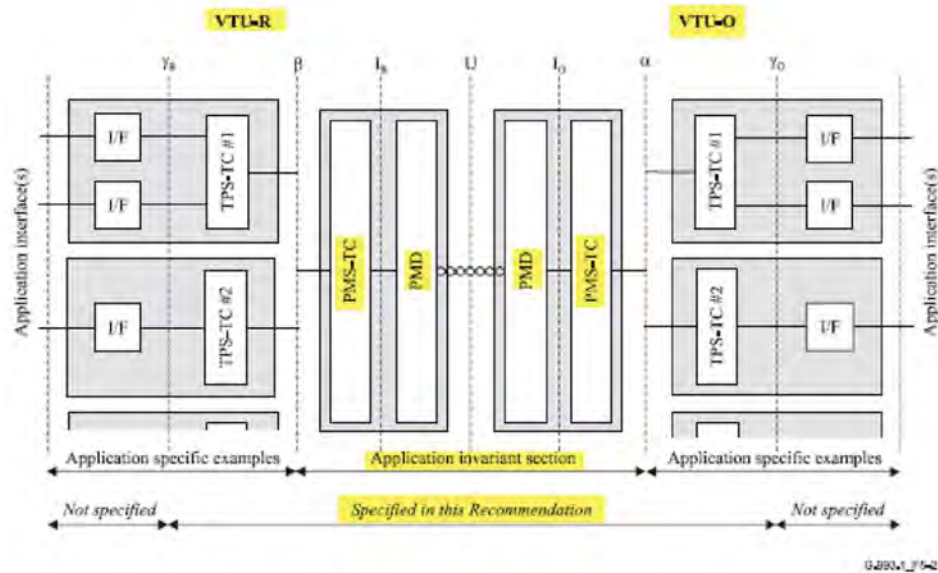


Figure 5-2/G.993.1 – VTU-x functional reference model

VDSL1 at NOK00078776, Figure 5-2.

216. The PMS-TC layer in the VDSL1 standard takes data from one of two payloads, the slow and fast payload, and (in the case of the mandatory slow payload) scrambles, applies FEC to, and interleaves that data. Fast payload data is not interleaved.

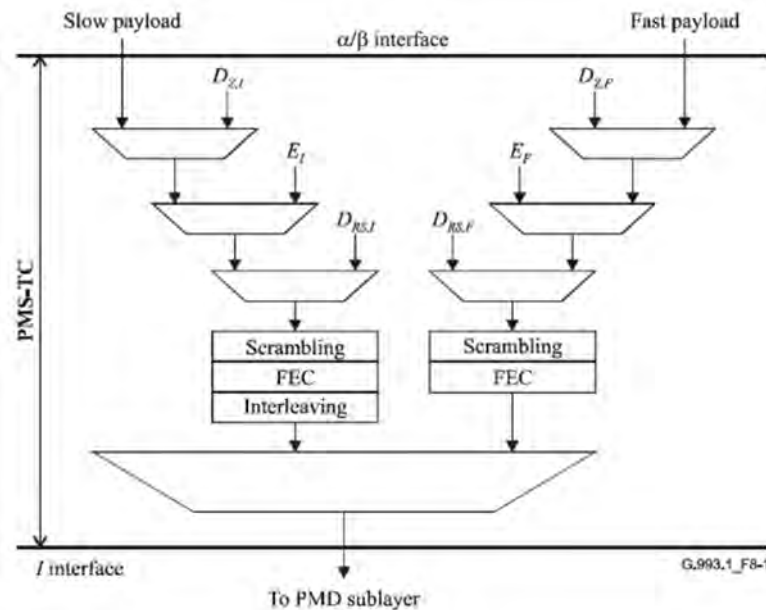


Figure 8-1/G.993.1 – Diagram of PMS-TC sublayer

VDSL1 at NOK00078784, Figure 8-1. See also:

8 PMS-TC sublayer

The PMS-TC sublayer provides transmission medium specific TC functions, such as framing, Forward Error Correction (FEC), and interleaving.

8.1 PMS-TC functional model

All data bytes shall be transmitted MSB first. All serial processing (e.g., scrambling, CRC calculation) shall however be performed LSB first, with the outside world MSB considered as the VDSL LSB. As a result, the first incoming bit (outside world MSB) shall be the first bit processed inside VDSL (VDSL LSB). The PMS-TC functional diagram is presented in Figure 8-1.

VDSL1 at NOK00078783, §§ 8-8.1 “PMS-TC sublayer” and “PMS-TC functional model.”

217. The PMS-TC sublayer uses a Reed-Solomon code as described in Section 8.3:

8.3 Forward error correction

A standard byte-oriented Reed-Solomon code shall be used to provide protection against random and burst errors.

A Reed-Solomon code word contains $N = K + R$ bytes, comprised of R redundant check bytes $c_0, c_1, \dots, c_{R-2}, c_{R-1}$ appended to K message bytes $m_0, m_1, \dots, m_{K-2}, m_{K-1}$. The check bytes shall be computed from the message bytes using the equation

$$C(D) = M(D)D^R \bmod G(D)$$

VDSL1 at NOK00078784-785, § 8.3.

218. The PMS-TC sublayer interleaving is described in detail in Section 8.4:

8.4 Interleaving

8.4.1 General

Interleaving shall be used to protect the data against bursts of errors by spreading the errors over a number of Reed-Solomon codewords. The interleave depth shall be programmable with a maximum interleave depth of 64 codewords when the number of octets per codeword (N) equals 255. For smaller values of N the interleave depth can grow nearly proportionately.

It shall be possible to adjust the interleave depth via the management system to meet latency requirements. The latency of the slow path is a function of the data rate and burst error correction capability. For data rates greater than or equal to 13 Mbit/s the latency between the α and β interfaces shall not exceed 10 ms when the interleaver depth is set to the maximum. At lower data rates there is a trade-off between higher latency and decreased burst error correction ability. At any data rate, the minimum latency occurs when the interleaver is turned off.

When the interleaver is on the codewords shall be interleaved before transmission to increase the immunity of RS codewords to bursts of errors. The convolutional interleaver is defined by two parameters: the interleaver block length, I , and the interleaving depth D . The block length I shall divide the RS codeword length N (i.e., N shall be an integer multiple of I). The convolutional interleaver uses a memory in which a block of I octets is written while an (interleaved) block of I octets is read. Details of the implementation are given in 8.4.2.

The same size interleaving memory (see Table 8-1) is needed for interleaving at the transmitter and de-interleaving at the receiver.

The convolutional interleaving introduces an absolute read-to-write delay, Δ_j , that increments linearly with the octet index within a block of I octet:

$$\Delta_j \equiv (D - 1) \times j$$

where $j = 0, 1, 2, \dots, I - 1$.

VDSL1 at NOK00078785, § 8.4.1 “Interleaving – General”.

219. Section 8.4.2 gives a detailed description of the typical “triangular” implementation of a convolutional interleaver. This implementation shows how the interleaver/deinterleaver memory size for this typical implementation can be deduced from the two parameters M and I as $M \times I \times (I - 1)/2$ bytes. In Figure 8-2 below, the interleaver memory is the left triangle of blocks of M -octet memories used as delay elements. The deinterleaver memory is the same size and is the inverted triangle on the right, again constructed of M -octet memories used as delay elements.

8.4.2 Triangular implementation

To decrease the implementation complexity, the delay increment ($D - 1$) shall be chosen as a multiple of the interleaver block length (I), i.e.: $D - 1 = M \times I$. The $(D - 1)$ to I ratio is the interleaving depth parameter (M). The characteristics of convolutional interleaving are shown in Table 8-1. The parameters t and q depend on the characteristics of the RS code and are defined as:

- t = number of bytes that can be corrected by RS codewords = half the number of redundancy bytes = $R/2$;
- q = length of RS codeword divided by the length of an interleaver block = N/I .

VDSL1 at TQD-ADT 066670, § 8.4.2.

Table 8-1/G.993.1 – Characteristics of convolutional interleaving

Parameter	Value
Interleaver block length (I)	I bytes (equal to or divisor of N)
Interleaving Depth (D)	$M \times I + 1$
(De)interleaver memory size	$M \times I \times (I - 1)/2$ bytes
Correction capability	$\lfloor t/q \rfloor \times (M \times I + 1)$ bytes
End-to-end delay	$M \times I \times (I - 1)$ bytes

The example in Figure 8-2 shows $I = 7$. I parallel branches (numbered $0 \dots I - 1$) are implemented with a delay increment of M octets per branch. Each branch shall be a FIFO shift register (delay line) with length $0 \times M \dots (I - 1) \times M$ bytes. The deinterleaver is similar to the interleaver, but the branch indices are reversed so that the largest interleaver delay corresponds to the smallest deinterleaver delay. Deinterleaver synchronization shall be achieved by routing the first byte of an interleaved block of I bytes into branch 0.

VDSL1 at NOK00078786, § 8.4.2.

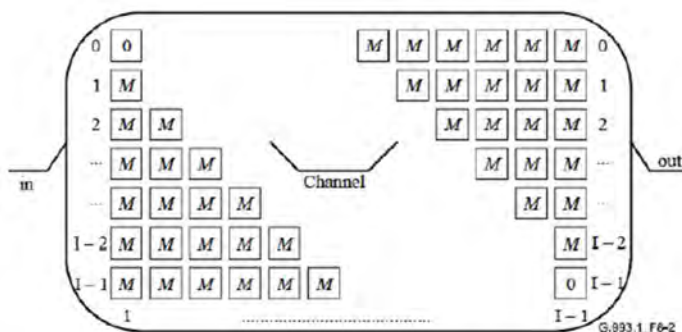


Figure 8-2/G.993.1 – Implementation example with $D - 1 = M \times I$ and $I = 7$

VDSL1 at NOK00078786, § 8.4.2.

220. Table 8-2 below shows the interleaver/deinterleaver memory sizes that correspond to various values of M and I using the equation $M \times I \times (I - 1)/2$ bytes to compute the interleaver/deinterleaver memory sizes.

Table 8-2/G.993.1 – Example of interleaver parameters with RS(144,128)

Rate [kbit/s]	Interleaver parameters	Interleaver depth	(De)interleaver memory size	Erasure correction	End-to-end delay
50×1024	$I = 72$ $M = 13$	937 blocks of 72 bytes	33 228 bytes	3 748 bytes 520 μ s	9.23 ms
24×1024	$I = 36$ $M = 24$	865 blocks of 36 bytes	15 120 bytes	1 730 bytes 500 μ s	8.75 ms
12×1024	$I = 36$ $M = 12$	433 blocks of 36 bytes	7 560 bytes	866 bytes 501 μ s	8.75 ms
6×1024	$I = 18$ $M = 24$	433 blocks of 18 bytes	3 672 bytes	433 bytes 501 μ s	8.5 ms
4×1024	$I = 18$ $M = 16$	289 blocks of 18 bytes	2 448 bytes	289 bytes 501 μ s	8.5 ms
2×1024	$I = 18$ $M = 8$	145 blocks of 18 bytes	1 224 bytes	145 bytes 503 μ s	8.5 ms

VDSL1 at NOK00078786, § 8.4.2.

221. The standard requires support of numerous values of / and + as described at the end of Section 8.4.2 (shown below). Thus the VTU-R and VTU-O must agree as to the particular values of M and I that will be used. This is accomplished through configuration messages including O-MSG2 and R-MSG2 described below. Because the memory allocation of an obvious convolutional interleaver implementation can be deduced from the parameters that describe which option for convolutional interleaving has been adopted, it would have been obvious that this allocation value could be included in the configuration message. This would have been an obvious alternative with predictable results. Indeed, a maximal interleaver memory was computed using such an equation and transmitted in R-MSG2 as shown in Table 12-32 below.

222. The following interleaver parameters shall be supported:

The following interleaver parameters shall be supported:

- For $(N,K) = (144,128)$ the following values for M and I shall be supported:
 $I = 36$ and M between 2 and 52.
- For $(N,K) = (240,224)$ the following values for M and I shall be supported:
 $I = 30$ and M between 2 and 62.

VDSL1 at NOK00078787, § 8.4.2.

223. VDSL1 uses an initialization procedure to set up the link between the VTU-O and VTU-R, during which the transceivers exchange parameters. In this procedure, messages are sent over the Special Operation Channel (SOC).

The VTU-R shall send the message R-MSG2 over the SOC channel to transfer information about its bit allocation capabilities and several other features. After receiving this message, the VTU-O shall do the same by sending the SOC message O-MSG2, containing the capabilities of the VTU-O.

VDSL1 at NOK00078847-848, § 12.4.6.1 “Sequence of messages and symbols during channel analysis and exchange state.”

224. The initialization procedure includes the VTU-O sending a message called the O-MSG2 to the VTU-R. The O-MSG2 includes information about the VTU-O’s capabilities such as interleaving/deinterleaving depths supported by the VTU-O and the maximum interleaver delay the VTU-O can use.

Table 12-23/G.993.1 – Description of O-MSG2

Field content	Field or macro-field type	Remark
Message descriptor	1 byte	See Table 12-12.
Minimal SNR margin	1 byte	In units of 0.5 dB
Maximal constellation size in downstream ($B_{max,d}$)	1 byte	Maximum number of bits per tone
RS setting supported by VTU-O	1 byte	0x00: Only mandatory settings 0xFF: All settings
Interleaver settings supported by the VTU-O	1 byte	0x00: Only mandatory settings 0xFF: All settings 0xNN NN = Number of additional (i.e., non-mandatory) settings in hexadecimal (NN $\neq$ 0x00 and NN $\neq$ 0xFF)
Detailed interleaver setting description	0 byte if NN = 0x00 or NN = 0xFF, 4xNN otherwise	Interleaver descriptor (see Table 12-24)
Maximal power in downstream	1 byte (unsigned)	In units of 0.25 dBm
Maximum interleaver delay	1 byte	In ms by steps of 0.5 ms (Note 1)
Maximum number of EOC bytes per frame in downstream	1 byte	Number of EOC bytes per frame
Maximum number of VOC bytes per frame in downstream	1 byte	Number of VOC bytes per frame

Table 12-23/G.993.1 – Description of O-MSG2

Field content	Field or macro-field type	Remark
Support of express bit swapping	1 byte	0x00: Not supported 0xFF: Supported
j_{max}	1 byte	Maximum value of j_{max} supported by the VTU-O (Note 2)
NOTE 1 – This field can be set to 0 in order to emulate the fast channel. This field is used for the creation of R-CONTRACT1 even if dual latency is used at the end.		
NOTE 2 – Specification of $j_{max} = k$ means that all values $0, 1, \dots, k$ are supported.		

j_{max} is defined in 12.4.6.2.1.3.

VDSL1 at NOK00078849-850, Table 12-23. See also id. Table 12-24:

Table 12-24/G.993.1 – Interleaver descriptor

Field	Field or macro-field type
I	1 byte
q	1 byte
$M_{\min}$	1 byte
$M_{\max}$	1 byte
NOTE – The four fields are repeated for each interleaver setting.	

225. The VTU-R responds to the O-MSG2 by sending the VTU-O a message called the R-MSG2.

Table 12-32/G.993.1 – Description of R-MSG2

Field	Field or macro-field type	Remark
Message descriptor	1 byte	See Table 12-12.
Maximal constellation size in upstream ($B_{\max u}$)	1 byte	Maximum number of bits per tone
RS setting supported by VTU-R	1 byte	0: Only mandatory settings 1: All settings
Interleaver settings supported by the VTU-R	1 byte	0x00: Only mandatory settings 0xFF: All settings 0xNN NN = Number of additional (i.e., non-mandatory) settings in hexadecimal (NN $\neq$ 0x00 and NN $\neq$ 0xFF)
Detailed interleaver setting description	0 byte if NN = 0x00 or NN = 0xFF; 4xNN otherwise	Interleaver descriptor (See Table 12-24.)
Maximal power transmitted	1 byte (unsigned)	In dBm by steps of 0.25 dBm
Maximal interleaver memory	3 bytes	In bytes (Note 1)
Maximum number of EOC bytes per frame in upstream	1 byte	Number of EOC bytes per frame
Maximum number of VOC bytes per frame in upstream	1 byte	Number of VOC bytes per frame
Support of express bit swapping	1 byte	0x00: Not supported 0xFF: Supported
$j_{\max}$	1 byte	Specify maximum value of $j_{\max}$ supported by the VTU-R (Note 2)
NOTE 1 – The interleaver memory is computed as $M \times I \times (I - 1)$.		
NOTE 2 – Specification of $j_{\max} = k$ means that all values from 0, 1, ..., k are supported.		

VDSL1 at NOKIA00078853, Table 12-32. See also *id.* NOK00078850, Table 12-24 *supra*.

226. VDSL1's initialization messages (including the O-MSG2 and R-MSG2) may be segmented, with each segment being given a unique identifier. The receiving transceiver will acknowledge each segment received, including an identification of the segments being acknowledged, or may transmit a repeat request for particular segments not correctly received.

12.4.2.6 Segmentation of messages

Some messages could potentially be large, and even larger than the maximal allowed frame size of an HDLC frame (1026 octets). It shall therefore be possible to segment messages before transmission. In order to do this, all messages transmitted during initialization shall receive a number that counts the message. This number shall be stored in one byte and will wrap around in case of overflow. The value 0 shall not be used because it has a special meaning (see later). This means that 255 shall be followed by 1.

This "message index" shall be transmitted in the Address Field of the HDLC frame (see Figure 12-3). The message index allows to track lost messages and to request the retransmission of a particular message. The index shall initially be set to one and shall be incremented with one after the transmission of a message. The index shall not be incremented in case of a REPEAT-REQUEST. The counting of messages shall start when the transmission starts using RQ mode instead of automatic repeat (AR) mode.

A segmentation index (1 byte) shall be included in the Control Field of the HDLC frame. The four MSBs of this field shall indicate the number of segments that make up the total message. The four LSBs shall indicate the index of the current segment. For instance, a value 0x93 indicates the third segment of a total of nine. In case the message is not segmented, the value of the field shall be 0x11.

The REPEAT-REQUEST message will behave differently from the other messages. The message index counter shall not be increased when a REPEAT-REQUEST message is sent. Also, the meaning of the message index and the segmentation index is different in this case.

The message index of the REPEAT-REQUEST message shall contain the message index of the message that should be retransmitted. The default value 0 indicates that the last unacknowledged message should be sent (this is why 0 shall never be used as message index for any other message). If this message contains several segments, only the last segment shall be retransmitted.

Likewise, the control field shall contain the segment that should be retransmitted. The Information payload of the REPEAT-REQUEST shall consist of one byte containing the value 0x55.

In the initialization procedure a transmitter should never send two consecutive messages without acknowledgement of the first message. It will always first receive some message from the other side before transmitting again. Therefore, an acknowledgement shall be sent for all but the last segment. Typically, the last segment signals the end of the message and it will therefore be acknowledged by the reply to this message. The ACK-SEG-message (see Table 12-12) shall be used to acknowledge the reception of the other segments. The ACK-SEG-message shall have its own message and segment index, which shall not refer to the segmented message that is being sent.

Once acknowledged, messages (or segments) are not expected to be retransmitted again. Also, a REPEAT-REQUEST that contains a number that is higher than the most recent value of the counter shall be ignored.

In AR mode, segmentation shall be done in the same way but there will be no acknowledgement (ACK-SEG) between different segments of the same message. Segments shall be sent in order.

VDSL1 at NOK00078835-836, § 12.4.2.6 “Segmentation of messages.”

E. VDSL2 Standard

227. The ITU-T released the G.993.2 standard (“VDSL2”) in February 2006 on their publicly available website. VDSL2 at TQD_TX00153386, cover page. VDSL2 is the second version of the very high speed digital subscriber line standard. *Id.* It is described as an enhancement to the VDSL1 standard. VDSL2 at TQD_TX00153394-395, § 1 “Scope.”

228. DSL transceivers (such as those described in the VDSL2 standard) specifically function over the existing copper wires originally deployed to provide telephone service (sometimes called “POTS service”). *Id.* § 1 “Scope.”

229. The VDSL2 standard uses multicarrier transceivers to transmit data in the form of discrete multitone (DMT) symbols.

10.4.1 Data sub-carriers

The sub-carriers shall be indexed from $i = 0$ to $i = \text{MSI}$, where MSI is the index of the highest loaded sub-carrier (i.e., the maximum index in the MEDLEY set). The values of MSI may be different for upstream and downstream transmission and are denoted as MSI_{us} and MSI_{ds} respectively. The index of the highest loaded sub-carrier (MSI_{us} or MSI_{ds}) will be restricted by the selected profile and band plan as shown in Table 6-1. Specifically, MSI_{us} shall be equal to or lower than the "index of the highest supported upstream data-bearing sub-carrier" (6.2.10) and MSI_{ds} shall be equal to or lower than the "index of the highest supported downstream data-bearing sub-carrier" (6.2.9). Transmission will take place on NSC sub-carriers, with $\text{NSC}_{\text{us}} \leq \text{MSI}_{\text{us}}$ and $\text{NSC}_{\text{ds}} \leq \text{MSI}_{\text{ds}}$; the sub-carrier with index $i=0$ shall not be used. $\text{NSC}_{\text{us}} + \text{NSC}_{\text{ds}}$ shall always be less than 4096.

The sub-carriers to be used for data transport in the upstream and downstream directions ($\text{MEDLEY}_{\text{us}}$ and $\text{MEDLEY}_{\text{ds}}$ sets, respectively) shall be determined during initialization, as specified in 12.3.3.

NOTE – The sub-carriers used for data transmission depend on channel characteristics, such as loop attenuation and noise, and on the specific requirements on the PSD of the transmit signal, such as notching of amateur radio bands, PSD reduction at low frequencies to share the loop with POTS or ISDN, and others.

10.4.2 Sub-carrier spacing

Sub-carrier spacing is the frequency spacing, Δf , between the sub-carriers. The sub-carriers shall be centered at frequencies $f = i \times \Delta f$. The sub-carrier index i takes the values $i = 0, 1, 2, \dots, \text{MSI}$. Valid values of sub-carrier spacing are 4.3125 kHz and 8.625 kHz, both with a tolerance of ± 50 ppm.

Sub-carrier spacing is profile dependent (see Table 6-1).

10.4.3 Modulation by the inverse discrete Fourier transform (IDFT)

The IDFT is used to modulate the output of the symbol encoder onto the DMT sub-carriers. It converts the NSC complex values Z_i (as defined in 10.3.4) generated by the symbol encoder (frequency domain representation) into $2N$ real values x_n ($n = 0, 1, \dots, 2N - 1$), which is a time domain representation. The conversion shall be performed with a $2N$ point IDFT, with $N - 1 \geq \text{MSI}$, as:

$$x_n = \sum_{i=0}^{2N-1} \exp\left(j \cdot 2 \cdot \pi \cdot \frac{n \cdot i}{2 \cdot N}\right) \cdot Z_i \text{ for } n = 0 \text{ to } 2N - 1$$

The valid values of N are $N = 2^{n+5}$, where n can take integer values from 0 to 7. The values of N used for upstream and downstream are exchanged during initialization (see 12.3.2, 12.3.3.2.1.3, 12.3.3.2.2.3).

For sub-carrier indices i that are not in the MEDLEY set and for $MSI < i < N$, the corresponding values of Z_i are not generated by the symbol encoder. These values are vendor discretionary, but shall comply with the constraints given in Table 10-4. Z_0 shall always be equal to zero and Z_N shall always be a real value.

In order to generate real values of x_n , the input values Z_i , where $i = 0, 1, \dots, N - 1$ and $Z_0 = 0$, shall be further augmented so that the vector Z_i has a Hermitian symmetry:

$$Z_i = \text{conj}(Z_{2N-i}) \text{ for } i = N + 1 \text{ to } 2N - 1$$

NOTE – Different values of N result in different transmit signal images above the Nyquist frequency. Knowledge of how the additional Z_i values are defined allows the receiver to better estimate the channel during initialization.

VDSL2 at TQD_TX00153471-472, §§ 10.4.1-10.4.3 “Data sub-carriers,” “Sub-carrier spacing,” and “Modulation by the inverse discrete Fourier transform (IDFT).”

230. The VDSL2 standard breaks down the multicarrier transceiver into multiple logical layers that perform certain functions. The transceiver at the customer premise is termed the VTU-R and the transceiver at the central office is termed the VTU-O. VDSL21 at TQD_TX00153401-405, §§ 4, 5.1 “Abbreviations,” “VTU functional.” The layers used in VDSL2 are shown generally below, with some of the relevant portions (such as the PMS-TC and PMD layers) highlighted.

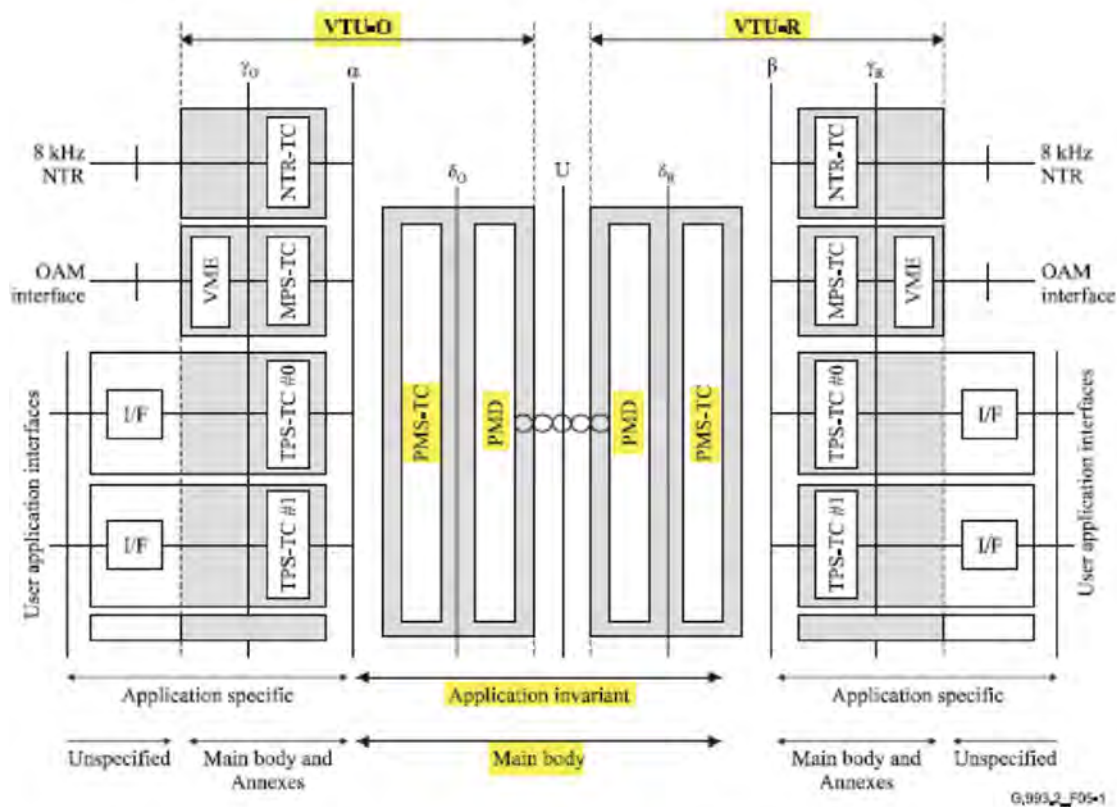


Figure 5-1/G.993.2 – VDSL2 and VTU functional model

VDSL2 at TQD_TX00153404, Figure 5-1.

231. The PMS-TC layer in the VDSL2 standard takes data from various bearers and forms that data into at least one latency path (the second latency path being optional) and combines that data with overhead data. The data in both latency paths is then scrambled, FEC is applied, and interleaved, before the two latency paths are combined into one stream to send to the next layer (the PMD layer) where it is prepared for transmission.

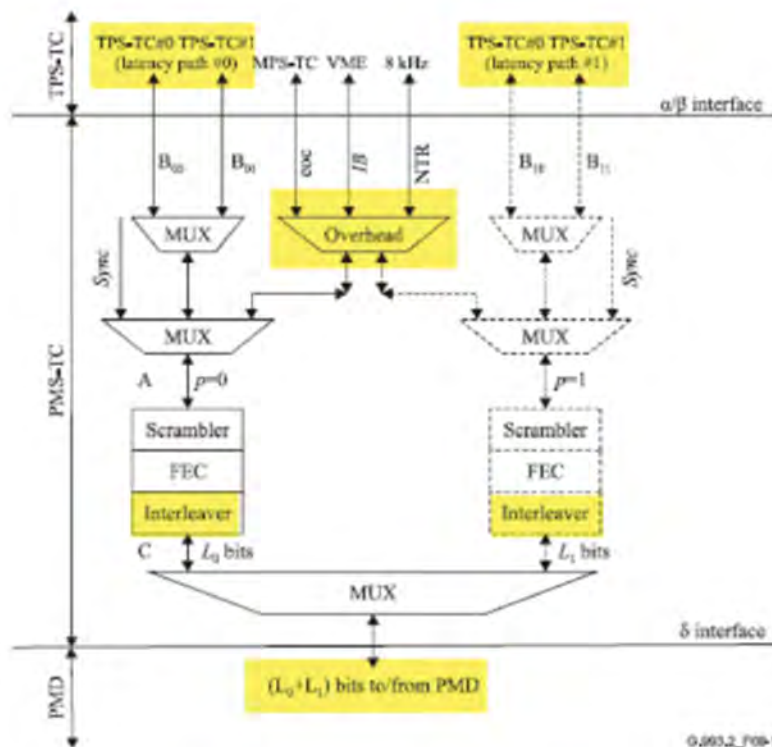


Figure 9-1/G.993.2 – PMS-TC functional model

VDSL2 at TQD_TX00153441, Figure 9-1. *See also:*

9 Physical media specific transmission convergence (PMS-TC) sub-layer

The PMS-TC sub-layer resides between the a/I3 interface and 8 interface of the VDSL2 transceiver reference model (see Figure 5-1). It provides transmission medium specific TC functions, such as scrambling, framing, forward error correction (FEC), and interleaving.

9.1 PMS-TC functional model

The PMS-TC functional model is presented in Figure 9-1. Up to two bearer channels of transmit user data originated by various TPS-TCs, management data originated by the MPS-TC, and NTR data are incoming via the a/I3 interface in a uniform format, as specified in 8.1.2. The incoming user data and the overhead data are multiplexed into one or two latency paths. Each bearer channel is carried over a single latency path (i.e., shall not be split across two latency paths). A Syncbyte is added to each latency path for OH frame alignment.

The VTU shall support at least one latency path; support of two latency paths is optional. If only one latency path is enabled, it shall be latency path #0.

NOTE 1 – When transporting two or more applications with different latency and impulse noise protection (INP) requirements and limited higher layer error

resilience, a VTU should implement dual latency because, in general, under these conditions dual latency will provide improved performance and/or quality of service.

The multiplexed data in each latency path is scrambled, encoded using Reed-Solomon forward error correction coding, and interleaved. The interleaved buffers of data of both latency paths are multiplexed into a bit stream to be submitted to the PMD sub-layer via the 8 interface.

VDSL2 at TQD_TX00153440-402, §§ 9-9.1 “Physical media specific transmission convergence (PMS-TC) sub-layer” and “PMS-TC functional model.”

232. The PMS-TC sublayer interleaving is described in detail.

9.4 Interleaving

Interleaving shall be provided in all supported latency paths to protect the data against bursts of errors by spreading the errors over a number of Reed-Solomon codewords. The convolutional interleaver adopted for VDSL2 shall follow the rule:

- I is the interleaver block size in bytes. Each of the I bytes in an interleaver block $B_0B_1 \dots B_{I-1}$ shall be delayed by the interleaver by an amount that varies linearly with the byte index. More precisely byte B_j (with index j) shall be delayed by $\Delta[j] = (D - 1) \times j$ bytes, where D is the interleaver depth in bytes, and D and I are co-prime (have no common divisor except for 1).

For any interleaver input of size $D \times I$ bytes, the relationship between the index of each input byte (n_{in}) and the index of each output byte (n_{out}) is given by $n_{out} = (n_{in} + \Delta[j])$, where $j = n_{in} \bmod I$ and $\Delta[j] = (D - 1) \times j$.

The total delay of the interleaver/de-interleaver combination is $(D - 1) \times (I - 1)$ bytes.

The RS codeword length N_{FEC} shall be an integer multiple of I , i.e., $N_{FEC} = q \times I$, where q is an integer between 1 and 8 inclusive. All values of q shall be supported. Codewords shall be mapped to interleaver blocks such that the first I bytes of the codeword map to the I bytes $B_0B_1 \dots B_{I-1}$ of the first interleaver block.

The interleaver depth shall be set to meet the requirements for error-burst protection and latency. The VTU shall support all integer values of D from 1 to D_{max} , as specified for the particular profile (see Table 6-1). At any data rate, the minimum latency occurs when the interleaver is turned off. If both latency paths are

supported, interleaving shall be supported on both latency paths. The same valid and mandatory configuration parameters shall apply to all supported latency paths.

A summary of interleaver parameters is given in Table 9-3.

Table 9-3/G.993.2 – Summary of interleaver parameters

Parameter(s)	Value
D and I	Co-prime
q	Integer between 1 and 8, inclusive
N_{FEC}	Integer between 32 and 255 inclusive, $N_{FEC} = q \times I$
Total delay of the interleaver/de-interleaver combination	$(D-1) \times (I-1)$ bytes

VDSL2 at TQD_TX00153444, § 9.4 “Interleaving”.

233. VDSL2 uses an initialization procedure to set up the link between the VTU-O and VTU-R, during which the transceivers exchange parameters.

12.1.3 Initialization procedures

During the G.994.1 handshake phase of the initialization procedure, the VTUs exchange capability lists and agree on a common mode for training and operation using the G.994.1 protocol. A successful completion of the G.994.1 handshake phase will lead to either the channel discovery phase of initialization or the loop diagnostic mode (depending on which one is selected). Failure of the G.994.1 handshake phase leads back to the L3 state. The handshake procedure is described in 12.3.2 and ITU-T Rec. G.994.1 [2].

During the channel discovery, training, and channel analysis & exchange phases of initialization, the VTUs train their respective transceivers after identifying the common mode of operation. During these phases, the transceivers identify channel conditions, exchange parameters for showtime operation, etc. After successful completion of the initialization procedure, the transceivers transition to the L0 state (showtime). Upon unsuccessful completion of the initialization procedure, the VTUs return to the L3 state. The initialization phases are described in 12.3.3 through 12.3.5.

VDSL2 at TQD_TX00153511, § 12.1.3 “Initialization procedures.”

234. The initialization procedure includes the VTU-O sending a message called the O-MSG1 to the VTU-R. The O-MSG1 includes information about the VTU-O’s capabilities such as

interleaving/deinterleaving depths supported by the VTU-O and the maximum interleaver delay the VTU-O can use.

12.3.5.2.1.1 O-MSG 1

The O-MSG 1 message contains the capabilities of the VTU-O and the requirements for downstream transmission (such as margin). The full list of parameters carried by the O-MSG 1 message is shown in Table 12-40.

Table 12-40/G.993.2 – Description of message O-MSG 1

	Field name	Format
1	Message descriptor	Message code
2	Downstream target SNR margin (TARSNRMds)	2 bytes
3	Downstream minimum SNR margin (MINSNRMds)	2 bytes
4	Downstream maximum SNR margin (MAXSNRMds)	2 bytes
5	RA-MODE	1 byte
6	NTR	1 byte
7	TPS-TC capabilities	see Table 12-41
8	PMS-TC capabilities	see Table 12-43

Table 12-41/G.993.2 – TPS-TC capabilities of the VTU-O

Field name	Format	Description
Maximum number of downstream TPS-TCs of each type	1 byte: [ssaapp00]	Indicates the maximum number of TPS-TCs of each type that the VTU-O supports in the downstream direction: - ss=max number of downstream STM TPS-TCs (0,1,2); - aa=max number of downstream ATM TPS-TCs (0,1,2); and - pp=max number of downstream PTM TPS-TCs (0,1,2)
Maximum number of upstream TPS-TCs of each type	1 byte: [ssaapp00]	Indicates the maximum number of TPS-TCs of each type that the VTU-O supports in the upstream direction: - ss=max number of upstream STM TPS-TCs (0,1,2); - aa=max number of upstream ATM TPS-TCs (0,1,2); and - pp=max number of upstream PTM TPS-TCs (0,1,2)
Supported combinations of downstream bearer channels and TPS-TCs	1 byte: [s ₀ a ₀ p ₀ 0 s ₁ a ₁ p ₁ 0]	s ₀ : equal to 1 if STM can be supported on bearer channel 0 a ₀ : equal to 1 if ATM can be supported on bearer channel 0 p ₀ : equal to 1 if PTM can be supported on bearer channel 0 s ₁ : equal to 1 if STM can be supported on bearer channel 1 a ₁ : equal to 1 if ATM can be supported on bearer channel 1 p ₁ : equal to 1 if PTM can be supported on bearer channel 1
Supported combinations of upstream bearer channels and TPS-TCs	1 byte: [s ₀ a ₀ p ₀ 0 s ₁ a ₁ p ₁ 0]	s ₀ : equal to 1 if STM can be supported on bearer channel 0 a ₀ : equal to 1 if ATM can be supported on bearer channel 0 p ₀ : equal to 1 if PTM can be supported on bearer channel 0 s ₁ : equal to 1 if STM can be supported on bearer channel 1 a ₁ : equal to 1 if ATM can be supported on bearer channel 1 p ₁ : equal to 1 if PTM can be supported on bearer channel 1
For each supported TPS-TC, a bearer channel descriptor (see Table 12-42) shall be appended to the message.		
Downstream STM TPS-TC capabilities	0, 1, or 2 bearer channel descriptors	Contains the capabilities of the supported downstream STM TPS-TCs.
Downstream ATM TPS-TC capabilities	0, 1, or 2 bearer channel descriptors	Contains the capabilities of the supported downstream ATM TPS-TCs.
Downstream PTM TPS-TC capabilities	0, 1, or 2 bearer channel descriptors	Contains the capabilities of the supported downstream PTM TPS-TCs.
Upstream STM TPS-TC capabilities	0, 1, or 2 bearer channel descriptors	Contains the capabilities of the supported upstream STM TPS-TCs.
Upstream ATM TPS-TC capabilities	0, 1, or 2 bearer channel descriptors	Contains the capabilities of the supported upstream ATM TPS-TCs.
Upstream PTM TPS-TC capabilities	0, 1, or 2 bearer channel descriptors	Contains the capabilities of the supported upstream PTM TPS-TCs.
NOTE – The number of bearer channel descriptors for the TPS-TC capabilities depends on the fields "Maximum number of downstream/upstream TPS-TCs".		

Table 12-42/G.993.2 – Bearer channel descriptor

Octet	Content of field
1-2	Minimum net data rate (<i>net_min_n</i>)
3-4	Maximum net data rate (<i>net_max_n</i>)
5-6	Reserved net data rate (<i>net_reserve_n</i>) (Note)
7	Maximum interleaving delay
8	Impulse noise protection
9	TPS-TC options
NOTE – This parameter is not used in this version of this Recommendation and shall be set to the value of the minimum net data rate in octets 1 and 2. The OLR procedures that utilize this parameter will be defined in a future revision of this Recommendation.	

Table 12-43/G.993.2 – PMS-TC capabilities of the VTU-O

Field name	Format	Description
Downstream dynamic interleaver	1 byte	Support of dynamic change of interleaver depth in the downstream direction (see 9.4.1). A value of 00 ₁₆ indicates not supported. All other values are for further study.
Upstream dynamic interleaver	1 byte	Support of dynamic change of interleaver depth in the upstream direction (see 9.4.1). A value of 00 ₁₆ indicates not supported. All other values are for further study.
Downstream message overhead data rate	1 byte	Minimum message overhead data rate that is needed by the VTU-O in the downstream direction. The unsigned 8-bit value is the message overhead data rate divided by 1000 bits per second minus 1 (covering the range 1 to 256 kbit/s).
Upstream message overhead data rate	1 byte	Minimum message overhead data rate that is needed by the VTU-O in the upstream direction. The unsigned 8-bit value is the message overhead data rate divided by 1000 bits per second minus 1 (covering the range 1 to 256 kbit/s).

Table 12-43/G.993.2 – PMS-TC capabilities of the VTU-O

Field name	Format	Description
Max DS net data rate for latency path 0	2 bytes	Parameter block of 2 octets that describes the maximum downstream net data rate supported in latency path #0. The unsigned 16-bit value is the net data rate divided by 8000 bits per second.
Max US net data rate for latency path 0	2 bytes	Parameter block of 2 octets that describes the maximum upstream net data rate supported in latency path #0. The unsigned 16-bit value is the net data rate divided by 8000 bits per second.
Max DS net data rate for latency path 1	2 bytes	Parameter block of 2 octets that describes the maximum downstream net data rate supported in latency path #1. The unsigned 16-bit value is the net data rate divided by 8000 bits per second.
Max US net data rate for latency path 1	2 bytes	Parameter block of 2 octets that describes the maximum upstream net data rate supported in latency path #1. The unsigned 16-bit value is the net data rate divided by 8000 bits per second.
DS (1/S) _{max}	1 byte	Parameter block of 1 octet that describes the maximum value of 1/S supported by the VTU-O in the downstream direction as defined in 9.5.5. The unsigned 8-bit value is coded as 1 to 64 in steps of 1.
US (1/S) _{max}	1 byte	Parameter block of 1 octet that describes the maximum value of 1/S supported by the VTU-O in the upstream direction as defined in 9.5.5. The unsigned 8-bit value is coded as 1 to 64 in steps of 1.
NOTE – If only one latency path is supported, the values for latency path 1 shall be set to ZERO.		

VDSL2 at TQD_TX00153564-568, § 12.3.5.2.1.1 “O-MSG1” (partial) and Tables 12-40, 12-41, 12-42, and 12-43.

235. The VTU-R responds to the O-MSG1 by sending the VTU-O a message called the R-MSG2.

12.3.5.2.2.1 R-MSG 2

The R-MSG 2 message conveys VTU-R information to the VTU-O. The full list of parameters carried by the R-MSG 2 message is shown in Table 12-49.

Table 12-49/G.993.2 – Description of message R-MSG 2

	Field name	Format
1	Message descriptor	Message code
2	TPS-TC capabilities	See Table 12-50
3	PMS-TC capabilities	See Table 12-51

Table 12-50/G.993.2 – TPS-TC capabilities of VTU-R

Field name	Format	Description
Maximum number of downstream TPS-TCs of each type	1 byte: [ssaapp00]	Indicates the maximum number of TPS-TCs of each type that the VTU-R supports in the downstream direction: - ss=max number of downstream STM TPS-TCs (0,1,2); - aa=max number of downstream ATM TPS-TCs (0,1,2); and - pp=max number of downstream PTM TPS-TCs (0,1,2).
Maximum number of upstream TPS-TCs of each type	1 byte: [ssaapp00]	Indicates the maximum number of TPS-TCs of each type that the VTU-R supports in the upstream direction: - ss=max number of upstream STM TPS-TCs (0,1,2); - aa=max number of upstream ATM TPS-TCs (0,1,2); and - pp=max number of upstream PTM TPS-TCs (0,1,2).
Supported combinations of downstream bearer channels and TPS-TCs	1 byte: [s ₀ a ₀ p ₀ 0 s ₁ a ₁ p ₁ 0]	s ₀ : equal to 1 if STM can be supported on bearer channel 0 a ₀ : equal to 1 if ATM can be supported on bearer channel 0 p ₀ : equal to 1 if PTM can be supported on bearer channel 0 s ₁ : equal to 1 if STM can be supported on bearer channel 1 a ₁ : equal to 1 if ATM can be supported on bearer channel 1 p ₁ : equal to 1 if PTM can be supported on bearer channel 1
Supported combinations of upstream bearer channels and TPS-TCs	1 byte: [s ₀ a ₀ p ₀ 0 s ₁ a ₁ p ₁ 0]	s ₀ : equal to 1 if STM can be supported on bearer channel 0 a ₀ : equal to 1 if ATM can be supported on bearer channel 0 p ₀ : equal to 1 if PTM can be supported on bearer channel 0 s ₁ : equal to 1 if STM can be supported on bearer channel 1 a ₁ : equal to 1 if ATM can be supported on bearer channel 1 p ₁ : equal to 1 if PTM can be supported on bearer channel 1
For each supported TPS-TC, a bearer channel descriptor (see Table 12-42) shall be appended to the message.		
Downstream STM TPS-TC capabilities	0, 1, or 2 bearer channel descriptors	Contains the capabilities of the supported downstream STM TPS-TCs.
Downstream ATM TPS-TC capabilities	0, 1, or 2 bearer channel descriptors	Contains the capabilities of the supported downstream ATM TPS-TCs.
Downstream PTM TPS-TC capabilities	0, 1, or 2 bearer channel descriptors	Contains the capabilities of the supported downstream PTM TPS-TCs.
Upstream STM TPS-TC capabilities	0, 1, or 2 bearer channel descriptors	Contains the capabilities of the supported upstream STM TPS-TCs.
Upstream ATM TPS-TC capabilities	0, 1, or 2 bearer channel descriptors	Contains the capabilities of the supported upstream ATM TPS-TCs.
Upstream PTM TPS-TC capabilities	0, 1, or 2 bearer channel descriptors	Contains the capabilities of the supported upstream PTM TPS-TCs.
NOTE – The number of bearer channel descriptors for the TPS-TC capabilities depends on the fields "Maximum number of downstream/upstream TPS-TCs".		

Table 12-51/G.993.2 – PMS-TC capabilities of VTU-R

Field name	Format	Description
Downstream dynamic interleaver	1 byte	Support of dynamic change of interleaver depth in the downstream direction (see 9.4.1). A value of 00 ₁₆ indicates not supported. All other values are for further study.
Upstream dynamic interleaver	1 byte	Support of dynamic change of interleaver depth in the upstream direction (see 9.4.1). A value of 00 ₁₆ indicates not supported. All other values are for further study.
Downstream message overhead data rate	1 byte	Minimum message overhead data rate that is needed by the VTU-R in the downstream direction. The unsigned 8-bit value is the message overhead data rate divided by 1000 bits per second minus 1 (covering the range 1 to 256 kbit/s).
Upstream message overhead data rate	1 byte	Minimum message overhead data rate that is needed by the VTU-R in the upstream direction. The unsigned 8-bit value is the message overhead data rate divided by 1000 bits per second minus 1 (covering the range 1 to 256 kbit/s).
Max DS net data rate for latency path 0	2 bytes	The maximum downstream net data rate supported in latency path #0. The unsigned 16-bit value is the net data rate divided by 8000 bits per second.
Max US net data rate for latency path 0	2 bytes	The maximum upstream net data rate supported in latency path #0. The unsigned 16-bit value is the net data rate divided by 8000 bits per second.
Max DS net data rate for latency path 1	2 bytes	Parameter block of 2 octets that describes the maximum downstream net data rate supported in latency path #1. The unsigned 16-bit value is the net data rate divided by 8000 bits per second.
Max US net data rate for latency path 1	2 bytes	Parameter block of 2 octets that describes the maximum upstream net data rate supported in latency path #1. The unsigned 16-bit value is the net data rate divided by 8000 bits per second.
DS (1/S) _{max}	1 byte	Parameter block of 1 octet that describes the maximum value of 1/S supported by the VTU-R in the downstream direction as defined in 9.5.5. The unsigned 8-bit value is coded as 1 to 64 in steps of 1.

Table 12-51/G.993.2 – PMS-TC capabilities of VTU-R

Field name	Format	Description
US (1/S) _{max}	1 byte	Parameter block of 1 octet that describes the maximum value of 1/S supported by the VTU-R in the upstream direction as defined in 9.5.5. The unsigned 8-bit value is coded as 1 to 64 in steps of 1.
NOTE – If only one latency path is supported, the values for latency path 1 shall be set to ZERO.		

VDSL2 at TQD_TX00153573-576, § 12.3.5.2.2.1 “R-MSG2” (partial) and Tables 12-49, 12-50, and 12-51.

236. VDSL2’s initialization messages (including the O-MSG1 and R-MSG2) may be segmented, with each segment being given a unique identifier. The receiving transceiver will acknowledge each segment received, including an identification of the segments being acknowledged, or may transmit a repeat request for particular segments not correctly received.

12.2.6 Segmentation of messages

Messages that are larger than the maximum allowed size (1024 bytes) shall be segmented before transmission; messages shorter than 1024 bytes may also be segmented to improve robustness. To allow segmentation, a segmentation index is included in the control field of the HDLC frame. The four MSBs of this field shall indicate the number of segments, to a maximum of 15, into which the message has been segmented. The four LSBs of this field shall indicate the index of the current segment, starting from 116. For example, a segmentation index value of 9316 indicates the third segment of a total of nine. In case the message is not segmented, the value of the field shall be 1116.

In RQ mode, an acknowledgement (O/R-ACK-SEG) shall be sent for all but the last segment. Typically, the last segment signals the end of the message and will therefore be acknowledged by the reply to the message. The O/R-ACK-SEG message (see Table 12-2) shall be used to acknowledge the reception of the other segments. The O/R-ACK-SEG message shall have its message index assigned by the generic rule defined in 12.2.2.2, and shall be increased by 1 when a new segment is received. The segmentation index of each O/R-ACK-SEG message shall be set to 1116. Once acknowledged, segments shall not be retransmitted and re-transmission shall not be requested.

In AR mode, segmentation shall be done in the same way, but there will be no acknowledgements (O/R-ACK-SEG) between different segments of the same message. Segments shall be sent in sequential order. All segments shall be sent before the message is repeated.

VDSL2 at TQD_TX00153515-516, § 12.2.6 “Segmentation of messages.”

1. O-PMS Message

237. The VDSL2 initialization procedure uses other messages in addition to the O-MSG1 and R-MSG2. One of these messages is the “O-PMS” message. The O-PMS message is transmitted from the VTU-O to the VTU-R.

12.3.5.2.1.3 O-PMS

The O-PMS message conveys the initial PMS-TC parameter settings that shall be used in the upstream direction during Showtime. It also specifies the portion of shared interleaver memory that the VTU-R can use to de-interleave the downstream data stream. The full list of parameters carried by the O-PMS message is shown in Table 12-46.

Table 12-46/G.993.2 – Description of message O-PMS

	Field name	Format
1	Message descriptor	Message code
2	MSGLP	1 byte
3	Mapping of bearer channels to latency paths	1 byte
4	B _{x0}	1 byte
5	B _{x1}	1 byte
6	LP ₀	Latency path descriptor
7	LP ₁	Latency path descriptor
8	MaxD ₀	3 bytes
9	MaxD ₁	3 bytes

Field #1 "Message descriptor" is a unique one-byte code that identifies the message. See Table 12-2 for a complete list of codes.

Field #2 "MSGLP" is a one-byte field that indicates which latency path is selected for OH frames of Type 1 (which carries message overhead) in the upstream direction. The seven MSBs of the byte shall always be set to ZERO. The LSB shall be set to ZERO to indicate latency path #0 or ONE to indicate latency path #1.

Field #3 "Mapping of bearer channels to latency paths" is a one-byte field that indicates which bearer channels shall be carried in each of the upstream latency paths. The byte is denoted as [cccc dddd]. The bits cccc shall be set to 0000 if bearer channel #0 is to be carried in latency path #0, and to 0001 if bearer channel #0 is to be carried in latency path #1. The bits cccc shall be set to 1111 if the bearer channel #0 is disabled. The bits dddd indicate which latency path carries bearer channel #1 using the same encoding method as used for cccc.

Field #4 "B_{x0}" is a one-byte field that indicates the number of octets from bearer channel #0 that shall be transported in each MDF in the upstream direction. The value shall be either zero or the non-zero value from the set {B00, B10}.

Field #5 "B_{x1}" is a one-byte field that indicates the number of octets from bearer channel #1 that shall be transported in each MDF in the upstream direction. The value shall be either zero or the non-zero value from the set {B01, B11}.

Field #6 "LP₀" is a 10-byte field that contains the PMS-TC parameters for latency path #0 in the upstream direction. The "Latency path descriptor" format specified in Table 12-47 shall be used.

Field #7 "LP₁" is a 10-byte field that contains the PMS-TC parameters for latency path #1 in the upstream direction. The "Latency path descriptor" format specified in Table 12-47 shall be used. If latency path #1 is not used, all bytes of LP₁ shall be set to ZERO.

Field #8 "MaxD0" is a 3-byte field that specifies the maximum interleaver delay that the VTU-R shall be allowed to use to de-interleave the data stream in downstream latency path #0. The maximum interleaver delay shall be specified in bytes as an unsigned integer.

Field #9 "MaxD1" is a 3-byte field that specifies the maximum interleaver delay that the VTU-R shall be allowed to use to de-interleave the data stream in downstream latency path #1. The maximum interleaver delay shall be specified in bytes as an unsigned integer. If the value of this field is FFFFFFFF16, the VTU-R shall autonomously partition the interleaver delay specified in Field #8 (MaxD0) between both downstream latency paths.

The latency path descriptor is described in Table 12-47. It contains the primary parameters of the framer, as specified in Table 9-6, and the interleaver settings for one latency path. All values are unsigned integers.

Table 12-47/G.993.2 – Latency path descriptor

Octet	Field	Format	Description
1	T	1 byte	The number of MDFs in an OH sub-frame for the latency path; $T = k \times M$, where k is an integer. The value of T shall not exceed 64.
2	G	1 byte	The total number of overhead octets in an OH sub-frame for the latency path; $1 \leq G \leq 32$.
3	F	1 byte	Number of OH frames in the OH superframe for the latency path. $1 \leq F \leq 255$.
4	M	1 byte	The number of MDFs in an RS codeword for the latency path. Only the values 1, 2, 4, 8, 16 are allowable.
5 & 6	L	2 bytes	Contains the value of L for the latency path.
7	R	1 byte	Contains the value of R for the latency path.
8	I	1 byte	Contains the value of I for the latency path.
9 & 10	D	2 bytes	Interleaver depth D for the latency path.

VDSL2 at TQD_TX00153570-572§ 12.3.5.2.1.3 “O-PMS.”

238. Fields 8 and 9 in the O-PMS message specify the maximum interleaver delay allowed in the downstream latency paths. *Id.*

IX. THE FAMILY 9 PATENTS

A. Overview of the Family 9 Patents and Asserted Claims

239. I have been informed by counsel that TQ Delta is asserting claims 10 and 18 of the '411 Patent, claim 1 of the '348 Patent, and claims 4 and 6 of the '809 Patent.

240. The application that matured into the '411 Patent was filed on May 20, 2010. The '411 Patent, entitled "Packet Retransmission," issued on June 18, 2013.

241. The asserted claims of the '411 Patent recite as follows:

'411 Patent, claim 10

A transceiver capable of packet retransmission comprising:

a transmitter portion capable of: transmitting a plurality of packets, identifying at least one packet of the plurality of packets as a packet that should be retransmitted and allocating a memory between a retransmission function and an interleaving and/or deinterleaving function, wherein at least a portion of the memory may be allocated to the retransmission function or to the interleaving and/or deinterleaving function at any one particular time, and wherein a message transmitted during initialization indicates how the memory has been allocated between the retransmission function and the interleaving and/or deinterleaving function in the transceiver.

'411 Patent, claim 18

A transceiver capable of packet retransmission comprising:

a receiver portion capable of: receiving a plurality of packets, identifying at least one packet of the plurality of packets as a packet that should be retransmitted and allocating a memory between a retransmission function and an interleaving and/or deinterleaving function

wherein the memory is allocated between the interleaving function and the deinterleaving function in accordance with a message received during an initialization of the transceiver and wherein at least a portion of the memory may be allocated between the retransmission function and the interleaving and/or deinterleaving function at any one particular time depending on the message.

242. The application that matured into the '348 Patent was filed on November 8, 2013.

The '348 Patent, entitled "Packet Retransmission," issued on July 28, 2015.

243. The asserted claim of the '348 Patent recite as follows:

'348 Patent, claim 1

An apparatus comprising:

a multicarrier transceiver including a processor and memory operable to:

transmit a packet using a forward error correction encoder and an interleaver, wherein the packet comprises a header field and a plurality of PTM-TC codewords, a plurality of ATM cells or a plurality of Reed-Solomon codewords, and wherein the header field comprises a sequence identifier (SID); and

receive a plurality of messages using a forward error correction decoder and without using a deinterleaver, wherein each message of the plurality of messages is received in a different DMT symbol and wherein at least one message of the plurality of messages includes an acknowledgement (ACK) or a negative acknowledgement (NACK) of the transmitted packet.

244. The application that matured into the '809 Patent was filed on December 3, 2019.

The '809 Patent, entitled "Techniques for Packet and Message Communication in a Multicarrier Transceiver Environment," issued on November 10, 2020.

245. The asserted claims of the '809 Patent recite as follows:

'809 Patent, claim 4⁴

The apparatus of claim 1, wherein a physical layer of the transceiver is capable of generating the packet and the message.

'809 Patent, claim 6⁵

The apparatus of claim 2, wherein a physical layer of the transceiver is capable of generating the packet and the message.

⁴ This claim depends from claim 1, which states:

An apparatus comprising:

a multicarrier transceiver including a processor and memory capable of:

transmitting a packet using forward error correction encoding and interleaving, wherein the packet comprises a header field and a plurality of Reed-Solomon codewords, and wherein the header field comprises a sequence identifier (SID); and

receiving a message using forward error correction decoding and without using deinterleaving, wherein the message is received in a single DMT symbol, and wherein the message includes an acknowledgement (ACK) or a negative acknowledgement (NACK) of the transmitted packet.

⁵ This claim depends from claim 2, which states: The apparatus of claim 1, wherein the received message has a higher immunity to noise than the transmitted packet.

B. File History

246. I have reviewed the prosecution histories of the Patents-in-Suit, and provide short summaries below. I note that, in all of the prosecution histories, Aware (the original assignee) and TQ Delta submitted multiple, lengthy Information Disclosure Statements (“IDS”) identifying prior art.

1. The ’411 Patent

247. I have reviewed the prosecution history of the ’411 Patent, focusing on the asserted claims. I present a brief summary below.

248. On May 20, 2010, the application for the ’411 Patent was filed. ’411 Pros. Hist. The then-pending claim was rejected on May 17, 2012 for double-patenting over U.S. Patent Application No. 2012/760,728 (which eventually issued as the ’577 Patent). ’411 Pros. Hist.

249. On November 16, 2012, the applicant filed a request for reconsideration amending the rejected claim and adding additional independent and dependent claims. ’411 Pros. Hist. The applicant argued that the additional limitation in the rejected claim, which required that the transceiver share memory with an interleaving/deinterleaving function as well as the retransmission function, overcame the double patenting rejection. *Id.*

250. On December 17, 2012, the examiner rejected claims 106, 107, 109-114, 116-121, and 123-126 under 102(b) as anticipated by Moon (U.S. Patent Application Publication No. 2003/0076870); claims 107, 114, and 121 were also rejected for non-statutory obviousness-type double patenting over U.S. Patent Application No. 12/783,758. ’411 Pros. Hist.

251. I note that in rejecting claim 107 the examiner determined that Moon teaches sharing a memory between a retransmission function and an interleaving/deinterleaving function. This is shown below in an excerpt from the ’411 Pros. Hist.

7. In regard to claim 107, Moon teaches:

- A method of packet retransmission comprising;
- transmitting or receiving, using a transceiver, a plurality of packets;

(Figure 9A, ref. S1 & S2 & S3 & S4 & P1 & P2 & P3 & P4 and section [0086] in Moon)

- identifying at least one packet of the plurality of packets as a packet that should be retransmitted; and

(Packets S1 & S2 & S3 in figure 9A in Moon)

- sharing a memory between a retransmission function and an interleaving/deinterleaving function.

(Section [0059] in Moon)

'411 Pros. Hist. The examiner indicated that claims 108, 115, and 122 would be allowable if they were amended to not depend from the rejected claims. '411 Pros. Hist. The examiner indicated that transmitting "a message during initialization, the message indicating how to share the memory between the retransmission function and the interleaving/deinterleaving function" rendered claims 108, 115, and 122 patentable. *Id.*

252. On April 24, 2013, the applicant amended claims 107 and 114 to incorporate the limitation found in claim 108 and 115, respectively, and added additional claims including the limitation of claims 108, 115, and 122. '411 Pros. Hist.

253. Subsequently, on May 9, 2013, the examiner issued a notice of allowance for all claims. '411 Pros. Hist. Pending claim 114 issued as claim 10, and pending claim 132 issued as claim 18. *Id.*

2. The '348 Patent

254. I have reviewed the prosecution history of the '348 Patent, focusing on the asserted claims. I present a brief summary below.

255. On November 8, 2013, the application for the '348 Patent was filed. The earliest priority date was April 12, 2006. Prosecution History of U.S. Patent Application No. 14/075,194 (“’194 application”), 305-30.

256. On January 2, 2014, the applicant filed a preliminary amendment deleting claims 1-105, and adding new claims 106-33. *Id.* 322-27.

257. I note there was only one Non-Final Office Action (“NFOA”) that substantively examined U.S. Patent App. No. 14/075,194, which was issued on January 28, 2015. The USPTO rejected claims 106-121 pursuant to 35 U.S.C. §101 as being directed to “software, per se,” which I understand is a type of non-statutory subject matter. *Id.* 354. The NFOA also allowed claims 122-133. *Id.*

258. On February 4, 2015, the applicant responded to the NFOA, and amended claims 106, 114, 122, and 128. *Id.* 407-12. The applicant did not change claims 107-13, 115-21, 123-27, and 129-33. *See id.* In the response, the applicant stated that claims 106 and 114 were amended to include a processor and memory to overcome the previous §101 rejection. *See id.* 412. The amendments to claims 106, 114, 122, and 128 are shown below:

106. (Currently Amended) An apparatus comprising:

a multicarrier transceiver including a processor and memory operable to:

transmit a packet using a forward error correction encoder and an interleaver, wherein the packet comprises a header field and a plurality of PTM-TC codewords ~~or~~, a plurality of ATM cells or a plurality of Reed-Solomon codewords, and wherein the header field comprises a sequence identifier (SID); and

receive a plurality of messages using a forward error correction decoder and without using a deinterleaver, wherein each message of the plurality of messages is received

in a different DMT symbol and wherein at least one message of the plurality of messages includes an acknowledgement (ACK) or a negative acknowledgement (NACK) of the transmitted packet.

114. (Currently Amended) An apparatus comprising:

a multicarrier transceiver including a processor and memory operable to:

receive a packet using a forward error correction decoder and a deinterleaver, wherein the packet comprises a header field and a plurality of PTM-TC codewords ~~or~~, a plurality of ATM cells or a plurality of Reed-Solomon codewords, and wherein the header field comprises a sequence identifier (SID); and

transmit a plurality of messages using a forward error correction encoder and without using an interleaver, wherein each message of the plurality of messages is transmitted in a different DMT symbol and wherein at least one message of the plurality of messages includes an acknowledgement (ACK) or a negative acknowledgement (NACK) of the received packet.

122. (Currently Amended) A method, in a multicarrier transceiver, comprising:

transmitting a packet using a forward error correction encoder and an interleaver, wherein the packet comprises a header field and a plurality of PTM-TC codewords ~~or~~, a plurality of ATM cells or a plurality of Reed-Solomon codewords, and wherein the header field comprises a sequence identifier (SID); and

receiving a plurality of messages using a forward error correction decoder and without using a deinterleaver, wherein each message of the plurality of messages is received in a different DMT symbol and wherein at least one message of the plurality of messages includes an acknowledgement (ACK) or a negative acknowledgement (NACK) of the transmitted packet.

128. (Currently Amended) A method, in a multicarrier transceiver, comprising:

receiving a packet using a forward error correction decoder and a deinterleaver, wherein the packet comprises a header field and a plurality of PTM-TC codewords ~~or~~, a plurality of ATM cells or a plurality of Reed-Solomon codewords, and wherein the header field comprises a sequence identifier (SID); and

transmitting a plurality of messages using a forward error correction encoder and without using an interleaver, wherein each message of the plurality of messages is transmitted in a different DMT symbol and wherein at least one message of the plurality of messages includes an acknowledgement (ACK) or a negative acknowledgement (NACK) of the received packet.

See id. 408-11.

259. On April 8, 2015, the USPTO issued a Notice of Allowance for U.S. Patent Application No. 14/075,194, indicating that claims 106-33 were allowed. *See id.* 417, 421. These claims were renumbered as claims 1-27 of the '348 Patent. I note the USPTO did not provide any reason for allowance.

260. Additionally, after the '348 Patent issued on July 28, 2015, the applicant requested, on August 28, 2015, a Certificate of Correction to remedy a typographical error in claim 23 (the term "HACK" was corrected to "NACK"). *See id.* 499. The Certificate of Correction was mailed on December 22, 2015. *Id.* 504.

261. During prosecution, the applicant disclosed and the USPTO considered several prior art references, including G.993.2, G.992.3, and G.992.5. *Id.* 211, 361. These were not subject to any substantive analysis by the USPTO, however.

3. The '809 Patent

262. I have reviewed the prosecution history of the '809 Patent, focusing on the asserted claims. I present a brief summary below.

263. On December 3, 2019, the application for the '809 Patent was filed. The earliest priority date was April 12, 2006. Prosecution History of U.S. Patent Application No. 16/701,343 ("343 application"), 107.

264. On December 3, 2019, the applicant also filed a preliminary amendment that deleted claims 1-105 and added new claims 106-33. *See id.* 1-7.

265. I note that there was only one substantive office action issued during prosecution of the '343 application. In particular, on February 2, 2020, the USPTO examiner issued a NFOA on objecting to the title and rejecting claims 106-133 on non-statutory double patenting grounds as being unpatentable over claims 1-28 of U.S. Patent No. 9,094,348 and claims 1-28 of U.S. Patent No. 9,749,235. *Id.* 114-119. The NFOA also suggested that the applicant may timely file a terminal

disclaimer in compliance with 37 CFR §§ 1.321(c) or 1.321(d) to overcome an actual or provisional rejection based on non-statutory double patenting—provided that the reference application or patent is shown to be commonly owned with the examined application, or claimed an invention made as a result of activities undertaken within the scope of a joint research agreement. *Id.*

266. The applicant responded to the NFOA on April 21, 2020. *Id.* 178-186. In its response, the applicant amended the title and also submitted an Electronic Terminal Disclaimer in view of U.S. Patent Nos. 9,094,348 and 9,749,235, which the applicant indicated were filed and approved. *Id.* 385-391. Claims 106-133 were left unchanged. *See id.*

267. On April 1, 2020 and May 25, 2020, the applicant also submitted a supplemental information disclosure statement. *Id.* 313-351, 395-396. Those statements, along with the corresponding transmittal letters, described cited non-patent references, foreign references, patent application references, and patent application publication references. *Id.*

268. On July 23, 2020, the USPTO examiner allowed the claims in an Ex Parte Quayle Action. *Id.* 410-415. In particular, pursuant to 37 C.F.R. § 1.105, the examiner requested that the applicant specify which of the cited references (from the information disclosure statements filed on April 1, 2020 and May 25, 2020) disclose “receiving a packet using FEC decoding and deinterleaving” and “transmitting a message using FEC encoding and without using interleaving,” or alternatively, “transmitting a packet using FEC encoding and interleaving” and “receiving a message using FEC decoding and without using deinterleaving” wherein “the message is received in a single DMT symbol.” The examiner also requested the applicant identify the 15 to 20 references that were most relevant to the claims in the application, including specifying any documents that involve the application or direct parents. Additionally, the examiner stated that claims 106-133 were allowed, and that the application was in condition for allowance except for pending formal matters

that included the request described above, and that prosecution on the merits was closed in accordance with the practice under Ex Parte Quayle, 25 USPQ 74, 453 O.G. 213, (Comm’r Pat. 1935). *Id.*

269. On July 31, 2020, the applicant responded to the Ex Parte Quayle Action, thanking the examiner for clarifications offered during a July 23, 2020 telephone interview. *Id.* 425-426. I understand that the applicant discussed the background of the application and explained how the submitted references were material to that application. *See id.*

270. On August 8, 2020, the USPTO issued a Notice of Allowance for U.S. Patent Application No. 16/701,343, indicating that claims 106-133 were allowed. *See id.* 446. These claims were renumbered as claims 1-28 of the ’809 Patent. I note the USPTO did not provide any reason for allowance.

271. The USPTO issued the ’809 Patent on November 10, 2020. *See id.* 529.

272. During prosecution, the applicant disclosed and the USPTO considered several prior art references, including G.993.2, G.992.3, and G.992.5. *Id.* 323-324. These were not subject to any substantive analysis by the USPTO, however.

C. Claim Construction

273. I understand that the Court issued an Order in this case construing certain terms, which are set forth below.

Patent and Claim	Term or Phrase	Court’s Construction
’411 Patent, claims 10 and 18; ’348 Patent, claim 1; ’809 Patent, claims 4 and 6; ’055 Patent, claim 17	“transceiver”	“communications device capable of transmitting and receiving data wherein the transmitter portion and receiver portion share at least some common circuitry”
’348 Patent, claim 1	“operable to”	“configured to”

Patent and Claim	Term or Phrase	Court's Construction
'348 Patent, claim 1; '055 Patent, claim 17	"PTM-TC [(Packet Transfer Mode Transmission Convergence)] codewords"	plain meaning
'348 Patent, claim 1	"transmit a packet using a forward error correction encoder and an interleaver"	plain meaning

274. In my analysis, I have applied the Court's constructions. I have interpreted the remaining claim terms as they would have been understood by a person having ordinary skill in the art on the priority date of the Family 9 Patents, considering the context of the claim itself, the specification, the figures, the prior art, and the prosecution history. Consistent with these constructions and interpretations, I have considered the claims in light of the ordinary meaning of the claims based on the perspective of a POSITA and consistent with my experience in the field.

X. OVERVIEW OF THE PRIOR ART

275. Below are summaries of the prior art that, in my opinion, renders obvious the asserted claims of the '411 Patent, the '348 Patent, and the '809 Patent.

A. ITU-T Standard G.993.1 ("VDSL1 Standard") (NOK00078763)

276. See my discussion of the VDSL1 Standard in the Technology Background section above. § VII.C *supra*.

277. G.993.1 is entitled "Very high speed digital subscriber line transceivers" and was officially approved on June 13, 2004, by the ITU-T Study Group 15. G.993.1 at Cover; *id.* at i. Drafts of G.993.1, including a final draft circulated for consent, were made available to and disseminated amongst ITU members prior to the June 2004 meeting. In fact, a final draft of G.993.1 was disseminated and consented to at the April 2004 Geneva meeting. Because no comments were

received, that draft of G.993.1 entered approved status on June 13, 2004. I thus understand G.993.1 to be prior art as of at least June 13, 2004.

278. For example, a draft version of G.993.1 was disseminated and made available to ITU members as SS-046 and SS-R11 in connection with the January 2004 meeting in Singapore. Additionally, prior to its official approval at the June 2004 meeting, the draft G.993.1 standard was disseminated amongst and made available to ITU members as C-107 and TD 71 (PLEN) for consent in connection with the April 2004 Geneva meeting. I understand that TD 71 (PLEN), which was submitted for consent, was ultimately approved without comment at the June 2004 meeting. The agenda for the January 2001 meeting in Singapore, SS-001, shows that the documents SS-046 and SS-R11 were slated for discussion at that meeting. The document list for the January 2004 meeting shows that SS-046 was available to ITU members through the ITU website as of no later than May 5, 2004, and the document was further indexed to the G.vdsl area and findable by descriptive name "Proposed draft for revision of G.993.1." The document list for the April 2004 meeting shows that C-0107 and TD 71 (PLEN) were available to ITU members through the ITU website as of no later than May 13, 2004, and the documents were further indexed to the G.vdsl area and findable by descriptive names "Draft revised Recommendation G.993.1" and "Draft revised Recommendation G.993.1 (for consent)."

279. Thus, the G.993.1 draft versions SS-046, SS-R11, C-0107, and TD 71 (PLEN) were all made available to ITU members prior to the June 2004 ITU meeting where G.993.1 was approved. Each of these draft versions include substantially all of the material that I have relied on in the G.993.1 document including, for example, the description of Reed-Solomon coding in Section 8.3, interleaving in Section 8.4, and the description of configuration messages in Section 12 including the description of R-MSG2 in Section 12.4.6.3.1.1. The documents are very similar

throughout with minor small differences except for differences in material near the end of the documents. For example, TD 71 PLEN includes the four-page Appendix IV: 8.625 kHz Tone Spacing (Informative) at the end that is not present in C-0107 or SS-046. C-0107 includes an “Attachment for G.994.1” at the end that is not present in TD-71 PLEN. Similar to C-0107, SS-046 includes “Annex J Handshake procedure for VDSL (normative)” towards the end that includes similar information to the “Attachment for G.994.1” found at the end of C-0107. None of these minor differences are related in any way to the teachings I have relied on from G.993.1. I further note that the document TD 71 PLEN includes the notation “(for consent)” in the title indicating that this is the version that was proposed at the April 2004 ITU-T SG15 meeting.

280. Evidence regarding the availability of these documents is shown in the Issues List TD-WP1-0021 for G.vdsl and G.vdsl2 from April 2004. In particular, TD-WP1-0021 lists a series of actions (shown below) taken between January 2004 and March 2004 regarding documents SS-046 and C-107. For example, TD-WP1-0021 states that “SS-046 with editorial revisions shall be submitted as a White Contribution to the April 2004 SG15 meeting proposing a revised recommendation G.993.1” and “MC-124R1 contained the agreed changes to C-107 (Draft revised G.993.1) and will be uploaded to the TSB prior to the Geneva meeting.”

1.12	Agreed 20-Jan-2004	Q4/15 agrees to develop a VDSL Recommendation for consent in April 2004 or sooner with full text specification for DMT in the main body and full text specification for QAM in a normative Annex, with the following in the Scope section: It has been agreed in the ITU-T to develop a subsequent VDSL2 Recommendation that specifies only DMT modulation, and is based on ITU Rec. G.993.1-2004 (VDSL) and ITU Rec. G.992.3 (ADSL2).	SS-022

1.13	Agreed 23-Jan-2004	SS-046 with editorial revisions shall be submitted as a White Contribution to the April 2004 SG15 meeting proposing a revised Recommendation G.993.1. Prior to the submission, the editor will perform editorial revisions and notify the committee by the email reflector for review.	SS-046
1.13.1	Agreed 23-Jan-2004	The accepted resolutions in SS-087 shall be applied to the draft revised Recommendation G.993.1. Item #13 and #18 are also accepted. The note in point 21 shall be modified to be: "For system using QAM implementation the DMT specific aspects of this annex is not required."	SS-046
1.13.2	Agreed 23-Jan-2004	Table 6-2 will be replaced with Figure-5(a)/SS-060. Table 5(b) and the picture will be placed in an informative appendix.	SS-046
1.13.3	Agreed 12-Mar-2004	MC-124R1 contained the agreed changes to C-107 (Draft revised G.993.1) and will be uploaded to the TSB prior to the Geneva meeting.	MC-042, MC-053, MC-124

(TD-WP1-0021 at 4-5)

281. As further evidence, the documents list SG15/Q4 for the April 2004 Geneva meeting lists the document C-0107 as the first item on the G.vdsl agenda.

G.vdsl

Num.	Source	Area	Title	Informal Filename	ITU filename
C-0107	G.vdsl Associate Rapporteur	G.vdsl	Draft revised Recommendation G.993.1	- N/A -	T01-SG15-C-0107R1-040420b.doc

Brief Agenda:

Area		Contribution Numbers
Admin	details	D-1252, TD-PLN-0003, TD-PLN-0016, TD-PLN-0017, TD-PLN-0018, TD-PLN-0019, TD-PLN-0020, TD-PLN-0057, TD-PLN-0086, TD-PLN-0087, TD-PLN-0094, TD-PLN-0095, TD-PLN-0096, TD-PLN-0097, TD-PLN-0105, TD-GEN-0001, TD-GEN-0002, TD-GEN-0003, TD-GEN-0004, TD-GEN-0005, TD-GEN-0006, TD-GEN-0007, TD-GEN-0008, TD-GEN-0013, TD-GEN-0014, TD-GEN-0015, TD-GEN-0016, TD-GEN-0017, TD-GEN-0018, TD-GEN-0019, TD-GEN-0020, TD-GEN-0021, TD-GEN-0022, TD-GEN-0023, TD-GEN-0024, TD-GEN-0025, TD-GEN-0026, TD-GEN-0027, TD-GEN-0028, TD-GEN-0029, TD-GEN-0030, TD-GEN-0031, TD-GEN-0032, TD-GEN-0033, TD-GEN-0034, TD-GEN-0035, TD-GEN-0036, TD-GEN-0037, TD-GEN-0038, TD-GEN-0040, TD-GEN-0041, TD-GEN-0042, TD-GEN-0043, TD-GEN-0058, TD-WP1-0001, TD-WP1-0003, TD-WP1-0004, TD-WP1-0005, TD-WP1-0008, TD-WP1-0009, TD-WP1-0015, TD-WP1-0023,
IPR	details	
G.shdsl.bis	details	
G.vdsl	details	C-0107, D-0993, D-0994, D-1015, D-1016, D-1018, D-1019, D-1020, D-1021, D-1023, D-1024, D-1025, D-1037, D-1038, D-1039, D-1041, D-1042, D-1045, D-1046, D-1047, D-1048, D-1049, D-1050, D-1051, D-1052, D-1053, D-1054, D-1060, D-1061, D-1062, D-1067, D-1068, D-1069, D-1079, D-1118, D-1144, D-1145, D-1151, D-1156, D-1180, D-1185, D-1186, D-1187, D-1188, D-1202, D-1219, D-1255, TD-PLN-0071, TD-WP1-0002, TD-WP1-0007, TD-WP1-0020, TD-WP1-0021, TD-WP1-0022, TD-WP1-0028,
G.vdsl	details	
G.voice	details	
G.bond	details	D-1063, D-1119, TD-WP1-0024, TD-WP1-0026,

282. In addition to my review of these documents, I have spoken with ADTRAN's engineers Richard Goodson and Kevin Schneider. Dr. Schneider and Mr. Goodson informed me that both were involved in ADTRAN's standardization activities, including ADTRAN's activities as an ITU member. Mr. Goodson informed me that he attended ITU meetings during the time period covered by the meetings I have referenced above and continuing into the time period of the alleged invention. He confirmed that the established practice of the ITU (including Study Group 15 that was developing the G.993.1 standard) was that as standard approached approval, drafts of the standard are made available through the ITU Telecommunication Information Exchange Service (TIES) website, which is how the documents SS-046, C-107-R1, and TD 71 PLEN were made

available. Such documents are available to any ITU Sector Member companies and Associate Member companies in the appropriate technical area. As described on the ITU website <https://www.itu.int/en/about/Pages/default.aspx>, the ITU today is a community of more than 20,000 professionals and includes 193 Member States as well as some 900 companies. In 2004, the ITU included 189 Member States and 730 companies including 640 that were Sector Members. <https://web.archive.org/web/20040627044216/http://www.itu.int/members/index.html>.

283. Mr. Goodson stated that in addition to documents such as SS-046, C-107-R1, and TD 71 PLEN being available to ITU members through the website, meetings about DSL standards where these documents were discussed prior to the time of the alleged invention were often “standing room only” because those engineers who were interested in implementing DSL technologies wanted to be in the room as decisions were being made in the adoption of a standard.

284. I also spoke with Nokia's engineers Frank Van der Putten, Danny Van Bruyssel, and Paul Spruyt. Mr. Van Bruyssel and Mr. Van der Putten informed me that they were present at the April 2004 meeting where G.993.1 was consented to and the June 2004 meeting where G.993.1 was approved. They informed me that, after approval, the document would undergo no substantive changes. Mr. Van Bruyssel also informed me that, on May 13, 2004, he received an email through an ITU, Study Group 15 email distribution list—tsg15g4@itu.ch. That email pointed to him to the draft version of G.993.1 for consent, which was ultimately approved without comment.

285. Nokia's engineers also informed me that the standard practice of the ITU, Study Group 15, was to access drafts of recommendations, and contributions, on the TIES website. They explained that these “ITU documents are effectively available to anyone that would be interested in getting them.” Based on my conversation with them, and my own independent assessment, I understand that ITU membership is broad and can be achieved in a number of different capacities,

and is intended to include anyone interested in and possessing ordinary skill regarding DSL standards. I also understand that, outside of receiving a membership, one may gain access to ITU documents as a guest of an ITU member.

286. Furthermore, the G.993.1 standard that was approved in June 2004 shares many concepts with the ETSI VDSL1 standard TS 101 270-2 that bears a copyright date of 2003 and was published on ETSI's website to the public at large in July 2003 (specifically July 24, 2003, according to the Work Programme details for the document available through the ETSI website: https://portal.etsi.org/webapp/workprogram/Report_WorkItem.asp?WKI_ID=11828.) The shared concepts found in ETSI VDSL1 that are included in and relevant to my invalidity analysis include support and symmetric and asymmetric rates by a VDSL transceiver (e.g., ETSI VDSL1 § 5.2.5.1.1), the use of Reed-Solomon Coding (e.g., ETSI VDSL1 § 6.4.2), the use of convolutional interleaving (specifically the triangular implementation of convolutional interleaving) (e.g., ETSI VDSL1 §§ 6.4.3.1, 6.4.3.2), and the initialization process (including specifically the O-MSG2, R-MSG2, and "maximum interleaver memory" field used in my analysis) (e.g., ETSI VDSL1 §§ 8.2.1, 8.2.4, 8.2.6 (generally), 8.2.6.2.1.1, 8.2.6.2.1.2, 8.2.6.2.1.3, 8.2.6.3.1.1, 8.2.6.3.1.2, 8.2.6.3.1.3, 8.2.6.3.1.4). For these concepts, ETSI VDSL1 includes substantially the same teachings, and indeed often identical content for concepts like RS coding, interleaving, and the initialization process, as described in the version of ITU 993.1 that was approved in June 2004.

287. I note that the connection between ETSI's VDSL1 standard TS 101 270-2 and ITU VDSL standards was documented, for example, in TD-WP1-0002. TD-WP1-0002 is a letter associated with the April 2004 ITU meeting in Geneva from ETSI TM6 Chairman Manfred Gindel to Paolo Rosa of the ITU describing a compromise proposal that would "enable ITU-T SG15Q4 to develop a VDSL specification similar to TS 101 270-2, but with DMT in the main body and QAM

in a normative Annex, and also to start work on VDSL2 with objectives similar to those agreed in ETSI.” G.993.1 is in fact a VDSL specification similar to TS 101 270-2, but with DMT in the main body and QAM in a normative Annex:

We understand that a compromise proposal described in an informal document named "VDSL a way forward", and discussed at the recent Q15 meeting held in Geneva during October 2003, exists. This could enable ITU-T SG15Q4 to develop a VDSL specification similar to TS 101 270-2, but with DMT in the main body and QAM in a normative Annex, and also to start work on VDSL2 with objectives similar to those agreed in ETSI. TM6 supports this compromise and we hope that it can be achieved in the very near future. (Excerpt from TD-WP1-0002)

288. The document SS-022R3 associated with the January 2004 ITU meeting in Singapore entitled “VDSL: A way forward” indeed includes the agreement to “develop a VDSL Recommendation for consent in April 2004 or sooner with full text specification for DMT in the main body and full text specification for QAM in a normative Annex,” and that recommendation is what was approved in June 2004 as G.993.1. The Issues List for G.vdsl and G.vdsl2 from the January 2004 meeting in Singapore, SS-U11R3 further documents the agreement described in “VDSL: A way forward” and indicates that a revised version of SS-046 (one of the three draft versions of G.993.1 I discussed above) will be the VDSL recommendation proposed for consent at the April 2004 SG15 meeting.

<u>1.12</u>	<u>Agreed</u> <u>20-Jan-2004</u>	<u>Q4/15 agrees to develop a VDSL Recommendation for consent in April 2004 or sooner with full text specification for DMT in the main body and full text specification for QAM in a normative Annex, with the following in the Scope section:</u> <u>It has been agreed in the ITU-T to develop a subsequent VDSL2 Recommendation that specifies only DMT modulation, and is based on ITU Rec. G.993.1-2004 (VDSL) and ITU Rec. G.992.3 (ADSL2).</u>	<u>SS-022</u>
<u>1.13</u>	<u>Agreed</u> <u>23-Jan-2004</u>	<u>SS-046 with editorial revisions shall be submitted as a White Contribution to the April 2004 SG15 meeting proposing a revised Recommendation G.993.1. Prior to the submission, the editor will perform editorial revisions and notify the committee by the email reflector for review.</u>	<u>SS-046</u>
<u>1.13.1</u>	<u>Agreed</u> <u>23-Jan-2004</u>	<u>The accepted resolutions in SS-087 shall be applied to the draft revised Recommendation G.993.1. Item #13 and #18 are also accepted. The note in point 21 shall be modified to be: "For system using QAM implementation the DMT specific aspects of this annex is not required."</u>	<u>SS-046</u>
<u>1.13.2</u>	<u>Agreed</u> <u>23-Jan-2004</u>	<u>Table 6-2 will be replaced with Figure-5(a)/SS-060. Table 5(b) and the picture will be placed in an informative appendix.</u>	<u>SS-046</u>

289. These documents provide further evidence that the teachings I have relied on from G.993.1 approved in June 2004 were published and publicly known no later than June 2004 because 1) substantially identical or identical teachings were published in ETSI VDSL1 in July 2003, 2) that publication was available to the public at large, 3) that publication and its contents were known by ETSI members that approved the ETSI VDSL1 standard prior to July 2003, and 4) the ETSI VDSL1 standard and its contents (including the substantially identical or identical teachings I have relied on from G.993.1) were known by, at a minimum, ITU members as indicated by the references to ETSI's VDSL1 standard during the development of G.993.1.

290. Accordingly, I am informed that G.993.1 qualifies as prior art under at least pre-AIA 35 U.S.C. §102(a) and §102(b) because it was described in a printed publication in this or a foreign country before the earliest priority date for the Family 9 Patents and was published more than one year before the priority date of the Family 9 Patents.

B. ITU-T Standard G.992.3 (“ADSL2 Standard”) (NOK00078327)

291. See my discussion of the ADSL2 Standard in the Technology Background section above. § VII.C *supra*. The ITU-T approved the G.992.3 standard in January 2005. ADSL2, cover page. It was published on their public website on January 4, 2006. For the same reasons described above with respect to the ITU-T G.993.1 standard, I am informed that G.992.3 qualifies as prior art under at least pre-AIA 35 U.S.C. §102(a) and §102(b) because it was described in a printed publication in this or a foreign country before the earliest priority date for the Family 9 Patents and was published more than one year before the priority date of the Family 9 Patents.

C. “Interleaved FEC/ARQ coding for QoS multicast over the Internet” by Chen et al. (“Chen”) (NOK00075559)

292. Chen was published in the July 2004 edition of the *Canadian Journal of Electrical and Computer Engineering*. Chen at NOK0007559. Specifically, I understand that it was published on July 1, 2004. See Appendix D (Hall-Ellis Decl.). Accordingly, I understand that Chen qualifies as §102(b) prior art. Chen describes using a combination of Reed-Solomon coding, interleaving, and automatic repeat request (ARQ) coding to enhance the quality of multicast transmissions over the Internet. *Id.* Chen describes how to apply its interleaved/ARQ method to a unicast system as well. *Id.* NOK00075562, col. 2 ¶ 2.

293. Chen includes a sequence number in data packets.

The frames transmitted from the sender (first sender or intermediate DR) are shown in Fig. 1. Fig. 1 shows clearly how the application multicast data packets are encapsulated within the frames. Each block of data, consisting of 217 bytes, is given a sequence number ranging from 1 to 65536. The sequence number field in a corresponding word is preceded by a type field. So, far, we have only two designations, where 11111111 and 10101010 denote a fresh and a retransmitted word respectively. A 1-byte data length (DL) field is also needed to indicate the end of receiving because the data may be less than 217 bytes. If the data are less than 217 bytes, the remainder of the 217 bytes is filled with padding. A 2-byte cyclic redundancy check (CRC) field is also formed at the end of the word. This CRC operates over the whole word. The resulting 223 bytes are fed to the (255, 223) Reed-Solomon encoder routine [1], which outputs a corresponding 255 bytes that we call

an encoded word. The RS code is assumed to be systematic, such that the 32 parity bytes are added by the encoder at the end of the word. One byte of dummy data is added to obtain a 256-byte encoded word.

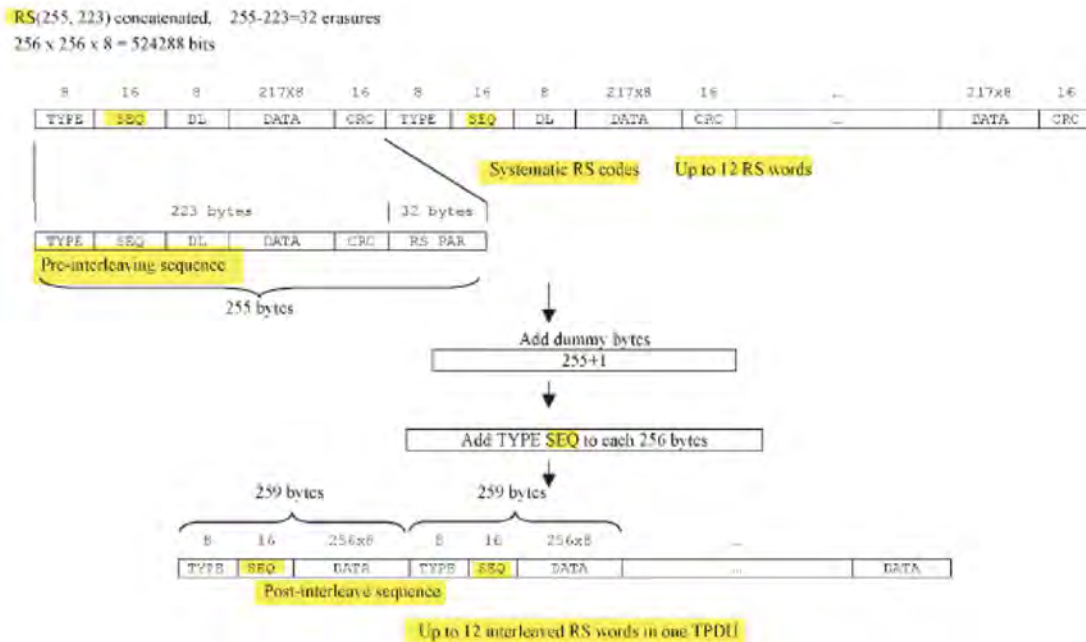
Chen at NOK00075559, col. 2 ¶ 4-NOK00075560, col. 1 ¶ 1.

294. The data packets are comprised of Reed-Solomon codewords (abbreviated “RSvwords” in Chen) that are interleaved and a new sequence number is inserted after interleaving.

Interleaving is next applied to a record of 256 such encoded RS words to yield a corresponding interleaved record. The first interleaved word of such a record consists of the first bytes of each of the 256 preinterleaving words. The 256-th interleaved word is the concatenation of the 256-th bytes of all the pre-interleaving words. It is also possible to have a smaller interleaving depth such as 128, in which case the first interleaved word will encompass the first and second bytes of every pre-interleaving word. Then the first 128 bytes of the 128-th interleaved word are bytes numbered 255 in the original pre-interleaving words, and the second 128 bytes of the 128-th interleaved word are bytes numbered 256 in the original pre-interleaving words.

If the interleaving depth is 256, it is easy to understand that up to 32 consecutive interleaved words can be lost without any ARQ retransmission request from the receiver. Because of interleaving, if any 32 words are lost, only 32 of the 255 bytes of each corresponding preinterleaving word will be lost. Because the RS decoder can conceal up to $255 - 223 = 32$ lost bytes, such loss is not detrimental and will not result in retransmission requests. For error concealment for multicast traffic over the Internet, minimizing the need for retransmissions is an essential asset, which subsequently leads to the minimizing of no acknowledgement (NAK) and repair-packet implosions [2]. Returning to Fig. 1, we see that the server routine groups up to 12 interleaved words into one UDP data unit (TPDU). A 2-byte interleaved sequence number field is added to each interleaved word, preceded by a 1-byte field denoting an interleaved word, i.e., 11111111. Although each original RS-encoded word carries a type field and a sequence field, these fields are scrambled later by the interleaving, and the need arises for new type and sequence fields. Occasionally or at the request of the client, retransmitted words are also included in some TPDU. If an interleaving depth of 256 is used, each TPDU contains eight interleaved words; in this case four consecutive TPDU can be lost without any retransmissions.

Chen at NOK00075560, col. 1 ¶¶ 2-3. *See also id.* Fig. 1 (excerpted for readability).



295. The receiver deinterleaves the received interleaved packets.

... If the type field indicates an interleaved word, this interleaved word is sent to the de-interleaving buffer. A timer is reset upon the arrival of the first byte of a certain deinterleaved RS word (this may not be the first byte of the word).

Routinely, the de-interleaved RS words are checked one after another. ...

Chen at NOK00075560, col. 2 ¶¶ 2-3 (excerpted).

296. If a message is received with too many corrupted packets, the system sends a NAK to request retransmission of the errored packets only; the NAK identifies the errored packets by their sequence number.

The format of the NAK frame used by the processing in Fig. 2(b) is shown in Fig. 3. As mentioned before, for the multicast applications at hand, this is the only packet type that the client can send. However, for a video conferencing system, data may also be sent from the client to the server. In Fig. 3, we use 01010101 as the value of the type field to denote NAK messages. A 1-byte field denoting the data length of the NAK message is also added. The subsequent fields give the sequence numbers of the lost words as requested by the client. A final 1-byte CRC field is used for error checking at the server. The client routine in Fig. 4(a) shows the process that takes place once a record of data is read from the UDP layer. Blocks of 259 bytes are sampled from such a record. If the type field indicates a repair word, and if further CRC indicates correct reception, the first bytes are stripped and the following 217 bytes are sent to the receiving buffer. The DL field may be used to separate real data

from padding. If the type field indicates an interleaved word, this interleaved word is sent to the de-interleaving buffer. A timer is reset upon the arrival of the first byte of a certain deinterleaved RS word (this may not be the first byte of the word).

Chen at NOK00075560, col. 2 ¶ 2.

Fig. 4(b) shows the step-by-step generation of the NAK request at the client. As shown in Fig. 2(b), we execute the processing for such a NAK message as it arrives at the server. Fig. 4(b) also shows the multiplexing of RS (interleaved) words and repair (retransmission) words in server transmission time. Priority is given to repair words, and interleaved words are transmitted when there are no repairs to be sent ($XZ = 0$).

The timing diagram of Fig. 5 shows a sequence of events taking place under the control of our new scheme for a typical scenario. Equal-length interleaving windows for data records N , $N + 1$, $N + 2$ are used. The end of the next window sets the time limit for playing received packets of the current record at the MPEG player. Since received packets of each record are reorganized in the interleaving window and then played in sequence, this technique has a buffer effect that can help to guarantee delay-jitter-free playing at the receiver. This figure also shows the NAK and retransmissions for record N due to the packet loss. For record $N + 1$, the RS FEC decoder can correct the loss and errors without asking for any retransmission. For record $N + 2$, no loss and errors are encountered for the first K bytes of RS symbols; therefore, the RS decoder is not called, and no NAKs are generated.

Chen at NOK00075561, col. 2 ¶¶ 1-2. *See also id.* Figs. 4b, 5 (5 excerpted).

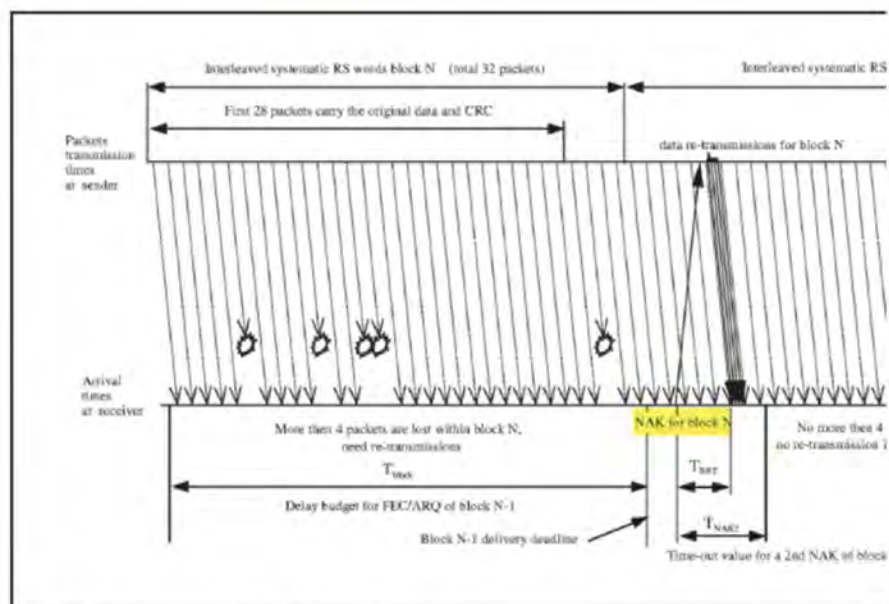
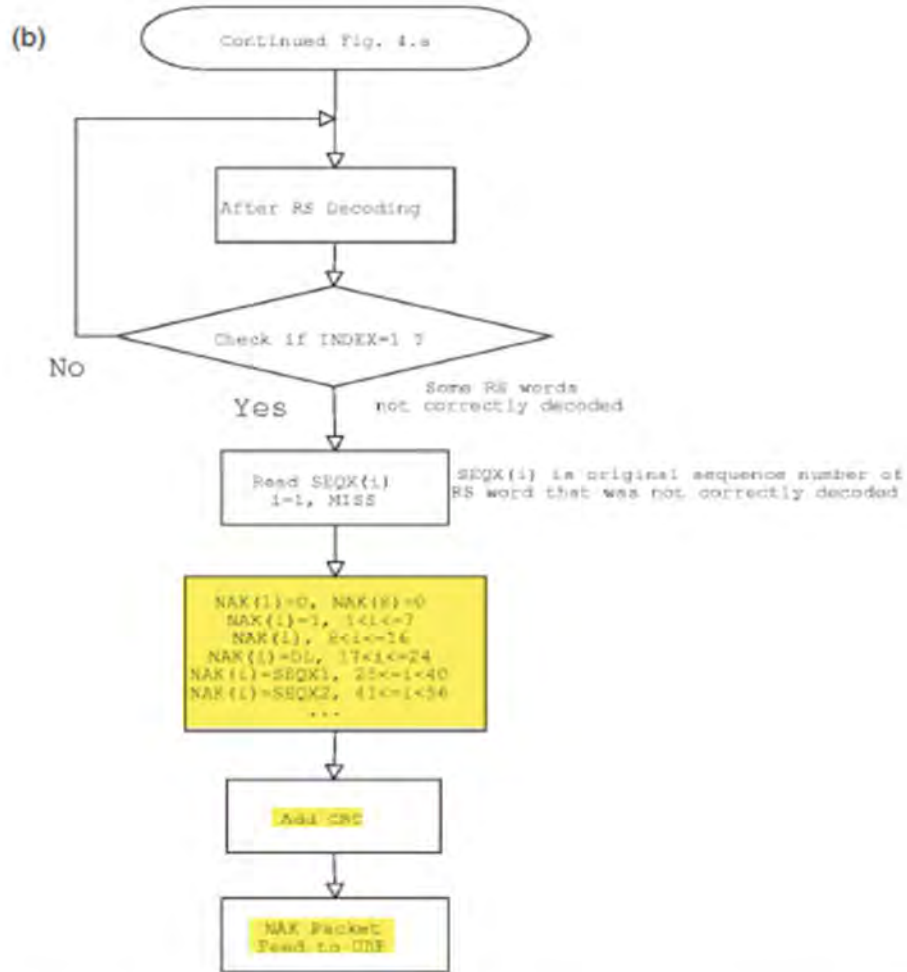


Figure 5: The hybrid FEC/ARQ system timing.

297. Figure 3 shows the NAK format, including the sequence number of incorrectly received packets.

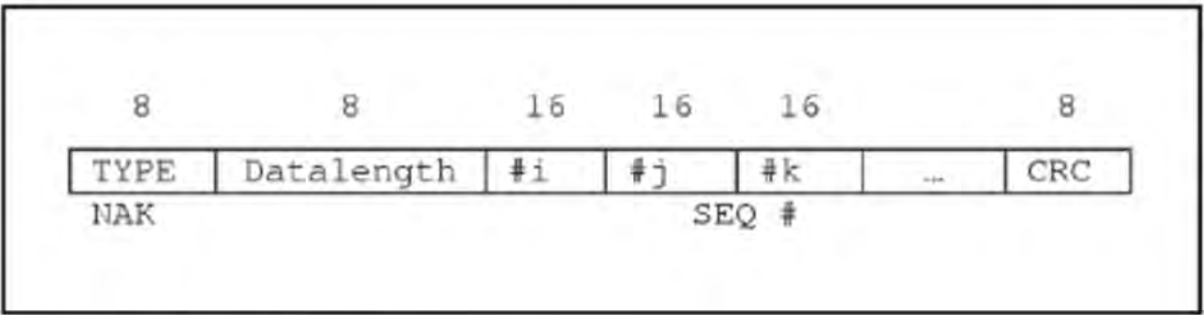


Figure 3: NAK packet format.

298. Chen performed experiments to validate these concepts. Chen at NOK00075561-562, page 3, col. 2 ¶ 3-page 4, col. 2 ¶ 1. In those experiments, the retransmission and interleaving functions were performed by general purpose CPUs that would, of course, used shared memory and therefore would have allocated shared memory between an interleaving function and a retransmission function. *Id.* Chen explicitly discusses how interleaving, FEC, and ARQ are performed in his experiments and expected to be performed in real systems by the server and client CPUs, as shown in the excerpt below:

mental results above show that the performance of our algorithms is sensitive mainly to packet loss and errors. Bandwidth expansion due to FEC/ARQ utilization may bring occasional congestion, packet loss and errors. However, test results indicate that the performance of our algorithms is not compromised as long as the CPU speeds of the server and clients are capable of handling the real-time interleaved FEC/ARQ coding/decoding algorithms therein.

D. 3GPP TS 25.401 v6.0.0 (“TS 25.401”) (NOK00075136)

299. The 3GPP published TS 25.401 v6.0.0 (“TS 25.401”) on their publicly available website March 26 of 2003. TS 25.401 at NOK00075136, cover page. The meta data associated with the public document as retrievable from the ETSI webpage indicates that it was last modified on March 15, 2003. Accordingly, I understand that TS 25.401 qualifies as §102(b) prior art. TS 25.401

provides a basic description of the overall Release 6 Universal Terrestrial Radio Access Network (“UTRAN”). *Id.*

300. As can be seen from the excerpt below, TS 25.401 describes the basic architecture of the UTRAN, which includes several communications transceivers, including the so-called “Node B” (the cellular equivalent of the CO side in a DSL connection), the Radio Network Subsystem (RNS), which controls the Node B, and user equipment (UE), which is a user device like a smartphone, which is the cellular equivalent of the CPE side of a DSL connection.

Node B: logical node in the RNS responsible for radio transmission / reception in one or more cells to/from the UE. [¶] The logical node terminates the Iub interface towards the RNC.

Radio Network Subsystem: RNS can be either a full UTRAN or only a part of a UTRAN [¶] An RNS offers the allocation and release of specific radio resources to establish means of connection in between an UE and the UTRAN. A Radio Network Subsystem contains one RNC and is responsible for the resources and transmission/reception in a set of cells.

User Equipment: Mobile Equipment with one or several UMTS Subscriber Identity Module(s) [¶] A device allowing a user access to network services via the Uu interface. The UE is defined in ref. [8]. If this term is used in the context of Iur-g, it means MS in case it uses radio resources of a DBSS.

TS 25.401 at NOK00075142-143, § 3.1 “Definitions.” *See also:*

RNS Radio Network Subsystem

UE User Equipment

301. TS 25.401 at NOK00075144, § 3.2 “Abbreviations. The RNS may control multiple Node Bs, as shown in Figure 4 from TS 25.401 below.

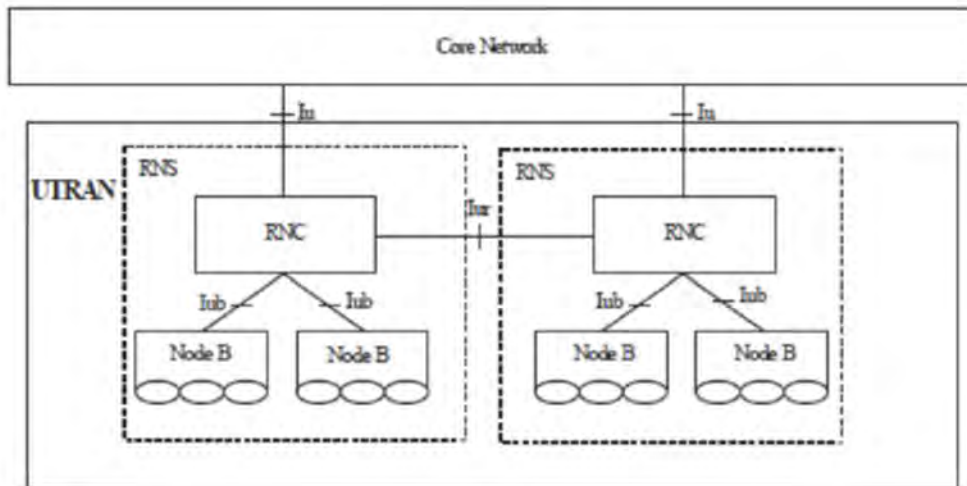


Figure 4: UTRAN Architecture

TS 25.401 at NOK00075148, Figure 4.

302. Additional details about the 3GPP Release 6 system can be found in other documents published by the 3GPP. For example, 3GPP TS 25.212 v6.0.0, published on the 3GPP's publicly available website in December 2003, describes how the Release 6 channels are coded. TS 25.212 at cover page; *see also* TR 25.896 at NOK00074965, § 8.4.1 "Physical layer structures in time domain (TS25.212)." Channels, including the DCH (which, as described below with respect to TR 25.896, is the predecessor to the E-DCH described in TR 25.896), are coded as described in Figure 1 in TS 25.212. *Id.* § 4.2 "General coding/multiplexing of TrCHs". As can be seen in Figure 1, the DCH is interleaved twice:

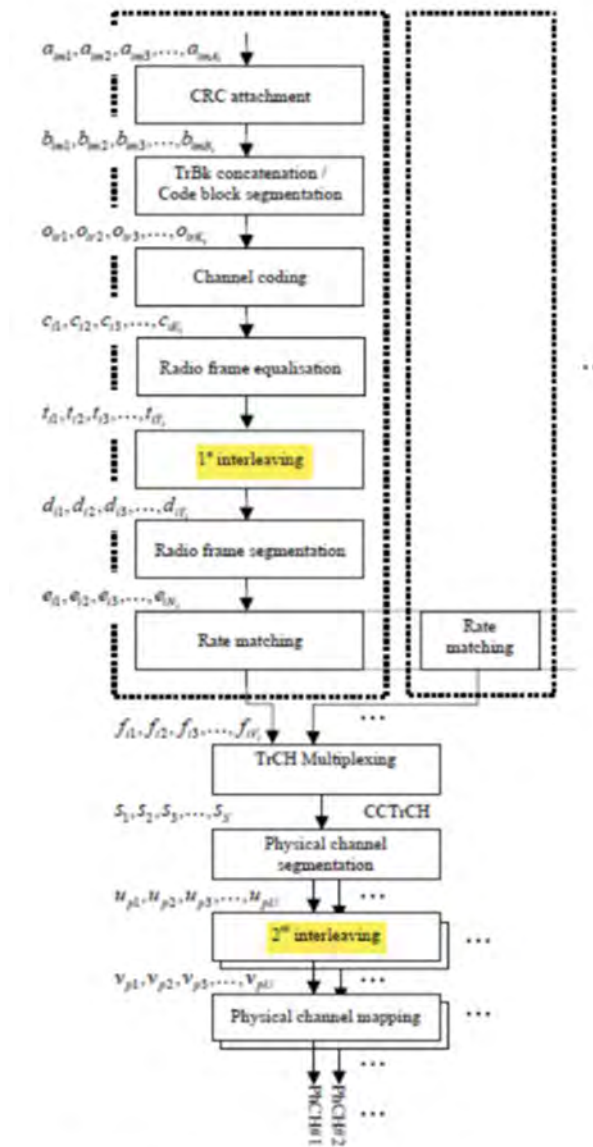


Figure 1: Transport channel multiplexing structure for uplink.

TS 25.212 at Figure 1.

303. The first interleaving in Release 6 is done with a block interleaver using inter-column permutations. The first interleaving is a five-step process, which is applied in both the uplink and downlink directions.

4.2.5.2 1st interleaver operation

The 1st interleaving is a block interleaver with inter-column permutations. The input bit sequence to the block interleaver is denoted by $x_{i,1}, x_{i,2}, x_{i,3}, \dots, x_{i,X_i}$, where i is TrCH number and X_i the number of bits. Here X_i is guaranteed to be an integer multiple of the number of radio frames in the TTI. The output bit sequence from the block interleaver is derived as follows:

- (1) Select the number of columns $C1$ from table 4 depending on the TTI. The columns are numbered 0, 1, ..., $C1 - 1$ from left to right.
- (2) Determine the number of rows of the matrix, $R1$ defined as

$$R1 = X_i / C1.$$

The rows of the matrix are numbered 0, 1, ..., $R1 - 1$ from top to bottom.

- (3) Write the input bit sequence into the $R1 \times C1$ matrix row by row starting with bit $x_{i,1}$ in column 0 of row 0 and ending with bit $x_{i,(R1 \times C1)}$ in column $C1 - 1$ of row $R1 - 1$:

$$\begin{bmatrix} x_{i,1} & x_{i,2} & x_{i,3} & \dots & x_{i,C1} \\ x_{i,(C1+1)} & x_{i,(C1+2)} & x_{i,(C1+3)} & \dots & x_{i,(2 \times C1)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ x_{i,((R1-1) \times C1+1)} & x_{i,((R1-1) \times C1+2)} & x_{i,((R1-1) \times C1+3)} & \dots & x_{i,(R1 \times C1)} \end{bmatrix}$$

- (4) Perform the inter-column permutation for the matrix based on the pattern $\langle P1_{C1}(j) \rangle_{j \in \{0,1,\dots,C1-1\}}$ shown in table 4, where $P1_{C1}(j)$ is the original column position of the j -th permuted column. After permutation of the columns, the bits are denoted by y_{ik} :

$$\begin{bmatrix} y_{i,1} & y_{i,(R1+1)} & y_{i,(2 \times R1+1)} & \dots & y_{i,((C1-1) \times R1+1)} \\ y_{i,2} & y_{i,(R1+2)} & y_{i,(2 \times R1+2)} & \dots & y_{i,((C1-1) \times R1+2)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ y_{i,R1} & y_{i,(2 \times R1)} & y_{i,(3 \times R1)} & \dots & y_{i,(C1 \times R1)} \end{bmatrix}$$

- (5) Read the output bit sequence $y_{i,1}, y_{i,2}, y_{i,3}, \dots, y_{i,(C1 \times R1)}$ of the block interleaver column by column from the inter-column permuted $R1 \times C1$ matrix. Bit $y_{i,1}$ corresponds to row 0 of column 0 and bit $y_{i,(R1 \times C1)}$ corresponds to row $R1 - 1$ of column $C1 - 1$.

Table 4 Inter-column permutation patterns for 1st interleaving

TTI	Number of columns $C1$	Inter-column permutation patterns $\langle P1_{C1}(0), P1_{C1}(1), \dots, P1_{C1}(C1-1) \rangle$
10 ms	1	$\langle 0 \rangle$
20 ms	2	$\langle 0, 1 \rangle$
40 ms	4	$\langle 0, 2, 1, 3 \rangle$
80 ms	8	$\langle 0, 4, 2, 6, 1, 5, 3, 7 \rangle$

TS 25.212 at § 4.2.5.2 “1st interleaver operation”; *See also id.* §§ 4.2.5.3-4 (“Relation between input and output of 1st interleaving in [uplink/downlink]”).

304. The second interleaving in Release 6 is also a block interleaving, but operates differently than the first interleaving. TS 25.212 at § 4.2.11 “2nd interleaving”. The second interleaving, like the first interleaving, is a five-step process applied in both the uplink and downlink directions.

4.2.11 2nd interleaving

The 2nd interleaving is a block interleaver and consists of bits input to a matrix with padding, the inter-column permutation for the matrix and bits output from the matrix with pruning. The bits input to the block interleaver are denoted by $u_{p,1}, u_{p,2}, u_{p,3}, \dots, u_{p,U}$, where p is PhCH number and U is the number of bits in one radio frame for one PhCH. The output bit sequence from the block interleaver is derived as follows:

- (1) Assign $C2 = 30$ to be the number of columns of the matrix. The columns of the matrix are numbered $0, 1, 2, \dots, C2 - 1$ from left to right.
- (2) Determine the number of rows of the matrix, $R2$, by finding minimum integer $R2$ such that:

$$U \leq R2 \times C2.$$

The rows of rectangular matrix are numbered $0, 1, 2, \dots, R2 - 1$ from top to bottom.

- (3) Write the input bit sequence $u_{p,1}, u_{p,2}, u_{p,3}, \dots, u_{p,U}$ into the $R2 \times C2$ matrix row by row starting with bit $y_{p,1}$ in column 0 of row 0:

$$\begin{bmatrix} y_{p,1} & y_{p,2} & y_{p,3} & \dots & y_{p,C2} \\ y_{p,(C2+1)} & y_{p,(C2+2)} & y_{p,(C2+3)} & \dots & y_{p,(2 \times C2)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ y_{p,((R2-1) \times C2+1)} & y_{p,((R2-1) \times C2+2)} & y_{p,((R2-1) \times C2+3)} & \dots & y_{p,(R2 \times C2)} \end{bmatrix}$$

where $y_{p,k} = u_{p,k}$ for $k = 1, 2, \dots, U$ and if $R2 \times C2 > U$, the dummy bits are padded such that $y_{p,k} = 0$ or 1 for $k = U + 1, U + 2, \dots, R2 \times C2$. These dummy bits are pruned away from the output of the matrix after the inter-column permutation.

- (4) Perform the inter-column permutation for the matrix based on the pattern $\langle P2(j) \rangle_{j \in \{0,1,\dots,C2-1\}}$ that is shown in table 7, where $P2(j)$ is the original column position of the j -th permuted column. After permutation of the columns, the bits are denoted by $y'_{p,k}$.

$$\begin{bmatrix} y'_{p,1} & y'_{p,(R2+1)} & y'_{p,(2 \times R2+1)} & \dots & y'_{p,((C2-1) \times R2+1)} \\ y'_{p,2} & y'_{p,(R2+2)} & y'_{p,(2 \times R2+2)} & \dots & y'_{p,((C2-1) \times R2+2)} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ y'_{p,R2} & y'_{p,(2 \times R2)} & y'_{p,(3 \times R2)} & \dots & y'_{p,(C2 \times R2)} \end{bmatrix}$$

- (5) The output of the block interleaver is the bit sequence read out column by column from the inter-column permuted $R2 \times C2$ matrix. The output is pruned by deleting dummy bits that were padded to the input of the matrix before the inter-column permutation, i.e. bits $y'_{p,k}$ that corresponds to bits $y_{p,k}$ with $k > U$ are removed from the output. The bits after 2nd interleaving are denoted by $v_{p,1}, v_{p,2}, \dots, v_{p,U}$, where $v_{p,1}$ corresponds to the bit $y'_{p,k}$ with smallest index k after pruning, $v_{p,2}$ to the bit $y'_{p,k}$ with second smallest index k after pruning, and so on.

Table 7 Inter-column permutation pattern for 2nd interleaving

Number of columns C2	Inter-column permutation pattern $\langle P2(0), P2(1), \dots, P2(C2-1) \rangle$
30	$\langle 0, 20, 10, 5, 15, 25, 3, 13, 23, 8, 18, 28, 1, 11, 21, 6, 16, 26, 4, 14, 24, 19, 9, 29, 12, 2, 7, 22, 27, 17 \rangle$

TS 25.212 at § 4.2.11.

E. 3GPP TR 25.896 v6.0.0 (“TR 25.896”) (NOK00074926)

305. The 3GPP published TR 25.896 v6.0.0 (“TR 25.896”) on their publicly available website on April 3 of 2004. TR 25.896 at NOK00074926, cover page. The meta data associated with the public document as retrievable from the ETSI webpage indicates that it was last modified on April 3, 2004. Accordingly, I understand that TR 25.896 qualifies as §102(b) prior art. TR 25.896 describes studies performed of enhanced uplink features for Release 6 of the 3GPP UTRAN (which, in its non-enhanced form, is generally described in TS 25.401). *Id.*; see also *id.* § 1 “Scope” (“The purpose of this TR is to help TSG RAN WG1 to define and describe the potential enhancements

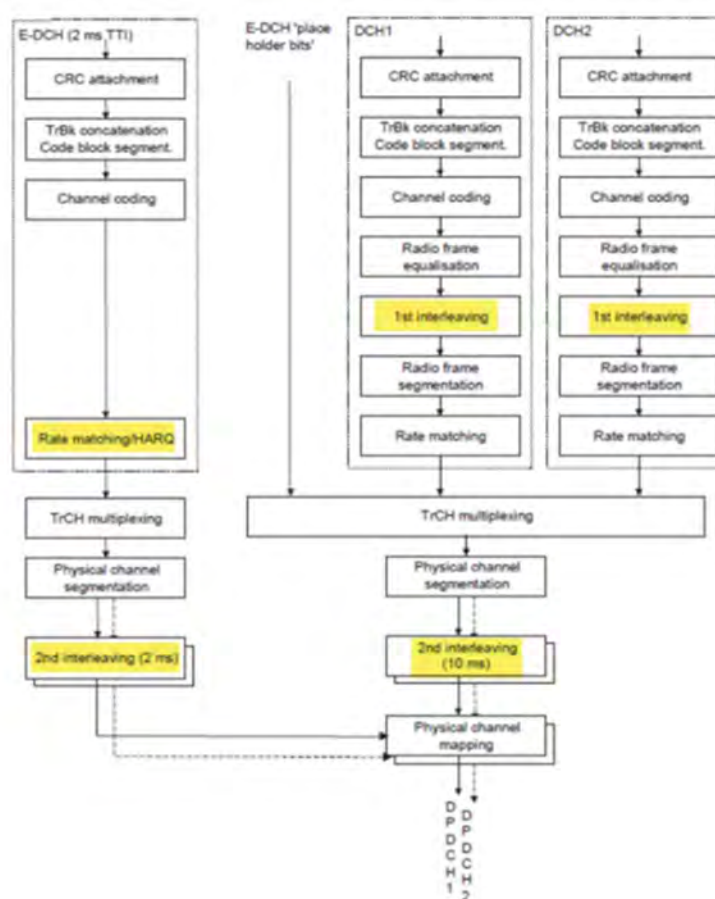
under consideration and compare the benefits of each enhancement with earlier releases for improving the performance of the dedicated transport channels in UTRA FDD uplink, along with the complexity evaluation of each technique. The scope is to either enhance uplink performance in general or to enhance the uplink performance for background, interactive and streaming based traffic.”).

306. One of the major enhancements proposed to the Release 6 uplink was the addition of hybrid ARQ protocols, which were already in use on the downlink side. TR 25.896 at NOK00074952, § 7.2.1 “Hybrid ARQ” (“Hybrid ARQ cannot be used by all [uplink] transport channels ... In the downlink, there is a separate CCTrCh carrying the HS-DSCH, while the assumption of a separate CCTrCh is not necessarily true in the uplink scenario.”). Note that CCTrCh stands for Coded Composite Transport Channel.

307. Hybrid ARQ is a type of retransmission, as described in § VII.A.4 above, which had been in use by communication systems for decades before TR 25.896 was published. TR 25.896, however, specifically studied the feasibility of adding hybrid ARQ, both with and without incremental redundancy, to the resource-limited Release 6 uplink. TR 25.896 at NOK00074951, § 7.2.1 “Hybrid ARQ” (“Node B controlled hybrid ARQ allows for rapid retransmissions of erroneously received data units, thus reducing the number of RLC retransmissions and the associated delays. This can improve the quality of service experienced by the end user. As a Node B controlled retransmission is less costly from a delay perspective, the physical channel can be operated with somewhat higher error probability than in Rel 5, which may result in improved system capacity.”).

308. TR 25.896 supports hybrid ARQ using a channel called the “E-DCH” or enhanced dedicated channel, which (other than the changed described in TR 25.896) is relatively unchanged

from its description in TS 25.401 and TS 25.212, including the use of first and second interleaving steps. TR 25.896 at NOK00074935, § 3 “Definitions, symbols and abbreviations” (“E-DCH – Enhanced DCH, a new dedicated transport channel type or enhancements to an existing dedicated transport channel type (if required by a particular proposal)”; *see also id.* § 8.1.1 “Transport Channel Structure” (“To support some of the enhancements currently under consideration, a new transport channel type, the E-DCH, is introduced. Depending on future decisions on which enhancements to support and how to support them, the E-DCH may or may not be identical to the DCH.”). For example, Figure 8.4.7 from TR 25.896 shows how a 2ms E-DCH could interact with the existing DCH structure, including the second interleaving step:



TR 25.896 at NOK00074969, Figure 8.4.7. Therefore, TR 25.896 describes using hybrid ARQ (retransmission) and interleaving to detect and correct errored packets in the uplink direction. *Id.* It

is worth noting that, by the time TR 25.896 was published, the Release 6 system already included the use of hybrid ARQ and interleaving to detect and correct errored packets in the downlink direction. *See, e.g.*, TS 25.212 at § 4.5 “Coding for HS-DSCH” and Figure 16 (copied below).

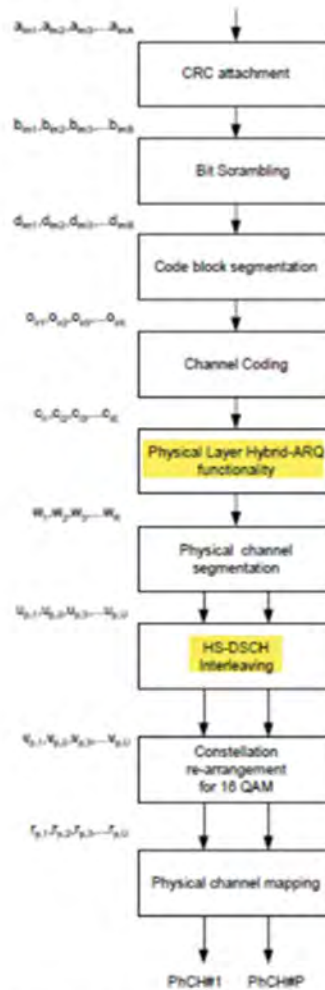


Figure 16: Coding chain for HS-DSCH

309. The uplink hybrid ARQ described in TR 25.896 uses a defined amount of memory in both the Node B and UE. The memory used at the Node B can be calculated based on a number of variables including the number of times a packet can be retransmitted (“N”).

The principle of hybrid ARQ is to buffer E-DCH TTIs that were not received correctly and consequently combine the buffered data with retransmissions. The actual method of doing soft combining depends on the HARQ combining scheme selected. In Chase combining scheme the receiver always combines the full

retransmission of the failed TTI, i.e. the amount of data in the receiver buffer remains the same. In the incremental redundancy schemes the receiver buffers coded symbols, which introduce new information to the E-DCH TTI transmitted first, i.e. the amount of data to be buffered increases with consecutive retransmissions. However, in practice the buffer in the receiver needs to be dimensioned considering the maximum number of coded bits of the E-DCH TTI after all the incremental redundancy has been introduced. Regardless of the HARQ combining scheme soft combining is done on L1 of Node B before the decoding stage of FEC. Prior to decoding these symbols are soft-valued, i.e. each symbol is represented by two or more bits. Here we call them soft symbols.

For N-process SAW HARQ, the number of soft symbols to be buffered in L1 receiver can be estimated generally as follows, since no new PDUs are transmitted on a subchannel before the previous packet is acknowledged. The receiver has to buffer one E-DCH TTI for each HARQ process and for each UE. The next transmission is either a new packet or a retransmission of an erroneous packet. In either case, the maximum buffering need is N E-DCH TTIs. The actual size of the buffer needed for each E-DCH TTI depends on the HARQ combining scheme as described above. Thus, for N-process stop-and-wait ARQ the L1 buffering at Node B for one UE can be expressed as:

$$buffer = (softsymbols_{TTI} \times N)$$

Table 9.2.4 shows some examples for the soft buffer size required at Node B per UE. It should be noted that soft buffer is needed for all UEs waiting for retransmission at a given time. Thus these values have to be multiplied by the number of active UEs using E-DCH. Depending on particular implementations, the buffer may be shared between UEs and thus the total size of soft buffer can be reduced given that not all active UEs transmit simultaneously using the most demanding format. Also, it may be possible to re-use some existing buffers.

Table 9.2.4 Node B soft buffer size (kilo soft symbols) per UE, for some example values of N (number of HARQ processes) and TTI lengths

	N=4, TTI=10 ms	N=3, TTI=10 ms	N=8, TTI=2 ms	N=6, TTI=2 ms
BPSK, SF=4	38.4	28.8	15.36	11.52
QPSK or 2*BPSK, SF=4	76.8	57.6	30.72	23.04
8PSK or 3*BPSK, SF=4	115.2	86.4	46.08	34.56
2*QPSK or 4*BPSK, SF=4; or QPSK, SF=2	153.6	115.2	61.44	46.08

TR 25.896 at NOK00074991, § 9.2.2.1.1 “Soft buffer at Node B”. Hybrid ARQ can also require reordering received packets, which (again) requires a set amount of memory.

Due to physical/MAC layer (H)ARQ, the MAC-e PDUs may be received in wrong order. Therefore, reordering of the MAC-e PDUs is required either in the Node B or in the RNC. Here the required reordering buffer size per UE is estimated. In the estimation it is assumed that only one high bit rate reordering queue is active for a

UE at a time. Furthermore, maximum of 3 retransmissions per MAC-e PDU is assumed and the buffer size is calculated for the worst case where one ARQ process requires retransmissions while the others get blocks through and forward them to reordering queue. The worst case buffer size is then calculated as

$$reorderingbuffer = [3(N - 1) + 1] \times (bits/T_{TI})$$

These values have to be multiplied by the number of E-DCH UEs served by the Node B or by the RNC, depending on the location of the reordering buffer. Depending on implementation and location of the re-ordering buffer, it may be possible to re-use some existing buffers.

TR 25.896 at NOK00074991-992, § 9.2.2.1.2 “Reordering buffer in radio network” and Table 9.2.5.

310. A reordering buffer is also necessary to deinterleave the interleaved data and ensure that all packets and data within packets are in the correct order. As with the reordering buffer, the size of the deinterleaving buffer will depend on the amount of data transmitted in a packet and the interleaving depth (which is similar to N, the number of times a packet can be retransmitted).

311. Similarly, the amount of memory used by the UE also depends on “N” (the number of times a packet can be retransmitted).

In addition to buffers in the receiver side, also some buffer at UE side is needed. The retransmission buffer can be either at MAC layer as info bits or at physical layer as coded bits. The total number of coded bits is typically 3 times the number of info bits. Table 9.2.4 shows also the required retransmission buffer size in kbits (assuming that coded bits are buffered). Depending on implementation it may be possible to re-use existing buffers.

TR 25.896 at NOK00074992, § 9.2.2.1.3 “Retransmission buffer in UE.”

312. TR 25.896 describes the signaling between the UE and Node B necessary to support uplink hybrid ARQ. For example, if the Node B decides to allow only a certain number of retransmissions of a packet (e.g., setting N to 3 to limit packets to three retransmissions), both the Node B and UE need to know N. Similarly, if the Node B wants to use four redundancy versions (which are numbered from 0, resulting in redundancy versions 0, 1, 2, and 3), the UE needs to know

this as well. Likewise, of course, the UE must tell the Node B which redundancy version is used with any particular packet transmitted.

Associated control signaling required for the operation a particular scheme consists of downlink and uplink signaling. Different proposals may have different requirements on the necessary signaling. Furthermore, the signaling structure may depend on other uplink enhancements considered.

The overhead required should be kept small in order not to waste power and code resources in the downlink and not to create unnecessary interference in the uplink.

...

The necessary information needed by the Node B to operate the hybrid ARQ mechanism can be grouped into two different categories: information required prior to soft combining/decoding (outband signaling), and information required after successful decoding (inband signaling). Depending on the scheme considered, parts of the information might either be explicitly signaled or implicitly deduced, e.g., from CFN or SFN.

The information required prior to soft combining consists of:

- Hybrid ARQ process number.
- New data indicator. The new data indicator is used to control when the soft combining buffer should be cleared in the same way as for the HS-DSCH.
- Redundancy version. If multiple redundancy versions are supported, the redundancy version needs to be known to the Node B. The potential gains with explicit support of multiple redundancy versions should be carefully weighted against the increase in overhead due to the required signaling. Note that, unlike the HS-DSCH, the number of users simultaneously transmitting data in the uplink using hybrid ARQ may be significant.
- Rate matching parameters (number of physical channel bits, transport block size). This information is required for successful decoding. In R99/4/5, there is a one-to-one mapping between the number of physical channel bits and the transport block size, given by the TFCI and attributes set by higher layer signaling. This assumption does not hold for hybrid ARQ schemes if the number of available channel bits varies between (re)transmissions, e.g., due to multiplexing with other transport channels. Hence, individual knowledge of these two quantities is required in the Node B.

The information required after successful decoding can be sent as a MAC header. The content is similar to the MAC-hs header, e.g., information for reordering, demultiplexing of MAC-d PDUs, etc.

The information needed by UE necessary to operate the hybrid ARQ mechanism is either explicitly signaled by Node B, or decided by the UE itself, depending on the scheme. It is noted that whether the UE will decide the parameter values or the Node B will signal them, could affect the round trip time for HARQ retransmissions.

TR 25.896 at NOK00074953, § 7.2.3 “Associated Signaling”.

313. TR 25.896 therefore describes, in detail, the amount of memory needed to support hybrid ARQ retransmission on both ends of a cellular system. As noted in the description of TS 25.401, the 3GPP has defined, in detail, two stages of interleaving applied to packets that are also protected from error using hybrid ARQ retransmission.

314. Well before the publication of TS 25.401 in 2003 and TR 25.896 in 2004, a POSITA would know that UMTS communications systems could be implemented using multicarrier communication. For example, the 2001 conference paper by Lacroix, Goudard, and Alard entitled “OFDM with guard interval vs. OFDM/Offset QAM for high data rate UMTS downlink transmission,” presented at VTC 2001 explicitly discussed using multicarrier transmission for UMTS. Below is an excerpt which includes the sentence “This study confirms the interest of considering OFDM modulation for high data rate UMTS downlink transmission.”

Abstract – OFDM modulation using guard interval, denoted OFDM/QAM in this paper, is well known for its robustness to multi-path time varying propagation channels, and therefore used in several broadcasting and high data rate radio LAN standards. OFDM/OQAM modulation is an alternative to it, which has the advantage not to require the use of a guard interval, which leads to a gain in spectral efficiency. In this paper, we recall the main characteristics of OFDM/OQAM, and compare it to the OFDM/QAM in a realistic 3G context. We propose optimized modulation parameters and provide simulation results. The study confirms the interest of considering OFDM modulation for high data rate UMTS downlink transmission. Besides, in that context, the gain in spectral efficiency allows OFDM/OQAM to outperform OFDM/QAM performances.

F. U.S. Patent No. 6,865,233 issued to Eriksson, et al. (“Eriksson”) (NOK00083199)

315. The application that matured into U.S. Patent No. 6,865,233 (“Eriksson”) was filed on February 19, 1999 and issued as a patent on March 8, 2005. Accordingly, I understand that Eriksson qualifies as §102(b) prior art. Eriksson describes and claims methods and systems for “control signaling enabling flexible link adaptation” in a communication system such as a cellular system. Eriksson at NOK00083199, title.

316. Eriksson describes multiple ways of detecting and correcting errored packets in a communication system, including the use of FEC such as interleaving and several types of hybrid ARQ retransmission.

Another proposed hybrid ARQ Scheme, sometimes referred to as incremental redundancy or type-II hybrid ARQ, provides for additional redundant bits to be transmitted if the originally transmitted block cannot be decoded. This scheme is conceptually illustrated in FIG. 3. Therein, three decoding attempts are made by the receiver. First, the receiver attempts to decode the originally received data block (with or without redundancy). Upon failure, the receiver then receives additional redundant bits R1, which it uses in conjunction with the originally transmitted data block to attempt decoding. As a third step, the receiver obtains another block of redundant information R2, which it uses in conjunction with the originally received data block and the block of redundant bits R1 to attempt decoding for a third time. This process can be repeated until successful decoding is achieved.

As compared with link adaptation, incremental redundancy does not require that link quality estimates be transmitted or used. However, one problem with this technique is the large memory requirement associated with storing the data block (and possibly additional blocks of redundant bits) until a successful decode occurs, which storage is needed since the subsequently transmitted redundancy blocks (e.g., R1 and R2) cannot be independently decoded to give the same performance as if combined decoding was used. The storage requirements are further increased if the receiver stores a multi-bit soft value associated with each received bit, the soft values indicating a confidence level associated with the decoding of the received bit.

Eriksson at NOK00083211, 3:23-52.

317. Eriksson describes why ARQ and FEC have competing advantages and disadvantages:

In addition to modulation schemes, digital communication systems also employ various techniques to handle erroneously received information, which techniques also affect the bit rate experienced by the user. Generally speaking, these techniques include those which aid a receiver to correct the erroneously received information, e.g., forward error correction (FEC) techniques, and those which enable the erroneously received information to be retransmitted to the receiver, e.g., automatic retransmission request (ARQ) techniques. FEC techniques include, for example, convolutional or block coding of the data prior to modulation. FEC coding involves representing a certain number of data bits using a certain number of code bits. Thus, it is common to refer to convolutional codes by their code rates, e.g., 1/2 and 1/3, wherein the lower code rates provide greater error protection but lower user bit rates for a given channel bit rate.

ARQ techniques involve analyzing received blocks of data for errors and requesting retransmission of blocks which contain errors. Consider, for example, the block mapping example illustrated in FIG. 2 for a radiocommunication system operating in accordance with the Generalized Packet Radio Service (GPRS) optimization which has been proposed as a packet data service for GSM. Therein, a logical link control (LLC) frame containing a frame header (FH), a payload of information and a frame check sequence (FCS) is mapped into a plurality of radio link control (RLC) blocks, each of which include a block header (BH), information field, and block check sequence (BCS), which can be used by a receiver to check for errors in the information field. The RLC blocks are further mapped into physical layer bursts, i.e., the radio signals which have been GMSK modulated onto the carrier wave for transmission. In this example, the information contained in each RLC block can be interleaved over four bursts (timeslots) for transmission.

Eriksson at NOK00083210, 2:15-49.

318. For example, interleaving and hybrid ARQ retransmission both require large amounts of memory. *See, e.g., §§ VII.A.3-4.*

319. Eriksson describes the advantages of using these techniques together in hybrid ARQ (which combines ARQ and FEC):

Strengths and weaknesses of these two error control schemes can be balanced by combining FEC and ARQ techniques. Such combined techniques, commonly referred to as hybrid ARQ techniques, permits correction of some received errors using the FEC coding at the receiver, with other errors requiring retransmission. Proper selection of FEC coding schemes with ARQ protocols thus results in a hybrid ARQ technique having greater reliability than a system employing a purely FEC coding scheme with greater throughput than a system employing a purely ARQ-type error handling mechanism.

Eriksson at NOK00083210-211, 2:65-3:8.

320. Eriksson acknowledges the problem of limited memory, which is especially acute in the uplink (UE to Node B) direction since UEs are meant to be easily movable by a user.

As compared with link adaptation, incremental redundancy does not require that link quality estimates be transmitted or used. However, one problem with this technique is the large memory requirement associated with storing the data block (and possibly additional blocks of redundant bits) until a successful decode occurs, which storage is needed since the subsequently transmitted redundancy blocks (e.g., R1 and R2) cannot be independently decoded to give the same performance as if combined decoding was used. The storage requirements are further increased if the receiver stores a multi-bit soft value associated with each received bit, the soft values indicating a confidence level associated with the decoding of the received bit.

Eriksson at NOK00083211, 3:39-52.

321. However, as noted in TR 25.896, Node Bs can also suffer from high amounts of overhead signaling in the downlink direction causing interference in the uplink, as well as limited memory due to the number of UEs communicating with each Node B. TR 25.896 at NOK00074953, § 7.2.3 “Associated Signaling.”

322. Eriksson suggests sending direct messages between Node Bs and UEs to allow for dynamic changes of the balance between FEC and ARQ based on both channel conditions and the memory status of either the UE or Node B.

MCS changes may or may not be based solely on reported link quality estimates. For example, when incremental redundancy is used and the receiver has limited memory it may be beneficial to increase the robustness of the MCS even though (in a system with unlimited memory) it would decrease throughput. Consider the following scenario. Using a less robust MCS, the number of required retransmissions for successful incremental redundancy combination will be higher. This, in turn, requires a lot of memory. If the receiver runs out of memory, it will begin to discard received blocks that have previously been stored for later incremental redundancy combination. Since the information transmitted using the relatively unrobust MCS probably relies in part upon incremental redundancy combining to achieve acceptable decoding performance, the result may be significant degradation in received signal quality. Hence it may be better under such circumstances to dynamically increase the robustness of the transmission's MCS, e.g., when the receiver starts to run out of memory.

Yet another factor which further complicates this process is the possibility of changing the MCS associated with blocks that are being retransmitted. If link

adaptation is used without incremental redundancy, then changing the MCS for retransmissions may be very desirable based upon the measured link quality. On the other hand, if incremental redundancy is employed, using a different MCS may make it impossible to combine the retransmitted block with the originally transmitted block. However, if the link quality changes significantly it still may be desirable to change the MCS even if some of the earlier transmitted data blocks cannot be used in the redundancy combining process at the receiver.

Thus it can be seen that there are many challenges associated with optimizing the manner in which these various techniques are employed. To enable dynamic changing of the MCS during a connection, some form of overhead signalling is necessary between the transmitter and receiver.

Eriksson at NOK00083211, 3:63-4:32.

G. PCT Publication WO2006/030478 / U.S. Patent No. 8,775,890 to Yap et al. (“Yap”) (NOK00084269)

323. U.S. Patent No. 8,775,890 issued to Yap, et al., claims priority to a PCT application filed on September 13, 2004. The PCT application published as WO2006/030478 on March 23, 2006. The '890 patent represents the applicant's English translation of the disclosure of PCT publication WO2006/030478. I have been informed and understand that based on the publication date of the PCT publication, Yap is prior art to the '348 and '809 patents under at least 35 U.S.C. § 102(a). For ease of reference, the citations to Yap below are to the '890 patent.

324. Yap describes an “automatic retransmission request control system in an OFDM-MIMO communication system.” Yap at NOK00084269, Abstract. To facilitate retransmission, Yap discloses the use of a CRC check at a receiver to check for errored transmissions and an ACK/NACK messaging scheme in which the receiver signals to the transmitter whether or not retransmission is required. As described by Yap, “[a]n ARQ controlling section module at the transmitter determines whether or not retransmission of signals is required. . . The ARQ module makes decisions based on ARQ feedback information from the receiver. . . In the system of the present invention, the receiver carries out cyclic redundancy checks (CRC) for antenna chains and the transmitter determines whether or not data retransmission is required for transmission antennas

by feedback information of acknowledgement (ACK) or negative acknowledgment (NACK) based on a result of CRC from the receiver.” Yap at NOK00084269, 2:35-49.

325. Yap further describes processing of incoming data prior to transmission, including the use of interleaving on data to be transmitted.

At transmitter 100, data processing is performed for each individual antenna chain. Different streams of independent data are transmitted from the individual transmission antennas. The input data is first attached the cyclic redundancy check (CRC) code at CRC attaching section 102. Then, channel coding such as convolutional coding and turbo coding is carried out at coding section 104. The coded data will then be interleaved by interleaver 106 to reduce burst errors in the data. M-ary modulation constellation symbol mapping is executed on the interleaved data at mapping section 108.

Yap at NOK00084270, 4:8-19.

326. Deinterleaving, decoding, and error detection is then applied in the receiver: “After carrying out demapping (demapping section 216), deinterleaving (deinterleaver 218) and decoding (decoding section 220), cyclic redundancy check (CRC processing section 222) is then performed on each packet to validate the data.” Yap at NOK00084270.

327. Based on the results of the CRC, the receiver will then send either an acknowledgement (ACK) or negative acknowledge (NACK) to indicate whether the transmitted packet needs to be retransmitted: “If it is determined that the checked packet does not include error, acknowledgment (ACK) is transmitted to the transmitter and the transmitter does not retransmit the packet. If there is an error, a negative acknowledgment (NACK) is transmitted to transmitter 100 for retransmission request.” Yap at NOK00084270, 4:49-53.

XI. PRIOR ART ANALYSIS

A. Motivations to Combine

1. Chen in View of ADSL2 Standard

328. A POSITA would have been motivated to combine the disclosure of Chen with ADSL2. Chen describes a combination of techniques that work with both unicast and multicast transmissions in wireless or wired transmission system. Chen at NOK0007559, col. 1 ¶ 1 (“Congestion, tunneling, and lack of high-speed backbones are among the reasons for occasional data loss [in wired systems]. On the wireless side ... Numerous techniques have been suggested for error and loss control over the Internet.”), page 4, col. 2 ¶ 2 (“In this section, we present the theoretical analysis of our interleaved FEC/ARQ scheme in a step-by-step way. First, we analyze the basic characteristics of our hybrid FEC/ARQ technique in unicast.”). ADSL2 describes a wired multicarrier communication system. ADSL2 at 89, §§ 8.8-8.8.1 “Modulation” and “Subcarriers.” ADSL2 further describes using interleaving and Reed-Solomon coding to protect transmissions from errors. ADSL2 at 38, § 7.7.1.5 “Interleaver.”

329. A POSITA would have understood that Chen’s techniques could be used with the multicarrier transceiver described in ADSL2 because Chen’s techniques are design to work with any transceiver that can communicate over the Internet. Chen at NOK0007559, col. 1 ¶ 1. ADSL2 systems were, in fact, widely used to transmit data and enable communications over the Internet. The primary difference between Chen and ADSL2 is that Chen suggests using retransmission to protect data from errors, while ADSL2 suggests using retransmission to protect overhead signaling from errors. Compare Chen at NOK00075561, col. 2 ¶¶ 1-2 with ADSL2 at 43, § 7.8.2.4.3 “Overhead message segmentation.” As already noted in §§ VIIA.4, IX.A, IX.C , retransmission was a common method to combat noise in a transmission system, and would have been well understood by a POSITA. A POSITA would have understood that using retransmission to correct

errors in a multicarrier system was no different than using retransmission in a non-multicarrier system. *See, e.g.*, §§ VII.A.1, V11.A.4, V11.B, and IX.A. Thus a POSITA would have been motivated to use Chen's error correction techniques, including interleaving, ARQ, and Reed-Solomon codewords, with ADSL2's multicarrier transceiver.

330. Combining Chen and ADSL2 would have been obvious for the separate reason that using Chen's techniques with a multicarrier system is one of a limited number of possibilities. For example, as set out clearly in Chen, there are two basic types of communications system: wired and wireless. Chen at NOK0007559, col. 1 ¶ 1. Among these systems, there are a limited number of methods to modulate and transmit data; one of the best-known and most advantageous techniques is "discrete multitone" or DMT modulation, which is the type of modulation used in a multicarrier system. Multicarrier systems have several large advantages over other systems, including efficient bandwidth use with less intersymbol interference than in a comparable single-carrier system. § VIIA.1 *supra*. A POSITA would have therefore been motivated to try using Chen's techniques in a known multicarrier system, such as the popular ADSL2 system, to improve data throughput such as (for example) better video delivery.

331. In fact, the prior art IEEE 802.16-2004 ("WiMAX") standard was a multicarrier system that used Reed-Solomon coding, interleaving, and ARQ to prevent and correct errors. § VII.B. The WiMAX standard would therefore suggest to a POSITA that Chen and ADSL2 could be combined because the WiMAX standard itself contains all the elements found in Chen and uses a multicarrier transceiver like that in ADSL2, but adds in the additional complication of wireless transmission. *Id.*; *see also* Chen at NOK0007559, col. 1 ¶ 1, and page 3, col. 2 ¶¶ 1-2 and ADSL2 at 38, §§7.7.1.5 "Interleaver" and 89, 8.8-8.8.1 "Modulation" and "Subcarriers."

332. A POSITA would have recognized that combining Chen and ADSL2 would have the predictable and expected result of adding Chen’s advantages—as described in Section IV of the Chen paper—to the ADSL2 system. Chen at NOK00075562, col. 2 ¶ 2-page 5, col. 2 ¶ 1. ADSL2 uses two of the three error correction methods found in Chen—interleaving and Reed-Solomon coding—to transmit regular data in a multicarrier system. ADSL2 at 38, §§ 7.7.1.5 “Interleaver” and 89, 8.8-8.8.1 “Modulation” and “Subcarriers.” Adding retransmission to the combined multicarrier system would have had a predictable result to a POSITA, as evidenced by the fact that ADSL2 uses retransmission to correct errors in overhead messages. ADSL2 at 43-46, § 7.8.2.4.3 “Overhead message segmentation”; *see also supra* §§ VII.C and IX.A. A POSITA would have further understood that the combination of Chen and ADSL2 was predictable from the WiMAX standard, which combined a multicarrier system with interleaving, retransmission, and Reed-Solomon coding. § VII.B. The use of retransmission (along with interleaving and Reed-Solomon coding) within the ADSL2 standard, and with the separate use of retransmission, interleaving, and Reed-Solomon coding in the WiMAX standard, both published before the priority date of the Family 9 Patents, is further evidence that a POSITA would have found the combination of Chen and ADSL2 to have predictable results.

2. Yap in View of VDSL1

333. A POSITA would have been motivated to combine the teachings of Yap with those of VDSL1. First, both Yap and VDSL1 relate to multicarrier transceivers (OFDM in the case of Yap and the related DMT in the case of VDSL1). Compare Yap at NOK00084270, 3:65-4:2 with VDSL1 at NOK00078792, § 9.2.1 “Multi-carrier modulation.” Further, both Yap and VDSL1 use interleaving and error correction coding to protect transmitted data from errors. *Compare* Yap at NOK00084270, 4:8-19, Fig. 1 with VDSL1 at NOK00078785, § 8.4.1 “Interleaving – General,”

NOK00078784, Figure 8-1 and §§ 8.2-8.4 “Scrambler,” “Forward error correction,” and “Interleaving” (incoming data is scrambled, FEC is applied, and then data is interleaved).

334. A POSITA would have been motivated to combine Yap’s technique of using retransmission as an error correction technique (in addition to error correction coding and interleaving) and its associated ACK/NACK messaging to VDSL1 in order to enjoy the additional error protection provided by retransmission. A POSITA would have a strong expectation of success in doing so as the combination is nothing more than applying a well-known technique in a substantially similar environment. Indeed, VDSL1 already applied retransmission during initialization. Yap simply provides a teaching to apply the retransmission technique to Showtime communications where other error correction techniques like error correction coding and interleaving are also applied.

3. VDSL1 in View of ADSL2

335. A POSITA would have been motivated to combine the teachings of VDSL1 with those of ADSL2. First, both ADSL2 and VDSL1 use the same type of transceiver—a multicarrier transceiver. VDSL1 at NOK00078792, § 9.2.1 “Multi-carrier modulation” *with* ADSL2 at 89, §§ 8.8-8.8.1 “Modulation” and “Subcarriers.” Both ADSL2 and VDSL2 use similar types of interleaving to protect transmitted data from errors. Compare VDSL1 at NOK00078785, § 8.4.1 “Interleaving – General” with ADSL2 at 38, § 7.7.1.5 “Interleaver.” Moreover, both ADSL2 and VDSL1 subject data for transmission to the same basic steps. ADSL2 at 227, Figure C.7-1 and 36-37, §§ 7.7.1.3-7.7.1.5 “Scrambler,” “Forward error correction function,” and “Interleaver” (overhead data and bearer data is multiplexed, the scrambled, FEC is applied, and then data is interleaved) *with* VDSL1 at NOK00078784-787, Figure 8-1 and §§ 8.2-8.4 “Scrambler,” “Forward error correction,” and “Interleaving” (incoming data is scrambled, FEC is applied, and then data is interleaved).

336. The main difference between ADSL2 and VDSL1 is that ADSL2 provides an overhead channel whose messages are protected from errors by a packet retransmission function. ADSL2 at 39-40, 43-44, §§ 7.8.2, 7.8.2.1, and 7.8.2.4.3 “Overhead channel access,” “Overhead channel structure,” and “Overhead message segmentation.” ADSL2’s overhead channel is formed from sync bytes in data frames. *Id.* In contrast, VDSL1 protects initialization messages with segmentation and retransmission. *See, e.g.*, VDSL1 at NOK00078835, § 12.4.2.6 “Segmentation of messages.” VDSL1 has a VDSL overhead control channel that can be used to change the transmitted power. VDSL1 at NOK00078802 and NOK00078819, §§ 10.2.3, 10.6 “VDSL overhead control (VOC) channel” and “VDSL Overhead Channel (VOC).” However, VDSL1 does not provide for retransmission in the overhead channel. *Id.* A POSITA would have been motivated to provide retransmission over the VOC channel like that used in ADSL2’s overhead channel to, for example, allow the VOC channel to be used to control more functions (similar to the ADSL2 overhead channel, which is used to control transmitted power and perform tasks like online reconfiguration). ADSL2 at 173, § 9.4.1 “Commands.” A POSITA would have had a reasonable expectation of success in modifying the VDSL1 overhead channel based on the ADSL2 overhead channel because of the similarity between both systems. As already noted, both VDSL1 and ADSL2 use the same type of modulation and both use extremely similar steps to ready data for transmission. Compare ADSL2 at 227 and 36-39 Figure C.7-1 and §§ 7.7.1.3-7.7.1.5 “Scrambler,” “Forward error correction function,” and “Interleaver” with VDSL1 at NOK00078784-787, Figure 8-1 and §§ 8.2-8.4 “Scrambler,” “Forward error correction,” and “Interleaving.” Further, retransmission is a long-studied method to ensure that data is received correctly which the VDSL1 system already used retransmission to ensure that initialization messages are correctly received. VDSL1 at

NOK00078835-836, § 12.4.2.6 “Segmentation of messages.” Thus it would be a routine matter for a POSITA to modify the VDSL1 overhead channel to utilize retransmission.

337. A POSITA would have further been motivated to provide for retransmission of data, rather than only overhead, in the combined VDSL1/ADSL2 system. For example, the effect of providing for data retransmissions has been extensively studied in the prior art. *See, e.g.*, §§ VII.A.4, VII.B, IX.C, and IX.G. Retransmitting data has been a standard tool in communications when, for example, essential data is received in a non-usable (errored beyond correction) form. Stephen Wicker, *Error Control Systems for Digital Communication and Storage* at page 393. For example, the Aramvith paper specifically studies the effect of using interleaving in conjunction with retransmission. § IX.G. Similarly, both ADSL2 and VDSL1 use retransmission to ensure that initialization messages are properly received. ADSL2 at 158, § 8.15.5.1.2 “Message flow, acknowledgement and retransmission” and VDSL1 at NOK00078835, § 12.4.2.6 “Segmentation of messages.” And, perhaps most significantly, the Internet (which uses the TCP/IP protocol) includes a retransmission mechanism and operates over VDSL1 and ADSL2 lines during Showtime. § VII.A.4 *supra*. Thus, a POSITA would have understood that the combined VDSL1/ADSL2 system was compatible with retransmitting data during Showtime.

338. A POSITA would have had a reasonable expectation of success in modifying the VDSL1/ADSL2 system to include retransmission during Showtime. First, both ADSL2 and VDSL1 already use retransmission to ensure that initialization messages (and overhead messages in the case of ADSL2) are correctly received. ADSL2 at 43-46 and 158, §§ 7.8.2.4.3 and 8.15.5.1.2 “Overhead message segmentation” and “Message flow, acknowledgement and retransmission” and VDSL1 at 066720-721, § 12.4.2.6 “Segmentation of messages.” ADSL2 overhead channel retransmission already functions during Showtime. ADSL2 at 43-46, § 7, 7.8.2.4.3 “Overhead message

segmentation.” Further, as already noted, when accessing the Internet over an ADSL2 or VDSL1 connection, the TCP/IP protocol provides for retransmission of data during Showtime. § VII.A.4 *supra*. Finally, examples of systems using retransmission during the equivalent of Showtime abound. §§ VII.B (IEEE 802.16 “WiMAX”), IX.C, IX.E, IX.F, IX.G. A POSITA would have therefore understood that using retransmission to protect transmitted data (rather than just overhead data) was a well-understood solution to the problem of burst errors in communication systems.

4. 3GPP TR 25.896 v6.0.0 and TS 25.401 v6.0.0 in View of Eriksson

339. A POSITA would have been motivated to combine the disclosure of TR 25.896 and TS 25.401. First, TR 25.896 specifically studies the implementation of “potential enhancements under consideration and compare the benefits of each enhancement with earlier releases for improving the performance of the dedicated transport channels in UTRA FDD uplink.” TR 25.896 at NOK00074934, § 1 “Scope.” TR 25.896 further explains that it “involves the Radio Access work area of the 3GPP studies and has impacts on both the Mobile Equipment and Access Network of the 3GPP systems.” *Id.* TR 25.896 describes how a channel called “E-DCH” or the “Enhanced DCH” provides enhanced uplink services. *Id.* § 3 “Definitions, symbols and abbreviations.”

340. TS 25.401 provides baseline definitions and descriptions of the 3GPP system that TR 25.896 enhances. TS 25.401 at NOK00075141, § 1 “Scope” (“The present document describes the overall architecture of the UTRAN, including internal interfaces and assumptions on the radio and Iu interfaces.”). For example, while TR 25.896 specifically enhances the FDD uplink in the UTRAN, TS 25.401 describes the baseline FDD uplink in the UTRAN. TS 25.401 at NOK00075147, § 6 “UTRAN Architecture” (“The UTRAN consists of a set of Radio Network Subsystems connected to the Core Network through the Iu. ¶A RNS consists of a Radio Network Controller one or more Node Bs and optionally one SAS. A Node B is connected to the RNC through the Iub interface. ¶A Node B can support FDD mode, TDD mode or dual-mode

operation.”). In fact, TS 25.401 describes non-enhanced dedicated channels—“DCH” rather than the “E-DCH” described in TR 25.896. TS 25.401 at NOK00075144 and NOK00075174, §§ 3.2 “Abbreviations” (“DCH – Dedicated Channel”) and 11.2.4 (“DCH Transport Channel”). Thus, a POSITA would have referred to TS 25.401 to understand the disclosure of TR 25.896, especially with respect to unchanged portions of the standard, which include the basic definitions of “UE” (user equipment), “Node B” (the portion of the RNS that communicates with the UE, often called a base station), “RNS” (radio network subsystem), and for background on how the proposals in TR 25.896 provide an enhanced uplink between the UE and Node B.

341. A POSITA would have been motivated to combine the suggestion to explicitly send messages allocating a shared memory between a retransmission function and an interleaving or deinterleaving function found in Eriksson with the disclosure of TR 25.896 and TS 25.401. TR 25.896 describes the advantages of using hybrid ARQ with incremental redundancy to correct and detect errors in a cellular environment. TR 25.896 at NOK00074951-952, § 7.2.1 General. Eriksson also discloses the advantages of using hybrid ARQ with incremental redundancy to correct and detect errors in a cellular environment. Eriksson at NOK00083210-211, 2:15-49, 2:65-3:8, and 3:23-52. Eriksson provides further discussion of the advantages of using FEC (such as interleaving) with hybrid ARQ retransmission, as well as potential incentives for a cellular transceiver to prefer using FEC versus hybrid ARQ to detect and correct errors. Eriksson at NOK00083210, 2:15-49, 2:65-3:8, 3:23-52, and 3:63-4:32.

342. Eriksson then suggests that having the transceivers used in a cellular environment exchange messages about which of the error correction/detection techniques is more advantageous and should therefore be allocated more memory—for example, if the transceiver is running out of memory to store transmitted packets for potential later retransmission, then FEC (which uses

interleaving and other techniques to prevent and correct errors) is preferable to hybrid ARQ. Eriksson at NOK00083211, 3:63-4:32. On the other hand, if the transceiver has plenty of memory available to store transmitted packets for potential later retransmission and the channel between the transceivers is causing bursts of interference that cause the loss of full packets, then the transceiver may prefer to use hybrid ARQ and apply less FEC (and therefore less interleaving) and rely more on retransmission to correct errors. *Id.*

343. 3GPP TR 25.896 and TS 25.401 benefit from using Eriksson's suggested messages conveying an explicit memory allocation described in Eriksson because they increase the flexibility of hybrid ARQ and allows for better adaptation to different link qualities where interleaving or retransmission may be more or less valuable in dealing with noise. Eriksson at NOK00083211, 3:63-4:32.

344. Incorporating explicit messages supporting memory sharing between interleaving and retransmission suggested by Eriksson would not interfere with the operation of the system and method described in 3GPP TS 25.401 and 3GPP TR 25.896, since the memory sharing described in Eriksson already works with a system that uses retransmission and interleaving and exchanges parameters to allow a shared memory to be allocated between retransmission and interleaving. Eriksson at NOK00083210-211, 2:65-3:8, 3:23-38.

345. Further, incorporating the messages that explicitly convey memory allocations suggested by Eriksson is a simple modification of the disclosure of TR 25.896 and TS 25.401 that has the advantage of making the system in TR 25.896 and TS 25.401 more responsive to changing conditions, therefore making the overall system more optimal and desirable. Eriksson at NOK00083210-211, 2:15-49, 2:65-3:8, 3:23-52, and 3:63-4:32.

B. The Asserted Claims of the '411 Patent Are Obvious

346. The asserted claims of the '411 Patent are claims 10 and 18. The following charts summarize my citations to the prior art for the asserted claims:

Ground	Claims	Summary Claim Chart
G.993.1 in view of G.992.3	10, 18	Exhibit A-1
TR 25.896 and TS 25.401 in view of Eriksson	10, 18	Exhibit A-2

347. The asserted claims of the '411 Patent are obvious. I address each limitation of the asserted claims in turn below.

1. Claim 10

[10.a] A transceiver capable of packet retransmission comprising:

348. To the extent the court determines that the preamble is limiting, as described above in Section VII.A.4, transceivers capable of packet retransmission were well-known in the art long before the priority date of the '411 patent. Examples of various systems that used packet retransmission are listed in Sections VII.A.4 and VII.B-E above, as well as in Sections IX.A-C.

i VDSL1 in View of ADSL2

349. VDSL1 and ADLS2 both disclose “a transceiver capable of packet retransmission.” Specifically, both VDSL1 and ADSL2 disclose using packet retransmission to protect various messages from errors in conjunction with transceivers. VDSL1 is replete with references to transceivers and directly teaches the use of transceivers as the title is “Very high speed digital subscriber line transceivers.” Additionally, VDSL1 teaches the use of common shared circuitry in a transceiver. For example, VDSL1 Fig. I.7 teaches the use of a common diplexer in a VDSL transceiver.

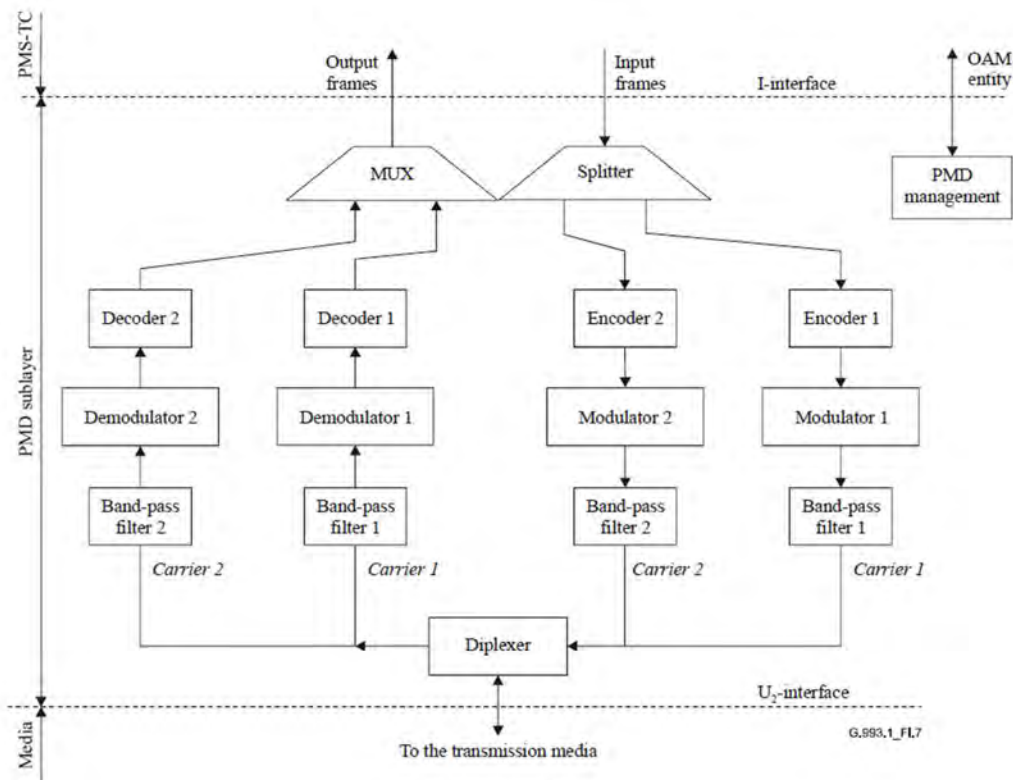


Figure I.7/G.993.1 – VTU PMD sublayer functional model

350. ADSL2 also expressly teaches transceivers as it is entitled “Asymmetric digital subscriber line transceivers.” ADSL2 is also replete with teachings of transceivers. For example, the summary states that “[t]his Recommendation describes Asymmetric Digital Subscriber Line (ADSL) transceivers on a metallic twisted pair that allows high-speed data transmission between the network operator end (ATU-C) and the customer end (ATU-R).” A POSITA would have known that it is common for DSL transceivers to have some common shared circuitry.

351. Moreover, it was a common practice well known by a POSITA at the time of the invention for the transmitter and receiver portion of a communications device such as a DSL transmitter/receiver to share circuitry. At the very least, such devices share circuitry related to providing power. In typical transceiver architectures known to a POSITA at the time of the invention, transmitting and receiving functions will additionally share a common processing resource such as a digital signal processing chip and/or a common memory resource. For example,

a single digital signal processing chip was commonly used for both transmitting and receiving functions. Both transmitting and receiving functions commonly share the same clock. Both transmitting and receiving functions commonly share the same antenna and associated analog circuitry. As another example, whenever the device utilizes shared memory for interleaving and deinterleaving, the device would of necessity share common circuitry related to the shared memory

352. ADSL2 uses retransmission to ensure that certain overhead messages (or segments of those messages) are received correctly. To support this, ADSL2 uses a timer and required acknowledgment or negative acknowledgment (ACK or NACK) signals.

353. Each transmitted message segment shall be acknowledged by the far-end with a segment acknowledge message, except the last one. The last command message segment shall be acknowledged by the far-end with the appropriate response message (identical to the non-segmented case). The last response message segment shall not be acknowledged. Table 7-17a defines the segment acknowledge message. The following shall apply to all encapsulated segment acknowledge messages:

- HDLC frame shall contain 5 message octets ($P = 5$);
- The two least significant bits of the address field shall be identical to the corresponding bits in the address field of the acknowledged message segment. All other bits of the address field shall be set to 0₂;
- The second least significant bit of the control field shall either indicate a command message (request to continue the response, e.g., L2 grant) or a response message (request to continue the command, e.g., OLR);
- The least significant bit of the control field shall toggle for each acknowledgement message as it normally toggles for each command/response message (see 7.8.2.4.2);
- All other bits of the control field shall be set to 0₂.

Table 7-17a/G.992.3 – Segment acknowledge message

Message octet number	Message octet definition
Octet 1	Message designator 1111 0000 _b for acknowledgement of high priority message segment 1111 0001 _b for acknowledgement of normal priority message segment 1111 0010 _b for acknowledgement of low priority message segment
Octet 2	Message type of segment acknowledge message 01 ₁₆
Octet 3	Acknowledged message segment ID (in range 1 to 7)
Octet 4	Message Designator (first message octet of acknowledged message segment)
Octet 5	Message Type (second message octet of acknowledged message segment)

Every message segment shall be acknowledged by the far-end before the next message segment is transmitted. The segment acknowledge message shall not be segmented.

Timeouts shall be defined as follows:

- For a non-segmented message, timeout shall apply between the last octet of the command message transmitted and the first octet of the response message received.
- For a segmented response message, the timeout shall apply between the last octet of the command message transmitted and the first octet of the first message segment received.
- A timeout corresponding to the command priority shall apply between the last octet of the message segment transmitted and the first octet of the segment acknowledge message received.
- Another timeout corresponding to the command priority shall apply between the last octet of the segment acknowledge message transmitted and the first octet of the next message segment received.
- If a timeout expires, the transmitter may repeat the last transmitted message. This repeated message can be an unsegmented command message, a command/response message segment (except the last segment of a response message), or a segment acknowledge message. Alternately, the ATU may abandon the message after an implementation-specific number of retransmissions.

ADSL2 at 43-46, § 7.8.2.4.3 “Overhead message segmentation.” ADSL2 provides for similar retransmission over other messages. *See, e.g.:*

At the start of the Exchange Phase, the ATU-C shall transition to the C-TREF1-LD state (in which C-TREF symbols shall be transmitted until the first R-MSGx-LD message is received).

If the ATU-C receives an R-MSGx-LD message, the ATU-C shall transition to the C-ACK or C-NACK state within 128 symbols from the end of the R-MSGx-LD state. If the R-MSGx-LD message is successfully received, the ATU-C shall transition to the C-ACK state (in which a positive acknowledgment C-ACK message shall be transmitted). Instead, if a decoding error occurs (i.e., the CRC locally computed at the ATU-C does not correspond to the CRC transmitted by the ATU-R), the ATU-C shall transition to the C-NACK state.

The C-ACK message shall be represented by the "01010101" octet and shall be transmitted over 64 symbol periods using the same modulation technique as the loop diagnostics information bearing messages. No CRC shall be added to the C-ACK message. In the C-NACK state, the ATU-C shall transmit 64 C-TREF symbols. Note that from the ATU-R's perspective, this is equivalent to the ATU-C not responding to the R-MSGx-LD message.

At the end of the C-ACK or C-NACK state, the ATU-C shall transition to the C-TREF2-LD state (in which 256 C-TREF symbols shall be transmitted). During the C-TREF2-LD state, the ATU-R transitions to the R-QUIET2-LD state (because the R-ACK message is successfully received and no more R-MSGx-LD messages remain to be transmitted) or the ATU-R transitions to the R-SEGUE-LD state (because no, or a corrupted C-ACK message is received or more R-MSGx-LD messages remain to be transmitted). At the end of the C-TREF2-LD state, the ATU-C shall transition to the C-SEGUE-LD state (if the ATU-R has transitioned to the R-QUIET2-LD state) or shall return to the C-TREF1-LD state (if the ATU-R has returned to the R-SEGUE-LD state).

Note that, as a result of a corrupted C-ACK message, the ATU-C could successfully receive the same message twice. In this case, the ATU-C shall ignore the second identical (same Sequence Number) message.

ADSL2 at 158-159, § 8.15.5.1.2 "Message flow, acknowledgement and retransmission."

354. VDSL1 also discloses using packet retransmission to protect initialization messages from errors. Specifically, VDSL1 may segment initialization messages; in that case, each segment is given a unique identifier. The receiving transceiver acknowledges each segment received, including an identification of the segments being acknowledged, or may transmit a repeat request for particular segments not correctly received.

12.4.2.6 Segmentation of messages

Some messages could potentially be large, and even larger than the maximal allowed frame size of an HDLC frame (1026 octets). It shall therefore be possible to segment messages before transmission. In order to do this, all messages transmitted during initialization shall receive a number that counts the message. This number shall be stored in one byte and will wrap around in case of overflow. The value 0 shall not be used because it has a special meaning (see later). This means that 255 shall be followed by 1.

This “message index” shall be transmitted in the Address Field of the HDLC frame (see Figure 12-3). The message index allows to track lost messages and to request the retransmission of a particular message. The index shall initially be set to one and shall be incremented with one after the transmission of a message. The index shall not be incremented in case of a REPEAT-REQUEST. The counting of messages shall start when the transmission starts using RQ mode instead of automatic repeat (AR) mode.

A segmentation index (1 byte) shall be included in the Control Field of the HDLC frame. The four MSBs of this field shall indicate the number of segments that make up the total message. The four LSBs shall indicate the index of the current segment. For instance a value 0x93 indicates the third segment of a total of nine. In case the message is not segmented, the value of the field shall be 0x11.

The REPEAT-REQUEST message will behave differently from the other messages. The message index counter shall not be increased when a REPEAT-REQUEST message is sent. Also, the meaning of the message index and the segmentation index is different in this case.

The message index of the REPEAT-REQUEST message shall contain the message index of the message that should be retransmitted. The default value 0 indicates that the last unacknowledged message should be sent (this is why 0 shall never be used as message index for any other message). If this message contains several segments, only the last segment shall be retransmitted.

Likewise, the control field shall contain the segment that should be retransmitted. The Information payload of the REPEAT-REQUEST shall consist of one byte containing the value 0x55.

In the initialization procedure a transmitter should never send two consecutive messages without acknowledgement of the first message. It will always first receive some message from the other side before transmitting again. Therefore, an acknowledgement shall be sent for all but the last segment. Typically, the last segment signals the end of the message and it will therefore be acknowledged by the reply to this message. The ACK-SEG-message (see Table 12-12) shall be used to acknowledge the reception of the other segments. The ACK-SEG-message shall have its own message and segment index, which shall not refer to the segmented message that is being sent.

Once acknowledged, messages (or segments) are not expected to be retransmitted again. Also, a REPEAT-REQUEST that contains a number that is higher than the most recent value of the counter shall be ignored.

In AR mode, segmentation shall be done in the same way but there will be no acknowledgement (ACK-SEG) between different segments of the same message. Segments shall be sent in order.

VDSL1 at NOK00078835-836, § 12.4.2.6 “Segmentation of messages.”

355. ADSL2 and VDSL1 therefore each disclose retransmitting messages to ensure they are received correctly. As already described in § VII.A.4 above, retransmission was well-known and routinely implemented in the prior art to ensure that packets are correctly received. For example, retransmission in the form of automatic repeat request (that is, “ARQ”) was in use before 1960 and has been a standard tool in communications for years.

It is difficult to date the earliest introduction of ARQ protocols in digital communications. They were almost certainly an immediate consequence of the development of parity-check codes in the early days of digital computers. The next major development in the area came in 1960, when Wozencraft and Horstein [Woz3],[Woz4] described and analyzed a system that allowed for both error correction and error detection with retransmission requests. Their system, now known as a type-I hybrid-AREQ protocol, provided significantly improved performance over the pure ARQ protocols in many applications.

Stephen Wicker, Error Control Systems for Digital Communication and Storage at page 393.

356. Thus, to the extent that ADSL2’s use of retransmission in the overhead channel does not disclose or render obvious a transceiver capable of packet retransmission, a POSITA would have been able to incorporate a packet retransmission function into VDSL1 or ADSL2 without issues.

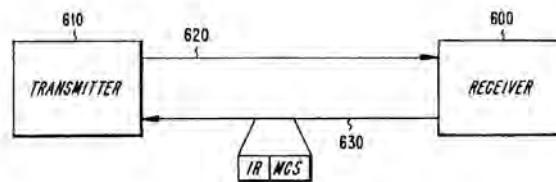
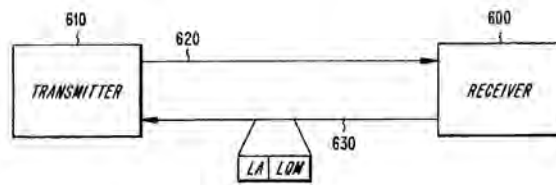
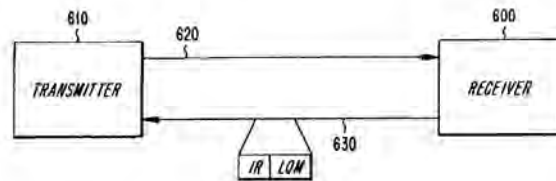
357. Finally, I note that, as specifically pointed out in § VII.A.4, the TCP/IP standard (which is the protocol used for Internet traffic) uses retransmission. Specifically, the transport control protocol (“TCP”) includes a retransmission mechanism. Both VDSL1 and ADSL2 were (and ADSL2 is) widely used by consumers to access the Internet. Thus, a POSITA would have

recognized that the data transmission in VDSL1 and ADSL2 was used with a retransmission protocol when VDSL1 and ADSL2 were used to for Internet access. § VII.A.4 *supra*.

358. Accordingly, VDSL1 and ADSL2 both disclose and (together) render obvious “a transceiver capable of packet retransmission” (ADSL2 and VDSL1 describe using a packet retransmission function, while a POSITA would have found it obvious to use both VDSL1 and ADSL2 with retransmission to protect non-overhead data, including as described in the TCP/IP protocol).

ii TR 25.896 and TS 25.401 in View of Eriksson

359. TR 25.896 and TS 25.401 in view of Eriksson discloses and/or renders obvious “a transceiver capable of packet retransmission.” Specifically, both TR 25.896 and Eriksson disclose a transceiver capable of packet retransmission (the UE stores packets for later retransmission packets if requested, while the Node B stores received packets for eventual soft combining). Packet retransmission inherently requires that the transceiver have a transmitter and a receiver sharing common circuitry. For example, Eriksson discloses a transceiver that may transmit packets to another transceiver, such as a mobile radio telephone. Eriksson at 2:50-64. It explains that if “there are errors, then a request is sent back to the transmitting entity, e.g., a base station in a radiocommunication system, denoting the block to be resent.” *Id.* Such a scheme is shown in Figs. 6(a)-6(c), below.



360. Here, each transmitter unit, such as a base station, both transmits and receives messages. Likewise, each receiver unit, such as a mobile radio telephone, transmits and receives messages. Because the receiver unit is able to detect errors that are received and subsequently transmit a message indicating that error, and because the transmitter unit is able to receive an error message and subsequently resend the packet, a POSITA would have understood that these are transceivers with a transmitter and receiver sharing common circuitry.

361. TR 25.896 Node Bs (base stations) and UEs (user equipment) are communications transceivers and use an uplink hybrid ARQ scheme. This is described in section 7.2 “Hybrid ARQ”:

Node B controlled hybrid ARQ allows for rapid retransmissions of erroneously received data units, thus reducing the number of RLC retransmissions and the associated delays. This can improve the quality of service experienced by the end user. As a Node B controlled retransmission is less costly from a delay perspective, the physical channel can be operated with somewhat higher error probability than in Rel 5, which may result in improved system capacity. The retransmission probability for the initial transmission is preferably in the order of 10-20% when evaluating hybrid ARQ as closed loop power control is used for the uplink, maintaining a given quality level. Significantly higher retransmission probabilities may lead to considerably reduced end user throughput, while at very small retransmission probabilities the Node B controlled hybrid ARQ will not provide any additional gains compared to R99/4/5. Soft combining can further improve the performance of a Node B controlled hybrid ARQ mechanism.

TR 25.896 at NOK00074951-952§ 7.2.1 “Hybrid ARQ – General”.

362. TR 25.896 further describes using multiple redundancy versions with hybrid ARQ:

Redundancy version. If multiple redundancy versions are supported, the redundancy version needs to be known to the Node B. The potential gains with explicit support of multiple redundancy versions should be carefully weighted against the increase in overhead due to the required signaling. Note that, unlike the HS-DSCH, the number of users simultaneously transmitting data in the uplink using hybrid ARQ may be significant.

TR 25.896 at NOK00074953, § 7.2.3 “Associated Signalling”.

363. Eriksson also describes that Node Bs and UEs (communications transceivers with a shared memory) at both ends of a cellular communication system use hybrid ARQ and incremental redundancy to correct errors:

Another proposed hybrid ARQ Scheme, sometimes referred to as incremental redundancy or type-II hybrid ARQ, provides for additional redundant bits to be transmitted if the originally transmitted block cannot be decoded. This scheme is conceptually illustrated in FIG. 3. Therein, three decoding attempts are made by the receiver. First, the receiver attempts to decode the originally received data block (with or without redundancy). Upon failure, the receiver then receives additional redundant bits R1, which it uses in conjunction with the originally transmitted data block to attempt decoding. As a third step, the receiver obtains another block of redundant information R2, which it uses in conjunction with the originally received data block and the block of redundant bits R1 to attempt decoding for a third time. This process can be repeated until successful decoding is achieved.

As compared with link adaptation, incremental redundancy does not require that link quality estimates be transmitted or used. However, one problem with this technique

is the large memory requirement associated with storing the data block (and possibly additional blocks of redundant bits) until a successful decode occurs, which storage is needed since the subsequently transmitted redundancy blocks (e.g., R1 and R2) cannot be independently decoded to give the same performance as if combined decoding was used. The storage requirements are further increased if the receiver stores a multi-bit soft value associated with each received bit, the soft values indicating a confidence level associated with the decoding of the received bit.

Eriksson at NOK00083211, 3:23-38.

364. Therefore TR 25.896 and Eriksson both disclose “a transceiver capable of packet retransmission” (both TR 25.896 and Eriksson disclose Node Bs and UEs using hybrid ARQ).

[10.b] a transmitter portion capable of: transmitting a plurality of packets,

365. Transceivers transmitting a plurality of packets were well-known in the art long before the priority date of the '411 patent. Examples of transceivers that transmit packets are replete in the prior art and described extensively in Sections VII.A.1-4, VII.B-E, and IX.A-I *supra* as well as below.

i VDSL1 in View of ADSL2

366. VDSL1 and ADSL2 both disclose “a transmitter portion capable of transmitting a plurality of packets.” Specifically, both VDSL1 and ADSL2 disclose transceivers that transmit a plurality of packets.

367. The ADSL2 standard transmits packets (groups of bytes) in the form of DMT symbols.

8.8 Modulation

The modulator shall modulate a constellation encoder output data frame or sync frame (containing $NSC - 1$ complex values Z_i , $i = 1$ to $NSC - 1$) into a DMT symbol. The data frame can be taken from the data symbol constellation encoder (68 per superframe) as defined in 8.6. The sync frame can be taken from the synchronization symbol constellation encoder (1 per superframe) as defined in 8.7. For (short) initialization and diagnostics mode signals, the frame is defined in 8.13, 8.14 and 8.15.

8.8.1 Subcarriers

A DMT symbol consists of a set of subcarriers, with index $i = 0$ to NSC. The DMT subcarriers spacing Δf , shall be 4.3125 kHz, with a tolerance of ± 50 ppm. The subcarrier frequencies shall be $f_i = i \times \Delta f$, i 0 to NSC.

ADSL2 at 89, §§ 8.8-8.8.1 “Modulation” and “Subcarriers.”

368. ADSL2 includes a Packet Transmission Convergence function (PTM-TC) that facilitates transport of a packet stream in either the upstream or downstream direction using the PMS-TC layer, where the packet streams are processed and (eventually) scrambled, interleaved, and encoded with FEC.

K.3.1 Scope

The PTM-TC function provides procedures for the transport of one unidirectional packet stream in either the upstream or downstream direction. Packet boundaries, octet boundaries, and the position of most significant bits are explicitly maintained across the transport for the PTM-TC stream. The PTM-TC stream is presented asynchronously across the T-R or V-C reference point with respect to the PMD bit clocks.

The PTM-TC function is defined in terms of the PTM-TC defined in Annex H.1/G.993.1 [13]. Referring to the reference model of that annex, the PTM-TC of VDSL is defined connecting above the PMS-TC function to either a fast or slow channel through the a/b interface. This same function is used for clause K.3 and is defined to connect to a single PMS-TC latency path function.

ADSL2 at 376, § K.3.1 “Scope.” *See also id.* 377 Figure:

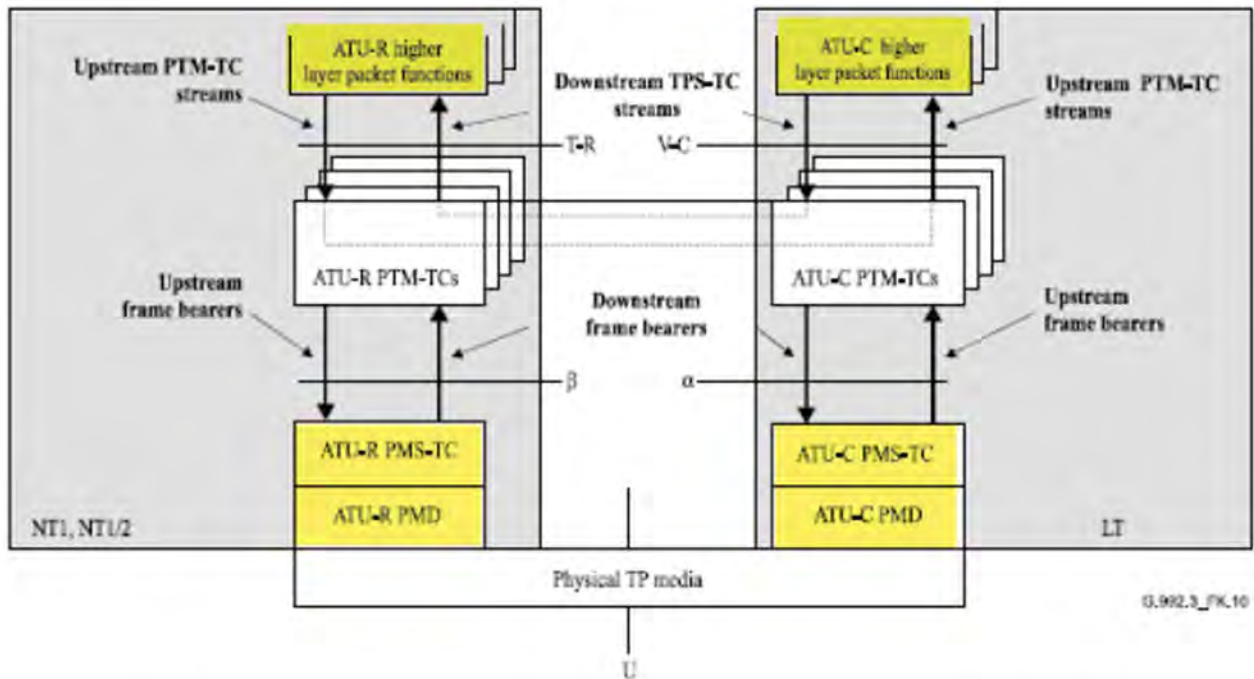


Figure K.10/G.992.3 – PTM-TC transport capabilities within the user plane

369. VDSL1 also breaks data into packets before passing that data to the PMS-TC layer, where the packet streams are processed and (eventually) scrambled, interleaved, and encoded with FEC.

H.1.1 Functional model

The functional model of packetized data transport is presented in Figure H.1. In the transmit direction, the PTM entity obtains data packets to be transported over VDSL from the application interface. The PTM entity processes each packet and applies it (depending on the latency requirements) to the y interface of the either Fast or Slow VDSL path intended for packetized data transport. The corresponding TPS-TC (PTM-TC) receives the packet from the y interface, encapsulates it into a special frame (PTM-TC frame) and maps into the PMS-TC frame (transmission frame) for transmission over the VDSL link.

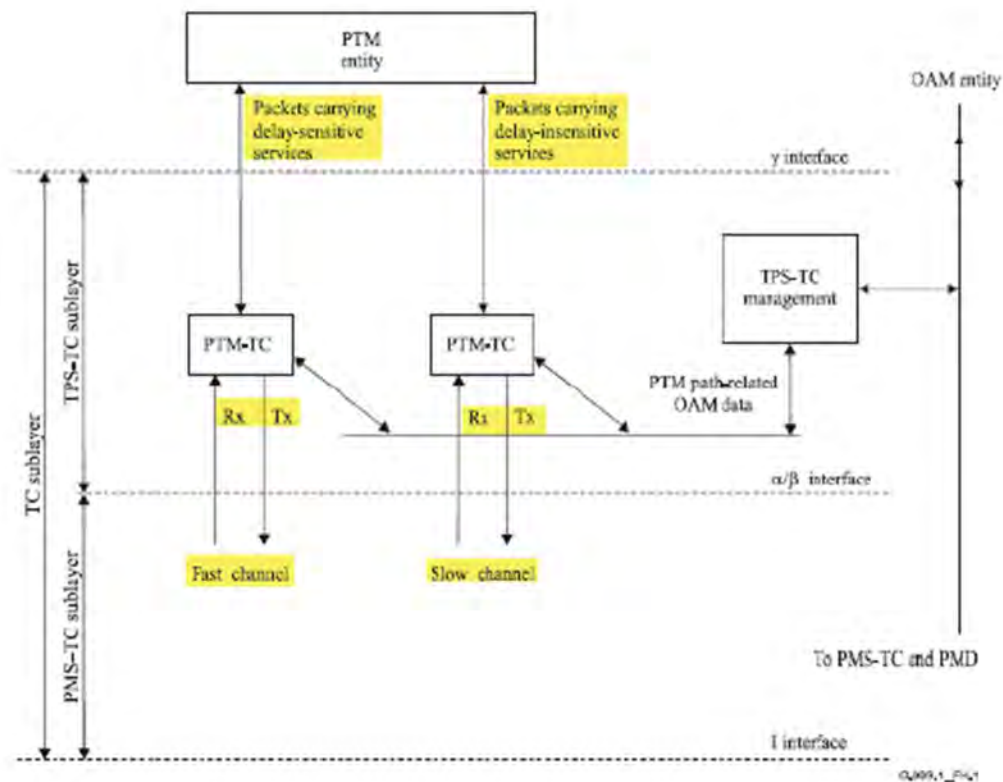


Figure H.1/G.993.1 – Functional model of PTM transport

In the receive direction, the PTM-TC frame extracted from the received PMS-TC frame is directed into the PTM-TC. The PTM-TC recovers the transported packet and delivers it to the PTM entity via the y interface.

VDSL1 at 066798-99, § H.1.1 “Functional model.”

370. Accordingly, VDSL1 and ADSL2 both disclose and render obvious “a transmitter portion capable of transmitting a plurality of packets” (ADSL2 and VDSL1 describe transmitting a plurality of packets).

ii TR 25.896 and TS 25.401 in View of Eriksson

371. TR 25.896 and TS 25.401 in view of Eriksson renders obvious “a transmitter portion capable of transmitting a plurality of packets.” Specifically, both TR 25.896 and Eriksson disclose a transmitter portion that transmits a plurality of packets.

372. TR 25.896 Node Bs (base stations) and UEs (user equipment) are communications transceivers and transmit a plurality of packets protected from errors by an uplink hybrid ARQ scheme. This is described in section 7.2 “Hybrid ARQ”:

Node B controlled hybrid ARQ allows for rapid retransmissions of erroneously received data units, thus reducing the number of RLC retransmissions and the associated delays. This can improve the quality of service experienced by the end user. As a Node B controlled retransmission is less costly from a delay perspective, the physical channel can be operated with somewhat higher error probability than in Rel 5, which may result in improved system capacity. The retransmission probability for the initial transmission is preferably in the order of 10-20% when evaluating hybrid ARQ as closed loop power control is used for the uplink, maintaining a given quality level. Significantly higher retransmission probabilities may lead to considerably reduced end user throughput, while at very small retransmission probabilities the Node B controlled hybrid ARQ will not provide any additional gains compared to R99/4/5. Soft combining can further improve the performance of a Node B controlled hybrid ARQ mechanism.

TR 25.896 at NOK00074951-952, § 7.2.1 “Hybrid ARQ – General”.

373. Eriksson also describes transmitting a plurality of packets protected from errors by an uplink hybrid ARQ scheme:

Another proposed hybrid ARQ Scheme, sometimes referred to as incremental redundancy or type-II hybrid ARQ, provides for additional redundant bits to be transmitted if the originally transmitted block cannot be decoded. This scheme is conceptually illustrated in FIG. 3. Therein, three decoding attempts are made by the receiver. First, the receiver attempts to decode the originally received data block (with or without redundancy). Upon failure, the receiver then receives additional redundant bits R1, which it uses in conjunction with the originally transmitted data block to attempt decoding. As a third step, the receiver obtains another block of redundant information R2, which it uses in conjunction with the originally received data block and the block of redundant bits R1 to attempt decoding for a third time. This process can be repeated until successful decoding is achieved.

As compared with link adaptation, incremental redundancy does not require that link quality estimates be transmitted or used. However, one problem with this technique is the large memory requirement associated with storing the data block (and possibly additional blocks of redundant bits) until a successful decode occurs, which storage is needed since the subsequently transmitted redundancy blocks (e.g., R1 and R2) cannot be independently decoded to give the same performance as if combined decoding was used. The storage requirements are further increased if the receiver stores a multi-bit soft value associated with each received bit, the soft values indicating a confidence level associated with the decoding of the received bit.

Eriksson at NOK00083211, 3:23-38.

374. Accordingly, TR 25.896 and Eriksson both disclose “a transmitter portion capable of transmitting a plurality of packets” (the Node Bs and UEs in both TR 25.896 and Eriksson transmit a plurality of packets).

[10.c] identifying at least one packet of the plurality of packets as a packet that should be retransmitted and allocating a memory between a retransmission function and an interleaving and/or deinterleaving function,

375. As described above in Sections VII.A.3-5, a POSITA would have understood that interleaving, retransmission, and sharing memory were all well-known in the art long before the priority date of the '411 patent. The claimed limitation of identifying at least one packet of the plurality of packets as a packet that should be retransmitted and allocating a memory between a retransmission function and an interleaving and/or deinterleaving function is therefore nothing more than the combination of known elements.

376. For example, as discussed above, the C programming language has explicit commands malloc and calloc to dynamically allocate memory which only make sense in the context of shared memory. § VII.A.5. It has been common practice for graduate students and engineers in industry to implement communications transceivers including multicarrier using the C programming language and its variants since at least the 1980's. As a practical example, consider a transmitter that is interleaving its packets before transmission and also saving transmitted packets for possible retransmission if needed. Having separate memories for interleaving and retransmission in the transmitter requires moving interleaved packets to a separate memory. A POSITA would therefore find it obvious to simply re-allocate that memory to the retransmission function, which accomplishes the task of saving the packets for retransmission without the additional and needless work of copying this large amount of data from one memory to another.

377. As another example, digital signal processors with a shared memory such as the AT&T DSP32 and the TI TMS 320 series of DSPs have been programmed to implement communications transceivers as a matter of course since DSPs were introduced no later than 1981 as discussed in ¶ 159. § VII.A.5. Moon and Mazzoni additionally describe transceivers with a shared memory. *Id.*

378. One example of a transceiver with a shared memory is the TNETD1500 ADSL Transceiver discussed in the 2001 Texas Instrument product bulletin entitled “AR5 Chipset Family: ADSL Router, Integrated Access Device and Residential Gateway Chipset Solutions”. § VII.A.5. The TNETD5100 includes a programmable 1600-MIPS TMS320C6200 DSP core and, as is common practice for DSPs in general and the TMS320 DSPs in particular, the DSP uses a shared memory which is a 1-Mbit on-chip instruction/data cache. *Id.* Thus a transceiver with a shared memory was well-known at the time of the alleged invention. *Id.*

379. Another example of a system to allocate memory between an interleaving function and a packet retransmission function is the Intel PRO/Wireless 5116 implementation of WiMAX (IEEE 802.16) as discussed above in §§ VII.A.5 and VII.B.

380. Another example is provided in §§ VII.A.5, including U.S. Patent No. 7,027,782 to Moon et. al. (“Moon”), which illustrates a system that allocates memory between an interleaving function and a packet retransmission function in a mobile communications system.

381. Another example is provided in §§ VII.A.5 ¶¶, which is U.S. Patent No. 7,269,208 to Mazzoni et al. (Mazzoni) also illustrates a system that allocates memory between an interleaving function and a packet retransmission function in a communications system.

382. Another example of allocating memory between an interleaving function and a retransmission function is provided in Chen, which is described in §§ VII.A.5 ¶¶ and in more detail in Section IX.C.

383. Further, as described below, many prior artists did in fact create a transceiver that identifies at least one packet of the plurality of packets as a packet that should be retransmitted and allocates a memory between a retransmission function and an interleaving and/or deinterleaving function. *See* Sections *i-iii* below. This shows that combining these old elements was predictable and within the skill of a POSITA. Thus, as shown below with the specific prior art combinations, this limitation is obvious as it is the combination of known elements in a known manner to reach a predictable result.

i VDSL1 in View of ADSL2

384. VDSL1 in view of ADSL2 renders obvious “identifying at least one packet of the plurality of packets as a packet that should be retransmitted and allocating a memory between a retransmission function and an interleaving and/or deinterleaving function.” Specifically, both ADSL2 and VDSL1 describe identifying at least one packet of the plurality of packets as a packet that should be retransmitted. In both cases the retransmission function requires the allocation of memory to retain information that may be retransmitted, and allocation to a shared memory would have been obvious. Both ADSL2 and VDSL1 implement convolutional interleaving functions with well-defined minimum memory requirements. Any implementation of such interleaving requires the allocation of memory, and allocation to a shared memory would have been obvious. Thus VDSL1 in view of ADSL2 renders obvious “identifying at least one packet of the plurality of packets as a packet that should be retransmitted” (ADSL2 and VDSL1 both identify messages/segments that should be retransmitted) and “allocating a memory between a retransmission function and an interleaving and/or deinterleaving function”.

385. ADSL2 uses retransmission to ensure that certain overhead messages (or segments of those messages) are received correctly. ADSL2 uses a timer and required acknowledgment or negative acknowledgment (ACK or NACK) signals to decide when to discard a segment or message being held for retransmission versus when to retransmit that segment/message.

Each transmitted message segment shall be acknowledged by the far-end with a segment acknowledge message, except the last one. The last command message segment shall be acknowledged by the far-end with the appropriate response message (identical to the non-segmented case). The last response message segment shall not be acknowledged. Table 7-17a defines the segment acknowledge message. The following shall apply to all encapsulated segment acknowledge messages:

- HDLC frame shall contain 5 message octets ($P = 5$);
- The two least significant bits of the address field shall be identical to the corresponding bits in the address field of the acknowledged message segment. All other bits of the address field shall be set to 02;
- The second least significant bit of the control field shall either indicate a command message (request to continue the response, e.g., L2 grant) or a response message (request to continue the command, e.g., OLR);
- The least significant bit of the control field shall toggle for each acknowledgement message as it normally toggles for each command/response message (see 7.8.2.4.2);
- All other bits of the control field shall be set to 02.

Table 7-17a/G.992.3 – Segment acknowledge message

Message octet number	Message octet definition
Octet 1	Message designator 1111 0000 _b for acknowledgement of high priority message segment 1111 0001 _b for acknowledgement of normal priority message segment 1111 0010 _b for acknowledgement of low priority message segment
Octet 2	Message type of segment acknowledge message 01 ₁₆
Octet 3	Acknowledged message segment ID (in range 1 to 7)
Octet 4	Message Designator (first message octet of acknowledged message segment)
Octet 5	Message Type (second message octet of acknowledged message segment)

Every message segment shall be acknowledged by the far-end before the next message segment is transmitted. The segment acknowledge message shall not be segmented.

Timeouts shall be defined as follows:

- For a non-segmented message, timeout shall apply between the last octet of the command message transmitted and the first octet of the response message received.
- For a segmented response message, the timeout shall apply between the last octet of the command message transmitted and the first octet of the first message segment received.
- A timeout corresponding to the command priority shall apply between the last octet of the message segment transmitted and the first octet of the segment acknowledge message received.
 - Another timeout corresponding to the command priority shall apply between the last octet of the segment acknowledge message transmitted and the first octet of the next message segment received.
 - If a timeout expires, the transmitter may repeat the last transmitted message. This repeated message can be an unsegmented command message, a command/response message segment (except the last segment of a response message), or a segment acknowledge message. Alternately, the ATU may abandon the message after an implementation-specific number of retransmissions.

ADSL2 at 43-46, § 7.8.2.4.3 “Overhead message segmentation.”

386. ADSL2 provides for similar retransmission for initialization messages. *See, e.g.:*

At the start of the Exchange Phase, the ATU-C shall transition to the C-TREF1-LD state (in which C-TREF symbols shall be transmitted until the first R-MSGx-LD message is received).

If the ATU-C receives an R-MSGx-LD message, the ATU-C shall transition to the C-ACK or C-NACK state within 128 symbols from the end of the R-MSGx-LD state. If the R-MSGx-LD message is successfully received, the ATU-C shall transition to the C-ACK state (in which a positive acknowledgment C-ACK message shall be transmitted). Instead, if a decoding error occurs (i.e., the CRC locally computed at the ATU-C does not correspond to the CRC transmitted by the ATU-R), the ATU-C shall transition to the C-NACK state.

The C-ACK message shall be represented by the "01010101" octet and shall be transmitted over 64 symbol periods using the same modulation technique as the loop diagnostics information bearing messages. No CRC shall be added to the C-ACK

message. In the C-NACK state, the ATU-C shall transmit 64 C-TREF symbols. Note that from the ATU-R's perspective, this is equivalent to the ATU-C not responding to the R-MSGx-LD message.

At the end of the C-ACK or C-NACK state, the ATU-C shall transition to the C-TREF2-LD state (in which 256 C-TREF symbols shall be transmitted). During the C-TREF2-LD state, the ATU-R transitions to the R-QUIET2-LD state (because the R-ACK message is successfully received and no more R-MSGx-LD messages remain to be transmitted) or the ATU-R transitions to the R-SEGUE-LD state (because no, or a corrupted C-ACK message is received or more R-MSGx-LD messages remain to be transmitted). At the end of the C-TREF2-LD state, the ATU-C shall transition to the C-SEGUE-LD state (if the ATU-R has transitioned to the R-QUIET2-LD state) or shall return to the C-TREF1-LD state (if the ATU-R has returned to the R-SEGUE-LD state).

Note that, as a result of a corrupted C-ACK message, the ATU-C could successfully receive the same message twice. In this case, the ATU-C shall ignore the second identical (same Sequence Number) message.

ADSL2 at 158-159, § 8.15.5.1.2 “Message flow, acknowledgement and retransmission.”

387. ADSL2 must save the previously transmitted packet in memory for retransmission to work. Therefore, ADSL2 allocates some memory to retransmission. Since products implementing multicarrier transmission with interleaving and retransmission such as the Intel 5116 chip implementing 802.16 and specifically products implementing ADSL transceivers such as the TI TNET5100 typically use shared memory, it would be obvious to allocate shared memory for this retransmission function. The obviousness of this choice is further illustrated by, for example, Mazzoni's and Moon's disclosures of using a shared memory for multiple error correction functions in a communications transceiver, and the corresponding memory savings that could be achieved.

388. VDSL1 describes identifying at least one packet of the plurality of packets as a packet that should be retransmitted. VDSL1 uses an initialization procedure to set up the link between the VTU-O and VTU-R, during which the transceivers exchange parameters. In this procedure, messages are sent over the Special Operation Channel (SOC) and retransmission is utilized so that VDSL1 identifies at least one packet of the plurality of packets as a packet that

should be retransmitted. This was already mentioned in my discussion of the preamble, and that discussion is included here by reference.

389. Additionally, Section 12.4.1 provides an overview of this initialization:

12.4.1 Overview

Initialization of a VTU-O/VTU-R pair includes a variety of tasks. The set of tasks consist of:

- Definition of a common mode of operation;
- Synchronization (sample clock alignment and symbol alignment);
- Transfer of frequency band allocation and PSD mask information from the VTU-O to the VTU-R;
- Channel identification;
- Noise identification;
- Calculation of bit loading and energy tables;
- Exchange of parameters (RS settings, interleaver parameters, VOC settings, bit loading and energy tables, ...).

VDSL1 at NOK00078832, § 12.4.1 “Overview.”

390. VDSL1 uses a repeat request (RQ) which we have referred to as a NACK above, when operating in RQ mode to request a retransmission on the SOC. This is described in Section 12.4.2.1 as excerpted below.

12.4.2 SOC protocol

12.4.2.1 Message format

The SOC shall use HDLC-like format with byte stuffing to delineate the messages as specified in ITU-T Rec. G.994.1. A reliable transmission shall be insured by using either an automatic repeat (AR) mode or a repeat request (RQ) mode. The maximum size of an HDLC frame shall be 1026 bytes (this also defines the maximum size of a SOC message segment). This is the size of the payload before octet stuffing and addition of any flags.

In the AR mode, the message encapsulated in the HDLC frame shall be automatically repeated. At least four idle flags (0x7E) shall be inserted between successive frames.

In the RQ mode, the messages encapsulated in HDLC frame shall be sent once. However, the VTU expecting the message shall have the possibility to request the remote side to repeat it by sending a REPEAT REQUEST message. This operation is necessary when the expected message has a wrong Check Sequence or when a timeout has expired. After two unsuccessful REPEAT REQUEST, the initialization shall be aborted. This means the initiating side will restart the G.994.1 handshake after a silent period. After a number of unsuccessful attempts, the modems shall stop all attempts. The number of attempts that is made before a final abort of the initialization shall be chosen by the initiating modem.

VDSL1 at NOK00078833, § 12.4.2.1 “Message format.”

391. VDSL1 must save the previously transmitted packet in memory for retransmission to work. Therefore, VDSL1 allocates some memory to retransmission. Since products implementing multicarrier transmission with interleaving and retransmission such as the Intel 5116 chip implementing 802.16 and specifically products implementing DSL transceivers such as the TI TNET5100 typically use shared memory, it would be obvious to allocate shared memory for this retransmission function. The obviousness of this choice is further illustrated by, for example, Mazzoni's and Moon's disclosures of using a shared memory for multiple error correction functions in a communications transceiver, and the corresponding memory savings that could be achieved.

392. Thus both ADSL2 and VDSL1 disclose retransmitting messages to ensure they are received correctly, which requires identifying the message or segment that must be retransmitted, and allocating shared memory to support these functions would have been obvious to a POSITA. As already described in § VII.A.4 above, retransmission was well-known and routinely implemented in the prior art to ensure that packets are correctly received. For example, retransmission in the form of automatic repeat request (that is, "ARQ") was in use before 1960 and has been a standard tool in communications for years.

It is difficult to date the earliest introduction of ARQ protocols in digital communications. They were almost certainly an immediate consequence of the development of parity-check codes in the early days of digital computers. The next major development in the area came in 1960, when Wozencraft and Horstein [Woz3],[Woz4] described and analyzed a system that allowed for both error correction and error detection with retransmission requests. Their system, now known as a type-I hybrid-AREQ protocol, provided significantly improved performance over the pure ARQ protocols in many applications.

Stephen Wicker, Error Control Systems for Digital Communication and Storage at page 393.

393. Thus, to the extent that the use in ADSL2 and VDSL1 of retransmission does not disclose or render obvious allocating a memory to a retransmission function, such that (when combined with the allocation of memory to an interleaving and/or deinterleaving function in both

ADSL2 and VDSL1, as discussed below) ADSL2 and VDSL1 allocate a memory between a retransmission function and an interleaving and/or deinterleaving function, a POSITA would have been able to incorporate retransmission using a shared memory into VDSL1 or ADSL2 using standard approaches with predictable results.

394. Finally, I note that, as specifically pointed out in § VII.A.4.a, the TCP/IP standard (which is the protocol used for Internet traffic) uses retransmission. Specifically, the transport control protocol (“TCP”) includes a retransmission mechanism. Both VDSL1 and ADSL2 were widely used by consumers to access the Internet. Thus, a POSITA would have recognized that the shared memory used to support data transmission in VDSL1 and ADSL2 was used with a retransmission protocol when VDSL1 and ADSL2 were used to for Internet access. § VII.A.4.a *supra*.

395. ADSL2 implements an interleaving function with a well-defined minimum memory requirement, specifically in the PMS-TC layer. The PMS-TC layer uses an interleaver, specifically a convolutional interleaver. This is described in Section 7.7.1.5, which is excerpted below. Note that different interleaver depths D_p and different FEC Codeword sizes will require different minimum amounts of shared memory to implement.

7.7.1.5 Interleaver

To spread the Reed-Solomon codeword and, therefore, reduce the probability of failure of the FEC in the presence of impulse noise, the FEC Output Data Frames shall be convolutionally interleaved. The interleaver creates the Interleaved FEC Output Data Frames at Reference point C, at the output of the latency path function. This procedure is followed by the frame multiplexing procedure.

Convolutional interleaving is defined by the rule (using the currently defined values of the framing control parameters D_p and the derived parameter N_{FECp}):

Each of the N_{FECp} octets $B_0, B_1, \dots, B_{N_{FECp}-1}$ in an FEC Output Data Frame is delayed by an amount that varies linearly with the octet index. More precisely, octet B_i (with index i) is delayed by $(D_p - 1) \times i$ octets, where D_p is the interleaver depth.

An example for $N_{FECp} = 5$, $D_p = 2$ is shown in Table 7-13, where B_i^j denotes the i -th octet of the j -th FEC Output Data Frame.

Table 7-13/G.992.3 – Convolutional interleaving example for $N_{FECp} = 5$, $D_p = 2$

Interleaver input	B_0^j	B_1^j	B_2^j	B_3^j	B_4^j	B_0^{j+1}	B_1^{j+1}	B_2^{j+1}	B_3^{j+1}	B_4^{j+1}
Interleaver output	B_0^j	B_3^{j-1}	B_1^j	B_4^{j-1}	B_2^j	B_0^{j+1}	B_3^j	B_1^{j+1}	B_4^j	B_2^{j+1}

With the above-defined rule, the output octets from the interleaver always occupy distinct time slots when N_{FECp} is odd and D_p is a power of 2. When N_{FECp} is even, a dummy octet shall be added at the beginning of the codeword at the input to the interleaver. The resultant odd-length codeword is then convolutionally interleaved, and the dummy octet shall then be removed from the output of the interleaver.

The interleaving procedure of the latency path function # p shall interleave a single FEC Output Data Frame, or $M_p \times K_p + R_p$ octets. This procedure is followed by the Frame Multiplexing procedure.

ADSL2 at 38, § 7.7.1.5 “Interleaver.”

396. Figure C.7-1 shows the PMS-TC function in ADSL2 including the interleaving function.

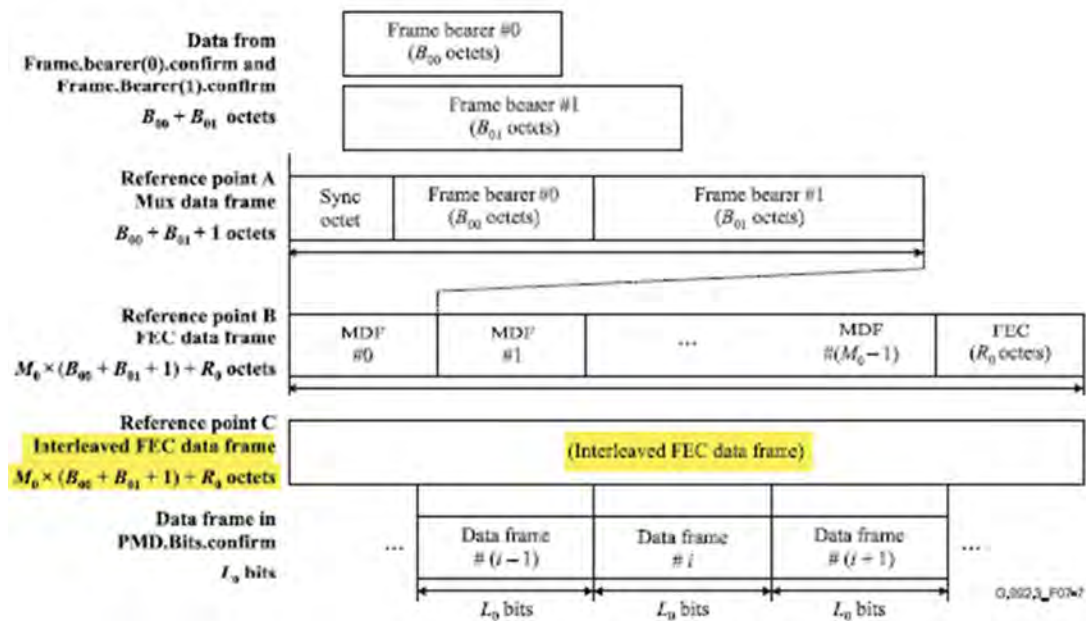


Figure 7-7/G.992.3 – Illustration of frame structure with single latency dual bearers and $T_p = 1$

ADLS2 at 29, Figure 7-7.

398. Figure 7-8 shows the same information, but where two bearers are being used over two latency paths using two different interleaver functions.

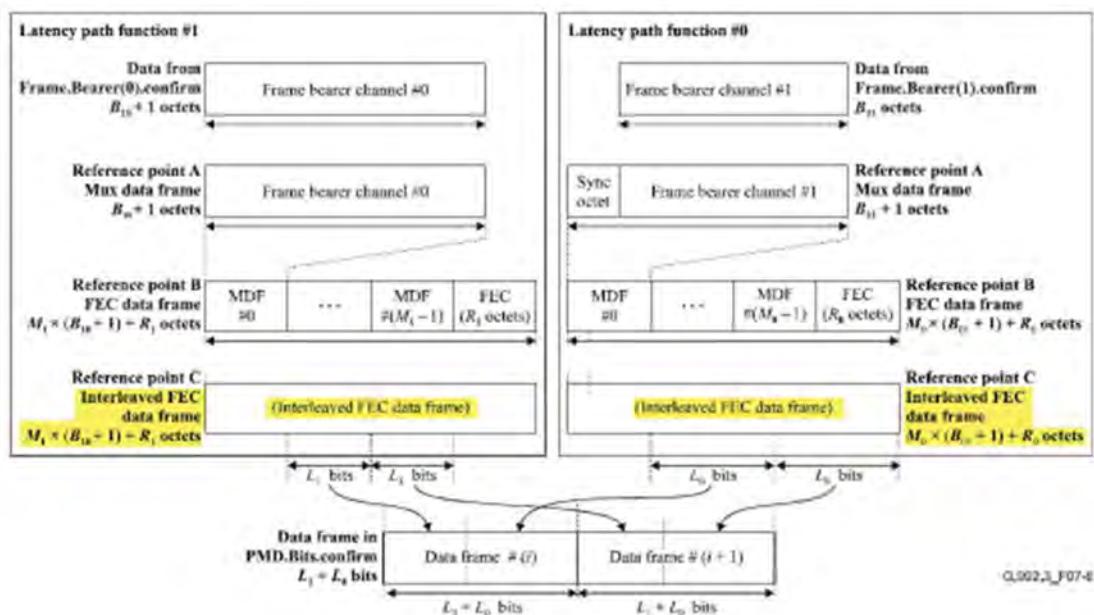


Figure 7-8/G.992.3 – Illustration of frame structure with dual latency and dual bearers

ADLS2 at 30, Figure 7-8.

399. ADSL2 uses a convolutional interleaver to protect data from impulse noise and describes the operation of the convolutional interleaver (and, consequently, the deinterleaver) in detail.

To spread the Reed-Solomon codeword and, therefore, reduce the probability of failure of the FEC in the presence of impulse noise, the FEC Output Data Frames shall be convolutionally interleaved. The interleaver creates the Interleaved FEC Output Data Frames at Reference point C, at the output of the latency path function. This procedure is followed by the frame multiplexing procedure.

Convolutional interleaving is defined by the rule (using the currently defined values of the framing control parameters D_p and the derived parameter $N_{FEC,p}$):

Each of the $N_{FEC,p}$ octets $B_0, B_1, \dots, B_{N_{FEC,p}-1}$ in an FEC Output Data Frame is delayed by an amount that varies linearly with the octet index. More precisely, octet B_i (with index i) is delayed by $(D_p - 1) \times i$ octets, where D_p is the interleaver depth.

An example for $N_{FEC,p} = 5, D_p = 2$ is shown in Table 7-13, where B_i^j denotes the i -th octet of the j -th FEC Output Data Frame.

Table 7-13/G.992.3 – Convolutional interleaving example for $N_{FEC,p} = 5, D_p = 2$

Interleaver input	B_0^j	B_1^j	B_2^j	B_3^j	B_4^j	B_0^{j+1}	B_1^{j+1}	B_2^{j+1}	B_3^{j+1}	B_4^{j+1}
Interleaver output	B_0^j	B_3^{j-1}	B_1^j	B_4^{j-1}	B_2^j	B_0^{j+1}	B_3^j	B_1^{j+1}	B_4^j	B_2^{j+1}

With the above-defined rule, the output octets from the interleaver always occupy distinct time slots when $N_{FEC,p}$ is odd and D_p is a power of 2. When $N_{FEC,p}$ is even, a dummy octet shall be added at the beginning of the codeword at the input to the interleaver. The resultant odd-length codeword is then convolutionally interleaved, and the dummy octet shall then be removed from the output of the interleaver.

The interleaving procedure of the latency path function #p shall interleave a single FEC Output Data Frame, or $M_p \times K_p + R_p$ octets. This procedure is followed by the Frame Multiplexing procedure.

ADSL2 at 38, § 7.7.1.5 “Interleaver.”

400. VDSL1 describes transceivers that use memory to support interleaving (or deinterleaving) functions. The PMS-TC sublayer of VDSL1 interleaves data being transmitted in the “slow payload,” which is mandatory. The PMS-TC layer in the VDSL1 standard takes data from one of two payloads, the slow and fast payload, and (in the case of the mandatory slow payload) scrambles, applies FEC to, and interleaves that data. Fast payload data is not interleaved.

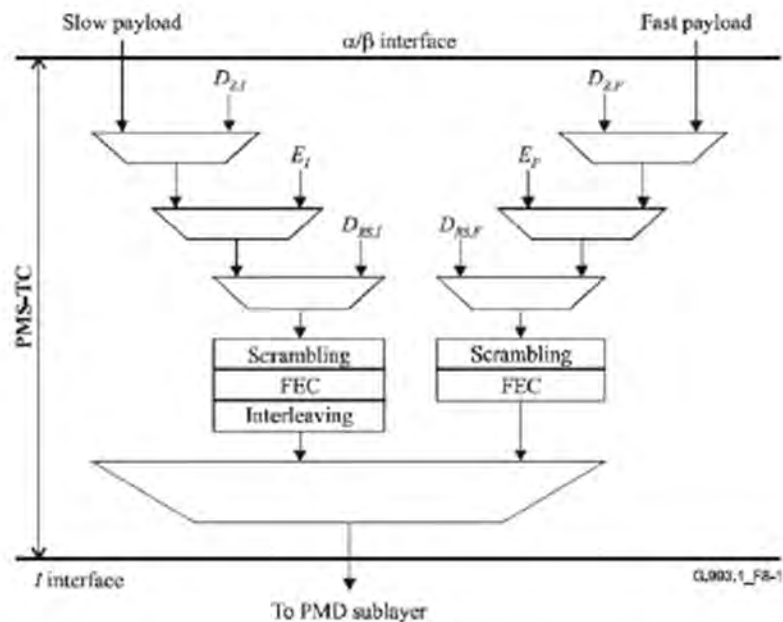


Figure 8-1/G.993.1 – Diagram of PMS-TC sublayer

VDSL1 at NOK00078784, Figure 8-1. *See also:*

8 PMS-TC sublayer

The PMS-TC sublayer provides transmission medium specific TC functions, such as framing, Forward Error Correction (FEC), and interleaving.

8.1 PMS-TC functional model

All data bytes shall be transmitted MSB first. All serial processing (*e.g.*, scrambling, CRC calculation) shall however be performed LSB first, with the outside world MSB considered as the YDSL LSB. As a result, the first incoming bit (outside world MSB) shall be the first bit processed inside VDSL (VDSL LSB). The PMS-TC functional diagram is presented in Figure 8-1.

VDSL1 at NOK00078783, §§ 8-8.1 “PMS-TC sublayer” and “PMS-TC functional model.”

401. The PMS-TC sublayer interleaving is described in detail in Section 8.4:

8.4 Interleaving

8.4.1 General

Interleaving shall be used to protect the data against bursts of errors by spreading the errors over a number of Reed-Solomon codewords. The interleave depth shall be programmable with a maximum interleave depth of 64 codewords when the number

of octets per codeword (N) equals 255. For smaller values of N the interleave depth can grow nearly proportionately.

It shall be possible to adjust the interleave depth via the management system to meet latency requirements. The latency of the slow path is a function of the data rate and burst error correction capability. For data rates greater than or equal to 13 Mbit/s the latency between the a and 13 interfaces shall not exceed 10 Ms when the interleaver depth is set to the maximum. At lower data rates there is a trade-off between higher latency and decreased burst error correction ability. At any data rate, the minimum latency occurs when the interleaver is turned off.

When the interleaver is on the codewords shall be interleaved before transmission to increase the immunity of RS codewords to bursts of errors. The convolutional interleaver is defined by two parameters: the interleaver block length, I , and the interleaving depth D . The block length I shall divide the RS codeword length N (i.e., N shall be an integer multiple of I). The convolutional interleaver uses a memory in which a block of I octets is written while an (interleaved) block of I octets is read. Details of the implementation are given in 8.4.2.

The same size interleaving memory (see Table 8-1) is needed for interleaving at the transmitter and de-interleaving at the receiver.

The convolutional interleaving introduces an absolute read-to-write delay, Δ_j , that increments linearly with the octet index within a block of I octet :

$$\Delta_j \equiv (D - 1) \times j$$

where $j = 0, 1, 2, \dots I - 1$.

VDSL1 at NOK00078785, § 8.4.1 “Interleaving – General”.

402. Section 8.4.2 gives a detailed description of the typical “triangular” implementation of a convolutional interleaver. This implementation shows how the interleaver/deinterleaver memory size for this typical implementation can be computed from the two parameters M and I as $M \times I \times (I - 1)/2$ bytes. In Figure 8-2 below, the interleaver memory is the left triangle of blocks of M -octet memories used as delay elements. The deinterleaver memory is the same size and is the inverted triangle on the right, again constructed of M -octet memories used as delay elements.

8.4.2 Triangular Implementation

To decrease the implementation complexity, the delay increment ($D-1$) shall be chosen as a multiple of the interleaver block length (I), i.e.: $D-1 = M \times I$. The $(D-1)$ to I ratio is the interleaving depth parameter (M). The characteristics of convolutional interleaving are shown in Table 8-1. The parameters t and q depend on the characteristics of the RS code and are defined as:

- t = number of bytes that can be corrected by RS codewords = half the number of redundancy bytes = $R/2$;
- q = length of RS codeword divided by the length of an interleaver block = N/I .

VDSL1 at TQD-ADT 066670, § 8.4.2 “Triangular implementation.”

Table 8-1/G.993.1 – Characteristics of convolutional interleaving

Parameter	Value
Interleaver block length (I)	I bytes (equal to or divisor of N)
Interleaving Depth (D)	$M \times I + 1$
(De)interleaver memory size	$M \times I \times (I-1)/2$ bytes
Correction capability	$\lfloor t/q \rfloor \times (M \times I + 1)$ bytes
End-to-end delay	$M \times I \times (I-1)$ bytes

The example in Figure 8-2 shows $I = 7$. I parallel branches (numbered $0 \dots I-1$) are implemented with a delay increment of M octets per branch. Each branch shall be a FIFO shift register (delay line) with length $0 \times M \dots (I-1) \times M$ bytes. The deinterleaver is similar to the interleaver, but the branch indices are reversed so that the largest interleaver delay corresponds to the smallest deinterleaver delay. Deinterleaver synchronization shall be achieved by routing the first byte of an interleaved block of I bytes into branch 0.

VDSL1 at NOK00078785, § 8.4.2 Triangular implementation.

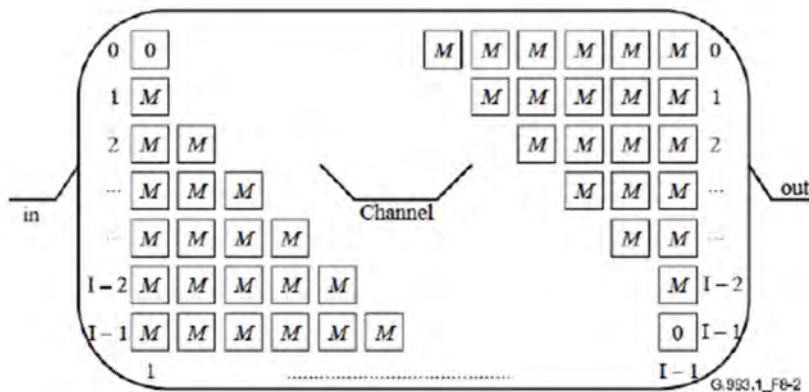


Figure 8-2/G.993.1 – Implementation example with $D-1 = M \times I$ and $I = 7$

VDSL1 at NOK00078785, § 8.4.2 -Triangular implementation.

403. Table 8-2 below shows the minimum interleaver/deinterleaver memory sizes that correspond to various values of M and I using the equation $M \times I \times (I - 1)/2$ bytes to compute the interleaver/deinterleaver memory sizes. Note that the interleaver and deinterleaver have the same minimum memory requirement.

Table 8-2/G.993.1 – Example of interleaver parameters with RS(144,128)

Rate [kbit/s]	Interleaver parameters	Interleaver depth	(De)interleaver memory size	Erasure correction	End-to-end delay
50×1024	$I = 72$ $M = 13$	937 blocks of 72 bytes	33 228 bytes	3 748 bytes 520 μ s	9.23 ms
24×1024	$I = 36$ $M = 24$	865 blocks of 36 bytes	15 120 bytes	1 730 bytes 500 μ s	8.75 ms
12×1024	$I = 36$ $M = 12$	433 blocks of 36 bytes	7 560 bytes	866 bytes 501 μ s	8.75 ms
6×1024	$I = 18$ $M = 24$	433 blocks of 18 bytes	3 672 bytes	433 bytes 501 μ s	8.5 ms
4×1024	$I = 18$ $M = 16$	289 blocks of 18 bytes	2 448 bytes	289 bytes 501 μ s	8.5 ms
2×1024	$I = 18$ $M = 8$	145 blocks of 18 bytes	1 224 bytes	145 bytes 503 μ s	8.5 ms

VDSL1 at NOK00078786, Table 8-2.

404. Numerous interleaver parameters were required to be supported by the VDSL1 standard.

The following interleaver parameters shall be supported:

- For $(N,K) = (144,128)$ the following values for M and I shall be supported:
 $I = 36$ and M between 2 and 52.
- For $(N,K) = (240,224)$ the following values for M and I shall be supported:
 $I = 30$ and M between 2 and 62.

VDSL1 at NOK00078785, § 8.4.2.

405. Thus, both ADSL2 and VDSL1 disclose implementing convolutional interleaving, which has a well-defined minimum memory requirement for the well-known triangular implementation. In order to implement this convolutional interleaving at all, any implementation would have been required to allocate memory to this function. Since products implementing

multicarrier transmission with interleaving and retransmission such as the Intel 5116 chip implementing 802.16 and specifically products implementing DSL transceivers such as the TI TNET5100 typically use shared memory, it would be obvious to allocate shared memory for this interleaving function. The obviousness of this choice is further illustrated by, for example, Mazzoni's and Moon's disclosures of using a shared memory for multiple error correction functions, including specifically interleaving and/or deinterleaving, in a communications transceiver, and the corresponding memory savings that could be achieved.

406. The VDSL1 standard requires support of numerous values of M and I as described at the end of Section 8.4.2 (shown below). Thus the VTU-R and VTU-O must agree as to the particular values of M and I that will be used. This is accomplished through configuration messages including O-MSG2 and R-MSG2 described below.

Table 12-23/G.993.1 – Description of O-MSG2

Field content	Field or macro-field type	Remark
Message descriptor	1 byte	See Table 12-12.
Minimal SNR margin	1 byte	In units of 0.5 dB
Maximal constellation size in downstream ($B_{max,d}$)	1 byte	Maximum number of bits per tone
RS setting supported by VTU-O	1 byte	0x00: Only mandatory settings 0xFF: All settings
Interleaver settings supported by the VTU-O	1 byte	0x00: Only mandatory settings 0xFF: All settings 0xNN NN = Number of additional (i.e., non-mandatory) settings in hexadecimal (NN $\neq$ 0x00 and NN $\neq$ 0xFF)
Detailed interleaver setting description	0 byte if NN = 0x00 or NN = 0xFF, 4xNN otherwise	Interleaver descriptor (see Table 12-24)
Maximal power in downstream	1 byte (unsigned)	In units of 0.25 dBm
Maximum interleaver delay	1 byte	In ms by steps of 0.5 ms (Note 1)
Maximum number of EOC bytes per frame in downstream	1 byte	Number of EOC bytes per frame
Maximum number of VOC bytes per frame in downstream	1 byte	Number of VOC bytes per frame

Table 12-23/G.993.1 – Description of O-MSG2

Field content	Field or macro-field type	Remark
Support of express bit swapping	1 byte	0x00: Not supported 0xFF: Supported
j_{max}	1 byte	Maximum value of j_{max} supported by the VTU-O (Note 2)
NOTE 1 – This field can be set to 0 in order to emulate the fast channel. This field is used for the creation of R-CONTRACT1 even if dual latency is used at the end.		
NOTE 2 – Specification of $j_{max} = k$ means that all values 0, 1, ..., k are supported.		

j_{max} is defined in 12.4.6.2.1.3.

VDSL1 at NOK00078848-849, Table 12-23. *See also id.* Table 12-24.

407. Accordingly, VDSL1 in view of ADSL2 renders obvious “identifying at least one packet of the plurality of packets as a packet that should be retransmitted” (ADSL2 identifies overhead or initialization messages/segments that should be retransmitted and VDSL1 identifies initialization messages/segments that should be retransmitted), “allocating a memory between a retransmission function” (ADSL2 and VDSL1 describe using a packet retransmission function, and a POSITA would have found it obvious to use both VDSL1 and ADSL2 with retransmission to protect non-overhead data, including as described in the TCP/IP protocol) “and an interleaving and/or deinterleaving function” (VDSL1 and ADSL2 both describe using interleaving functions to protect and correct transmitted data from errors, and these interleaving functions require a well-defined minimum amount of memory for each possible configuration and the multiple configurations as well as standard industry practice make shared memory the most obvious implementation approach).

ii TR 25.896 and TS 25.401 in View of Eriksson

408. TR 25.896 and TS 25.401 in view of Eriksson render obvious “identifying at least one packet of the plurality of packets as a packet that should be retransmitted and allocating a memory between a retransmission function and an interleaving and/or deinterleaving function.” Specifically, TR 25.896 and Eriksson both disclose a transceiver that uses hybrid ARQ

retransmission and therefore identify errored packets which should be retransmitted. In addition, both TR 25.896 and Eriksson disclose allocating a memory between a retransmission function and an interleaving and/or deinterleaving function (the UE stores packets for later retransmission packets if requested, while the Node B stores received packets for eventual soft combining, and the UE interleaves packets before transmitting them while the Node B deinterleaves received packets).

409. TR 25.896 describes that hybrid ARQ is used to identify erroneously received (that is, errored) data packets, which should be retransmitted.

Node B controlled hybrid ARQ allows for rapid retransmissions of erroneously received data units, thus reducing the number of RLC retransmissions and the associated delays. This can improve the quality of service experienced by the end user. As a Node B controlled retransmission is less costly from a delay perspective, the physical channel can be operated with somewhat higher error probability than in Rel 5, which may result in improved system capacity. The retransmission probability for the initial transmission is preferably in the order of 10-20% when evaluating hybrid ARQ as closed loop power control is used for the uplink, maintaining a given quality level. Significantly higher retransmission probabilities may lead to considerably reduced end user throughput, while at very small retransmission probabilities the Node B controlled hybrid ARQ will not provide any additional gains compared to R99/4/5. Soft combining can further improve the performance of a Node B controlled hybrid ARQ mechanism.

TR 25.896 at NOK00074951-952, § 7.2.1 “Hybrid ARQ – General”.

410. Eriksson similarly describes that hybrid ARQ is used to identify packets that cannot be decoded, which should be retransmitted.

Another proposed hybrid ARQ Scheme, sometimes referred to as incremental redundancy or type-II hybrid ARQ, provides for additional redundant bits to be transmitted if the originally transmitted block cannot be decoded. This scheme is conceptually illustrated in FIG. 3. Therein, three decoding attempts are made by the receiver. First, the receiver attempts to decode the originally received data block (with or without redundancy). Upon failure, the receiver then receives additional redundant bits R1, which it uses in conjunction with the originally transmitted data block to attempt decoding. As a third step, the receiver obtains another block of redundant information R2, which it uses in conjunction with the originally received data block and the block of redundant bits R1 to attempt decoding for a third time. This process can be repeated until successful decoding is achieved.

As compared with link adaptation, incremental redundancy does not require that link quality estimates be transmitted or used. However, one problem with this technique is the large memory requirement associated with storing the data block (and possibly additional blocks of redundant bits) until a successful decode occurs, which storage is needed since the subsequently transmitted redundancy blocks (e.g., R1 and R2) cannot be independently decoded to give the same performance as if combined decoding was used. The storage requirements are further increased if the receiver stores a multi-bit soft value associated with each received bit, the soft values indicating a confidence level associated with the decoding of the received bit.

Eriksson at NOK00083211, 3:23-38.

411. TR 25.896 Node Bs (base stations) and UEs (user equipment) are communications transceivers and use an uplink hybrid ARQ scheme. This is described in section 7.2 “Hybrid ARQ”:

Node B controlled hybrid ARQ allows for rapid retransmissions of erroneously received data units, thus reducing the number of RLC retransmissions and the associated delays. This can improve the quality of service experienced by the end user. As a Node B controlled retransmission is less costly from a delay perspective, the physical channel can be operated with somewhat higher error probability than in Rel 5, which may result in improved system capacity. The retransmission probability for the initial transmission is preferably in the order of 10-20% when evaluating hybrid ARQ as closed loop power control is used for the uplink, maintaining a given quality level. Significantly higher retransmission probabilities may lead to considerably reduced end user throughput, while at very small retransmission probabilities the Node B controlled hybrid ARQ will not provide any additional gains compared to R99/4/5. Soft combining can further improve the performance of a Node B controlled hybrid ARQ mechanism.

TR 25.896 at NOK00074951-952, § 7.2.1 “Hybrid ARQ – General”.

412. TR 25.896 further describes using multiple redundancy versions with hybrid ARQ:

Redundancy version. If multiple redundancy versions are supported, the redundancy version needs to be known to the Node B. The potential gains with explicit support of multiple redundancy versions should be carefully weighted against the increase in overhead due to the required signaling. Note that, unlike the HS-DSCH, the number of users simultaneously transmitting data in the uplink using hybrid ARQ may be significant.

TR 25.896 at NOK00074953, § 7.2.3 “Associated Signalling”.

413. TR 25.896 interleaves packets sent using hybrid ARQ twice, as shown in Figure 8.4.7:

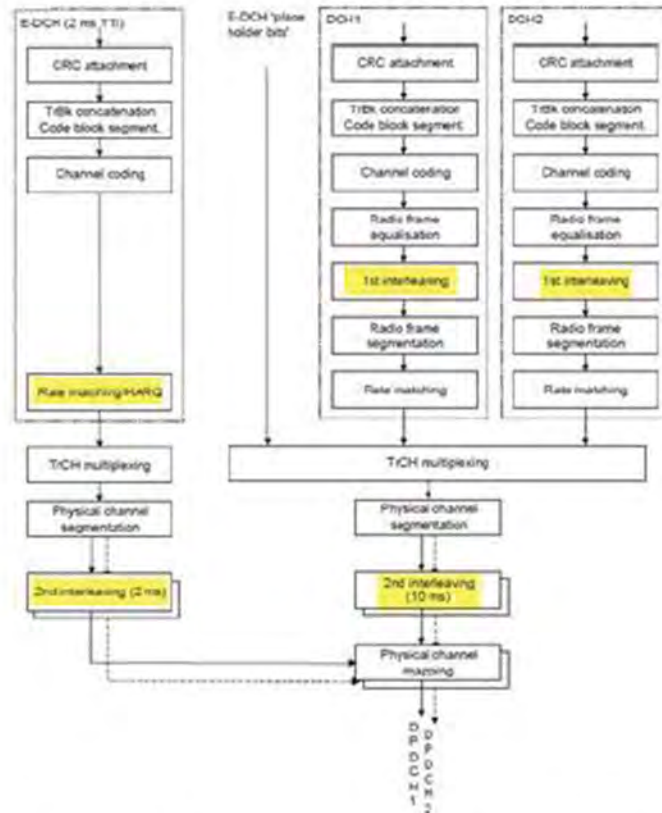


Figure 8.4.7. Time multiplexing of E-DCH (2ms TTI) and DCH (10ms TTI) with place holder bits.

TR 25.896 at NOK00074969, Figure 8.4.7. The first and second interleaving performed in the Release 6 system are described in detail in TS 25.212 §§ 4.2.5 and 4.2.11. *See §§ IX.D-E supra.* The minimum amount of memory required to support the two interleaving steps depends on a number of factors including the interleaver depth and codeword size. Once these parameters are known, the minimum amount of memory needed to support interleaving can be deduced so the appropriate amount of memory is reserved. §§ IX.D-E.

414. TR 25.896 further describes that a portion of the shared memory is allocated to the hybrid ARQ retransmission function. The minimum amount of memory needed to support hybrid ARQ retransmission depends on several factors. TR 25.896 describes that the Node B uses a buffer that a POSITA would understand to be within a shared memory to store the received data for which a retransmission has been or will be requested using hybrid ARQ, thereby using a shared memory

for both interleaving and retransmission. The minimum amount of memory needed to support the hybrid ARQ retransmission function at the Node B depends on the number of hybrid ARQ processes and the number of symbols transmitted per TTI.

The design of an uplink hybrid ARQ scheme should take the following aspects into account:

- Memory requirements, both in the UE and the Node B. Rapid retransmissions reduce the amount of buffer memory required in the Node B for buffering of soft bits when a retransmission has been requested.

TR 25.896 at NOK00074951-952, § 7.2.1 “Hybrid ARQ – General”.

415. Specifically, TR 25.896 describes that either the Node B or the UE will determine parameters necessary to use hybrid ARQ and that the end that chooses the parameters (either Node B or UE) must signal the other end with the selected parameters. No matter which end (Node B or UE) decides the parameters used, both the Node B and UE must set aside a sufficient portion of the shared memory (referred in the 3GPP standards as a “buffer”) to support the interleaving applied to messages also subject to hybrid ARQ retransmission. The Node B must set aside data based on the number of retransmissions allowed and the number of hybrid ARQ processes used, which further shows that the Node B and UE must exchange a message setting these parameters; otherwise, the Node B will not be able to estimate the amount of memory to set aside to support hybrid ARQ retransmission.

The principle of hybrid ARQ is to buffer E-DCH TTIs that were not received correctly and consequently combine the buffered data with retransmissions. The actual method of doing soft combining depends on the HARQ combining scheme selected. In Chase combining scheme the receiver always combines the full retransmission of the failed TTI, i.e. the amount of data in the receiver buffer remains the same. In the incremental redundancy schemes the receiver buffers coded symbols, which introduce new information to the E-DCH TTI transmitted first, i.e. the amount of data to be buffered increases with consecutive retransmissions. However, in practice the buffer in the receiver needs to be dimensioned considering the maximum number of coded bits of the E-DCH TTI after all the incremental redundancy has been introduced. Regardless of the HARQ combining scheme soft combining is done on L1 of Node B before the decoding stage of FEC. Prior to

decoding these symbols are soft-valued, i.e. each symbol is represented by two or more bits. Here we call them soft symbols.

For N-process SAW HARQ, the number of soft symbols to be buffered in L1 receiver can be estimated generally as follows, since no new PDUs are transmitted on a subchannel before the previous packet is acknowledged. The receiver has to buffer one E-DCH TTI for each HARQ process and for each UE. The next transmission is either a new packet or a retransmission of an erroneous packet. In either case, the maximum buffering need is N E-DCH TTIs. The actual size of the buffer needed for each E-DCH TTI depends on the HARQ combining scheme as described above. Thus, for N-process stop-and-wait ARQ the L1 buffering at Node B for one UE can be expressed as:

$$buffer = (softsymbols_{TTI} \times N)$$

Table 9.2.4 shows some examples for the soft buffer size required at Node B per UE. It should be noted that soft buffer is needed for all UEs waiting for retransmission at a given time. Thus these values have to be multiplied by the number of active UEs using E-DCH. Depending on particular implementations, the buffer may be shared between UEs and thus the total size of soft buffer can be reduced given that not all active UEs transmit simultaneously using the most demanding format. Also, it may be possible to re-use some existing buffers.

Table 9.2.4 Node B soft buffer size (kilo soft symbols) per UE, for some example values of N (number of HARQ processes) and TTI lengths

	N=4, TTI=10 ms	N=3, TTI=10 ms	N=8, TTI=2 ms	N=6, TTI=2 ms
BPSK, SF=4	38.4	28.8	15.36	11.52
QPSK or 2*BPSK, SF=4	76.8	57.6	30.72	23.04
8PSK or 3*BPSK, SF=4	115.2	86.4	46.08	34.56
2*QPSK or 4*BPSK, SF=4; or QPSK, SF=2	153.6	115.2	61.44	46.08

TR 25.896 at NOK00074991, § 9.2.2.1.1 “Soft buffer at Node B”. Either the RNC or the Node B must also set aside memory for a reordering buffer, which is used to correct the order of packets when they are received out of order due to retransmission. This number is dependent on the number of hybrid ARQ processes, the maximum number of retransmissions allowed (assumed 3 in the below example), and the amount of data sent in a single TTI (essentially, the size of packets being transmitted). Again, this shows that the Node B and UE must exchange the number of hybrid ARQ processes and the number of retransmissions allowed.

Due to physical/MAC layer (H)ARQ, the MAC-e PDUs may be received in wrong order. Therefore, reordering of the MAC-e PDUs is required either in the Node B or

in the RNC. Here the required reordering buffer size per UE is estimated. In the estimation it is assumed that only one high bit rate reordering queue is active for a UE at a time. Furthermore, maximum of 3 retransmissions per MAC-e PDU is assumed and the buffer size is calculated for the worst case where one ARQ process requires retransmissions while the others get blocks through and forward them to reordering queue. The worst case buffer size is then calculated as

$$\text{reorderingbuffer} = [3(N - 1) + 1] \times (\text{bits}/\text{TTI})$$

These values have to be multiplied by the number of E-DCH UEs served by the Node B or by the RNC, depending on the location of the reordering buffer. Depending on implementation and location of the re-ordering buffer, it may be possible to re-use some existing buffers.

Table 9.2.5 Reordering buffer size (kB) per UE, for some example values of N (number of HARQ processes) and TTI lengths

User data rate	N=4, TTI=10 ms	N=3, TTI=10 ms	N=8, TTI=2 ms	N=6, TTI=2 ms
384 kbit/s	4.8	3.36	2.1	1.5
768 kbit/s	9.6	6.72	4.2	3.1
1 Mbit/s	12.5	8.8	5.5	4.0
1.5 Mbit/s	18.8	13.1	8.3	6.0

TR 25.896 at NOK00074991, § 9.2.2.1.2 “Reordering buffer in radio network” and Table 9.2.5.

Both interleaving steps applied to the data will also require a reordering buffer to deinterleave the interleaved data and put the interleaved packets/data in the correct order; as with the hybrid ARQ reordering buffer, the minimum size of the [de]interleaving buffer depends on the amount of data transmitted in a packet and the interleaving depth (which is similar to N, the number of times a packet can be retransmitted).

416. The UE must set aside a sufficient portion of the shared memory (“buffer” in the 3GPP standards) to support the interleaving and retransmission used by hybrid ARQ with incremental redundancy. The minimum amount of memory the UE must set aside to support uplink hybrid ARQ with incremental redundancy depends on the amount of information transmitted in each TTI.

In addition to buffers in the receiver side, also some buffer at UE side is needed. The retransmission buffer can be either at MAC layer as info bits or at physical layer as coded bits. The total number of coded bits is typically 3 times the number of info bits. Table 9.2.4 shows also the required retransmission buffer size in kbits

(assuming that coded bits are buffered). Depending on implementation it may be possible to re-use existing buffers.

TR 25.896 at NOK00074992, § 9.2.2.1.3 “Retransmission buffer in UE.”

417. Eriksson also describes that Node Bs and UEs (communications transceivers with a shared memory) at both ends of a cellular communication system use hybrid ARQ and incremental redundancy to correct errors:

Another proposed hybrid ARQ Scheme, sometimes referred to as incremental redundancy or type-II hybrid ARQ, provides for additional redundant bits to be transmitted if the originally transmitted block cannot be decoded. This scheme is conceptually illustrated in FIG. 3. Therein, three decoding attempts are made by the receiver. First, the receiver attempts to decode the originally received data block (with or without redundancy). Upon failure, the receiver then receives additional redundant bits R1, which it uses in conjunction with the originally transmitted data block to attempt decoding. As a third step, the receiver obtains another block of redundant information R2, which it uses in conjunction with the originally received data block and the block of redundant bits R1 to attempt decoding for a third time. This process can be repeated until successful decoding is achieved.

As compared with link adaptation, incremental redundancy does not require that link quality estimates be transmitted or used. However, one problem with this technique is the large memory requirement associated with storing the data block (and possibly additional blocks of redundant bits) until a successful decode occurs, which storage is needed since the subsequently transmitted redundancy blocks (e.g., R1 and R2) cannot be independently decoded to give the same performance as if combined decoding was used. The storage requirements are further increased if the receiver stores a multi-bit soft value associated with each received bit, the soft values indicating a confidence level associated with the decoding of the received bit.

Eriksson at NOK00083211, 3:23-38. Eriksson notes that limited memory is a problem in hybrid ARQ systems.

As compared with link adaptation, incremental redundancy does not require that link quality estimates be transmitted or used. However, one problem with this technique is the large memory requirement associated with storing the data block (and possibly additional blocks of redundant bits) until a successful decode occurs, which storage is needed since the subsequently transmitted redundancy blocks (e.g., R1 and R2) cannot be independently decoded to give the same performance as if combined decoding was used. The storage requirements are further increased if the receiver stores a multi-bit soft value associated with each received bit, the soft values indicating a confidence level associated with the decoding of the received bit.

Eriksson at NOK00083211at 3:39-52. Combined ARQ and interleaving also contributes to the problem of limited memory in Eriksson, which notes that ARQ and FEC such as interleaving have competing advantages and disadvantages:

In addition to modulation schemes, digital communication systems also employ various techniques to handle erroneously received information, which techniques also affect the bit rate experienced by the user. Generally speaking, these techniques include those which aid a receiver to correct the erroneously received information, e.g., forward error correction (FEC) techniques, and those which enable the erroneously received information to be retransmitted to the receiver, e.g., automatic retransmission request (ARQ) techniques. FEC techniques include, for example, convolutional or block coding of the data prior to modulation. FEC coding involves representing a certain number of data bits using a certain number of code bits. Thus, it is common to refer to convolutional codes by their code rates, e.g., $1/2$ and $1/3$, wherein the lower code rates provide greater error protection but lower user bit rates for a given channel bit rate.

ARQ techniques involve analyzing received blocks of data for errors and requesting retransmission of blocks which contain errors. Consider, for example, the block mapping example illustrated in FIG. 2 for a radiocommunication system operating in accordance with the Generalized Packet Radio Service (GPRS) optimization which has been proposed as a packet data service for GSM. Therein, a logical link control (LLC) frame containing a frame header (FH), a payload of information and a frame check sequence (FCS) is mapped into a plurality of radio link control (RLC) blocks, each of which include a block header (BH), information field, and block check sequence (BCS), which can be used by a receiver to check for errors in the information field. The RLC blocks are further mapped into physical layer bursts, i.e., the radio signals which have been GMSK modulated onto the carrier wave for transmission. In this example, the information contained in each RLC block can be interleaved over four bursts (timeslots) for transmission.

Eriksson at NOK00083210, 2:15-49. Eriksson describes the advantages of using these techniques together (referred to as hybrid ARQ):

Strengths and weaknesses of these two error control schemes can be balanced by combining FEC and ARQ techniques. Such combined techniques, commonly referred to as hybrid ARQ techniques, permits correction of some received errors using the FEC coding at the receiver, with other errors requiring retransmission. Proper selection of FEC coding schemes with ARQ protocols thus results in a hybrid ARQ technique having greater reliability than a system employing a purely FEC coding scheme with greater throughput than a system employing a purely ARQ-type error handling mechanism.

418. Eriksson at NOK00083210-211, 2:65-3:8. However, as already noted, one disadvantage of using hybrid ARQ with incremental redundancy is increased demand for shared memory based on both the number of retransmissions and the degree of interleaving applied. Eriksson at NOK00083211, 3:23-52. As described in Section VII.A.5 above, a POSITA would have understood that it was common to allocate memory from a common source to support multiple functions. A POSITA would therefore have understood that the shared memory must be allocated between the ARQ and interleaving/deinterleaving functions described in Eriksson.

419. The standard implementation of cellular technology uses a baseband processor that employs shared memory. Examples include the Zyray spinner baseband processor that supported 3GPP WCDMA technology such as that disclosed in TR 25.896. A press release from April 2004 describes the WCDMA baseband processor as supporting 3GPP Release 99 and providing an upgrade path to release 4.

WCDMA Baseband Processor

Wed, 04/07/2004 - 11:04am by

Zyray introduces the SPINNERchip 1.1. The product enables device manufacturers to develop single-mode WCDMA, dual-mode GSM/GPRS & WCDMA, and multi-mode GSM/GPRS, EDGE & WCDMA devices. SPINNERchip 1.1 is housed in a 10 × 10 mm BGA including integrated analog technology and is flexible enough to mate with various RF transceivers, baseband processors, and dual-mode protocol stacks. Other features include the following: 3GPP Rel 99 December 2003 compliant WCDMA FDD modem; firmware upgrade path via to 3GPP Rel 4; 384 kbps class modem operation in both uplink and downlink; supports Circuit-Switched Voice, Packet-Switched Data and Circuit-Switched Data; Generic Layer 1 Abstraction Layer allows for 3rd party protocol stack support; complete firmware suite including PHY layer and Layer 1; support for USIM interface (optional); on-chip, programmable power management; supply voltage: 1.8 V / 3.0 V.

Zyray Wireless

420. TR 25.896 explicitly discusses retransmission (hybrid ARQ) as an enhancement of 3GPP Release 99, for example on page 39, excerpted below.

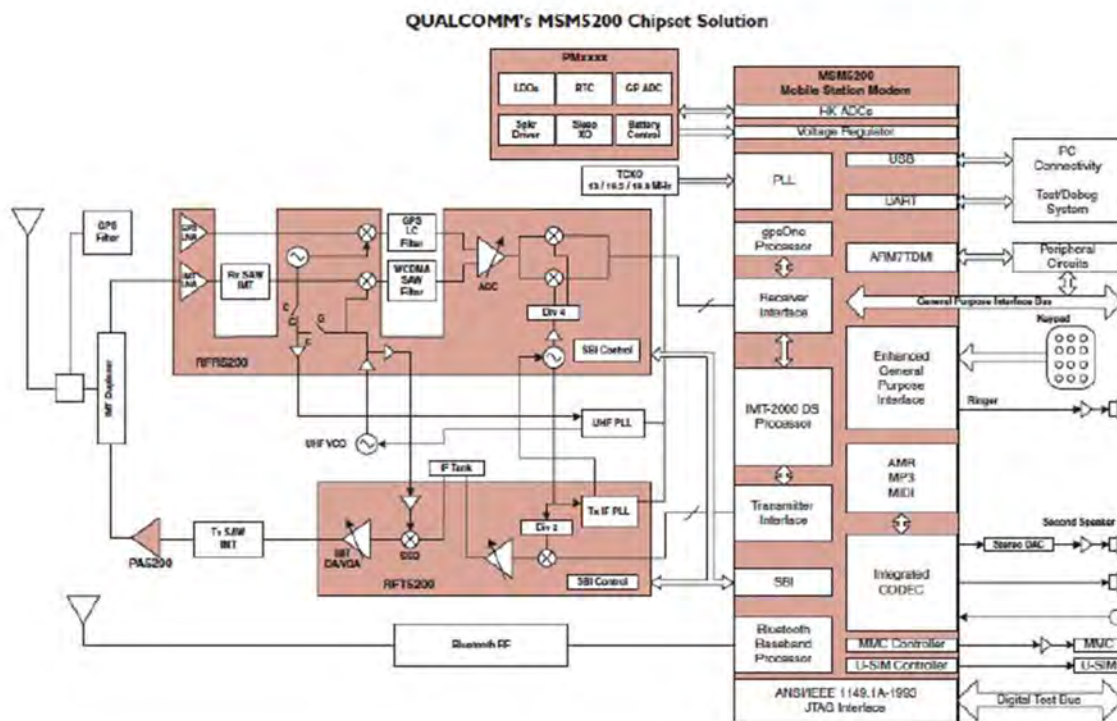
8.3.1 Reuse of current physical layer structure

In this alternative the E-DCH is time multiplexed into the same coded composite transport channel (CCTrCH) as the other DCHs if present. The TFCI indirectly informs where and how many bits of each DCH within the CCTrCH are, regardless of the DCH being a Rel99 DCH or an E-DCH.

Time multiplexing is easiest to implement if the TTI length is 10/20/40/80 ms, since then the Rel99 transport channel multiplexing chain can be used. There may naturally be some enhancements, e.g., to rate matching, to support the potential new enhanced uplink features, e.g., hybrid ARQ.

TR 25.896 at NOK00074964, § 8.3.1 "Reuse of current physical layer structure."

421. Another example is the Qualcomm Mobile Station Modem (MSM) 5200 baseband processor chip that was available by 2001 and was 3GPP Release 99 compliant. Shown below is an excerpt from Qualcomm's product literature about the MSM 5200:



MSM5200 DEVICE DESCRIPTION



The MSM5200 chipset and system software builds on the successful architecture of QCT's MSM3xxx and MSM5xxx families of CDMA devices, and is based on QCT's core CDMA systems expertise gained from years of field experience with this technology. The MSM5200 solution integrates both digital and analog

functions into a single chip.

The MSM5200 solution includes advanced technologies such as GPS-based position location and Bluetooth, as well as multimedia features such as Qtones™ MP3 player software and Compact Media Extension™ (CMX™) MIDI-based multimedia software. Along with an

optimized software solution for the WCDMA modem, QCT offers system development software, verification, test, debug, calibration, manufacturing, and field test support using the WCDMA designer development tools, which reduce time-to-market for a complete WCDMA product.

MSM5200 General Device Enhancements

• Integrated audio codec • Supports low-power, low-frequency crystal to enable TCXO shutoff • Enhanced I/O support for faster RS-232	• 16-bit-wide memory support • Embedded Bluetooth baseband processor • Voice Recognition (optional) –Speaker-dependent, speaker-independent and	- voice-prompt support –Multiple-language support • Integrated mass-storage device (MMC) controller • Audio enhancement technology support (Qtones and	- CMX software) • Integrated USB device controller for fast and simple PC interconnect • Interfaces with the MSP™ family of applications processors

MSM5200 IMT-2000 DS Device Enhancements

• 3GPP Release 99 compliant • Full 3GPP protocol stack including L1, L2:MAC/RLC/BMC/PDCP,	- L3:RRC, (G)MM, SS, CC, SMS, SM, RABM • Data rates: 384 kbps on both the uplink and downlink	• Adaptive Multiple Rate (AMR) speech codec • Support for convolutional and turbo coding	• USIM interface

422. Thus it would be obvious to a POSITA to implement the system disclosed in TR 25.896 and TS 25.401, which uses both interleaving and retransmission, on a baseband processor. These baseband processors are programmable devices that would naturally use shared memory and would naturally allocate memory between a retransmission function and an interleaving and/or deinterleaving function.

423. Accordingly, TR 25.896 and TS 25.401 in view of Eriksson renders obvious “identifying at least one packet of the plurality of packets as a packet that should be retransmitted” (both TR 25.896 and Eriksson identify errored packets as packets that should be retransmitted) “and allocating a memory between a retransmission function” (the Node Bs and UEs in both TR 25.896 and Eriksson use a shared memory called a buffer to support hybrid ARQ retransmission, with the UE storing packets for retransmission and the Node B storing packets for soft combining, with the

size of these buffers being set based on transmission parameters) “and an interleaving and/or deinterleaving function” (the Node Bs and UEs in TR 25.896 support interleaving/deinterleaving, which uses a shared memory as taught by Eriksson).

[10.d] wherein at least a portion of the memory may be allocated to the retransmission function or to the interleaving and/or deinterleaving function at any one particular time, and

424. As described above in VII.A.3-5, a POSITA would have understood that interleaving, retransmission, and sharing memory were all well-known in the art long before the priority date of the '411 patent. The claimed limitation that at least a portion of the memory may be allocated to the retransmission function or to the interleaving and/or deinterleaving function at any one particular time is therefore nothing more than the combination of known elements.

425. In particular, having a portion of the memory that may be allocated to the retransmission function or to the interleaving and/or deinterleaving function at any one particular time was an inherent feature of any programmable device with a shared memory resource that implemented both an interleaving function and a retransmission function. The use of such programmable devices featuring a shared memory was prevalent long before the alleged time of the invention. Examples include DSPs such as the TI TMS 320 which implemented transceivers including the TNET5100 as part of the AR5 chipset, the Intel 5110 implementation of 802.16 which used Dual-core ARM 946E-S engines and a DSP engine which are programmable processors that use shared memories, and baseband processors used to implement cellular communications. I have discussed all of these above with respect to this claim and in the technology background and I include that discussion by reference.

426. Further, as described further below, many prior artists did in fact create transceivers wherein at least a portion of the memory may be allocated to the retransmission function or to the interleaving and/or deinterleaving function at any one particular time. *See* Sections *i-iii* below. This

shows that combining these old elements was predictable and within the skill of a POSITA. Thus, as shown below with the specific prior art combinations, this limitation is obvious as it is the combination of known elements in a known manner to reach a predictable result.

i VDSL1 in View of ADSL2

427. VDSL1 in view of ADSL2 renders obvious “wherein at least a portion of the memory may be allocated to the retransmission function or to the interleaving and/or deinterleaving function at any one particular time.” Specifically, ADSL2 and VDSL1 both describe a transceiver with memory that can be used for interleaving/deinterleaving and retransmission functions as well as other functions. I have discussed this above in the earlier limitations of this claim, and I include that discussion here by reference.

428. VDSL1 transceivers interleave data in the required “slow” payload in the PMS-TC sublayer.

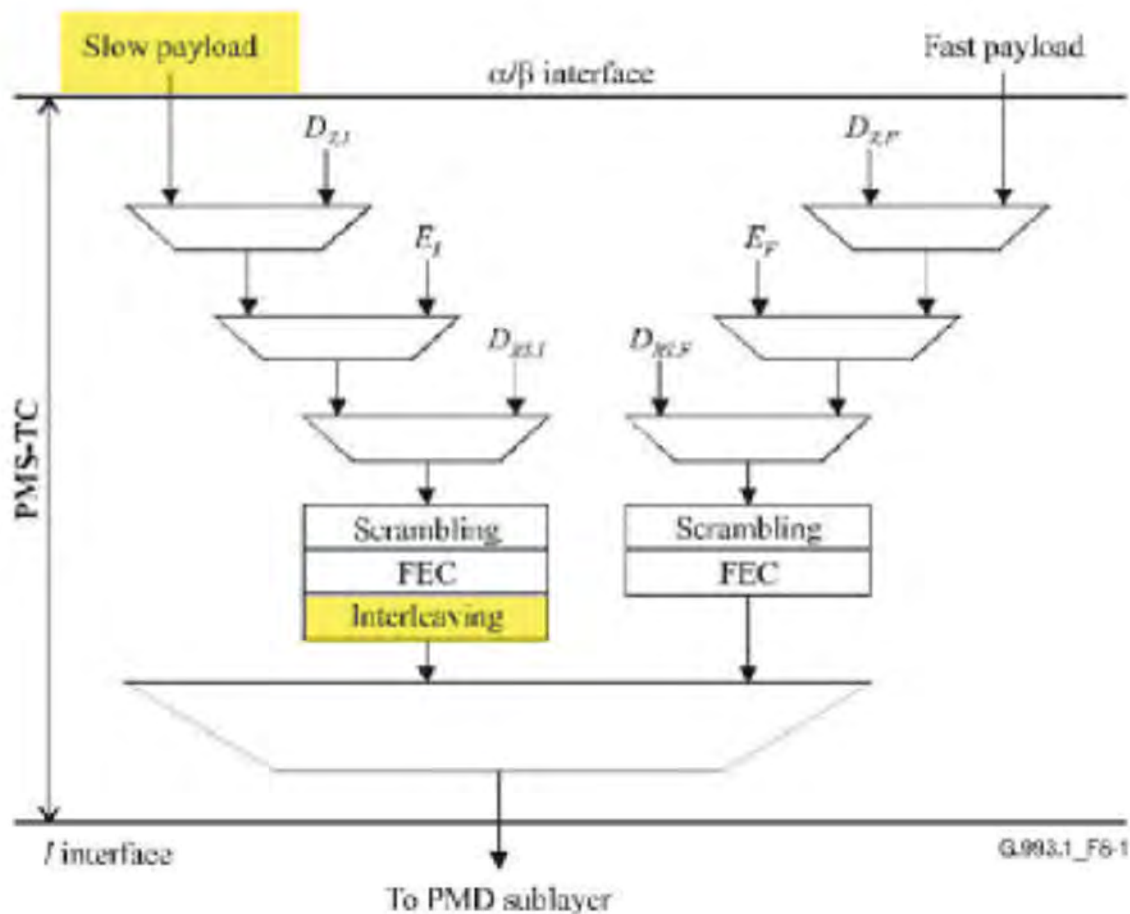


Figure 8-1/G.993.1 – Diagram of PMS-TC sublayer

VDSL1 at NOK00078784, Figure 8-1. *See also:*

8 PMS-TC sublayer

The PMS-TC sublayer provides transmission medium specific TC functions, such as framing, Forward Error Correction (FEC), and interleaving.

8.1 PMS-TC functional model

All data bytes shall be transmitted MSB first. All serial processing (e.g., scrambling, CRC calculation) shall however be performed LSB first, with the outside world MSB considered as the VDSL LSB. As a result, the first incoming bit (outside world MSB) shall be the first bit processed inside VDSL (VDSL LSB). The PMS-TC functional diagram is presented in Figure 8-1.

VDSL1 at NOK00078783, §§ 8-8.1 “PMS-TC sublayer” and “PMS-TC functional model.”

429. VDSL1 describes the interleaver (and, consequently, the deinterleaver) in detail.

Interleaving shall be used to protect the data against bursts of errors by spreading the errors over a number of Reed-Solomon codewords. The interleave depth shall be programmable with a maximum interleave depth of 64 codewords when the number of octets per codeword (N) equals 255. For smaller values of N the interleave depth can grow nearly proportionately.

It shall be possible to adjust the interleave depth via the management system to meet latency requirements. The latency of the slow path is a function of the data rate and burst error correction capability. For data rates greater than or equal to 13 Mbit/s the latency between the a and 13 interfaces shall not exceed 10 Ms when the interleaver depth is set to the maximum. At lower data rates there is a trade-off between higher latency and decreased burst error correction ability. At any data rate, the minimum latency occurs when the interleaver is turned off.

When the interleaver is on the codewords shall be interleaved before transmission to increase the immunity of RS codewords to bursts of errors. The convolutional interleaver is defined by two parameters: the interleaver block length, I, and the interleaving depth D. The block length I shall divide the RS codeword length N (i.e., N shall be an integer multiple of I). The convolutional interleaver uses a memory in which a block of I octets is written while an (interleaved) block of I octets is read. Details of the implementation are given in 8.4.2.

The same size interleaving memory (see Table 8-1) is needed for interleaving at the transmitter and de-interleaving at the receiver.

The convolutional interleaving introduces an absolute read-to-write delay, A_j , that increments linearly with the octet index within a block of I octet :

$$\Delta_j \equiv (D - 1) \times j$$

where $j = 0, 1, 2, \dots, I - 1$.

VDSL1 at NOK00078785, § 8.4.1 “Interleaving – General”. The amount of memory required to support interleaving depends on the interleaver parameters, data rate, and interleaver depth, and VDSL1 provides this information in a table. Note that the interleaver/deinterleaver have identical memory requirements.

Table 8-2/G.993.1 – Example of interleaver parameters with RS(144,128)

Rate [kbit/s]	Interleaver parameters	Interleaver depth	(De)interleaver memory size	Erasure correction	End-to-end delay
50×1024	$I = 72$ $M = 13$	937 blocks of 72 bytes	33 228 bytes	3 748 bytes 520 μ s	9.23 ms
24×1024	$I = 36$ $M = 24$	865 blocks of 36 bytes	15 120 bytes	1 730 bytes 500 μ s	8.75 ms
12×1024	$I = 36$ $M = 12$	433 blocks of 36 bytes	7 560 bytes	866 bytes 501 μ s	8.75 ms
6×1024	$I = 18$ $M = 24$	433 blocks of 18 bytes	3 672 bytes	433 bytes 501 μ s	8.5 ms
4×1024	$I = 18$ $M = 16$	289 blocks of 18 bytes	2 448 bytes	289 bytes 501 μ s	8.5 ms
2×1024	$I = 18$ $M = 8$	145 blocks of 18 bytes	1 224 bytes	145 bytes 503 μ s	8.5 ms

VDSL1 at NOK00078786, Table 8-2.

430. As in VDSL1, in ADSL2 the PMS-TC sublayer in ADSL2 interleaves data being transmitted.

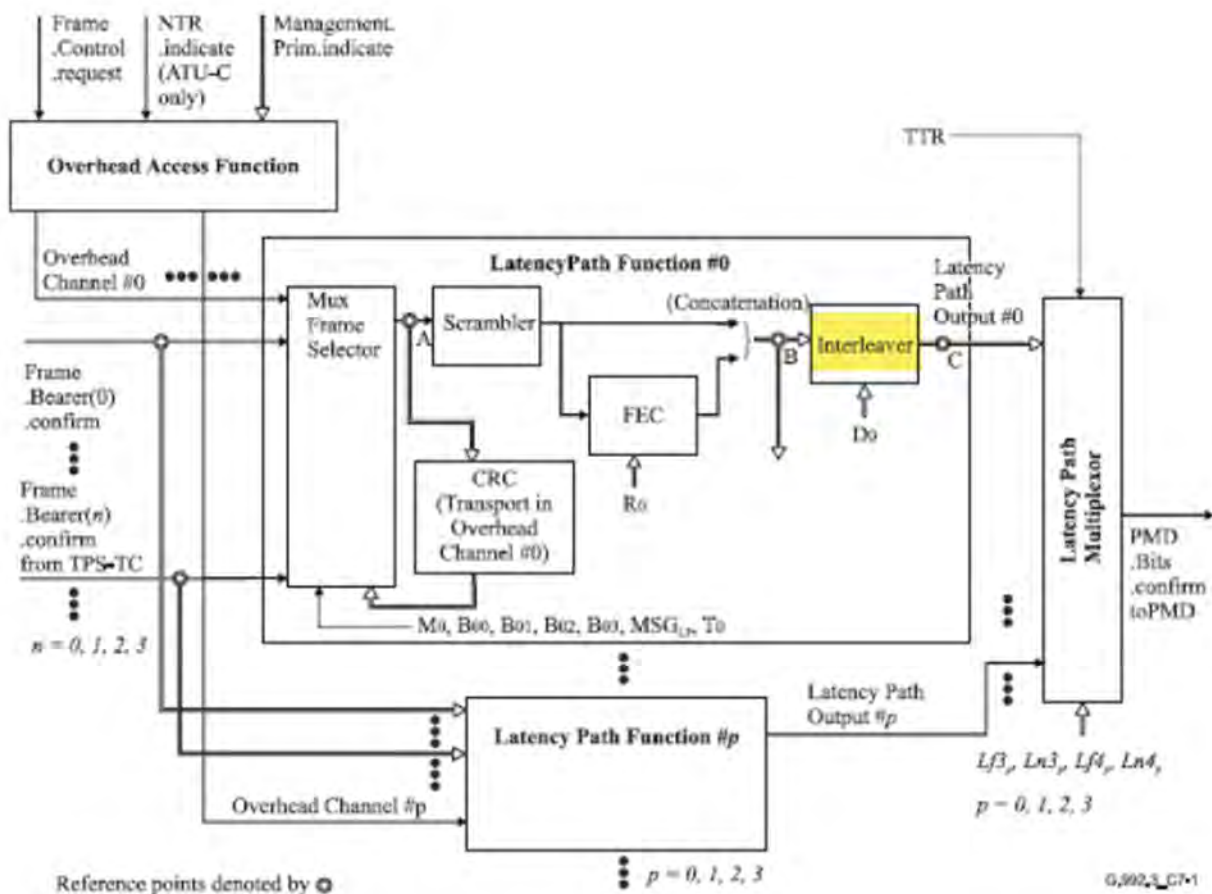


Figure C.7-1/G.992.3 – Block diagram of transmit PMS-TC function

ADSL2 at 227, Figure C.7-1.

431. ADSL2 uses a convolutional interleaver to protect data from impulse noise and describes the operation of the convolutional interleaver (and, consequently, the deinterleaver) in detail.

To spread the Reed-Solomon codeword and, therefore, reduce the probability of failure of the FEC in the presence of impulse noise, the FEC Output Data Frames shall be convolutionally interleaved. The interleaver creates the Interleaved FEC Output Data Frames at Reference point C, at the output of the latency path function. This procedure is followed by the frame multiplexing procedure.

Convolutional interleaving is defined by the rule (using the currently defined values of the framing control parameters D_p and the derived parameter $N_{FEC,p}$):

Each of the $N_{FEC,p}$ octets $B_0, B_1, \dots, B_{N_{FEC,p}-1}$ in an FEC Output Data Frame is delayed by an amount that varies linearly with the octet index. More precisely, octet B_i (with index i) is delayed by $(D_p - 1) \times i$ octets, where D_p is the interleaver depth.

An example for $N_{FEC,p} = 5, D_p = 2$ is shown in Table 7-13, where B_i^j denotes the i -th octet of the j -th FEC Output Data Frame.

Table 7-13/G.992.3 – Convolutional interleaving example for $N_{FEC,p} = 5, D_p = 2$

Interleaver input	B_0^j	B_1^j	B_2^j	B_3^j	B_4^j	B_0^{j+1}	B_1^{j+1}	B_2^{j+1}	B_3^{j+1}	B_4^{j+1}
Interleaver output	B_0^j	B_3^{j-1}	B_1^j	B_4^{j-1}	B_2^j	B_0^{j+1}	B_3^j	B_1^{j+1}	B_4^j	B_2^{j+1}

With the above-defined rule, the output octets from the interleaver always occupy distinct time slots when $N_{FEC,p}$ is odd and D_p is a power of 2. When $N_{FEC,p}$ is even, a dummy octet shall be added at the beginning of the codeword at the input to the interleaver. The resultant odd-length codeword is then convolutionally interleaved, and the dummy octet shall then be removed from the output of the interleaver.

The interleaving procedure of the latency path function #p shall interleave a single FEC Output Data Frame, or $M_p \times K_p + R_p$ octets. This procedure is followed by the Frame Multiplexing procedure.

ADSL2 at 38, § 7.7.1.5 “Interleaver.”

432. ADSL2 protects overhead messages (or segments of overhead messages) from errors using retransmission as well as interleaving.

Each transmitted message segment shall be acknowledged by the far-end with a segment acknowledge message, except the last one. The last command message segment shall be acknowledged by the far-end with the appropriate response message (identical to the non-segmented case). The last response message segment shall not be acknowledged. Table 7-17a defines the segment acknowledge message. The following shall apply to all encapsulated segment acknowledge messages:

- HDLC frame shall contain 5 message octets ($P = 5$);
- The two least significant bits of the address field shall be identical to the corresponding bits in the address field of the acknowledged message segment. All other bits of the address field shall be set to 02;

- The second least significant bit of the control field shall either indicate a command message (request to continue the response, e.g., L2 grant) or a response message (request to continue the command, e.g., OLR);
- The least significant bit of the control field shall toggle for each acknowledgement message as it normally toggles for each command/response message (see 7.8.2.4.2);
- All other bits of the control field shall be set to 02.

Table 7-17a/G.992.3 – Segment acknowledge message

Message octet number	Message octet definition
Octet 1	Message designator 1111 0000 _b for acknowledgement of high priority message segment 1111 0001 _b for acknowledgement of normal priority message segment 1111 0010 _b for acknowledgement of low priority message segment
Octet 2	Message type of segment acknowledge message 01 ₁₆
Octet 3	Acknowledged message segment ID (in range 1 to 7)
Octet 4	Message Designator (first message octet of acknowledged message segment)
Octet 5	Message Type (second message octet of acknowledged message segment)

Every message segment shall be acknowledged by the far-end before the next message segment is transmitted. The segment acknowledge message shall not be segmented.

Timeouts shall be defined as follows:

- For a non-segmented message, timeout shall apply between the last octet of the command message transmitted and the first octet of the response message received.
- For a segmented response message, the timeout shall apply between the last octet of the command message transmitted and the first octet of the first message segment received.
- A timeout corresponding to the command priority shall apply between the last octet of the message segment transmitted and the first octet of the segment acknowledge message received.
- Another timeout corresponding to the command priority shall apply between the last octet of the segment acknowledge message transmitted and the first octet of the next message segment received.

If a timeout expires, the transmitter may repeat the last transmitted message. This repeated message can be an unsegmented command message, a command/response message segment (except the last segment of a response message), or a segment acknowledge message. Alternately, the ATU may abandon the message after an implementation-specific number of retransmissions.

ADSL2 at 43-46, § 7.8.2.4.3 “Overhead message segmentation.”

433. The amount of memory required to support ADSL2’s overhead message retransmission depends on the data rate of the channel as well as the length of time that messages are stored.

The information in C-PARAM includes:

- The latency path MSGLP to carry the upstream message-oriented portion of the overhead channel.
- Assignment of upstream frame bearers to upstream latency paths.
- The number of message octets MSGc included in the upstream overhead structure.
- Bpn for each upstream latency path and frame bearer.
- Mp for each upstream latency path.
- Rp for each upstream latency path.
- Dp for each upstream latency path.
- TP for each upstream latency path.
- · Lf3p, Ln3p, Lf4p, Ln4p corresponding to each upstream latency path.

The information in R-PARAM includes:

- The latency path MSGLP to carry the downstream message oriented portion of the overhead channel.
- Assignment of downstream frame bearers to downstream latency paths.
- The number of message octets MSGc included in the downstream overhead structure.
- Bpn for each downstream latency path and frame bearer.

- M_p for each downstream latency path.
- R_p for each downstream latency path.
- D_p for each downstream latency path.
- TP for each downstream latency path.
- $Lf3p, Ln3p, Lf4p, Ln4p$ corresponding to each downstream latency path.

This C-PARAMS and R-PARAMS information is represented as a parameter block as in Table C.7-2. The information is transmitted in the order shown during C-PARAM and R-PARAM as described in the PMD initialization procedure.

ADSL2 at 229, § C.7.10.3 “Exchange phase.” *See also id.* 228, Table C.7-1 (excerpted):

K_p	The number of octets per Mux Data Frame in latency path function # p is always $K_p = \sum_{l=0}^{N_{EC}-1} B_{p,l} + 1$
$N_{FEC,p}$	The number of octets per FEC Data Frame and Interleaved FEC Data Frame in latency path function # p is always $N_{FEC,p} = M_p \times K_p + R_p$.
L_p	Average number of bits per data symbol $L_p = (96 \times Lf4_p + 30 \times Lf3_p + 144 \times Ln4_p + 70 \times Ln3_p) / 340$
S_p	Not accounting for the interleaving procedure, the number of PMD.Bits.request primitives (and correspondingly the number of PMD symbols) over which the FEC Data Frame spans is always $S_p = \frac{8 \times N_{FEC,p}}{L_p}$. The value of S_p may represent a non-integer value.

Overhead rate OR_p of latency path function # p	Overhead rate is always $OR_p = \frac{M_p}{T_p \times S_p} \times 32 \text{ kbit/s} = \frac{M_p \times L_p}{T_p \times (K_p \times M_p + R_p)} \times 4 \text{ kbit/s}$

434. Thus, both ADSL2 and VDSL1 disclose implementing convolutional interleaving, which has a well-defined minimum memory requirement for the well-known triangular implementation. In order to implement this convolutional interleaving at all, any implementation would have been required to allocate memory to this function. Both ADLS2 and VDSL1 also implemented retransmission. Since products implementing multicarrier transmission with

interleaving and retransmission such as the Intel 5116 chip implementing 802.16 and specifically products implementing DSL transceivers such as the TI TNET5100 typically use shared memory, it would be obvious to allocate shared memory for this interleaving function. The obviousness of this choice is further illustrated by, for example, Mazzoni's and Moon's disclosures of using a shared memory for multiple error correction functions in a communications transceiver, and the corresponding memory savings that could be achieved. A POSITA would have found it obvious that each portion of memory is used for a particular one of the retransmission, interleaving (on the transmit side) or deinterleaving (on the receive side) functions at any one time.

435. In particular, having a portion of the memory that may be allocated to the retransmission function or to the interleaving and/or deinterleaving function at any one particular time was an inherent feature of any programmable device with a shared memory resource that implemented both an interleaving function and a retransmission function. The use of such programmable devices featuring a shared memory was prevalent long before the alleged time of the invention. Examples include DSPs such as the TI TMS 320 which implemented transceivers including the TNET5100 as part of the AR5 chipset, the Intel 5110 implementation of 802.16 which used Dual-core ARM 946E-S engines and a DSP engine which are programmable processors that use shared memories, and baseband processors used to implement cellular communications.

436. Accordingly, VDSL1 in view of ADSL2 renders obvious "wherein at least a portion of the memory may be allocated to the retransmission function or to the interleaving and/or deinterleaving function at any one particular time" (VDSL1 and ADSL2 disclose or render obvious using the memory to support interleaving/deinterleaving and retransmission such that it would have been obvious to a POSITA that at least a portion of that memory is allocated to one of those functions at any given time).

437. TR 25.896 and TS 25.401 in view of Eriksson renders obvious “wherein at least a portion of the memory may be allocated to the retransmission function or to the interleaving and/or deinterleaving function at any one particular time.” Specifically, both TR 25.896 and Eriksson render obvious a transceiver with memory wherein a portion of that memory may be used to support either hybrid ARQ retransmission or interleaving/deinterleaving at a given time.

438. TR 25.896 interleaves packets sent using hybrid ARQ twice, as shown in Figure 8.4.7:

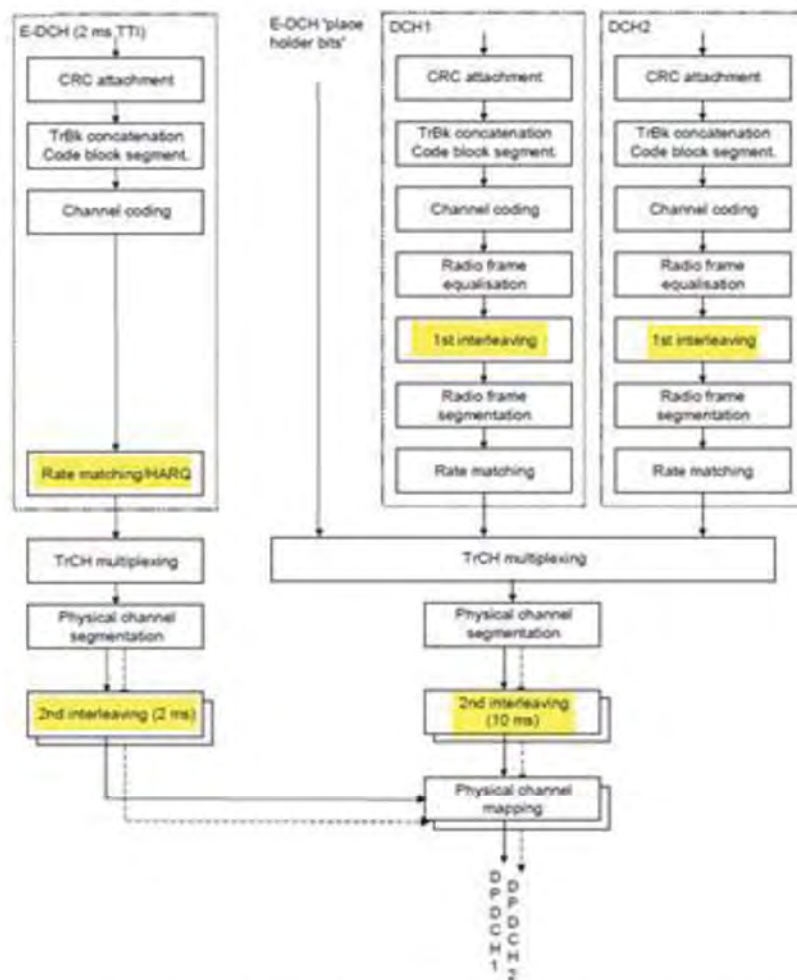


Figure 8.4.7. Time multiplexing of E-DCH (2ms TTI) and DCH (10ms TTI) with place holder bits.

TR 25.896 at NOK00074969, Figure 8.4.7. The first and second interleaving performed in the Release 6 system are described in detail in TS 25.212 §§ 4.2.5 and 4.2.11. *See* §§ IX.D-E.

439. TR 25.896 further describes that a portion of memory is allocated to the hybrid ARQ retransmission function.

The principle of hybrid ARQ is to buffer E-DCH TTIs that were not received correctly and consequently combine the buffered data with retransmissions. The actual method of doing soft combining depends on the HARQ combining scheme selected. In Chase combining scheme the receiver always combines the full retransmission of the failed TTI, i.e. the amount of data in the receiver buffer remains the same. In the incremental redundancy schemes the receiver buffers coded symbols, which introduce new information to the E-DCH TTI transmitted first, i.e. the amount of data to be buffered increases with consecutive retransmissions. However, in practice the buffer in the receiver needs to be dimensioned considering the maximum number of coded bits of the E-DCH TTI after all the incremental redundancy has been introduced. Regardless of the HARQ combining scheme soft combining is done on L1 of Node B before the decoding stage of FEC. Prior to decoding these symbols are soft-valued, i.e. each symbol is represented by two or more bits. Here we call them soft symbols.

For N-process SAW HARQ, the number of soft symbols to be buffered in L1 receiver can be estimated generally as follows, since no new PDUs are transmitted on a subchannel before the previous packet is acknowledged. The receiver has to buffer one E-DCH TTI for each HARQ process and for each UE. The next transmission is either a new packet or a retransmission of an erroneous packet. In either case, the maximum buffering need is N E-DCH TTIs. The actual size of the buffer needed for each E-DCH TTI depends on the HARQ combining scheme as described above. Thus, for N-process stop-and-wait ARQ the L1 buffering at Node B for one UE can be expressed as:

$$buffer = (softsymbols_{TTI} \times N)$$

Table 9.2.4 shows some examples for the soft buffer size required at Node B per UE. It should be noted that soft buffer is needed for all UEs waiting for retransmission at a given time. Thus these values have to be multiplied by the number of active UEs using E-DCH. Depending on particular implementations, the buffer may be shared between UEs and thus the total size of soft buffer can be reduced given that not all active UEs transmit simultaneously using the most demanding format. Also, it may be possible to re-use some existing buffers.

Table 9.2.4 Node B soft buffer size (kilo soft symbols) per UE, for some example values of N (number of HARQ processes) and TTI lengths

	N=4, TTI=10 ms	N=3, TTI=10 ms	N=8, TTI=2 ms	N=6, TTI=2 ms
BPSK, SF=4	38.4	28.8	15.36	11.52
QPSK or 2*BPSK, SF=4	76.8	57.6	30.72	23.04
8PSK or 3*BPSK, SF=4	115.2	86.4	46.08	34.56
2*QPSK or 4*BPSK, SF=4; or QPSK, SF=2	153.6	115.2	61.44	46.08

TR 25.896 at NOK00074991, § 9.2.2.1.1 “Soft buffer at Node B”. Either the RNC or the Node B must also set aside memory for a reordering buffer, which is used to correct the order of packets when they are received out of order due to retransmission.

Due to physical/MAC layer (H)ARQ, the MAC-e PDUs may be received in wrong order. Therefore, reordering of the MAC-e PDUs is required either in the Node B or in the RNC. Here the required reordering buffer size per UE is estimated. In the estimation it is assumed that only one high bit rate reordering queue is active for a UE at a time. Furthermore, maximum of 3 retransmissions per MAC-e PDU is assumed and the buffer size is calculated for the worst case where one ARQ process requires retransmissions while the others get blocks through and forward them to reordering queue. The worst case buffer size is then calculated as

$$\text{reorderingbuffer} = [3(N - 1) + 1] \times (\text{bits}/\text{TTI})$$

These values have to be multiplied by the number of E-DCH UEs served by the Node B or by the RNC, depending on the location of the reordering buffer. Depending on implementation and location of the re-ordering buffer, it may be possible to re-use some existing buffers.

Table 9.2.5 Reordering buffer size (kB) per UE, for some example values of N (number of HARQ processes) and TTI lengths

User data rate	N=4, TTI=10 ms	N=3, TTI=10 ms	N=8, TTI=2 ms	N=6, TTI=2 ms
384 kbit/s	4.8	3.36	2.1	1.5
768 kbit/s	9.6	6.72	4.2	3.1
1 Mbit/s	12.5	8.8	5.5	4.0
1.5 Mbit/s	18.8	13.1	8.3	6.0

TR 25.896 at NOK00074991, § 9.2.2.1.2 “Reordering buffer in radio network” and Table 9.2.5. Both interleaving steps applied to the data will also require a reordering buffer to deinterleave the interleaved data and put the interleaved packets/data in the correct order.

440. The UE must set aside a sufficient portion of the shared memory (“buffer” in the 3GPP standards) to support the interleaving and retransmission used by hybrid ARQ with

incremental redundancy. The minimum amount of memory the UE must set aside to support uplink hybrid ARQ with incremental redundancy depends on the amount of information transmitted in each TTI.

In addition to buffers in the receiver side, also some buffer at UE side is needed. The retransmission buffer can be either at MAC layer as info bits or at physical layer as coded bits. The total number of coded bits is typically 3 times the number of info bits. Table 9.2.4 shows also the required retransmission buffer size in kbits (assuming that coded bits are buffered). Depending on implementation it may be possible to re-use existing buffers.

TR 25.896 at NOK00074992, § 9.2.2.1.3 “Retransmission buffer in UE.”

441. Eriksson suggests changing the MCS or FEC scheme to accommodate link quality or transceiver memory limitations. This may cause the Node B to direct the UE to use a different MCS scheme or more robust FEC, therefore allocating less memory to hybrid ARQ retransmission.

MCS changes may or may not be based solely on reported link quality estimates. For example, when incremental redundancy is used and the receiver has limited memory it may be beneficial to increase the robustness of the MCS even though (in a system with unlimited memory) it would decrease throughput. Consider the following scenario. Using a less robust MCS, the number of required retransmissions for successful incremental redundancy combination will be higher. This, in turn, requires a lot of memory. If the receiver runs out of memory, it will begin to discard received blocks that have previously been stored for later incremental redundancy combination. Since the information transmitted using the relatively unrobust MCS probably relies in part upon incremental redundancy combining to achieve acceptable decoding performance, the result may be significant degradation in received signal quality. Hence it may be better under such circumstances to dynamically increase the robustness of the transmission's MCS, e.g., when the receiver starts to run out of memory.

Yet another factor which further complicates this process is the possibility of changing the MCS associated with blocks that are being retransmitted. If link adaptation is used without incremental redundancy, then changing the MCS for retransmissions may be very desirable based upon the measured link quality. On the other hand, if incremental redundancy is employed, using a different MCS may make it impossible to combine the retransmitted block with the originally transmitted block. However, if the link quality changes significantly it still may be desirable to change the MCS even if some of the earlier transmitted data blocks cannot be used in the redundancy combining process at the receiver.

Thus it can be seen that there are many challenges associated with optimizing the manner in which these various techniques are employed. To enable dynamic changing of the MCS during a connection, some form of overhead signalling is necessary between the transmitter and receiver.

Eriksson at NOK00083211, 3:63-4:32. A POSITA would have understood that Eriksson's suggestion to change the initial allocation of memory between interleaving/deinterleaving and hybrid ARQ retransmission has the advantage of allowing a system (such as TR 25.896) to more easily respond to changes in link quality. A POSITA would therefore be motivated to modify the system of TR 25.896 based on Eriksson so that the initial hybrid ARQ/interleaving setup (described further with respect to limitation [10.e]) could be modified to respond to changing channel conditions. A POSITA would have understood that allowing the Node B to provide similar signaling to the UE, for example when the Node B is experiencing severe interference, i.e., at any one particular time, from other UEs and it would prefer the UE use retransmission rather than increase transmit power (or otherwise modify its MCS or FEC scheme to increase the likelihood of correct packet reception). *Id.*

442. The typical implementation of cellular technology uses a baseband processor that employs shared memory. Examples include the Zyray spinner baseband processor that supported 3GPP WCDMA technology such as that disclosed in TR 25.896. A press release from April 2004 describes the WCDMA baseband processor as supporting 3GPP Release 99 and providing an upgrade path to release 4.

WCDMA Baseband Processor

Wed, 04/07/2004 - 11:04am by

Zyray introduces the SPINNERchip 1.1. The product enables device manufacturers to develop single-mode WCDMA, dual-mode GSM/GPRS & WCDMA, and multi-mode GSM/GPRS, EDGE & WCDMA devices. SPINNERchip 1.1 is housed in a 10 × 10 mm BGA including integrated analog technology and is flexible enough to mate with various RF transceivers, baseband processors, and dual-mode protocol stacks. Other features include the following: 3GPP Rel 99 December 2003 compliant WCDMA FDD modem; firmware upgrade path via to 3GPP Rel 4; 384 kbps class modem operation in both uplink and downlink; supports Circuit-Switched Voice, Packet-Switched Data and Circuit-Switched Data; Generic Layer 1 Abstraction Layer allows for 3rd party protocol stack support; complete firmware suite including PHY layer and Layer 1; support for USIM interface (optional); on-chip, programmable power management; supply voltage: 1.8 V / 3.0 V.

Zyray Wireless

443. TR 25.896 explicitly discusses retransmission (hybrid ARQ) as an enhancement of 3GPP Release 99, for example on page 39, excerpted below.

Release 6

39

3GPP TR 25.896 V6.0.0 (2004-03)

8.3.1 Reuse of current physical layer structure

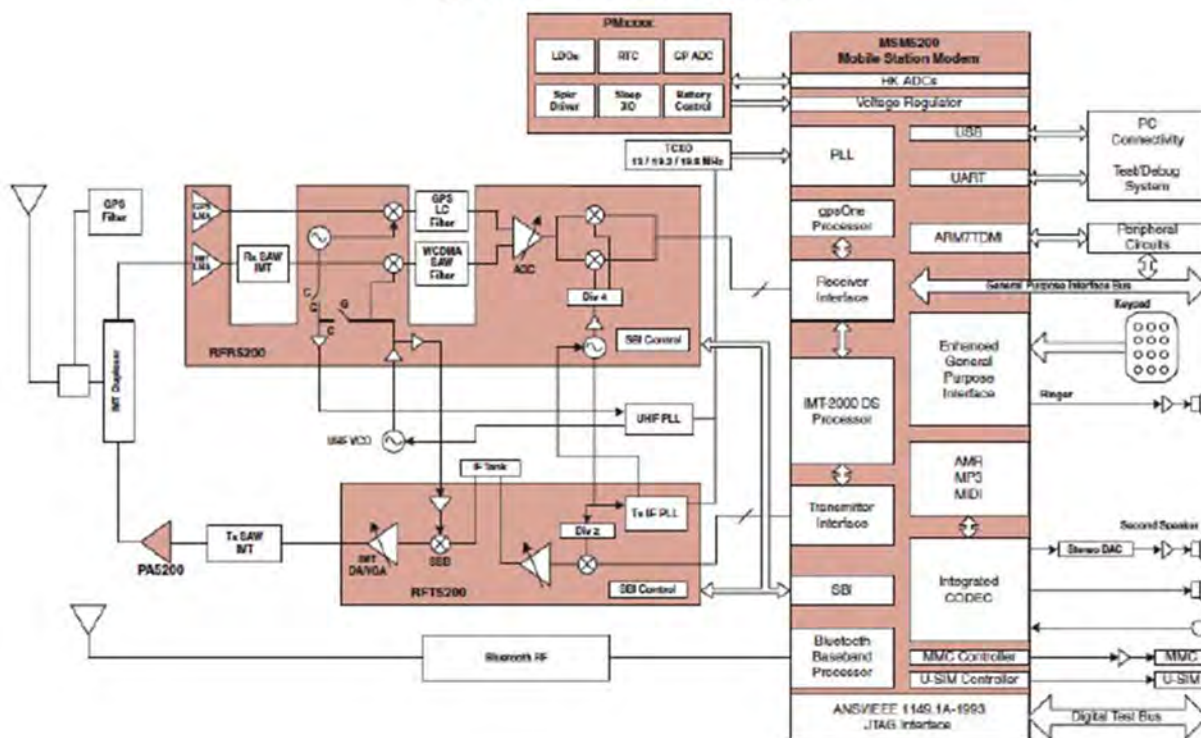
In this alternative the E-DCH is time multiplexed into the same coded composite transport channel (CCTrCH) as the other DCHs if present. The TFCI indirectly informs where and how many bits of each DCH within the CCTrCH are, regardless of the DCH being a Rel99 DCH or an E-DCH.

Time multiplexing is easiest to implement if the TTI length is 10/20/40/80 ms, since then the Rel'99 transport channel multiplexing chain can be used. There may naturally be some enhancements, e.g., to rate matching, to support the potential new enhanced uplink features, e.g., hybrid ARQ.

TR 24.896 at NOK00074964, § 8.3.1 “Reuse of current physical layer structure.”

444. Another example is the Qualcomm Mobile Station Modem (MSM) 5200 baseband processor chip that was available by 2001 and was 3GPP Release 99 compliant. Shown below is an excerpt from Qualcomm’s product literature about the MSM 5200:

QUALCOMM's MSM5200 Chipset Solution



MSM5200 DEVICE DESCRIPTION



The MSM5200 chipset and system software builds on the successful architecture of QCT's MSM3xxx and MSM5xxx families of CDMA devices, and is based on QCT's core CDMA systems expertise gained from years of field experience with this technology. The MSM5200 solution integrates both digital and analog

functions into a single chip.

The MSM5200 solution includes advanced technologies such as GPS-based position location and Bluetooth, as well as multimedia features such as Qtones™ MP3 player software and Compact Media Extension™ (CMX™) MIDI-based multimedia software. Along with an

optimized software solution for the WCDMA modem, QCT offers system development software, verification, test, debug, calibration, manufacturing, and field test support using the WCDMA designer development tools, which reduce time-to-market for a complete WCDMA product.

MSM5200 General Device Enhancements

- Integrated audio codec
- Supports low-power, low-frequency crystal to enable TXO shutoff
- Enhanced I/O support for faster RS-232
- 16-bit-wide memory support
- Embedded Bluetooth baseband processor
- Voice Recognition (optional)
 - Speaker-dependent, speaker-independent and voice-prompt support
- Integrated mass-storage device (MMC) controller
- Audio enhancement technology support (Qtones and CMX software)
- Integrated USB device controller for fast and simple PC interconnect
- Interfaces with the MSP™ family of applications processors

MSM5200 IMT-2000 DS Device Enhancements

- 3GPP Release 99 compliant
- Full 3GPP protocol stack including L1, L2:MAC/RLC/BMC/PDCP,
- L3:RRC, (G)MM, SS, CC, SMS, SM, RABM
- Data rates: 384 kbps on both the uplink and downlink
- Adaptive Multiple Rate (AMR) speech codec
- Support for convolutional and turbo coding
- USIM interface

445. Thus it would be obvious to a POSITA to implement the system disclosed in TR 25.896 and TS 25.401, which uses both interleaving and retransmission, on a baseband processor. These baseband processors are programmable devices that would naturally use shared memory and would naturally allocate memory between a retransmission function and an interleaving and/or deinterleaving function “wherein at least a portion of the memory may be allocated to the retransmission function or to the interleaving and/or deinterleaving function at any one particular time.”

446. Accordingly, TR 25.896 and TS 25.401 in view of Eriksson renders obvious “wherein at least a portion of the memory may be allocated to the retransmission function or to the interleaving and/or deinterleaving function at any one particular time” (the Node Bs and UEs in TR 25.896 support interleaving/deinterleaving and hybrid ARQ retransmission, which uses a shared memory as taught by Eriksson, and a POSITA would be motivated to modify TR 25.896 to incorporate Eriksson’s changing of hybrid ARQ retransmission and interleaving memory allocations to better accommodate limited memory or channel conditions).

[10.e] wherein a message transmitted during initialization indicates how the memory has been allocated between the retransmission function and the interleaving and/or deinterleaving function in the transceiver.

447. As described above in Sections VII.A.3-6, a POSITA would have understood that interleaving, retransmission, sharing memory, and messages allocating memory between functions were all well-known in the art long before the priority date of the ’411 patent. The claimed limitation that a message transmitted during initialization indicates how the memory has been allocated between the retransmission function and the interleaving and/or deinterleaving function in the transceiver is therefore nothing more than the combination of known elements.

448. As was known by a POSITA long before the alleged invention, in order for interleaving to work, both systems have to agree on the exact interleaving parameters so that the

deinterleaver in the receiver properly reverses the action of the interleaver in the transmitter. As long as there is more than one possible set of interleaving parameters that can be selected by the transceivers at the time of transmission, configuration messages must be used so that the systems agree on those parameters. As was known by a POSITA long before the alleged invention, for both block interleaving and convolutional interleaving, these parameters determine the minimum amount of memory required for a typical implementation.

449. Thus, it would be obvious that configuration messages that any such system would require to exchange the interleaving parameters could additionally include the amount of memory that has been allocated to the interleaver/deinterleaver. This would have had predictable results because it provided information that could have been computed by the other transceiver as well for a typical implementation.

450. As was known by a POSITA long before the alleged invention, in order for retransmission to work well, both systems have to agree on how long transmitted symbols, codewords, or packets will be retained for possible retransmission so that the system will behave in a predictable way when the transmitter requests a retransmission or when a timeout occurs that would trigger a retransmission. As long as there is more than one possible retransmission protocol that can be selected by the transceivers at the time of transmission, configuration messages must be used so that the systems agree on which protocol (i.e., which retransmission parameters) will be selected.

451. As was known by a POSITA long before the alleged invention, these parameters determine the amount of memory required by a typical implementation, i.e., the number of symbols that must be retained in the worst case determines how much memory is required by the transmitter for a typical implementation of retransmission. Similarly, the maximum number of received

symbols retained for possible improvement by a retransmitted symbol, codeword, or packet (and the size with which those symbols are represented) determines the minimum amount of memory required by the receiver using a typical implementation to properly benefit from retransmission.

452. Thus, it would be obvious that the configuration messages that any such system would require to exchange the retransmission parameters could additionally include the amount of memory that has been allocated to the retransmission function. This would have had predictable results because it provided information that could have been computed by the other transceiver as well for a typical implementation.

453. Numerous systems discussed in Sections VII.A.3-4, VII.B-E, and IX.A-I above and again below for specific prior art references demonstrate that it was well established and would have been obvious for a POSITA to implement a system with more than one interleaving option and more than one retransmission option. Examples include many wireless and wired communication standards including 3GPP and IEEE 802.16, and many articles in the literature discussed in the background above. For any of these systems configuration messages must be exchanged so that the two transceivers agree specifically on the exact interleaver parameters and the exact retransmission parameters. From these parameters the minimum memory requirements for a typical implementation are determined in a manner that is well understood by a POSITA. It would have been obvious that those memory allocations could be exchanged along with the parameters that were already exchanged. This would have predictable results providing information that could have been computed by the other transceiver as well.

454. Further, as described below, many prior artists did in fact create transceivers that made the addition of an explicit indication of how the memory has been allocated between the retransmission function and the interleaving and/or deinterleaving function obvious. *See* Sections

i-iii below. Thus, combining these old elements was predictable and within the skill of a POSITA. Thus, as shown below with the specific prior art combinations, this limitation is obvious as it is the combination of known elements in a known manner to reach a predictable result.

i VDSL1 in View of ADSL2

455. VDSL1 in view of ADSL2 renders obvious “wherein a message transmitted during initialization indicates how the memory has been allocated between the retransmission function and the interleaving and/or deinterleaving function in the transceiver.”

456. Specifically, VDSL1 discloses an initialization procedure in which the VTU-R and the VTU-O exchange messages concerning their capabilities, including their respective interleaving capabilities and, eventually, selected interleaving parameters. As discussed above, the typical triangular implementation of convolutional interleaving has a well-defined minimum memory requirement. Thus, these messages enable the transceivers at both the VTU-O and VTU-R side to deduce the minimum amount of the shared memory in the transceiver required to have been allocated to the interleaving/deinterleaving functions for this typical implementation. It is obvious that the explicit memory allocation information could be also included in these messages with predictable results. Thus, these messages render this limitation obvious.

The VTU-R shall send the message R-MSG2 over the SOC channel to transfer information about its bit allocation capabilities and several other features. After receiving this message, the VTU-O shall do the same by sending the SOC message O-MSG2, containing the capabilities of the VTU-O.

VDSL1 at NOK00078847-848, § 12.4.6.1 “Sequence of messages and symbols during channel analysis and exchange state.”

457. One of the initialization messages sent is called the O-MSG2. The O-MSG2 is sent from the VTU-O to the VTU-R and includes information about the VTU-O’s capabilities, including the interleaving/deinterleaving setting the VTU-O supports as well as the maximum interleaver

delay the VTU-O can use (in milliseconds, which with the data rate and other information can be translated into a number of bits).

Table 12-23/G.993.1 – Description of O-MSG2

Field content	Field or macro-field type	Remark
Message descriptor	1 byte	See Table 12-12.
Minimal SNR margin	1 byte	In units of 0.5 dB
Maximal constellation size in downstream ($B_{max,d}$)	1 byte	Maximum number of bits per tone
RS setting supported by VTU-O	1 byte	0x00: Only mandatory settings 0xFF: All settings
Interleaver settings supported by the VTU-O	1 byte	0x00: Only mandatory settings 0xFF: All settings 0xNN NN = Number of additional (i.e., non-mandatory) settings in hexadecimal (NN ≠ 0x00 and NN ≠ 0xFF)
Detailed interleaver setting description	0 byte if NN = 0x00 or NN = 0xFF, 4xNN otherwise	Interleaver descriptor (see Table 12-24)
Maximal power in downstream	1 byte (unsigned)	In units of 0.25 dBm
Maximum interleaver delay	1 byte	In ms by steps of 0.5 ms (Note 1)
Maximum number of EOC bytes per frame in downstream	1 byte	Number of EOC bytes per frame
Maximum number of VOC bytes per frame in downstream	1 byte	Number of VOC bytes per frame

Table 12-23/G.993.1 – Description of O-MSG2

Field content	Field or macro-field type	Remark
Support of express bit swapping	1 byte	0x00: Not supported 0xFF: Supported
j_{max}	1 byte	Maximum value of j_{max} supported by the VTU-O (Note 2)
NOTE 1 – This field can be set to 0 in order to emulate the fast channel. This field is used for the creation of R-CONTRACT1 even if dual latency is used at the end.		
NOTE 2 – Specification of $j_{max} = k$ means that all values 0, 1, ..., k are supported.		

j_{max} is defined in 12.4.6.2.1.3.

VDSL1 at NOK00078848-849, Table 12-23. See also *id.* NOK00078849, Table 12-24:

Table 12-24/G.993.1 – Interleaver descriptor

Field	Field or macro-field type
I	1 byte
q	1 byte
M_{min}	1 byte
M_{max}	1 byte
NOTE – The four fields are repeated for each interleaver setting.	

458. The VTU-R sends a response message called the R-MSG2 to the VTU-O in response to the O-MSG2.

Table 12-32/G.993.1 – Description of R-MSG2

Field	Field or macro-field type	Remark
Message descriptor	1 byte	See Table 12-12.
Maximal constellation size in upstream ($B_{max,u}$)	1 byte	Maximum number of bits per tone
RS setting supported by VTU-R	1 byte	0: Only mandatory settings 1: All settings
Interleaver settings supported by the VTU-R	1 byte	0x00: Only mandatory settings 0xFF: All settings 0xNN NN = Number of additional (i.e., non-mandatory) settings in hexadecimal (NN $\neq$ 0x00 and NN $\neq$ 0xFF)
Detailed interleaver setting description	0 byte if NN = 0x00 or NN = 0xFF; 4xNN otherwise	Interleaver descriptor (See Table 12-24.)
Maximal power transmitted	1 byte (unsigned)	In dBm by steps of 0.25 dBm
Maximal interleaver memory	3 bytes	In bytes (Note 1)
Maximum number of EOC bytes per frame in upstream	1 byte	Number of EOC bytes per frame
Maximum number of VOC bytes per frame in upstream	1 byte	Number of VOC bytes per frame
Support of express bit swapping	1 byte	0x00: Not supported 0xFF: Supported
j_{max}	1 byte	Specify maximum value of j_{max} supported by the VTU-R (Note 2)
NOTE 1 – The interleaver memory is computed as $M \times I \times (I - 1)$.		
NOTE 2 – Specification of $j_{max} = k$ means that all values from 0, 1, ..., k are supported.		

VDSL1 at NOK00078853, Table 12-32. *See also id.* NOK00078849, Table 12-24 *supra*.

459. Furthermore, in VDSL1 explicit messages about memory were exchanged. By conveying the maximal interleaver memory as shown above in Table 12-32, the R-MSG2 conveys

information related to memory for the interleaving/deinterleaving function. At the very least, therefore, the R-MSG2 described in the VDSL1 standard renders transmitting or receiving a message indicating how the shared memory in the transceiver has been allocated to the one or more of interleaving and deinterleaving functions obvious.

460. Like VDSL1, ADSL2 also discloses an initialization procedure in which the ATU-O (CO) and ATU-R (CPE) transceivers exchange messages concerning their capabilities, including interleaving/deinterleaving capabilities that are eventually used by the CPE to select interleaving parameters. A POSITA would understand that these initialization messages could be used at both the CO and CPE side to deduce the minimum amount of the shared memory in the transceiver required to have been allocated to the interleaving/deinterleaving functions for a typical implementation of convolutional interleaving. It would have been simple and obvious for a POSITA include the explicit memory allocation for interleaving/deinterleaving in these messages. Thus, these message render this limitation obvious.

TPS-TC functions shall be fully configured prior to the initialization of the PMS-TC and PMD functions or be configured after initialization of the PMS-TC and PMS function in a manner that is outside the scope of this Recommendation. The configuration prior to initialization is performed via a G.994.1 MS message. Information may be exchanged prior to the mode select to ascertain capabilities using G.994.1 CL or CLR messages. Most of the information conveyed through G.994.1 messages is dependent on the TPS-TC type and is defined in Annex K.

ADSL2 at 19, § 6.6 “Initialization procedure.”

461. The CO and CPE exchange capabilities in the “G.994.1 capabilities message,” which includes capabilities including maximal supported upstream (CPE to CO) and downstream (CO to CPE) data rates and data rates for each latency path in each direction, which can be used to calculate the interleaving depth supported by the CO and CPE.

Table 7-18/C.992.3 – Format for PMS-TC capability list information

Npar(2) bit	Definition of Npar(2) bit
NTR	This bit is set to a one if the ATU has the capability to transport the NTR signal in the downstream direction.
Spar(2) bit	Definition of related Npar(3) octets
Downstream Overhead Data Rate	Parameter block of 2 octets that describes the minimum message based data rate that is needed by the ATU. The unsigned 6-bit value is the data rate divided by 1000 bits per second minus 1 (covering the range 1 to 64 kbit/s) – see Note.
Upstream Overhead Data Rate	Parameter block of 2 octets that describes the minimum message based data rate that is needed by the ATU. The unsigned 6-bit value is the data rate divided by 1000 bits per second minus 1 (covering the range 1 to 64 kbit/s) – see Note.
Downstream PMS-TC latency path #0 supported (always set to 1)	Parameter block of 2 octets that describes the maximum net_max downstream rate supported in the latency path #0. The unsigned 12-bit net_max value is the data rate divided by 4000. The net_max downstream rate shall be greater than or equal to the maximum required downstream data rate for each TPS-TC type that is supported by the ATU.
Upstream PMS-TC latency path #0 Supported (always set to 1)	Parameter block of 2 octets that describes the maximum net_max upstream rate supported in the latency path #0. The unsigned 12-bit net_max value is the data rate divided by 4000. The net_max upstream rate shall be greater than or equal to the maximum required upstream data rate for each TPS-TC type that is supported by the ATU.
Downstream PMS-TC latency path #1 Supported	Parameter block of 4 octets that describes the maximum net_max downstream rate, downstream $R_{1,max}$, and downstream $D_{1,max}$ supported in the latency path #1. The unsigned 12-bit net_max value is the data rate divided by 4000. $R_{1,max}$ is an unsigned 4-bit value and shall be one of the valid R_p values divided by 2. $D_{1,max}$ is an unsigned 3-bit value and shall be the logarithm base 2 of one of the valid D_p values.
Upstream PMS-TC latency path #1 Supported	Parameter block of 4 octets that describes the maximum net_max upstream rate, upstream $R_{1,max}$, and upstream $D_{1,max}$ supported in the latency path #1. The unsigned 12-bit net_max value is the data rate divided by 4000. $R_{1,max}$ is an unsigned 4-bit value and shall be one of the valid R_p values divided by 2. $D_{1,max}$ is an unsigned 3-bit value and shall be the logarithm base 2 of one of the valid D_p values.
Downstream PMS-TC latency path #2 Supported	Parameter block of 4 octets that describes the maximum net_max downstream rate, downstream $R_{2,max}$, and downstream $D_{2,max}$ supported in the latency path #2. The unsigned 12-bit net_max value is the data rate divided by 4000. $R_{2,max}$ is an unsigned 4-bit value and shall be one of the valid R_p values divided by 2. $D_{2,max}$ is an unsigned 3-bit value and shall be the logarithm base 2 of one of the valid D_p values.
Upstream PMS-TC latency path #2 Supported	Parameter block of 4 octets that describes the maximum net_max upstream rate, upstream $R_{2,max}$, and upstream $D_{2,max}$ supported in the latency path #2. The unsigned 12-bit net_max value is the data rate divided by 4000. $R_{2,max}$ is an unsigned 4-bit value and shall be one of the valid R_p values divided by 2. $D_{2,max}$ is an unsigned 3-bit value and shall be the logarithm base 2 of one of the valid D_p values.
Downstream PMS-TC latency path #3 Supported	Parameter block of 4 octets that describes the maximum net_max downstream rate, downstream $R_{3,max}$, and downstream $D_{3,max}$ supported in the latency path #3. The unsigned 12-bit net_max value is the data rate divided by 4000. $R_{3,max}$ is an unsigned 4-bit value and shall be one of the valid R_p values divided by 2. $D_{3,max}$ is an unsigned 3-bit value and shall be the logarithm base 2 of one of the valid D_p values.

Table 7-18/G.992.3 – Format for PMS-TC capability list information

Upstream PMS-TC latency path #3 Supported	Parameter block of 4 octets that describes the maximum net_max upstream rate, upstream R_{3_max} and upstream D_{3_max} supported in the latency path #3. The unsigned 12-bit net_max value is the data rate divided by 4000. R_{3_max} is an unsigned 4-bit value and shall be one of the value R_p values divided by 2. D_{3_max} is an unsigned 3-bit value and shall be the logarithm base 2 of one of the valid D_p values.
NOTE – By construction of the ADSL2 framing, the message-based overhead data rate is strictly smaller than 64 kbit/s. Hence, the G.994.1 phase of initialization should not request a minimum value of 64 kbit/s.	

ADSL2 at 48-49, Table 7-18. After the G.994.1 capabilities message is sent, the CO sends the C-PARAM message to the CPE, which includes parameter selections for the upstream (CPE to CO) direction.

The information in C-PARAM includes:

- The latency path MSGLP to carry the upstream message-oriented portion of the overhead channel.
- Assignment of upstream frame bearers to upstream latency paths.
- The number of message octets MSGc included in the upstream overhead structure.
- B_{pn} for each upstream latency path and frame bearer.
- M_p for each upstream latency path.
- R_p for each upstream latency path.
- D_p for each upstream latency path.
- TP for each upstream latency path.
- $Lf3_p, Ln3_p, Lf4_p, Ln4_p$ corresponding to each upstream latency path.

ADSL2 at 229-230, § C.7.10.3 “Exchange phase.” The R-PARAM message is sent in response to the C-PARAM message and includes selected parameters for the downstream (CO to CPE) direction.

The information in R-PARAM includes:

- The latency path MSGLP to carry the downstream message oriented portion of the overhead channel.

- Assignment of downstream frame bearers to downstream latency paths.
- The number of message octets MSGc included in the downstream overhead structure.
- B_{pn} for each downstream latency path and frame bearer.
- M_p for each downstream latency path.
- R_p for each downstream latency path.
- D_p for each downstream latency path.
- TP for each downstream latency path.
- $Lf3_p, Ln3_p, Lf4_p, Ln4_p$ corresponding to each downstream latency path.

This C-PARAMS and R-PARAMS information is represented as a parameter block as in Table C.7-2. The information is transmitted in the order shown during C-PARAM and R-PARAM as described in the PMD initialization procedure.

ADSL2 at 229-230, § C.7.10.3 “Exchange phase.” *See also id.* 231-231, Table C.7-2 (excerpted).

Table C.7-2/G.992.3 – Format for PMS-TC PARAMS Information

Octet Number [i]	Format PMS-TC Bits $[8 \times i + 7 \text{ to } 8 \times i + 0]$	Description
Octet 0	[pfff 00bb] bit 7 to 0	The bits bb encode the value of MSG_{LP} . MSG_{LP} indicates the latency path in which the message-based overhead information is to be transmitted. The values 00, 01, 10, and 11 correspond to latency path #0, #1, #2, #3, respectively. The bits ffff encode the initialization success/failure code as defined in this clause. The bit p is the probing bit. A value 1 indicates that the current initialization is used for automode probing. A value 0 indicates that the current initialization is normal initialization.
Octet 1	[cccc dddd] bit 7 to 0	The bits cccc are set to 0000, 0001, 0010, or 0011 if the frame bearer #0 is to be carried in latency path #0, #1, #2, or #3 respectively. The bits cccc are set to 1111 if $type_0$ is zero (i.e., disabled frame bearer, see Table 6-1). The bits dddd describe where the frame bearer #1 is to be carried using the same encoding method as cccc.
Octet 2	[eeee ffff] bit 7 to 0	The bits eeee and ffff describe where the frame bearers #2 and #3, respectively, are to be carried using the same encoding method as cccc of octet 1.
Octet 3	[gggg gggg] bit 7 to 0	The bits gggggggg encode the value of MSG_C , the number of octets in the message-based portion of the overhead structure. The latency path # MSG_{LP} is used to transport the message based overhead information.
Octet 4	[hhhh hhhh] bit 7 to 0	The bits hhhhhhhh give the number of octets from bearer #0 per Mux Data Frame being transported. This value is zero or the non-zero value from the value of the set $\{B_{00}, B_{10}, B_{20}, B_{30}\}$.
Octet 5	[iiii iii] bit 7 to 0	The bits iiiiii give the number of octets from bearer #1 per Mux Data Frame being transported. This value is zero or the non-zero value from the value of the set $\{B_{01}, B_{11}, B_{21}, B_{31}\}$.
Octet 6	[jjjj jjj] bit 7 to 0	The bits jjjjjj give the number of octets from bearer #2 per Mux Data Frame being transported. This value is zero or the non-zero value from the value of the set $\{B_{02}, B_{12}, B_{22}, B_{32}\}$.
Octet 7	[kkkk kkk] bit 7 to 0	The bits kkkkkk give the number of octets from bearer #3 per Mux Data Frame being transported. This value is zero or the non-zero value from the value of the set $\{B_{03}, B_{13}, B_{23}, B_{33}\}$.
Octet 8	[mmmm mmm] bit 7 to 0	The bits mmmmmmm give the value of M_P for latency path #0. They are always present and set to zero if not used.
Octet 9	[tttt ttt] bit 7 to 0	The bits tttttt give the value of T_P for latency path #0. They are always present and set to zero if not used.
Octet 10	[rrrr 0DDD] bit 7 to 0	The bits rrrr0DDD give the value of R_P and D_P for latency path #0. The rrrr and DDD bits are coded as defined in Table 7-18. They are always present and set to zero if not used.
Octet 11	[llll lll] bit 7 to 0	The bits llllll give the LSB of the value of $Lf3_P$ for latency path #0. They are always present and set to zero if not used.
Octet 12	[llll lll] bit 15 to 8	The bits llllll give the MSB of the value of $Lf3_P$ for the latency path #0. These are always present and set to zero if not used.

462. The values exchanged during initialization allow the CO and CPE to deduce the minimum amount of the shared memory needed to support the selected interleaving parameters for each frame.

Table C.7-1/G.992.3 – Derived Characteristics of the ATU data frame

K_p	The number of octets per Mux Data Frame in latency path function # p is always $K_p = \sum_{i=0}^{N_{sc}-1} B_{p,i} + 1$
$N_{FEC,p}$	The number of octets per FEC Data Frame and Interleaved FEC Data Frame in latency path function # p is always $N_{FEC,p} = M_p \times K_p + P_F$
L_p	Average number of bits per data symbol $L_p = (96 \times L_f 4_p + 30 \times L_f 3_p + 144 \times L_n 4_p + 70 \times L_n 3_p) / 340$
S_p	Not accounting for the interleaving procedure, the number of PMD.Bits.request primitives (and correspondingly the number of PMD symbols) over which the FEC Data Frame spans is: always $S_p = \frac{8 \times N_{FEC,p}}{L_p}$. The value of S_p may represent a non-integer value.

ADSL2 at 228, Table C.7-1 (excerpted).

463. By conveying information used to calculate the number of octets per interleaved data frame, the C-PARAM and R-PARAM messages render obvious that a message transmitted during initialization indicates how the memory has been allocated to the interleaving and/or deinterleaving function in the transceiver. For example, as described in §§ VII.A.6, a POSITA would have understood that exchanging messages to enable inter-operability between communicating transceivers was well-known in the prior art. One example of this sort of message is the message conveying the interleaving memory requirements described in Aramvith. § IX.G.

464. VDSL1 and ADSL2 also both use retransmission to ensure that initialization messages are correctly received. For example, ADSL2 uses ACK or NACK messages to request retransmission of an initialization message or message segment that is not correctly received. This

helps ensure that the ADSL2 system will reach Showtime even in the presence of impulse noise or other channel errors because, during initialization, messages are not protected using interleaving.

At the start of the Exchange Phase, the ATU-C shall transition to the C-TREF1-LD state (in which C-TREF symbols shall be transmitted until the first R-MSGx-LD message is received).

If the ATU-C receives an R-MSGx-LD message, the ATU-C shall transition to the C-ACK or C-NACK state within 128 symbols from the end of the R-MSGx-LD state. If the R-MSGx-LD message is successfully received, the ATU-C shall transition to the C-ACK state (in which a positive acknowledgment C-ACK message shall be transmitted). Instead, if a decoding error occurs (i.e., the CRC locally computed at the ATU-C does not correspond to the CRC transmitted by the ATU-R), the ATU-C shall transition to the C-NACK state.

The C-ACK message shall be represented by the "01010101" octet and shall be transmitted over 64 symbol periods using the same modulation technique as the loop diagnostics information bearing messages. No CRC shall be added to the C-ACK message. In the C-NACK state, the ATU-C shall transmit 64 C-TREF symbols. Note that from the ATU-R's perspective, this is equivalent to the ATU-C not responding to the R-MSGx-LD message.

At the end of the C-ACK or C-NACK state, the ATU-C shall transition to the C-TREF2-LD state (in which 256 C-TREF symbols shall be transmitted). During the C-TREF2-LD state, the ATU-R transitions to the R-QUIET2-LD state (because the R-ACK message is successfully received and no more R-MSGx-LD messages remain to be transmitted) or the ATU-R transitions to the R-SEGUE-LD state (because no, or a corrupted C-ACK message is received or more R-MSGx-LD messages remain to be transmitted). At the end of the C-TREF2-LD state, the ATU-C shall transition to the C-SEGUE-LD state (if the ATU-R has transitioned to the R-QUIET2-LD state) or shall return to the C-TREF1-LD state (if the ATU-R has returned to the R-SEGUE-LD state).

Note that, as a result of a corrupted C-ACK message, the ATU-C could successfully receive the same message twice. In this case, the ATU-C shall ignore the second identical (same Sequence Number) message.

ADSL2 at 158-159, § 8.15.5.1.2 "Message flow, acknowledgement and retransmission."

465. VDSL1 uses a similar scheme to ensure that initialization messages are correctly received and the transceiver reaches Showtime.

12.4.2.6 Segmentation of messages

Some messages could potentially be large, and even larger than the maximal allowed frame size of an HDLC frame (1026 octets). It shall therefore be possible to segment messages before transmission. In order to do this, all messages transmitted during initialization shall receive a number that counts the message. This number shall be stored in one byte and will wrap around in case of overflow. The value 0 shall not be used because it has a special meaning (see later). This means that 255 shall be followed by 1.

This "message index" shall be transmitted in the Address Field of the HDLC frame (see Figure 12-3). The message index allows to track lost messages and to request the retransmission of a particular message. The index shall initially be set to one and shall be incremented with one after the transmission of a message. The index shall not be incremented in case of a REPEAT-REQUEST. The counting of messages shall start when the transmission starts using RQ mode instead of automatic repeat (AR) mode.

A segmentation index (1 byte) shall be included in the Control Field of the HDLC frame. The four MSBs of this field shall indicate the number of segments that make up the total message. The four LSBs shall indicate the index of the current segment. For instance a value 0x93 indicates the third segment of a total of nine. In case the message is not segmented, the value of the field shall be 0x11.

The REPEAT-REQUEST message will behave differently from the other messages. The message index counter shall not be increased when a REPEAT-REQUEST message is sent. Also, the meaning of the message index and the segmentation index is different in this case.

The message index of the REPEAT-REQUEST message shall contain the message index of the message that should be retransmitted. The default value 0 indicates that the last unacknowledged message should be sent (this is why 0 shall never be used as message index for any other message). If this message contains several segments, only the last segment shall be retransmitted.

Likewise, the control field shall contain the segment that should be retransmitted. The Information payload of the REPEAT-REQUEST shall consist of one byte containing the value 0x55.

In the initialization procedure a transmitter should never send two consecutive messages without acknowledgement of the first message. It will always first receive some message from the other side before transmitting again. Therefore, an acknowledgement shall be sent for all but the last segment. Typically, the last segment signals the end of the message and it will therefore be acknowledged by the reply to this message. The ACK-SEG-message (see Table 12-12) shall be used to acknowledge the reception of the other segments. The ACK-SEG-message shall have its own message and segment index, which shall not refer to the segmented message that is being sent.

Once acknowledged, messages (or segments) are not expected to be retransmitted again. Also, a REPEAT-REQUEST that contains a number that is higher than the most recent value of the counter shall be ignored.

In AR mode, segmentation shall be done in the same way but there will be no acknowledgement (ACK-SEG) between different segments of the same message. Segments shall be sent in order.

VDSL1 at NOK00078835-836, § 12.4.2.6 “Segmentation of messages.”

466. However, ADSL2 also separately provides for an overhead channel that uses the “sync octets” in data packets to carry overhead information.

7.8.2 Overhead channel access

Each latency path that is enabled carries an overhead channel structure. Various primitives and messages are signalled over these overhead channels via the overhead channel access procedures described in this clause.

7.8.2.1 Overhead channel structure

Each latency path that is enabled carries an overhead channel to be transported in the sync octets. Generally, each overhead channel can contain a CRC portion, a bit-oriented portion, and a message-oriented portion over a repeating sequence of sync octets of length SEQp. The specific structure of the overhead channel for latency path #p shall have one of four formats as displayed in Table 7-14 depending upon the value of the derived parameter SEQp.

ADSL2 at 39-40, §§ 7.8.2, 7.8.2.1 “Overhead channel access” and “Overhead channel structure.”

467. These overhead messages, if too long to carry in a single set of sync octets, are broken into segments and acknowledged separately.

Each transmitted message segment shall be acknowledged by the far-end with a segment acknowledge message, except the last one. The last command message segment shall be acknowledged by the far-end with the appropriate response message (identical to the non-segmented case). The last response message segment shall not be acknowledged. Table 7-17a defines the segment acknowledge message. The following shall apply to all encapsulated segment acknowledge messages:

- HDLC frame shall contain 5 message octets ($P = 5$);
- The two least significant bits of the address field shall be identical to the corresponding bits in the address field of the acknowledged message segment. All other bits of the address field shall be set to 02;

- The second least significant bit of the control field shall either indicate a command message (request to continue the response, e.g., L2 grant) or a response message (request to continue the command, e.g., OLR);
- The least significant bit of the control field shall toggle for each acknowledgement message as it normally toggles for each command/response message (see 7.8.2.4.2);
- All other bits of the control field shall be set to 02.

Table 7-17a/G.992.3 – Segment acknowledge message

Message octet number	Message octet definition
Octet 1	Message designator 1111 0000 _b for acknowledgement of high priority message segment 1111 0001 _b for acknowledgement of normal priority message segment 1111 0010 _b for acknowledgement of low priority message segment
Octet 2	Message type of segment acknowledge message 01 ₁₆
Octet 3	Acknowledged message segment ID (in range 1 to 7)
Octet 4	Message Designator (first message octet of acknowledged message segment)
Octet 5	Message Type (second message octet of acknowledged message segment)

Every message segment shall be acknowledged by the far-end before the next message segment is transmitted. The segment acknowledge message shall not be segmented.

Timeouts shall be defined as follows:

- For a non-segmented message, timeout shall apply between the last octet of the command message transmitted and the first octet of the response message received.
- For a segmented response message, the timeout shall apply between the last octet of the command message transmitted and the first octet of the first message segment received.
- A timeout corresponding to the command priority shall apply between the last octet of the message segment transmitted and the first octet of the segment acknowledge message received.
- Another timeout corresponding to the command priority shall apply between the last octet of the segment acknowledge message transmitted and the first octet of the next message segment received.

If a timeout expires, the transmitter may repeat the last transmitted message. This repeated message can be an unsegmented command message, a command/response message segment (except the last segment of a response message), or a segment

acknowledge message. Alternately, the ATU may abandon the message after an implementation-specific number of retransmissions.

ADSL2 at 43-46, § 7.8.2.4.3 “Overhead message segmentation.”

468. The data rate of ADSL2’s overhead channel can be derived from information transmitted in the C-PARAM and R-PARAM messages. *See, e.g.:*

The information in C-PARAM includes:

- The latency path MSGLP to carry the upstream message-oriented portion of the overhead channel.
- Assignment of upstream frame bearers to upstream latency paths.
- The number of message octets MSGc included in the upstream overhead structure.
- B_{pn} for each upstream latency path and frame bearer.
- M_p for each upstream latency path.
- R_p for each upstream latency path.
- D_p for each upstream latency path.
- TP for each upstream latency path.
- $Lf3p$, $Ln3p$, $Lf4p$, $Ln4p$ corresponding to each upstream latency path.

The information in R-PARAM includes:

- The latency path MSGLP to carry the downstream message oriented portion of the overhead channel.
- Assignment of downstream frame bearers to downstream latency paths.
- The number of message octets MSGc included in the downstream overhead structure.
- B_{pn} for each downstream latency path and frame bearer.
- M_p for each downstream latency path.
- R_p for each downstream latency path.
- D_p for each downstream latency path.
- TP for each downstream latency path.

- Lf3_p, Ln3_p, Lf4_p, Ln4_p corresponding to each downstream latency path.

This C-PARAMS and R-PARAMS information is represented as a parameter block as in Table C.7-2. The information is transmitted in the order shown during C-PARAM and R-PARAM as described in the PMD initialization procedure.

ADSL2 at 229-230, § C.7.10.3 “Exchange phase.” *See also id.* 228-229, Table C.7-1 (excerpted):

K_p	The number of octets per Mux Data Frame in latency path function # p is always $K_p = \sum_{i=0}^{N_{BC}-1} B_{p,i} + 1$
$N_{FEC,p}$	The number of octets per FEC Data Frame and Interleaved FEC Data Frame in latency path function # p is always $N_{FEC,p} = M_p \times K_p + R_p$.
L_p	Average number of bits per data symbol $L_p = (96 \times Lf4_p + 30 \times Lf3_p + 144 \times Ln4_p + 70 \times Ln3_p) / 340$
S_p	Not accounting for the interleaving procedure, the number of PMD.Bits.request primitives (and correspondingly the number of PMD symbols) over which the FEC Data Frame spans is always $S_p = \frac{8 \times N_{FEC,p}}{L_p}$. The value of S_p may represent a non-integer value.

Overhead rate OR_p of latency path function # p	Overhead rate is always $OR_p = \frac{M_p}{T_p \times S_p} \times 32 \text{ kbit/s} = \frac{M_p \times L_p}{T_p \times (K_p \times M_p + R_p)} \times 4 \text{ kbit/s}$

469. The data rate of the overhead channel, combined with the maximum number of times that an overhead segment can be retransmitted, and the length of the timer that controls whether a timeout retransmission is triggered, together can be used to deduce the minimum amount of memory required to support the overhead channel’s retransmission function. The C-PARAM and R-PARAM messages therefore convey information used to deduce the minimum amount of the shared memory required to support overhead channel retransmission.

470. The C-PARAM and R-PARAM messages in the ADSL2 standard therefore at least render that a message transmitted during initialization indicates how the memory has been allocated between the retransmission function and the interleaving and/or deinterleaving function in the

transceiver obvious. For example, as described in §§ VII.A.6, a POSITA would have understood that exchanging messages to enable inter-operability between communicating transceivers were well-known in the prior art. A POSITA would have understood that it would be advantageous to remove the requirement to perform multiple arithmetic steps to learn how much of the shared memory should be allocated to the packet retransmission function by, for example, explicitly including in the C-PARAM and R-PARAM messages the total memory needed to support the overhead channel's packet retransmission function.

471. To the extent that ADSL2's overhead channel packet retransmission and the initialization messages related to it does not render this limitation obvious, it is nonetheless obvious in view of a POSITA's knowledge and skill in the art. As already described in § VII.A.4 above, retransmission was well-known and routinely implemented in the prior art to ensure that packets are correctly received. For example, retransmission in the form of automatic repeat request (that is, "ARQ") was in use before 1960 and has been a standard tool in communications for years.

It is difficult to date the earliest introduction of ARQ protocols in digital communications. They were almost certainly an immediate consequence of the development of parity-check codes in the early days of digital computers. The next major development in the area came in 1960, when Wozencraft and Horstein [Woz3],[Woz4] described and analyzed a system that allowed for both error correction and error detection with retransmission requests. Their system, now known as a type-I hybrid-AREQ protocol, provided significantly improved performance over the pure ARQ protocols in many applications.

Stephen Wicker, Error Control Systems for Digital Communication and Storage at page 393. Furthermore, as discussed in the background section VIII, 802.16 included multiple options for both interleaving and retransmission of data packets which are decided during initialization.

472. Thus, to the extent that the overhead channel packet retransmission described in ADSL2 is somehow different from the claimed retransmission function, a POSITA would have found it routine to modify ADSL2 to work with a general retransmission function rather than one

that only operates for overhead transmissions. For example, as specifically pointed out in § VII.A.4.a, the TCP/IP standard (which is the protocol used for Internet traffic) uses retransmission. Specifically, the transport control protocol (“TCP”) includes a retransmission mechanism. Both VDSL1 and ADSL2 were widely used by consumers to access the Internet. The amount of memory required to support retransmission has long been studied, and a message exchange during initialization to ensure that both sides (transmitter and receiver) have sufficient memory to support packet retransmission would have been well within the skill of a POSITA. *See, e.g., §§ VII.A.4, VII.A.6, VII.B supra.*

473. Accordingly, VDSL1 in view of ADSL2 renders obvious “wherein a message transmitted during initialization indicates how the memory has been allocated between the retransmission function” (ADSL2 describes initialization messages that can be used to deduce the minimum amount of memory necessary for the overhead retransmission function, while a POSITA would have found it obvious to use both VDSL1 and ADSL2 with retransmission to protect non-overhead data, including as described in the TCP/IP protocol, and would have understood that the transceivers must exchange initialization data like that set out in the ADSL2 C-PARAM and R-PARAM messages to ensure they both have sufficient memory to support packet retransmission) “and the interleaving and/or deinterleaving function in the transceiver” (VDSL1 and ADSL2 both describe initialization messages that can be used to deduce the minimum amount of memory necessary to support the interleaving/deinterleaving functions).

ii TR 25.896 and TS 25.401 in View of Eriksson

474. TS 25.401 and TR 25.896 in view of Eriksson renders obvious “wherein a message transmitted during initialization indicates how the memory has been allocated between the retransmission function and the interleaving and/or deinterleaving function in the transceiver.”

475. Whenever there is a choice between alternative functionalities, the transceivers must find a way to agree, and the typical approach is a message indicating which choice will be selected. In order for a retransmission function to work properly, both sides have to agree about the exact retransmission policy including how far back in time information will be stored on the transmitter side for possible retransmission. A POSITA would be able to deduce the minimum amount of memory to support retransmission from this information. Similarly, for interleaving to work properly both sides have to agree on the specific interleaver implementation. From this information, a POSITA would be able to deduce the minimum amount of memory required to implement the interleaver for a typical implementation.

476. Each choice has memory requirements associated with it, and including those memory requirements in the message would be an obvious choice with predictable results. Specifically, whenever there are multiple retransmission choices such as the choice between the number of retransmission processes to employ or the transmission time interval, the message informing the other transceiver of this choice could obviously include its memory requirements. Similarly, whenever there is a choice between whether to have interleaving or between multiple interleavers, the message indicating whether and how to perform interleaving could obviously include the associated memory requirements.

477. Specifically, TR 25.896 describes that either the Node B or the UE will determine parameters necessary to use hybrid ARQ and that the end that chooses those parameters (either Node B or UE) must signal the other end with the selected parameters. For example, TR 25.896 describes the use of stop-and-wait (“SAW”) hybrid ARQ processes. In SAW hybrid ARQ, if (for example) packet 4 in hybrid ARQ process 3 is not received correctly, hybrid ARQ process 3 will retransmit packet 4 until it is correctly received (which is signaled by an ACK or

acknowledgement). Thus the number of hybrid ARQ processes, N , must be set. TR 25.896 describes that tradeoffs between using a large number of hybrid ARQ processes, which may cause significant overhead with the number of ACKs and NAKs being sent, but also ensures more continuous data is provided because the other hybrid ARQ processes can continue transmitting data even when one process is hung up retransmitting a packet.

If E-DCH utilizes physical layer HARQ, there is a need to transmit ACK/NACK signaling in a downlink physical channel. N defines the minimum number of HARQ processes required to provide continuous transmission. However, increasing the number of HARQ channels also adds to round trip time and thus N cannot be arbitrarily large. A compromise between round trip time and processing time is of main importance when considering the selection of N .

If the available time for downlink signaling and UE/Node B processing is made long enough through suitable selection of N , ACK/NACK could be embedded in existing Rel'99 downlink channel structure, i.e. within a 10 ms TTI. Another option is to reserve a specific field shorter than 10 ms time period, in a downlink physical channel for ACK/NACK as is done in HS-DPCCH for uplink in Rel'5 HSDPA. The downlink ACK/NACK field length is naturally independent of TTI length in uplink.

Figure 8.2.1 depicts the general concept of timing for E-DCH HARQ process. After having received transport block(s) on E-DCH the Node B has T_{NBP} for processing and sending acknowledgement to the UE. In here no assumption is made on which downlink physical channel the ACK/NACK in DL would be sent. Based on the acknowledgement and possible other information provided by the UTRAN, the UE decides whether it resends the transport block(s) or transmits new transport block(s). The processing time available for the UE between receiving the acknowledgement and transmitting the next TTI in the same HARQ process is T_{UEP} .

The length of the acknowledgement field in DL directly affects the available processing time in Node B and UE. The length of the acknowledgement field might also affect the required power offset for transmitting it, relative to DL DPCCH, depending on the scheme. With 10 ms TTI and high enough N , acknowledgement could e.g. be embedded in existing multiplexing structure within a 10 ms TTI. This might allow more space for coding and smaller power offset for transmitting ACK/NACK than in the case where ACK/NACK is inserted into downlink physical channel within a shorter time period than 10ms.

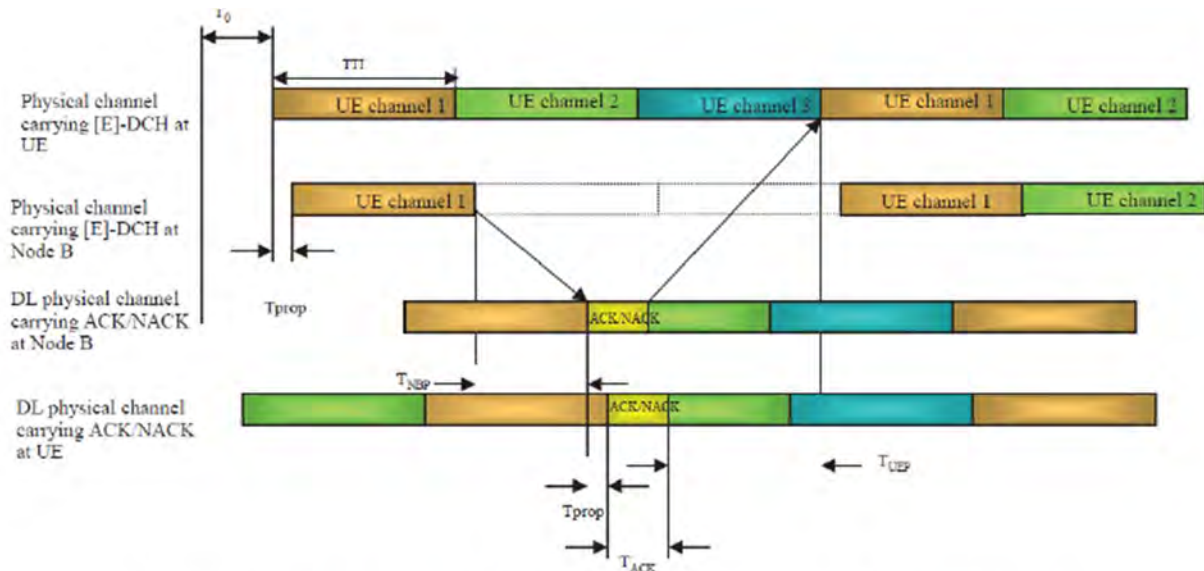


Figure 8.2.1. HARQ timing schematic for N=3, TTI=10 ms, as an example.

Table 8.2.1 presents some estimations for available processing time TTI lengths 10 ms and 2 ms, with N=2,3,4,5. The timing calculations assume a roundtrip delay of 0.1 ms. The acknowledgement signal from the Node B may be spread over one of more slots. However, the longer TACK becomes, the less processing time there is available for UE and RNS. For TTI=10 ms case, a TACK = 10 ms is possible if N=3 or larger. With TTI=2 ms, TACK necessarily has to be shorter.

Table 8.2.1. Examples of UE and Node B processing times with E-DCH

TTI length (ms)	N	T _{ack} (ms)	T _{NBP} +T _{UEP} (ms)
10	2	2	7.9 (0.8xTTI)
10	3	2	17.9 (1.8xTTI)
10	3	10	9.9 (1.0xTTI)
10	4	2	27.9 (2.8xTTI)
10	4	10	19.9 (2.0xTTI)
2	4	2 (3 slots)	3.9 (1.95 x TTI)
2	5	2/3 (1 slot)	7.23 (3.6xTTI)
2	5	4/3 (2 slots)	6.56 (3.3xTTI)
2	6	4/3 (2 slots)	8.56 (4.3xTTI)
2	6	2 (3 slots)	7.90 (4.0xTTI)
2	7	2 (3 slots)	9.90 (5.0xTTI)

The table shows examples of the total time available for UE and Node B processing in the case of implicit scheduling. Thus, the figures in Table 8.2.1 represent minimum round trip time. Other methods with e.g. additional control channels would increase the round trip time or reduce available processing time. These methods are investigated separately. Note that the length of the E-DCH TTI also has an impact on the processing time needed. Since a shorter TTI contains fewer bits

than a longer one, the processing load for baseband processing such as interleaving and turbo decoding is smaller and less time is consumed. On the other hand, interleaving gain is impacted when short TTI length is employed.

The choice of TTI and N should be done in conjunction with selecting the structure of the downlink ACK/NAK transmission. Furthermore, the maximum data rate supported will affect the required processing times. Herein, the assumption that maximum data rate would be around 1-2Mbit/s was used.

More detailed analysis of the required processing times are needed in the future, but this gives some rough estimate how the TTI length affects the HARQ physical layer structure. In addition to processing times, important issues to consider are the physical layer structure for sending the L1 signaling in uplink and downlink, and the performance and complexity related to that.

TR 25.896 at NOK00074962-963, § 8.2 “TTI length vs. HARQ physical channel structure.” TR 25.896 does not explicitly discuss a message where the Node B and UE set N, but a POSITA would have understood that N must be agreed on by the Node B and UE before hybrid ARQ retransmission can begin.

478. No matter which end (Node B or UE) decides the parameters used, both the Node B and UE must set aside a sufficient portion of the shared memory (referred in the 3GPP standards as a “buffer”) to support the interleaving applied to messages also subject to hybrid ARQ retransmission. The Node B must set aside data based on the number of retransmissions allowed and the number of hybrid ARQ processes used, which further shows that the Node B and UE must exchange a message setting these parameters; otherwise, the Node B will not be able to estimate the minimum amount of memory to set aside to support hybrid ARQ retransmission. A POSITA would have understood that the minimum amount of memory to be set aside should be set during initialization so that all data transmitted is properly protected from errors.

The principle of hybrid ARQ is to buffer E-DCH TTIs that were not received correctly and consequently combine the buffered data with retransmissions. The actual method of doing soft combining depends on the HARQ combining scheme selected. In Chase combining scheme the receiver always combines the full retransmission of the failed TTI, i.e. the amount of data in the receiver buffer remains the same. In the incremental redundancy schemes the receiver buffers coded

symbols, which introduce new information to the E-DCH TTI transmitted first, i.e. the amount of data to be buffered increases with consecutive retransmissions. However, in practice the buffer in the receiver needs to be dimensioned considering the maximum number of coded bits of the E-DCH TTI after all the incremental redundancy has been introduced. Regardless of the HARQ combining scheme soft combining is done on L1 of Node B before the decoding stage of FEC. Prior to decoding these symbols are soft-valued, i.e. each symbol is represented by two or more bits. Here we call them soft symbols.

For N-process SAW HARQ, the number of soft symbols to be buffered in L1 receiver can be estimated generally as follows, since no new PDUs are transmitted on a subchannel before the previous packet is acknowledged. The receiver has to buffer one E-DCH TTI for each HARQ process and for each UE. The next transmission is either a new packet or a retransmission of an erroneous packet. In either case, the maximum buffering need is N E-DCH TTIs. The actual size of the buffer needed for each E-DCH TTI depends on the HARQ combining scheme as described above. Thus, for N-process stop-and-wait ARQ the L1 buffering at Node B for one UE can be expressed as:

$$buffer = (softsymbols_{TTI} \times N)$$

Table 9.2.4 shows some examples for the soft buffer size required at Node B per UE. It should be noted that soft buffer is needed for all UEs waiting for retransmission at a given time. Thus these values have to be multiplied by the number of active UEs using E-DCH. Depending on particular implementations, the buffer may be shared between UEs and thus the total size of soft buffer can be reduced given that not all active UEs transmit simultaneously using the most demanding format. Also, it may be possible to re-use some existing buffers.

Table 9.2.4 Node B soft buffer size (kilo soft symbols) per UE, for some example values of N (number of HARQ processes) and TTI lengths

	N=4, TTI=10 ms	N=3, TTI=10 ms	N=8, TTI=2 ms	N=6, TTI=2 ms
BPSK, SF=4	38.4	28.8	15.36	11.52
QPSK or 2*BPSK, SF=4	76.8	57.6	30.72	23.04
8PSK or 3*BPSK, SF=4	115.2	86.4	46.08	34.56
2*QPSK or 4*BPSK, SF=4; or QPSK, SF=2	153.6	115.2	61.44	46.08

TR 25.896 at NOK00074991, § 9.2.2.1.1 “Soft buffer at Node B”. Either the RNC or the Node B must also set aside memory for a reordering buffer, which is used to correct the order of packets when they are received out of order due to retransmission. This number is dependent on the number of hybrid ARQ processes, the maximum number of retransmissions allowed (assumed 3 in the below example), and the amount of data sent in a single TTI (essentially, the size of packets being

transmitted). Again, this shows that the Node B and UE must exchange the number of hybrid ARQ processes and the number of retransmissions allowed.

Due to physical/MAC layer (H)ARQ, the MAC-e PDUs may be received in wrong order. Therefore, reordering of the MAC-e PDUs is required either in the Node B or in the RNC. Here the required reordering buffer size per UE is estimated. In the estimation it is assumed that only one high bit rate reordering queue is active for a UE at a time. Furthermore, maximum of 3 retransmissions per MAC-e PDU is assumed and the buffer size is calculated for the worst case where one ARQ process requires retransmissions while the others get blocks through and forward them to reordering queue. The worst case buffer size is then calculated as

$$\text{reorderingbuffer} = [3(N - 1) + 1] \times (\text{bits}/\text{TTI})$$

These values have to be multiplied by the number of E-DCH UEs served by the Node B or by the RNC, depending on the location of the reordering buffer. Depending on implementation and location of the re-ordering buffer, it may be possible to re-use some existing buffers.

Table 9.2.5 Reordering buffer size (kB) per UE, for some example values of N (number of HARQ processes) and TTI lengths

User data rate	N=4, TTI=10 ms	N=3, TTI=10 ms	N=8, TTI=2 ms	N=6, TTI=2 ms
384 kbit/s	4.8	3.36	2.1	1.5
768 kbit/s	9.6	6.72	4.2	3.1
1 Mbit/s	12.5	8.8	5.5	4.0
1.5 Mbit/s	18.8	13.1	8.3	6.0

TR 25.896 at NOK00074991, § 9.2.2.1.2 “Reordering buffer in radio network” and Table 9.2.5.

Both interleaving steps applied to the data will also require a reordering buffer to deinterleave the interleaved data and put the interleaved packets/data in the correct order; as with the hybrid ARQ reordering buffer, the minimum size of the [de]interleaving buffer depends on the amount of data transmitted in a packet and the interleaving depth (which is similar to N, the number of times a packet can be retransmitted).

479. The UE must set aside a sufficient portion of the shared memory (“buffer” in the 3GPP standards) to support the interleaving and retransmission used by hybrid ARQ with incremental redundancy. The minimum amount of memory the UE must set aside to support uplink

hybrid ARQ with incremental redundancy depends on the amount of information transmitted in each TTI.

In addition to buffers in the receiver side, also some buffer at UE side is needed. The retransmission buffer can be either at MAC layer as info bits or at physical layer as coded bits. The total number of coded bits is typically 3 times the number of info bits. Table 9.2.4 shows also the required retransmission buffer size in kbits (assuming that coded bits are buffered). Depending on implementation it may be possible to re-use existing buffers.

TR 25.896 at NOK00074992, § 9.2.2.1.3 “Retransmission buffer in UE.”

480. Messages conveying the parameters set by the UE or Node B and transmitted to the other side (that is, from the UE to the Node B or vice versa) are sufficient for the transceiver to compute the minimum amount of the shared memory in the transceiver (the buffer) that will be used by the hybrid ARQ function by a typical implementation (or by whatever implementation the transceiver has selected). It would be obvious to include this memory requirement in these messages. Including such a message would have predictable results providing information that could have been computed by the other transceiver as well.

481. Eriksson further suggests sending messages indicating the amount of memory available for use by the hybrid ARQ and incremental redundancy functions to provide more direct (and efficient) memory and spectrum usage by the Node B and UE.

MCS changes may or may not be based solely on reported link quality estimates. For example, when incremental redundancy is used and the receiver has limited memory it may be beneficial to increase the robustness of the MCS even though (in a system with unlimited memory) it would decrease throughput. Consider the following scenario. Using a less robust MCS, the number of required retransmissions for successful incremental redundancy combination will be higher. This, in turn, requires a lot of memory. If the receiver runs out of memory, it will begin to discard received blocks that have previously been stored for later incremental redundancy combination. Since the information transmitted using the relatively unrobust MCS probably relies in part upon incremental redundancy combining to achieve acceptable decoding performance, the result may be significant degradation in received signal quality. Hence it may be better under such circumstances to dynamically increase the robustness of the transmission's MCS, e.g., when the receiver starts to run out of memory.

Yet another factor which further complicates this process is the possibility of changing the MCS associated with blocks that are being retransmitted. If link adaptation is used without incremental redundancy, then changing the MCS for retransmissions may be very desirable based upon the measured link quality. On the other hand, if incremental redundancy is employed, using a different MCS may make it impossible to combine the retransmitted block with the originally transmitted block. However, if the link quality changes significantly it still may be desirable to change the MCS even if some of the earlier transmitted data blocks cannot be used in the redundancy combining process at the receiver.

Thus it can be seen that there are many challenges associated with optimizing the manner in which these various techniques are employed. To enable dynamic changing of the MCS during a connection, some form of overhead signalling is necessary between the transmitter and receiver.

Eriksson at NOK00083211, 3:63-4:32.

482. Thus, a POSITA would have understood from TR 25.896 that the UE and Node B transmit (or receive, depending on which end decided the hybrid ARQ and incremental redundancy parameters) a message that describes the ARQ scheme. It would be obvious to include the specific memory allocation of that scheme and to set initial values for those parameters during initialization to ensure that all transmitted data is protected from errors.

483. Accordingly, TR 25.896 and TS 25.401 in view of Eriksson renders obvious “wherein a message transmitted during initialization indicates how the memory has been allocated between the retransmission function and the interleaving and/or deinterleaving function in the transceiver” (TR 25.896 describes that the Node B and UE must exchange messages setting certain parameters that determine the amount of shared memory used by hybrid ARQ versus the amount of memory used by interleaving; a POSITA would understand that these messages are transmitted during the transceiver’s initialization. Eriksson further suggests modifying this messaging to include an explicit indication of how the shared memory should be allocated to the hybrid ARQ retransmission versus the interleaving function, depending on the memory available to the Node B and UE).

2. Claim 18

[18.a] A transceiver capable of packet retransmission comprising:

484. Claim 18 contemplates a receiver as opposed to a transmitter. Because this is two ends of the same retransmission process—that is, a transmitter transmits packets and a receiver receives those packets—this claim is largely duplicative of claim 10 and is thus rendered invalid for the same reasons.

i VDSL1 in View of ADSL2

485. This limitation is identical to limitation [10.a] and is therefore disclosed or rendered obvious for the same reasons set forth with respect to that limitation.

ii TR 25.896 and TS 25.401 in View of Eriksson

486. This limitation is identical to limitation [10.a] and is therefore disclosed or rendered obvious for the same reasons set forth with respect to that limitation.

[18.b] a receiver portion capable of: receiving a plurality of packets,

487. The only difference between this limitation and limitation [10.b] is the fact that this limitation requires a receiver portion capable of receiving a plurality of packets, rather than a transmitter portion capable of transmitting a plurality of packets. This difference does not change my analysis of whether this limitation was disclosed or rendered obvious by the prior art. Indeed, the claims merely describe two ends of the same retransmission process. A system that has a transmitter portion capable of retransmission will therefore have a receiver portion capable of retransmission. As it relates to this claim, each of the prior art references disclose a system that transmits and receives a plurality of packets.

i VDSL1 in View of ADSL2

488. VDSL1 discloses a VTU-O and VTU-R—which are two ends of retransmission—for “Very high speed digital subscriber line transceivers.” Likewise, ADSL2 discloses

“Asymmetric Digital Subscriber Line (ADSL) transceivers on a metallic twisted pair that allows high-speed data transmission between the network operator end (ATU-C) and the customer end (ATU-R).” Accordingly, for the reasons discussed above with respect to limitation [10.b], and because these references discuss both ends of the retransmission process, this limitation is disclosed or rendered obvious.

ii TR 25.896 and TS 25.401 in View of Eriksson

489. TR 25.896 discloses both Node Bs (base stations) and UEs (user equipment), where the base station is one side of the retransmission scheme and the user equipment is on the other end. *See* TR 25.896 at NOK00074951-952, § 7.2.1 “Hybrid ARQ – General”. Likewise, Eriksson describes a transmitter to transmit bits as well as a “receiver” to decode those bits. *See* Eriksson at NOK00083211, 3:23-38. Accordingly, for the reasons discussed above with respect to limitation [10.b], this limitation is disclosed or rendered obvious.

[18.c] identifying at least one packet of the plurality of packets as a packet that should be retransmitted and allocating a memory between a retransmission function and an interleaving and/or deinterleaving function

490. This limitation is identical to limitation [10.c] and is therefore disclosed or rendered obvious for the same reasons set forth with respect to that limitation. Although the memory allocation might differ—as the receiver requires a memory allocation sufficient to store packets based on a corrupted packet (such as storing those packets around the corrupted packet or storing the actual corrupted packet) while the transmitter requires a memory allocation sufficient to store packets while waiting for an ACK message—because it is the same system the disclosure remains consistent.

i VDSL1 in View of ADSL2

491. ADSL2 uses retransmission to ensure that certain overhead messages (or segments of those messages) are received correctly. ADSL2 accomplishes this on both the transmitter and

receiver side. For example, as explained with respect to limitation [10.c], ADSL2 uses a timer and required acknowledgment or negative acknowledgment (ACK or NACK) signals to decide when to discard a segment or message being held for retransmission versus when to retransmit that segment/message.

Each transmitted message segment shall be acknowledged by the far-end with a segment acknowledge message, except the last one. The last command message segment shall be acknowledged by the far-end with the appropriate response message (identical to the non-segmented case). The last response message segment shall not be acknowledged. Table 7-17a defines the segment acknowledge message. The following shall apply to all encapsulated segment acknowledge messages:

- HDLC frame shall contain 5 message octets ($P = 5$);
- The two least significant bits of the address field shall be identical to the corresponding bits in the address field of the acknowledged message segment. All other bits of the address field shall be set to 02;
- The second least significant bit of the control field shall either indicate a command message (request to continue the response, e.g., L2 grant) or a response message (request to continue the command, e.g., OLR);
- The least significant bit of the control field shall toggle for each acknowledgement message as it normally toggles for each command/response message (see 7.8.2.4.2);
- All other bits of the control field shall be set to 02.

Table 7-17a/G.992.3 – Segment acknowledge message

Message octet number	Message octet definition
Octet 1	Message designator 1111 0000 _b for acknowledgement of high priority message segment 1111 0001 _b for acknowledgement of normal priority message segment 1111 0010 _b for acknowledgement of low priority message segment
Octet 2	Message type of segment acknowledge message 01 ₁₆
Octet 3	Acknowledged message segment ID (in range 1 to 7)
Octet 4	Message Designator (first message octet of acknowledged message segment)
Octet 5	Message Type (second message octet of acknowledged message segment)

Every message segment shall be acknowledged by the far-end before the next message segment is transmitted. The segment acknowledge message shall not be segmented.

Timeouts shall be defined as follows:

- For a non-segmented message, timeout shall apply between the last octet of the command message transmitted and the first octet of the response message received.
- For a segmented response message, the timeout shall apply between the last octet of the command message transmitted and the first octet of the first message segment received.
- A timeout corresponding to the command priority shall apply between the last octet of the message segment transmitted and the first octet of the segment acknowledge message received.
 - Another timeout corresponding to the command priority shall apply between the last octet of the segment acknowledge message transmitted and the first octet of the next message segment received.
 - If a timeout expires, the transmitter may repeat the last transmitted message. This repeated message can be an unsegmented command message, a command/response message segment (except the last segment of a response message), or a segment acknowledge message. Alternately, the ATU may abandon the message after an implementation-specific number of retransmissions.

ADSL2 at 43-46, § 7.8.2.4.3 “Overhead message segmentation.”

492. ADSL2 provides for similar retransmission for initialization messages. *See, e.g.:*

At the start of the Exchange Phase, the ATU-C shall transition to the C-TREF1-LD state (in which C-TREF symbols shall be transmitted until the first R-MSGx-LD message is received).

If the ATU-C receives an R-MSGx-LD message, the ATU-C shall transition to the C-ACK or C-NACK state within 128 symbols from the end of the R-MSGx-LD state. If the R-MSGx-LD message is successfully received, the ATU-C shall transition to the C-ACK state (in which a positive acknowledgment C-ACK message shall be transmitted). Instead, if a decoding error occurs (i.e., the CRC locally computed at the ATU-C does not correspond to the CRC transmitted by the ATU-R), the ATU-C shall transition to the C-NACK state.

The C-ACK message shall be represented by the "01010101" octet and shall be transmitted over 64 symbol periods using the same modulation technique as the loop diagnostics information bearing messages. No CRC shall be added to the C-ACK

message. In the C-NACK state, the ATU-C shall transmit 64 C-TREF symbols. Note that from the ATU-R's perspective, this is equivalent to the ATU-C not responding to the R-MSGx-LD message.

At the end of the C-ACK or C-NACK state, the ATU-C shall transition to the C-TREF2-LD state (in which 256 C-TREF symbols shall be transmitted). During the C-TREF2-LD state, the ATU-R transitions to the R-QUIET2-LD state (because the R-ACK message is successfully received and no more R-MSGx-LD messages remain to be transmitted) or the ATU-R transitions to the R-SEGUE-LD state (because no, or a corrupted C-ACK message is received or more R-MSGx-LD messages remain to be transmitted). At the end of the C-TREF2-LD state, the ATU-C shall transition to the C-SEGUE-LD state (if the ATU-R has transitioned to the R-QUIET2-LD state) or shall return to the C-TREF1-LD state (if the ATU-R has returned to the R-SEGUE-LD state).

Note that, as a result of a corrupted C-ACK message, the ATU-C could successfully receive the same message twice. In this case, the ATU-C shall ignore the second identical (same Sequence Number) message.

ADSL2 at 158-159, § 8.15.5.1.2 “Message flow, acknowledgement and retransmission.”

493. ADSL2 requires that the receiver save either the packets that are corrupted, or the packets surrounding the corrupted packet. Therefore, ADSL2 allocates some memory to retransmission for the receiver, as described in limitation [10.c].

494. Likewise, VDSL1 describes identifying at least one packet of the plurality of packets as a packet that should be retransmitted. VDSL1 uses an initialization procedure to set up the link between the VTU-O and VTU-R, during which the transceivers exchange parameters. In this procedure, messages are sent over the Special Operation Channel (SOC) and retransmission is utilized so that VDSL1 identifies at least one packet of the plurality of packets as a packet that should be retransmitted.

495. Additionally, Section 12.4.1 provides an overview of this initialization:

12.4.1 Overview

Initialization of a VTU-O/VTU-R pair includes a variety of tasks. The set of tasks consist of:

- Definition of a common mode of operation;
- Synchronization (sample clock alignment and symbol alignment);
- Transfer of frequency band allocation and PSD mask information from the VTU-O to the VTU-R;
- Channel identification;
- Noise identification;
- Calculation of bit loading and energy tables;
- Exchange of parameters (RS settings, interleaver parameters, VOC settings, bit loading and energy tables, ...).

VDSL1 at NOK00078832, § 12.4.1 “Overview.”

496. VDSL1 uses a repeat request (RQ) which we have referred to as a NACK above, when operating in RQ mode to request a retransmission on the SOC. This is described in Section 12.4.2.1 as excerpted below.

12.4.2 SOC protocol

12.4.2.1 Message format

The SOC shall use HDLC-like format with byte stuffing to delineate the messages as specified in ITU-T Rec. G.994.1. A reliable transmission shall be insured by using either an automatic repeat (AR) mode or a repeat request (RQ) mode. The maximum size of an HDLC frame shall be 1026 bytes (this also defines the maximum size of a SOC message segment). This is the size of the payload before octet stuffing and addition of any flags.

In the AR mode, the message encapsulated in the HDLC frame shall be automatically repeated. At least four idle flags (0x7E) shall be inserted between successive frames.

In the RQ mode, the messages encapsulated in HDLC frame shall be sent once. However, the VTU expecting the message shall have the possibility to request the remote side to repeat it by sending a REPEAT REQUEST message. This operation is necessary when the expected message has a wrong Check Sequence or when a timeout has expired. After two unsuccessful REPEAT REQUEST, the initialization shall be aborted. This means the initiating side will restart the G.994.1 handshake after a silent period. After a number of unsuccessful attempts, the modems shall stop all attempts. The number of attempts that is made before a final abort of the initialization shall be chosen by the initiating modem.

VDSL1 at NOK00078833, § 12.4.2.1 “Message format.”

497. VDSL1 requires that the receiver save either the packets that are corrupted, or the packets surrounding the corrupted packet. Therefore, as explained with respect to limitation [10.c], VDSL1 allocates some memory to retransmission.

ii TR 25.896 and TS 25.401 in View of Eriksson

498. TR 25.896 and Eriksson both disclose a transceiver that uses hybrid ARQ retransmission and therefore identify errored packets which should be retransmitted. In addition, both TR 25.896 and Eriksson disclose allocating a memory between a retransmission function and an interleaving and/or deinterleaving function (the UE stores packets for later retransmission packets if requested, while the Node B stores received packets for eventual soft combining, and the UE interleaves packets before transmitting them while the Node B deinterleaves received packets). Here, the receiver is Node B, and that Node B requires a memory to store the received packets.

[18.d] wherein the memory is allocated between the retransmission function and the interleaving function and/or the deinterleaving function in accordance with a message received during an initialization of the transceiver and

499. This limitation is a simple rewording of limitation [10.e]. The differences between this limitation and [10.e] do not change my analysis of whether this limitation was disclosed or rendered obvious by the prior art. Indeed, if the memory is allocated between the retransmission function and the interleaving function it is referencing the transmitter side of the system. If the memory is allocated between the retransmission function and the deinterleaving function it is referencing the receiver side of the system.

iii VDSL1 in View of ADSL2

500. VDSL1, for example, discloses an initialization procedure in which the VTU-R and the VTU-O exchange messages concerning their capabilities, including their respective interleaving capabilities and, eventually, selected interleaving parameters. As discussed above, the typical triangular implementation of convolutional interleaving has a well-defined minimum memory requirement. Thus, these messages enable the transceivers at both the VTU-O and VTU-R side to deduce the minimum amount of the shared memory in the transceiver required to have been allocated to the interleaving/deinterleaving functions for this typical implementation. Likewise, ,

ADSL2 discloses an initialization procedure in which the ATU-O (CO) and ATU-R (CPE) transceivers exchange messages concerning their capabilities, including interleaving/deinterleaving capabilities that are eventually used by the CPE to select interleaving parameters. Thus, because VDSL1 and ADSL2 disclose this limitation with respect to both the transmitter and receiving side, for the same reasons discussed in limitation [10.e], this limitation is disclosed.

iv TR 25.896 and TS 25.401 in View of Eriksson

501. As explained in limitation [10.e], whenever there is a choice between alternative functionalities, the transceivers must find a way to agree, and the typical approach is a message indicating which choice will be selected. In order for a retransmission function to work properly, both sides have to agree about the exact retransmission policy including how far back in time information will be stored on the transmitter side for possible retransmission. A POSITA would be able to deduce the minimum amount of memory to support retransmission from this information. Similarly, for deinterleaving to work properly, both sides have to agree on the specific deinterleaver implementation. From this information, a POSITA would be able to deduce the minimum amount of memory required to implement the deinterleaver for a typical implementation.

502. Specifically, TR 25.896 describes that either the Node B or the UE will determine parameters necessary to use hybrid ARQ and that the end that chooses those parameters (either Node B or UE) must signal the other end with the selected parameters. As explained in limitation [18.c], Node B is the receiver, and thus the parameters may be determined based on a message received during initialization.

[18.e] wherein at least a portion of the memory may be allocated between the retransmission function and the interleaving and/or deinterleaving function at any one particular time depending on the message.

503. As already explained, because the receiver and transmitter are a part of the same system, my analysis largely does not change. The only significant difference between this limitation

and limitation [10.d] is the fact that this limitation includes “depending on the message.” This difference does not change my analysis of whether this limitation was disclosed or rendered obvious by the prior art. Indeed, here, the only difference might be the fact that the memory allocation could be different. That, however, was already addressed with respect to limitation [18.d], which I hereby incorporate.

i VDSL1 in View of ADSL2

504. This limitation is effectively identical to limitations [10.d] and [18.d], which I incorporate. As explained in limitation [18.d], VDSL1 and ADSL2 both disclose allocating memory between a retransmission function and a deinterleaver on a receiver based on a message. As I explained in limitation [10.d], it was known and obvious to make such an allocation at any one particular time, and my conclusion is not affected by the fact that this occurs at the receiver.

ii TR 25.896 and TS 25.401 in View of Eriksson

505. This limitation is effectively identical to limitations [10.d] and [18.d], which I incorporate. As I explained in limitation [10.d], a POSITA would have understood that Eriksson’s suggestion to change the initial allocation of memory between interleaving/deinterleaving and hybrid ARQ retransmission has the advantage of allowing a system (such as TR 25.896) to more easily respond to changes in link quality. A POSITA would therefore be motivated to modify the system of TR 25.896 based on Eriksson so that the initial hybrid ARQ/interleaving setup (described further with respect to limitation [10.e]) could be modified to respond to changing channel conditions. A POSITA would have understood that allowing the Node B to provide similar signaling to the UE, for example when the Node B is experiencing severe interference, i.e., at any one particular time, from other UEs and it would prefer the UE use retransmission rather than increase transmit power (or otherwise modify its MCS or FEC scheme to increase the likelihood of correct packet reception).

C. The Asserted Claims of the '348 Patent Are Obvious

506. The asserted claim of the '348 Patent is claim 1. The following charts summarize my citations to the prior art for the asserted claim:

Ground	Claim	Summary Claim Chart
VDSL	1	Exhibit B-1
Chen in view of ADSL2	1	Exhibit B-2
Yap in view of VDSL1	1	Exhibit B-3

507. The asserted claim of the '348 Patent is obvious. I address each limitation of the asserted claim in turn below.

1. Claim 1

[1.pre] An apparatus comprising:

i VDSL1

508. To the extent that the preamble is limiting, VDSL1 discloses an apparatus.

Figure 5-1 shows the reference configuration used for G.993.1. It is essentially a Fibre to the Node architecture with an Optical Network UnZit (ONU) sited in the existing metallic access network (or at the serving Local Exchange or Central Office). The first architectural model covers Fibre-to-the-cabinet (FTTCab) type of deployment; the second one is Fibre-to-the-exchange (FTTEx) type of deployment. Existing unscreened twisted metallic access wire-pairs are used to convey the signals to and from the customer's premises.

VDSL1 at NOK00070775-776, § 5.1 "General reference models." *See also id.* Figs 5-1 and 5-2:

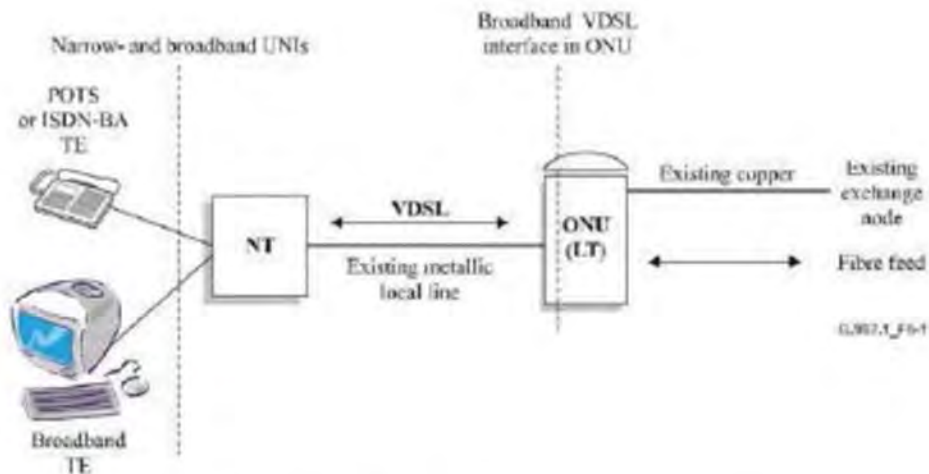


Figure 5-1/G.993.1 – General reference model

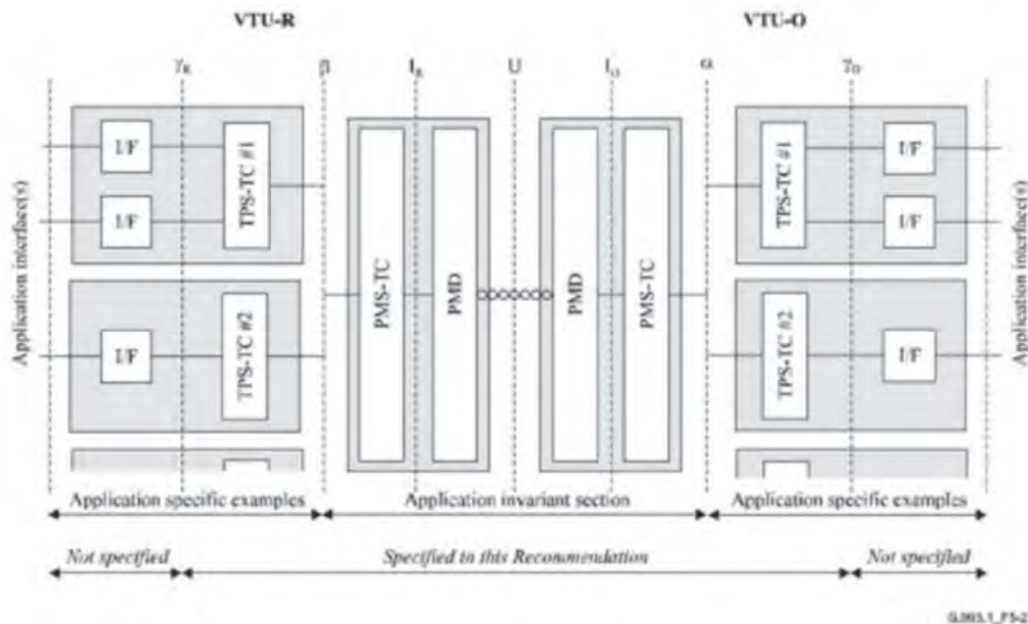


Figure 5-2/G.993.1 – VTU-x functional reference model

ii Chen in view of ADSL2

509. To the extent that the preamble is limiting, Chen discloses an apparatus.

The technology of real-time reliable multicast over a best-effort service network has become more popular recently. In this paper, a new technique is introduced that integrates word interleaving, forward error correction (FEC) and automatic repeat request (ARQ) to mitigate the error and loss effects encountered in wired and wireless Internet applications. For multicast video sessions spanning tens of routers, the integration of interleaving, FEC and ARQ as well as fine tuning of the various

parameters of each mechanism will play a major role in the move towards successful deployment. In this work, the basic algorithms for integrated error control are first derived. Then the corresponding real-time implementations for an experimental test bed to achieve the practical performance of these techniques are completed. The performance of the scheme is analyzed, and laboratory results are compared with analytical results.

Chen at NOK00075559, Abstract. *See also id.* NOK00075559, col. 1 ¶ 2-col. 2 ¶ 1:

Although work has been done on employing forward error correction (FEC) and automatic repeat request (ARQ) in multicast [8]–[9], few actual buildups and experiments which integrate all the techniques, such as interleaving, FEC and ARQ, have been reported. This paper concentrates on an actual experiment involving interleaved RS/ARQ error correction. The results show that RS/ARQ combined with interleaving can provide strong multicast error control and excellent QoS guarantees at high user data rates.

510. The experimental test bed shown in Figure 6 of Chen is evidence that Chen discloses an apparatus.

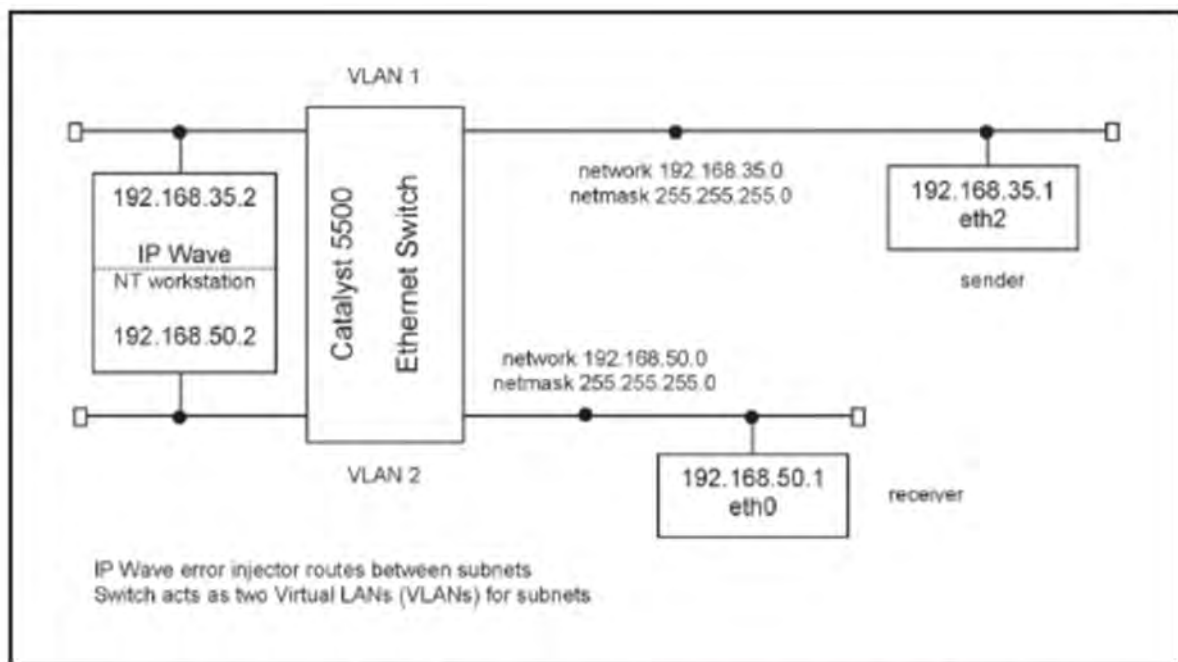


Figure 6: *Block diagram of the experimental test bed used (IP Wave used for error and loss injection).*

Chen at NOK00075563, Fig. 6.

511. Chen describes the combination of interleaving, Reed-Solomon codes, and automatic repeat request (“ARQ”) to protect from and correct errors in both multicast and unicast systems.

In this section, we present the theoretical analysis of our interleaved FEC/ARQ scheme in a step-by-step way. First, we analyze the basic characteristics of our hybrid FEC/ARQ technique in unicast. Then we evaluate its performance in a multicast environment. ...

Chen at NOK00075562, col. 2 ¶ 2 (partial).

512. Chen performed experiments using the apparatus on the testbed. Chen at NOK00075561-562, page 3, col. 2 ¶ 3-page 4, col. 2 ¶ 1. In those experiments the retransmission and interleaving functions were performed by general purpose CPUs that would, of course, use shared memory and therefore would have allocated shared memory between an interleaving function and a retransmission function. *Id.* Chen explicitly discusses how interleaving, FEC, and ARQ are performed in his experiments and expected to be performed in real systems by the server and client CPUs, as shown in the excerpt below:

mental results above show that the performance of our algorithms is sensitive mainly to packet loss and errors. Bandwidth expansion due to FEC/ARQ utilization may bring occasional congestion, packet loss and errors. However, test results indicate that the performance of our algorithms is not compromised as long as the CPU speeds of the server and clients are capable of handling the real-time interleaved FEC/ARQ coding/decoding algorithms therein.

Chen at NOK00075562, col. 2 ¶ 1 (partial).

513. In addition, ADSL2 also discloses an apparatus.

This Recommendation describes Asymmetric Digital Subscriber Line (ADSL) transceivers on a metallic twisted pair that allows high-speed data transmission between the network operator end (ATU-C) and the customer end (ATU-R).

ADSL2 at i, Summary. *See also id.* 9-10, § 5.1 “ATU functional model.”

Figure 5-1 shows the functional blocks and interfaces of an ATU-C and ATU-R that are referenced in this Recommendation. It illustrates the most basic functionality of the ATU-R and the ATU-C. Each ATU contains both an application invariant

section and an application-specific section. The application invariant section consists of the PMS-TC and PMD layers and are defined in clauses 7 and 8, while the application-specific aspects that are confined to the TPS-TC layer and device interfaces, are defined in Annex K. Management functions, which are typically controlled by the operator's management system (EMS or NMS), are not shown in Figure 5-1. Figure 5-3 provides a high level view that includes the management interface.

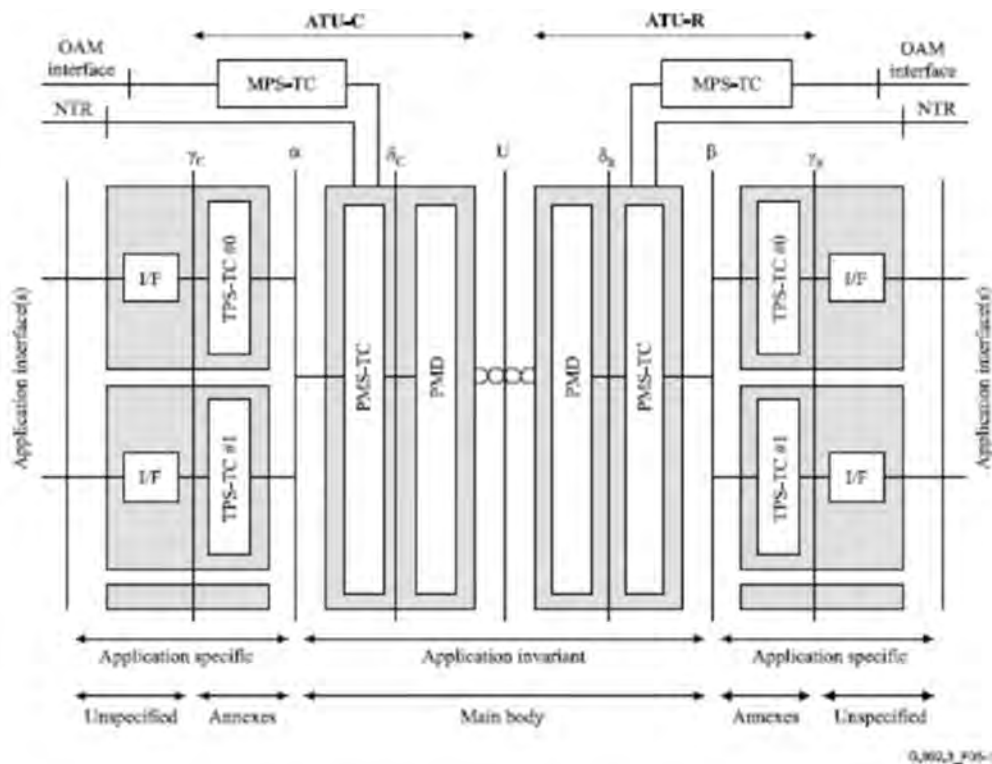


Figure 5-1/G.992.3 – ATU functional model

ADSL2 at 9, Figure 5-1.

514. Because both Chen and ADSL2 disclose an apparatus, this limitation is rendered obvious by the combination of Chen and ADSL2.

iii Yap in view of VDSL1

515. Yap discloses an apparatus. For example, Yap describes an OFDM transceiver: “FIG. 1 is a diagram of transmitter 100 for a multiple-input multiple-output communication system that utilizes orthogonal frequency division multiplexing (namely, a MIMO-OFDM system).” Yap at NOK00084270, 3:65-4:2.

[1.a] a multicarrier transceiver including a processor and memory operable to:

i VDSL1

516. VDSL1 discloses “a multicarrier transceiver including a processor and memory.” Specifically, VDSL1 uses a transceiver with multi-carrier modulation (which is a multicarrier transceiver) that includes a processor and memory.

517. VDSL1 specifies the number of subcarriers used in multi-carrier modulation and uses the subcarriers to transmit on four frequency bands.

9.2.1 Multi-carrier modulation

The modulation shall use a maximum number of subcarriers equal to $N_{SC} = 2^{n+8}$ where n can take the values 0, 1, 2, 3, 4. Disjoint subsets of the N_{SC} sub-carriers shall be defined for use in the downstream and upstream directions. These subsets are determined by the frequency plan (see 6.1). The exact subsets of sub-carriers used to modulate data in each direction shall be determined during initialization and shall be based on management system settings and the signal-to-noise ratios (SNRs) of the sub-channels. In many cases the number of sub-carriers used in a direction will be less than the maximum number allowed by the partitioning.

VDSL1 at NOK00078792, § 9.2.1 “Multi-carrier modulation.” *See also:*

G.993.1 transceivers shall use Frequency Division Duplexing (FDD) in separating upstream and downstream transmission.

G.993.1 systems use a four-band plan that starts at 138 kHz and extends up to 12 MHz. The four frequency bands denoted as DS1, US1, DS2, and US2, for the first downstream band, the first upstream band, the second downstream band, and the second upstream band, respectively, as shown in Figure 6-1, shall be allocated according to the band separating frequencies f_1 , f_2 , f_3 , f_4 , and f_5 .

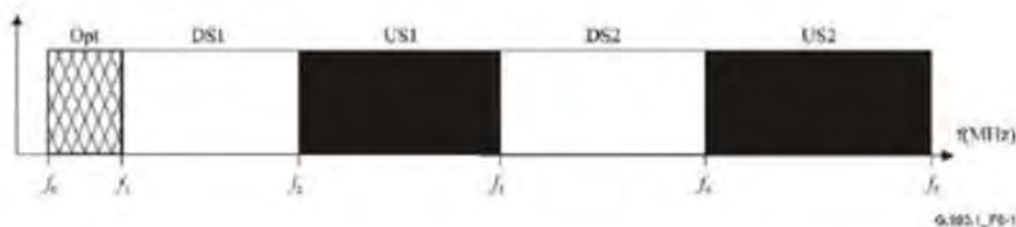


Figure 6-1/G.993.1 – G.993.1 band allocation

VDSL1 at NOK00078777, § 6.1 “Duplexing method.”

ii Chen in view of ADSL2

518. Chen in view of ADSL2 renders obvious “a multicarrier transceiver including a processor and memory.” Specifically, Chen discloses a transceiver that includes a processor and memory. ADSL2 also separately discloses a multicarrier transceiver that includes a processor and memory. Therefore, Chen in view of ADSL2 renders this element obvious.

519. Chen does not explicitly state that it works with a multicarrier transceiver; rather, the techniques described in Chen work with any physical layer, wired or wireless, including a DSL system like ADSL2. Such systems have a processor and memory. *See, e.g.*, Chen, Abstract, page 159, col. 1 ¶ 1-col. 2 ¶ 1, Fig. 1.

520. ADSL2 describes transmitting data using a multicarrier transceiver, which has a processor and memory.

8.8 Modulation

The modulator shall modulate a constellation encoder output data frame or sync frame (containing $NSC - 1$ complex values Z_i , $i = 1$ to $NSC - 1$) into a DMT symbol. The data frame can be taken from the data symbol constellation encoder (68 per superframe) as defined in 8.6. The sync frame can be taken from the synchronization symbol constellation encoder (1 per superframe) as defined in 8.7. For (short) initialization and diagnostics mode signals, the frame is defined in 8.13, 8.14 and 8.15.

8.8.1 Subcarriers

A DMT symbol consists of a set of subcarriers, with index $i = 0$ to NSC . The DMT subcarriers spacing Δf , shall be 4.3125 kHz, with a tolerance of ± 50 ppm. The subcarrier frequencies shall be $f_i = i \times \Delta f$, $i = 0$ to NSC .

8.8.1 Subcarriers

A DMT symbol consists of a set of subcarriers, with index $i = 0$ to NSC . The DMT subcarriers spacing Δf , shall be 4.3125 kHz, with a tolerance of ± 50 ppm. The subcarrier frequencies shall be $f_i = i \times \Delta f$, $i = 0$ to NSC .

ADSL2 at 89-90, §§ 8.8-8.8.1 “Modulation” and “Subcarriers.”

521. Chen in view of ADSL2 therefore renders obvious “a multicarrier transceiver” (disclosed by ADSL2) “including a processor and memory” (disclosed by both Chen and ADSL2).

iii Yap in view of VDSL1

522. Yap discloses “a multicarrier transceiver including a processor and memory.” In particular, Yap discloses a MIMO-OFDM system. Yap at NOK00084262, abstract, 3:65-4:2. A POSITA would understand that OFDM is a multicarrier communication technique. Further, this system includes a processor and memory. Yap at 4:8-20, 4:32-53, Figs. 1 and 2.

[1.b] transmit a packet using a forward error correction encoder and an interleaver

i VDSL1

523. VDSL1 discloses “transmit a packet using a forward error correction encoder and an interleaver.” In particular, VDSL1 describes how the transmitted data is interleaved; the receiving transceiver deinterleaves received data using the same process in reverse.

Interleaving shall be used to protect the data against bursts of errors by spreading the errors over a number of Reed-Solomon codewords. The interleave depth shall be programmable with a maximum interleave depth of 64 codewords when the number of octets per codeword (N) equals 255. For smaller values of N the interleave depth can grow nearly proportionately.

It shall be possible to adjust the interleave depth via the management system to meet latency requirements. The latency of the slow path is a function of the data rate and burst error correction capability. For data rates greater than or equal to 13 Mbit/s the latency between the a and 13 interfaces shall not exceed 10 Ms when the interleaver depth is set to the maximum. At lower data rates there is a trade-off between higher latency and decreased burst error correction ability. At any data rate, the minimum latency occurs when the interleaver is turned off.

When the interleaver is on the codewords shall be interleaved before transmission to increase the immunity of RS codewords to bursts of errors. The convolutional interleaver is defined by two parameters: the interleaver block length, I, and the interleaving depth D. The block length I shall divide the RS codeword length N (i.e., N shall be an integer multiple of I). The convolutional interleaver uses a memory in which a block of I octets is written while an (interleaved) block of I octets is read. Details of the implementation are given in 8.4.2.

The same size interleaving memory (see Table 8-1) is needed for interleaving at the transmitter and de-interleaving at the receiver.

The convolutional interleaving introduces an absolute read-to-write delay, A_j , that increments linearly with the octet index within a block of I octets :

$$\Delta_j \equiv (D - 1) \times j$$

where $j = 0, 1, 2, \dots, I - 1$.

VDSL1 at NOK00078785, § 8.4.1 “Interleaving – General”. *See also id.* NOK00078979, § I.5.4.2:

De-interleaving is performed at the receive side by writing the octets of the incoming codeword into a bank of I virtual shift registers numbered $j = 0, 1, \dots, (I - 1)$. The length of virtual shift register j in the deinterleaving memory is: $M \times (I - 1 - j)$.

524. VDSL also explains that Reed-Solomon (RS) coding shall be used for forward error correction (FEC) to provide protection against random and burst errors. *See* VDSL, §§ 8.3, I.1.2.7.

ii Chen in view of ADSL2

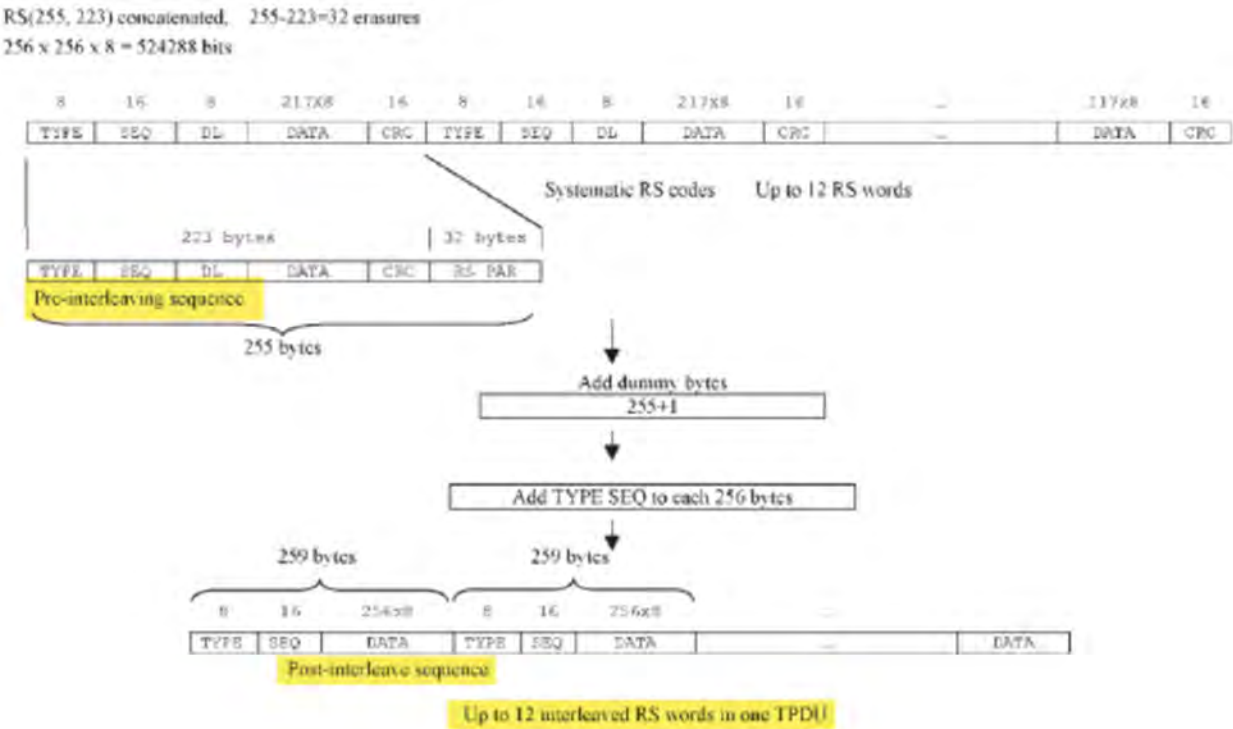
525. Chen discloses “transmit a packet using a forward error correction encoder and an interleaver.” For example, Chen describes using a combination of FEC including interleaving and automatic repeat request (“ARQ”) to detect and correct errors in packets. Chen describes using interleaving to protect transmitted packets from errors and, subsequently, deinterleaving the received packets.

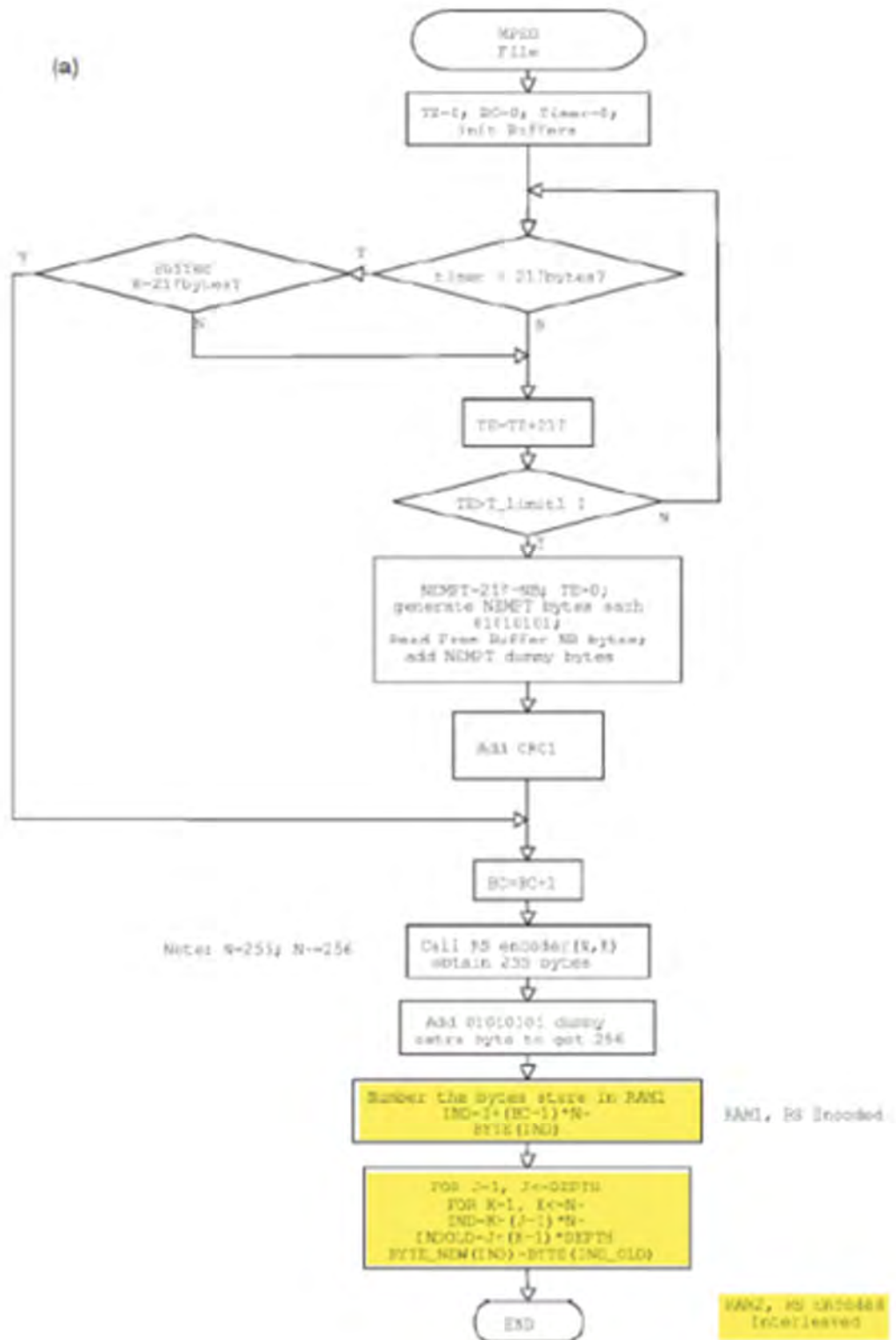
Interleaving is next applied to a record of 256 such encoded RS words to yield a corresponding interleaved record. The first interleaved word of such a record consists of the first bytes of each of the 256 preinterleaving words. The 256-th interleaved word is the concatenation of the 256-th bytes of all the pre-interleaving words. It is also possible to have a smaller interleaving depth such as 128, in which case the first interleaved word will encompass the first and second bytes of every pre-interleaving word. Then the first 128 bytes of the 128-th interleaved word are bytes numbered 255 in the original pre-interleaving words, and the second 128 bytes of the 128-th interleaved word are bytes numbered 256 in the original pre-interleaving words.

If the interleaving depth is 256, it is easy to understand that up to 32 consecutive interleaved words can be lost without any ARQ retransmission request from the receiver. Because of interleaving, if any 32 words are lost, only 32 of the 255 bytes of each corresponding preinterleaving word will be lost. Because the RS decoder

can conceal up to $255 - 223 = 32$ lost bytes, such loss is not detrimental and will not result in retransmission requests. For error concealment for multicast traffic over the Internet, minimizing the need for retransmissions is an essential asset, which subsequently leads to the minimizing of no acknowledgement (NAK) and repair-packet implosions [2]. Returning to Fig. 1, we see that the server routine groups up to 12 interleaved words into one UDP data unit (TPDU). A 2-byte interleaved sequence number field is added to each interleaved word, preceded by a 1-byte field denoting an interleaved word, i.e., 11111111. Although each original RS-encoded word carries a type field and a sequence field, these fields are scrambled later by the interleaving, and the need arises for new type and sequence fields. Occasionally or at the request of the client, retransmitted words are also included in some TPDU's. If an interleaving depth of 256 is used, each TPDU contains eight interleaved words; in this case four consecutive TPDU's can be lost without any retransmissions.

Chen at NOK00075560., col. 1 ¶¶ 2-3. *See also id.* Figs. 1-2a and 5 (Figs. 1 and 5 excerpted for readability):





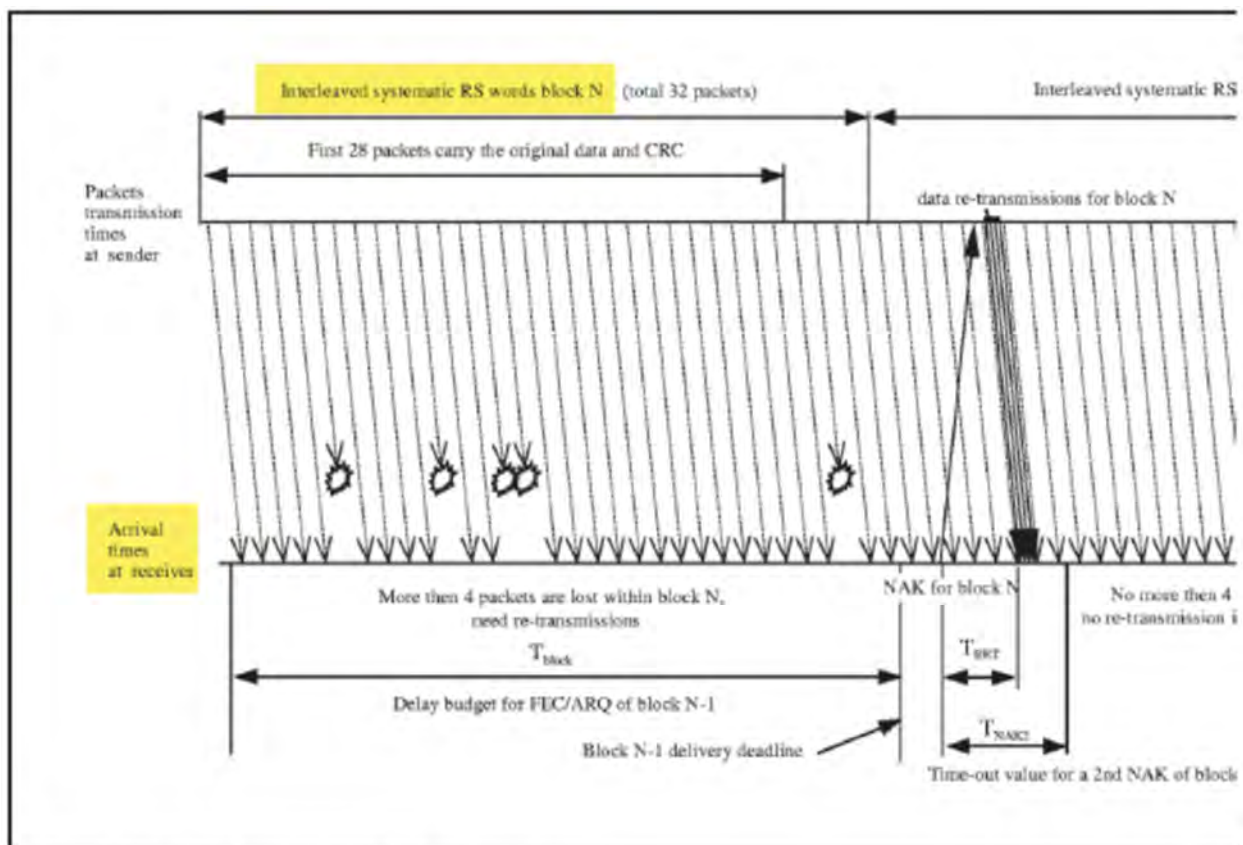


Figure 5: The hybrid FEC/ARQ system timing.

526. Once the interleaved words are received, they are deinterleaved.

... If the type field indicates an interleaved word, this interleaved word is sent to the de-interleaving buffer. A timer is reset upon the arrival of the first byte of a certain deinterleaved RS word (this may not be the first byte of the word).

Routinely, the de-interleaved RS words are checked one after another. ...

Chen at NOK00075560, col. 2 ¶¶ 2-3 (excerpted). See also *id.* NOK00075562, Fig.

4(a).



527. ADSL2 also describes transmitting packets using interleaving; on the receive side, the received packets are deinterleaved. For example, Figure 7-7 in ADSL2 shows data being interleaved at the PMS-TC layer prior to transmission where data is sent over two frame bearers and a single latency path.

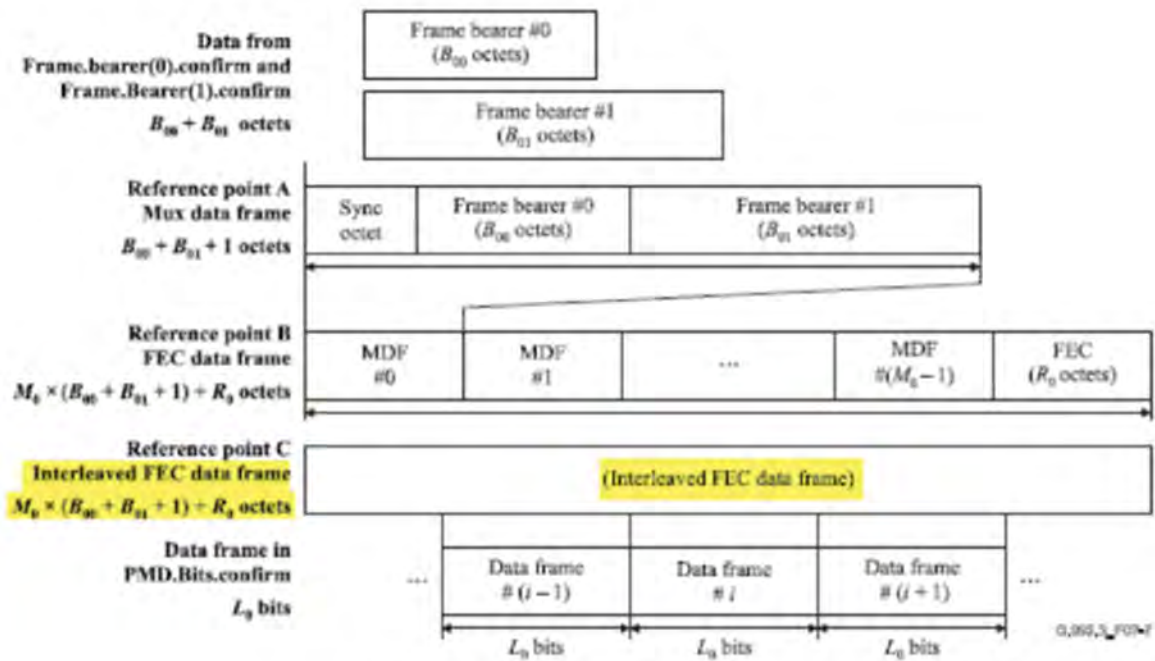


Figure 7-7/G.992.3 – Illustration of frame structure with single latency dual bearers and $T_p = 1$

ADLS2 at 29, Figure 7-7. Figure 7-8 shows the same information, but for a system using two frame bearers and two latency paths.

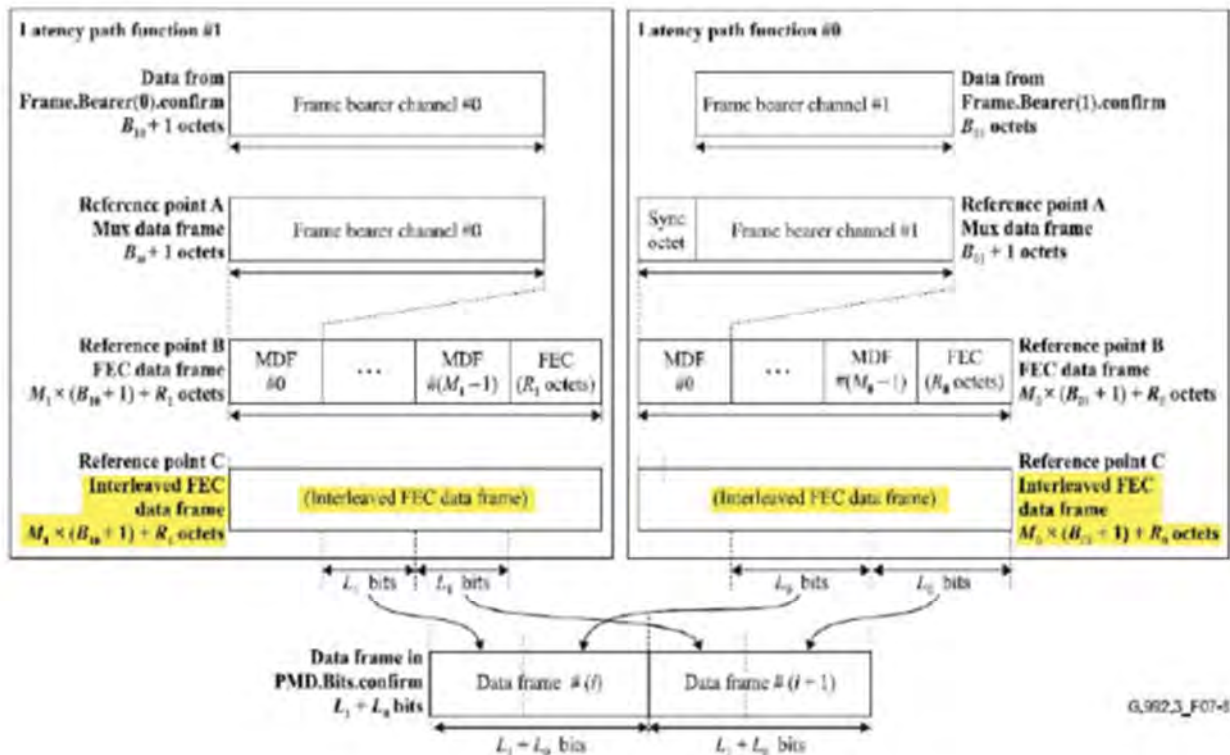


Figure 7-8/G.992.3 – Illustration of frame structure with dual latency and dual bearers

ADLS2 at 30, Figure 7-8.

528. The interleaving used by ADSL2 is a convolutional interleaver.

To spread the Reed-Solomon codeword and, therefore, reduce the probability of failure of the FEC in the presence of impulse noise, the FEC Output Data Frames shall be convolutionally interleaved. The interleaver creates the Interleaved FEC Output Data Frames at Reference point C, at the output of the latency path function. This procedure is followed by the frame multiplexing procedure.

Convolutional interleaving is defined by the rule (using the currently defined values of the framing control parameters D_p and the derived parameter $N_{FEC,p}$):

Each of the $N_{FEC,p}$ octets $B_0, B_1, \dots, B_{N_{FEC,p}-1}$ in an FEC Output Data Frame is delayed by an amount that varies linearly with the octet index. More precisely, octet B_i (with index i) is delayed by $(D_p - 1) \times i$ octets, where D_p is the interleaver depth.

An example for $N_{FEC,p} = 5, D_p = 2$ is shown in Table 7-13, where B_i^j denotes the i -th octet of the j -th FEC Output Data Frame.

Table 7-13/G.992.3 – Convolutional interleaving example for $N_{FEC,p} = 5$, $D_p = 2$

Interleaver input	B_0^j	B_1^j	B_2^j	B_3^j	B_4^j	B_0^{j+1}	B_1^{j+1}	B_2^{j+1}	B_3^{j+1}	B_4^{j+1}
Interleaver output	B_0^j	B_3^{j-1}	B_1^j	B_4^{j-1}	B_2^j	B_0^{j+1}	B_3^j	B_1^{j+1}	B_4^j	B_2^{j+1}

With the above-defined rule, the output octets from the interleaver always occupy distinct time slots when $N_{FEC,p}$ is odd and D_p is a power of 2. When $N_{FEC,p}$ is even, a dummy octet shall be added at the beginning of the codeword at the input to the interleaver. The resultant odd-length codeword is then convolutionally interleaved, and the dummy octet shall then be removed from the output of the interleaver.

The interleaving procedure of the latency path function #p shall interleave a single FEC Output Data Frame, or $M_p \times K_p + R_p$ octets. This procedure is followed by the Frame Multiplexing procedure.

ADSL2 at 38, § 7.7.1.5 “Interleaver.” Deinterleaving is performed on the received packet to obtain the transmitted data (which may have some errors due to the transmission channel).

529. ADSL2 also describes the forward error correction function:

7.7.1.4 Forward error correction function

The FEC procedure inserts Reed-Solomon FEC redundancy octets to provide coding gain as an outer coding function to the PMD function. The FEC procedure of latency path function #p shall calculate R_p octets from $M_p \times K_p$ input octets. The octets are appended to the end of the input octets in the structure of FEC Output Data Frame at Reference Point B.

When $R_p = 0$, no redundancy octets are appended and the values in the FEC Output Data Frame are identical to the input values. For all other values of R_p , the following encoding procedure shall be used to create the R_p octets:

The FEC procedure shall take in M_p scrambled Mux Data Frames comprising message octets, $m_0, m_1, \dots, m_{M_p \times K_p - 2}, m_{M_p \times K_p - 1}$. The procedure shall produce R_p redundancy octets $c_0, c_1, \dots, c_{R_p - 2}, c_{R_p - 1}$. These two taken together comprise the FEC codeword of size $M_p \times K_p + R_p$ octets. The R_p redundancy octets shall be appended to the message octets to form the FEC Output Data Frame at Reference Point B.

At the end of the initialization sequence, the FEC Function always starts with the first of M_p Mux Data Frames.

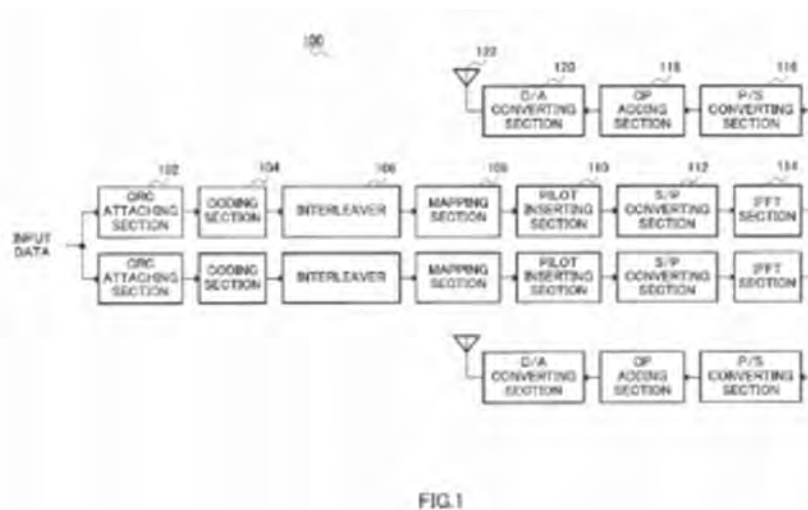
ADSL2, § 7.7.1.5 “Forward Error Correction Function.”

530. Chen in view of ADSL2 therefore renders obvious “transmit a packet using a forward error correction encoder and an interleaver.”

531. Yap describes and depicts a multicarrier transceiver that transmits packets using interleaving.

At transmitter 100, data processing is performed for each individual antenna chain. Different streams of independent data are transmitted from the individual transmission antennas. The input data is first attached the cyclic redundancy check (CRC) code at CRC attaching section 102. Then, channel coding such as convolutional coding and turbo coding is carried out at coding section 104. The coded data will then be interleaved by interleaver 106 to reduce burst errors in the data. M-ary modulation constellation symbol mapping is executed on the interleaved data at mapping section 108.

Yap at NOK00084270, 4:8-19. The various functions of the transmitter, including the interleaver, are also shown in Figure 1:



532. Yap further describes and depicts a multicarrier transceiver that receives packets using deinterleaving. “After carrying out demapping (demapping section 216), deinterleaving (deinterleaver 218) and decoding (decoding section 220), cyclic redundancy check (CRC processing section 222) is then performed on each packet to validate the data.” Yap at NOK00084270, 4:45-47. The various functions of the receiver, including the deinterleaver, are also shown in Figure 2:

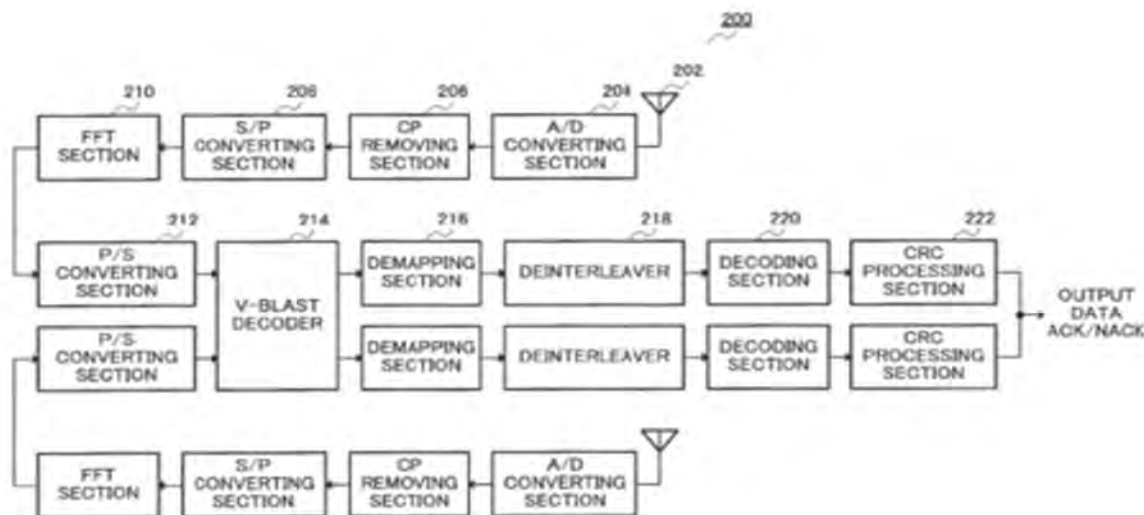


FIG.2

[1.c] wherein the packet comprises a header field and a plurality of PTM-TC codewords, a plurality of ATM cells or a plurality of Reed-Solomon codewords,

i VDSL1

533. VDSL1 discloses “wherein the packet comprises a header field and a plurality of PTM-TC codewords, a plurality of ATM cells or a plurality of Reed-Solomon codewords.”

534. The VDSL1 specification covers the functional requirements for the transport of ATM cells and for PTM-TC codewords as described in Section 5.1 General reference models NOK00078775, shown below:

It is envisaged that VDSL will find applications in the transport of various protocols. For each application different functional requirements must be developed for the Transport Protocol Specific-Transmission Convergence Layer (TPS-TC). This specification covers the functional requirements for the transport of ATM and PTM. However, the G.993.1 core transceiver shall be capable of supporting future additional applications.

535. VDSL1 discloses how packets may be comprised of ATM cells in Appendix G entitled ATM-TC. VDSL1, pages 138-142, NOK00078908-12. For example, Section G.1 Scope

(VDSL1, page 138) explains that Annex G describes the ATM-based service transmission over a VDSL link.

This annex specifies a VDSL ATM Transport Protocol Specific Transmission Convergence sublayer (ATM-TC), which describes the ATM-based service transmission over a VDSL link. This annex defines a minimum set of requirements to deliver an ATM service from the ONU to the remote customer premises. It is based on ITU-T Rec. I.432.1. The ATM-TC specification is applicable at both the VTU-O and the VTU-R sides.

536. VDSL1 discloses how packets may be comprised of Packet Transfer Mode (PTM) Transmission Convergence (TC) codewords in Annex H entitled PTM-TC at pages 143-149 (NOK00078913-19). Annex H describes how the PTM entity obtains data packets to be transported over VDSL from the application interface. The PTM-TC receives the packet:

The functional model of packetized data transport is presented in Figure H.1. In the transmit direction, the *PTM entity* obtains data packets to be transported over VDSL from the application interface. The PTM entity processes each packet and applies it (depending on the latency requirements) to the interface of the either Fast or Slow VDSL path intended for packetized data transport. The corresponding TPS-TC (PTM-TC) receives the packet from the γ interface, encapsulates it into a special frame (PTM-TC frame) and maps into the PMS-TC frame (transmission frame) for transmission over the VDSL link.

VDSL1, page 143.

537. Figure 5-2 of Section 5.2 Functional reference model illustrates how the Transport Protocol Specific-Transmission Convergence Layer (TPS-TC) is connected to the Physical Media Specific-Transmission Convergence layer (PMS-TC) at the α/β interfaces.

5.2 Functional reference model

One of the TPS-TC in Figure 5-2 may be assigned for management purposes and is called Overhead Channel TC (OC-TC).

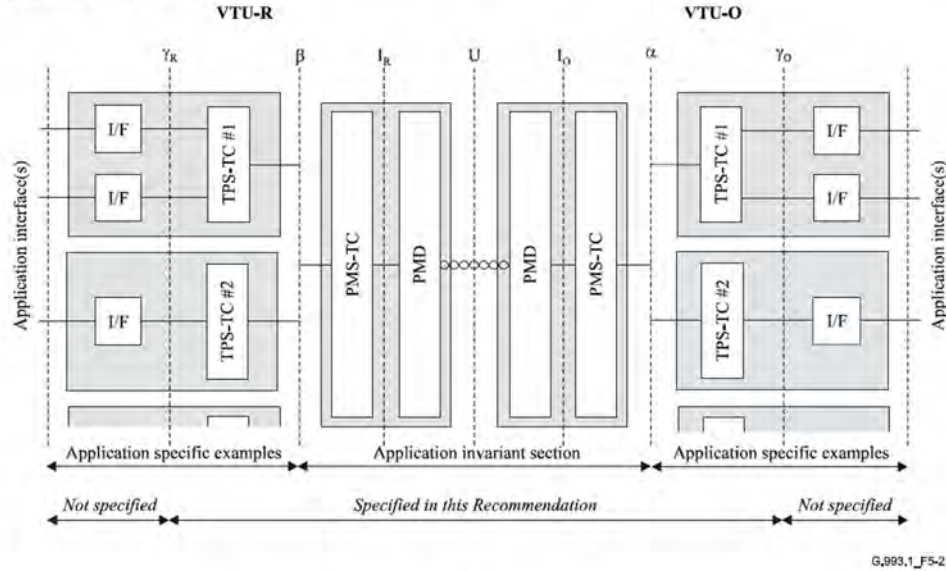


Figure 5-2/G.993.1 – VTU-x functional reference model

538. Within the PMS-TC, Reed-Solomon forward error correction is employed so that VDSL1 packets will be comprised of a plurality of Reed-Solomon codewords. This is described in Section 8.3 Forward error correction, on pages 14-15:

8.3 Forward error correction

A standard byte-oriented Reed-Solomon code shall be used to provide protection against random and burst errors.

A Reed-Solomon code word contains $N = K + R$ bytes, comprised of R redundant check bytes $c_0, c_1, \dots, c_{R-2}, c_{R-1}$ appended to K message bytes $m_0, m_1, \dots, m_{K-2}, m_{K-1}$. The check bytes shall be computed from the message bytes using the equation

$$C(D) = M(D)D^R \bmod G(D)$$

where:

$M(D) = m_0D^{K-1} \oplus m_1D^{K-2} \oplus \dots \oplus m_{K-2}D \oplus m_{K-1}$ is the message polynomial

$C(D) = c_0D^{R-1} \oplus c_1D^{R-2} \oplus \dots \oplus c_{R-2}D \oplus c_{R-1}$ is the check polynomial

$G(D) = \prod (D \oplus \alpha^i)$ is the generator polynomial of the Reed-Solomon code, where the index of the product runs from $i = 0$ to $R - 1$.

This means that $C(D)$ is the remainder obtained from dividing $M(D)D^R$ by $G(D)$. The arithmetic shall be performed in the Galois Field GF(256), where α is a primitive element that satisfies the

primitive binary polynomial $x^8 \oplus x^4 \oplus x^3 \oplus x^2 \oplus 1$. A data byte $(d_7, d_6, \dots, d_1, d_0)$ is identified with the Galois Field element $d_7\alpha^7 \oplus d_6\alpha^6 \oplus \dots \oplus d_1\alpha \oplus d_0$.

Both K and R shall be programmable parameters. Redundancy values of $R = 0, 2, 4, 6, 8 \dots 16$ shall be supported. The following codeword parameters specified as (N, K) shall be supported: (144,128) and (240,224). Other values for N and K are optional. However, N shall be less than or equal to 255.

539. VDSL1 discloses packets that comprises a header field and a plurality of PTM-TC codewords, a plurality of ATM cells or a plurality of Reed-Solomon codewords. For example, VDSL1 places ATM cells, PTM-TC codewords, and Reed-Solomon codewords in packets, also called frames in VDSL1. This is described in Section 8.5 Framing, on pages 17-18, shown below. In discussing the payload adaptation, the terms “packet” and “frame” are used interchangeably.

8.5 Framing

8.5.1 Frame description

A *frame* is a set of bytes carried by one DMT symbol. The frame frequency depends on the total length of the cyclic extension (see 9.2.2). A frame shall be composed of two sources: the "fast" buffer and the "interleaved" (or "slow") buffer. The index i refers to parameters related to the fast or interleaved buffers ($i \in \{F, I\}$). The inclusion of the fast buffer shall be optional. When the fast buffer is not included, the interleaved buffer shall have the capability to carry non-interleaved data by setting the interleaver depth to zero.

Both fast and interleaved buffer shall contain an integer number of RS-encoded bytes. Neither the fast nor the interleaved buffer is required to carry an integer number of RS codewords. To reduce the end-to-end delay, it is recommended that the fast buffer (or the interleaved buffer when the interleaver depth is zero) carries at least one RS codeword. The framing parameters shall be exchanged between the VTU-O and VTU-R during initialization.

The framing rules described in this clause are represented in Figure 8-3.

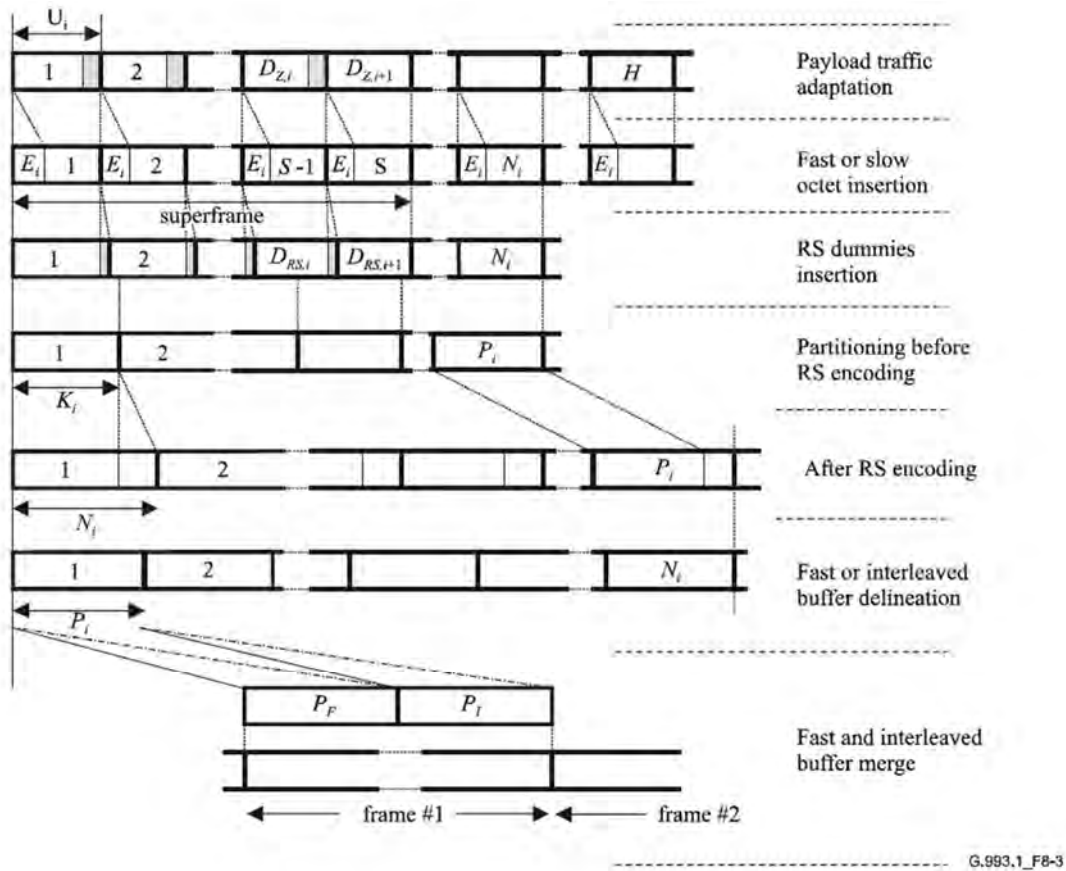


Figure 8-3/G.993.1 – Framing description

8.5.2 Payload adaptation

The α/β interface provides bytes at a rate multiple of 64 kbit/s. In order to map an integer number of bytes into a frame, the TPS-TC byte flow shall be stuffed with the appropriate number of dummies.

For an $n_i \times 64$ kbit/s rate, we have on average $n_i \times 8000/f_s$ bytes per frame, with f_s the symbol frequency. This number will not be integer for a general value of f_s . Since the cyclic extension $L_{CP} + L_{CS} - \beta$ is a multiple of 2^{n+1} however, (see 9.2.2), we always have an integer number of bytes every $H = 138$ frames. If we define k as:

$$k = \frac{8 \text{ kbytes} \times H}{f_s}$$

we can transport $n_i \times k$ TPS-TC payload bytes in H frames. In order to transport an integer number of bytes per frame, we have to insert an appropriate number of dummy bytes. Every frame will contain a total of U_i bytes (TPS-TC bytes + dummy bytes), with:

$$U_i = \left\lceil \frac{n_i \times k}{H} \right\rceil$$

The number of dummy bytes $D_{Z,i}$ to be inserted every H packets shall therefore be:

$$D_{Z,i} = \left\lceil \frac{n_i \times k}{H} \right\rceil \times H - (n_i \times k)$$

These dummy bytes shall be inserted in the last position of the first $D_{Z,i}$ packets of U_i bytes in a sequence of H packets. The value of the $D_{Z,i}$ dummies shall be 0x3A.

540. For example, the overhead bytes, also called fast or slow bytes, comprise the header field for the VDSL1 packets. The overhead bytes are described in Section 8.5.5 “Contents of fast and slow bytes.” on pages 19-21. An excerpt of Section 8.5.5 is shown below.

8.5.5 Contents of fast and slow bytes

Each of the packets in a superframe shall transport E_i overhead bytes, called fast or slow bytes, depending on the channel. The content of these bytes is summarized in Table 8-3. If the fast buffer is empty, the F-EOC bytes shall be transported in the S-EOC bytes. Otherwise, the S-EOC bytes shall be replaced with payload bytes.

There shall be V VOC bytes per packet. They shall always be transported in the slow channel. A setting of $V = 1$ shall be supported, other values for V should be allowed as optional. The value of V shall be exchanged during initialization (see 12.4.6.2.1.1).

If the fast path is active, the NTR byte in the slow channel shall be replaced with a dummy byte. Similarly for the IB bytes.

The fast and slow dummy bytes shall have the value 0xFF.

Table 8-3/G.993.1 – Contents of fast and slow bytes

Packet	Fast bytes		Slow bytes		
	First byte	Other bytes (if any)	First byte	2nd up to (V + 1)st byte	Other bytes (if any)
1	F-CRC	F-EOC	S-CRC	VOC	S-EOC/payload
2	Synch byte	F-EOC	Synch byte	VOC	S-EOC/payload
3-5	IB	F-EOC	IB/dummy	VOC	S-EOC/payload
6	NTR	F-EOC	NTR/dummy	VOC	S-EOC/payload
7-10	Dummy	F-EOC	Dummy	VOC	S-EOC/payload

541. As another example, each 53-byte ATM cell includes a header byte that is used for a header error check (HEC). This is discussed in Section g.4.1.1 Data flow on pages 140-141. For example, in the excerpt below:

The ATM cell format is identical in both transmit and receive directions: 52 out of the 53 octets carry ATM layer data (user data). Octet number 5 is undefined (intended for HEC insertion in the TC sublayer).

542. As another example, the PTM-TC frame format uses an HDLC-type mechanism which includes a header field that comprises an opening flag sequence, an address field, a control field, frame-check-sequence octets, and a closing flag sequence. This is described in section H.4.1.1 “Frame structure” (pages 146-147), an excerpt of which is shown below:

H.4.1 Packet encapsulation

For packet encapsulation, an HDLC-type mechanism shall be used with detailed characteristics as specified in the following subclauses.

H.4.1.1 Frame structure

The PTM-TC frame format shall be as shown in Figure H.2. The opening and closing Flag Sequences shall be set to $7E_{16}$. They identify the start and the end of the frame. Only one Flag Sequence is required between two consecutive frames.

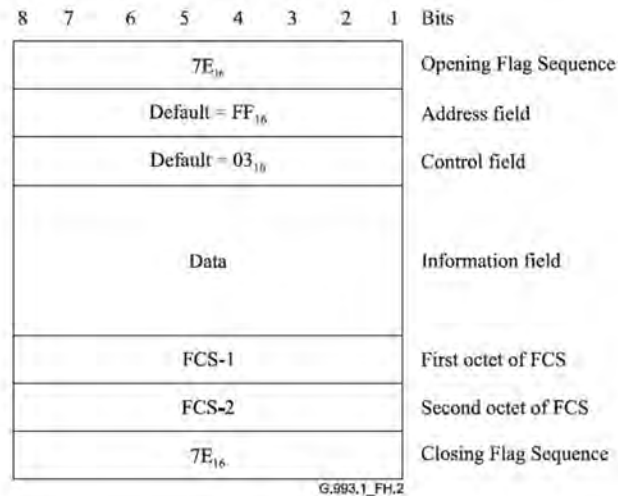


Figure H.2/G.993.1 – PTM-TC frame format

The Address and Control octets are intended for auxiliary information. They shall be set to their default values of hexadecimal FF₁₆ and 03₁₆ respectively if not used.

NOTE 1 – The address and Control fields may be used for different auxiliary OAM functions.

The Information field shall be filled with the transported data packet. Prior to encapsulation the octets of the data packet shall be numbered sequentially. Octets shall be transmitted in ascending numerical order.

The Frame Check Sequence (FCS) octets are used for packet level error monitoring, and shall be set as described in H.4.3.

543. VDSL1 therefore discloses “wherein the packet comprises a header field and a plurality of PTM-TC codewords, a plurality of ATM cells or a plurality of Reed-Solomon codewords.”

ii Chen in view of ADSL2

544. Chen and ADSL2 each disclose “wherein the packet comprises a header field and a plurality of PTM-TC codewords, a plurality of ATM cells or a plurality of Reed-Solomon codewords.” Specifically, each of Chen and ADSL2 separately disclose that the packet which is transmitted using interleaving and received using deinterleaving is comprised of Reed-Solomon codewords.

545. Chen builds the packets that are sent and interleaved from data encoded in Reed-Solomon codewords; the packets are deinterleaved at the receiver.

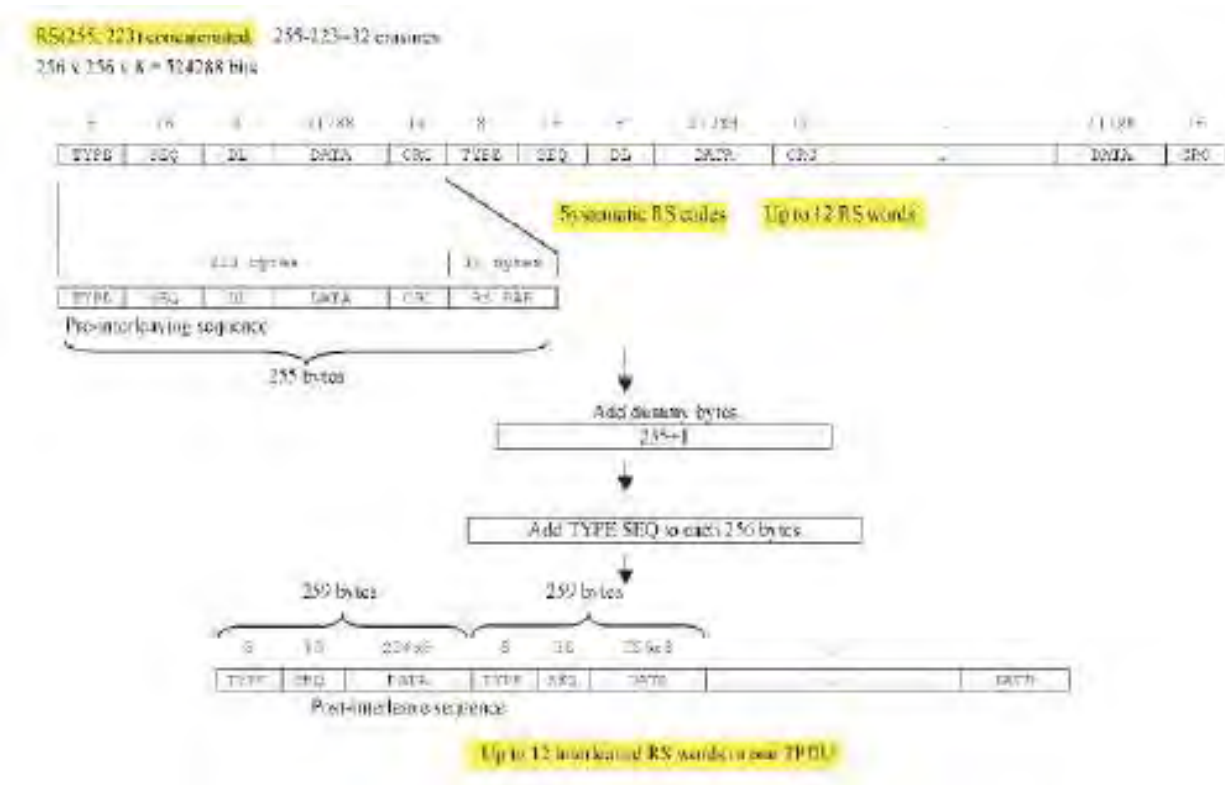
Nowadays, multicast is becoming more and more important to realtime multimedia communication applications, such as interactive video conferencing, video-on-demand systems, and so on. These applications usually have stringent quality of service (QoS) requirements in terms of bandwidth, end-to-end delay, delay jitter and packet loss rate. As one of the most challenging aspects of multicast, the technology of controlling packet loss has attracted more and more attention [1]–[6]. Congestion, tunnelling, and lack of high-speed backbones are among the reasons for occasional data loss. On the wireless side, bursty channel errors combined with packet loss make stronger error and loss control more crucial. Numerous techniques have been suggested for error and loss control over the Internet. In the past, mere packet replication was usually suggested to combat the best-effort delivery and subsequent transportation-layer protocol data unit (TPDU) loss of the Internet, but now this approach has given way to powerful erasure codes. These erasure codes, such as Reed-Solomon (RS) codes, have strong inherent erasure-correction capability [7].

Although work has been done on employing forward error correction (FEC) and automatic repeat request (ARQ) in multicast [8]–[9], few actual buildups and experiments which integrate all the techniques, such as interleaving, FEC and ARQ, have been reported. This paper concentrates on an actual experiment involving interleaved RS/ARQ error correction. The results show that RS/ARQ combined with interleaving can provide strong multicast error control and excellent QoS guarantees at high user data rates.

Chen at NOK00075559, col. 1 ¶¶ 1-2-col. 2 ¶ 1. *See also id.* NOK00075560, col. 1 ¶ 2:

Interleaving is next applied to a record of 256 such encoded RS words to yield a corresponding interleaved record. The first interleaved word of such a record consists of the first bytes of each of the 256 preinterleaving words. The 256-th interleaved word is the concatenation of the 256-th bytes of all the pre-interleaving words. It is also possible to have a smaller interleaving depth such as 128, in which case the first interleaved word will encompass the first and second bytes of every pre-interleaving word. Then the first 128 bytes of the 128-th interleaved word are bytes numbered 255 in the original pre-interleaving words, and the second 128 bytes of the 128-th interleaved word are bytes numbered 256 in the original pre-interleaving words.

See also id. Fig. 1 (cropped to show that Reed-Solomon codewords are input to the interleaver):



546. ADSL2 also discloses that the packet that is transmitted using interleaving (and subsequently received using deinterleaving) is comprised of Reed-Solomon codewords.

To spread the Reed-Solomon codeword and, therefore, reduce the probability of failure of the FEC in the presence of impulse noise, the FEC Output Data Frames shall be convolutionally interleaved. ...

ADSL2 at 38, § 7.7.1.5 “Interleaver” (excerpted).

547. Chen in view of ADSL2 therefore each disclose “wherein the packet comprises a header field and a plurality of PTM-TC codewords, a plurality of ATM cells or a plurality of Reed-Solomon codewords.”

iii Yap in view of VDSL1

548. Yap discloses that error correction coding is applied to the packets prior to transmission. Describing the data processing steps, Yap explains that “channel coding such as

convolutional coding and turbo coding is carried out at coding section 104” prior to interleaving. Yap at NOK00084270, 4:12-14.

549. To the extent that Yap does not explicitly disclose the use of Reed-Solomon codewords, as described above in Section VII.A.2, the use of Reed-Solomon coding for error correction coding was well known and would have been an obvious alternative design choice for a POSITA implementing Yap’s invention. A POSITA would have had a reasonable expectation of success that the well known Reed-Solomon coding could be employed in placed of convolutional coding or turbo coding. This would have been nothing more than the substitution of a well known technique to serve a similar purpose. Moreover, VDSL1 discloses a multicarrier communication apparatus in which transmitted and received packets comprise one or more Reed-Solomon codewords.

550. VDSL1 Reed-Solomon encodes transmitted data before that data is interleaved.

After payload adaptation, E_i overhead bytes (see 8.5.5) shall be added to the head-end of each packet of U_i bytes (see Figure 8-3). These bytes are called fast and slow bytes for the fast and slow channel respectively. Next, a sequence of N_i packets of $(E_i + U_i)$ bytes shall be RS-encoded. In order to achieve an integer number of RS-codewords per N_i packets, RS-dummy bytes may have to be inserted.

The number of RS-encoded bytes, B_i , per N_i packets is given by:

$$B_i = \left[N_i \times (E_i + U_i) + D_{RS,i} \right] \times \frac{N_i}{K_i}$$

VDSL1 at NOK00078788, § 8.5.3 “RS encoding.” *See also id.* NOK00078785, §§ 8.4-8.4.1

“Interleaving – General”:

Interleaving shall be used to protect the data against bursts of errors by spreading the errors over a number of Reed-Solomon codewords. The interleave depth shall be programmable with a maximum interleave depth of 64 codewords when the number of octets per codeword (N) equals 255. For smaller values of N the interleave depth can grow nearly proportionately.

551. Thus, Yap in view of VDSL1 renders obvious “wherein the packet comprises a header field and a plurality of PTM-TC codewords, a plurality of ATM cells or a plurality of Reed-Solomon codewords.”

[1.d] and wherein the header field comprises a sequence identifier (SID);

i VDSL1

552. VDSL1 discloses “wherein the header field comprises a sequence identifier (SID).” For example, VDSL1 discloses placing sequence identifiers in the header field of HDLS frames containing special operations channel (SOC) messages. Specifically, the message index (a sequence identifier that identifies the message) is placed in the Address Field of the HDLC frame, which is a component of the header field of the HDLC frame. Figure 12-3 (on page 63) shows the structure of an HDLC frame.

<i>Meaning</i>	<i>Value</i>
	← 1 byte →
<i>Flag</i>	0x7E
<i>Address Field</i>	Address
<i>Control Field</i>	Control
<i>Information Payload</i>	Payload bytes
<i>Check Sequence</i>	FCS
<i>Check Sequence</i>	FCS
<i>Flag</i>	0x7E

Figure 12-3/G.993.1 – Structure of an HDLC frame

553. As another example, messages that are too large for one HDLC frame are broken into segments, each with a specific segment index, which is a sequence identifier (SID), that is in the header field of the HDLC frame. Specifically, the segment index is placed in the Control Field of the HDLC frame, which is a component of the header field of the HDLC frame.

554. Each repeat request message (which is equivalent to a NACK) must contain the message index and the segment index of the segment to be retransmitted; similarly, an acknowledgment of a segment must contain the message index and the segment index of the segment being acknowledged. This is described in Section 12.4.2.6 “Segmentation of messages” (on pages 65-66), and shown below.

12.4.2.6 Segmentation of messages

Some messages could potentially be large, and even larger than the maximal allowed frame size of an HDLC frame (1026 octets). It shall therefore be possible to segment messages before transmission. In order to do this, all messages transmitted during initialization shall receive a number that counts the message. This number shall be stored in one byte and will wrap around in case of overflow. The value 0 shall not be used because it has a special meaning (see later). This means that 255 shall be followed by 1.

This "message index" shall be transmitted in the Address Field of the HDLC frame (see Figure 12-3). The message index allows to track lost messages and to request the retransmission of a particular message. The index shall initially be set to one and shall be incremented with one after the transmission of a message. The index shall not be incremented in case of a REPEAT-REQUEST. The counting of messages shall start when the transmission starts using RQ mode instead of automatic repeat (AR) mode.

A segmentation index (1 byte) shall be included in the Control Field of the HDLC frame. The four MSBs of this field shall indicate the number of segments that make up the total message. The four LSBs shall indicate the index of the current segment. For instance a value 0x93 indicates the third segment of a total of nine. In case the message is not segmented, the value of the field shall be 0x11.

The REPEAT-REQUEST message will behave differently from the other messages. The message index counter shall not be increased when a REPEAT-REQUEST message is sent. Also, the meaning of the message index and the segmentation index is different in this case.

The message index of the REPEAT-REQUEST message shall contain the message index of the message that should be retransmitted. The default value 0 indicates that the last unacknowledged message should be sent (this is why 0 shall never be used as message index for any other message). If this message contains several segments, only the last segment shall be retransmitted.

Likewise, the control field shall contain the segment that should be retransmitted. The Information payload of the REPEAT-REQUEST shall consist of one byte containing the value 0x55.

In the initialization procedure a transmitter should never send two consecutive messages without acknowledgement of the last message. It will always first receive some message from the other side before transmitting again. Therefore, an acknowledgement shall be sent for all but the last segment. Typically, the last segment signals the end of the message and it will therefore be acknowledged by the reply to this message. The ACK-SEG-message (see Table 12-12) shall be used to acknowledge the reception of the other segments. The ACK-SEG-message shall have its own message and segment index, which shall not refer to the segmented message that is being sent.

Once acknowledged, messages (or segments) are not expected to be retransmitted again. Also, a REPEAT-REQUEST that contains a number that is higher than the most recent value of the counter shall be ignored.

In AR mode, segmentation shall be done in the same way but there will be no acknowledgement (ACK-SEG) between different segments of the same message. Segments shall be sent in order.

VDSL1, pages 65-66.

555. Thus, VDSL1 explicitly used sequence identifiers in the headers of initialization messages in HDLC frames. It would have been obvious to a POSITA that such sequential identifiers could also be used on the packets comprising a header field and a plurality of PTM-TC codewords, a plurality of ATM cells or a plurality of Reed-Solomon codewords. For example, PTM-TC codewords have the same frame structure as HDLC frames, as shown in Figure H.2 and accompanying text shown below (VDSL1, page 146), and it would be obvious to a POSITA that the Address Field and Control Field of the PTM-TC codewords can be used for sequence identifiers in exactly the way these fields were used for SOC messages.

H.4.1 Packet encapsulation

For packet encapsulation, an HDLC-type mechanism shall be used with detailed characteristics as specified in the following subclauses.

H.4.1.1 Frame structure

The PTM-TC frame format shall be as shown in Figure H.2. The opening and closing Flag Sequences shall be set to 7E₁₆. They identify the start and the end of the frame. Only one Flag Sequence is required between two consecutive frames.

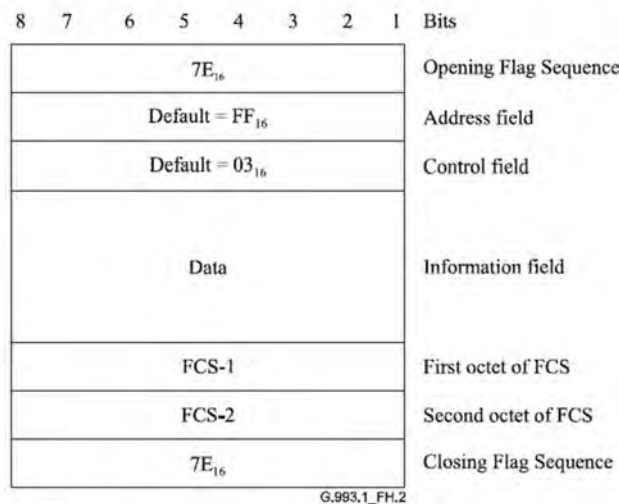


Figure H.2/G.993.1 – PTM-TC frame format

556. It would also have been obvious to a POSITA that sequence identifiers could be used in the same way in the overhead bytes of the frames of the PMS-TC. Thus VDSL1 discloses, or at least makes obvious, “wherein the header field comprises a sequence identifier (SID).”

ii Chen in view of ADSL2

557. Chen and ADSL2 each disclose “wherein the header field comprises a sequence identifier (SID).”

558. Chen describes server and client algorithms which reside on top of the UDP layer, and notes the server routine typically exists at the sender of a video/audio multicast session or at one of the intermediate routers, and the client routine exists within the receiver part of the DR or the end user. Chen at page 159, col. 2 ¶ 3.

559. Chen goes on to describe the application of interleaving as follows:

Interleaving is next applied to a record of 256 such encoded RS words to yield a corresponding interleaved record. The first interleaved word of such a record consists of the first bytes of each of the 256 pre-interleaving words. The 256-th interleaved word is the concatenation of the 256-th bytes of all the pre-interleaving words. It is also possible to have a smaller interleaving depth such as 128, in which case the first interleaved word will encompass the first and second bytes of every pre-interleaving word. Then the first 128 bytes of the 128-th interleaved word are bytes numbered 255 in the original pre-interleaving words, and the second 128 bytes of the 128-th interleaved word are bytes numbered 256 in the original pre-interleaving words.

If the interleaving depth is 256, it is easy to understand that up to 32 consecutive interleaved words can be lost without any ARQ retransmission request from the receiver. Because of interleaving, if any 32 words are lost, only 32 of the 255 bytes of each corresponding pre-interleaving word will be lost. Because the RS decoder can conceal up to $255 - 223 = 32$ lost bytes, such loss is not detrimental and will not result in retransmission requests. For error concealment for multicast traffic over the Internet, minimizing the need for retransmissions is an essential asset, which subsequently leads to the minimizing of no acknowledgement (NAK) and repair-packet implosions [2]. Returning to Fig. 1, we see that the server routine groups up to 12 interleaved words into one UDP data unit (TPDU). A 2-byte interleaved sequence number field is added to each interleaved word, preceded by a 1-byte field denoting an interleaved word, i.e., 11111111. Although each original RS-encoded word carries a type field and a sequence field, these fields are scrambled later by the interleaving, and the need arises for new type and sequence fields. Occasionally or at the request of the client, retransmitted words are also included in some TPDUs. If an interleaving depth of 256 is used, each TPDU contains eight interleaved words; in this case four consecutive TPDUs can be lost without any retransmissions.

Fig. 2(a) shows the byte processing of a transmitted multicast file, as outlined above. Meanwhile, Fig. 2(b) underlines the processing of retransmission packets and shows how to combine them with new interleaved words. Most of the processing above is done offline by multicast applications. The whole video or voice file is FEC-encoded, interleaved and stored before the real-time multicast is started. The pro-

cedure of mixing repairs and interleaving words has to be executed in real time when the multicast session commences.

The format of the NAK frame used by the processing in Fig. 2(b) is shown in Fig. 3. As mentioned before, for the multicast applications at hand, this is the only packet type that the client can send. However, for a video conferencing system, data may also be sent from the client to the server. In Fig. 3, we use 01010101 as the value of the type field to denote NAK messages. A 1-byte field denoting the data length of the NAK message is also added. The subsequent fields give the sequence numbers of the lost words as requested by the client. A final 1-byte CRC field is used for error checking at the server. The client routine in Fig. 4(a) shows the process that takes place once a record of data is read from the UDP layer. Blocks of 259 bytes are sampled from such a record. If the type field indicates a repair word, and if further CRC indicates correct reception, the first 4 bytes are stripped and the following 217 bytes are sent to the receiving buffer. The DL field may be used to separate real data from padding. If the type field indicates an interleaved word, this interleaved word is sent to the de-interleaving buffer. A timer is reset upon the arrival of the first byte of a certain de-interleaved RS word (this may not be the first byte of the word).

Chen at page 160, col. 1 ¶ 2-col. 2 ¶ 1.

560. Moreover, “an ADSL transmission unit can simultaneously convey all of the following: a number of downstream frame bearers, a number of upstream frame bearers, a baseband POTS/ISDN duplex channel, and ADSL line overhead for framing, error control, operations, and maintenance.” ADSL2 at pages i-ii, Summary.

561. In particular, the FEC procedure is detailed below:

7.7.1.4 Forward error correction function

The FEC procedure inserts Reed-Solomon FEC redundancy octets to provide coding gain as an outer coding function to the PMD function. The FEC procedure of latency path function # p shall calculate R_p octets from $M_p \times K_p$ input octets. The octets are appended to the end of the input octets in the structure of FEC Output Data Frame at Reference Point B.

When $R_p = 0$, no redundancy octets are appended and the values in the FEC Output Data Frame are identical to the input values. For all other values of R_p , the following encoding procedure shall be used to create the R_p octets:

The FEC procedure shall take in M_p scrambled Mux Data Frames comprising message octets, $m_0, m_1, \dots, m_{M_p \times K_p - 2}, m_{M_p \times K_p - 1}$. The procedure shall produce R_p redundancy octets $c_0, c_1, \dots, c_{R_p - 2}, c_{R_p - 1}$. These two taken together comprise the FEC codeword of size $M_p \times K_p + R_p$ octets. The R_p redundancy octets shall be appended to the message octets to form the FEC Output Data Frame at Reference Point B.

At the end of the initialization sequence, the FEC Function always starts with the first of M_p Mux Data Frames.

The redundancy octets are computed from the message octets using the equation:

$$C(D) = M(D)D^{R_p} \text{ modulo } G(D)$$

where:

$M(D) = m_0 D^{M_p \times K_p - 1} + m_1 D^{M_p \times K_p - 2} + \dots + m_{M_p \times K_p - 2} D + m_{M_p \times K_p - 1}$ is the message polynomial,

$C(D) = c_0 D^{R_p - 1} + c_1 D^{R_p - 2} + \dots + c_{R_p - 2} D + c_{R_p - 1}$ is the check polynomial, and

$G(D) = \prod (D + \alpha^i)$ is the generator polynomial of the Reed-Solomon code,

where the index of the product runs from $i = 0$ to $R_p - 1$.

That is, $C(D)$ is the remainder obtained from dividing $M(D)D^{R_p}$ by $G(D)$. The arithmetic is performed in the Galois Field GF(256), where α is a primitive element that satisfies the primitive binary polynomial $x^8 + x^4 + x^3 + x^2 + 1$. A data octet ($d_7, d_6, \dots, d_1, d_0$) is identified with the Galois

Field element $d_7\alpha^7 + d_6\alpha^6 + \dots + d_1\alpha + d_0$.

The FEC procedure of the latency path # p creates N_{FECp} octets in the structure of a FEC Output Data Frame at Reference Point B. This procedure is followed by the interleaver procedure.

ADSL2 at page 37, § 7.7.1.4.

562. The overhead channel structure is detailed below:

7.8.2.1 Overhead channel structure

Each latency path that is enabled carries an overhead channel to be transported in the sync octets. Generally, each overhead channel can contain a CRC portion, a bit-oriented portion, and a message-oriented portion over a repeating sequence of sync octets of length SEQ_p . The specific structure of the overhead channel for latency path # p shall have one of four formats as displayed in Table 7-14 depending upon the value of the derived parameter SEQ_p .

The value of SEQ_p shall be calculated as shown in Table 7-14 and depends upon the value of $MSG_{L,p}$ as well as the latency of all paths. The value of SEQ_p shall be implicitly defined through a PARAMS message exchanged during initialization, and shall not be updated otherwise. To determine the value of SEQ_p , the indicator bits shall be allocated to the latency path that has the lowest value of the derived parameter $delay_p$, and the message-based overhead shall be allocated to latency path # $MSG_{L,p}$. If more than one latency path has the same value of $delay_p$, the path with the lowest latency shall be the latency path with lowest $delay_p$ and lowest label p . The values of SEQ_p shall be determined during the initialization procedures, and shall not be changed through on-line reconfiguration or power management transitions not involving the initialization procedures (although the latency path with the lowest delay may change).

An overhead structure frame counter is maintained in each latency path with the frame counter incremented by one for each sync octet transmitted. The overhead structure frame counter starts

from zero at the end of the initialization procedure. When the counter reaches the maximum value SEQ_p and the end of the sequence is reached, the counter is reset and the information sequence is begun again from octet sequence 0. This same counter shall be used to control the behavior of the CRC procedure in 7.7.1.2 and the behavior of the NTR transport procedure in 7.8.1. The value of MSG_C is identified during initialization and shall result in a message-based overhead data rate in the MSG_{min} to MSG_{max} range.

The first sync octet following the initialization sequence shall always contain a CRC octet in each latency path. The value of the CRC octet for the first sync octet following initialization is implementation specific.

ADSL2 at pages 39-40, § 7.8.2.1.

563. Finally, the short initialization procedures are included below:

8.14 Short initialization procedures

A short initialization sequence is defined to allow the ATUs to quickly enter Showtime from a L3 power management state or as a fast recovery procedure from changing of line conditions during Showtime. The Short Initialization Sequence shall be optional for both ATU-C and ATU-R (with indication in G.994.1, see 8.13.2). If the Short Initialization Sequence is supported, the ATU should also support unbalanced bitswap (i.e., type 3 On-Line Reconfiguration with restriction to change b_i , g_i and L_p only, see 9.4.1.1).

The state diagram of the short sequence shall be the same as the one shown in Figure 8-26 to Figure 8-30, with the exception of the entry procedures which shall be as depicted in Figures 8-31 and 8-32. Figure 8-31 shows the entry procedure for an ATU-C initiated short initialization. The ATU-C shall keep transmitting 128 symbols of C-COMB1 followed by 256 symbols of silence (C-QUIET2) until either the ATU-R responds with R-COMB1 during one of the C-QUIET2 states or a vendor discretionary timeout C-T1 is reached. If the short initialization is used as a fast recovery procedure from showtime, the ATU-R should reply to the first transmission of the C-COMB initialization signal.

..

The short initialization procedure may be used for the link state transition from the L3 state to the L0 state (see 9.5.3). Fast error recovery (during the L0 or L2 link state) is through the short initialization procedure. At the start of the short initialization procedure, the ADSL link state shall be changed to the L3 state. When the ATU reaches the Showtime state through the short initialization procedure, the ADSL link shall be in the L0 state (see Figure 9-5).

The short initialization procedure should be completed within 3 s. However, to meet this requirement, proper time budget balancing between ATU-C and ATU-R is required. Table 8-41 lists recommended time budgets for the variable portions of each ATU initialization sequence. Figures 8-33 and 8-34 show the recommended timing diagram for the short initialization procedure.

ADSL2 at pages 147-148, § 8.14.

564. Chen in view of ADSL2 therefore each disclose “wherein the header field comprises a sequence identifier (SID).”

iii Yap in view of VDSL1

565. Yap discloses “wherein the header field comprises a sequence identifier (SID).” As described in Yap:

At transmitter 100, data processing is performed for each individual antenna chain. Different streams of independent data are transmitted from the individual transmission antennas. The input data is first attached the cyclic redundancy check (CRC) code at CRC attaching section 102. Then, channel coding such as convolutional coding and turbo coding is carried out at coding section 104. The coded data will then be interleaved by interleaver 106 to reduce burst errors in the data. M-ary modulation constellation symbol mapping is executed on the interleaved data at mapping section 108. A pilot signal is inserted in the mapped signal at pilot inserting section 110. Pilot signal insertion makes channel evaluation at the receiver straightforward.

Yap at 4:8-20.

[1.e] and receive a plurality of messages using a forward error correction decoder and without using a deinterleaver,

i VDSL1

566. VDSL1 discloses that the transceiver “receive[s] a plurality of messages using a forward error correction decoder and without using a deinterleaver.” For example, VDSL1 discloses transmitting data messages in a “fast” buffer without interleaving the messages. VDSL1 separately discloses transmitting ACK/NACK response messages during initialization without interleaving.

567. Specifically, the PMS-TC sublayer of VDSL1 interleaves data being transmitted in the “slow payload” and does not interleave data in the “fast payload.” The “fast” payload is faster, of course, because it does not include the delay from interleaving, which is an absolute delay that depends on the interleaving depth and the size of the block of data being interleaved.

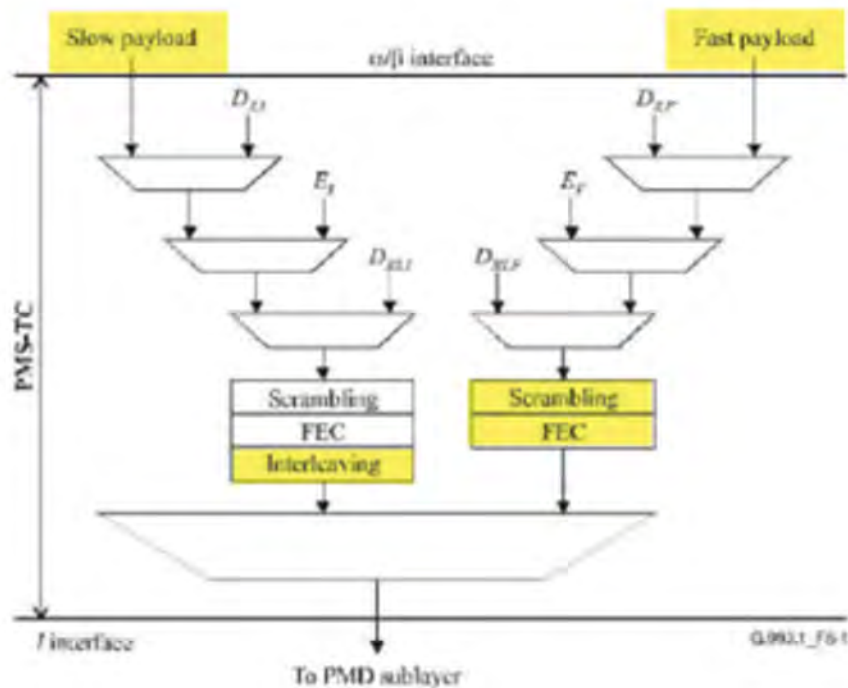


Figure 8-1/G.993.1 – Diagram of PMS-TC sublayer

VDSL1 at NOK00078784, Figure 8-1.

A frame is a set of bytes carried by one DMT symbol. The frame frequency depends on the total length of the cyclic extension (see 9.2.2). A frame shall be composed of two sources: the "fast" buffer and the "interleaved" (or "slow") buffer. The index i refers to parameters related to the fast or interleaved buffers ($i \in \{F, I\}$). The inclusion of the fast buffer shall be optional. When the fast buffer is not included, the interleaved buffer shall have the capability to carry non-interleaved data by setting the interleaver depth to zero.

VDSL1, page 17, § 8.5.1 "Frame description."

568. Similarly, VDSL1 teaches sending initialization messages and ACK or NACK messages without using interleaving. An ACK for a given initialization message means that message was correctly received; if a NACK is sent, it means the message was incorrectly received.

The SOC shall use HDLC-like format with byte stuffing to delineate the messages as specified in ITU-T Rec. G.994.1. A reliable transmission shall be insured by using either an automatic repeat (AR) mode or a repeat request (RQ) mode. The maximum size of an HDLC frame shall be 1026 bytes (this also defines the maximum size of a SOC message segment). This is the size of the pay load before octet stuffing and addition of any flags.

In the AR mode, the message encapsulated in the HDLC frame shall be automatically repeated. At least four idle flags (0x7E) shall be inserted between successive frames.

In the RQ mode, the messages encapsulated in HDLC frame shall be sent once. However, the VTU expecting the message shall have the possibility to request the remote side to repeat it by sending a REPEAT_REQUEST message. This operation is necessary when the expected message has a wrong Check Sequence or when a timeout has expired. After two unsuccessful REPEAT_REQUEST, the initialization shall be aborted. This means the initiating side will restart the G.994.1 handshake after a silent period. After a number of unsuccessful attempts, the modems shall stop all attempts. The number of attempts that is made before a final abort of the initialization shall be chosen by the initiating modem.

A SOC message shall contain an integer number of octets (8 bits per octet). The octets shall be sent least significant bit first. A message is subdivided in fields. A field can contain more than one byte. In this case, the field shall be split in bytes with the byte containing the most significant bits sent first. For example a field of 16 bits made of the bits $m_{15}, \dots, m_0$ shall be segmented in a first byte $B_0 = m_{15} \dots m_8$ and a second byte $B_1 = m_7 \dots m_0$. Some fields can be merged together to form a logical entity called a macro-field, such as "PSD descriptor", "Band descriptor".

The structure of an HDLC frame is illustrated in Figure 12-3.

<i>Meaning</i>	<i>Value</i>
	← 1 byte →
<i>Flag</i>	0x7E
<i>Address Field</i>	Address
<i>Control Field</i>	Control
<i>Information Payload</i>	Payload bytes
<i>Check Sequence</i>	FCS
<i>Check Sequence</i>	FCS
<i>Flag</i>	0x7E

Figure 12-3/G.993.1 – Structure of an HDLC frame

VDSL1 at NOK00078833-834, § 12.4.2.1 “Message format.”

569. These messages include a type field:

The information payload of every SOC message shall start with a field (with a length of one byte) containing a unique code to identify the message and to allow fast and easy recognition of each SOC message. The message codes of all the messages sent during the initialization sequence are shown in Table 12-12 in hexadecimal notation. They are numbered in the order in which they appear. The messages originating at the VTU-O have the MSB equal to zero, the messages originating at VTU-R have MSB equal to one. Some one-byte messages have special codes.

Table 12-12/G.993.1 – Message codes for the SOC messages

SOC message	Message code
O/R-REPEAT_REQUEST	0x55 (Note)
R-ACK	0x00 (Note)
R-NACK	0xFF (Note)
O/R-ACK-SEG	0x0F (Note)
O-SIGNATURE	0x01
O-UPDATE _n	0x02
O-MSG1	0x03
O-MSG2	0x04

Table 12-12/G.993.1 – Message codes for the SOC messages

SOC message	Message code
O-CONTRACT _n	0x05
O-B&G	0x06
R-MSG1	0x81
R-MSG2	0x82
R-CONTRACT1	0x83
R-MARGIN _n	0x84
R-B&G	0x85
NOTE – This is the entire payload of the message.	

VDSL1 at NOK00078834, § 12.4.2.4 “Message codes.

570. VDSL1 describes the initialization messages listed in the above tables in detail. One of these messages is the R-MSG2, which is sent from the VTU-R to the VTU-O.

12.4.6.3.1.1 R-MSG2

This message contains information about the capabilities of the VTU-R for bit allocation. The content of R-MSG2 shall be as specified in Table 12-32.

Table 12-32/G.993.1 – Description of R-MSG2

Field	Field or macro-field type	Remark
Message descriptor	1 byte	See Table 12-12.
Maximal constellation size in upstream ($B_{\text{max u}}$)	1 byte	Maximum number of bits per tone
RS setting supported by VTU-R	1 byte	0: Only mandatory settings 1: All settings
Interleaver settings supported by the VTU-R	1 byte	0x00: Only mandatory settings 0xFF: All settings 0xNN NN = Number of additional (i.e., non-mandatory) settings in hexadecimal (NN $\neq$ 0x00 and NN $\neq$ 0xFF)
Detailed interleaver setting description	0 byte if NN = 0x00 or NN = 0xFF; 4xNN otherwise	Interleaver descriptor (See Table 12-24.)
Maximal power transmitted	1 byte (unsigned)	In dBm by steps of 0.25 dBm
Maximal interleaver memory	3 bytes	In bytes (Note 1)
Maximum number of EOC bytes per frame in upstream	1 byte	Number of EOC bytes per frame
Maximum number of VOC bytes per frame in upstream	1 byte	Number of VOC bytes per frame
Support of express bit swapping	1 byte	0x00: Not supported 0xFF: Supported
f_{max}	1 byte	Specify maximum value of f_{max} supported by the VTU-R (Note 2)
NOTE 1 – The interleaver memory is computed as $M \times I \times (I - 1)$.		
NOTE 2 – Specification of $f_{\text{max}} = k$ means that all values from 0, 1, ..., k are supported.		

VDSL1 at NOK00078853, § 12.4.6.3.1.1 “R-MSG2.”

571. The VTU-O then sends the O-MSG2 to the VTU-R.

12.4.6.2.1.1 O-MSG2

This message contains information about the capability of VTU-O for contract negotiation. The content of O=MSG2 shall be as shown in Table 12-23.

Table 12-23/G.993.1 – Description of O-MSG2

Field content	Field or macro-field type	Remark
Message descriptor	1 byte	See Table 12-12.
Minimal SNR margin	1 byte	In units of 0.5 dB
Maximal constellation size in downstream ($B_{\text{max},d}$)	1 byte	Maximum number of bits per tone
RS setting supported by VTU-O	1 byte	0x00: Only mandatory settings 0xFF: All settings
Interleaver settings supported by the VTU-O	1 byte	0x00: Only mandatory settings 0xFF: All settings 0xNN NN = Number of additional (i.e., non-mandatory) settings in hexadecimal (NN $\neq$ 0x00 and NN $\neq$ 0xFF)
Detailed interleaver setting description	0 byte if NN = 0x00 or NN = 0xFF, 4xNN otherwise	Interleaver descriptor (see Table 12-24)
Maximal power in downstream	1 byte (unsigned)	In units of 0.25 dBm
Maximum interleaver delay	1 byte	In ms by steps of 0.5 ms (Note 1)
Maximum number of EOC bytes per frame in downstream	1 byte	Number of EOC bytes per frame
Maximum number of VOC bytes per frame in downstream	1 byte	Number of VOC bytes per frame

Table 12-23/G.993.1 – Description of O-MSG2

Field content	Field or macro-field type	Remark
Support of express bit swapping	1 byte	0x00: Not supported 0xFF: Supported
j_{max}	1 byte	Maximum value of j_{max} supported by the VTU-O (Note 2)
NOTE 1 – This field can be set to 0 in order to emulate the first channel. This field is used for the creation of R-CONTRACT1 even if dual latency is used at the end.		
NOTE 2 – Specification of $j_{\text{max}} = k$ means that all values 0, 1, ..., k are supported.		

j_{max} is defined in 12.4.6.2.1.3.

The structure of the interleaver descriptor shall be as given in Table 12-24. I , q and M are the interleaver parameters (see 8.4.2).

Table 12-24/G.993.1 – Interleaver descriptor

Field	Field or macro-field type
I	1 byte
q	1 byte
M_{min}	1 byte
M_{max}	1 byte
NOTE – The four fields are repeated for each interleaver setting.	

VDSL1 at NOK00078848-849, 12.4.6.2.1.1 “O-MSG2”.

572. VDSL1 allows for segmenting initialization messages that are large – which the R-MSG2 and O-MSG2 may very well be – into multiple frames, with each segment being given a “segment index” and the overall message having a “message index” that tracks the structure of the message’s segments, so they can be reassembled in order if they are received out of order. The receiving side will acknowledge each segment received (except for the last segment) with a segment acknowledgement message. The receiver requests retransmission of segments not correctly received by sending a “REPEAT-REQUEST” message. The last segment of a message is acknowledged with a standard acknowledgment message.

12.4.2.6 Segmentation of messages

Some messages could potentially be large, and even larger than the maximal allowed frame size of an HDLC frame (1026 octets). It shall therefore be possible to segment messages before transmission. In order to do this, all messages transmitted during initialization shall receive a number that counts the message. This number shall be stored in one byte and will wrap around in case of overflow. The value 0 shall not be used because it has a special meaning (see later). This means that 255 shall be followed by 1.

This "message index" shall be transmitted in the Address Field of the HDLC frame (see Figure 12-3). The message index allows to track lost messages and to request the retransmission of a particular message. The index shall initially be set to one and shall be incremented with one after the transmission of a message. The index shall not be incremented in case of a REPEAT-REQUEST. The counting of messages shall start when the transmission starts using RQ mode instead of automatic repeat (AR) mode.

A segmentation index (1 byte) shall be included in the Control Field of the HDLC frame. The four MSBs of this field shall indicate the number of segments that make up the total message. The four LSBs shall indicate the index of the current segment. For instance a value 0x93 indicates the third segment of a total of nine. In case the message is not segmented, the value of the field shall be 0x11.

The REPEAT-REQUEST message will behave differently from the other messages. The message index counter shall not be increased when a REPEAT-REQUEST message is sent. Also, the meaning of the message index and the segmentation index is different in this case.

The message index of the REPEAT-REQUEST message shall contain the message index of the message that should be retransmitted. The default value 0 indicates that

the last unacknowledged message should be sent (this is why 0 shall never be used as message index for any other message). If this message contains several segments, only the last segment shall be retransmitted.

Likewise, the control field shall contain the segment that should be retransmitted. The Information payload of the REPEAT-REQUEST shall consist of one byte containing the value 0x55.

In the initialization procedure a transmitter should never send two consecutive messages without acknowledgement of the last message. It will always first receive some message from the other side before transmitting again. Therefore, an acknowledgement shall be sent for all but the last segment. Typically, the last segment signals the end of the message and it will therefore be acknowledged by the reply to this message. The ACK-SEG-message (see Table 12-12) shall be used to acknowledge the reception of the other segments. The ACK-SEG-message shall have its own message and segment index, which shall not refer to the segmented message that is being sent.

Once acknowledged, messages (or segments) are not expected to be retransmitted again. Also, a REPEAT-REQUEST that contains a number that is higher than the most recent value of the counter shall be ignored.

In AR mode, segmentation shall be done in the same way but there will be no acknowledgement (ACK-SEG) between different segments of the same message. Segments shall be sent in order.

VDSL1 at NOK00078835-836, § 12.4.2.6 “Segmentation of messages.”

573. VDSL1 thus discloses that the transceiver “receive[s] a plurality of messages using a forward error correction decoder and without using a deinterleaver.”

ii Chen in view of ADSL2

574. Chen in view of ADSL2 renders obvious that the transceiver “receive[s] a plurality of messages using a forward error correction decoder and without using a deinterleaver.” Specifically, both Chen and ADSL2 disclose receiving ACKs or NAKs that are not interleaved.

575. Chen discloses sending a NAK message that is not interleaved. Instead, Chen uses a CRC to protect NAK messages from errors.

The format of the NAK frame used by the processing in Fig. 2(b) is shown in Fig. 3. As mentioned before, for the multicast applications at hand, this is the only packet type that the client can send. However, for a video conferencing system, data may

also be sent from the client to the server. In Fig. 3, we use 01010101 as the value of the type field to denote NAK messages. A 1-byte field denoting the data length of the NAK message is also added. The subsequent fields give the sequence numbers of the lost words as requested by the client. A final 1-byte CRC field is used for error checking at the server. The client routine in Fig. 4(a) shows the process that takes place once a record of data is read from the UDP layer. Blocks of 259 bytes are sampled from such a record. If the type field indicates a repair word, and if further CRC indicates correct reception, the first bytes are stripped and the following 217 bytes are sent to the receiving buffer. The DL field may be used to separate real data from padding. If the type field indicates an interleaved word, this interleaved word is sent to the de-interleaving buffer. A timer is reset upon the arrival of the first byte of a certain deinterleaved RS word (this may not be the first byte of the word).

Chen at NOK00075560, col. 2 ¶ 2. *See also id.* Figs. 1, 3 (Fig. 1 shows an interleaved packet format, while Fig. 3 shows a specific NAK packet format, which is not interleaved).

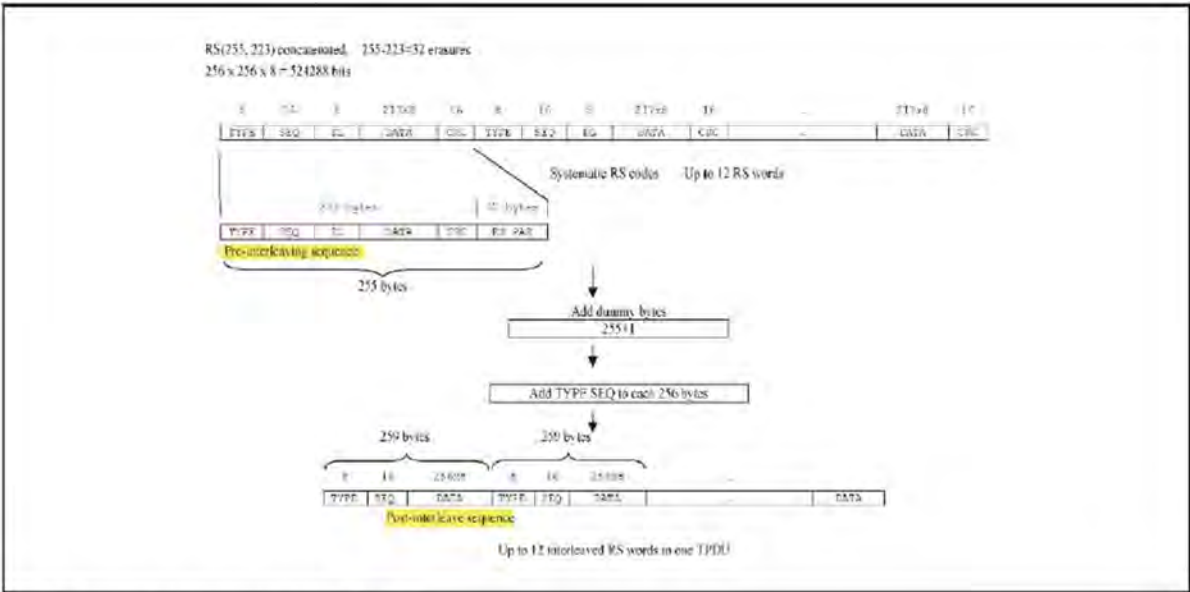


Figure 1: Transportation-layer protocol data unit (TPDU) format.

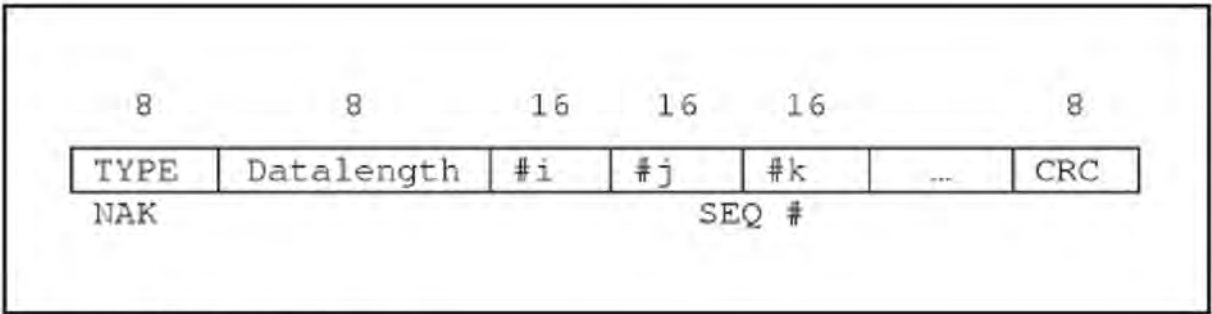


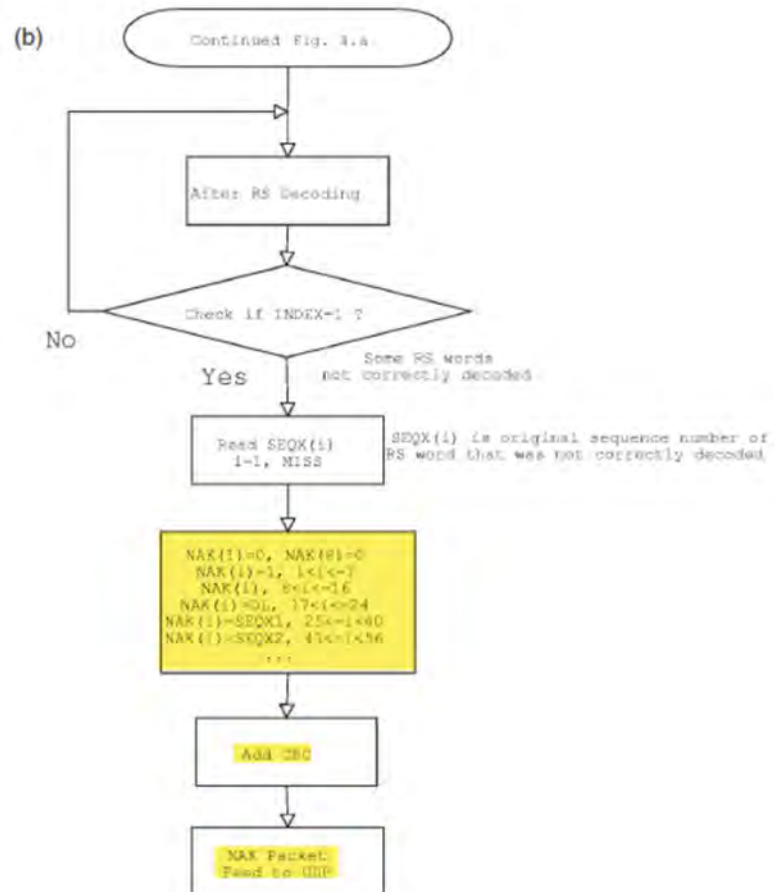
Figure 3: NAK packet format.

576. Chen describes generating a NAK message which, as described above, is not interleaved.

Fig. 4(b) shows the step-by-step generation of the NAK request at the client. As shown in Fig. 2(b), we execute the processing for such a NAK message as it arrives at the server. Fig. 4(b) also shows the multiplexing of RS (interleaved) words and repair (retransmission) words in server transmission time. Priority is given to repair words, and interleaved words are transmitted when there are no repairs to be sent ($XZ = 0$).

The timing diagram of Fig. 5 shows a sequence of events taking place under the control of our new scheme for a typical scenario. Equal-length interleaving windows for data records N , $N + 1$, $N + 2$ are used. The end of the next window sets the time limit for playing received packets of the current record at the MPEG player. Since received packets of each record are reorganized in the interleaving window and then played in sequence, this technique has a buffer effect that can help to guarantee delay-jitter-free playing at the receiver. This figure also shows the NAK and retransmissions for record N due to the packet loss. For record $N + 1$, the RS FEC decoder can correct the loss and errors without asking for any retransmission. For record $N + 2$, no loss and errors are encountered for the first K bytes of RS symbols; therefore the RS decoder is not called, and no NAKs are generated.

Chen at NOK00075561, col. 2 ¶¶ 1-2. *See also id.* Fig. 4b (showing that NAK packets are protected by a CRC check and not interleaved prior to transmission).



See also id. Fig. 5 (NAK is not described as being interleaved) (cropped).

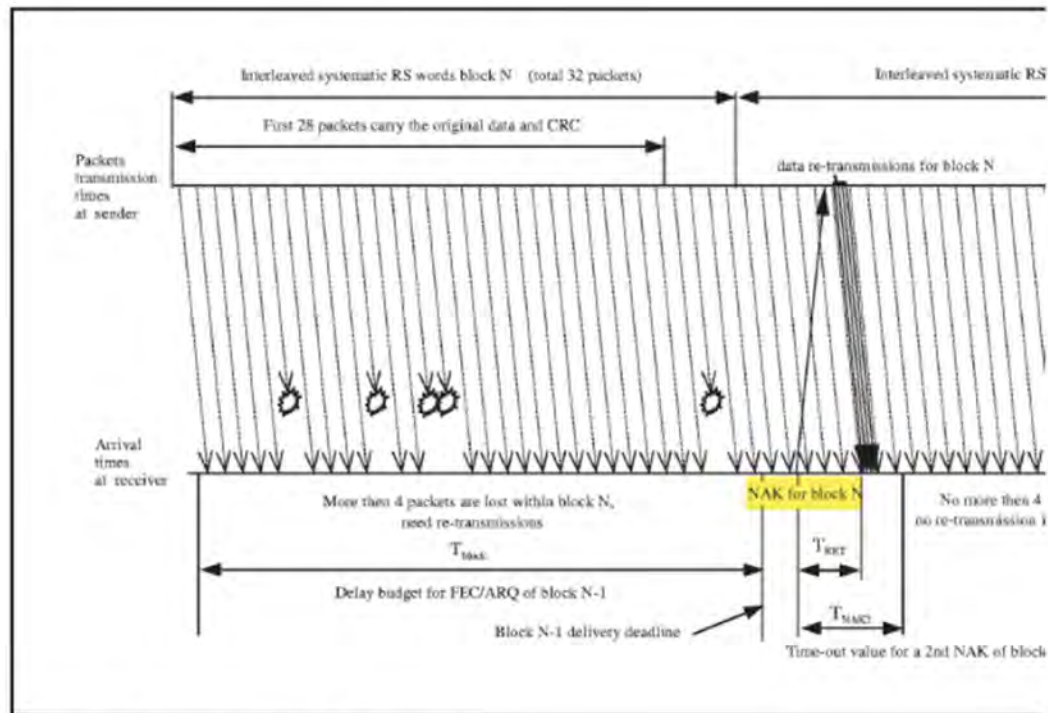


Figure 5: The hybrid FEC/ARQ system timing.

577. ADSL2 sends non-interleaved ACK or NACK (which is another abbreviation for NAK) messages to indicate whether certain overhead messages (or segments of those messages) were correctly received.

Each transmitted message segment shall be acknowledged by the far-end with a segment acknowledge message, except the last one. The last command message segment shall be acknowledged by the far-end with the appropriate response message (identical to the non-segmented case). The last response message segment shall not be acknowledged. Table 7-17a defines the segment acknowledge message. The following shall apply to all encapsulated segment acknowledge messages:

- HDLC frame shall contain 5 message octets ($P = 5$);
- The two least significant bits of the address field shall be identical to the corresponding bits in the address field of the acknowledged message segment. All other bits of the address field shall be set to 02;
- The second least significant bit of the control field shall either indicate a command message (request to continue the response, e.g., L2 grant) or a response message (request to continue the command, e.g., OLR);

- The least significant bit of the control field shall toggle for each acknowledgement message as it normally toggles for each command/response message (see 7.8.2.4.2);
- All other bits of the control field shall be set to 0₂.

Table 7-17a/G.992.3 – Segment acknowledge message

Message octet number	Message octet definition
Octet 1	Message designator 1111 0000 _b for acknowledgement of high priority message segment 1111 0001 _b for acknowledgement of normal priority message segment 1111 0010 _b for acknowledgement of low priority message segment
Octet 2	Message type of segment acknowledge message 01 ₁₆
Octet 3	Acknowledged message segment ID (in range 1 to 7)
Octet 4	Message Designator (first message octet of acknowledged message segment)
Octet 5	Message Type (second message octet of acknowledged message segment)

Every message segment shall be acknowledged by the far-end before the next message segment is transmitted. The segment acknowledge message shall not be segmented.

Timeouts shall be defined as follows:

- For a non-segmented message, timeout shall apply between the last octet of the command message transmitted and the first octet of the response message received.
- For a segmented response message, the timeout shall apply between the last octet of the command message transmitted and the first octet of the first message segment received.
- A timeout corresponding to the command priority shall apply between the last octet of the message segment transmitted and the first octet of the segment acknowledge message received.
- Another timeout corresponding to the command priority shall apply between the last octet of the segment acknowledge message transmitted and the first octet of the next message segment received.

If a timeout expires, the transmitter may repeat the last transmitted message. This repeated message can be an unsegmented command message, a command/response message segment (except the last segment of a response message), or a segment

acknowledge message. Alternately, the ATU may abandon the message after an implementation-specific number of retransmissions.

ADSL2 at 43-46, § 7.8.2.4.3 “Overhead message segmentation.”

578. ADSL2 provides for similar ACKs or NACKs for other messages. Here, because the ACKs and NACKs are generated during the exchange phase, they are not interleaved. *See, e.g.:*

At the start of the Exchange Phase, the ATU-C shall transition to the C-TREF1-LD state (in which C-TREF symbols shall be transmitted until the first R-MSGx-LD message is received).

If the ATU-C receives an R-MSGx-LD message, the ATU-C shall transition to the C-ACK or C-NACK state within 128 symbols from the end of the R-MSGx-LD state. If the R-MSGx-LD message is successfully received, the ATU-C shall transition to the C-ACK state (in which a positive acknowledgment C-ACK message shall be transmitted). Instead, if a decoding error occurs (i.e., the CRC locally computed at the ATU-C does not correspond to the CRC transmitted by the ATU-R), the ATU-C shall transition to the C-NACK state.

The C-ACK message shall be represented by the "01010101" octet and shall be transmitted over 64 symbol periods using the same modulation technique as the loop diagnostics information bearing messages. No CRC shall be added to the C-ACK message. In the C-NACK state, the ATU-C shall transmit 64 C-TREF symbols. Note that from the ATU-R's perspective, this is equivalent to the ATU-C not responding to the R-MSGx-LD message.

At the end of the C-ACK or C-NACK state, the ATU-C shall transition to the C-TREF2-LD state (in which 256 C-TREF symbols shall be transmitted). During the C-TREF2-LD state, the ATU-R transitions to the R-QUIET2-LD state (because the R-ACK message is successfully received and no more R-MSGx-LD messages remain to be transmitted) or the ATU-R transitions to the R-SEGUE-LD state (because no, or a corrupted C-ACK message is received or more R-MSGx-LD messages remain to be transmitted). At the end of the C-TREF2-LD state, the ATU-C shall transition to the C-SEGUE-LD state (if the ATU-R has transitioned to the R-QUIET2-LD state) or shall return to the C-TREF1-LD state (if the ATU-R has returned to the R-SEGUE-LD state).

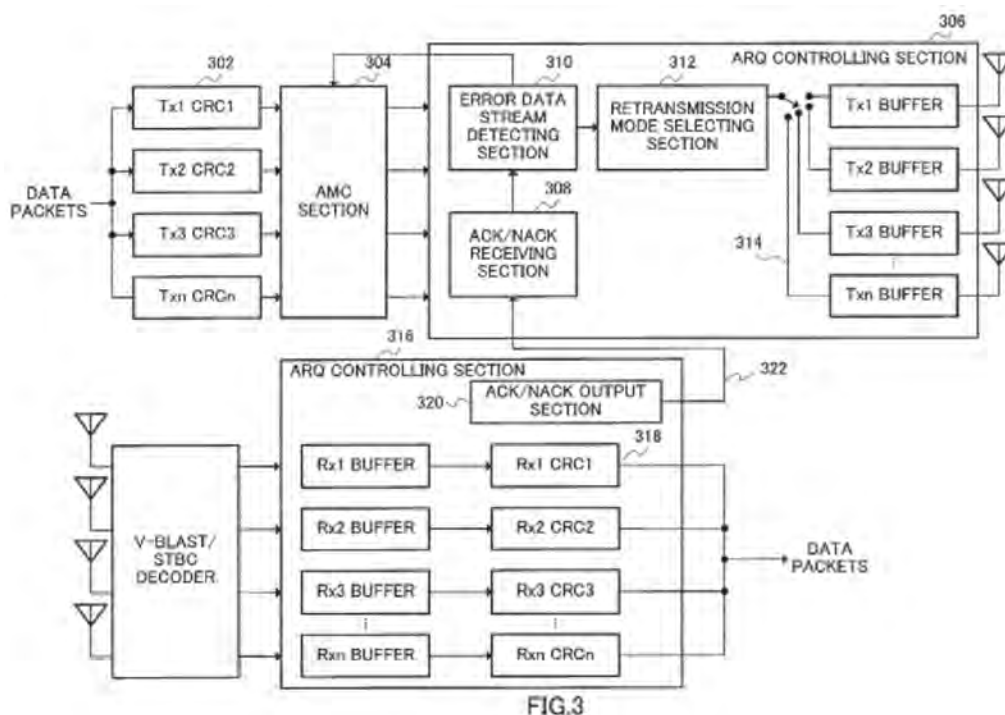
Note that, as a result of a corrupted C-ACK message, the ATU-C could successfully receive the same message twice. In this case, the ATU-C shall ignore the second identical (same Sequence Number) message.

ADSL2 at 158, § 8.15.5.1.2 “Message flow, acknowledgement and retransmission.”

Chen in view of ADSL2 therefore renders obvious that the transceiver “receive[s] a plurality of messages using a forward error correction decoder and without using a deinterleaver.” Both Chen and ADSL2 disclose receiving ACKs or NACKs/NAKs without interleaving.

iii Yap in view of VDSL1

579. Yap discloses a multicarrier transceiver that “receive[s] a plurality of messages using a forward error correction decoder and without using a deinterleaver.” After CRC error detection is performed at the receiver, the receiver then transmits either an acknowledgement or negative acknowledgement (ACK/NACK) to the transmitter to indicate whether retransmission is required. “If it is determined that the checked packet does not include error, acknowledgment (ACK) is transmitted to the transmitter and the transmitter does not retransmit the packet. If there is an error, a negative acknowledgment (NACK) is transmitted to transmitter 100 for retransmission request.” Yap at NOK00084270, 4:49-53. The message transmission is also shown at element 322 in Figure 3:



With regard to the ACK/NACK messaging in Figure 3, Yap explains: “As illustrated in FIG. 3, each antenna chain will have its dedicated CRC attaching section 302. Therefore, at ARQ controlling section 316 of the receiver, every data packet on each individual receiving antenna chain will undergo CRC for error detection in CRC processing section 318. The receiver will then feedback ARQ information related to each of the data streams from ACK/NACK output section 320 via fast ARQ feedback channel 322 to a plurality of ACK/NACK receiving sections 308 at ARQ controlling section 306 of the transmitter.” Yap at NOK00084270, 4:57-66. *See also* Figures 4-7. A POSITA would understand that the “fast ARQ feedback channel 322” transmits the ACK/NACK message to the transmitter without using interleaving.

[1.f] wherein each message of the plurality of messages is received in a different DMT symbol

i VDSL1

580. VDSL1 discloses “wherein each message of the plurality of messages is received in a different DMT symbol.” The SOC messages discussed above are each received in a different DMT symbol. Indeed, during the channel analysis and exchange phase, only one message is sent from a transceiver to another transceiver at a time. This is illustrated in Figure 12-7 (VDSL1, page 77), shown below.

12.4.6 Channel analysis and exchange

Figure 12-7 gives an overview of the sequence of SOC messages and symbol types during the channel analysis and exchange state.

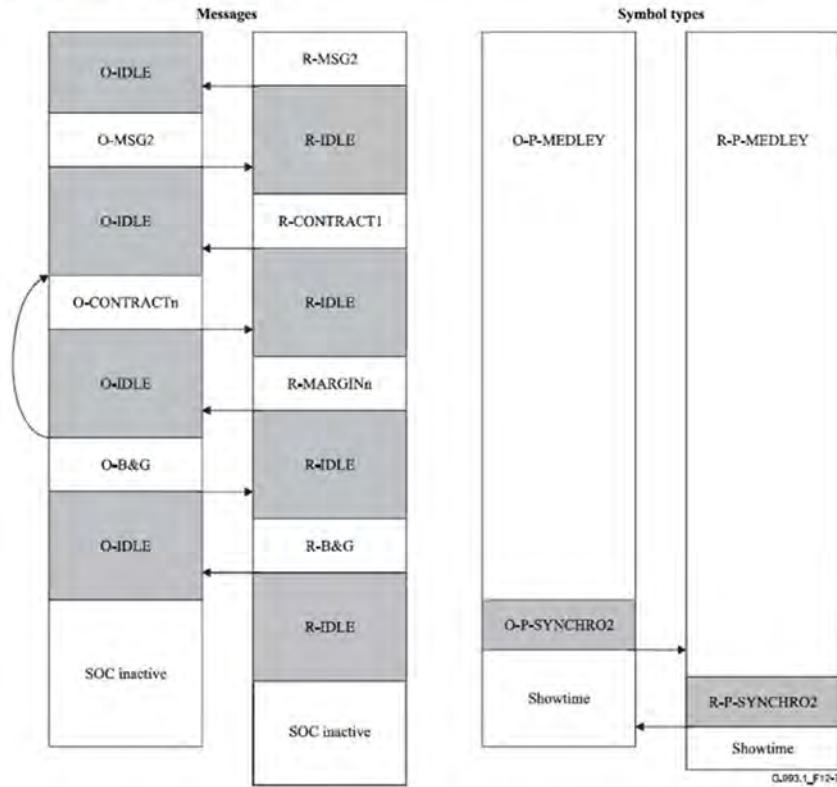


Figure 12-7/G.993.1 – Timeline of the channel analysis and exchange phase

581. Additionally, for the PMS-TC, VDSL1 defines a “frame” as “a set of bytes carried by one DMT symbol. VDSL1 at 17 (frame description). VDSL1 then provides:

In the transmit direction, the PMD layer shall receive input frames from the PMS-TC sublayer. A frame shall contain exactly the number of bytes that will be modulated onto one DMT symbol. This shall be an integer number. Each carrier shall have a number of bits assigned to it during initialization. The bits that are to be modulated on a carrier shall be encoded into constellation points according to the rules given in 9.2.5. After encoding, the carriers shall be modulated and summed using an IDFT. The resulting digital signal shall be cyclically extended and windowed before being sent towards the transmission medium over the U interface.

VDSL1 at 22.

ii Chen in view of ADSL2

582. Chen in view of ADSL2 renders obvious that the transceiver “wherein each message of the plurality of messages is received in a different DMT symbol.”

583. Chen explains that “[e]ach packet contains eight interleaved RS words, so that the loss of one packet implies the loss of only eight symbols in each of 256 RS-encoded words”:

Each packet contains eight interleaved RS words, so that the loss of one packet implies the loss of only eight symbols in each of 256 RS-encoded words. Since (255, 223) Reed-Solomon codes can correct 32 erasures per RS word, the loss of four packets will lead to 32 erasures in each de-interleaved RS word, which is a correctable level of loss. When the number of packets lost is greater than four, a NAK will be generated. The sender will retransmit interleaved words containing only original data (no parity bytes). Equation (1) gives the probability that a packet cannot be RS decoded correctly and has to be retransmitted. Note that, as mentioned above, if 0 or 1 or 2, 3, 4 (less than $(N - K)$) packets are lost, then 0 or 8 or 16, 24, 32 bytes will be lost in the de-interleaved RS words, and there are no final decoding errors:

$$P' = \frac{1}{N} \sum_{i=N-K+1}^N i \binom{N}{i} (1-P)^{N-i} P^i. \quad (1)$$

For a systematic-RS-words block, no NAK is generated when the number of packets lost is less than $(N - K)$. Therefore, the probability that no NAK is generated for a block is

$$P_{\text{no NAK}} = \sum_{i=0}^{N-K} \binom{N}{i} (1-P)^{N-i} P^i. \quad (2.a)$$

Chen at page 163, col. 1 ¶¶ 1-2.

584. Further, ADSL2 discloses this limitation. First, ADSL2 defines DMT as discrete multitone. ADSL2 at page 6. With regard to modulation, ADSL2 provides that the modulator shall modulate a constellation encoder output data frame or sync frame into a DMT symbol, and that a DMT symbol consists of a set of subcarriers. ADSL2 at page 89, §§ 8.8 and 8.8.1. The inverse discrete fourier transform (IDFT) is used to modulate a constellation encoder output data frame onto the DMT subcarriers. ADSL2 at page 91, § 8.8.2.

585. Additionally, ADSL2 covers the basic functions of initialization for a physically connected ATU-R and ATU-C to establish a communications link (ADSL2 at pages 104-105, § 8.13.1.1), along with initialization with hyperframe (ADSL2 at pages 254-255, § C.8.13.1).

586. Yap discloses “wherein each message of the plurality of messages is received in a different DMT symbol.” As explained in Yap:

At transmitter **100**, data processing is performed for each individual antenna chain. Different streams of independent data are transmitted from the individual transmission antennas. The input data is first attached the cyclic redundancy check (CRC) code at CRC attaching section **102**. Then, channel coding such as convolutional coding and turbo coding is carried out at coding section **104**. The coded data will then be interleaved by interleaver **106** to reduce burst errors in the data. M-ary modulation constellation symbol mapping is executed on the interleaved data at mapping section **108**. A pilot signal is inserted in the mapped signal at pilot inserting section **110**. Pilot signal insertion makes channel evaluation at the receiver straightforward.

Yap at 4:8-20.

[1.g] and wherein at least one message of the plurality of messages includes an acknowledgement (ACK) or a negative acknowledgement (NACK) of the transmitted packet.

587. VDSL1 renders obvious “wherein at least one message of the plurality of messages includes an acknowledgement (ACK) or a negative acknowledgement (NACK) of the transmitted packet.” VDSL1 explicitly discloses wherein at least one of the plurality of messages includes an acknowledgement (ACK) or negative acknowledgement (NACK) because the SOC messages include ACK and NACK messages. This is disclosed, for example, in Table 12-12 and accompanying text shown below (VDSL1, page 64).

12.4.2.4 Message codes

The information payload of every SOC message shall start with a field (with a length of one byte) containing a unique code to identify the message and to allow fast and easy recognition of each SOC message. The message codes of all the messages sent during the initialization sequence are shown in Table 12-12 in hexadecimal notation. They are numbered in the order in which they appear. The messages originating at the VTU-O have the MSB equal to zero, the messages originating at VTU-R have MSB equal to one. Some one-byte messages have special codes.

Table 12-12/G.993.1 – Message codes for the SOC messages

SOC message	Message code
O/R-REPEAT_REQUEST	0x55 (Note)
R-ACK	0x00 (Note)
R-NACK	0xFF (Note)
O/R-ACK-SEG	0x0F (Note)
O-SIGNATURE	0x01
O-UPDATE _n	0x02
O-MSG1	0x03
O-MSG2	0x04

588. As another example, VDSL1 discloses that the initialization messages can be broken into segments, with each segment having a segment index. Any repeat request message (which is equivalent to a NACK) will contain the segment index of the segment to be retransmitted; similarly, an acknowledgment of a segment will contain the segment index of the segment being acknowledged.

12.4.2.6 Segmentation of messages

Some messages could potentially be large, and even larger than the maximal allowed frame size of an HDLC frame (1026 octets). It shall therefore be possible to segment messages before transmission. In order to do this, all messages transmitted during initialization shall receive a number that counts the message. This number shall be stored in one byte and will wrap around in case of overflow. The value 0 shall not be used because it has a special meaning (see later). This means that 255 shall be followed by 1.

This “message index” shall be transmitted in the Address Field of the HDLC frame (see Figure 12-3). The message index allows to track lost messages and to request the retransmission of a particular message. The index shall initially be set to one and shall be incremented with one after the transmission of a message. The index shall not be incremented in case of a REPEAT-REQUEST. The counting of messages shall start when the transmission starts using RQ mode instead of automatic repeat (AR) mode.

A segmentation index (1 byte) shall be included in the Control Field of the HDLC frame. The four MSBs of this field shall indicate the number of segments that make

up the total message. The four LSBs shall indicate the index of the current segment. For instance a value 0x93 indicates the third segment of a total of nine. In case the message is not segmented, the value of the field shall be 0x11.

The REPEAT-REQUEST message will behave differently from the other messages. The message index counter shall not be increased when a REPEAT-REQUEST message is sent. Also, the meaning of the message index and the segmentation index is different in this case.

The message index of the REPEAT-REQUEST message shall contain the message index of the message that should be retransmitted. The default value 0 indicates that the last unacknowledged message should be sent (this is why 0 shall never be used as message index for any other message). If this message contains several segments, only the last segment shall be retransmitted.

Likewise, the control field shall contain the segment that should be retransmitted. The Information payload of the REPEAT-REQUEST shall consist of one byte containing the value 0x55.

In the initialization procedure a transmitter should never send two consecutive messages without acknowledgement of the fast message. It will always first receive some message from the other side before transmitting again. Therefore, an acknowledgement shall be sent for all but the last segment. Typically, the last segment signals the end of the message and it will therefore be acknowledged by the reply to this message. The ACK-SEG-message (see Table 12-12) shall be used to acknowledge the reception of the other segments. The ACK-SEG-message shall have its own message and segment index, which shall not refer to the segmented message that is being sent.

Once acknowledged, messages (or segments) are not expected to be retransmitted again. Also, a REPEAT-REQUEST that contains a number that is higher than the most recent value of the counter shall be ignored.

In AR mode, segmentation shall be done in the same way but there will be no acknowledgement (ACK-SEG) between different segments of the same message. Segments shall be sent in order.

VDSL1 at NOK00078835-836, § 12.4.2.6 “Segmentation of messages.”

589. VDSL1 does not specify that data packets transmitted during Showtime using interleaving (such as those sent in the “slow” payload) are retransmitted. However, as noted above, retransmission was a common solution to channel noise such as impulse noise, or unpredictable burst errors in a channel. § VII.A.4. And as already discussed, VDSL1 uses retransmission to ensure

that initialization messages are received, which will ensure that the DSL transceiver reaches “Showtime” (when data is transmitted) more reliably and faster, even though the initialization messages are not protected from channel noise by interleaving. A POSITA would have found adding retransmission using non-interleaved ACK/NACK messages to VDSL1 an obvious, even routine, solution that allowed for greater flexibility in responding to channel noise. For example, the IEEE 802.16 “WiMAX” standard used interleaving with hybrid ARQ retransmission using non-interleaved ACK/NACK messages, so that interleaved packets that are not correctly received are re-sent. § VII.B. WiMAX used a combined interleaving/hybrid ARQ mechanism to provide high data rates in a wireless system, which suggests that the combination of these elements would be desirable in a wired system like VDSL1.

590. Specifically, a POSITA would have used VDSL1’s “fast” (non-interleaved) data payload to send ACK/NACK messages corresponding to data transmitted during Showtime. *See* VDSL1 at NOK00078787, § 8.5.1 “Framing description.” A POSITA would have further incorporated the message and segment index system VDSL1 used to enable the retransmission of initialization messages/segments. VDSL1 at NOK00078835-836, § 12.4.2.6 “Segmentation of messages.” Each of these elements were already present in VDSL1 and their functionality is not changed by their use with a system that allows retransmission of Showtime data.

591. A POSITA would have understood that this modification would have had a predictable result. Indeed, as described in §§ VII.A.4, VII.B, IX.C, and IX.G *supra*, the prior art included many examples where retransmission was incorporated into a data communication system such as WiMAX or a wired or wireless Internet connection. Thus a POSITA would have had a reasonable expectation of success in modifying VDSL1 as described herein.

592. VDSL1 therefore renders obvious discloses “wherein at least one message of the plurality of messages includes an acknowledgement (ACK) or a negative acknowledgement (NACK) of the transmitted packet.”

ii Chen in view of ADSL2

593. Chen discloses “wherein at least one message of the plurality of messages includes an acknowledgement (ACK) or a negative acknowledgement (NACK) of the transmitted packet.”

594. Chen discloses that the NAK sent without interleaving is sent in response to an incorrectly-received interleaved message.

The format of the NAK frame used by the processing in Fig. 2(b) is shown in Fig. 3. As mentioned before, for the multicast applications at hand, this is the only packet type that the client can send. However, for a video conferencing system, data may also be sent from the client to the server. In Fig. 3, we use 01010101 as the value of the type field to denote NAK messages. A 1-byte field denoting the data length of the NAK message is also added. The subsequent fields give the sequence numbers of the lost words as requested by the client. A final 1-byte CRC field is used for error checking at the server. The client routine in Fig. 4(a) shows the process that takes place once a record of data is read from the UDP layer. Blocks of 259 bytes are sampled from such a record. If the type field indicates a repair word, and if further CRC indicates correct reception, the first bytes are stripped and the following 217 bytes are sent to the receiving buffer. The DL field may be used to separate real data from padding. If the type field indicates an interleaved word, this interleaved word is sent to the de-interleaving buffer. A timer is reset upon the arrival of the first byte of a certain deinterleaved RS word (this may not be the first byte of the word).

Routinely, the de-interleaved RS words are checked one after another. Let E be a suitably selected number in the range 1 to $(N - K)$, where $N = 255$, $K = 223$. $TH1$ is a period of time corresponding to a number of bits in the range $((K + E) * 256 * 8)$ to $(N * 256 * 8)$, while $TH2$ corresponds to the range $(N * 256 * 8)$ to $((N + E) * 256 * 8)$. If, in a certain RS word, the first $K = 223$ bytes have been received before the expiry of a certain time out $TH1$ on the timer above, and CRC indicates correct reception, the first 4 bytes are stripped and the following 217 data bytes are delivered to the receiver buffer. If CRC checking fails, erasure-based RS decoding is tried, and later CRC checking is performed again. If CRC works in this second trial after RS decoding, the 217 bytes are delivered to the receiver buffer. If CRC fails for the second time in a row, the client waits for more bytes before trying RS and CRC decoding again.

If $(K + E)$ bytes out of $N = 255$ bytes are received (not necessarily the first K bytes) before the expiry of time out $TH2$, erasure-base RS decoding is performed first, after

which CRC checking is verified and 217-byte data delivery to the receiver buffer takes place. If CRC checking fails in the latter case, or time out TH2 is exceeded without receiving $(K + E)$ bytes, then a NAK (repair request) is formulated and sent to the server. Because of interleaving, one has to pay for the loss of resilience by waiting a longer time before a meaningful number of bytes (at least $(K + E)$) of the same RS word are received. Only two NAKs or repair requests are allowed for the same RS word. The processing of the received repairs will be similar to the above, with the exception that no RS decoding takes place. Finally, as to video or voice applications, if all error correction trials for a certain RS word fail and playing time for this word expires, the previous correctly received 217 bytes in the receiver's buffer should be replayed. It is also possible that too many data arrive at the client. In this case, due to being busy with de-interleaving and RS decoding, the client may suffer from data loss.

Chen at NOK00075560-561, page 2, col. 2 ¶¶ 2-3-page 3, col. 1 ¶ 1.

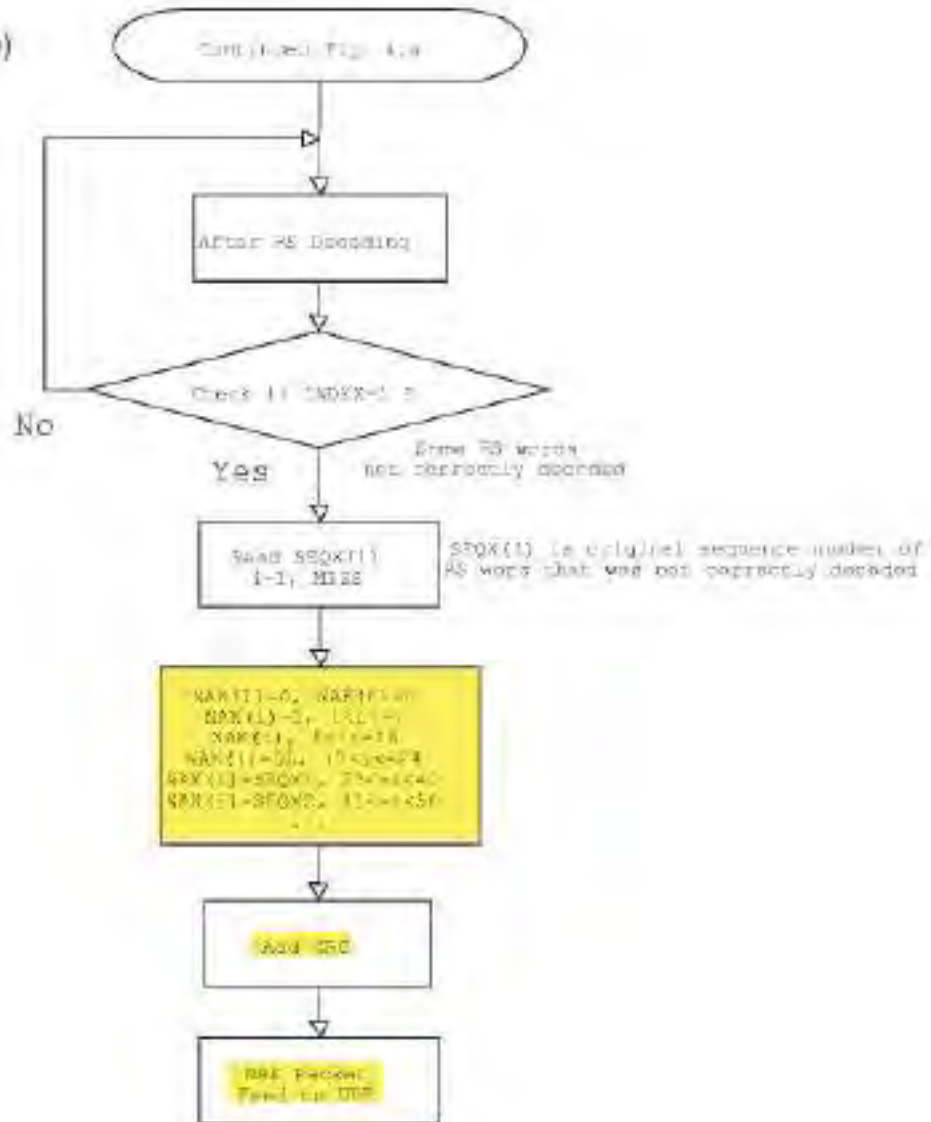
595. Chen further describes that, if a NAK message is required, it is generated with a sequence number of each incorrectly received packet.

Fig. 4(b) shows the step-by-step generation of the NAK request at the client. As shown in Fig. 2(b), we execute the processing for such a NAK message as it arrives at the server. Fig. 4(b) also shows the multiplexing of RS (interleaved) words and repair (retransmission) words in server transmission time. Priority is given to repair words, and interleaved words are transmitted when there are no repairs to be sent ($XZ = 0$).

The timing diagram of Fig. 5 shows a sequence of events taking place under the control of our new scheme for a typical scenario. Equal-length interleaving windows for data records N , $N + 1$, $N + 2$ are used. The end of the next window sets the time limit for playing received packets of the current record at the MPEG player. Since received packets of each record are reorganized in the interleaving window and then played in sequence, this technique has a buffer effect that can help to guarantee delay-jitter-free playing at the receiver. This figure also shows the NAK and retransmissions for record N due to the packet loss. For record $N + 1$, the RS FEC decoder can correct the loss and errors without asking for any retransmission. For record $N + 2$, no loss and errors are encountered for the first K bytes of RS symbols; therefore the RS decoder is not called, and no NAKs are generated.

Chen at NOK00075561, page 3, col. 2 ¶¶ 1-2. *See also id.* Figs. 4b and 5.

(b)



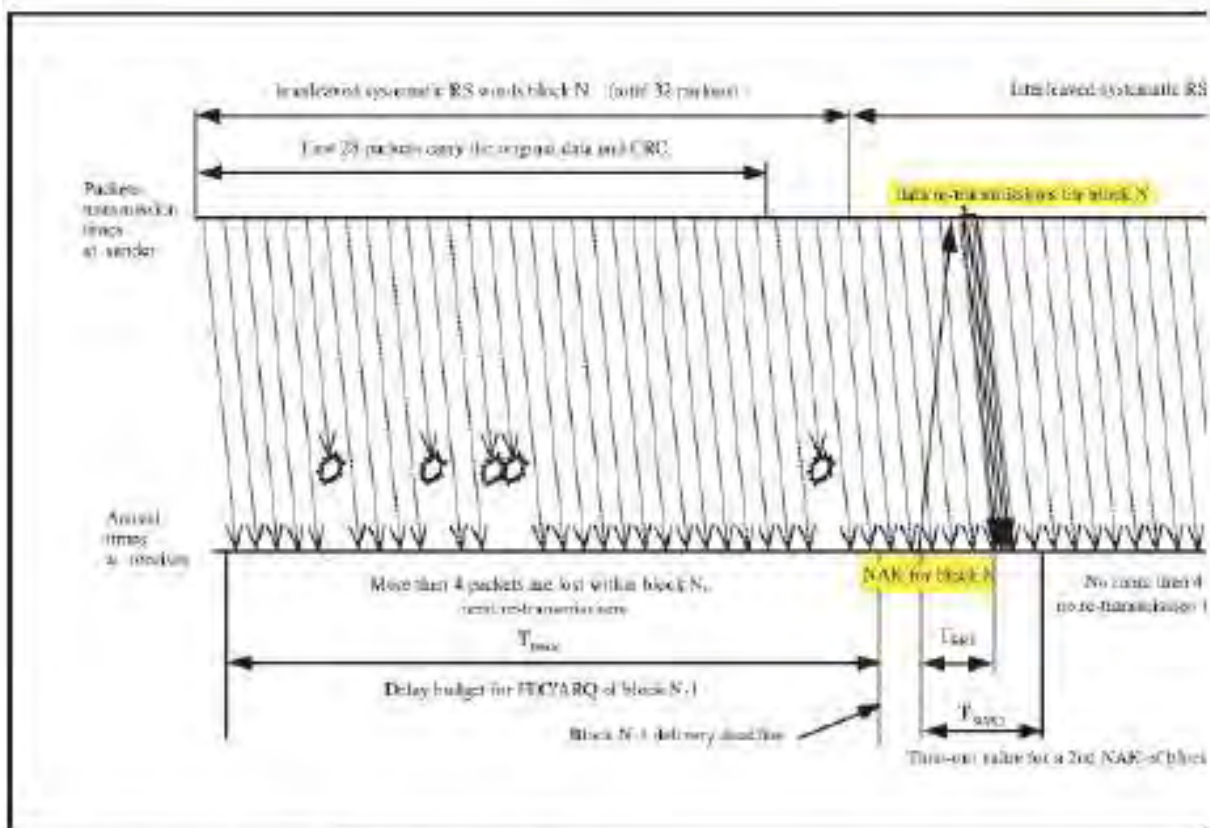


Figure 5: The hybrid FEC/ARQ system timing.

596. The NAK message(s) that indicate incorrectly received packets are then received by the transmitter. The data that needs to be retransmitted is identified so only the NAK data is re-sent.

Fig. 2(a) shows the byte processing of a transmitted multicast file, as outlined above. Meanwhile, Fig. 2(b) underlines the processing of retransmission packets and shows how to combine them with new interleaved words. Most of the processing above is done offline by multicast applications. The whole video or voice file is FEC-encoded, interleaved and stored before the real-time multicast is started. The procedure of mixing repairs and interleaving words has to be executed in real time when the multicast session commences.

Chen at NOK00075560, page 2, col. 1 ¶ 4-col. 1 ¶ 1. See also *id.* Fig. 2(b):

corresponding word is preceded by a type field. So, far, we have only two designations, where 11111111 and 10101010 denote a fresh and a retransmitted word respectively. A 1-byte data length (DL) field is also needed to indicate the end of receiving because the data may be less than 217 bytes. If the data are less than 217 bytes, the remainder of the 217 bytes is filled with padding. A 2-byte cyclic redundancy check (CRC) field is also formed at the end of the word. This CRC operates over the whole word. The resulting 223 bytes are fed to the (255, 223) Reed-Solomon encoder routine [1], which outputs a corresponding 255 bytes that we call an encoded word. The RS code is assumed to be systematic, such that the 32 parity bytes are added by the encoder at the end of the word. One byte of dummy data is added to obtain a 256-byte encoded word.

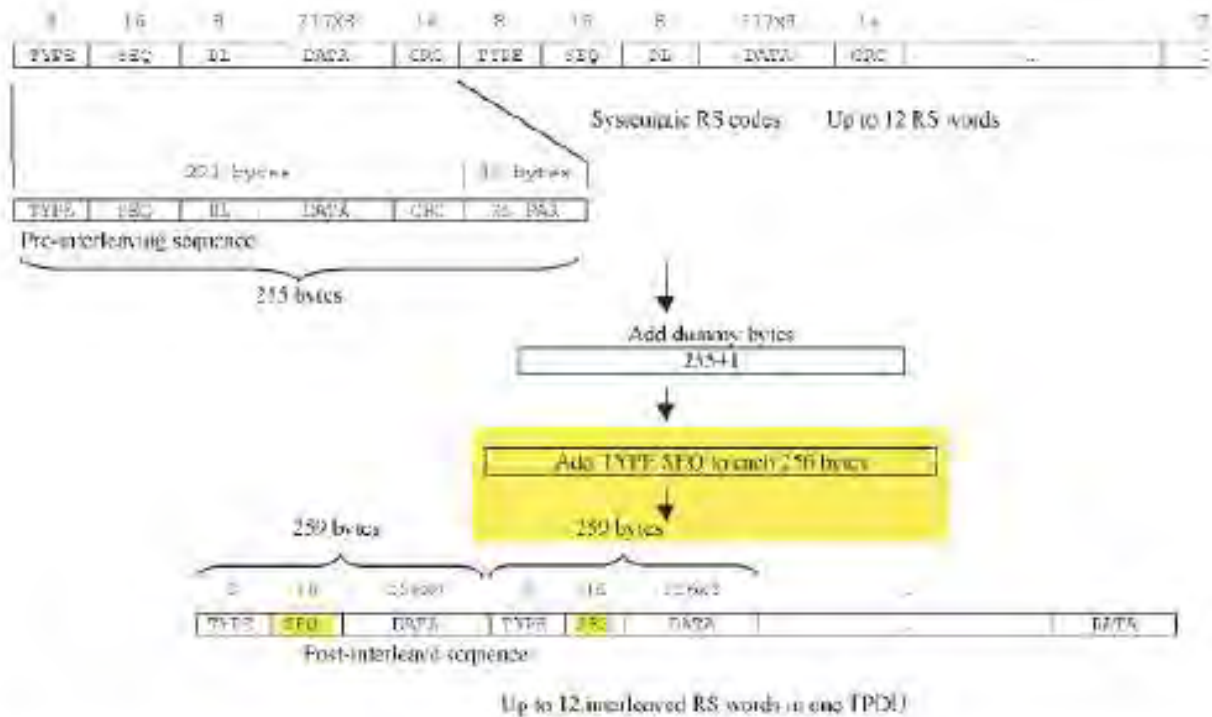
Chen at NOK00075559, page 2, col. 2 ¶ 4 - NOK00075560, page 2, col. 1 ¶ 1.

The format of the NAK frame used by the processing in Fig. 2(b) is shown in Fig. 3. As mentioned before, for the multicast applications at hand, this is the only packet type that the client can send. However, for a video conferencing system, data may also be sent from the client to the server. In Fig. 3, we use 01010101 as the value of the type field to denote NAK messages. A 1-byte field denoting the data length of the NAK message is also added. The subsequent fields give the sequence numbers of the lost words as requested by the client. A final 1-byte CRC field is used for error checking at the server. The client routine in Fig. 4(a) shows the process that takes place once a record of data is read from the UDP layer. Blocks of 259 bytes are sampled from such a record. If the type field indicates a repair word, and if further CRC indicates correct reception, the first bytes are stripped and the following 217 bytes are sent to the receiving buffer. The DL field may be used to separate real data from padding. If the type field indicates an interleaved word, this interleaved word is sent to the de-interleaving buffer. A timer is reset upon the arrival of the first byte of a certain deinterleaved RS word (this may not be the first byte of the word).

Chen at NOK00075560, page 2, col. 2 ¶ 2. *See also id.* Figs. 1, 3 (Fig. 1 excerpted to show sequence numbers that correspond to the data following that number)

RS(255, 223) concatenated, $255-223=32$ erasures

$256 \times 256 \times 8 = 524288$ bits



protocol data unit (TPDU) format.

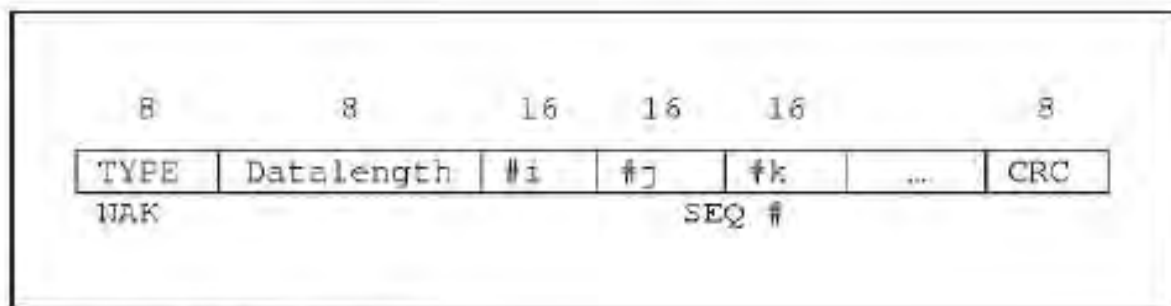


Figure 3: NAK packet format.

598. Chen therefore discloses a “wherein at least one message of the plurality of messages includes an acknowledgement (ACK) or a negative acknowledgement (NACK) of the transmitted packet” (Chen sends negative acknowledgement messages including sequence numbers for packets not correctly received).

iii Yap in view of VDSL1

599. Yap discloses “wherein at least one message of the plurality of messages includes an acknowledgement (ACK) or a negative acknowledgement (NACK) of the transmitted packet.”

600. As described above, after CRC error detection is performed at the receiver, the receiver then transmits either an acknowledgement or negative acknowledgement (ACK/NACK) to the transmitter to indicate whether retransmission of the packet is required. “If it is determined that the checked packet does not include error, acknowledgment (ACK) is transmitted to the transmitter and the transmitter does not retransmit the packet. If there is an error, a negative acknowledgment (NACK) is transmitted to transmitter 100 for retransmission request.” Yap at NOK00084270, 4:49-53.

D. The Asserted Claims of the '809 Patent Are Obvious

601. The asserted claims of the '809 Patent are claims 4 and 6. The following charts summarize my citations to the prior art for the asserted claims:

Ground	Claims	Summary Claim Chart
VDSL	4, 6	Exhibit C-1
Chen in view of ADSL 2	4, 6	Exhibit C-2
Yap in view of VDSL1	4, 6	Exhibit C-3

602. The asserted claims of the '809 Patent are obvious. I address each limitation of the asserted claims in turn below.

1. Claim 4

[4.pre] The apparatus of claim 1,

603. Claim 4 depends from claim 1. As shown in the table below, the limitations of claim 1 of the 809 Patent are similar to the limitations claim 1 of the 348 Patent. For the reasons set forth above with respect to claim 1 of the 348 Patent, VDSL1, Chen in view of ADSL2, and Yap in view of VDSL1 disclose the limitations of claim 1 of the 809 Patent.

Limitation	348 Patent	809 Patent
1.pre	An apparatus comprising:	An apparatus comprising:
1.a	a multicarrier transceiver including a processor and memory operable to:	a multicarrier transceiver including a processor and memory capable of:
1.b	transmit a packet using a forward error correction encoder and an interleaver,	transmitting a packet using forward error correction encoding and interleaving,
1.c	wherein the packet comprises a header field and a plurality of PTM-TC codewords, a plurality of ATM cells or a plurality of Reed-Solomon codewords,	wherein the packet comprises a header field and a plurality of Reed-Solomon codewords,
1.d	and wherein the header field comprises a sequence identifier (SID);	and wherein the header field comprises a sequence identifier (SID);
1.e	and receive a plurality of messages using a forward error correction decoder and without using a deinterleaver,	and receiving a message using forward error correction decoding and without using deinterleaving,
1.f	wherein each message of the plurality of messages is received in a different DMT symbol	wherein the message is received in a single DMT symbol,
1.g	and wherein at least one message of the plurality of messages includes an acknowledgement (ACK) or a negative acknowledgement (NACK) of the transmitted packet.	and wherein the message includes an acknowledgement (ACK) or a negative acknowledgement (NACK) of the transmitted packet.

[4.a] wherein a physical layer of the transceiver is capable of generating the packet and the message.

i VDSL1

604. VDSL1 discloses “wherein a physical layer of the transceiver is capable of generating the packet and the message.” For example, VDSL provides for a physical layer that “shall be able to transport at least one of ATM or PTM signals.” VDSL1 at 4, 11; *see also id.* 30 (describing OAM functional model of a VDSL link), 139 (showing transport of ATM data), 149-50 (describing PMS-TC sublayer functional model for VTU-O and VTU-R), 186 (detailing link activation/deactivation process to establish/terminate a VDSL link with physically connected and

powered VTU-O and VTU-R), 210-12 (detailing UTOPIA implementation of the ATM-TC interface).

ii Chen in view of ADSL2

605. Chen discloses “wherein a physical layer of the transceiver is capable of generating the packet and the message.” As discussed above, Chen discloses an FEC/ARQ error transmission scheme. Chen at page 159, col. 1 ¶ 1-col. 2 ¶ 2. Chen describes how to apply this method to a unicast system. *Id.* NOK00075562, col. 2 ¶ 2. Additionally, the transportation-layer protocol data unit format is shown in Figure 1:

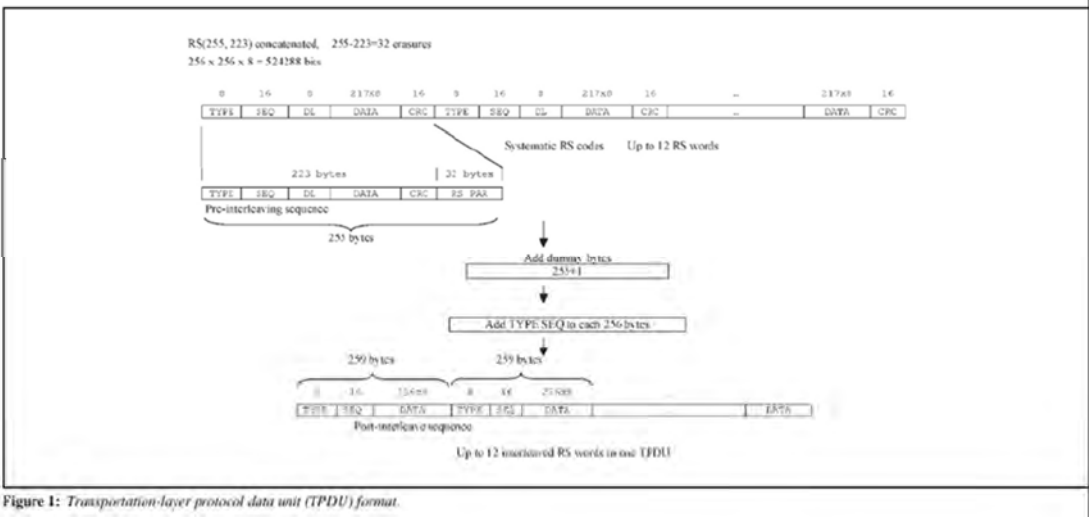


Figure 1: Transportation-layer protocol data unit (TPDU) format.

606. In addition, ADSL2 discloses “wherein a physical layer of the transceiver is capable of generating the packet and the message.” For example, ADSL2 describes ADSL transceivers that allow high-speed data transmission between the network operator end (ATU-C) and the customer end (ATU-R), and it also specifies the physical layer characteristics of the ADSL interface. ADSL2 at page i, Summary. In Section 5.1, ADSL2 describes the principal functions of the PMD and PMS-TC layers, which “may generate or receive control messages via the overhead channel of the PMS-TX layer” and “transport control messages generated in the TPS-TC, PMS-TX or PMD layers as well as messages generated at the management interface,” respectively. ADSL2 at pages 9, 5.1 (also

describing the TPS-TC layer and MPS-TC function). Additionally, the functional blocks and interfaces of an ATU-C and ATU-R are shown in Figure 5-1:

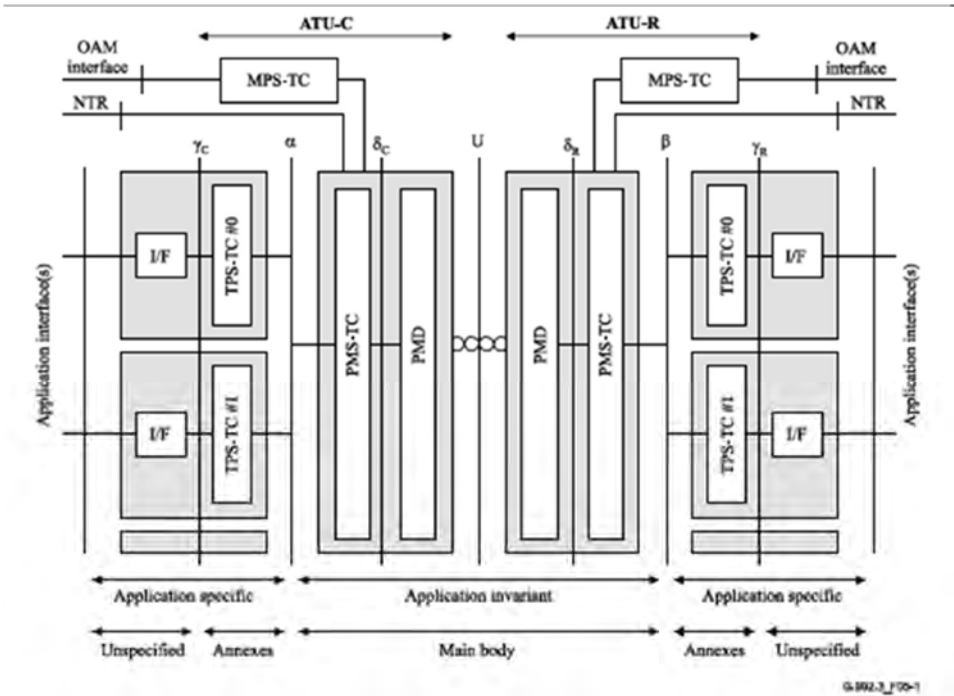
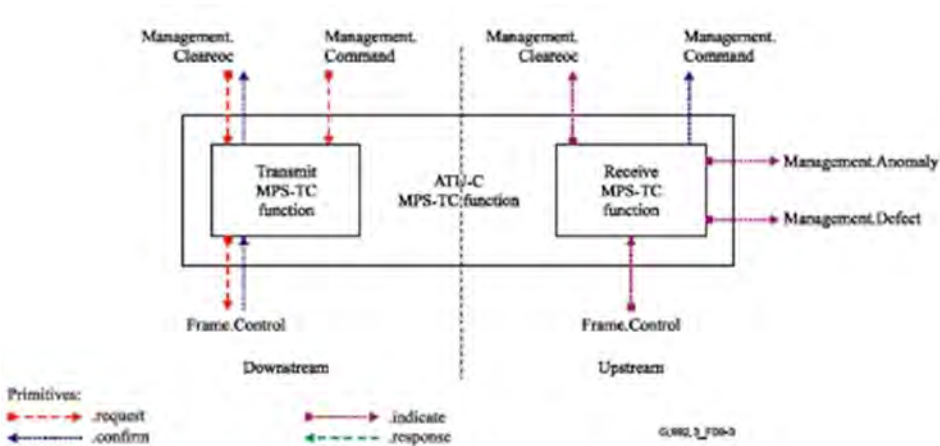


Figure 5-1/G.992.3 – ATU functional model

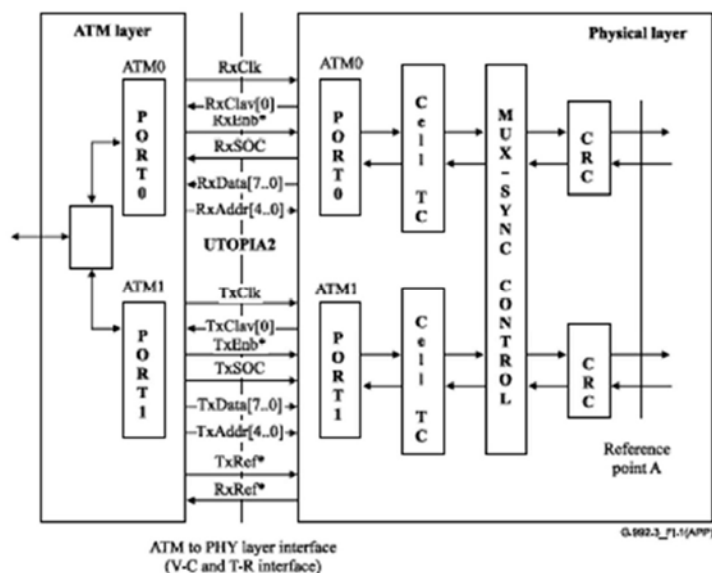
607. The MPS-TC function, also described in Section 5.1, has interface signals, which are shown in Figure 9-3 (ATU-C) and Figure 9-4 (ATU-R).



ADSL2, pages 171, 9.3.

608. Further, Annex F of ADSL2 describes performance requirements for operation, and also references testing of physical layer aspects with the defined ATU control parameters. ADSL2, page 329, Annex F.

609. Finally, Appendix I describes the logical interface between the ATM layer and the physical layer (i.e., ATU) at ATU-C and ATU-R. ADLS2, page 408, Appendix I; *see also id.* Figure I.1.



610. Because both Chen and ADSL2 disclose “wherein a physical layer of the transceiver is capable of generating the packet and the message,” this limitation is rendered obvious by the combination of Chen and ADSL2.

iii Yap in view of VDSL1

611. Yap discloses “wherein a physical layer of the transceiver is capable of generating the packet and the message.” As described above, after CRC error detection is performed at the receiver, the receiver then transmits either an acknowledgement or negative acknowledgement (ACK/NACK) to the transmitter to indicate whether retransmission of the packet is required. “If it is determined that the checked packet does not include error, acknowledgment (ACK) is transmitted to the transmitter and the transmitter does not retransmit the packet. If there is an error, a negative

acknowledgment (NACK) is transmitted to transmitter 100 for retransmission request.” Yap at NOK00084270, 4:49-53; *see also* Yap at 2:44-49, 3:63-4:7, 4:8-20, 4:32-53, 4:57-5:4, 5:41-48, Figs. 1-7.

2. Claim 6

[6.pre] The apparatus of claim 2,

612. Claim 2 recites: [2pre] the apparatus of claim 1, [2.a] wherein the received message has a higher immunity to noise than the transmitted packet.

613. Limitation [2pre] is identical to limitation [4.pre] and is therefore disclosed or rendered obvious for the same reasons set forth with respect to that limitation.

614. As described below, the identified prior art references disclose [2.a].

i VDSL1

615. VDSL1 discloses “wherein the received message has a higher immunity to noise than the transmitted packet.” In particular, VDSL1-compliant systems “shall provide protection against disturbance from impulse noise,” in at least two levels of protection: first, a lower level of protection to support latency-sensitive services like voice, and second, a higher level of protection to support burst error-sensitive services like video. VDSL1 at 53-54. The standard provides an impulse noise model (in which an impulse noise generator is required to prove the burst noise immunity of the VDSL transceiver with repetitive bursts of white Gaussian noise injected onto the line with sufficient power to ensure effective erasure of the data for the period of time of the burst), and requires transmission performance testing to measure noise margins and verify that minimum requirements are satisfied. VDSL1 at 93-96.

ii Chen in view of ADSL2

616. Chen discloses “wherein the received message has a higher immunity to noise than the transmitted packet.” As has already been explained, Chen describes using a combination of

interleaving and ARQ coding to enhance the quality of multicast and unicast transmissions, which enables the received message to have a higher immunity to noise than the transmitted packet. Chen at Abstract.

617. In addition, ADSL2 discloses “wherein the received message has a higher immunity to noise than the transmitted packet.” ADSL2 describes interleaving “[t]o spread the Reed-Solomon codeword and, therefore, reduce the probability of failure of the FEC in the presence of impulse noise.” ADSL2 at 38, 7.7.1.5. ADSL2 further provides that the receiver should select values based on enumerated constraints and optimization priorities, including related to impulse noise protection. ADSL2 at 54, 7.10.3. ADSL2 specifies the signal-to-noise plus distortion ratio of the transmitted signal, and measures that ratio using a MultiTone Power Ratio (MTPR) test. ADSL2 at 93, 8.9.2. Moreover, the signal-to-noise-ratio margin (defined as the maximum increase (in dB) of the received noise power, such that the ATU can still meet all the target BERs over all the frame bearers) is measured by the receive PMD function during initialization and diagnostics mode. ADSL2 at 102, 8.12.3.6.

618. Because both Chen and ADSL2 disclose “wherein the received message has a higher immunity to noise than the transmitted packet,” this limitation is rendered obvious by the combination of Chen and ADSL2.

iii Yap in view of VDSL1

619. Yap discloses “wherein the received message has a higher immunity to noise than the transmitted packet.” An automatic retransmission request control system is described in Yap, which seeks to ensure that the received message has a higher immunity to noise than the transmitted packet. In particular, Yap discloses a retransmission mode selection made from among “(a) a mode in which to transmit the data, which are to be retransmitted, via the same antenna as in the previous transmission, while transmitting, at the same time, new data by use of an antenna via which no data

retransmission is requested; (b) a mode in which to transmit the data, which are to be retransmitted, via an antenna via which no retransmission is requested, while transmitting new data via another antenna at the same time; (c) a mode in which to use STBC to retransmit the data via an antenna via which no retransmission is requested; and (d) a mode in which to use STBC to retransmit the data via all the available antennas.” Yap at Abstract. The transmitter is shown as Figure 1, and the receiver as Figure 2. Yap at Fig. 1, Fig. 2, 3:63-4:7.

[6.a] wherein a physical layer of the transceiver is capable of generating the packet and the message.

620. This limitation is identical to limitation [4.a] and is therefore disclosed or rendered obvious for the same reasons set forth with respect to that limitation.

XII. SECONDARY CONSIDERATIONS OF NON-OBVIOUSNESS

621. I have been informed that, under certain circumstances, a showing of obviousness can be rebutted based on one or more secondary considerations of non-obviousness that may indicate that the patented invention was not obvious to a person of ordinary skill in the art at the time of the invention.

622. At this time, I am not aware of any secondary considerations of non-obviousness that would be relevant to my analysis of the validity of the asserted claims. If TQ Delta or its experts should present any other evidence relating to secondary considerations of non-obviousness, either through its experts or otherwise, I reserve the right to respond and/or to supplement my report.

623. I have reviewed TQ Delta’s response to Nokia’s Interrogatory No. 6, in which TQ Delta responded to a question seeking the factual basis for any contention by TQ Delta that there exists any secondary consideration or objective evidence of non-obviousness with respect to the Patents-in-Suit. After reviewing TQ Delta’s response, I have not seen any evidence that shows non-obviousness of the asserted claims.

624. At this time, I am not aware of any secondary considerations of non-obviousness that would be relevant to my analysis of the validity of the asserted claims. If TQ Delta or its experts should present any other evidence relating to secondary considerations of non-obviousness, either through its experts or otherwise, I reserve the right to respond and/or to supplement my report.

A. Commercial Success

625. In its interrogatory response, TQ Delta states that the Patents-in-Suit have been commercially successful. The only evidence of that TQ Delta points to is the sales of Nokia and other companies that TQ Delta has sued. TQ Delta has not shown that these products practice the Family 9 Patents. Even if TQ Delta had shown that these products practice the Family 9 Patents (which it has not), TQ Delta has not shown the sales of those products are due to the functionality it alleges infringes the Family 9 Patents. In sum, I see no evidence that the Family 9 inventions have been commercially successful.

B. Licensing

626. TQ Delta also contends that licensing is evidence of commercial success, failure of others, long felt but unresolved needs, and recognition of a problem. I understand that the licensing evidence must be tied to the asserted patents. In its interrogatory response, TQ Delta identifies various Lantiq licenses with Aware. I understand, however, that TQ Delta filed a summary judgment motion in prior litigation arguing that these licenses do not include the Family 9 Patents. Therefore, according to TQ Delta, the Lantiq licenses have no relationship to the Family 9 Patents and cannot be evidence that the inventions in the Family 9 Patents were licensed. With regard to the Zhone, Fujitsu, Siemens, ZyXEL, and DZS licenses, since TQ Delta acquired the Family 9 Patents in 2012, I understand that TQ Delta has only received three royalty reports from Siemens totaling less than \$6,000, covering the time period of December 2016 through December 2021. *See* A. Divine Depo. Tr. at 138:5-18. TQ Delta also stated that the Siemen's product developed under

the license failed to sell in the marketplace. *See id.* Further, the licenses that TQ Delta cites are portfolio-based, and TQ Delta has not shown that any royalty received from Siemens relates to the use of the Family 9 Patents alone. TQ Delta has not received any other royalty reports from any of the other licenses.

627. Additionally, TQ Delta has not demonstrated any nexus between the commercial success of the licensed products as a result of the incorporation of the claimed technology.

C. Standardization

628. TQ Delta also claims that the ITU-T's adoption of DSL standards "that have incorporated the inventions of many of the Asserted Patents into them as indicated by TQ Delta's infringement claim charts which identify the specific portions of the standards on which the patent claims read" is evidence of industry praise. I have reviewed the Family 9 Patents and TQ Delta's infringement contentions, but note TQ Delta has failed to map any Accused Product to any ITU-T standard. Moreover, TQ Delta has not shown that the success of any such product is attributable solely to the mandatory features of the DSL standards.

629. I see no evidence that the identified licenses evidence commercial success, failure of others, long felt but unresolved needs, recognition of a problem, or industry praise with respect to the Family 9 Patents.

XIII. CONCLUSION

630. In my opinion, claims 10 and 18 of the '411 Patent, claim 1 of the '348 Patent, and claims 4 and 6 of the '809 Patent are invalid in light of the prior art.

631. I reserve the right to supplement my opinions in the future to respond to any arguments or positions that TQ Delta or its experts may raise, taking account of new information as it becomes available to me.

632. If I am called to testify about this report by deposition or at a hearing before the Court, I may cite other documents or information similar to that specifically identified in this report. I may also use graphics, animations, pictures, demonstrations, and/or other audio/visual aids to explain my analysis and opinions.

I declare under penalty of perjury of the law of the United States that all statements made herein of my own knowledge are true, and that any statements made on information and belief are believed to be true.

Date: August 29, 2022



Richard D. Wesel, Ph.D.

APPENDIX A

RICHARD DALE WESEL

ELECTRICAL AND COMPUTER ENGINEERING DEPARTMENT
AND
OFFICE OF ACADEMIC AND STUDENT AFFAIRS

HENRY SAMUELI SCHOOL OF ENGINEERING
UNIVERSITY OF CALIFORNIA, LOS ANGELES

Room 6426 Boelter Hall • Box 951594 • Los Angeles, CA 90095-1594
Phone (310) 267-2150 • Cell (310) 922-7831 • Email: wesel@ucla.edu

EDUCATION

1991 – 1996, **Stanford University**, Stanford, CA

Ph.D. in Electrical Engineering

Trellis Code Design for Correlated Fading and Achievable Rates for Tomlinson-Harashima Precoding.

1984 – 1989, **Massachusetts Institute of Technology**, Cambridge, MA

S. M. and S. B. in Electrical Engineering

Thesis: *Adaptive Equalization for Modem Constellation Identification.*

EMPLOYMENT

1996 – present, **University of California, Los Angeles**, Los Angeles, CA

- **Associate Dean** of Academic and Student Affairs for the Henry Samueli School of Engineering and Applied Science since July 2007
- **Acting Director** HSSEAS MS Online Program, July 2007 – July 2008 (inaugural year of operation admitting students and offering classes)
- **Professor** of Electrical and Computer Engineering since July 2006
- **Associate Professor** of Electrical Engineering 2002-2006
- **Assistant Professor** of Electrical Engineering 1996-2001

1991 – 1996, **Stanford University**, Stanford, CA, Research Assistant and Teaching Assistant.

1986 – 1994, **AT&T Bell Laboratories**, Holmdel, NJ

- Member of Technical Staff summer 1994
- Member of Technical Staff 1989-1991
- Intern 1986-1989, two summer internships and a culminating 6-month internship.

AWARDS

- Fellow of the Institute of Electrical and Electronics Engineers (IEEE)
- Qualcomm Faculty Award
- Selected for the National Academy of Engineering Frontiers of Engineering Program
- TRW Excellence in Teaching Award (UCLA School of Engineering)
- Okawa Foundation Award for Excellence in Telecommunications Research
- National Science Foundation CAREER Award
- AT&T Foundation Ph.D. Fellow.
- Tau Beta Pi MIT chapter president 1987-1988, Eta Kappa Nu, Sigma Xi, National Merit Scholar.

ACADEMIC SERVICE

- **Associate Dean, Leading Office of Academic and Student Affairs**, including admissions and counseling for all engineering majors, July 2007 to present
- **Chair, Samueli COVID-19 Task Force** 2020-2021
- **Member, UCLA COVID-19 Academic Continuity Task Force** 2020-2021
- **Member, HSSEAS SEASnet review committee**, November 2018 to 2020
- **Chair, Summer Sessions Faculty Advisory Committee**, Sept. 2018 to present.
- **Member, Advisory Committee on Immigration Policy**, 2017 to 2021
- **HSSEAS Strategic Planning Committee for Education**, Jan. 2017-June 2018
- **Member, Center for the Integration of Research, Teaching, and Learning at UCLA, Steering Committee**, January 2017-present
- **Member, Classroom Advisory Committee**, November 2016 to present
- **Chair, UCLA Special Programs Task Force**, Feb. 2013- Sept. 2014
- **Member, UCLA Enrollment Planning Committee** October 2011-present
- **Member, UCLA Undergraduate Non-Resident Implementation Task Force** August 2010 – July 2011
- **Member of UCLA Undergraduate Council** July 2006- July 2008.
- **Member of the Committee on Undergraduate Admissions and Relations with Schools** July 2006- July 2008
- **Electrical Engineering Department Vice Chair for Undergraduate Affairs** July 2005 – July 2007. Successfully managed the 2006 ABET Accreditation visit for EE.
- **Member of the School of Engineering Faculty Executive Committee** 2003-2006.

- **Chair of the Electrical Engineering Department Courses and Curriculum Committee** 2003-2005.
- **Chair of the Communications Major Field** in the Electrical Engineering Department at the University of California, Los Angeles, 1999-2004.
- **Chair of the Cubicle Allocation Committee** for the Electrical Engineering Department at UCLA, managing the allocation of 150 student cubicles among approximately 20 professors who share this space, 1998-2005.
- **Chair of 2002 Annual Research Review** (annual departmental research symposium). Also Vice Chair of 2001 Annual Research Review.
- **Member of 2001 UCLA EE Annual Report Committee.**
- **Elected Member of the Legislative Assembly** of the UCLA Academic Senate, 1997-2001.
- **Chair for quarterly Seminar Series** in Signals and Systems. Established this seminar series in spring 1997. Recruit a professor each quarter to organize speakers for the series. Personally organized speakers for four of these quarters.
- **Local Exhibits Chair**, 1997 UCLA EE Research Symposium

GRADUATED PH.D. STUDENTS

1. Christina Fragouli, Ph.D. Sept. 2000, Dissertation: *Turbo Code Design for High Spectral Efficiency*, 2000-2001 UCLA EE Dept. Best Ph.D. Student 2001. **Professor at UCLA.**
2. Christos Komninakis, Ph.D. Dec. 2000, Dissertation: *Joint Channel Estimation and Decoding for Wireless Channels*, Senior Director of Technology, Qualcomm.
3. Xueting Liu, Ph.D. Dec. 2000, Dissertation: *Trellis Code Design for Periodic Erasures and Adaptive Coded, Modulation Schemes for Time-Varying Channels*, Apple Inc., CA
4. Wei Shi, Ph.D. Dec. 2000, Dissertation: *New Results in Wireless Communications*, Qualcomm
5. Tom Sun, Ph.D. Dec. 2002, Dissertation: *Error Protection Techniques for Source and Channel Coding*, Qualcomm, San Diego, CA
6. Chris Jones, Ph.D. Dec. 2003, Dissertation: *Constructions, applications, and implementations of low-density parity-check codes*, Co-founder Chilicon Power, Los Angeles, CA
7. Adina Matache, Ph. D. June 2004, Dissertation: *Coding Techniques for High Data Rates in Wireless Multiple-Input Multiple-Output Communications*, Aerospace Corp.
8. Cenk Kose, Ph.D. Dec. 2004, Dissertation: *Universal trellis codes and concatenated trellis-coded modulations for the compound linear vector Gaussian channel*
9. Aditya Ramamoorthy, June 2005, *Generalized ACE Codes and Theoretic Results in Network Coding*, **Associate Professor at Iowa State University**
10. Jun Shi, Ph.D. Sept. 2005, Dissertation: *Universal Channel Codes and Trellis State-Diagram Reduction*, Senior Principle Scientist at the Broadcom Corporation
11. Wen-Yen Weng, Ph.D. March 2007, Dissertation: *Universal Serially Concatenated Trellis Coded Modulations and Rate-Compatible High-Rate LDPC Codes*

12. Esteban Valles (Primary Advisor John Villasenor), Ph.D. March 2007, Dissertation: *Timing Recovery Using Soft Information Feedback and Efficiency of Array Codes*, Aerospace Corporation.
13. Andres Vila Casado, Ph.D. December 2007, Dissertation: *Improving LDPC Decoders: Informed Dynamic Message-Passing Scheduling and Multiple-Rate Code Design*
14. Herwin Chan (Primary Advisor Ingrid Verbauwhede), Ph.D. December 2007, Dissertation: *Accelerating Applications Through Cross-Layer Co-Design*
15. Miguel Griot, Ph.D. Sept. 2008, Dissertation: *Nonlinear Codes for Multiple Access to Binary Channels and Higher-Order Modulations over the AWGN Channel*, Qualcomm Corporation.
16. Bike Xie, Ph.D. June 2010, Dissertation: *Encoding for Degraded Broadcast Channels and Resource Allocation for content Distribution in Peer-To-Peer Networks*, Senior Staff Engineer-Manager at Marvell Corp
17. Thomas Courtade, Ph.D. June 2012, Dissertation: *Two Problems in Multiterminal Information Theory*, **Assistant Professor at UC Berkeley.**
18. Jiadong Wang, Ph.D. June 2012, Dissertation: *Absorbing Set Analysis of LDPC Codes and Read-Channel Quantization in Flash Memory*, Senior Engineer at Qualcomm.
19. Tsung-Yi Chen, Ph. D. September 2013, Dissertation: *Achieving Low-Latency Communication with Feedback: from Information Theory to Practical System Design*, SpiderCloud Wireless.
20. Adam Williamson, Ph. D June 2014, Dissertation: *Reliability-output Decoding and Low-latency Variable-Length Coding Schemes for Communication with Feedback*, Northrop Grumman Corp.
21. Kasra Vakiliinia, Ph. D June 2014, Dissertation: *Coding Schemes to Approach Capacity in Short Blocklength with Feedback and LDPC Coding for Flash Memory*, Huawei Technologies.
22. Haobo Wang, Ph. D Decembner, 2018, Dissertation: *Optimizing Flash-Based Storage Systems*, SK Hynix.
23. Sudarsan V.S. Ranganathan, Ph. D December, 2018, Dissertation: *Advances in Protograph-Based LDPC Codes and a Rate Allocation Problem*, Post Doctoral Scholar and Lecturer at the Massachusetts Institute of Technology.

RESEARCH FUNDING SOURCES 1996 - PRESENT

- Mercury Systems
- Qualcomm
- Zeta Corporation
- Physical Optics Corporation
- SA Photonics
- Micron Semiconductor
- National Science Foundation
- Western Digital Corporation
- INPHI Corporation
- Rockwell Collins
- The Broadcom Foundation

- Boeing
- Texas Instruments
- Conexant
- ST Microelectronics
- Northrop Grumman
- Skyworks
- Xetron Corporation
- Pacific Bell
- Honeywell

COURSES TAUGHT 1996 - PRESENT

- EE131A Probability
- EE132A Communications Systems
- EE231A Information Theory
- EE231E Channel Coding
- EE232A Stochastic Processes

PROFESSIONAL ACTIVITIES

- **General Chair**, 2022 North American School of Information Theory, Los Angeles, CA, August 15-19, 2022.
- **Associate Editor**, IEEE Transactions on Information Theory, March 2020-present
- **Presenter** of half-day tutorial on Incremental Redundancy at Globecom 2019, December 2019.
- **Panel member** for National Science Foundation Proposal Review Panels.
- **Invited Speaker** half-day tutorial on Incremental Redundancy at the 2018 European School of Information Theory, Bertinoro, Italy, May 2018
- **Invited Speaker** at IEEE UCSD ITA Workshop, annually 2006-present
- **Reviewer** for various IEEE conferences and journals. Regularly reviewing submissions to Trans. on Information Theory, Trans. on Communications, Journal on Selected Areas of Communications, Communications Letters, Globecom, and International Conference on Communications, 1994-present.
- **Technical Program Committee Member**, numerous times for *Globecom*, *ICC*, and *ISIT*.

- **Organizer and lecturer** for UCLA Extension course on Error Control Coding (2000-2006). Received an award for being among the top 10% of UCLA extension lecturers.
- **Associate Editor**, *IEEE Transactions on Communications* 1999-2005.
- **Technical Program Chair**, *Communication Theory Symposium at Globecom 2002*.
- **Organizer and Session Chair** for Special Session on Concatenated codes and iterative decoding at the *2001 Asilomar Conf. on Signals, Systems, and Computers*.
- **Session Organizer and Chair** for Communication Theory Symposium at the 2001 International Conference on Communications.
- **Organizer and Session Chair** for Special Session on Communication over Time Varying Channels at the *1999 Asilomar Conf. on Signals, Systems, and Computers*.
- **Invited speaker** 1998 and 2000 *IEEE Communication Theory Workshops*.
- **Invited speaker** Office of Naval Research, Naval Research Labs 1998 Turbo Codes Workshop.
- **Invited speaker** 1998 DARPA GloMo workshop on emerging technologies for hand-held wireless devices in military communication.
- **Instructor** for 1997 UCLA Extension course on wireless multimedia communications.
- **Invited speaker** at various universities and companies including Stanford, UC-Berkeley, UC-San Diego, Ohio State University, University of Arizona, Johns Hopkins University, Cornell, Telia Research, Lulea, Sweden, Lucent, Boeing, Xetron, Texas Instruments, Conexant, and Microsoft Research.

JOURNAL PUBLICATIONS (see all publications at <http://www.seas.ucla.edu/csl>)

1. A. Baldauf, A. Belhouchat, S. Kalantarmoradian, A. Sung-Miller, D. Song, N. Wong, and R. D. Wesel, "Efficient Computation of Viterbi Decoder Reliability with an Application to Variable-Length Coding", *IEEE Transactions on Communications*, Early Access, July 2022 (accepted).
2. H. Yang, M. Pan, A. Antonini, and R. D. Wesel, "Sequential Transmission Over Binary Asymmetric Channels With Feedback", *IEEE Transactions on Information Theory*, Early Access, June 2022 (accepted).
3. H. Yang, E. Liang, M. Pan and R. D. Wesel, "CRC-Aided List Decoding of Convolutional Codes in the Short Blocklength Regime," in *IEEE Transactions on Information Theory*, **DOI**: 10.1109/TVT.2022.3150717.
4. L. Wang, C. Terrill, M. Stark, Z. Li, S. Chen, C. Hulse, C. Kuo, R. Wesel, G. Bauch, R. Pitchumani, "Reconstruction-Computation-Quantization (RCQ): A Paradigm for Low Bit Width LDPC Decoding," *IEEE Transactions on Communications*, 2022 Early Access Article, **DOI**: 10.1109/TCOMM.2022.3149913.
5. N. Farsad, W. Chuang, A. Goldsmith, C. Komninakis, M. Médard, C. Rose, L. Vandenberghe, E. E. Wesel, R. D. Wesel, "Capacities and Optimal Input Distributions for Particle-Intensity

- Channels,” *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*. December 2020, vol. 6, no. 3, pp. 220-232, DOI: [10.1109/TMBMC.2020.3035371](https://doi.org/10.1109/TMBMC.2020.3035371).
6. T. Wiegart, F. D. Ros, M. P. Yankov, F. R. Steiner, S. Gaiarin, R. D. Wesel, “Probabilistically Shaped 4-PAM for Short-Reach IM/DD Links with a Peak Power Constraint”, *Journal of Lightwave Technology*, October 2020 (Early Access) . DOI: [10.1109/JLT.2020.3029371](https://doi.org/10.1109/JLT.2020.3029371).
 7. M. Stark, L. Wang, G. Bauch, R. D. Wesel, "Decoding Rate-Compatible 5G-LDPC Codes with Coarse Quantization Using the Information Bottleneck Method", *IEEE Open Journal of the Communications Society*, May 2020, vol. 1, pp. 646-660, DOI: [10.1109/OJCOMS.2020.2994048](https://doi.org/10.1109/OJCOMS.2020.2994048).
 8. H. Wang, S. V. S. Ranganathan, R. D. Wesel, “Variable-Length Coding with Shared Incremental Redundancy: Design Methods and Examples,” *IEEE Transactions on Communications*, June 2019, vol. 67, no. 9, pp. 5981-5995. DOI: [10.1109/TCOMM.2019.2919626](https://doi.org/10.1109/TCOMM.2019.2919626).
 9. S. V. S. Ranganathan, D. Divsalar, R. D. Wesel, "Quasi-Cyclic Protograph-Based Raptor-Like LDPC Codes for Short Block-Lengths", *IEEE Transactions on Information Theory*, June 2019, vol. 65, no. 6, pp. 3758-3777, DOI: [10.1109/TIT.2019.2895322](https://doi.org/10.1109/TIT.2019.2895322).
 10. A. Heidarzadeh, J. Chamberland, R. D. Wesel, and P. Parag, "A Systematic Approach to Incremental Redundancy with Application to Erasure Channels", *IEEE Transactions on Communications*, April 2019, vol. 67, no. 4, pp. 2620-2631, DOI: [10.1109/TCOMM.2018.2889254](https://doi.org/10.1109/TCOMM.2018.2889254).
 11. S. V. S. Ranganathan, “Allocating Redundancy Between Erasure Coding and Channel Coding when Fading Channel Diversity Grows with Codeword Length,” *IEEE Transactions on Communications*, May 2017, vol. 65, no. 8, pp. 3226 - 3237. DOI: [10.1109/TCOMM.2017.2706728](https://doi.org/10.1109/TCOMM.2017.2706728).
 12. H. Wang, N. Wong, T.-Y. Chen, R. D. Wesel, “Using Dynamic Allocation of Write Voltage to Extend Flash Memory Lifetime”, *IEEE Transactions on Communications*, November 2016, vol. 64, no. 11, pp. 4474-4486, DOI: [10.1109/TCOMM.2016.2607707](https://doi.org/10.1109/TCOMM.2016.2607707).
 13. K. Vakulinia, S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel, "Optimizing Transmission Lengths for Limited Feedback with Non-Binary LDPC Examples” *IEEE Transactions on Communications*, June 2016, vol. 64, no. 6, pp. 2245-2257, DOI: [10.1109/TCOMM.2016.2538770](https://doi.org/10.1109/TCOMM.2016.2538770).
 14. C.-Y. Lou, B. Daneshrad, and R. D. Wesel, "Convolutional-Code-Specific CRC Code Design", *IEEE Transactions on Communications*, October 2015, vol. 63, no. 10, pp. 3459 – 3469, DOI: [10.1109/TCOMM.2015.2459058](https://doi.org/10.1109/TCOMM.2015.2459058).
 15. A. R. Williamson, T.-Y. Chen and R. D. Wesel, "Variable-length Convolutional Coding for Short Blocklengths with Decision Feedback", *IEEE Transactions on Communications*, July 2015, vol. 63, no. 7, pp. 2389 – 2403, DOI: [10.1109/TCOMM.2015.2429583](https://doi.org/10.1109/TCOMM.2015.2429583).
 16. T.-Y. Chen, K. Vakulinia, D. Divsalar, and R. D. Wesel, "Protograph-Based Raptor-Like LDPC Codes", *IEEE Transactions on Communications*, Vol. 63, No. 5, pp 1522-1530, May 2015, DOI: [10.1109/TCOMM.2015.2404842](https://doi.org/10.1109/TCOMM.2015.2404842).
 17. A. R. Williamson, M. Marshall, and R. D. Wesel, “Reliability-output Decoding of Tail-biting Convolutional Codes,” *IEEE Transactions on Communications*, Vol. 62, No. 6, pp 1768-1778, June 2014, DOI: [10.1109/TCOMM.2014.2319264](https://doi.org/10.1109/TCOMM.2014.2319264).
 18. J. Wang, K. Vakulinia, T.-Y. Chen, T. Courtade, G. Dong, T. Zhang, H. Shankar, and R. D. Wesel, “Enhanced Precision Through Multiple Reads for LDPC Decoding in Flash Memories” *IEEE*

Journal on Selected Areas in Communications, Vol. 32, No. 5, pp 880-891, May 2014,

DOI: [10.1109/JSAC.2014.140508](https://doi.org/10.1109/JSAC.2014.140508).

19. T.A. Courtade and R. D. Wesel, "Coded Cooperative Data Exchange in Multihop Networks," *IEEE Transactions on Information Theory*, Vol. 60, No. 2, pp. 1136-1158, February 2014, **DOI:** [10.1109/TIT.2013.2290993](https://doi.org/10.1109/TIT.2013.2290993).
20. J. Wang, L. Dolecek and R. Wesel, "The Cycle Consistency Matrix Approach to LDPC Absorbing Sets in Separable Circulant-Based Codes," *IEEE Transactions on Information Theory*, Vol. 59, No. 4, pp 2293 - 2314, April 2013, **DOI:** [10.1109/TIT.2012.2235122](https://doi.org/10.1109/TIT.2012.2235122).
21. B. Xie, T. A. Courtade, and R. D. Wesel, "Optimal Encoding Schemes for Several Classes of Discrete Degraded Broadcast Channels," *IEEE Transactions on Information Theory*, Vol. 59, No. 3, pp. 1360-1378, March 2013, **DOI:** [10.1109/TIT.2012.2237095](https://doi.org/10.1109/TIT.2012.2237095).
22. M. Griot, A. I. Vila Casado, W.-Y. Weng, H. Chan and R. D. Wesel, "Nonlinear Trellis Codes for Binary-Input Binary-Output Multiple Access Channels With Single-User Decoding," *IEEE Transactions on Communications*, Vol. 60, No. 2, Feb. 2012, **DOI:** [10.1109/TCOMM.2012.010512.070116](https://doi.org/10.1109/TCOMM.2012.010512.070116).
23. T. A. Courtade and R. D. Wesel, "Optimal Allocation of Redundancy Between Packet-Level Erasure Coding and Physical-Layer Channel Coding in Fading Channels," *IEEE Transactions on Communications*, Vol. 59, No. 8, pp. 2101-2109, August 2011, **DOI:** [10.1109/TCOMM.2011.062311.090277](https://doi.org/10.1109/TCOMM.2011.062311.090277).
24. A. I. Vila Casado, M. Griot, and R. D. Wesel, "LDPC Decoders with Informed Dynamic Scheduling," *IEEE Transactions on Communications*, Vol. 58, No. 12, pp 3470-3479, December 2010, **DOI:** [10.1109/TCOMM.2010.101910.070303](https://doi.org/10.1109/TCOMM.2010.101910.070303).
25. A. I. Vila Casado, W.-Y. Weng, S. Valle, and R. D. Wesel, "Multiple-Rate Low-Density Parity-Check Codes with Constant Blocklength," *IEEE Transactions on Communications*, Vol. 57, No. 1, pp 75-83, January 2009, **DOI:** [10.1109/TCOMM.2009.0901.060256](https://doi.org/10.1109/TCOMM.2009.0901.060256).
26. H. Chan, A. I. Vila Casado, J. Basak, M. Griot, W.-Y. Weng, R. D. Wesel, B. Jalali, E. Yablonovitch, I. Verbauwhede, "Demonstration of Uncoordinated Multiple Access in Optical Communications," *IEEE Transactions on Circuits and Systems-I: Regular Papers*, Vol. 55, No. 10, pp 3259-3269, November 2008, **DOI:** [10.1109/TCSI.2008.925365](https://doi.org/10.1109/TCSI.2008.925365).
27. W.-Y. Weng, C. Kose, B. Xie and R. D. Wesel, "Universal Serially Concatenated Trellis Coded Modulation for Space-Time Channels," *IEEE Transactions on Communications*, Vol. 56, No. 10, pp 1636-1646, October 2008, **DOI:** [10.1109/TCOMM.2008.060530](https://doi.org/10.1109/TCOMM.2008.060530).
28. B. Xie, M. Griot, A. I. Vila Casado, and R. D. Wesel, "Optimal Transmission Strategy and Explicit Capacity Region for Broadcast Z Channels," *IEEE Transactions on Information Theory*, Vol. 53, No. 9, pp 4296-4304, September 2008, **DOI:** [10.1109/TIT.2008.928298](https://doi.org/10.1109/TIT.2008.928298).
29. J. Shi and R. D. Wesel, "A Study on Universal Codes with Finite Block Lengths," *IEEE Transactions on Information Theory*, Vol. 54, No. 9, pp 3066-3074, September 2007, **DOI:** [10.1109/TIT.2007.903156](https://doi.org/10.1109/TIT.2007.903156).

30. M. Griot, W.-Y. Weng and R. D. Wesel, "A Tighter Bhattacharyya Bound for Decoding Error Probability," *IEEE Communications Letters*, Vol.11, No. 4, pp 346-347, April 2007, DOI: [10.1109/LCOM.2007.348296](https://doi.org/10.1109/LCOM.2007.348296).
31. C. R. Jones, T. Tian, J. Villaseñor and R. D. Wesel, "The Universal Operation of LDPC Codes Over Scalar Fading Channels," *IEEE Transactions on Communications*, Vol. 55, no. 1, pp 122-132, Jan. 2007, DOI: [10.1109/TCOMM.2006.885081](https://doi.org/10.1109/TCOMM.2006.885081).
32. Kose C. and Wesel R. D., "Universal Space-Time Codes from Demultiplexed Trellis Codes," *IEEE Trans. on Communications*. Vol.54. No 7. July 2006, DOI: [10.1109/TCOMM.2006.877967](https://doi.org/10.1109/TCOMM.2006.877967).
33. F. Peng, W. Ryan. and R. D. Wesel," Surrogate-Channel Design of Universal LDPC Codes," *IEEE Comm. Letters*, Jun. 2006, vol. 10, no. 6, pp 480-482, DOI: [10.1109/LCOMM.2006.1638622](https://doi.org/10.1109/LCOMM.2006.1638622)
34. A. Ramamoorthy, J. Shi and R. D. Wesel, "On the Capacity of Network Coding for Random Networks," *IEEE Transactions on Information Theory*, Aug. 2005, Vol. 51, no. 8, pp 2878-2885, DOI: [10.1109/TIT.2005.851725](https://doi.org/10.1109/TIT.2005.851725).
35. Wesel R. D., "Reduced-State Representations for Trellis Codes Using Constellation Symmetry," *IEEE Transactions on Communications*, Aug. 2004, vol. 52, no. 8 pp 1302-1310, DOI: [10.1109/TCOMM.2004.833023](https://doi.org/10.1109/TCOMM.2004.833023).
36. Tian T., Jones C., Villaseñor J. D. and Wesel R. D., "Selective Avoidance of Cycles in Irregular LDPC Code Construction," *IEEE Transactions on Communications*, Aug. 2004, vol. 52(8), pp 1242-1247, DOI: [10.1109/TCOMM.2004.833048](https://doi.org/10.1109/TCOMM.2004.833048).
37. Sun T. W., Wesel R. D., Shane M. R. and Jarett K., "Superposition Turbo-TCM for Multi-Rate Broadcast," *IEEE Transactions on Communications*, Mar. 2004, vol. 52(3), pp 368-371, DOI: [10.1109/TCOMM.2004.823646](https://doi.org/10.1109/TCOMM.2004.823646).
38. Shi J. and Wesel R. D., "Efficient Computation of Trellis Code Generating Functions," *IEEE Transactions on Communications*, Feb. 2004, vol. 52(2), pp 219-227, DOI: [10.1109/TCOMM.2003.822702](https://doi.org/10.1109/TCOMM.2003.822702).
39. Matache A. and Wesel R. D., "Universal Trellis Codes for Diagonally Layered Space-Time Systems," *IEEE Transactions on Signal Processing*, Nov. 2003, vol. 51(11) pp 2773-2783, Special Issue on "MIMO Wireless", DOI: [10.1109/TSP.2003.818159](https://doi.org/10.1109/TSP.2003.818159).
40. Kose C. and Wesel R. D., "Universal Space-Time Trellis Codes," *IEEE Trans. on Info. Theory*, Oct. 2003, vol. 40(10) pp 2717-2727 Special Issue on "Space-Time Transmission, Reception, Coding and Signal Design," DOI: [10.1109/TIT.2003.817459](https://doi.org/10.1109/TIT.2003.817459).
41. Q. Zhang, P. K. Varshney, and R. D. Wesel, "Optimal Bi-level quantization of I.I.D Sensor Observations for Binary Hypothesis Testing," *IEEE Transactions on Information Theory*, July 2002, 48(7) pp. 2105-2110, DOI: [10.1109/TIT.2002.1013153](https://doi.org/10.1109/TIT.2002.1013153).
42. C. Komninakis, C. Fragouli, A. H. Sayed, and R. D. Wesel, "Multi-Input, Multi-output Fading Channel Tracking and Equalization Using Kalman Estimation," *IEEE Transactions on Signal Processing*, May 2002, vol. 50(5), pp. 1065-1076, DOI: [10.1109/78.995063](https://doi.org/10.1109/78.995063).
43. A. Bernard, X. Liu, R. D. Wesel, and A. Alwan, "Multi-rate Transmission of Speech Using Rate-Compatible Trellis Codes and Embedded Source Codes," *IEEE Transactions on Communications*, February 2002, vol. 50(2) pp. 309-320, DOI: [10.1109/26.983326](https://doi.org/10.1109/26.983326).
44. P. Ormeci, D. L. Goeckel, X. Liu, and R. D. Wesel, "Adaptive Bit-Interleaved Coded Modulation for Time-Varying Channels Using Outdated Fading Estimates" *IEEE Transactions on Communications*, Sept. 2001, 49(9), pp. 1572-1581, DOI: [10.1109/26.950344](https://doi.org/10.1109/26.950344).

45. R. D. Wesel, X. Liu, J. M. Cioffi, and C. Komninakis, "Constellation Labeling for Linear Encoders," *IEEE Transactions on Information Theory*, Sept. 2001, 47(6), pp. 2417-2431, DOI: [10.1109/18.945255](https://doi.org/10.1109/18.945255).
46. C. Komninakis and R.D. Wesel, "Joint Iterative Channel Estimation and Decoding in Flat Correlated Rayleigh Fading," *IEEE Journal on Selected Areas in Communications*, Sept. 2001, 19(9), pp. 1706-17, DOI: [10.1109/49.947035](https://doi.org/10.1109/49.947035).
47. C. Fragouli and R. D. Wesel, "Turbo encoder design for symbol interleaved parallel concatenated trellis coded modulation," *IEEE Transactions on Communications*, 49(3), Mar. 2001, pp. 425-435, DOI: [10.1109/26.911450](https://doi.org/10.1109/26.911450).
48. W. Shi, T. W. Sun, and R. D. Wesel, "Quasi-convexity and Optimal Binary Fusion for Distributed Detection with Identical Sensors in Generalized Gaussian Noise," *IEEE Transactions on Information Theory*, Jan. 2001, 47(1), pp. 446-450, DOI: [10.1109/18.904560](https://doi.org/10.1109/18.904560).
49. R. D. Wesel, X. Liu, and W. Shi, "Trellis Codes for Periodic Erasures," *IEEE Transactions on Communications*, June 2000, 48(6), pp. 938-947, DOI: [10.1109/26.848553](https://doi.org/10.1109/26.848553).
50. R. D. Wesel and J. M. Cioffi, "Trellis Codes for Periodic Interleavers," *IEEE Communications Letters*, April 1999, 3(4), pp. 103-105, DOI: [10.1109/4234.757202](https://doi.org/10.1109/4234.757202).
51. R. D. Wesel, and J. Cioffi, "Achievable Rates for Tomlinson-Harashima Precoding," *IEEE Transactions on Information Theory*, Mar. 1998, 44(2), pp. 824-831, DOI: [10.1109/18.661530](https://doi.org/10.1109/18.661530).

CONFERENCE PUBLICATIONS

1. D. Song, F. Areces, L. Wang, R. D. Wesel, "Shaped TCM with List Decoding that Exceeds the RCU Bound by Optimizing a Union Bound on FER", 2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, Dec. 4-8, 2022 (accepted).
2. J. A. King, W. R. Ryan, R. D. Wesel, "CRC-Aided Short Convolutional Codes and RCU Bounds for Orthogonal Signaling", 2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, Dec. 4-8, 2022 (accepted).
3. F. Areces, D. Song, R. D. Wesel, and A. B. Wagner, "Efficiently Computable Converse for Finite-Blocklength Communication", 2022 IEEE International Symposium on Information Theory (ISIT), Espoo, Finland, June 26 - July 1, 2022.
4. H. Yang, R. C. Yavas, V. Kostina, and R. D. Wesel, "Variable-Length Stop-Feedback Codes With Finite Optimal Decoding Times for BI-AWGN Channels", 2022 IEEE International Symposium on Information Theory (ISIT), Espoo, Finland, June 26 - July 1, 2022.
5. J. Nguyen, L. Wang, C. Hulse, S. Dani, A. Antonini, T. Chauvin, D. Dariush, R. Wesel, "Neural Normalized Min-Sum Message-Passing vs. Viterbi Decoding for the CCSDS Line Product Code", ICC 2022, Seoul, South Korea, May 16–20, 2022.
6. L. Wang, D. Song, F. Areces, and R. D. Wesel. "Achieving Short-Blocklength RCU bound via CRC List Decoding of TCM with Probabilistic Shaping", ICC 2022, Seoul, South Korea, May 16–20, 2022.
7. W. Sui, H. Yang, B. Towell, A. Asmani, R. Wesel, "High-Rate Convolutional Codes with CRC-Aided List Decoding for Short Blocklengths", ICC 2022, Seoul, South Korea, May 16–20, 2022.

8. J. King, A. Kwon, H. Yang, W. Ryan, R. Wesel, "CRC-Aided List Decoding of Convolutional and Polar Codes for Short Messages in 5G", ICC 2022, Seoul, South Korea, May 16-20, 2022.
9. C. Terrill, L. Wang, S. Chen, C. Hulse, C. Kuo, R. Wesel, D. Divsalar, "FPGA Implementations of Layered MinSum LDPC Decoders Using RCQ Message Passing", 2021 IEEE Global Communications Conference (GLOBECOM), Madrid, Spain, Dec. 7-11, 2021.
10. L. Wang, S. Chen, J. Nguyen, D. Dariush, R. Wesel, "Neural-Network-Optimized Degree-Specific Weights for LDPC MinSum Decoding", 2021 International Symposium on Topics in Coding (ISTC), Montréal, Canada, Aug. 30-Sept. 3, 2021.
11. A. Antonini, R. Gimelshein, R. D. Wesel, "Causal (Progressive) Encoding over Binary Symmetric Channels with Noiseless Feedback", 2021 IEEE International Symposium on Information Theory (ISIT), Melbourne, Victoria, Australia, July 12-20, 2021.
12. D. Xiao, L. Wang, D. Song, R. D. Wesel, "Finite-Support Capacity-Approaching Distributions for AWGN Channels," 2020 IEEE Information Theory Workshop, Online, April 11-15 2021.
13. A. Arafa, R. D. Wesel, "Timely Transmission Using Optimized Variable Length Coding," 2021 IEEE INFOCOM Age of Information Workshop, Online, May 10, 2021.
14. L. Wang, M. Stark, R. D. Wesel, G. Bauch, "A Reconstruction-Computation-Quantization (RCQ) Approach to Node Operations in LDPC Decoding," 2020 IEEE Global Communications Conference, Taipei, Taiwan, Dec. 7-11, 2020.
15. J. Nguyen, E. Liang, L. Wang, T. Drullinger, T. Chauvin, R. D. Wesel, "Comparison of Integrated and Independent RF/FSOTransceivers on a Fading Optical Channel," 2020 Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, Nov. 1-5, 2020.
16. Hengjie Yang, Linfang Wang, Vincent Lau, Richard D. Wesel, "An Efficient Algorithm for Designing Optimal CRCs for Tail-Biting Convolutional Codes", 2020 IEEE International Symposium on Information Theory, June 21-26, 2020, Los Angeles, CA.
17. Amael Antonini, Hengjie Yang, Richard D. Wesel, "Low Complexity Algorithms for Transmission of Short Blocks over the BSC with Full Feedback," 2020 IEEE International Symposium on Information Theory, June 21-26, 2020, Los Angeles, CA.
18. Hengjie Yang, Richard D. Wesel, "Finite-Blocklength Performance of Sequential Transmission over BSC with Noiseless Feedback", 2020 IEEE International Symposium on Information Theory, June 21-26, 2020, Los Angeles, CA.
19. Maximilian Stark, Gerhard Bauch, Linfang Wang, Richard D. Wesel, "Information Bottleneck Decoding of Rate-Compatible 5G-LDPC Codes", 2020 IEEE Int. Conf. Commun., Virtual Conference, June 7-11, 2020.
20. H. Yang, E. Liang, H. Yao, A. Vardy, D. Divsalar, and R. D. Wesel, "A List-Decoding Approach to Low-Complexity Soft Maximum-Likelihood Decoding of Cyclic Codes," 2019 IEEE Global Communications Conference, Dec. 9-13, 2019, Waikoloa, HI, USA.
21. E. Liang, H. Yang, H. Yao, D. Divsalar, and R. D. Wesel, "List-Decoded Tail-Biting Convolutional Codes with Distance-Spectrum Optimal CRCs for 5G," 2019 IEEE Global Communications Conference, Dec. 9-13, 2019, Waikoloa, HI, USA.

22. N. Wong, H. Wang, E. Liang, S. V. S. Ranganathan, and R. D. Wesel, "Decoding Flash Memory with Progressive Reads and Independent vs. Joint Encoding of Bits in a Cell," 2019 IEEE Global Communications Conference, Dec. 9-13, 2019, Waikoloa, HI, USA.
23. H. Yang and R. D. Wesel, "On the Most Informative Boolean Functions of the Very Noisy Channel," 2019 IEEE International Symposium on Information Theory, July 7-12, 2019, Paris, France.
24. H. Yang and R. D. Wesel, "Serial List Viterbi Decoding with CRC: Managing Errors, Erasures, and Complexity," 2018 IEEE Global Communications Conference, Dec. 9-13, 2018, Abu Dhabi, UAE.
25. H. Wang and R. D. Wesel, "Channel Code Analysis and Design using Multiple Variable-Length Codes in Parallel without Feedback," 2018 IEEE Global Communications Conference, Dec. 9-13, 2018, Abu Dhabi, UAE.
26. R. D. Wesel, N. Wong, A. Baldauf, A. Belhouchat, A. Heidarzadeh, and J.-F. Chamberland, "Transmission Lengths that Maximize Throughput of Variable-Length Coding & ACK/NACK Feedback," 2018 IEEE Global Communications Conference, Dec. 9-13, 2018, Abu Dhabi, UAE.
27. A. Baldauf, A. Belhouchat, N. Wong, R. D. Wesel, "Efficient Computation of Convolutional Decoder Reliability Without a CRC," Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, October 28-31, 2018.
28. S. V. S. Ranganathan, R. D. Wesel and D. Divsalar, "Linear Rate-Compatible Codes with Degree-1 Extending Variable Nodes Under Iterative Decoding," 2018 IEEE International Symposium on Information Theory (ISIT), Vail, CO, USA, 2018, pp. 1166-1170.
29. A. Heidarzadeh, J. Chamberland, P. Parag and R. D. Wesel, "A Systematic Approach to Incremental Redundancy over Erasure Channels," 2018 IEEE International Symposium on Information Theory (ISIT), Vail, CO, USA, 2018, pp. 1176-1180.
30. R. D. Wesel, E. E. Wesel, L. Vandenberghe, C. Kominakis, and M. Medard, "Efficient Binomial Channel Capacity Computation with an Application to Molecular Communication," IEEE 2018 Inf. Theory and Applications (ITA) Workshop. La Jolla, CA. February 11 - February 16, 2018.
31. S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel, "Design of Improved Quasi-Cyclic Protograph-Based Raptor-Like LDPC Codes for Short Block-Lengths," IEEE Int. Symp. Inf. Theory (ISIT). June 2017.
32. H. Wang, S. V. S. Ranganathan, and R. D. Wesel, "Approaching Capacity Using Incremental Redundancy without Feedback," IEEE Int. Symp. Inf. Theory (ISIT).). June 2017.
33. H. Wang, N. Wong, A. M. Baldauf, C. K. Bachelor, S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel, "An Information Density Approach to Analyzing and Optimizing Incremental Redundancy with Feedback," IEEE Int. Symp. Inf. Theory (ISIT).). June 2017.
34. N. Wong, K. Vakili, H. Wang, S. V. S. Ranganathan, and R. D. Wesel, "Sequential Differential Optimization of Incremental Redundancy Transmission Lengths: An Example with Tail-Biting Convolutional Codes," IEEE 2016 Inf. Theory and Applications (ITA) Workshop. La Jolla, CA. February 12 - February 17, 2017.
35. S. V. S. Ranganathan, K. Vakili, D. Divsalar, and R. D. Wesel, "Universal Rate-Compatible LDPC Code Families for Any Increment Ordering". 9th Int. Symp. Turbo Codes & Iterative Information Processing (ISTC), Brest, France, September 5 - September 9, 2016.
36. R. D. Wesel, K. Vakili, S. V. S. Ranganathan, T. Mu, and D. Divsalar, "Resource-Aware Incremental Redundancy in Feedback and Broadcast," In Proceedings of the 2016 International Zurich Seminar on Communications, March 2-4, 2016, pp 63-67.

37. S. V. S. Ranganathan, K. Vakulinia, L. Dolecek, D. Divsalar, and R. D. Wesel, "Some Results on Spatially Coupled Protograph LDPC Codes". IEEE 2016 Inf. Theory and Applications (ITA) Workshop. La Jolla, CA. February 12 - 17, 2016.
38. H. Wang, N. Wong, and R. D. Wesel, "Dynamic Voltage Allocation with Quantized Voltage Levels and Simplified Channel Modeling," 49th Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA November 8-11, 2015.
39. S. V. S. Ranganathan, D. Divsalar, and R. D. Wesel, "On the Girth of $(3,L)$ Quasi-Cyclic LDPC Codes Based on Complete Photographs". IEEE Int. Symp. Inf. Theory (ISIT). Wanchai, Hong Kong. June 14 - 19, 2015. DOI: 10.1109/ISIT.2015.7282491
40. K. Vakulinia, D. Divsalar, and R. D. Wesel, "RCA Analysis of the Polar Codes and the use of Feedback to aid Polarization at Short Blocklengths", IEEE Int Symp. Inf. Theory (ISIT), Wanchai, Hong Kong, June 14-19, 2015.
41. H. Wang, T.-Y. Chen, and R. D. Wesel, "Histogram-Based Flash Channel Estimation", IEEE Int. Conf. Commun., London, UK, June 8-12, 2015.
42. C.-Y. Lou, B. Daneshrad, and R. D. Wesel, "Optimizing Pilot Length for a Go/No-Go Decision in Two-State Block Fading Channels with Feedback", IEEE Int. Conf. Commun., London, UK, June 8-12, 2015.
43. K. Vakulinia, D. Divsalar, and R. D. Wesel, "Protograph-Based Raptor-Like LDPC Codes for the Binary Erasure Channel". IEEE 2015 Inf. Theory and Applications (ITA) Workshop. La Jolla, CA. February 1 - February 6, 2015.
44. K. Vakulinia, A. R. Williamson, S. V. S. Ranganathan, D. Divsalar, R. D. Wesel, "Feedback Systems using Non-Binary LDPC Codes with a Limited Number of Transmissions". IEEE Information Theory Workshop (ITW). Hobart, Tasmania, Australia. Nov. 2 – Nov. 5, 2014.
45. K. Vakulinia, D. Divsalar, and R. D. Wesel, "Optimized Degree Distributions for Binary and Non-Binary LDPC Codes in Flash Memory". IEEE Int. Symp. Inf. Theory and Applicat. (ISITA), Melbourne, Australia, October 26 – 29, 2014.
46. K. Vakulinia, T.-Y. Chen, S. V. S. Ranganathan, A. R. Williamson, D. Divsalar, and R. D. Wesel, "Short-Blocklength Non-Binary LDPC Codes with Feedback-Dependent Incremental Transmissions". IEEE Int. Symp. Inf. Theory (ISIT), Honolulu, Hawaii, USA, June 29 - July 4, 2014.
47. S. V. S. Ranganathan, D. Divsalar, K. Vakulinia, and R. D. Wesel, "Design of High-Rate Irregular Non-binary LDPC Codes Using Algorithmic Stopping-Set Cancellation". IEEE Int. Symp. Inf. Theory (ISIT), Honolulu, Hawaii, USA, June 29 - July 4, 2014.
48. T.-Y. Chen, A. R. Williamson, and R. D. Wesel, "Asymptotic Expansion and Error Exponent of Two-Phase Variable-Length Coding with Feedback for Discrete Memoryless Channels". IEEE Int. Symp. Inf. Theory (ISIT), Honolulu, Hawaii, USA, June 29 - July 4, 2014.
49. T.-Y. Chen, D. Divsalar, and R. D. Wesel, "Finite-blocklength analysis of rate-compatible codes". IEEE Int. Symp. Inf. Theory (ISIT), Honolulu, Hawaii, USA, June 29 - July 4, 2014.
50. T.-Y. Chen, A. R. Williamson, and R. D. Wesel, "Increasing Flash Memory Lifetime by Dynamic Voltage Allocation for Constant Mutual Information". IEEE Inf. Theory and Applications (ITA) Workshop, San Diego, CA, USA, February 9 - 14, 2014.
51. T.-Y. Chen, A. R. Williamson, and R. D. Wesel, "Variable-length Coding with Feedback: Finite-length Codewords and Periodic Decoding". IEEE Int. Symp. Inf. Theory (ISIT), Istanbul, Turkey, July 7 - 12, 2013.

52. A. R. Williamson, T.-Y. Chen, and R. D. Wesel, "Reliability-based Error Detection for Feedback Communication with Low Latency". IEEE Int. Symp. Inf. Theory (ISIT), Istanbul, Turkey, July 7 - 12, 2013.
53. A. R. Williamson, T.-Y. Chen, and R. D. Wesel, "Firing the Genie: Two-Phase Short-blocklength Convolutional Coding with Feedback". IEEE Inf. Theory and Applications (ITA) Workshop, San Diego, CA, USA, February 12-16, 2013.
54. T.-Y. Chen, D. Divsalar, and R. D. Wesel, "Chernoff Bounds for Analysis of Rate- Compatible Sphere-Packing with Numerous Transmissions". IEEE Inf. Theory Workshop (ITW), Lausanne, Switzerland September 3 - 7, 2012.
55. A. R. Williamson, T.-Y. Chen, and R. D. Wesel, "A Rate-Compatible Sphere-Packing Analysis of Feedback Coding with Limited Retransmissions". IEEE Int. Symp. Inf. Theory (ISIT), Cambridge, MA, USA, July 1-6, 2012.
56. A. Marinoni, P. Savazzi and R. D. Wesel, "On q-ary LDPC Code Design for a Low Error Floor". Globecom 2011, Houston, Texas, USA, December 2011.
57. T.-Y. Chen, D. Divsalar, J. Wang and R. D. Wesel, "Protograph-Based Raptor-Like LDPC Codes for Rate Compatibility with Short Blocklengths". Globecom 2011, Houston, Texas, USA, December 2011.
58. J. Wang, H. Shankar and R. D. Wesel, "Soft Information for LDPC Decoding in Flash: Mutual-Information Optimized Quantization". Globecom 2011, Houston, Texas, USA, December 2011.
59. T. A. Courtade, J. Wang, and R. D. Wesel, "Superposition Coding to Support Multiple Streams, Priorities, and Channel Capacities in the Context of GMSK", to appear in MILCOM, Nov 2011.
60. T. A. Courtade and R. D. Wesel, "Weighted Universal Recovery, Practical Secrecy, and an Efficient Algorithm for Solving Both". Forty-Ninth Annual Allerton Conference on Communication, Control, and Computing, 2011.
61. B. Xie, M. van der Schaar, T. A. Courtade and R. D. Wesel, "Minimizing Weighted Sum Finish Time for One-To-Many File Transfer in Peer-To-Peer Networks". Forty-Ninth Annual Allerton Conference on Communication, Control, and Computing, 2011.
62. T. A. Courtade and R. D. Wesel, "Multiterminal Source Coding with an Entropy-Based Distortion Measure". ISIT 2011, Saint-Petersburg, Russia, August 2011.
63. J. Wang, L. Dolecek, Z. Zhang and R. D. Wesel, "Absorbing Set Spectrum Approach for Practical Code Design". ISIT 2011, Saint-Petersburg, Russia, August 2011.
64. J. Wang, L. Dolecek and R. D. Wesel, "Controlling LDPC Absorbing Sets via the Null Space of the Cycle Consistency Matrix". ICC 2011, Kyoto, Japan, June 2011.
65. T.-Y. Chen, N. Seshadri and R. D. Wesel, "A Sphere-Packing Analysis of Incremental Redundancy with Feedback". ICC 2011, Kyoto, Japan, June 2011.
66. J. Wang, L. Dolecek and R. D. Wesel, "LDPC Absorbing Sets, the Null Space of the Cycle Consistency Matrix, and Tanner's Constructions". ITA 2011, San Diego, CA, USA, Feb. 2011.
67. T.-Y. Chen, N. Seshadri and R. D. Wesel, "Incremental Redundancy: A Comparison of a Sphere-Packing Analysis and Convolutional Codes". ITA 2011, San Diego, CA, USA, Feb 2011.
68. T. A. Courtade and R. D. Wesel, "Efficient Universal Recovery in Broadcast Networks". Forty-Eighth Annual Allerton Conference on Communication, Control, and Computing: Monticello, IL, Sept. 29-Oct. 1, 2010.

69. E. L. Valles, R. D. Wesel, J. D. Villasenor, C. R. Jones and M. Simon, "Pilotless Carrier Phase-Synchronization via LDPC Code Feedback". MILCOM 2010: San Jose, CA, Oct. 31 - Nov 3, 2011.
70. T. A. Courtade, B. Xie, and R. D. Wesel, "Optimal Exchange of Packets for Universal Recovery in Broadcast Networks". MILCOM 2010: San Jose, CA, Oct. 31 - Nov 3, 2010.
71. A. Marinoni, P. Savazzi and R. D. Wesel, "Protograph-based q-ary LDPC Codes for Higher-Order Modulation". ISTC 2010: Brest, France, Sept 6-10, 2010.
72. T. A. Courtade and R. D. Wesel, "A Deterministic Approach to Rate-Compatible Fountain Communication". ITA 2010: La Jolla, CA, Jan. 31 - Feb. 5, 2010.
73. B. Xie and R. D. Wesel, "Optimal Natural Encoding Scheme for Discrete Multiplicative Degraded Broadcast Channels". ISIT 2009: Seoul, Korea, June 28-July 3, 2009.
74. A. Marinoni, T. A. Courtade and R. D. Wesel, "Spectrally Efficient LDPC Coded Modulations". GTTI 2009: sessione Trasmissione, Parma, Italy, June 23-25, 2009. ***Winner of the Francesco Carassa award for the best paper from a young scientist in the sessione Trasmissione.***
75. T. A. Courtade and R. D. Wesel, "A Cross-Layer Perspective on Rateless Coding for Wireless Channels". ICC 2009, Dresden, Germany, June 14-18, 2009.
76. B. Xie and R. D. Wesel, "Optimal Independent-Encoding Schemes for Input-Symmetric Degraded Broadcast Channels". ITA 2008, San Diego, USA, Jan. 27 - Feb. 1, 2008.
77. B. Xie and R. D. Wesel, "A Mutual Information Invariance Approach to Symmetry in Discrete Memoryless Channels". ITA 2008, San Diego, USA, Jan. 27 - Feb. 1, 2008.
78. Miguel Griot, A. I. Vila Casado, and R. D. Wesel, "Nonlinear Turbo Codes for Higher-Order Modulations". ICC 2008, Beijing, China, May 2008.
79. Y.M. Chang, A. I. Vila Casado, M.F. Chang, and R. D. Wesel, "Lower-Complexity Layered Belief Propagation Decoding of LDPC Codes". ICC 2008, Beijing, China, May 2008.
80. M. Griot, A. I. Vila Casado, and R. D. Wesel, "On the Design of Arbitrarily Low-Rate Turbo-Codes". Globecom 2007, Washington, D.C., November 2007.
81. A. I. Vila Casado, M. Griot and R. D. Wesel, "Improving LDPC Decoders via Informed Dynamic Scheduling". ITW 2007, Lake Tahoe, CA, USA, September 2007, pp 208-213.
82. B. Xie, M. Griot, A. I. Vila Casado and R. D. Wesel, "Optimal Transmission Strategy and Capacity Region for Broadcast Z Channels". ITW 2007, Lake Tahoe, CA, USA, September 2007, pp 390-395.
83. A. I. Vila Casado, M. Griot and R. D. Wesel, "Informed Dynamic Scheduling for Belief-Propagation Decoding of LDPC Codes". ICC 2007, Glasgow, Scotland, July 2007, pp 932-937.
84. H. Chan, M. Griot, A. Vila Casado, R. Wesel, I. Verbauwhede "High Speed Channel Coding Architectures for the Uncoordinated OR Channel". IEEE 17th INTERNATIONAL CONFERENCE ON Application-specific Systems, Architectures and Processors (ASAP), Steamboat Springs, Colorado, September 2006.
85. M. Griot, A. I. Vila Casado and R. D. Wesel "Non-linear Turbo Codes for Interleaver-Division Multiple Access on the OR Channel". Globecom 2006, 27 Nov. - 1 Dec., San Francisco, USA.
86. W.-Y. Weng, B. Xie and R. D. Wesel "Universal Space-Time Serially Concatenated Trellis Coded Modulations". Globecom 2006, 27 Nov. - 1 Dec., San Francisco, USA.

87. A. I. Vila Casado, S. Valle, W.-Y. Weng, and R. D. Wesel "Constant-Blocklength Multiple-Rate LDPC Codes for Analog-Decoding Implementations". Proceedings Analog Decoding Workshop, June 2006.
88. E. Valles, C. Jones, R. Wesel and J. Villasenor "Carrier and Timing Synchronization of BPSK via LDPC Code Feedback". IEEE 40th Asilomar Conference on Signals, Systems and Computers. Pacific Grove, CA. Nov. 2006.
89. M. Simon, E. Valles, C. Jones, R. Wesel and J. Villasenor "Information-Reduced Carrier Synchronization of BPSK and QPSK Using Soft Decision Feedback" The IEEE 44th Annual Allerton Conference on Communication, Control and Computing. Urbana, IL Sep. 2006.
90. M. Griot, A. I. Vila Casado, W.-Y. Weng, H. Chan, J. Basak, E. Yablonovitch, I. Verbauwhede, B. Jalali, and R. D. Wesel "Trellis Codes with Low Ones Density for the OR Multiple Access Channel", IEEE ISIT, Seattle, USA, July 2006.
91. M. Griot, A. I. Vila Casado, W.-Y. Weng, H. Chan, J. Basak, E. Yablonovitch, I. Verbauwhede, B. Jalali, and R. D. Wesel "Interleaver-Division Multiple Access on the OR Channel" Information Theory and Applications workshop. San Diego, 2006.
92. Yang Han, William E. Ryan and Wesel R. D. "Dual-Mode Decoding of Product Codes with Application to Tape Storage", IEEE GlobeCom 2005. St Louis, MO., vol. 3, 28 Nov.-2 Dec. 2005, pp 1255-1260.
93. Valles E., Vila Casado A. I., Blaum M., Villasenor J and Wesel R. D., "Hamming Codes Are Rate-Efficient Array Codes", IEEE GlobeCom 2005. St Louis, MO., Vol. 3, 28 Nov.-2 Dec. 2005, pp 1320-1324.
94. Kose C. and Wesel R. D., "Universal Space-Time Codes from standard trellis codes", *GlobeCom 2004*, Dallas, TX, Nov. 29-Dec. 3, 2004, pp 391-395.
95. Shi J. and Wesel R. D., "Channel Eigenvector-Invariant Space-Time Constellations", *GlobeCom 2004*, 29 Nov.-3 Dec. 2004, pp 530-534
96. Shi J. and Wesel R. D., "Universal Codes with Finite Block Lengths", *MilCom 2004*, 31 Oct.-3 Nov. 2004, pp 1338-1344
97. Ramamoorthy A. and Wesel R. D., "Analysis of an Algorithm for Irregular LDPC Code Construction," ISIT 2004, Chicago, Illinois
98. Shi J. and Wesel R. D., "Rotationally Invariant Space-Time Constellations," ISIT 2004, Chicago, Illinois
Shi J. and Wesel R. D., "Channel Eigenvector-Invariant Space-Time Constellations" *GlobeCom 2004*, Dallas, TX, Nov. 29-Dec. 3, 2004, pp 391-395.
99. Vila Casado A. I., Weng W. and Wesel R. D., "Multiple-Rate Low-Density Parity-Check Codes with Constant Block Length", *Conference Record of Thirty-Eighth Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 7-10, 2004, pp 2010-2014.
100. Matache A., Jones C and Wesel R. D., "Reduced Complexity MIMO Detectors for LDPC Coded Systems", *MilCom 2004*, 31 Oct.-3 Nov. 2004, pp 1073-1079
101. Ramamoorthy A. and Wesel R. D., "Expansion Properties of Generalized ACE Codes", *42nd Allerton Conference on Communication, Control and Computing 2004*, Monticello, Illinois
102. Weng W., Ramamoorthy A. and Wesel R. D., "Lowering the Error Floors of High-Rate LDPC Codes by Graph Conditioning," *VTC 2004*, September 2004, Los Angeles, California, pp 2549-2553.

- 103.Chan H., Hodjat A., Shi J., Wesel R. D. and Verbaauwhede I., "Streaming Encryption for a Secure Wavelength and Time Domain Hopped Optical Network," *IEEE Intl. Conf. on Information Technology (ICIT)* 2004.
- 104.Ramamoorthy A. and Wesel R. D., "Construction of Short Block Length Irregular LDPC Codes," *ICC 2004*, Paris, June 2004.
- 105.Ramamoorthy A., Shi J., and Wesel R. D., "On the Capacity of Network Coding for Random Networks," *Allerton Conference on Communication,Control and Computing*, October 2003
- 106.Jones C., Matache A., Tian T., Villasenor J. and Wesel R. D., "The Universality of LDPC Codes on Wireless Channels," *MilCom 2003*, Boston, MA, October 2003
- 107.J. Shi and R. D. Wesel, "Further Error Event Diagram Reduction Using Algorithmic Techniques," *International Conference on Communications (ICC) 2003*, Ankorage, AK, May 2003.
- 108.C. Kose, W. Weng, and R. D. Wesel, "Serially Concatenated Trellis Coded Modulation for the Compound Periodic Erasures Channel," *International Conference on Communications (ICC) 2003*, Ankorage, AK, May 2003,pp2953-2957.
- 109.T. W. Sun, R. D. Wesel, M. R. Shane, and K. Jarett, "Superposition Turbo TCM for Multi-Rate Broadcast," *International Conference on Communications (ICC) 2003*, Ankorage, AK, May 2003.
- 110.T. Tian, C. Jones, J. D. Villasenor, and R. D. Wesel, "Construction of Irregular LDPC Codes with Low Error Floors," *International Conference on Communications (ICC) 2003*, Ankorage, AK, May 2003.
- 111.C. Kose and R. D. Wesel, "Universal Space-Time Trellis Codes," *Globecom 2002*, Taipei, Taiwan, November 2002.
- 112.C. Jones, T. Tian, A. Matache, R.D. Wesel, and J. Villasenor, "Robustness of LDPC Codes on Periodic Fading Channels," *Globecom 2002*, Taipei, Taiwan, November 2002.
- 113.M. R. Shane and R. D. Wesel,, "Reduced Complexity Iterative Demodulation and Decoding of Serial Concatenated CPM," *International Conference on Communications (ICC) 2002*, New York, NY, April 28-May 2, 2002.
- 114.A. Matache and R. Wesel, "Trellis Coding for Diagonally Layered Space-Time Systems," *International Conference on Communications (ICC) 2002*, New York, NY, April 28-May 2, 2002.
- 115.C. Kose and R. D. Wesel,, "Performance of Likelihood Ratio Tests under Incorrect Models" *35th Asilomar Conference on Signals,Systems, and Computers*, Pacific Grove, CA, Nov. 2001.
- 116.M. R. Shane and R. D. Wesel, "Reduced Complexity Iterative Demodulation and Decoding of Serial Concatenated CPM," *35th Asil. Conf. on Signals,Systems, and Computers*, Pacific Grove, CA, Nov. 2001.
- 117.A. Matache and R. Wesel, "Trellis Coding for Layered Space-Time Systems," *35th Asilomar Conference on Signals,Systems, and Computers*, (invited) Pacific Grove, CA, Nov. 2001.
- 118.C. Komninakis and R. D. Wesel, "Trellis Turbo-codes in Flat Rayleigh Fading with Diversity," *Globecom 2001*, San Antonio, TX, November 25-29, 2001.
- 119.C. Fragouli and R. Wesel, "Bit vs. Symbol Interleaving for Parallel Concatenated Trellis Coded Modulation," *Globecom 2001*, San Antonio, TX, November 25-29, 2001.
- 120.C. Komninakis, R. D. Wesel, and L. Vandenberghe, "Capacity of the Binomial Channel, or Minimax Redundancy for Memoryless Sources," *IEEE International Symposium on Information Theory*, June 24-29, 2001, Washington D. C., page 127.
- 121.C. Fragouli, R. Wesel, Dirk Sommer, and Gerhard Fettweis, "Turbo Codes with Non-Uniform QAM Constellations," *International Conference on Communications (ICC) 2001*, Helsinki, Finland, June 2001, pp.70-73.

- 122.C. Kose and R. D. Wesel, "Code Design Metrics For Space-Time Systems Under Arbitrary Fading," *ICC 2001*, Helsinki, Finland, June 2001, pp. 1099-1103.
- 123.C. Fragouli, C. Komninakis, and R. D. Wesel, "Minimality Under Periodic Puncturing," *ICC 2001*, Helsinki, Finland, June 2001, pp. 300-304.
- 124.X. Liu, P. Ormeci, R. Wesel, and D. Goeckel, "Bandwidth-Efficient, Low-Latency Adaptive Coded Modulation Schemes for Time-Varying Channels," *ICC 2001*, Helsinki, Finland, June 2001, pp. 2211-2215.
- 125.W. Shi, C. Komninakis, R. Wesel, and B. Daneshrad, "Robustness of Space-Time Turbo Codes," *ICC 2001*, Helsinki, Finland, June 2001, pp. 1700-1704.
- 126.T. W. Sun and R. D. Wesel, "Constellation Labeling for Error-Resilient Source Coding of PAM," *35th Annual Conference on Information Sciences and Systems*, March 21-23, 2001, The Johns Hopkins University, Baltimore, Maryland, page 295.
- 127.M. Shane and R. D. Wesel, "Tight Bounds on the Mutual Information of the Binary Input AWGN Channel," *35th Annual Conference on Information Sciences and Systems*, March 21-23, 2001, The Johns Hopkins University, Baltimore, Maryland, page 117.
- 128.M. Shane and R. Wesel, "Parallel Concatenated Turbo Codes for Continuous Phase Modulation," (invited) *2000 Wireless Comm. and Networking Conf.*, Sept 24-28, 2000, 6 pages.
- 129.W. Shi, T. Sun, and R. D. Wesel, "Optimal Binary Distributed Detection," *2000 International Symposium on Information Theory*, Sorrento, Italy, June 2000, pp. 437.
- 130.R. D. Wesel and X. Liu, "Edge-Profile Optimal Constellation Labeling," *IEEE International Conference on Communications 2000*, New Orleans, LA, June 2000, pp. 1198-1202.
- 131.C. Komninakis, C. Fragouli, A. H. Sayed, and R. D. Wesel, "Adaptive Multi-Input Multi-Output Fading Channel Equalization Using Kalman Estimation," *ICC 2000*, New Orleans, LA, June 2000, pp. 1655-1659.
- 132.Q. Zhang, P. K. Varshney, and R. D. Wesel, "Optimal Distributed Binary Hypothesis Testing with Independent Identical Sensors," *2000 Conference on Information Sciences and Systems*, Princeton University, March 15-17, 2000, 6 pages.
- 133.C. Komninakis and R. D. Wesel, "Pilot-Aided Joint Data and Channel Estimation in Flat Fading," *Comm. Theory Symp. at Globecom 99*, Rio de Janeiro, Brazil, Dec. 5-9, 1999, pp. 2534-9.
- 134.C. Fragouli and R. D. Wesel, "Semi-Random Interleaver Design Criteria," *Communication Theory Symposium at Globecom 99*, Rio de Janeiro, Brazil, December 5-9, 1999, pp. 2352-6.
- 135.C. Komninakis and R. D. Wesel, "Non-Pilot-Aided Iterative Decoding for Joint Data Recovery and Channel Estimation in Fading," *33rd Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA. October 24-27, 1999, pp. 1122-1126.
- 136.C. Fragouli, C. Komninakis, A. H. Sayed, R.D. Wesel, "Channel Estimation and Equalization in Fading," *33rd Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA. October 24-27, 1999, pp. 1159-1163.
- 137.W. Shi, T. Sun, R. Wesel, "Optimal Binary Thresholds for Distributed Detection in Gaussian Noise," *33rd Asilomar Conf. on Sig., Sys., Comp.*, Pac. Grove, CA. Oct. 24-27, 1999, pp. 675-9.
- 138.K. Lakovic, J. Villaseñor, and R. D. Wesel, "Robust Joint Huffman and Convolutional Decoding," *VTC-Fall-99*, Amsterdam, The Netherlands, September 19-22, 1999, pp. 2551-5.
- 139.C. Komninakis and R. D. Wesel, "Iterative Joint Data and Channel Estimation in Correlated Flat Rayleigh Fading," in proceedings of the *7th International Conference on Advances in Communications and Control*, June 28-July 2, 1999, Athens, Greece, pp. 385-390.

- 140.C. Fragouli and R. D. Wesel, "Convolutional Codes and Matrix Control Theory," in proceedings of the *7th International Conference on Advances in Communications and Control*, June 28-July 2, 1999, Athens, Greece, pp. 317-328.
- 141.R. D. Wesel, "Reduced Complexity Trellis Code Transfer Function Computation," in the *Communication Theory Miniconference in conjunction with ICC 99*, June 6-10, 1999, pp. 37-41.
142. C. Fragouli and R. D. Wesel, "Symbol Interleaved Parallel Concatenated Trellis Coded Modulation," in the *Comm. Theory Miniconf. in conj. with ICC 99*, June 6-10, 1999, pp. 42-46.
143. P. Ormeci, D. L. Goeckel, and R. D. Wesel, "Adaptive Bit-Interleaved Coded Modulation for Time-Varying Channels Using Outdated Fading Estimates," in proceedings of the *33rd Annual Conference on Information Sciences and Systems*, 6 pages.
- 144.H. Zou, H. J. Kim, S. Kim, B. Daneshrad, R. Wesel, W. Mangione-Smith, "Equalized GMSK, Equalized QPSK, and OFDM: A Comparative Study for High-speed Wireless Indoor Data Communications," in *1999 Veh. Tech. Conf.*, May 16-20, 1999, Houston, TX, pp.1106-10.
- 145.A. Bernard, X. Liu, R. Wesel and A. Alwan, "Embedded Joint Source-Channel Coding of Speech using Symbol Puncturing in Trellis Code," in the Proceedings of *ICASSP 99*, vol. 5, Phoenix, AZ, March 1999., pp. 2427-2430.
- 146.A. Bernard, X. Liu, R. Wesel and A. Alwan, "Channel Adaptive Joint Source-Channel Coding of Speech," Proc. *32nd Asilomar Conf. on Sig., Sys, and Comp.*, Pac. Grove, CA, Nov. 98, pp. 357-361.
- 147.W. Shi and R. D. Wesel, "The Effect of Mismatch on Decision-Feedback Equalization and Tomlinson-Harashima Precoding," in *Thirty-Second Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, November 2-5, 1998, pp. 1743-1747.
- 148.R. D. Wesel and X. Liu, "Analytic Techniques for Periodic Trellis Codes," in *Thirty-Sixth Annual Allerton Conf. on Communications, Control, and Computing*, Sept. 23-25, 1998, pp. 39-48.
- 149.X. Liu and R. D. Wesel, "Profile Optimal 8-QAM and 32-QAM Constellations," in *Thirty-Sixth Annual Allerton Conf. on Comm., Control, and Computing*, Sept. 23-25, 1998, pp. 136-145.
- 150.W. Shi and R. D. Wesel, "When the Best Decision Feedback Equalizer is a Linear Equalizer," in *Thirty-Sixth Annual Allerton Conference on Control, and Computing*, Sept. 23-25, 1998, pp. 338-39.
- 151.R. D. Wesel, X. Liu, W. Shi, and J. M. Cioffi, "Trellis Codes for Compound Periodic Gaussian Channels," in Proceedings of the *1998 International Symposium on Information Theory*, August 16-21, 1998, Cambridge, MA, pp. 462.
- 152.R. D. Wesel, X. Liu, and W. Shi, "Periodic Symbol Puncturing of Trellis Codes," in proceedings of *Thirty-First Asilomar Conf. on Sig., Sys., and Computers*, Nov. 2-5, 1997, pp. 172-6.
- 153.R. D. Wesel, C. Komninakis, and X. Liu, "Towards Optimality in Constellation Labeling," in proceedings of the *Communication Theory Mini Conference at Globecom 97*, Phoenix, AZ, November 3-8, 1997, pp. 23-27.
- 154.R. D. Wesel, and J. M. Cioffi, "Joint Interleaver and Trellis Code Design," in proceedings of *Globecom 97*, Phoenix, AZ, November 3-8, 1997, pp. 939-943.
- 155.R. D. Wesel, and J. Cioffi, "Fundamentals of Coding for Broadcast OFDM," in *Twenty-Ninth Asilomar Conference on Signals, Systems, and Computers*, October 30, 1995, pp. 2-6.
- 156.R. D. Wesel, and J. Cioffi, "Achievable Rates for Tomlinson-Harashima Precoding," *1995 International Symposium on Information Theory*, September 1995, pp. 399.
- 157.R. D. Wesel, and J. Cioffi, "Precoding and the MMSE-DFE," Invited paper in *Twenty-Eighth Asilomar Conference on Signals, Systems, and Computers*, November 1, 1994, pp. 1144-1148.

158. R. D. Wesel and R. M. Gray, "Bayes risk weighted VQ and Learning VQ," in *Data Compression Conference*, Snowbird, Utah, March 1994, pp. 400-409.
159. T. M. Cover and R. D. Wesel, "A Gambling Estimate of the Rate-Distortion Function for Images," in *Data Compression Conference*, Snowbird, Utah, March 1994 (abstract only).

BOOK CHAPTERS

1. H. Chan, A. Hodjat, J. Shi, R. D. Wesel and I. Verbaauwhede, "Streaming Encryption for a Secure Wavelength and Time Domain Hopped Optical Network," Ch. 14 in *Embedded Cryptographic Hardware* (N. Nedjah and L. Mourelle, eds.). Nova Science Publishers, 2004
2. R. D. Wesel, "Convolutional Encoding" in *Wiley Encyclopedia of Telecommunications*, Edited by John Proakis, John Wiley, 2003.
3. R. D. Wesel, "Error Control" Chapter 6 in *Wireless Multimedia Communications*, Addison Wesley Longman, 1998.

PATENTS

1. R. D. Wesel, K. Vakiliinia, S. Ranganathan, D. Divsalar, Haobo Wang, "High Throughput Communication System". U.S. Patent #10,374,759. August 6, 2019.
2. R. D. Wesel, K. Vakiliinia, S. Ranganathan, D. Divsalar, "High Throughput Communication System". U.S. Patent #9998260. June 12, 2018.
3. R. D. Wesel, M.-C. F. Chang, Y.-M. Chang, A. I. Vila Casado, "Lower Complexity Layered Belief Propagation Decoding LDPC Codes". U.S. Patent #8489957, July 16, 2013.
4. A. I. Vila Casado, W.-Y. Weng, R. D. Wesel, N. Moschini, M. Siti, S. Valle, E. Yeo, "Variable-Rate Low-Density Parity Check Codes with Constant Blocklength". U.S. Patent #7802172, September 21, 2010.
5. I. S. Djokovic, R. D. Wesel, E. J. Infusino, M. K. Tsatsanis, "Multi-channel communication system for multiple input, multiple output processing of an encoded signal". U.S. Patent #7415086, August 19, 2008.
6. G. G. Raleigh, Michael A. Pollak, V. K. Jones, R. D. Wesel, "System and Method for I/Q Trellis Coded Modulation," U.S. Patent #6158041, December 5, 2000.
7. R. D. Wesel and J. M. Cioffi, "Transmission System Using Code Designed for Transmission with Periodic Interleaving," U.S. Patent #6125150, September 26, 2000.
8. R. D. Wesel, "Adaptive Frequency Dependent Compensation for Telephone Channels," U. S. Patent #5280525, January 18, 1994.
9. R. L. Daggett and R. D. Wesel, "Echo Protection Tone Detection and Regeneration for Digital Transmission of Facsimile Calls," U. S. Patent #5216519, June 1, 1993.

APPENDIX B

Materials Considered

In addition to the materials cited in my report, I considered the following:

Patents and File Histories

- PCT Publication WO2006/030478 / U.S. Patent No. 8,775,890 to Yap et al.
- U.S. Patent App. Publication No. 2005/0254508 to Aksu et al.
- U.S. Patent No. 10,833,809 and File History
- U.S. Patent No. 6,865,233 issued to Eriksson, et al.
- U.S. Patent No. 8,468,411 and File History
- U.S. Patent No. 9,094,348 and File History
- U.S. Provisional Patent App. No. 60/792,236
- U.S. Provisional Patent App. No. 60/849,650

Other Prior Art References

- 3GPP TR 25.896 v6.0.0
- 3GPP TS 25.401 v6.0.0
- G.992.3 (ADSL2)
- G.993.1 (VDSL1)
- Interleaved FEC/ARQ coding for QoS multicast over the Internet by Chen et al.
- Wireless Video Transport Using Conditional Retransmission and Low-Delay Interleaving by Aramvith et. al.

Discovery Materials

- Declaration of Dr. Sylvia Hall-Ellis, dated August 27, 2022
- Deposition Transcript of Abha Divine, dated August 4, 2022
- Deposition Transcript of Marcos Tzannes, dated July 13, 2022
- Nokia Defendants' Amended Invalidity Contentions, served July 28, 2022 and all exhibits cited therein
- Nokia Defendants' Amended Invalidity Contentions, served March 23, 2022 and all exhibits cited therein
- Nokia Defendants' Initial Invalidity Contentions, served January 13, 2022 and all exhibits cited therein
- TQ Delta's Objections and Responses to Nokia's Second Set of RFAs, dated August 15, 2022
- TQ Delta's Second Supplemental Objections and Responses to CommScope's First Set of Interrogatories, served August 19, 2022
- TQ Delta's Second Supplemental Objections and Responses to Nokia's Second Set of Interrogatories (Nos. 4-17), served August 19, 2022

Claim Construction Materials

- TQ Delta's Opening Claim Construction Brief (Dkt. 124), filed April 26, 2022 and all exhibits cited therein
- Defendants' Responsive Claim Construction Brief (Dkt. 135), filed May 6, 2022 and all exhibits cited therein
- TQ Delta's Reply Claim Construction Brief (Dkt. 140), filed May 13, 2022 and all exhibits cited therein
- Markman Hearing Transcript (Dkt. 172), dated June 1, 2022
- Claim Construction Memorandum and Order (Dkt. 169), filed June 8, 2022

APPENDIX C

DEPOSITION TESTIMONY OF RICHARD WESEL DURING PREVIOUS FIVE YEARS

TQ Delta, LLC v. ADTRAN Inc. (2019, 2022)
Case 1:14-cv-954-RGA (D. Del.)

Core Wireless Licensing S.A.R.L. v. LG Electronics, Inc. (2015, 2018) Case
No. 6:14-cv-911-JRG-RSP (E.D. Tex.) Lead Case
Case No. 6:14-cv-912-JRG-RSP (E.D. Tex.) Consolidated

Realtime Data LLC d/b/a IXO, v. Echostar Corporation, Hughes Network Systems, LLC
(2018)
Case No. 6:17-cv-00084-JDL

Realtime Data LLC d/b/a IXO, v. Rackspace US, INC.; Netapp, Inc.; and Solidfire, LLC
(2017)
Case No. 6:16-cv-00961-RWS-JDL (E.D. Tex.)

TQ Delta, LLC v. Comcast Cable Communications LLC, COXCOMM LLC and COX
Communications Inc., DirecTV, LLC, Time Warner Cable, Inc. and Time Warner Cable
Enterprises, LLC, Verizon Services Corp. (2016)
Case Nos. 15-611, 15-612, 15-613, 15-614, 15-615, 15-616 (RGA) (Delaware)

Core Wireless Licensing S.a.r.l. v. Apple Inc. (2016)
Case 5:15-cv-05008-NC (N.D. Cal.)

TRIAL TESTIMONY OF RICHARD WESEL DURING PREVIOUS FIVE YEARS

Core Wireless Licensing S.A.R.L. v. LG Electronics, Inc. (2016, 2019) Case
No. 6:14-cv-911-JRG-RSP (E.D. Tex.) Lead Case
Case No. 6:14-cv-912-JRG-RSP (E.D. Tex.) Consolidated

Core Wireless Licensing S.a.r.l. v. Apple Inc. (2016)
Case 5:15-cv-05008-NC (N.D. Cal.)

APPENDIX D

**UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION**

TQ DELTA, LLC,

Plaintiff,

v.

**COMMSCOPE HOLDING COMPANY, INC.,
COMMSCOPE INC., ARRIS US HOLDINGS,
INC., ARRIS SOLUTIONS, INC., ARRIS
TECHNOLOGY, INC., and ARRIS
ENTERPRISES, LLC**

Defendants.

CIV. A. NO. 2:21-CV-310-JRG
(Lead Case)

TQ DELTA, LLC,

Plaintiff,

v.

**NOKIA CORP., NOKIA SOLUTIONS AND
NETWORKS OY, and NOKIA OF AMERICA
CORP.,**

Defendants.

CIV. A. NO. 2:21-CV-309-JRG
(Member Case)

DECLARATION OF SYLVIA HALL-ELLIS, PH.D.

I, Sylvia D. Hall-Ellis, Ph.D., declare as follows:

I. INTRODUCTION

1. My name is Sylvia D. Hall-Ellis. I have been retained as an expert by Defendants Nokia of America Corporation, Nokia Corporation, and Nokia Solutions and Networks, Oy (collectively, “Nokia”).

2. I have written this Declaration at the request of Nokia to provide my expert opinion regarding the public availability of two publications, identified below. My Declaration sets forth my opinions in detail and provides the basis for my opinions regarding the public availability of these publications.

3. I reserve the right to supplement or amend my opinions, and bases for them, in response to any additional evidence, testimony, discovery, argument, and/or other additional information that may be provided to me after the date of this Declaration.

4. I am being compensated for my time spent working on this matter at my normal consulting rate of \$325 per hour, plus reimbursement for any additional reasonable expenses. My compensation is not in any way tied to the content of this report, the substance of my opinions, or the outcome of this litigation. I have no other interests in this proceeding or with any of the parties.

5. All of the materials that I considered and relied upon are discussed explicitly in this Declaration.

II. QUALIFICATIONS

6. I am currently an Adjunct Professor in the School of Information at San José State University in San José, California. I obtained a Master of Library Science from the University of North Texas in 1972 and a Ph.D. in Library and Information Science from the University of Pittsburgh in 1985. Over the last fifty years, I have held various positions in the field of library and information resources. I was first employed as a librarian in 1966 and have been involved in the field of library sciences since, holding numerous positions.

7. I am a member of the American Library Association (ALA) and its Association for Library Collections & Technical Services (ALCTS) Division, and I served on the Committee on Cataloging: Resource and Description (which wrote the new cataloging rules) and as the chair of the Committee for Education and Training of Catalogers and the Competencies and Education for a Career in Cataloging Interest Group. I also served as the Chair of the ALCTS Division's Task Force on Competencies and Education for a Career in Cataloging. Additionally, I have served as the Chair for the ALA Office of Diversity's Committee on Diversity, as a member of the REFORMA National Board of Directors, as a member of the Editorial Board for the ALCTS premier cataloging journal, *Library Resources and Technical Services*, and as a Co-Chair for the Library Research Round Table of the American Library Association.

8. I have also given over one hundred presentations in the field, including several on library cataloging systems and Machine-Readable Cataloging (“MARC”) standards. My current research interests include library cataloging systems, metadata, and organization of electronic resources.

9. My full curriculum vitae is attached hereto as **Exhibit A**.

III. PRELIMINARIES

A. Scope of Declaration and Legal Standards

10. I am not an attorney and will not offer opinions on the law. I am, however, rendering my expert opinion on the authenticity of the documents referenced herein and on when and how each of these documents was disseminated or otherwise made available to the extent that persons interested and ordinarily skilled in the subject matter or art, exercising reasonable diligence, could have located the documents before the dates discussed below with respect to the specific documents.

11. I am informed by counsel that a printed publication qualifies as publicly accessible as of the date it was disseminated or otherwise made available such that a person interested in and ordinarily skilled in the relevant subject matter could locate it through the exercise of ordinary diligence.

12. While I understand that the determination of public accessibility under the foregoing standard rests on a case-by-case analysis of the facts particular to an

individual publication, I also understand that a printed publication is rendered “publicly accessible” if it is cataloged and indexed by a library such that a person interested in the relevant subject matter could locate it (*i.e.*, I understand that cataloging and indexing by a library is sufficient, though there are other ways that a printed publication may qualify as publicly accessible). One manner of sufficient indexing is indexing according to subject matter category. I understand that the cataloging and indexing by a single library of a single instance of a particular printed publication is sufficient, even if the single library is in a foreign country. I understand that, even if access to a library is restricted, a printed publication that has been cataloged and indexed therein is publicly accessible so long as a presumption is raised that the portion of the public concerned with the relevant subject matter would know of the printed publication. I also understand that the cataloging and indexing of information that would guide a person interested in the relevant subject matter to the printed publication, such as the cataloging and indexing of an abstract for the printed publication, is sufficient to render the printed publication publicly accessible.

13. I understand that routine business practices, such as general library cataloging and indexing practices, can be used to establish an approximate date on which a printed publication became publicly accessible.

B. Persons of Ordinary Skill in the Art

14. I am told by counsel that the subject matter of this proceeding generally relates to packet transmission and retransmission.

15. I have been informed by counsel that a “person of ordinary skill in the art at the time of the invention” (POSITA) is a hypothetical person who is presumed to be familiar with the relevant field and its literature at the time of the inventions. This hypothetical person is also a person of ordinary creativity, capable of understanding the scientific principles applicable to the pertinent field.

16. I am told by counsel that a POSITA as of April 12, 2006 , would have possessed a bachelor’s degree in electrical or computer engineering, or the equivalent, and at least 5-6 years of experience in telecommunications or a related field; a master’s degree in electrical or computer engineering, or the equivalent, and at least 2-3 years of experience in telecommunications or a related field; or a Ph.D. in electrical or computer engineering, or the equivalent, with at least 1-2 years of experience in telecommunications or a related field. I have been further informed by counsel that a POSITA would have been familiar with and able to understand the information known in the art relating to these fields, including the publications discussed in this Declaration. I have been further informed by counsel that a POSITA would have been familiar with and able to understand the information known in the art relating to these fields, including the publication discussed in this

Declaration.

C. Use of Authoritative Databases and Indexing

17. In preparing this report, I used authoritative databases, such as the OCLC bibliographic database, the Library of Congress Online Catalog, the *IEEE Xplore* database,¹ *ResearchGate*,² and *Semantic Scholar*,³ to confirm citation details of the publication discussed.

18. A researcher may discover material relevant to his or her topic in a variety of ways. One common means of discovery is to search for relevant information in an index of periodical and other publications. Having found relevant material, the researcher will then normally obtain it online, look for it in libraries, or purchase it from the publisher, a bookstore, a document delivery service, or other provider. Sometimes, the date of a document's public accessibility will involve both indexing and library date information.

19. Indexing services use a wide variety of controlled vocabularies to provide subject access and other means of discovering the content of documents. The formats in which these access terms are presented vary from service to service.

20. Online indexing services and digital repositories commonly provide

¹ <https://ieeexplore.ieee.org/Xplore/home.jsp>

² <https://www.researchgate.net/>

³ <https://www.semanticscholar.org>

bibliographic information, abstracts, and full-text copies of the indexed publications, along with a list of the documents cited in the indexed publication. These services also often provide lists of publications that cite a given document. A citation of a document is evidence that the document was publicly available and in use by researchers no later than the publication date of the citing document.

D. Summary of Opinions

21. I am informed by counsel that the priority date for the patents at issue is April 12, 2006. As I will explain below, it is my opinion that the printed publication discussed in my Declaration was publicly accessible more than one year before the April 12, 2006, priority date.

IV. LIBRARY CATALOGING PRACTICES

A. MARC Records and OCLC

22. I am fully familiar with the library cataloging standard known as the MARC standard, which is an industry-wide standard method of storing and organizing library catalog information. MARC was first developed in the 1960's by the Library of Congress. A MARC-compatible library is one that has a catalog consisting of individual MARC records for works made available at that library.

23. Since at least the early 1970s and continuing to the present day, MARC has been the primary communications protocol for the transfer and storage of

bibliographic metadata in libraries.⁴ As explained by the Library of Congress:

You could devise your own method of organizing the bibliographic information, but you would be isolating your library, limiting its options, and creating much more work for yourself. Using the MARC standard prevents duplication of work and allows libraries to better share bibliographic resources. Choosing to use MARC enables libraries to acquire cataloging data that is predictable and reliable. If a library were to develop a “home-grown” system that did not use MARC records, it would not be taking advantage of an industry-wide standard whose primary purpose is to foster communication of information.

Using the MARC standard also enables libraries to make use of commercially available library automation systems to manage library operations. Many systems are available for libraries of all sizes and are designed to work with the MARC format. Systems are maintained and improved by the vendor so that libraries can benefit from the latest advances

⁴ A complete history of the development of MARC can be found in *MARC: Its History and Implications* by Henrietta D. Avram (Washington, DC: Library of Congress, 1975) and available online from the Hathi Trust (<https://babel.hathitrust.org/cgi/pt?id=mdp.39015034388556;view=1up;seq=1>; last visited July 25, 2022).

in computer technology. The MARC standard also allows libraries to replace one system with another with the assurance that their data will still be compatible.

Why Is a MARC Record Necessary? LIBRARY OF CONGRESS, <http://www.loc.gov/marc/umb/um01to06.html#part2> (last visited July 25, 2022).

24. Thus, almost every major library in the world is MARC-compatible. See, e.g., *MARC Frequently Asked Questions (FAQ)*, LIBRARY OF CONGRESS, <https://www.loc.gov/marc/faq.html> (last visited July 25, 2022) (“MARC is the acronym for MACHine-Readable Cataloging. It defines a data format that emerged from a Library of Congress-led initiative that began nearly fifty years ago. It provides the mechanism by which computers exchange, use, and interpret bibliographic information, and its data elements make up the foundation of most library catalogs used today.”). MARC is the ANSI/NISO Z39.2-1994 standard (reaffirmed in 2016) for Information Interchange Format. The full text of the standard is available from the Library of Congress at <http://www.loc.gov/marc/bibliographic/> (last visited July 25, 2022).

25. A MARC record comprises several fields, each of which contains specific data about the work. Each field is identified by a standardized, unique, three-digit code corresponding to the type of data that follow. See, e.g., <http://www.loc.gov/marc/umb/um07to10.html> (last visited July 25, 2022);

<http://www.loc.gov/marc/bibliographic/> (last visited July 25, 2022). For example, a work's title is recorded in field 245, the primary author of the work is recorded in field 100, a work's International Standard Book Number ("ISBN") is recorded in field 020, a work's International Standard Serial Number ("ISSN") is recorded in field 022, and the publication date is recorded in field 260 under the subfield "c." *Id.*⁵ If a work is a periodical, then its publication frequency is recorded in field 310, and the publication dates (e.g., the first and last publication) are recorded in field 362, which is also referred to as the enumeration/chronology field. *See* <http://www.loc.gov/marc/bibliographic/bd3xx.html> (last visited July 25, 2022).⁶

⁵ In some MARC records, field 264 is used rather than field 260 to record publication information. *See* <http://www.loc.gov/marc/bibliographic/bd264.html> (last visited July 25, 2022) ("Information in field 264 is similar to information in field 260 (Publication, Distribution, etc. (Imprint)). Field 264 is useful for cases where the content standard or institutional policies make a distinction between functions").

⁶ Upwards of two-thirds to three-quarters of book sales to libraries come from a jobber or wholesaler for online and print resources. These resellers make it their business to provide books to their customers as fast as possible, often providing turnaround times of only a single day after publication. Libraries purchase a significant portion of the balance of their books directly from publishers themselves, which provide delivery on a similarly expedited schedule. In general, libraries make these purchases throughout the year as the books are published and shelve the books as soon thereafter as possible in order to make the books available

26. The library that initially created the MARC record is reflected in field 040 in subfield “a” with that library’s unique library code. *See, e.g.,* <http://www.loc.gov/marc/umb/um07to10.html> (last visited July 25, 2022); <http://www.loc.gov/marc/bibliographic/> (last visited July 25, 2022). Once a MARC record for a particular work is originally created by one library, other libraries can use that original MARC record to then create their own MARC records for their own copies of the same work. These other libraries may modify or add to the original MARC record as necessary to reflect data specific to their own copies of the work. However, the library that created the original MARC record would still be reflected in these modified MARC records (corresponding to other copies of the same work at other libraries) in field 040, subfield “a”. The modifying library (or libraries) is reflected in field 040, subfield “d”. *See* <http://www.loc.gov/marc/bibliographic/bd040.html> (last visited July 25, 2022).

27. I consulted the Directory of OCLC Libraries (<http://www.oclc.org/contacts/libraries.en.html>; last visited July 25, 2022) in order to identify the institution that created or modified the MARC record. Moreover, when viewing the MARC record online via Online Computer Library Center’s (“OCLC”) bibliographic database, which I discuss further below, hovering over a

to their patrons. Thus, books are generally available at libraries across the country within just a few days of publication.

library code in field 040 with the mouse reveals the full name of the library. I also used this method of “mousing over” the library codes in the OCLC database to identify the originating and modifying libraries for the MARC records discussed in this report.

28. MARC records also include one or more fields that show information regarding subject matter classification. For example, 6XX fields are termed “Subject Access Fields.” See <http://www.loc.gov/marc/bibliographic/bd6xx.html> (last visited July 25, 2022). Among these, for example, is the 650 field; this is the “Subject Added Entry – Topical Term” field. See <http://www.loc.gov/marc/bibliographic/bd650.html> (last visited July 25, 2022). The 650 field is a “[s]ubject added entry in which the entry element is a topical term.” *Id.* These entries “are assigned to a bibliographic record to provide access according to generally accepted thesaurus-building rules (e.g., *Library of Congress Subject Headings* (LCSH), *Medical Subject Headings* (MeSH)).” *Id.*

29. Further, MARC records can include call numbers, which themselves contain a classification number. For example, a MARC record may identify a 050 field, which is the “Library of Congress Call Number.” See <http://www.loc.gov/marc/bibliographic/bd050.html> (last visited July 25, 2022). A defined portion of the Library of Congress Call Number is the classification number, and “source of the classification number is *Library of Congress Classification* and

the *LC Classification-Additions and Changes*.” *Id.* Thus, the 050 field may be used to show information regarding subject matter classification.

30. Each item in a library has a single classification number. A library selects a classification scheme (e.g., the Library of Congress Classification scheme just described or a similar scheme such as the Dewey Decimal Classification scheme) and uses it consistently. When the Library of Congress assigns the classification number, it appears as part of the 050 field, as discussed above. For MARC records created by libraries other than the Library of Congress (e.g., a university library or a local public library), the classification number may appear in a 09X (e.g., 090) field. See <http://www.loc.gov/marc/bibliographic/bd09x.html> (last visited July 25, 2022).

31. When a MARC-compatible library acquires a work, it creates a MARC record for its copy of the work in its computer catalog system in the ordinary course of its business. This MARC record (for the copy of a work available at the particular library) may be later accessed by researchers in a number of ways. For example, many libraries, including the Library of Congress, make their MARC records available through their website. As an example, the MARC record for the copy of *The Unlikely Spy*, by Daniel Silva,⁷ available at the Library of Congress can be

⁷ *The Unlikely Spy* is a 1996 novel written by Daniel Silva, who happens to be one of my favorite authors.

viewed through the Library of Congress website, at <https://catalog.loc.gov/vwebv/staffView?searchId=20265&recPointer=1&recCount=25&bibId=2579985> (last visited July 25, 2022). One could, of course, always physically visit the library at which the work is available, and request to see that library's MARC record for the work. Moreover, members of the Online Computer Library Center ("OCLC") can access the MARC records of other member institutions through OCLC's online bibliographic database, as I explain further below.

32. The OCLC was created "to establish, maintain and operate a computerized library network and to promote the evolution of library use, of libraries themselves, and of librarianship, and to provide processes and products for the benefit of library users and libraries, including such objectives as increasing availability of library resources to individual library patrons and reducing the rate of rise of library per-unit costs, all for the fundamental public purpose of furthering ease of access to and use of the ever-expanding body of worldwide scientific, literary and educational knowledge and information."⁸ Among other services, OCLC and

⁸ Third Article, Amended Articles of Incorporation of OCLC Online Computer Library Center, Incorporated (available at <https://www.oclc.org/content/dam/oclc/membership/articles-of-incorporation.pdf>; last visited July 25, 2022).

its members are responsible for maintaining the WorldCat database (<http://www.worldcat.org/>; last visited July 25, 2022), used by independent and institutional libraries throughout the world. All libraries that are members of OCLC are MARC-compatible. *See, e.g.,* https://help.oclc.org/Metadata_Services/OCLC-MARC_records/About_OCLC-MARC_records (last visited July 25, 2022) (“OCLC-MARC records describes records produced since November 1993.”).

33. When an OCLC member institution acquires a publication, like the other MARC-compatible libraries discussed above, it creates a MARC record for this publication in its computer catalog system in the ordinary course of its business. MARC records created at the Library of Congress were tape-loaded into the OCLC database through a subscription to MARC Distribution Services daily or weekly. Once the MARC record is created by a cataloger at an OCLC member library or can be tape-loaded from the Library of Congress, the MARC record is then made available to any other OCLC members online, and thereby made available to the public. Accordingly, once the MARC record is created by a cataloger at an OCLC member library or can be tape-loaded from the Library of Congress, any publication corresponding to the MARC record has been cataloged and indexed according to its subject matter such that a person interested in that subject matter could, with reasonable diligence, locate and access the publication through any library with access to the OCLC bibliographic database or through the Library of Congress.

34. *Fields 008 and 005 in MARC Records as Indicators of Public Accessibility.* When a MARC-compatible library creates an original MARC record for a work, the library records the date of creation of that MARC record in **field 008**, characters 00 through 05, in the ordinary course of its business. See <http://www.loc.gov/marc/bibliographic/bd008a.html> (last visited July 25, 2022). For OCLC member institutions that use OCLC software to create original MARC records, the date of creation in field 008 is automatically supplied by the OCLC software. The MARC record creation date in field 008 thus reflects the date on which, or shortly after which, a work was first acquired and cataloged by the library that created the original MARC record.

35. When other MARC-compatible libraries subsequently acquire their own copies of the same work, as mentioned, they create MARC records in their own computer catalog systems for their copies in the ordinary course of business.⁹ They may use a MARC record previously created for that work (by another MARC-compatible library) to create their own MARC records for their own copies of that

⁹ Initial contributions to the bibliographic database for a work are called “master records.”

same work.¹⁰ The previously created MARC record used by subsequently-acquiring libraries to create MARC records for their own copies may be obtained through the OCLC bibliographic database, as described above. If, when creating a MARC record to represent its own copy of the work, the subsequently-acquiring library uses the master MARC record in its original form, the subsequently-acquiring library cannot reenter data into the 008 field; therefore, the date in the 008 field will continue to reflect the date the MARC record was initially created by the originating library. On the other hand, if the subsequently-acquiring library modifies the previously created MARC record when creating its own MARC record for its own copy of the work, the subsequently-acquiring library may enter into the 008 field of its own MARC record the date its own MARC record was created.¹¹ But the library that created the original MARC record used by the subsequently-acquiring library would still be reflected in the MARC record of the subsequently-acquiring library in field

¹⁰ When a local library uses a master record in OCLC and produces (or downloads) it to the in-house system, the three-character symbol for the subsequent library is added to the holdings for the work.

¹¹ This practice is not required by, but is nevertheless consistent with, the MARC standard. Many MARC records exist today whose 008 fields indicate when the first original MARC record for a work was created, rather than when a derivative record was created based on the original MARC record by a subsequently-acquiring library for its own computer catalog system.

040, subfield “a”. Thus, the work identified by any MARC record possessed by any MARC-compatible library would have been accessible to the public at least as of the date shown in the 008 field, or shortly thereafter, either from the library that possesses the MARC record itself, or from the originating library indicated in field 040, subfield “a”. As discussed, a MARC-compatible library in the ordinary course of its business creates a MARC record in its own catalog system for a work when it acquires a copy of that work.

36. Moreover, when a MARC record is created by a library for its own copy of a work, **field 005** is automatically populated with the date that MARC record was created in year, month, day format (YYYYMMDD). See <http://www.loc.gov/marc/bibliographic/bd005.html> (last visited July 25, 2022).¹² Thereafter, the library’s computer system may automatically update the date in field 005 every time the library updates the MARC record (*e.g.*, to reflect that an item has been moved to a different shelving location within the library). *Id.*¹³ Thus, the work identified by any MARC record possessed by any MARC-compatible library would

¹² Some of the newer library catalog systems also include hour, minute, second (HHMMSS).

¹³ Field 005 is visible when viewing a MARC record via an appropriate computerized interface. But when a MARC record is printed directly to hardcopy from the OCLC database, the “005” label is not shown. The date in the 005 field instead appears next to the label “Replaced.”

have been accessible to the public at least as of the date shown in the 005 field, or shortly thereafter, from the library that possesses the MARC record itself. As noted, because the 005 field may be updated each time the library updates its MARC record, the work identified by the MARC record may, in fact, have been accessible to the public from that library much earlier than the date indicated in the 005 field.

37. Based on my personal experience as a professional librarian using the MARC and OCLC resources, it has been my experience that both of these resources were continuously operational and available since at least 1992. Indeed, in the course of my work, I have extensively used both of these resources over the past 30+ years, and I have consistently found the information contained within these resources to be complete and reliable. I have never found the date of accessibility as indicated in fields 008, 005, or 955 to be incorrect. And in only a minute number of cases have I found any errors at all in these records – none of which affected my ability to render an accurate opinion as to accessibility, indexing, or subject headings.

38. *Journals.* In addition to creating records for books, OCLC also hosts records for serial publications. Serial publications have the same collective title but are intended to continue indefinitely with enumeration such as a volume or issue number (*e.g.*, magazines, journals, etc.). In the OCLC bibliographic database, the first issue of the serial publication is typically cataloged, but the date is left open-ended with the use of a punctuation mark such as a dash. Individual issues are not

cataloged according to the current national cataloging rules outlined in *RDA: Resource and Access*.

39. MARC records for serial publication in OCLC represent the entire run of the serial title. With knowledge of the first issue published, future issues can be predicted. In my extensive professional experience, is it highly unusual for a library to stop collecting and shelving a serial publication prior to the end of its publication run. If a subscription to a serial publication ends its run or is cancelled before the end of its run, the library will denote that it has stopped receiving new volumes by filling in the end date in the local MARC record.

40. The handling of printed journal subscriptions is shown on the cover or the first few pages of individual issues. As was the best practice among libraries, issues arrived at a central facility and were immediately received, verified as part of a subscription, checked in, and stamped or labeled with the institution's name and date. Determining that the issue was part of the library subscription ensured that the entire set of publications for the year had been received so that they could be professionally bound and retained. This process also verified that all of the published issues arrived so that the library staff did not have to request or claim an issue that did not arrive as expected. In large public libraries with branches and multi-campus libraries within academic institutions, the journals were sorted and delivered to the subscribing unit. The new issue was placed in the public area; the

older issue was stored so that it remained available.

41. The foregoing process has been standard library practice longer than I have been working in the profession. I first learned the steps in the process in the late 1970s and later supervised it. Although the checking in process has become automated and now links electronically to holdings records for the MARC record for each serial title, the manual placement of a stamp or label and placing the issue in a public area has not changed for 50 years. Unless I note otherwise below in reference to a specific serial publication, it is my expert opinion that this standard protocol was followed for the serial publication discussed below.

V. PUBLICATION IN THIS PROCEEDING

A. “Interleaved FEC/ARQ Coding for QoS Multicast Over the Internet” by Donghui Chen, Bo Rong, N. Shayan, M. Bennani, J. Cabral, M. Kadoch, and A. K. Elhakeem [Exhibit 1001]

42. Attached hereto as Exhibit 1001 is a copy of the journal article, “Interleaved FEC/ARQ Coding for QoS Multicast Over the Internet” by Donghui Chen, Bo Rong, N. Shayan, M. Bennani, J. Cabral, M. Kadoch, and A. K. Elhakeem (hereafter “Chen”). The Chen article was published in Volume 29, Number 1 of the bilingual (English and French) *Canadian Journal of Electrical and Computer Engineering* and was found in the Linda Hall Library of Science, Engineering & Technology (Kansas City, Missouri). The Chen article appears beginning on page 159 of this issue dated July 2004. Exhibit 1001 is a true and correct copy of the

Chen article (pages 159-166). Specifically, the text of the article is complete; no pages are missing, and the text on each page appears to flow seamlessly from one page to the next; further, there are no visible alterations to the document. Exhibit 1001 was found within the custody of a library – a place where an authentic copy of this journal would likely be. Exhibit 1001 is a true and correct copy in a condition that creates no suspicion about its authenticity.

43. Attached hereto as Attachment 1a is a true and correct copy of the MARC record for the *Canadian Journal of Electrical and Computer Engineering* in the Linda Hall Library of Science, Engineering & Technology online catalog. The library ownership is indicated by the presence of the library's code (LHL) in field 049. The library continues to update and enhance the MARC record to meet current cataloging rules. The most recent enhancement to Attachment 1a occurred on November 23, 2020, as shown in field 005 ("20201123"). I personally identified and retrieved the MARC record that is Attachment 1a.

44. Based on finding a print copy of Exhibit 1001 in the Linda Hall Library of Science, Engineering & Technology and MARC record in its online library catalog attached as Attachment 1a, it is my opinion that the Chen article published in the *Canadian Journal of Electrical and Computer Engineering* was publicly available on or shortly after July 1, 2004, as shown in *Semantic Scholar*. Attachment 1a also shows that Exhibit 1001 was catalogued with four descriptor terms reading

“Electrical engineering \$z Canada \$v Periodicals” (see Attachment 1b, Library of Congress subject heading sh85041666, Attachment 1c, Library of Congress name authority n79007233, and Attachment 1d, Library of Congress subject heading sh85099890), “Computer engineering \$z Canada \$v Periodicals” (see Attachment 1e, Library of Congress subject heading sh85029495, Attachment 1c, Library of Congress name authority n79007233, and Attachment 1d, Library of Congress subject heading sh85099890), “Electrical engineering \$v Periodicals” (see Attachment 1f, Library of Congress subject heading sh2008102896), and “Computer engineering \$v Periodicals” (see Attachment 1g, Library of Congress subject heading sh2009121194) in the 650 fields.

45. As noted in the holdings information (362 field), the Linda Hall Library of Science, engineering & Technology has received the *Canadian Journal of Electrical and Computer Engineering* since 1976.¹⁴ In view of the MARC record for Exhibit 1001, the Chen article was publicly on or shortly after July 1, 2004, because the serial title had been received, cataloged, and indexed in the Linda Hall Library of Science, Engineering & Technology and made part of its online catalog database.

¹⁴ The publication was originally titled *Canadian Electrical Engineering Journal* (January 1976-January 1988) and adopted the current title in 1988. The publication ceased operations in 2017.

46. Attached hereto as Attachment 1h is a true and correct copy of the MARC record for the *Canadian Journal of Electrical and Computer Engineering* obtained from the OCLC bibliographic database. As previously noted, the library that created the record is recorded in field 040 with a unique library code. For Attachment 1h, that library code is “LRU,” which means that the MARC record for this serial was cataloged in the Tulane University Libraries (New Orleans, Louisiana). As can be seen in the “Entered” field in the MARC record for this exhibit, a cataloger in the Tulane University Libraries created OCLC record number 18830717 on December 2, 1988, as shown in the “Entered” field (“19881202”). The library continues to update this MARC record and enhanced the MARC record to meet current cataloging rules. The most recent enhancement to Attachment 1h occurred on September 28, 2020, as shown in the “Replaced” field (“20200928”). I personally identified and retrieved the MARC record that is Attachment 1h. The “BLvl” entry in Attachment 1h is “s,” which indicates that the *Canadian Journal of Electrical and Computer Engineering* is a serial publication. Field 310 of Attachment 1g reads “Bimonthly, \$b <1999-.” Accordingly, the MARC record for Exhibit 1001 corresponds to the *Canadian Journal of Electrical and Computer Engineering* from the time the serial title began using the current title and includes the date of the Chen article.

47. Attachment 1h includes an entry in field 050 (“TK1 \$b .C34”)—as

described above, a subject matter classification number consistent with the Library of Congress classification system (analogous to the Dewey Decimal classification system) and an entry in field 082 (“621.3”)—as described above, a subject matter classification number consistent with the Dewey Decimal classification system. Attachment 1h further includes two descriptor terms reading “Electrical engineering \$v Periodicals” (see Attachment 1f, Library of Congress subject heading sh2008102896), and “Computer engineering \$v Periodicals” (see Attachment 1g, Library of Congress subject heading sh2009121194) in the 650 fields. Thus, as of its cataloging, the publication corresponding to the MARC record attached hereto as Attachment 1h was indexed according to its subject matter by virtue of at least three independently sufficient classifications: the field 050 entry, the field 082 entry, and the field 650 entries. As of December 2, 1988, the MARC record attached hereto as Attachment 1h was accessible through any library with access to the OCLC bibliographic database or the online catalog at a library that subscribed to the serial, which means that the corresponding publication was publicly available on or before that same date through any library with access to the OCLC bibliographic database or through an individual library.

48. Attachment 1h indicates that the *Canadian Journal of Electrical and Computer Engineering* as cataloged at the Tulane University Libraries is currently available from 86 libraries. In view of the above, this issue of the *Canadian Journal*

of Electrical and Computer Engineering was publicly available on or shortly after July 1, 2004, because by that date it had been cataloged and indexed at the Tulane University Libraries, made part of the OCLC bibliographic database, and received at the Linda Hall Library of Science, Engineering & Technology. For these reasons, it is my opinion that Exhibit 1001 was published and accessible to the public on or shortly after July 1, 2004.

B. “Wireless Video Transport Using Conditional Retransmission and Low-Delay Interleaving” by Supavadee Aramvith, S., Chia-Wen Lin, Sumit Roy, and Ming-Ting Sun [Exhibit 1002]

49. Attached hereto is Exhibit 1002, which I understand is being submitted along with my Declaration, is a copy of a conference paper published in the five-volume set titled *2001 IEEE International Symposium on Circuits and Systems* found in the Linda Hall Library of Science, Engineering & Technology (Kansas City, Missouri). The conference paper titled “Wireless Video Transport Using Conditional Retransmission and Low-Delay Interleaving” by Supavadee Aramvith, S., Chia-Wen Lin, Sumit Roy, and Ming-Ting Sun (hereafter “Aramvith”) appears beginning on page 101 in volume 5. The 2001 IEEE International Symposium on Circuits and Systems was held on May 6-9, 2001, in Sydney, New South Wales, Australia. Exhibit 1002 is a true and correct copy of the Aramvith conference paper (volume 5, pages 101-104). I obtained this copy of the conference paper from the Linda Hall Library of Science, Engineering & Technology. Specifically, the text of

the Aramvith conference paper is complete; no pages are missing, and the text on each page appears to flow seamlessly from one page to the next; further, there are no visible alterations to the document. Exhibit 1002 was found within the custody of a library – a place where, if authentic, a copy of this conference proceedings volume would likely be. Exhibit 1002 is a true and correct copy in a condition that creates no suspicion about authenticity.

50. The Aramvith conference paper is also available in the *IEEE Xplore* database,¹⁵ and the digital repositories *ResearchGate*¹⁶ and *Semantic Scholar*.¹⁷

51. Attached hereto as Attachment 2a is a true and correct copy of the MARC record for this conference proceedings five-volume set from the Linda Hall Library of Science, Engineering & Technology online catalog. The library ownership is indicated by the presence of the library's code (LHL) in the 049 field. The library continues to update this MARC record and enhanced the MARC record to meet current cataloging rules. The most recent enhancement to the MARC record occurred on March 20, 2002, as shown in field 005 ("20020320"). I personally

¹⁵ <https://ieeexplore.ieee.org/document/1013860>

¹⁶

https://www.researchgate.net/publication/3308410_Wireless_video_transport_using_conditional_retransmission_and_low-delay_interleaving

¹⁷ <https://www.semanticscholar.org/paper/Wireless-video-transport-using-conditional-and-Aramvith-Lin/b4f6ee73d4d5fd6c4a9bbda7b5f6c6507c369348>

identified and retrieved the library catalog record which is Attachment 2a.

52. Based on finding a print copy of Exhibit 1002 in the Linda Hall Library of Science, Engineering & Technology and MARC record in its online library catalog attached as Attachment 2a, it is my opinion that the five-volume set titled *2001 IEEE International Symposium on Circuits and Systems* was publicly available at the Linda Hall Library of Science, Engineering & Technology as of April 26, 2001, as shown in field 008 (“010426”).

53. Attached hereto as Attachment 2b is a true and correct copy of the MARC record for the five-volume set titled *2001 IEEE International Symposium on Circuits and Systems* obtained from the OCLC bibliographic database. I personally identified and retrieved the MARC record that is Attachment 2b. As previously noted, the library that created the record is recorded in field 040 with a unique library code. For Attachment 2b, that library code is “NSA,” which means that the MARC record for this conference proceedings volume was cataloged at the National Security Agency Library (Washington, DC). As can be seen in the “Entered” field in the MARC record for this exhibit, a cataloger at the National Security Library created OCLC record number 46823428 on April 26, 2001, as shown in the “Entered” field (“20010426”). The library continues to update this MARC record the MARC record to meet current cataloging rules. The most recent enhancement to the MARC record occurred on June 17, 2022, as shown in the “Replaced” field

(“20220617”).

54. Attachment 2b further includes an entry in field 050 (“TK7801 \$b .I245 2001”)—as described above, a subject matter classification number consistent with the Library of Congress classification system (analogous to the Dewey Decimal classification system). Attachment 2b also shows that the volume was catalogued with five descriptor terms reading “Electronics \$v Congresses” (see Attachment 2c, Library of Congress subject heading sh2008102883), “Signal processing \$x Digital techniques \$v Congresses” (see Attachment 2d, Library of Congress subject heading sh2008111654), “Electronic circuits \$v Congresses” (see Attachment 2e, Library of Congress subject heading sh2008102920), “Neural networks (Computer science) \$v Congresses” (Attachment 2f, Library of Congress subject heading sh2008108385), and “Integrated circuits \$x Very large scale integration \$v Congresses” (Attachment 2g, Library of Congress subject heading sh2008104263) in the 650 fields. Thus, as of its cataloging, the publication corresponding to the MARC record attached hereto as Attachment 2b was indexed according to its subject matter by virtue of at least two independently sufficient classifications: the field 050 entry and the field 650 entries. Further, as of April 26, 2001, the MARC record attached hereto as Attachment 2b was accessible through any library with access to the OCLC bibliographic database or the online catalog at a library that acquired the conference proceedings five-volume set, which means that the corresponding publication was

publicly available on or before that same date through any library with access to the OCLC bibliographic database or through an individual library.

55. Attachment 2b indicates that the five-volume set titled *2001 IEEE International Symposium on Circuits and Systems* as cataloged at the National Security Agency Library is currently available from 35 libraries. In view of the above, the five-volume set titled *2001 IEEE International Symposium on Circuits and Systems* was publicly available on or shortly after April 26, 2001, because by that date it had been received, cataloged, and indexed at the National Security Agency Library, made part of the OCLC bibliographic database, and received at the Linda Hall Library of Science, Engineering & Technology. For these reasons, it is my opinion that Exhibit 1002 was published and publicly accessible on or shortly after April 26, 2001.

VI. CONCLUSION

56. In signing this Declaration, I recognize that the Declaration will be filed as evidence in a case before the District Court for the Eastern District of Texas. I also recognize that I may be subject to cross-examination in the case and that cross-examination will take place within the United States. If cross-examination is required of me, I will appear for cross-examination within the United States during the time allotted for cross-examination.

57. I hereby declare that all statements made herein of my own knowledge

are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Dated: August 27, 2022

Respectfully submitted,

A handwritten signature in blue ink, reading "Sylvia D. Hall-Ellis". The signature is written in a cursive style with a large initial "S".

Sylvia D. Hall-Ellis, Ph.D.

EXHIBIT A-1

SUMMARY CLAIM CHART A-1
U.S. Patent No. 8,468,411 versus ITU-T G.992.3 and G.993.1

Asserted Claims of the '411 Patent	Citations to ITU-T G.992.3 and G.993.1
Claim 10	
[10.a] A transceiver capable of packet retransmission comprising:	ADSL2, § 7.8.2.4.3 "Overhead message segmentation" ADSL2, § 8.15.5.1.2 "Message flow, acknowledgement and retransmission" VDSL1, § 12.4.2.6 "Segmentation of messages"
[10.b] a transmitter portion capable of: transmitting a plurality of packets,	ADSL2, §§ 8.8-8.8.1 "Modulation" and "Subcarriers" ADSL2, § K.3.1 "Scope" ADSL2, Fig. K.10 VDSL1, § H.1.1 "Functional model"
[10.c] identifying at least one packet of the plurality of packets as a packet that should be retransmitted and allocating a memory between a retransmission function and an interleaving and/or deinterleaving function,	ADSL2, Figs. 7-7, 7-8 ADSL2, § 7.7.1.5 "Interleaver" ADSL2, § 7.8.2.4.3 "Overhead message segmentation" ADSL2, § 8.15.5.1.2 "Message flow, acknowledgement and retransmission" ADSL2, Fig. C.7-1 VDSL1, § 8-8.1 "PMS-TC sublayer" and "PMS-TC functional model" VDSL1, Fig. 8-1 VDSL1, § 8.4.1 "Interleaving - General" VDSL1, § 8.4.2 "Triangular implementation" VDSL1, Table 8-2 VDSL1, § 12.4.1 "Overview" VDSL1, § 12.4.2.1 "Message format" VDSL1, Tables 12-23, 12-24

Asserted Claims of the '411 Patent	Citations to ITU-T G.992.3 and G.993.1
[10.d] wherein at least a portion of the memory may be allocated to the retransmission function or to the interleaving and/or deinterleaving function at any one particular time, and	ADSL2, § 7.7.1.5 “Interleaver” ADSL2, § 7.8.2.4.3 “Overhead message segmentation” ADSL2, Fig. C.7-1 ADSL2, § C.7.10.3 “Exchange phase” ADSL2, Table C.7-1 VDSL1, §§ 8-8.1 “PMS-TC sublayer” and “PMS-TC functional model” VDSL1, Fig. 8-1 VDSL1, § 8.4.1 “Interleaving - General” VDSL1, Table 8-2
[10.e] wherein a message transmitted during initialization indicates how the memory has been allocated between the retransmission function and the interleaving and/or deinterleaving function in the transceiver.	ADSL2, § 6.6 “Initialization procedure” ADSL2, §§ 7.8.2, 7.8.2.1 “Overhead channel access” and “Overhead channel structure” ADSL2, § 7.8.2.4.3 “Overhead message segmentation” ADSL2, Table 7-18 ADSL2, § 8.15.5.1.2 “Message flow, acknowledgement and retransmission” ADSL2, Tables C.7-1, C.7-2. ADSL2, § C.7.10.3 “Exchange phase” VDSL1, § 12.4.2.6 “Segmentation of messages” VDSL1, § 12.4.6.1 “Sequence of messages and symbols during channel analysis and exchange state” VDSL1, Tables 12-23, 12-24, 12-32.

Asserted Claims of the '411 Patent	Citations to ITU-T G.992.3 and G.993.1
Claim 18	
[18.a] A transceiver capable of packet retransmission comprising:	See [10.a]
[18.b] a receiver portion capable of: receiving a plurality of packets,	See [10.b]
[18.c] identifying at least one packet of the plurality of packets as a packet that should be retransmitted and allocating a memory between a retransmission function and an interleaving and/or deinterleaving function,	See [10.c]
[18.d] wherein the memory is allocated between the interleaving function and the deinterleaving function in accordance with a message received during an initialization of the transceiver and	See [10.e]
[18.e] wherein at least a portion of the memory may be allocated between the retransmission function and the interleaving and/or deinterleaving function at any one particular time depending on the message.	See [10.d]

EXHIBIT A-2

SUMMARY CLAIM CHART A-2

U.S. Patent No. 8,468,411 versus 3GPP TR 25.896, 3GPP TS 25.401, and U.S. Patent No. 6,865,233 issued to Eriksson, et al. (“Eriksson”)

Asserted Claims of the '411 Patent	Citations to 3GPP TR 25.896, TS 25.401, and Eriksson
Claim 10	
[10.a] A transceiver capable of packet retransmission comprising:	TR 25.896, § 7.2.1 “Hybrid ARQ - General” TR 25.896, § 7.2.3 “Associated Signalling” Eriksson, 3:23-38
[10.b] a transmitter portion capable of: transmitting a plurality of packets,	TR 25.896, § 7.2.1 “Hybrid ARQ - General” Eriksson, 3:23-38
[10.c] identifying at least one packet of the plurality of packets as a packet that should be retransmitted and allocating a memory between a retransmission function and an interleaving and/or deinterleaving function,	TR 25.896, § 7.2.1 “Hybrid ARQ - General” TR 25.896, § 7.2.3 “Associated Signalling” TR 24.896, § 8.3.1 “Reuse of current physical layer structure” TR 25.896, Fig. 8.4.7 TR 25.896, § 9.2.2.1.1 “Soft buffer at Node B” TR 25.896, § 9.2.2.1.2 “Reordering buffer in radio network” and Table 9.2.5 TR 25.896, § 9.2.2.1.3 “Retransmission buffer in UE” Eriksson, 2:15-49 Eriksson, 2:65-3:8 Eriksson, 3:23-38, 3:39-52, 3:23-52

Asserted Claims of the '411 Patent	Citations to 3GPP TR 25.896, TS 25.401, and Eriksson
[10.d] wherein at least a portion of the memory may be allocated to the retransmission function or to the interleaving and/or deinterleaving function at any one particular time, and	TR 24.896, § 8.3.1 “Reuse of current physical layer structure” TR 25.896, Fig. 8.4.7 TR 25.896, § 9.2.2.1.1 “Soft buffer at Node B” TR 25.896, § 9.2.2.1.2 “Reordering buffer in radio network” and Table 9.2.5 TR 25.896, § 9.2.2.1.3 “Retransmission buffer in UE” Eriksson, 3:63-4:32
[10.e] wherein a message transmitted during initialization indicates how the memory has been allocated between the retransmission function and the interleaving and/or deinterleaving function in the transceiver.	TR 25.896, § 8.2 “TTI length vs. HARQ physical channel structure” TR 25.896, § 9.2.2.1.1 “Soft buffer at Node B” TR 25.896, § 9.2.2.1.2 “Reordering buffer in radio network” and Table 9.2.5 TR 25.896, § 9.2.2.1.3 “Retransmission buffer in UE” Eriksson, 3:63-4:32
Claim 18	
[18.a] A transceiver capable of packet retransmission comprising:	See [10.a]
[18.b] a receiver portion capable of: receiving a plurality of packets,	See [10.b]
[18.c] identifying at least one packet of the plurality of packets as a packet that should be retransmitted and allocating a memory between a retransmission function and an interleaving and/or deinterleaving function,	See [10.c]
[18.d] wherein the memory is allocated between the interleaving function and	See [10.e]

Asserted Claims of the '411 Patent	Citations to 3GPP TR 25.896, TS 25.401, and Eriksson
the deinterleaving function in accordance with a message received during an initialization of the transceiver and	
[18.e] wherein at least a portion of the memory may be allocated between the retransmission function and the interleaving and/or deinterleaving function at any one particular time depending on the message.	See [10.d]

EXHIBIT B-1

SUMMARY CLAIM CHART B-1
U.S. Patent No. 9,094,348 versus G. 993.1

Asserted Claims of the '348 Patent	Citations to ITU-T G.993.1
Claim 1	
[1.pre] An apparatus comprising:	VDSL1, § 5.1, Figs. 5-1 and 5-2
[1.a] a multicarrier transceiver including a processor and memory operable to:	VDSL1, § 7.2.2 VDSL1, §§ 9.2.1, 9.2.1.3 VDSL1, §§ 8.4, 8.4.1 VDSL1, Fig. 8-2 VDSL1, §§ 8.5, 8.5.1, 8.5.6 VDSL1, § 1.5.4.2 VDSL1, Fig. I.27
[1.b] transmit a packet using a forward error correction encoder and an interleaver	VDSL1, §§ 8, 8.3 VDSL1, §§ 8.4, 8.4.1 VDSL1, §§ 8.5, 8.5.1, 8.5.6 VDSL1, § 12.4.2.4, Table 12-12 VDSL1, §§ I.1.2.7, I.1.2.8 VDSL1, page 5
[1.c] wherein the packet comprises a header field and a plurality of PTM-TC codewords, a plurality of ATM cells or a plurality of Reed-Solomon codewords,	VDSL1, § 7 VDSL1, § 8.5.3 VDSL1, ATM-TC Annex, §§ G.1, Fig. G.1, G.2, G.4, G.4.1 VDSL1, PTM-TC Annex, §§ H.1, H.1.1, Fig. H.1, H.2, H.3.1.1, H.4.1.1 VDSL1, §§ I.1.2, I.1.2.1

Asserted Claims of the '348 Patent	Citations to ITU-T G.993.1
[1.d] and wherein the header field comprises a sequence identifier (SID);	VDSL1, ATM-TC Annex, §§ G.1, Fig. G.1, G.2, G.4, G.4.1 VDSL1, pages 175, 210
[1.e] and receive a plurality of messages using a forward error correction decoder and without using a deinterleaver,	VDSL1, §§ 8, 8.3, 8.5, 8.5.1, 8.5.6 VDSL1, § 12.4.2.4 VDSL1, § I.1.2.7 VDSL1, page 5
[1.f] wherein each message of the plurality of messages is received in a different DMT symbol	VDSL1, § 1 VDSL1, §§ 8.5, 8.5.1
[1.g] and wherein at least one message of the plurality of messages includes an acknowledgement (ACK) or a negative acknowledgement (NACK) of the transmitted packet.	VDSL1, § 8.5.6 VDSL1, §§ 12.4.2, 12.4.2.1, 12.4.2.3, 12.4.2.4, 12.4.4.1, 12.4.4.3.1.2, 12.4.4.3.1.3, Fig. 12-4 VDSL1, § 10.3.2.3 VDSL1, pages 67-68

EXHIBIT B-2

SUMMARY CLAIM CHART B-2

U.S. Patent No. 9,094,348 versus “Interleaved FEC/ARQ coding for QoS multicast over the Internet” by Chen et al. (“Chen”) and G.992.3

Asserted Claims of the '348 Patent	Citations to Chen and ITU-T G.992.3
Claim 1	
[1.pre] An apparatus comprising:	Chen, Abstract ADSL2, Title ADSL2, Summary
[1.a] a multicarrier transceiver including a processor and memory operable to:	Chen, Abstract Chen, page 159, col. 1 ¶ 1-col. 2 ¶ 1 Chen, Fig. 1 ADSL2, I, Scope
[1.b] transmit a packet using a forward error correction encoder and an interleaver	Chen, Abstract Chen, page 159, col. 1 ¶ 2-col. 2 ¶ 1 Chen, page 160, col. 2 ¶ 2 Chen, page 161, col. 2 ¶¶ 1-2 Chen, Figs. 1, 3, 4(b), 5 ADSL2, § 3.8 ADSL2, § 7.2 ADSL2, Fig. 7-6 ADSL2, Tables 7-5, 7-8, 7-12, 7-13, 7-17a ADSL2, §§ 7.7.1.4, 7.7.1.5 ADSL2, § 7.8.2.4.3 ADSL2, § 7.9.1 ADSL2, §§ 8.15.5.1, 8.15.5.1.2, 8.15.5.2.2 ADSL2, § 9.4.1.8

Asserted Claims of the '348 Patent	Citations to Chen and ITU-T G.992.3
[1.c] wherein the packet comprises a header field and a plurality of PTM-TC codewords, a plurality of ATM cells or a plurality of Reed-Solomon codewords,	Chen, page 159, col. 1 ¶ 2-col. 2 ¶ 1 Chen, page 159, col. 2 ¶ 3-page 160, col. 2 ¶ 2 Chen, page 161, col. 2 ¶¶ 1-2 Chen, Figs. 1-3, 4(b) ADSL2, § 7.2 ADSL2, §§ 7.7.1.4, 7.7.1.5 ADSL2, § K.2.5.1 ADSL2, §§ K.2.8.2, K.2.8.3, K.2.8.4, K.2.8.5 ADSL2, § K.2.9.1 ADSL2, §§ K.3, K.3.1, K.3.5 ADSL, Fig. K.10 ADSL2, Table K.13
[1.d] and wherein the header field comprises a sequence identifier (SID);	Chen, page 159, col. 2 ¶ 3 Chen, page 160, col. 1 ¶ 2-col. 2 ¶ 1 Chen, Fig. 1 ADSL2, Summary ADSL2, § 7.7.1.4 ADSL2, § 7.8.2.1 ADSL2, § 8.14
[1.e] and receive a plurality of messages using a forward error correction decoder and without using a deinterleaver,	Chen, Abstract Chen, page 159, col. 1 ¶ 2-col. 2 ¶ 1 Chen, page 159, col. 2 ¶ 3-page 160, col. 1 ¶ 2 Chen, Figs. 1, 5 ADSL2, § 3.8 ADSL2, § 7.2 ADSL2, §§ 7.7.1.4, 7.7.1.5 ADSL2, § 7.9.1 ADSL2, Tables 7-6, 7-17a

Asserted Claims of the '348 Patent	Citations to Chen and ITU-T G.992.3
	ADSL2, Figs. 7-8, 7-12, 7-13 ADSL2, §§ 8.15.5.1, 8.15.5.1.2, 8.15.5.2, 8.15.5.2.2 ADSL2, § 9.4.1.8
[1.f] wherein each message of the plurality of messages is received in a different DMT symbol	Chen, page 163, col. 1 ¶ 1-2 ADSL2, §§ 8.8, 8.8.1, 8.8.2 ADSL2, § 8.13.1.1 ADSL2, § C.8.13.1
[1.g] and wherein at least one message of the plurality of messages includes an acknowledgement (ACK) or a negative acknowledgement (NACK) of the transmitted packet.	Chen, Abstract Chen, page 159, col. 1 ¶ 2-col. 2 ¶ 1 Chen, page 160, col. 2 ¶¶ 1-2 Chen, Figs. 1-3, 4(b), 5 ADSL2, §§ 8.15.5.1, 8.15.5.1.2, 8.15.5.2, 8.15.5.2.2 ADSL2, § 9.4.1.8 ADSL2, Tables 7-12, 7-13, 7-17a

EXHIBIT B-3

SUMMARY CLAIM CHART B-3
U.S. Patent No. 9,094,348 versus G.993.1 and PCT Publication WO2006/030478 / U.S.
Patent No. 8,775,890 to Yap et al. ("Yap")

Asserted Claims of the '348 Patent	Citations to ITU-T G.993.1 and Yap
Claim 1	
[1.pre] An apparatus comprising:	VDSL1, § 5.1, Figs. 5-1 and 5-2 Yap, Abstract Yap, 3:63-4:7
[1.a] a multicarrier transceiver including a processor and memory operable to:	VDSL1, § 7.2.2 VDSL1, §§ 9.2.1, 9.2.1.3 VDSL1, §§ 8.4, 8.4.1 VDSL1, Fig. 8-2 VDSL1, §§ 8.5, 8.5.1, 8.5.6 VDSL1, § 1.5.4.2 VDSL1, Fig. I.27 Yap, 3:63-4:20 Yap, 4:32-53 Yap, Figs. 1-2
[1.b] transmit a packet using a forward error correction encoder and an interleaver	VDSL1, §§ 8, 8.3 VDSL1, §§ 8.4, 8.4.1 VDSL1, §§ 8.5, 8.5.1, 8.5.6 VDSL1, § 12.4.2.4, VDSL, Table 12-12 VDSL1, §§ I.1.2.7, I.1.2.8 VDSL1, page 5 Yap, 3:63-4:20 Yap, 4:32-53 Yap, Figs. 1-2

Asserted Claims of the '348 Patent	Citations to ITU-T G.993.1 and Yap
[1.c] wherein the packet comprises a header field and a plurality of PTM-TC codewords, a plurality of ATM cells or a plurality of Reed-Solomon codewords,	VDSL1, § 7 VDSL1, § 8.5.3 VDSL1, ATM-TC Annex, §§ G.1, Fig. G.1, G.2, G.4, G.4.1 VDSL1, PTM-TC Annex, §§ H.1, H.1.1, Fig. H.1, H.2, H.3.1.1, H.4.1.1 VDSL1, §§ I.1.2, I.1.2.1 Yap, 4:8-20
[1.d] and wherein the header field comprises a sequence identifier (SID);	VDSL1, ATM-TC Annex, §§ G.1, Fig. G.1, G.2, G.4, G.4.1 VDSL1, pages 175, 210 Yap, 4:8-20
[1.e] and receive a plurality of messages using a forward error correction decoder and without using a deinterleaver,	VDSL1, §§ 8, 8.3, 8.5, 8.5.1, 8.5.6 VDSL1, § 12.4.2.4 VDSL1, § I.1.2.7 VDSL1, page 5 Yap, 2:44-49 Yap, 3:63-4:20 Yap, 4:32-53 Yap, 4:57-5:4 Yap, 5:41-48 Yap, Figs. 1-7
[1.f] wherein each message of the plurality of messages is received in a different DMT symbol	VDSL1, § 1 VDSL1, §§ 8.5, 8.5.1 Yap, 4:8-20

Asserted Claims of the '348 Patent	Citations to ITU-T G.993.1 and Yap
[1.g] and wherein at least one message of the plurality of messages includes an acknowledgement (ACK) or a negative acknowledgement (NACK) of the transmitted packet.	VDSL1, § 8.5.6 VDSL1, §§ 12.4.2, 12.4.2.1, 12.4.2.3, 12.4.2.4, 12.4.4.1, 12.4.4.3.1.2, 12.4.4.3.1.3, Fig. 12-4 VDSL1, § 10.3.2.3 VDSL1, pages 67-68 Yap, Abstract Yap, 2:44-49 Yap, 4:45-53 Yap, 4:57-5:4 Yap, 5:41-48 Yap, Figs. 4-7

EXHIBIT C-1

SUMMARY CLAIM CHART C-1
U.S. Patent No. 10,833,809 versus G. 993.1

Asserted Claims of the '809 Patent	Citations to ITU-T G.993.1
Claim 4	
[4.pre] The apparatus of claim 1,	See claim 1 of the '348 Patent
[4.a] wherein a physical layer of the transceiver is capable of generating the packet and the message.	VDSL1, § 7 VDSL1, § 10.1 VDSL1, § G.3 VDSL1, §§ 1.1, 1.1.1 VDSL1, § 1.4 VDSL1, pages 4, 210-12
Claim 6	
[6.pre] The apparatus of claim 2,	See [4.pre] VDSL1, § 11.3 VDSL1, § 14.2.6 VDSL1, §§ 14.3, 14.3.2 VDSL1, Fig. 14-5 VDSL1, § 1.4
[6.a] wherein a physical layer of the transceiver is capable of generating the packet and the message.	See [4.a]

EXHIBIT C-2

SUMMARY CLAIM CHART C-2

U.S. Patent No. 10,833,809 versus “Interleaved FEC/ARQ coding for QoS multicast over the Internet” by Chen et al. (“Chen”) and G.992.3

Asserted Claims of the '809 Patent	Citations to Chen and ITU-T G.992.3
Claim 4	
[4.pre] The apparatus of claim 1,	See claim 1 of the '348 Patent
[4.a] wherein a physical layer of the transceiver is capable of generating the packet and the message.	Chen, page 159, col. 1 ¶ 1-col. 2 ¶ 2 Chen, page 160, col. 2 ¶ 2 Chen, Fig. 1 ADSL2, Summary ADSL2, § 5.1 ADSL2, Fig. 5-1 ADSL2, § 9.3 ADSL2, Annex F ADSL2, Appendix I ADSL2, Fig. I.1
Claim 6	
[6.pre] The apparatus of claim 2,	See [4.pre] Chen, Abstract ADSL2, Summary ADSL2, § 7.2 ADSL2, § 7.7.1.5 ADSL2, § 7.10.3 ADSL2, § 8.9.2

Asserted Claims of the '809 Patent	Citations to Chen and ITU-T G.992.3
	ADSL2, § 8.12.3.6
[6.a] wherein a physical layer of the transceiver is capable of generating the packet and the message.	See [4.a]

EXHIBIT C-3

SUMMARY CLAIM CHART C-3
U.S. Patent No. 10,833,809 versus G.993.1 and PCT Publication WO2006/030478 / U.S.
Patent No. 8,775,890 to Yap et al. (“Yap”)

Asserted Claims of the '809 Patent	Citations to ITU-T G.993.1 and Yap
Claim 4	
[4.pre] The apparatus of claim 1,	See claim 1 of the '348 Patent
[4.a] wherein a physical layer of the transceiver is capable of generating the packet and the message.	VDSL1, § 7 VDSL1, § 10.1 VDSL1, § G.3 VDSL1, §§ 1.1, 1.1.1 VDSL1, § 1.4 VDSL1, pages 4, 210-12 Yap, 2:44-49 Yap, 3:63-4:20 Yap, 4:32-53 Yap, 4:57-5:4 Yap, 5:41-48 Yap, Figs. 1-7
Claim 6	
[6.pre] The apparatus of claim 2,	See [4.pre] VDSL1, § 11.3 VDSL1, § 14.2.6 VDSL1, §§ 14.3, 14.3.2 VDSL1, Fig. 14-5 VDSL1, § 1.4 Yap, Abstract

Asserted Claims of the '809 Patent	Citations to ITU-T G.993.1 and Yap
	Yap, 3:63-4:7
[6.a] wherein a physical layer of the transceiver is capable of generating the packet and the message.	See [4.a]