

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE PATENT TRIAL AND APPEAL BOARD

---

MICROSOFT CORPORATION

Petitioner

v.

NONEND INVENTIONS, N.V.

Patent Owner

---

PTAB Case No. IPR2016-00223

Patent No. 7,590,752 B2

---

**PETITION FOR *INTER PARTES* REVIEW  
OF U.S. PATENT NO. 7,590,752 B2**

## TABLE OF CONTENTS

	Page
EXHIBIT LIST .....	ii
I. INTRODUCTION .....	1
II. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(B) .....	1
III. REQUIREMENTS FOR INTER PARTES REVIEW .....	2
IV. OVERVIEW OF THE 752 PATENT.....	3
V. THERE IS A REASONABLE LIKELIHOOD THAT AT LEAST ONE CLAIM OF THE 752 PATENT IS UNPATENTABLE .....	6
A. IDENTIFICATION OF THE REFERENCES AS PRIOR ART.....	6
B. SUMMARY OF INVALIDITY POSITIONS .....	9
VI. DETAILED EXPLANATION OF PROPOSED GROUNDS .....	10
A. GROUND 1: CLAIMS 1, 3-8, 10 AND 14-16 ARE obvious OVER TUTORIAL AND HACKER.....	12
B. GROUND 2: CLAIMS 2, 11-13 ARE OBVIOUS OVER GNUTELLA TUTORIAL, HACKER AND SCHUSTER.....	38
C. GROUND 3: CLAIM 9 IS OBVIOUS OVER GNUTELLA TUTORIAL, GNUTELLA PROTOCOL AND HACKER .....	41
D. GROUND 4: CLAIMS 1, 3-10 AND 14-16 ARE OBVIOUS OVER CLARKE AND EHRMAN .....	43
E. GROUND 5: CLAIMS 2, 11-13 ARE OBVIOUS OVER CLARKE, EHRMAN AND SCHUSTER .....	58
VII. CONCLUSION.....	60

## EXHIBIT LIST

- Ex. 1001 U.S. Patent No. 7,590,752B2 to Oldenborgh et al. (“752 Patent”)
- Ex. 1002 through Ex. 1006 are not used in this proceeding
- Ex. 1007 Netherlands Patent Application No. 1017388, Filed Feb. 16, 2001, and English translation thereof submitted by Nonend Inventions in prosecution of European Patent Application No. 02 711 547.6
- Ex. 1008 not used in this proceeding
- Ex. 1009 Gnutella Tutorial (June 20, 2000) (“Tutorial” or “Gnutella Tutorial”)†
- Ex. 1010 The Gnutella Protocol Specification v0.4 (August 18, 2000)†
- Ex. 1011 Hacker MP3 The Definitive Guide (“Hacker”)
- Ex. 1012 I. Clarke, O. Sandberg, B. Wiley, and T. Hong, *Freenet: A Distributed Anonymous Information Storage and Retrieval System*, (Jul. 2000) (“Clarke”) †
- Ex. 1013 U.S. Prov. App. No. 60/237,365 (Oct. 4, 2000) (“Ehrman”)
- Ex. 1014 through Ex. 1018 are not used in this proceeding
- Ex. 1019 Declaration of Dr. George Kesidis on U.S. Patent No. 7,590,752B2 (“Kesidis Declaration”)
- Ex. 1020 Declaration of Sara Hare
- Ex. 1021 not used in this proceeding
- Ex. 1022 Affidavit of Christopher Butler (“Archive Declaration”)
- Ex. 1023 Declaration of Spencer Kimball (“Kimball Gnutella Declaration”)
- Ex. 1024 Declaration of Ziad Elkurjie (“Gnutella Protocol Declaration”)
- Ex. 1025 not used in this proceeding
- Ex. 1026 Joint Claim Construction Chart in Case 1:12-cv-01041-GMS

Ex. 1027 through Ex. 1049 are not used in this proceeding

Ex. 1050 U.S. Patent No. 6,914,897 to Schuster et al. (“Schuster”)

Ex. 1051 is not used in this proceeding

Ex. 1052 Declaration of Nathan Moinvaziri (“Moinvaziri Declaration”)

Ex. 1053 through Ex. 1064 are not used in this proceeding

Ex. 1065 Declaration of Andy Oram (“Oram Declaration”)

Ex. 1066 Declaration of Michael Freedman (“Freedman Declaration”)

Ex. 1067 Declaration of Brandon Wiley (“Wiley Declaration”)

Ex. 1068 Declaration of Theodore Hong (“Hong Declaration”)

†For ease of reference, paragraph numbers have been added to these references.

## I. INTRODUCTION

Pursuant to 35 U.S.C. § 311 and 37 C.F.R. § 42.100, Microsoft Corporation (“Petitioner”) petitions for *inter partes* review (“IPR”) of Claims 1-16 of U.S. Pat. No. 7,590,752 B2 (“the 752 Patent,” Ex. 1001), assigned to Nonend Inventions, N.V. This Petition shows a reasonable likelihood that Petitioner will prevail with respect to at least one of Claims 1-16 challenged under 35 U.S.C. § 314(a).

Petitioner has established the invalidity of each ground under Pre-AIA 35 U.S.C. §103 by a preponderance of the evidence in compliance with 35 U.S.C. § 316(e).

## II. MANDATORY NOTICES UNDER 37 C.F.R. § 42.8(B)

Pursuant to 35 U.S.C. §312(a)(2) and 37 C.F.R. §42.8(b)(1), all real parties in interest for this petition are Microsoft Corporation.

**RELATED MATTERS:** Patent Owner asserts the 752 Patent against Microsoft Corporation and others in 17 pending lawsuits which have been consolidated under the lead case of *Nonend Inventions, N.V. v. Apple, Inc.*, No. 15-cv-00466-JRG-RSP (E.D. Tex. Apr. 9. 2015). Petitioner is pursuing IPR petitions on U.S. Patent Nos. 7,779,138 B2 and 8,099,513 B2 asserted in 16 of these 17 lawsuits. *See*, PTAB Case Nos. IPR2016-00224 and IPR2016-00225.

**NOTICE OF COUNSEL AND SERVICE INFORMATION:** Pursuant to 37 C.F.R. §§ 42.8(b)(3), 42.8(b)(4) and 42.10(a), Petitioner appoints Bing Ai (Reg. No. 43,312) as its lead counsel, and Matthew C. Bernstein (*pro hac vice*), Kevin J.

Patariu (Reg. No. 63,210), Christopher L. Kelley (Reg. No. 42,714), Vinay P. Sathe (Reg. No. 55,595), and Philip A. Morin (Reg. No. 45,926) as back-up counsel, all at the mailing address of Perkins Coie LLP, 11988 El Camino Real, Suite 350, San Diego, CA 92130, contact numbers of 858-720-5700 (phone) and 5 858-720-5799 (fax), and the following email for service and all communications:

[PerkinsServiceMSFT-Nonend-IPR@perkinscoie.com](mailto:PerkinsServiceMSFT-Nonend-IPR@perkinscoie.com). Petitioner hereby consents to electronic service under 37 C.F.R. § 42.6(e). Pursuant to 37 C.F.R. § 42.10(b), a power of attorney is concurrently filed.

### **III. REQUIREMENTS FOR *INTER PARTES* REVIEW**

10 This Petition complies with all statutory and rule-based requirements under 37 C.F.R. §§ 42.104, 42.105 and 42.15 and thus should be accorded a filing date as the date of filing of this Petition. 37 C.F.R. § 42.106.

**GROUND FOR STANDING:** Pursuant to § 42.104(a), Petitioner hereby certifies that the 752 Patent is available for IPR and that it is not barred or estopped 15 from requesting IPR challenging claims of the 752 Patent. Petitioner has the standing and meets all requirements under 35 U.S.C. §§ 315(a)(1), 315(b), 315(e)(1) and 325(e)(1); and 37 C.F.R. §§42.73(d)(1), 42.101 and 42.102.

**IDENTIFICATION OF CHALLENGE:** Pursuant to 37 C.F.R. §§ 42.104(b) and 42.22, the precise relief requested is that the Board institute an IPR 20 trial on Claims 1-16 and cancel Claims 1-16. **Claims Challenged:** Claims 1-16 are

challenged in this Petition. **The Prior Art:** The prior art relied upon is discussed or referred to in papers filed with the Petition. *See* the Exhibit List and Kesidis

Declaration (Ex. 1019). **Supporting Evidence Relied Upon For The Challenge:**

The supporting evidence includes the Kesidis Declaration, the Hare Declaration

5 (Ex. 1020) and other supporting evidence in the Exhibit List or referred in the

papers filed with this Petition. **Statutory Ground(s) Of Challenge And Legal**

**Principles:** Pursuant to 37 C.F.R. § 42.104 (b)(2), the requested review is

governed by pre-AIA 35 U.S.C. §§ 102 and 103 and AIA 35 U.S.C. §§ 311 to 319  
and 325.

10 **Claim Construction:** The 752 Patent has not expired. In an IPR, “[a] claim in an unexpired patent shall be given its broadest reasonable construction in light of the specification of the patent in which it appears.” 37 C.F.R. § 42.100(b); *see also In re Cuozzo Speed Techs., LLC*, 778 F.3d 1271, 1280 (Fed. Cir. 2015).

**How Claims Are Unpatentable Under Statutory Grounds:** Pursuant to 37

15 C.F.R. § 42.104 (b)(4), Section VI provides an explanation of how Claims 1-16 are unpatentable pursuant to 35 U.S.C. § 103, including the identification of where each element of the claim is found in the cited prior art.

#### **IV. OVERVIEW OF THE 752 PATENT**

**SUMMARY OF THE 752 PATENT:** The 752 Patent relates to a system

20 for distributing media content from a first media player device to multiple other

media player devices. (Ex. 1001, Abstract). The first media player device retrieves an item of media content from another media player device. *Id.* A second media player device then requests and receives the item of media content from the first media player device. *Id.* The second media player device plays back the item on the second media player device while the first media player device also plays back the item of media content on the first media player device. *Id.*

**SUMMARY OF PROSECUTION HISTORY:** The 752 Patent was granted from Appl. No. 11/613,799 filed Dec. 20, 2006 and claimed priority to a Netherlands application of Feb. 16, 2001 (Ex. 1007). Figures 7 through 15 of the 138 Patent and text describing these figures were added to the U.S. Application No. 09/967,600 filed on Sep. 28, 2001 and do not appear in the original Netherlands application. The Patent Office rejected Patent Owner's amended and newly added claims over U.S. Patent No. 6,353,174 to Schmidt. Patent Owner distinguished Schmidt by arguing it "does not disclose a first media player that can both retrieve media content in response to a play - back request and receive requests from other media players for the media content" and amended claims to recite that the requests for content made by other devices are "sent in an uncoordinated order with respect to one another." The Patent Office allowed the amended claims. As shown by the cited prior art in this Petition, the Examiner relied on an incomplete record of relevant prior art during the examination. A person of ordinary skill in



the art (“POSITA”) in the technology of the 752 Patent would recognize highly relevant prior art cited in this Petition that reveal fatal defects of the issued claims. This Petition seeks to correct the unfortunate issuance of the invalid Claims 1-16 by providing a fuller and more complete record of relevant prior art to the Office and the public.

**LEVEL OF ORDINARY SKILL IN THE ART:** A POSITA with regard to the 752 Patent is an individual with a bachelor’s degree in computer science or related field and two years of working experience in computer inter-networking, or with equivalent total experience in computer inter-networking. (Ex. 1019, § VI).

**PROPOSED CLAIM CONSTRUCTION:** Petitioner construes certain claim terms below pursuant to the broadest reasonable interpretation (BRI) standard to comply with 37 C.F.R. §§ 42.100(b) and 42.104(b)(3) and for the sole purpose of this Petition. Such BRI constructions do not necessarily reflect appropriate claim constructions to be used in litigation and other proceedings where a different claim construction standard applies. Under BRI, “streaming” should be construed as “Transmission of content that may be played by the receiver while the transmission is occurring,” (Ex. 1001 at 2:38-45, 7:8-27); “Media player” should be construed under BRI as “Television, radio, or playback equipment for media content or a personal computer that is suitable to that end,” (*Id.*, 4:29-31); “Stream target” should be construed as “Media player software

and/or hardware that receives content,” (*Id.* 13:1-4, 14:15-19). The above proposed constructions are also consistent with Patent Owner’s previous proposed or stipulated constructions in litigation. *See* Ex. 1026, Ex. A (“media player,” “stream target”), Ex. B at 1-2 (“streaming”). While “streaming” was previously given a narrower litigation construction, “the continuous transmission of data packages that may be played by the receiver while the transmission is occurring,” Dr. Kesidis explains that that this element is taught by the prior art for each Ground even under this narrower construction. Ex. 1019, ¶¶26,99-105. All remaining terms in Claims 1-16 should be construed to have their respective plain and ordinary meaning under BRI.

**V. THERE IS A REASONABLE LIKELIHOOD THAT AT LEAST ONE CLAIM OF THE 752 PATENT IS UNPATENTABLE**

Claims 1-16 are unpatentable under 35 U.S.C. § 103(a) for merely reciting known, predictable and/or obvious combinations of the cited prior art references.

**A. IDENTIFICATION OF THE REFERENCES AS PRIOR ART**

**Gnutella Tutorial As Prior Art Under 35 U.S.C. § 102(a):** The Gnutella

Tutorial (Ex. 1009) was published at the domain gnutella.wego.com on Jun. 20, 2000 or earlier. The affidavit of C. Butler (Ex. 1022, *e.g.*, pp. 18-25), a representative of the Internet Archive, confirms that the document was captured by the Archive on that date. Users arriving at the home page <http://gnutella.wego.com> were presented with a link to the Tutorial on the left hand side of the page which, if

clicked, took them directly to the Tutorial. (Ex. 1019, ¶¶126-132). As explained in the declarations of S. Kimball (Ex. 1023, *e.g.*, Attachment F) and N. Moinvaziri (Ex. 1052, *e.g.*, Attachment F), the “gnutella.wego.com” site was designed as a destination for individuals interested in Gnutella and provided information about, and downloads of, Gnutella clients. (*Id.*, Item 19, p. 6; Ex. 1019, ¶133.) The Tutorial webpage provides a description of the operation of the Gnutella file sharing software. (Ex. 1009, ¶¶ 6, 7, FIG. 1) There was tremendous interest in the Tutorial in public and technical literature in the years and months leading up to the priority date of the 752 Patent. (*Id.*, ¶¶134-135) The WeGo website was frequently identified as a place for public to obtain and learn about Gnutella, and the URL for the website was given in many printed publications in 2000 and 2001, demonstrating that the WeGo site was well known, publicly accessible and was frequently accessed. (*Id.*, ¶136). The Oram Declaration also confirms public availability of Gnutella before the priority date (Ex. 1065, *e.g.*, Appendix B).

**Gnutella Protocol As Prior Art Under 35 U.S.C. § 102(a)**: The Gnutella Protocol Specification v0.4 (Ex. 1010) was published at the URL <http://gnutellahosts.com/GnutellaProtocol04.pdf> on Aug. 18, 2000, or earlier, as confirmed by the affidavit of C. Butler (Ex. 1022, pp. 29-36), who affirms that the document was captured by the Archive on that date. The Kimball Declaration (*e.g.*, Ex. 1023, Attachment J) also confirms this date. As explained in the Declaration of

Z. Elkurjie (Ex. 1024, *e.g.* Attachment F), this document was prepared and posted online by Clip2, a company that provided Gnutella-related technical data to developers and users. Clip2 posted this document to the site gnutellahosts.com, which gave information about hosts that users could access to connect to the

5 Gnutella network. The document was first directly accessible from the “front page” of gnutellahosts.com and later from an “Articles” tab on the front page. (Ex. 1019, ¶¶143-149). The Moinvaziri declaration also confirms this date (*e.g.*, Ex. 1052, Attachment F). The Oram Declaration also confirms public availability of Gnutella before the priority date of the 752 Patent (Ex. 1065).

10       **Hacker As Prior Art Under 35 U.S.C. § 102(a):** Hacker was published by O’Reilly in at least March 2000. Hacker describes features of MP3 audio encoding, downloading and playback technologies, applicable to many different computer operating systems. (Ex. 1011, p. 9). Hacker also describes operation of Winamp MP3 player (*Id.*, Chapter 8). (Ex. 1019, ¶¶137-141)

15       **Clarke As Prior Art Under 35 U.S.C. § 102(a):** Clarke was published as early as Jul. 1, 2000 (Ex. 1012, Front page), as confirmed by the Butler affidavit (Ex. 1022, pp. 37-48, 165-181, Exhibit B), the Oram Declaration (Ex. 1056, Attachments B and D), the Freedman Declaration (Ex. 1066, Attachment C and E, pp. 43-58), the Wiley Declaration (Ex. 1067), and the Hong Declaration (Ex. 1068, 20 Attachment C and D). Clarke discloses a publicly available software platform

called “Freenet” that allowed user computers to share, by requesting and downloading, files in a computer network. (Ex. 1012, Abstract, ¶¶ 1, 9; Ex. 1019, ¶¶149-152). The Oram Declaration also confirms public availability of Freenet prior to the critical date of the 752 Patent (Ex. 1065).

5        **Ehrman As Prior Art Under 35 U.S.C. § 102(e)**: Ehrman was filed as a U.S. Prov. Appl. 60/237,365 on Oct. 4, 2000 and published on Apr. 4, 2002 (Ex. 1013). Ehrman discloses a peer-to-peer (P2P) network of computers in which user devices, acting as nodes of the P2P network, can share video files by downloading over the Internet. (*Id.*, 1:8-10, 4:1-6, 4:10-14, 5:19-20; Ex. 1019, ¶¶153-157.)

10        **Schuster As Prior Art Under 35 U.S.C. § 102(e)**: Schuster was filed as a patent application on Feb. 20, 2000, and discloses a portable information device (PID) that is used for receiving content, *e.g.*, audio and video, and playing the content back to a user. (Ex. 1050, Cover Page and Abstract). PID can be a personal digital assistant (PDA), a phone, a network telephone. (*Id.*, Title, Abstract, 7:43-61, 15    9:19-36, 9:56 -10:7, 14:9-15, 18:4-22, 20:49-63; Ex. 1019, ¶158.)

## **B. SUMMARY OF INVALIDITY POSITIONS**

This Petition uses two primary references, (1) Gnutella Tutorial and (2) Clarke, neither of which was explicitly considered by the Office, to form independent and distinctive invalidity grounds against Claims 1-16. This Petition  
20    also combines these references with (3) Hacker; (4) Ehrman, (5) Gnutella Protocol

and (6) Schuster to show invalidity of Claims 1-16. Under the BRI claim construction and from the perspective of a POSTIA as stated in prior sections, this Petition determines the scope and content of the cited prior art in the proposed invalidity grounds, analyzes the differences between the claimed invention and the cited prior art by considering the claimed invention as a whole and the cited prior art as a whole under the BRI claim construction, and concludes that the prior art in the proposed invalidity grounds discloses all elements recited in Claims 1-16 and renders the claimed subject matter as a whole obvious and unpatentable.

## **VI. DETAILED EXPLANATION OF PROPOSED GROUNDS**

An object of the 752 Patent is “to at least partially remove the drawbacks [of prior art] mentioned explicitly or implicitly.” (Ex. 1001, 1:55-57). Such drawbacks are discussed in its Background section (*Id.* 1:15-52): growing and maintaining a network required “server duplication” (*Id.* 1:18-28), distributed processing that is in contact with a common server (*Id.* 1:29-36), a centrally controlled hierarchical system (*Id.* 1:37-40), and hierarchical systems of agents that requires an overhead structure (*Id.* 1:41-52).

The primary references, the Tutorial and Clarke, do not suffer from the alleged shortcomings of the prior art discussed by the 752 Patent. (Ex. 1019, ¶¶201-204.) The Tutorial teaches a P2P file sharing technology called Gnutella.

Gnutella allows users to share files (including MP3 files) with each other without

performing server duplication, without being controlled by a common server and without using a hierarchical system, as the name “peer-to-peer” indicates. (Ex. 1019, ¶¶205-206.) The Tutorial explicitly states that “every client is a server and every server is a client” and “Gnutella does not use servers to connect to.” (Ex.

5 1009, ¶¶ 6, 8; Ex. 1019, ¶207.) In **Ground 1**, the Tutorial and Hacker together illustrate the applicability of Gnutella’s MP3 media file sharing with Winamp player, a free, publicly available, downloadable software. In **Ground 2**, the Tutorial-Hacker -Schuster combination demonstrates that a POSITA already knew to use portable, wireless computers for MP3 file sharing and playback. In **Ground**  
10 **3**, Gnutella Protocol is combined with the Tutorial and Hacker to establish obviousness of making specific requests for parts of media files.

Clarke teaches a P2P file sharing technology, called “Freenet.” A software implementation of Freenet was available to the public via a free download from the Internet. (Ex. 1019, ¶211.) Freenet divided large files into multiple parts and  
15 allowed search and downloading of files from a network-connected computer. (Ex. 1012, ¶ 40; Ex. 1019, ¶212.) Freenet, being a “distributed” “peer-to-peer” network application, operated without being controlled by a central server and without using a hierarchical system or the need for server duplication, the limitations of the prior art alleged in the 752 Patent. While files propagated through the P2P network,  
20 computers along the way locally stored the files for further sharing. (Ex. 1019,

¶213.) In **Ground 4**, Clarke is combined with Ehrman, a P2P technology for distributing video files, to demonstrate

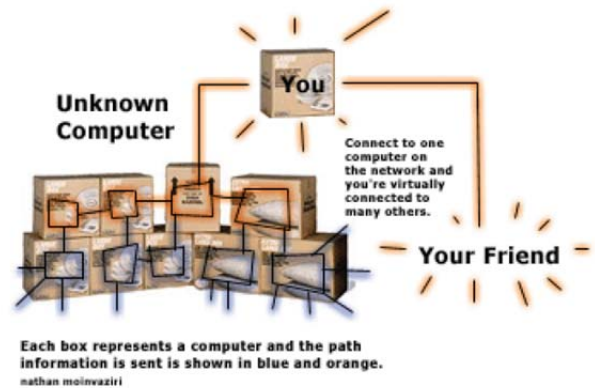
that it was obvious to use Freenet for

sharing media files. In **Ground 5**,

5 Clarke and Ehrman are combined with

Schuster to establish obviousness of

making specific requests for parts of media files.



**A. GROUND 1: CLAIMS 1, 3-8, 10 AND 14-16 ARE OBVIOUS OVER TUTORIAL AND HACKER**

10 The Tutorial describes the Gnutella software and messaging protocol for use in a P2P network of user computers for sharing audio files among peer computers.

(Ex. 1009 at ¶¶ 1-15). Gnutella application was freely available to the public for download and installation on computers. (Ex. 1019, ¶¶ 301-302.) To run a Gnutella

network of peers, no central computer was required and users could search for and

15 download files in the P2P network using Gnutella software application running on

their computers. (Ex. 1009, ¶¶ 7, 8; Ex. 1019, ¶¶ 303-304.) For user convenience,

MP3 files could be played back using the Winamp software application during or after file downloading. (Ex. 1009, ¶23, Ex. 1019, ¶¶ 303-304.)

The P2P network in FIG. 1 of the Tutorial is reproduced above and shows

20 the Gnutella as “a peer-to-per[sic] network.” (Ex. 1009, FIG. 1). Every computer

on the Gnutella network is considered a hub and everybody on the network is both



a client and a server. (Ex. 1009 at ¶¶ 6-8). The Tutorial teaches two modes by which MP3 files are transferred between peers: streaming, by which a user can listen to a song as it downloads, and downloading, without simultaneous playback.

*Id.* Such teachings directly go to the retrieval of streaming media content as

5 claimed in the 752 Patent. Hacker, is in the same field of MP3 file-sharing and playback as the Tutorial, describes software apps, including Winamp, executed on a user computer. (Ex. 1011, pp. 48-54, 61-65, 122, 152-154, 301-310, 88, 125, 206). Like the Tutorial, Hacker expressly discloses both ways to receive MP3 files: streaming and downloading. (Ex. 1011, pp. 88-89.) Both Tutorial and Hacker  
10 similarly describe the “playlist” feature of Winamp, which shows to the user a list of MP3 files cued up for download and playback on the Winamp player. (Ex. 1009, ¶23). Hacker teaches that the Winamp player included a playlist editor utility to allow a user to place MP3 file names in the playlist and download the files from the Internet. (Ex. 1011, Ch. 4, pp. 10-151).

15 As explained in detail below, the teachings in Tutorial and Hacker, and their combination, demonstrate that Claims 1-16 simply combine prior art elements according to known methods/techniques to yield predictable results or improvements, choose from a finite number of identified, predictable solutions, with a reasonable expectation of success, and merely use known techniques in the  
20 same or similar technical fields. The teachings in Tutorial and Hacker also suggest

or motivate a POSITA to modify or combine prior art to arrive at the claimed invention and a POSITA would have found it obvious to combine the teachings of Tutorial and Hacker. For example, a POSITA would have been guided or motivated by Tutorial's express teaching of combining Gnutella software and

5 Winamp player to have a reasonable expectation of success based on the teachings in Tutorial and Hacker and the general information available on Winamp. A POSITA would have recognized the convenience offered by pairing of Winamp with Gnutella for MP3 file sharing and playback. (Ex. 1019, ¶¶ 306-307). Such a combination was not only explicitly taught by Tutorial but also consists of no more  
10 than applying prior art elements according to known methods to yield predictable results. (Ex. 1019, ¶ 308). The following analyses and discussions of the teachings in Tutorial and Hacker in connection with each claim on an element-by-element basis provide additional reasons or motivations to combine.

**Claim 1: Element 1.**Preamble A system for distributing media content

15 from a first media player to other media players that include a second media player, comprising: The Tutorial discloses the Gnutella P2P network that allows computers, which are media players, to search and download MP3 files from peers within the network. (Ex. 1009, ¶¶ 6, 7, 8, 11, 13, 14, 17, 19). Computers in the Gnutella network function both as a client and a server, and can include a Winamp  
20 MP3 player app. (Ex. 1009, ¶¶ 6, 7, 8, 11). Hacker discloses additional details of

the Winamp app for downloading and playing back MP3 files. (Ex. 1011, pp. 48-54, 61-65, 122, 152-154, 301-310, 88, 125, 206). The Tutorial teaches a “Stream selected files” option in Gnutella, which invokes Winamp for playback of MP3 files. Tutorial thus explicitly teaches combining Gnutella with Winamp. Both

5 Gnutella and Winamp were written by Nullsoft, and thus a POSITA would have found it obvious and straightforward to combine their operations for MP3 file sharing. (Ex. 1019, ¶¶ 309-313). A POSITA would have been guided by those and other teachings to combine Tutorial and Hacker.

**Element 1.A:** the first media player configured to: receive a first play-back request for an item of media content from a component of the first media player, wherein the first media player includes a first stream target: Tutorial discloses that a peer computer in the Gnutella network, which is the first media player, receives play-back request for an MP3 file, which is an item of media content. The request may be received by the Gnutella user interface (UI) by a user selecting an MP3 file and then selecting “Stream selected files” which causes the file to be played back (while being downloaded) using Winamp app, which is the first stream target, while being downloaded. (Ex. 1009, ¶¶ 1,6,7,14,16, 23, FIG. 3). As further explained in Hacker, the playback request may also be received using a playlist feature by which a user can add an MP3 file to a list for playback by Winamp by specifying file name or location in the Internet. (Ex. 1011, pp. 6, 7, 19, 20, 41, 51,

63, 66, 88, 89, 120, 121, 150-152, Ch. 8). A POSITA would have recognized the benefit of combining Gnutella's file streaming Winamp's playlists and playback capability to search for and download MP3 files from the Internet, according to the explicit teachings of Tutorial and using known techniques with predictable results.

5 (Ex. 1019, ¶¶ 314-318). This element is thus disclosed and a POSITA would have been guided by those and other teachings to combine Tutorial and Hacker.

**Element 1.B:** in response to the first play-back request, initiate a first process to retrieve at least a part of the item of media content from at least one device selected from the group consisting of the first media player and one or more

10 further devices: The Tutorial discloses that a user is able to request an MP3 file that may be available at the user's computer, or another computer in the Gnutella network. (Ex. 1009, ¶¶ 1,6,7,14,16-22). For example, the Tutorial shows that the software can search the network for files matching a given search string (*e.g.*, "29 items" for the file name that the user was searching for, as depicted in FIG. 3). (Ex. 15 1009, ¶16, FIG. 3). Upon receiving results of the request, user can save the file or directly stream it to a player (as detailed by Hacker). (Ex. 1009, ¶23, FIG. 3), (Ex. 1011, pp. 48-54, 61-65, 122, 152-154, 301-310, 88, 125, 206). A POSITA would have recognized that Tutorial teaches to combine the file retrieval process of the Tutorial with the MP3 file downloading feature of Winamp (as taught by Hacker) 20 to be able to download and enjoy MP3 music via the Internet. (Ex. 1019, ¶¶ 319-

321.) This element is thus disclosed and a POSITA would have been guided by those and other teachings to combine Tutorial and Hacker.

**Element 1.C:** output at least said part of the item of media content retrieved as a result of said first process to the first stream target to provide streaming of

5 media content: Tutorial teaches that the Gnutella application may output at least part of an item of media content such as an MP3 file to the Winamp application to stream the MP3 file while it is downloading using the “Stream selected files” button. (Ex. 1009, ¶23, FIG. 3). A POSITA would have recognized that the Tutorial teaches to combine this feature with the Winamp player, taught by Hacker,  
10 to stream and play back audio files in real time. (Ex. 1019, ¶¶ 322-324.) This element is thus disclosed and a POSITA would have been guided by those and other teachings to combine the Tutorial and Hacker.

**Element 1.D:** receive, from a plurality of the other media players, a plurality of further requests for the item of media content that are sent in an uncoordinated

15 order with respect to one another from the plurality of other media players to the first media player, the further requests being received by the first media player after receiving the first play-back request: Tutorial discloses that any user’s computer is a peer in the Gnutella network and therefore can transmit (upload) and receive (download) search requests from many other peers in the Gnutella network.  
20 (Ex. 1009, ¶¶ 7,8,23). Because other computers are operated by other users, the

MP3 file requests from the other computers are generally not organized,  
independent and therefore uncoordinated with each other. (Ex. 1019, ¶¶ 325-326.)

Because the other users may request and download an MP3 file after a user has  
downloaded it, it is disclosed and obvious that the player may receive a plurality of  
5 further requests after the first user has requested the MP3 to be downloaded. (*Id.*,  
¶327.) This element is thus disclosed and a POSITA would have been guided by  
those and other teachings to combine the Tutorial and Hacker.

**Element 1.E** in reply to the further requests for the item of media content,  
transmit one or more parts of the item of media content, retrieved as a result of said

10 first process, to the requesting other media players: Tutorial discloses that a user's  
computer may provide multiple simultaneous uploads to other users. (Ex. 1009, ¶¶  
15,23, FIG. 3, "outgoing" and "incoming" items from FIG.2). Hacker teaches that  
users can set up their own computers to be servers also. Ex (1011, 296-300) As  
disclosed both in Hacker and Tutorial, the uploads can be MP3 files that are stored  
15 on the user's computer. (Ex. 1009, ¶¶ 20, 24, FIG. 4), (Ex. 1011, pp. 48-54, 61-65,  
122, 152-154, 301-310, 88, 125, 206). Once a file gets stored in a directory that the  
user has marked for sharing, the file becomes available to other users for search  
and download, as disclosed in Tutorial. (Ex. 1009, ¶24, FIG. 4). The Tutorial  
teaches that files are transmitted in one or more parts using packetized

20 transmission. (Ex. 1009, ¶15, Fig. 2). In a P2P network, such as Gnutella, where

hundreds or thousands of music files were being downloaded and uploaded, it would have been obvious to a POSITA that a user's computer on which both Gnutella and Winamp are implemented will receive requests for a media file that the computer downloaded from the network (*i.e.*, retrieved as a result of the "first" download process), and will in turn provide the file to other requesting peers. (Ex. 1019, ¶¶ 328-332). Therefore, it is disclosed that these features of the Tutorial could be combined with Hacker's Winamp player to allow multiple users of the Internet to share MP3 files for enjoyment. (Ex. 1019, ¶ 332.) This element is thus disclosed and a POSITA would have been guided by the teachings to combine the Tutorial and Hacker.

**Element 1.F** the second media player configured to: receive a second play-back request for the item of media content from a component of the second media player, wherein the second media player includes a second stream target: See Claim element 1.A. The "second media player" is simply a second computer in the Gnutella network equipped with similar software. (Ex. 1019, ¶¶ 333-334.) This element is thus disclosed and a POSITA would have been guided by those and other teachings to combine the Tutorial and Hacker.

**Element 1.G** in response to the second play-back request, initiate a second process to retrieve at least a part of the item of media content from the first media player, the second process including sending a request for at least said part of the

item of media content to the first media player: See Claim elements 1.B,

1.Preamble and 1.F. In addition, the Tutorial teaches that a user can use the Gnutella application to search for MP3 files in the Gnutella network and both transmit (upload) and receive (download). (Ex. 1009, ¶¶ 7,8,16-23, Figs. 3,4). The user's Gnutella application search for the MP3 file on every other computer, called Gnutella "hosts" in this context, in the Gnutella network. (*Id.* ¶ 23). When a Winamp player is installed on the user's computer, the user can receive and play back an MP3 file that is found by the Gnutella application by selecting the "Stream selected files" option. (*Id.* ¶ 23, FIG. 3). Thus, when the second computer in the Gnutella network is instructed by its user to find and stream an MP3 file, and when the requested file is available on the first computer in the Gnutella network, the second computer will initiate the process of retrieving the file from the first computer. (Ex. 1019, ¶¶ 335-338.) This element is thus disclosed and a POSITA would have been guided by prior art teachings to combine the Tutorial and Hacker.

**Element 1.H** receive at least said part of the item of media content retrieved as a result of said second process: See Claim element 1.B. Further, when the user has selected the "Stream selected files" option, the Gnutella software begins to receive the requested MP3 file from the computer on which the MP3 file is located. (Ex. 1019, ¶¶ 339-341.) This element is thus disclosed and a POSITA would have been guided by those and other teachings to combine the Tutorial and Hacker.



**Element 1.I** output at least said part of the received media content to the second stream target to provide streaming of media content: See elements 1.C and 1.F. (Ex. 1019, ¶¶ 342-343.)

**Element 1.J** wherein the first media player is adapted to output at least a part of the media content to the first stream target while the second media player outputs at least a part of the media content to the second stream target: The Tutorial discloses a P2P network of multiple computers connected to each other via a computer network. (Ex. 1009, ¶¶ 14, 15, 19, 23, Figs. 2,3.). Each peer computer is capable of performing searches for files. *Id.* As discussed above in regard to Claim elements 1.C, 1.G and 1.I, the Tutorial in view of Hacker discloses that a first peer computer on a P2P network may play a media file using Winamp, while a second similarly-equipped (peer) computer on the same P2P network may download the same song and play back using its own Winamp. The Tutorial also shows that the user may have multiple uploads and downloads in progress and simultaneously have multiple outgoing and incoming connections, and teaches that a song may be played using Winamp as it simultaneously downloads. (Ex. 1009, ¶¶ 14, 15, 19, 23, Figs. 2,3.) Therefore, it is disclosed and obvious to a POSITA that an MP3 file could be played on a first computer in the network (output to a first stream target) at the same time that a second computer was downloading and streaming the same song (output to a second stream target). (Ex. 1019, ¶¶ 344-

352.) This element is thus disclosed and a POSITA would have been guided by those and other teachings to combine the Tutorial and Hacker.

**Claim 3:** The system of claim 1, wherein the first media player includes a plurality of output managers for receiving the plurality of further requests: The

5 Tutorial discloses that a user computer with Gnutella software installed on it can simultaneously download and upload multiple files. (Ex. 1009, ¶¶ 6, 7, 14, 15, 23, FIG. 2.) The Tutorial also shows a user interface on which a user can see progress of such data traffic, which is characterized by the IP address of the other computer and a port ID on which the data traffic is occurring. (Ex. 1009, ¶¶ 9, FIG. 2.) Hacker  
10 similarly teaches the use of TCP/IP protocol for streaming. (Ex. 1011, pp. 259, 323.) The combination of the Tutorial's and Hacker's use of TCP protocol would therefore have been obvious to a POSITA and would have produced predictable results. (Ex. 1019, ¶¶ 353-355.) Since Gnutella allows for multiple uploads, each distinguishable based on the combination of the other computer's IP address and a  
15 port ID, it would be obvious that traffic management in the Tutorial is performed by routines or processes in the software for regulating the outgoing content, each routine or process controlling and monitoring data traffic for a given combination of IP address, port ID. (Ex. 1019, ¶¶ 356-357.)

**Claim 4:Element 4.Preamble** A computer-readable medium having  
20 computer executable instructions for sharing streaming media content comprising

data packages, from a first media player playing the media content to a plurality of other media players requesting at least a part of the media content: See Claim element 1.Preamble. In addition, the Gnutella Tutorial teaches that the Gnutella P2P functionality can be downloaded as a software application and installed on a computer. (Ex. 1009, ¶¶ 1-5.) Hacker also discloses that Winamp is a software application that can be installed on a Windows computer. (Ex. 1011, pp. 61-62.) A POSITA would have found it obvious that the combination of the Tutorial's downloadable software and Winamp's installable software application can thus be embodied into a computer-readable medium. (Ex. 1019, ¶¶ 358-360.)

**Element 4.A** said computer executable instructions when running on the first media player comprising instructions that cause the first media player to: receive a play-back request for an item of media content from a component of the first media player, wherein the first media player includes a stream target: See Claim elements 1.A and 4.Preamble. (Ex. 1019, ¶¶ 361-362.)

**Element 4.B** in response to the play-back request, initiate a first process to retrieve at least a part of the item of media content from at least one device selected from the group consisting of the first media player and one or more further devices: See Claim element 1.B. (Ex. 1019, ¶¶ 363-364.)

**Element 4.C** retrieve at least said part of the item of said media content:

Tutorial discloses a computer that is able to download files from peer computers in

the Gnutella network. (Ex. 1009, ¶ 9, 10, 15, 23, FIG. 2.) As disclosed in Hacker and the Tutorial, these files can be MP3 music files. (Ex. 1009, ¶ 23, FIG. 3, FIG. 4), (Ex. 1011, pp. 48-54, 61-65, 122, 152-154, 301-310, 88, 125, 206.) In some cases, these files may represent parts of a large audio file that was split into  
5 multiple tracks. (Ex. 1011, p. 201.) Tutorial discloses that the file transfer may be performed using IP protocol for Internet-based data transfer. (Ex. 1009, ¶¶ 6, 11, 12, 13, 15.) It would have been obvious to a POSITA that typical MP3 files downloaded by users will occupy multiple IP packets. Therefore, a typical user computer, as taught by the Tutorial, would retrieve an MP3 file in multiple parts,  
10 or data packets, *e.g.*, corresponding to individual IP packets. (Ex. 1019, ¶¶ 355-369.) Moreover, retrieving an entire file is the retrieval of at least a part of the item of said media content.

**Element 4.D** pass at least said part of the retrieved media content to the stream target in reply to the play-back request for the item of media content, after  
15 the first media player has received the play-back request: See Claim element 1.C.  
In addition, due to causality, it would have been obvious to a POSITA that the streaming of media content to the Winamp occurs after the user selects the “Stream selected files.” (Ex. 1019, ¶¶ 370-371.)

**Element 4.E** receive from the plurality of other media players a plurality of  
20 further requests for the item of media content that are sent in an uncoordinated

order with respect to one another from the plurality of other media players to the first media player: See Claim element 1.D. (Ex. 1019, ¶¶ 372-373.)

**Element 4.F** in reply to each of the plurality of further requests for the item of media content, transmit from the first media player to a corresponding requesting other media player one or more parts of the item of media content retrieved at the first media player in response to the play-back request: See Claim element 1.E. (Ex. 1019, ¶¶ 374-375.)

**Element 4.G** wherein the one or more parts of the item of media content are transmitted from the first media player to the corresponding requesting other media

player while the first media player passes at least said part of the retrieved media content to the stream target to provide streaming of media content: The Tutorial

discloses Gnutella, a file-sharing software, that can be run on a general purpose computer such as a desktop. (Ex. 1009, ¶¶ 4, 5, 7, 11, 13, 14, 15.) The Tutorial also discloses that MP3 files are received and stored in a local directory (Ex. 1009,

¶¶ 9, 24.) and that files from a local directory are made available for upload to other requesting nodes in the file-sharing network. *Id.* The Tutorial also discloses simultaneous operation of an MP3 player (Winamp) on the same computer and simultaneous file access by the file-sharing software and the player software (listening while downloading and listing it in the playlist). *Id.* The Tutorial further

teaches that an MP3 file can be played while it is being downloaded, or streamed,

so that the playback and download functions occur simultaneously. (Ex. 1009, ¶23, Fig. 3) A POSITA would have understood that the file-sharing software disclosed by the Tutorial is implemented on a computer, *e.g.*, on a Windows based desktop computer, where data stored in the file system is accessible to multiple software functions, *e.g.*, an uploading function and a playlist function, so that an MP3 file could be simultaneously played and uploaded. (Ex. 1019, ¶¶ 376-379.) A POSITA would have understood such concurrent use of data by the Tutorial's software and Hacker's player application at least because prevalent operating systems, *e.g.*, Windows by Microsoft and Unix, were well known to be multi-tasking operating systems that provided simultaneous access to data stored in a data file stored in the file system and also allowed multiple processes being run simultaneously. *Id.* The Tutorial also teaches simultaneous uploading and downloading being done by Gnutella (Ex. 1009, FIG. 2 showing uploading and downloading menus). *Id.* Further, a POSITA would have been motivated to combine the Tutorial with Hacker to allow data upload while data is provided to the player application to benefit from the ability to perform both file sharing and file playback functions without having to block each other. *Id.*

**Claim 5:** The computer-readable medium of claim 4, wherein said instructions further cause the first media player to stream at least said part of the item of media content to at least one of the plurality of other media players using at

least one of a plurality of connections, each connection corresponding to one media

player of the plurality of other media players: The Tutorial discloses the use of

outbound connections from a user computer using a TCP port / IP address

combination, each of which is a separate connection corresponding to one other

5 computer on the network. (Ex. 1009, ¶¶ 6, 11, 12, 13, 15.) The Tutorial's

description of unique connections to one or more computers in the network based

on address is consistent with the disclosure of the 752 Patent that the software

establishes a connection with one or more devices on the network based on

addresses of those devices if the quality of a first connection deteriorates. (Ex.

10 1001, 3:66-4:11, Ex. 1019, ¶¶ 380-385.) Hacker further discloses that the data

transfer can be done by streaming directly to a Winamp player. (Ex. 1011, pp. 5, 6,

51, 61, 63, 64, 88, 89.) A POSITA would have found it obvious that a combination

of the Gnutella software for sharing files disclosed in Tutorial and Hacker, would

result in a connection over which a media file is streamed. *Id.* A POSITA would

15 have been motivated to combine the Tutorial with Hacker's streaming capability to

allow multiple users to share MP3 files. *Id.*

**Claim 6.** The computer-readable medium of claim 4, wherein the

instructions further cause the first media player to receive the plurality of further

requests at a plurality of respective output managers: The Tutorial discloses that

20 any user's computer is a peer in the Gnutella network and therefore can transmit

and receive search requests from many other peers in the Gnutella network. (Ex. 1009, ¶¶ 6, 12, 14, 15, 19, FIG. 2.) A POSITA would have understood that Internet users would request MP3 file sharing at their convenience and liking, and users of the Gnutella network would therefore be performing file sharing in an uncoordinated manner. (Ex. 1009, ¶¶ 6, 12, 14, 15.) Hacker teaches that MP3 file streaming can be performed using the TCP/IP protocol. (Ex. 1011, pp. 259, 323.) A POSITA would further recognize that each request for data from a peer computer would be managed by a separate software resources, *e.g.*, processes or threads, in order to manage the outgoing packet transmission corresponding to that request. (Ex. 1019, ¶¶ 386-389.)

**Element 7.Preamble** A media player for playing streaming media content comprising data packages and transmitting at least a part of the media content to a plurality of other devices, comprising: a data processing unit that executes software having a routine for: *See* Claim element 1.Preamble. The Tutorial teaches that the

Gnutella P2P functionality can be downloaded as a software and installed on a computer. (Ex. 1009, ¶¶ 1-5.) Hacker also discloses that Winamp is a software application that can be installed on a Windows computer. (Ex. 1011, pp. 61-62.) A POSITA would have found it obvious that the combination of the Tutorial's downloadable software and Winamp's installable software application will thus be executed by a processor on a user computer, which is the data processing unit. (Ex.



1019, ¶¶ 390-392.)

**Element 7.A** (i) receiving a play-back request for an item of media content from a component of the media player, wherein the media player includes a stream target: See Claim element 1A. (Ex. 1019, ¶¶ 393-394.)

5 **Element 7.B** (ii) in response to the play-back request, initiate a first process to retrieve at least a part of the item of media content from at least one device selected from the group consisting of the media player and one or more further devices: See claim element 1.B. (Ex. 1019, ¶¶ 395-396.)

**Element 7.C** a retrieving routine for retrieving said item of media content:  
10 *See Claim element 4.C. In addition, a POSITA would have found it obvious to implement the task of retrieving into a software routine. (Ex. 1019, ¶¶ 397-398.)*

**Element 7.D** a management routine for managing retrieved media content:  
The Tutorial discloses management of received MP3 files because, upon reception, a user computer stores the file to a directory and shares the files with other  
15 computers. (Ex. 1009, ¶¶ 14, 15, 20, 23, 24, FIG. 4.) The Tutorial also discloses a user interface that displays information about the media files being transferred to and from other users, including, routing errors, upload count, download count, IP addresses and (OSI layer 4) port IDs to which files are being sent. (Ex. 1009, ¶9, FIG. 2.) A POSITA would have found it obvious to embody this task into a  
20 software routine, which is the management routine. (Ex. 1019, ¶¶ 399-401.)

**Element 7.E** a routine for making at least a part of the retrieved media content ready for play-back and providing the media content made ready for play-back to the stream target: *See* Claim element 1.C. The Tutorial teaches using IP packets for media file reception. (Ex. 1009, ¶¶ 12, 13.) A POSITA would have found it obvious that the user's computer makes the received MP3 files ready for playback during the "Stream selected files" operation by removing IP headers, and buffering the media data for playback. A POSITA would have found it obvious to embody this task into a software routine. (Ex. 1019, ¶¶ 402-403.)

**Element 7.F** a request handling routine for handling, after receiving the play-back request, a plurality of further requests for at least a part of the item of media content, wherein the plurality of further requests are sent from a plurality of other devices in an uncoordinated order with respect to one another to the media player: *See* Claim element 1.D. A POSITA would have found it obvious to embody this task into a software routine. (Ex. 1019, ¶¶ 404-405.)

**Element 7.G** a transmitting routine for streaming at least a part of the item of media content to each of the plurality of other devices: *See* Claim elements 1.C. and 1.G. In addition, when a requesting computer in the Gnutella network requests an MP3 file by using the "Stream selected files" option, the sender computer that transmits the requested MP3 file to the requesting computer, is streaming the requested MP3 file because the requesting computer plays back the MP3 file while

being downloaded. (Ex. 1019, ¶¶ 406-407.) A POSITA would have found it obvious to embody this task into a software routine. *Id.*

**Element 7.H** wherein the media player includes software that, when running on the media player allows the media player to: receive at least the play-back

5 request for the item of media content from the component of the media player: See  
Claim element 1.F. Ex. 1019, ¶¶ 408-409.

**Element 7.I** in response to the play-back request, initiate the first process to retrieve at least a part of the item of media content from at least one device

selected from the group consisting of the media player and the one or more further

10 devices: See Claim elements 1.B and 1.G. It would have been obvious to a

POSITA that, due to causality, the streaming of a requested MP3 file is initiated in response to the user request to “Stream selected files.” (Ex. 1019, ¶¶ 410-411.)

**Element 7.J** retrieve at least a part of the item of media content from at least one device selected from the group consisting of the media player and the one or

15 more further devices: See Claim elements 1.B, 1.E and 4.C. Ex. 1019, ¶¶ 412-413.

**Element 7.K** manage the retrieved media content: See Claim element 7.E.

Further, the Tutorial teaches that management functions are performed on files being received by a peer computer, including maintaining statistics about messages, packets sent, packets received, lost or dropped packets, etc. (Ex. 1009, Fig. 2, ¶15.)

20 This element is thus disclosed and a POSITA would have been guided by those

and other teachings to combine the Tutorial and Hacker. Ex. 1019, ¶¶ 414-415.

**Element 7.L** make the retrieved media content ready for play-back: The Tutorial discloses retrieval of media files by a computer node in a P2P network (Ex. 1009, ¶¶ 8, 19, 20, 23, 24, FIG. 4) and the use of a Winamp player application to play back MP3 files (*Id.*, ¶¶ 1,6,7,14,16-24, FIG. 4.). Hacker provides details about Winamp player operating upon a received MP3 file and making the file ready for play-back by, *e.g.*, adding the file to the playlist, indicating playback of the MP3 file, and reading header field tags for presenting title, artist, and album information to the user. (Ex. 1011, pp. 48-54, 61-65, 122, 152-154, 301-310, 88, 125, 206.) A POSITA would recognize that the user friendly playback audio interface features of Winamp taught by Hacker could be utilized in a computer utilizing the file sharing of the Tutorial according to known methods with predictable results, particularly in light of the explicit teaching in the Tutorial to combine Winamp with Gnutella. (Ex. 1019, ¶¶ 416-418.)

**Element 7.M** provide at least a part of the retrieved media content made ready for play-back to the stream target: Claim element 7.E. Ex. 1019, ¶¶ 419-420.

**Element 7.N** receive, from the plurality of other devices in any order, a plurality of requests for the item of media content after having retrieved at least a part of the item of media content at the media player: *See* Claim elements 1.D and

7.F. Ex. 1019, ¶¶ 421-422.

**Element 7.O** in reply to a request for the item of media content from a first of the plurality of other devices, stream from the media player to the first device at least a part of the item of media content retrieved at the media player in response to the request from the component of the media player for the media content while the media player is providing media content made ready for playback to the stream target: See Claim elements 1.G, 4.G and 7.G. As discussed in Claim elements 1.G and 7.G, due to symmetry of operation, when a requesting Gnutella peer node operates to “Stream selected files,” the corresponding file is being streamed to the requesting Gnutella peer node from the Gnutella peer node on which the file is available. (Ex. 1019, ¶¶ 423-425.)

**Claim 8.** The media player of claim 7, wherein the software, when running on the media player, further allows the media player to stream one or more data packages to the plurality of other devices, using a plurality of connections, each connection corresponding to one of the plurality of other devices. In addition to reasons provided for Claim 7, the Tutorial shows that a user computer with Gnutella software can simultaneously allow multiple uploads, using multiple connections, each connection having a connection with its own IP address and (OSI layer 4) port ID. (Ex. 1019, ¶¶ 426-429.) Hacker additionally provides the teaching that when an MP3 file is being played back on a computer, processor utilization is relatively low. (Ex. 1011, Fig. 4.11, pp. 154-155.) The combination of

Hacker's teaching with the Tutorial's teaching of multiple uploads would have thus provided predictable results and a reasonable expectation of success. *Id.*

**Claim 10.** The media player of claim 7, wherein said retrieving of said item of media content by said processing unit and streaming from the media player to the first device at least a part of the item of media content occur simultaneously.

The Tutorial teaches that a user may stream (listen to) selected files while the files downloads, i.e. playback and downloading occur simultaneously. (Ex. 1009, ¶23, Fig. 3.) A POSITA would have understood that the file-sharing software disclosed by the Tutorial is implemented on a computer, *e.g.*, on Windows based desktop computer or a Unix workstation, where data stored in the file system is accessible to multiple software functions, *e.g.*, an uploading function and a playback function. (Ex. 1019, ¶¶ 430-436.) A POSITA would have understood such concurrent use of data by the Tutorial's software and Hacker's player application at least because prevalent operating systems, *e.g.*, Windows by Microsoft and Unix, were well known to be multi-tasking operating systems that provided simultaneous read access to data stored in a data file stored in the file system. *Id.* A POSITA would have further understood that both the file uploading application and the playback operation do not attempt to make any changes to the content of the media file and therefore do not perform conflicting operations on the same memory location and thus are readily implemented simultaneously with each other. *Id.* Hacker

additionally provides the teaching that when an MP3 file is being played back, and multiple other tasks are simultaneously being performed on a computer, the processor utilization is relatively low at 15%. (Ex. 1011, Fig. 4.11, pp. 154-155.)

The combo of Hacker with Tutorial's teaching of multiple uploads would have provided predictable results and a reasonable expectation of success. *Id.* The Tutorial provides an explicit teaching to combine the download feature with the streaming playback of Winamp, which is described by Hacker (Ex. 1009, ¶23, Fig. 3; Ex. 1019, ¶¶ 435-436.)

**Claim 14.** The media player of claim 7, wherein the management routine places the media content retrieved in a buffer from which at least a part of the item of media content is being transmitted to at least one of the plurality of other devices: The Tutorial discloses that files are uploaded from a user computer using IP communication protocol (a destination IP address and a port number.) (Ex. 1009, ¶¶ 6, 9, 11, 12, 13, 15, FIG. 2.) A POSITA would have understood that a buffering operation is performed in a computer in which data is copied from a file into a buffer that is used by the transmission circuitry, *e.g.*, a TCP implementation stack on the computer. (Ex. 1019, ¶¶ 437-440.) This claim element would therefore have been obvious to POSITA in view of Tutorial. A POSITA would have also recognized that buffering was routinely used by computers when reading or writing files to reduce disk access delays, and thus buffering would have been used in the

combination of the Tutorial with Hacker for streaming of media files to listen to the files using Winamp while being downloaded. *Id.*

**Claim 15.** The media player of claim 7, wherein the request handling routine further includes a routine for handling the plurality of further requests at a plurality of respective output managers. *See* discussion for Claim 3. Ex. 1019, ¶¶ 441-443.

**Claim 16: Element 16.Preamble.** A computer-readable medium having computer executable instructions for sharing streaming media content comprising data packages, from a first media player playing the media content to a plurality of other media players requesting at least a part of the media content: *See* Claim elements 1.B and 4.Preamble. A POSITA would have been guided by those and other teachings to combine Tutorial and Hacker. Ex. 1019, ¶¶ 444-446.

**Element 16. A** said computer executable instructions when running on the first media player comprising instructions that cause the first media player to: receive a play-back request for an item of media content from a component of the first media player, wherein the first media player includes a stream target: *See* element 1.A. Ex. 1019, ¶¶ 447-448.

**Element 16.B** in response to the play-back request, initiate a first process to retrieve at least a part of the item of media content from at least one stream source selected from the group consisting of the first media player and one or more further devices: *See* Claim element 1.B. Ex. 1019, ¶¶ 449-450.



**Element 16.C** retrieve at least said part of the item of said media content from the stream source: *See* Claim element 4.C. Ex. 1019, ¶¶ 451-452.

**Element 16.D** manage the retrieved media content by using a buffer: *See* Claim element 7.D. Further, Hacker discloses that prevalent MP3 player software used buffers in which file data is stored during playback. (Ex. 1011, pp. 6, 38, 41, 69, 142, 144, 151, 159, 163, 175, 204, 219, 366.) A POSITA also knew that buffers are used by Windows and UNIX operating systems to reduce performance hits due to disk access. Ex. 1019, ¶¶ 453-454.

**Element 16.E** get a part of the retrieved media content from the buffer and make that part ready for play-back and pass it to the stream target in reply to the play-back request for the item of media content from the stream target to provide streaming of media content: *See* Claim elements 16.D and 7.E. *Id.*, ¶¶ 455-456.

**Element 16.F** after the first media player has received the playback request, receive from-the a plurality of other media players, a plurality of further requests, sent in an uncoordinated order with respect to one another, for the item of media content at one or more output managers, each output manager corresponding to one of the plurality of media players: *See* Claim element 1.D. Ex. 1019, ¶¶ 457-458.

**Element 16.G** in reply to each of the plurality of further requests for the item of media content, transmit from the corresponding output manager one or more parts of the item of media content: Claim element 1.E. Ex. 1019, ¶¶ 459-460.

**Element 16.H** wherein the buffer is both used by the first media player to make a part of the retrieved media content ready for play-back and by at least one output manager to get a section of the retrieved media content in order to transmit the section of the retrieved media content to the corresponding requesting other

5 media player: *See* Claim elements 16. D, 7.E, 7.L, 7.M. Furthermore, a POSITA knew that Unix and Windows file systems used buffers to store data during use by software applications to overcome slow access time and wear and tear associated with frequent hard drive accesses, Ex. 1019, ¶¶ 461-462. A POSITA would have therefore found it obvious that two software operations being performed on the  
10 same operating system; viz., media playback and data transmission to requesting computer; would use data from the same buffer rather than accessing data separately from the hard drive. *Id.*

**B. GROUND 2: CLAIMS 2, 11-13 ARE OBVIOUS OVER GNUTELLA TUTORIAL, HACKER AND SCHUSTER**

15 Schuster discloses a PID for receiving media content and playing it back to a user. (Ex. 1050, Abstract.) The PID can be a PDA, a phone or a network telephone. (*Id.*, Title, Abstract, 7:43-61, 9:20-36, 9:56 - 10:7, 14:9-15, 18:4-22, 20:49-63.) A POSITA would have realized that, similar to the Gnutella Tutorial and Hacker, Schuster also teaches the use of a media player application for playing back media.  
20 *Id.* A POSITA would have been guided or motivated by the express teaching of the use of Media Player application in Hacker and Schuster and would have a

reasonable expectation of success based on the teaching of the Tutorial, Hacker and Schuster and general information at the time. (Ex. 1019, ¶¶ 463-464.)

**Claim 2:** The system as recited in claim 1, wherein furthermore the first media player is connected through a wireless connection to at least one of said

5 plurality of other media players. Hacker, as discussed in combination with Tutorial, in regard to Claim 1, further teaches that some MP3 players can connect with other MP3 players through a wireless network connection. (Ex. 1011, pp. 244, 258.)

Further, Schuster's PID includes a media player, and is wirelessly connected to the Internet. (Ex. 1050, 11:47-55.) The Tutorial discloses a P2P network in which

10 multiple devices are connected to each other via a P2P connection. (Ex. 1009, ¶¶ 6, 8.) The combination of Schuster and the Tutorial thus discloses the claimed feature.

(Ex. 1019, ¶¶ 465-469.) A POSITA would have recognized at the time that

wireless Internet connections such as those disclosed in Schuster could have been used to transmit and receive a wide range of data using the same protocols (*e.g.*

15 TCP/IP) that would be used for a wired connection, and therefore could have been used to transmit the necessary data for the P2P network of the Tutorial. A POSITA

would have recognized the benefits of combining the wireless portable information device of Schuster with the P2P network of the Tutorial and player application of

Hacker, using known methods with predictable results, to extend media sharing to

20 additional Internet-connected devices. *Id.*

**Claim 11:** The media player of claim 7, wherein the media player is part of mobile communication equipment: Hacker, as discussed in combination with the Tutorial, in regard to Claim 7, further teaches that some handheld computers, cell phones, PDAs, which are mobile communication equipment, are able to play MP3 audio files. (Ex. 1011, pp. 7, 209, 230-233.) Schuster discloses a PDA that is wirelessly connected to the Internet and is equipped with media playback capability. (Ex. 1050, 8:21-34; 9:19-26,11:47-55.) A user of this device is mobile. (*Id.*, 11:47-55, 13:1-8.) The combination of Schuster's mobile device with the Tutorial's P2P network and Hacker's media playback application thus renders the claim obvious. (Ex. 1019, ¶¶ 470-473.) A POSITA would have combined Schuster's mobile device with the P2P network disclosed by the Tutorial and the player application disclosed by Hacker to provide media sharing capability to a variety of users, for the reasons discussed in regard to Claim 2. *Id.*

**Claim 12:** The media player of claim 7, wherein the media player is part of a pocket computer: Hacker, as discussed in combination with Tutorial, in regard to Claim 7, further teaches that some handheld computers, cell phones, portable digital assistants, *e.g.*, a PalmPilot, all of which are "pocket computers" and are able to play MP3 audio files. (Ex. 1011, pp. 7, 209, 230-233.) Schuster discloses a PDA and a wireless phone that is wirelessly connected to the Internet and is capable of media playback. (Ex. 1050, 8:21-34; 9:19-26,11:47-55.) A POSITA

would have understood a PDA to be a pocket computer. (Ex. 1019, ¶¶ 474-478.)

The combination of Schuster's PDA with the Tutorial's P2P network and Hacker's media playback application thus renders the claim obvious. *Id.* A POSITA would have combined Schuster's PDA with the P2P network disclosed by the Tutorial and the player application disclosed by Hacker to provide media sharing capability to a variety of users, for the reasons discussed in regard to Claim 2. *Id.*

**Claim 13:** The media player of claim 7, wherein the streaming comprises wireless streaming: As discussed in Claims 11 and 12, a POSITA would have understood the combination of the Tutorial, Hacker and Schuster as disclosing a user computer with wireless connection being able to receive MP3 files, streamed to the user's computer, *e.g.*, the Winamp player. (Ex. 1019, ¶¶ 479-481.)

**C. GROUND 3: CLAIM 9 IS OBVIOUS OVER GNUTELLA TUTORIAL, GNUTELLA PROTOCOL AND HACKER**

In the same field of P2P file sharing as the Gnutella Tutorial, the Gnutella Protocol specifies the format of messages exchanged among Gnutella peers. (Ex. 1010, ¶¶ 7-34.) The Gnutella Protocol specifies the format for exchange of messages by the same Gnutella program that is described by the Tutorial and thus provides an explicit suggestion to combine the teachings of the two references. A POSITA designing a P2P file sharing program would understand that computers using Gnutella would need to communicate using a defined protocol for network messages, and would therefore naturally look to Gnutella Protocol to solve that

issue. Further, combining the Gnutella program described by the Tutorial with the Gnutella Protocol would be a straightforward combination of the prior art elements described by each reference according to their intended purpose using known techniques to yield predictable results with a reasonable expectation of success, further using known features in the same or similar technical field. It would have been obvious, therefore, to use the protocol of Gnutella Protocol with the Gnutella program of the Tutorial. (Ex. 1019, ¶¶482-483.)

**Claim 9:** The media player of claim 7, wherein the plurality of requests received from the plurality of other devices for at least a part of the item of media content further comprise a plurality of specific requests received from the plurality of other devices for at least a part of the item of media content: Gnutella Protocol states that Gnutella can use the HTTP Range parameter so that “interrupted downloads may be resumed at the point at which they terminated.” (Ex. 1010, ¶34.)

A POSITA would be familiar with HTTP’s Range header, which allows a requesting device, such as a Gnutella peer node operating the program described in the Tutorial (Ex. 1009, Fig. 3, ¶23) to specify a specific subset of a media file, that is to be downloaded. (Ex. 1019, ¶¶484-485.) It is therefore disclosed and obvious that the Tutorial program would detect interrupted downloads and restart them at the point that they were discontinued, which would require using the HTTP Range parameter to request a specific part of the content, *e.g.*, the un-transferred portion.

*Id.* This element is thus disclosed and a POSITA would have been guided by those and other teachings to combine the Tutorial, Hacker and Gnutella Protocol.

**D. GROUND 4: CLAIMS 1, 3-10 AND 14-16 ARE OBVIOUS OVER CLARKE AND EHRMAN**

5           Clarke teaches a P2P network technology, called Freenet, which provides the same P2P technology as in the 752 Patent. Freenet operates as a network of identical nodes and allows users to share files via the network. (Ex. 1012, Abstract, ¶¶ 1, 10.) A software implementation of Freenet was freely available for others to download and use. (*Id.*, ¶¶ 4, 10.) Clarke explicitly discusses the usefulness of  
10   Freenet to P2P distribution of large files by splitting them into parts. (*Id.*, ¶ 40.) In the same field of file sharing, using a similar P2P file sharing technology in Clarke, Ehrman discloses sharing of music/video files (Ex. 1013, 1:8-13, 2:1-8, 4:10-14, 9:7-12) and teaches that video files tended to be large and inconvenient because downloading of video files over then-existing connection speeds resulted in  
15   network congestion, or required large bandwidths (*Id.*, 1:1-10, 2:1-8.) To remedy this, Ehrman provides a technique by which video files are divided into parts, called streams, and downloaded to a requester from peer computers. (*Id.*, 4:10-14.)

Both Clarke and Ehrman recognize bandwidth and storage problems associated with download of large files, which could be media files (Ex. 1012, ¶¶  
20   40, 41; Ex. 1013, 1:8-13, 2:1-8, 4:15-18.) A POSITA would have recognized that Clarke and Ehrman provide similar solutions (splitting a file in multiple parts for

downloading) to a similar problem of downloading a large file over a peer to peer network and would have thus combined Clarke with Ehrman for downloading video files. (Ex. 1019, ¶¶500-503.) A POSITA would have further realized that Clarke's feature of dividing large files into smaller subparts made them convenient  
5 for downloading to requesting computers and uploading from source computers and would alleviate upstream/downstream bandwidth problems associated with downloading video files, as discussed by Ehrman. *Id.* A POSITA would have thus readily adopted Clarke's file sharing technology for use by Ehrman's video file distribution network. *Id.* A POSITA would have a reasonable expectation of  
10 success in combining teachings of Clarke and Ehrman because both Clarke and Ehrman teach similar techniques of splitting large files into multiple parts and downloading them over a computer network. *Id.* A POSITA would have downloaded Clarke's freely available software as documented in Clarke for use with Ehrman's teaching. *Id.* Further, Ehrman explicitly teaches usefulness of such  
15 technique to video file downloading and playback. *Id.* A POSITA would have also recognized that Clarke's implementation of P2P network formation and request-resolution for forwarding P2P queries by consistent hashing, was freely available via a publicly accessible website (Ex. 1012, ¶¶ 4, 35-40) A POSITA would have combined Clarke's file sharing technology with Ehrman's video file downloading  
20 and a content display player for downloading and viewing video over the Internet.



*Id.*

The following analyses and discussions of Clarke and Ehrman in connection with each claim on an element-by-element basis provide additional reasons or motivations that would cause a POSITA to combine the Clarke and Ehrman in the manner as suggested in this Petition. For claim text, refer to Grounds 1 and 3.

**Claim 1: Element 1.Preamble:** Clarke teaches a P2P network for distributing files to user computers over a network. (Ex. 1012, Abstract, ¶¶ 2, 9, 10.) Ehrman discloses a P2P system in which multiple peer computers, one of which is “a first media player” and another one of which is “a second media player,” can share video files and playback received content using a playback application. (Ex. 1013, 1:9-11, 2:9-12, 4:1-6, 4:15-25, 5:11-18.) It would have been obvious to a POSITA to combine Ehrman’s peer computers that are capable of content playback with Clarke’s file-sharing technology to benefit from being able to share large video files over the Internet. (Ex. 1019, ¶¶504-507.) A POSITA would have been guided by prior art teachings to combine Clarke and Ehrman.

**Element 1.A:** Ehrman teaches that a peer computer, whose user desires to receive a particular video (which is the first media player), sends a request for downloading the video. (Ex. 1013, 5:19-22, 9:4-12.) A POSITA would have understood that the video-watching request would be received at the receiving peer computer from its user interface. Ehrman teaches that the receiving peer computer

would be equipped with a content play back capability, *e.g.*, a player application (which is the first stream target). A POSITA would have combined this teaching of Ehrman with Clarke's file sharing network to find and download the user-requested content from the Internet. (Ex. 1019, ¶¶508-510.)

5           **Element 1.B:** Ehrman teaches that, upon receiving a user request for a video download, a streaming content manager conveys a list of other peer computers from which to obtain the video (supplying peers for that video), to the requesting peer computer. Ehrman discloses that a receiving peer, *i.e.*, a peer computer that retrieves a media file, sends a request for downloading a content stream, such as a  
10 movie. (Ex. 1013, 9:4-12.) It would have been obvious that the requesting peer, which is the first media player, would then initiate a process to retrieve the video or movie requested by the user. (Ex. 1019, ¶¶511-514.) Clarke teaches that peer computers can query files from other peer computers and retrieve data files, when found. (Ex. 1012, ¶¶12, 13.) A POSITA would have combined teachings of  
15 Ehrman and Clarke to enable video and movie file searches by a user (peer computer) and downloading files found as a result of the searches over a P2P network. *Id.*

**Element 1.C:** Ehrman discloses that a requested stream content is displayed by a content display unit (player application) which is the first stream target,  
20 generally as it arrives. (Ex. 1013, 5:12-16.) Ehrman also discloses that the received

chunks of media content are combined by stripping of any transmission packaging and made ready to be played on the content display as they arrive. (Ex. 1013, 8:4-9, Ex. 1019, ¶¶515-518.) Clarke teaches that files can be downloaded and locally stored for use at the local computer upon retrieval. (Ex. 1012, ¶¶ 13, 20-27.) A POSITA would have thus understood that files retrieved from the P2P network, as taught by Clarke, are then outputted to a content playback device, as taught by Ehrman, to allow a user to enjoy the downloaded content. *Id.* This element is thus disclosed and a POSITA would have been guided by the teachings to combine Clarke and Ehrman.

**Element 1.D:** Clarke teaches that peer nodes query each other for data file retrieval (Ex. 1012, Abstract, ¶¶ 9, 13, 14) and that the peers may be in geographically different locations. (*Id.*, ¶17.) A POSITA would have understood file query and retrieval activity by different users of different computers at different locations would generally be independent of each other, and thus uncoordinated from each other (Ex. 1019, ¶¶519-521) and would have thus found it obvious that a peer computer that requests and receives media content, as taught by Ehrman, combined with the P2P network, as taught by Clarke, would thereafter receive requests for that and other media data files from other peer computers. *Id.*

**Element 1.E** Clarke discloses that nodes in the P2P network share files with each other (Ex. 1012, Abstract, ¶¶ 9, 13, 14.) and that one goal of shared file

serving is to store files for a long time. ( *Id.*, ¶¶ 40.) A POSITA would have understood that nodes in the P2P network not only are able to retrieve files from other nodes and locally store them, but also make locally stored files available to other nodes. (Ex. 1019, ¶¶522-525.) When a file is sent from one node to another,  
5 it is copied and locally stored at nodes along the way. *Id.* The combination of Clarke and Ehrman thus shows that a receiving node can make its content available to future requests from other user computers. A POSITA would have combined Clarke with Ehrman to enable eternal storage and redistribution of video files, as disclosed in Clarke and Ehrman. *Id.*

10       **Element 1.F:** *See* Claim element 1.A. The “second media player” is simply another peer computer in the network equipped with similar software. (Ex. 1019, ¶¶526-527.)

**Element 1.G:** *See* Claim elements 1.B , 1.Preamble and 1.F. Ehrman also discloses that the process of video retrieval is started by presenting a list of  
15 available videostreams to a user and receiving the customer’s choice. (Ex. 1013, 9:7-12.) Then, the list of supplying peers is provided for the videostream selected by the customer. *Id.* Further, Clarke describes splitting large files into multiple parts and allowing them to be found and received in the P2P network. (Ex. 1012, ¶40.) It would have been obvious that once the first media player has retrieved a  
20 given videostream, or a portion thereof, as described in step Claim 1.B, then the

first media player would be included in the list of peer nodes from which the current list of the videostream can be satisfied. (Ex. 1019, ¶¶528-529.)

**Element 1.H:** *See* Claim elements 1.B and 1.F. Ehrman also discloses that a requested videostream is provided to the requesting peer node using a list of supplying peers, each of whom as a part of the videostream. (Ex. 1013, 9:7-18, 5:19-6:5, 8:20-23.) Thus, Ehrman teaches that the second requesting peer will receive a part of the requested videostream. (Ex. 1019, ¶¶530-531.)

**Element 1.I:** *See* Claim elements 1.C and 1.F. Ex. 1019, ¶¶532-533.

**Element 1.J:** As discussed above in regard to Claim elements 1.C, 1.G and 1.I, in the combination of Clarke and Ehrman, Clarke discloses a first requesting peer may display retrieved video while a second requesting peer may retrieve the same video and display it using its content display unit. Ehrman shows that a peer computer may, at the same time, be both downloading and uploading content. (Ex. 1013, 6:3-5, 2:9-12, 4:1-17.) A POSITA would have thus found it obvious that two peer computers in the network would display their own locally retrieved video or movie files on their local players at the same time. (Ex. 1019, ¶¶534-536.)

**Claim 3:** Clarke discloses that computers in a P2P network are connected via Internet with other user computers and respond to request messages from other nodes. (Ex. 1012, Abstract, ¶¶ 9, 10, 13.) Each request message includes a transaction ID, which is a unique ID. (*Id.* ¶¶ 28, 5.) The requesting node and the

local node exchange messages as defined by the Freenet protocol. (*Id.* Abstract, ¶¶ 9, 10, 13.) Each node is configured to accept requests from multiple other nodes at a given time. (*Id.*, Abstract, ¶¶ 9, 10, 28.) It would have been obvious that Clarke's peer node teaches using multiple simultaneously-communicating software processes, which would be the output managers, each assigned a unique OSI layer-4 (TCP or UDP) port. (Ex. 1019, ¶¶537-540.) A POSITA would have combined Clarke's software resources to manage output data traffic with Ehrman to monitor the download of video files to a user. *Id.*

**Claim 4: Element 4.Preamble:** *See* Claim element 1.Preamble. Clarke teaches that the Freenet P2P network was implemented in software and be downloaded as a free software from a web site. (Ex. 1012, ¶ 4.) Ehrman's P2P network for downloading and displaying video files is taught to be implemented on a computer. (Ex. 1013, 2:4-12, 4:2-6, 4:15-18, 5:8-11.) A POSITA would have found it obvious that Clarke's downloadable software and Ehrman's teaching of providing a computer with video downloading capability can thus be embodied into a computer-readable medium. (Ex. 1019, ¶¶541-543.)

**Element 4.A, 4B:** *See*, respectively Claim elements 1.A and 4.Preamble and 1B. (Ex. 1019, ¶¶544-547).

**Element 4.C:** Clarke discloses that a node in the P2P network, after its desired file has been found at another node, can receive data of the file using a

packet-based transport mechanism such as TCP or UDP. (Ex. 1012, ¶¶ 28, 5.)

Clarke therefore discloses this claim element. (Ex. 1019 ¶¶548-549.)

**Elements 4.D, 4.E, 4.F** : *See*, respectively, Claim elements 1.C; 1.D and 1.E.  
Ex. 1019, ¶¶550-555.

5        **Element 4.G**: Clarke discloses the operation of a node computer in a file  
sharing network. (Ex. 1012, Abstract, ¶¶ 9, 10, 13.) After a node receives a file via  
the Freenet file sharing network, the node makes the file available to other nodes  
for upload. (Ex. 1012, ¶¶ 9, 10, 13.) At the same time, the file is locally stored. (Ex.  
1012, ¶¶ 20-23.) As explained by Dr. Kesidis, a POSITA would have understood  
10    that the file sharing protocol taught by Clarke is implemented on a computer, *e.g.*,  
on a Windows-based desktop computer or a Unix machine, where data stored in  
the file system is accessible to multiple software functions, *e.g.*, an uploading  
function and a playback function, so that a content file could be simultaneously  
played and uploaded. (Ex. 1019, ¶¶556-560.) A POSITA would have understood  
15    such concurrent use of data by Clarke's software and Ehrman's player, *e.g.*, a  
Microsoft Media Player, at least because prevalent operating systems, *e.g.*,  
Windows by Microsoft and Unix, were well known to be multi-tasking operating  
systems that provided simultaneous read access to data stored in a data file stored  
in the file system. *Id.* A POSITA would have been motivated to combine Clarke  
20    with Ehrman to allow data upload while data is provided to the player application

to benefit from ability to perform both video sharing and video playback functions without having to block each other. *Id.*

**Claim 5:** Clarke discloses the use of outbound connections from a user computer using a TCP port / IP address combination. (Ex. 1012, ¶¶ 28, 5.) Clarke's description of unique connections to one or more computers in the network based on address is consistent with the disclosure of the 752 Patent that the software establishes a connection with one or more devices on the network based on addresses of those devices if the quality of a first connection deteriorates. (Ex. 1001, 3:66-4:11, Ex. 1019, ¶¶562-564.) Ehrman discloses that the data transfer can be done by streaming media to a peer computer (that requests it) having a content display unit for playing back the content. (Ex. 1013, Title, 1:5-6, 2:1-8, 3:5-6, 5:12-16, 6:6-10, 7:18-22, 8:4-9, 8:14-23.) A POSITA would have found it obvious that the peer computer from which the requesting computer is receiving the content is therefore streaming the content to the requesting computer (Ex. 1019, ¶¶562-564) and would have combined Clarke's teaching of TCP based uploading with the P2P network of Ehrman, to benefit from being able to view video as it is being downloaded, as explicitly taught by Ehrman. (Ex. 1013, 1:3, Ex. 1019, ¶¶562-564.)

**Claim 6:** Clarke discloses that all nodes in the file sharing network are peers, or equals, and are able to both receive and transmit files from many other peer nodes in the network. (Ex. 1012, Abstract, ¶¶ 9-11.) A peer computer operating in



Clarke's P2P network would therefore receive multiple data retrieving requests from many other peer nodes in the network. (Ex. 1019, ¶¶565-568.) A POSITA would further recognize that each request for data from another peer computer in the P2P network would be managed by separate software resources or routines

5 running in software in order to manage the outgoing packet transmission corresponding to that request, and perform tasks such as file segmentation, routing table, etc. (Ex. 1012, ¶¶ 13, 18, Ex. 1019, ¶¶565-568.)

**Claim 7: Element 7.Preamble:** *See* Claim element 1.Preamble. Clarke teaches that the Freenet P2P network can be implemented in software and  
10 downloaded as a free software from a web site. (Ex. 1012, ¶ 4.) Ehrman's P2P network for downloading and displaying video files can be implemented on a PC. (Ex. 1013, 2:4-12, 4:2-6, 4:15-18, 5:8-11.) A POSITA would have found it obvious that the combination of Clarke's Freenet software and Ehrman's media player software application will thus be executed by a CPU on a user computer, which is  
15 the data processing unit. (Ex. 1019, ¶¶569-571.)

**Elements 7.A, 7B:** *See*, respectively, Claim elements 1.A and 1.B. Ex. 1019, ¶¶572-575.

**Element 7.C:** Claim element 4.C. A POSITA would have found it obvious to embody the task of retrieving into a software routine. Ex. 1019, ¶¶576-577.

20 **Element 7.D:** Clarke discloses that a peer computer allows other computers

to search for files on the user computer and uploads the files to other computers, when requested by the other computers. (Ex. 1012, Abstract, ¶¶ 9-11.) Ehrman further discloses that the files can be media files, such as video files. (Ex. 1013, 1:8-13, 2:1-8, 4:15-18.) Clarke discloses several management functions being performed on the retrieved file, including computing a hash key and updating a routing table. (Ex. 1012, ¶¶ 12, 13, 19, 20, 23, 27, 36, 38-40, 47, 52.) A POSITA would have combined the management functions of Clarke with Ehrman's video content sharing P2P network to manage video content distribution in the network. (Ex. 1019, ¶¶578-581.)

10           **Element 7.E:** *See* Claim element 1.C. (Ex. 1019, ¶¶582-583.)

**Element 7.F:** *See* Claim element 1.D. In addition, a POSITA would have found it obvious to embody this ask into a software routine. (Ex. 1019, ¶¶584-585.)

**Element 7.G:** *See* Claim elements 1.C and 1.G. Furthermore, Ehrman teaches that the requesting peer node receives the requested video stream as a number of segments, with each segment being divided into chunks, or smaller portions of 4K to 10K bytes for transmission. (Ex. 1013, 6:11-25.) Each of these portions are transmitted by the transmitting peers as a stream of video. (Ex. 1013, Title, 1:5-6, 2:1-3, 3:5-6, 4:19-25, 5:2-7, 5:12-16, 5:19-25, 6:3-10, 6:19-25, 7:-7, 6:11-25; Ex. 1019, ¶¶586-587.)

20           **Element 7.H:** *See* Claim element 1.F. Ex. 1019, ¶¶588-589.

**Element 7.I:** *See* Claim elements 1.B and 1.G. In addition, it would have been obvious that, due to causality, the process of streaming a requested MP3 file is initiated in response to the user request to view a video (Ex. 1019, ¶¶590-591.)

**Element 7.J:** *See* Claim element 1.B. In addition, it would have been  
5 obvious that, due to causality, the process of streaming a requested MP3 file is initiated in response to the user request to “Stream selected files.” (Ex. 1019, ¶592.)

**Element 7.K:** *See* Claim element 7.E. Clarke teaches performing data management function on the received files, including, *e.g.*, hash key table management and updating. (Ex. 1012, Section 5.) Ehrman teaches checking of  
10 received data, recombining for playback, removing of transmission packaging, and making section ready for playback. Ex. 1013, 8:4-9; Ex. 1019, ¶¶593-594.

**Element 7.L:** Both Ehrman and Clarke disclose operations performed on received files to make the files usable at the receiving node computer, which is similar to the functionality described in the 752 Patent (Ex. 1001, 8:38-42; Ex.  
15 1012, ¶¶ 12, 13, 19, 20, 23, 27, 36, 38-40, 47, 52), (Ex. 1013, 8:1 - 9:16.) A POSITA would have readily combined these features of Clarke and Ehrman to make a received file available for user enjoyment. (Ex. 1019, ¶¶595-597.)

**Elements 7.M, 7.N and 7.O:** *See*, respectively, Claim element 7.E; Claim elements 1.D and 7.F; and *See* Claim element 4.G. (Ex. 1019, ¶¶598-603.)

**Claim 8:** *See* Claim element 5. (Ex. 1019, ¶¶604-606.)  
20

**Claim 9:** Clarke in view of Ehrman teaches the operation of the media

player in Claim 7. Both Clarke and Ehrman disclose that large files, *e.g.*, video files, can be split into multiple segments, with each segment divided into smaller “chunks” for transmission. (Ex. 1012, ¶¶ 40, 41; Ex. 1013, 5:12-16, 6:3-5, 15-18; 5 8:4-9:3.) The information about which transmitting peers have which segments is provided to the requesting peer via a list. (*Id.* 5:22-6:2.) This is a listing of parts of the requested item of media content. (Ex. 1019, ¶¶ 607-610.) Ehrman also discloses that, upon receiving segments, a requesting node computer can also specifically request replacement of portions of the requested video that it determines have not 10 been received or are arriving slowly, which are also specific requests for a part of the item of media content. (Ex. 1013, 8:10-23.) A POSITA would have combined this feature of Clarke with Ehrman to ensure reliable reception of large files. *Id.*

**Claim 10:** A POSITA would have understood that Clarke’s file-sharing

node, or Ehrman’s node that requests and plays back video files, is a computer 15 such as a desktop computer that uses Windows or Unix operating system, where data stored in the file system is accessible to multiple software functions, processes or threads, *e.g.*, an uploading function and a playback. (Ex. 1019, ¶¶ 611-615.) A POSITA would have understood such concurrent use of data by Clarke’s software and Ehrman’s content display application at least because prevalent operating 20 systems, *e.g.*, Windows and Unix, were well known to be multi-tasking operating

systems that provided simultaneous read access to data stored in a data file. *Id.* A POSITA would have understood that both the file uploading application and the playback operation do not attempt to make changes to the content of the media file and therefore do not perform conflicting operations on the same memory location  
5 and thus are readily implemented simultaneously with each other. *Id.*

**Claim 14:** Clarke discloses that files are uploaded from a user computer using IP / TCP protocol UDP and a port ID. (Ex. 1012, ¶¶ 5, 28.) A POSITA would have understood that a buffering operation is performed in a computer in which data is copied from a file into a buffer used by the transmission circuitry,  
10 *e.g.*, a TCP protocol stack on the computer. (Ex. 1019, ¶¶ 616-621.) This claim element would therefore have been obvious in view of Clarke. *Id.* A POSITA would have understood that Ehrman's node computers, which transmit content chunks via the Internet, also include similar transmission buffers corresponding to OSI layer 4 (TCP or UDP) port numbers. *Id.* Ehrman also discloses the use of a  
15 stream storage unit and a network packager that packages streams for network transmission, each of which would have been understood by a POSITA to include a buffer for transmitting media streams. (Ex. 1013, 7:1-17; Ex. 1019, ¶¶ 619-621.)

**Claim 15:** *See* Claim element 3. (Ex. 1019, ¶¶ 622-624.)

**Claim 16: Element 16.Preamble:** *See* Claim elements 1.Preamble and  
20 4.Preamble. Ex. 1019, ¶¶ 625-627. **Elements 16.A, 16.B, and 16.C:** *See,*

respectively, Claim element 1.A; 1.B; and 4.C. Ex. 1019, ¶¶628-632. **Element 16.D** : *See* Claim element 7.D. Clarke further teaches storage of received files. (Ex. 1009, Abstract, § 3.2.) The use of a buffer for storage and processing of data in a computer was generally known. (Ex. 1019, ¶¶633-634.) **Elements 16.E, 16.F, 16.G and 16.H**: *See*, respectively, Claim elements 16.D and 7.E; Element 1.D; Element 1.E; and Elements 16.D, 7.E, 7.L and 7.M. (Ex. 1019, ¶¶635-642.) A POSITA knew that Unix and Windows file systems used buffers to store data during use by software applications to overcome slow access time and wear and tear associated with frequent hard drive accesses. *Id.* A POSITA would have therefore found it obvious that two software operations being performed on the same operating system; viz., media playback and data transmission to requesting computer; would use data from the same buffer rather than accessing data separately from the hard drive. *Id.*

**E. GROUND 5: CLAIMS 2, 11-13 ARE OBVIOUS OVER CLARKE, EHRMAN AND SCHUSTER**

Schuster discloses a PID for receiving content, *e.g.*, audio and video, and playing the content back to a user. (Ex. 1050, Abstract.) In various embodiments, the PID can be a PDA, a phone and a network telephone. (*Id.*, Title, Abstract, 7:43-61, 9:20-36, 9:56 - 10:7, 14:9-15, 18:4-22, 20:49-63.) A POSITA would have realized that, similar to Ehrman, Schuster teaches using Microsoft Media Player for media playback (*Id.*) and would have been guided or motivated by the express

teaching of the use of the Media Player application in Ehrman and Schuster and would have a reasonable expectation of success based on the teachings of Clarke, Ehrman and Schuster and general information at the time. (Ex. 1019, ¶¶643-646.)

**Claim 2:** Schuster discloses a PID that includes a media player, and is

5 wirelessly connected to the Internet using protocols such as TCP/IP. (Ex. 1050, FIG. 6,7A,7B,8, 8:8-20,20:1-12,22:1-14,20:1-54,25:1-3). Clarke discloses a file sharing network in which multiple devices are connected to each other via a P2P connection using TCP/IP protocol. (Ex. 1012, ¶¶ 28, 5). The combined teaching of Schuster and Clarke thus disclose the claimed feature. (Ex. 1019, ¶¶643-646). A  
10 POSITA would have combined the portable information device of Schuster with the file sharing network of Clarke and content display application of Ehrman to allow video playback on user devices. *Id.*

**Claim 11:** Schuster discloses a personal digital assistant that is wirelessly connected to the Internet and is equipped with a software application for media  
15 playback. (Ex. 1050, 8:21-34; 9:19-26,11:47-55.) A user of this device is mobile. (Ex. 1050, 11:47-55). The combination of Schuster's mobile device with Clarke's file sharing and Ehrman's media playback application thus renders the claim obvious. (Ex. 1019, ¶¶647-650.) A POSITA would have combined Schuster's mobile device with the file sharing disclosed by Clarke and the player application  
20 disclosed by Ehrman to provide media sharing capability to a variety of users. *Id.*

**Claim 12:** Schuster discloses a PDA and a wireless phone that is wirelessly connected to the Internet and is equipped with a software application for media playback. (Ex. 1050, 8:21-34; 9:19-26,11:47-55.) A POSITA would have understood a PDA to be a pocket computer. The combination of Schuster's PDA with Clarke's file sharing and Ehrman's media playback application thus renders the claim obvious. (Ex. 1019, ¶¶61-654.) A POSITA would have combined Schuster's PDA with the file sharing network of Clarke and the player application of Ehrman to provide media sharing capability to a variety of users. *Id.*

**Claim 13:** Schuster discloses a PID that is wireless and receives streaming media, *e.g.*, sent using Real Time Streaming Protocol. (Ex. 1050, 8:15-20, 16:16-34; 20:49-55; 23:28-35). The combined teachings of Schuster's PID, operating in the file sharing network disclosed by Clarke and the media playback application disclosed by Ehrman renders this claim obvious. (Ex. 1019, ¶¶655-658.) A POSITA would have combined Schuster's PID with the file sharing network disclosed by Clarke and the player application disclosed by Ehrman to provide media sharing to users. *Id.* This element is thus disclosed and a POSITA would have been guided by the teachings to combine Clarke, Ehrman and Schuster.

## VII. CONCLUSION

In light of all the evidence and analyses, Petitioner respectfully requests Claims 1-16 be canceled.



Dated: November 20, 2015

Respectfully submitted,

/Bing Ai/

Lead Counsel

Bing Ai, Reg. No. 43,312

Back-up Counsel

Matthew C. Bernstein, *Pro Hac Vice*

Kevin J. Patariu, Reg. No. 63,210

Christopher L. Kelley, Reg. No. 42,714

Vinay P. Sathe, Reg. No. 55,595

Philip A. Morin, Reg. No. 45,926

PERKINS COIE LLP  
11988 El Camino Real, Suite 350  
San Diego, CA 92130  
(858) 720-5700

Attorneys for Petitioner  
Microsoft Corporation

## **CERTIFICATE OF SERVICE**

The undersigned hereby certifies that a true copy of the foregoing  
PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 7,590,752 B2  
and supporting materials have been served in its entirety this Twentieth Day of  
November, 2015, by FedEx® on Patent Owner at the correspondence address for  
the attorney of record for the 752 Patent shown in USPTO PAIR:

FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP  
901 NEW YORK AVENUE, NW  
WASHINGTON DC 20001-4413

an additional courtesy copy is also being sent to the maintenance fee address as  
listed in USPTO PAIR:

NEDERLANDSCH OCTROOIBUREAU  
P.O. BOX 29720  
2502 LS THE HAUGE  
THE NETHERLANDS

and sent electronically to the attorneys of record for Plaintiff in the concurrent  
litigation matter:

Matthew J Antonelli  
Zachariah Harrington  
Califf Teal Cooper  
Kris Yue Teng  
Larry Dean Thompson , Jr  
Antonelli, Harrington & Thompson LLP  
4306 Yoakum Blvd., Ste. 450  
Houston, TX 77006

/Bing Ai/

Lead Counsel

Bing Ai, Reg. No. 43,312

Back-up Counsel

Matthew C. Bernstein, *Pro Hac Vice*

Kevin J. Patariu, Reg. No. 63,210

Christopher L. Kelley, Reg. No. 42,714

Vinay P. Sathe, Reg. No. 55,595

Philip A. Morin, Reg. No. 45,926

Attorneys for Petitioner

Microsoft Corporation