

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Activision Blizzard, Inc., Electronic Arts Inc., Take-Two Interactive Software,
Inc., 2K Sports, Inc., and Rockstar Games, Inc.,
Petitioners

v.

Acceleration Bay, LLC,
Patent Owner

Case Nos. IPR2016-00747

**DECLARATION OF DAVID R. KARGER IN SUPPORT OF THE
PETITION FOR AN *INTER PARTES* REVIEW OF
UNITED STATES PATENT NOS. 6,732,147 AND 6,910,069**

MAIL STOP PATENT BOARD
Patent Trial and Appeal Board
United States Patent and Trademark Office
Post Office Box 1450
Alexandria, Virginia 22313-1450
Submitted Electronically via the Patent Review Processing System

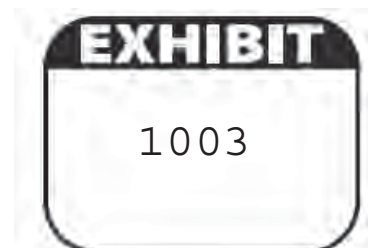


TABLE OF CONTENTS

I.	BACKGROUND AND QUALIFICATIONS	7
II.	LEVEL OF ORDINARY SKILL	11
III.	MATERIALS RELIED UPON	14
IV.	SUMMARY OF THE '147 AND '069 PATENTS.....	14
	A. Overview Of The '147 Patent.....	14
	B. Overview Of The '147 Patent Prosecution History	18
	C. Overview Of The '069 Patent.....	20
	D. Overview Of The '069 Patent Prosecution History	23
V.	OVERVIEW OF THE TECHNICAL FIELD OF THE CLAIMED INVENTIONS AND BRIEF DISCUSSION OF SOME OF THE RELEVANT PRIOR ART	26
VI.	CLAIM CONSTRUCTION	40
VII.	LEGAL STANDARDS	41
VIII.	INVALIDITY OF THE '147 PATENT UNDER 35 U.S.C. § 103	47
	A. The Challenged Claims Of The '147 Patent.....	47
	B. Overview Of Shoubridge	47
	C. Overview Of Denes.....	52
	D. Overview Of Rufino	53
	E. Combination Of The Teachings Of Shoubridge, Denes, And Rufino...	56
	F. Grounds 1-3: Invalidity Of Claims 1-16.....	58
	G. Ground 4: Invalidity Of Claims 1-16.....	92
IX.	INVALIDITY OF THE '069 PATENT UNDER 35 U.S.C. § 103	94
	A. The Challenged Claims Of The '069 Patent.....	94
	B. Overview Of Obraczka	95
	C. Overview Of Denes.....	101
	D. Overview Of Shoubridge	101
	E. Combination Of The Teachings Of Obraczka, Denes, And Shoubridge	103
	F. Grounds 1-2: Invalidity Of Claims 1-17.....	108
X.	SECONDARY CONSIDERATIONS	139

XI. CONCLUSION.....	139
---------------------	-----

LIST OF EXHIBITS

Exhibit	Description
Ex. 1001	U.S. Patent No. 6,732,147 (“the ’147 patent”)
Ex. 1002	U.S. Patent No. 6,732,147 File History
Ex. 1003	<i>This Exhibit</i> – Expert Declaration of David R. Karger (“Karger”)
Ex. 1004	Declaration of Scott Bennett, Ph.D
Ex. 1005	Peter J. Shoubridge & Arek Dadej, “Hybrid Routing in Dynamic Networks,” IEEE International Conference on Communications, Montreal, 1997 (“Shoubridge”)
Ex. 1006	Declaration of Steven Silvio Pietrobon attaching as Exhibit F Peter J. Shoubridge, “Adaptive Strategies for Routing in Dynamic Networks” (Ph.D. Thesis, University of South Australia, December 1996) (“Shoubridge Thesis”)
Ex. 1007	John M. McQuillan, <i>et al.</i> , “The New Routing Algorithm for the ARPANET,” IEEE Transactions Comms., Vol. 28, No. 5, 1980 (“McQuillan”)
Ex. 1008	Yogen Kantilal Dalal, “Broadcast Protocols in Packet Switched Computer Networks,” (Ph.D. Thesis, Stanford University 1977) (“Dalal”)
Ex. 1009	Katia Obraczka, <i>et al.</i> , “A Tool for Massively Replicating Internet Archives: Design, Implementation, and Experience,” Proceedings of the 16th International Conference on Distributed Computing Systems, 27-30 May 1996, Hong Kong (New York, NY: 1996), 657-664 (“Obraczka Paper”)
Ex. 1010	Katia Obraczka, “Massively Replicating Services In Wide-Area Internetworks,” (Ph.D. Thesis, University of Southern California December 1994) (“Obraczka”)
Ex. 1011	Jose Rufino, <i>et al.</i> , “A Study On The Inaccessibility Characteristics Of ISO 8802/4 Token-Bus LANs,” IEEE INFOCOM ’92: The Conference on Computer Communications. One World through Communications. Eleventh Annual Joint Conference of the IEEE Computer and Communication Societies, Florence, Italy, Vol. 2 (Picataway, NJ: IEEE Service Center, 1992), 0958-0967 (“Rufino”)
Ex. 1012	U.S. Patent No. 6,829,634 (“the ’634 patent”)
Ex. 1013	Kuo-Jui Raymond Lin, “Routing and Broadcasting in Two-dimensional Linear Congruential Graphs of Degree Four,” (Master’s Thesis, Concordia University, June 1994) (“Kuo-Jui Lin”)

Exhibit	Description
Ex. 1014	William S. Davis and David C. Yen, <u>The Information System Consultant's Handbook: Systems Analysis and Design</u> , CRC Press, 1998 ("Davis")
Ex. 1015	Topological Design Considerations in Computer Communication Networks, Computer Communication Networks (V.G. Cerf, D.D. Cown, R.C. Mullin), 1975 ("Cerf")
Ex. 1016	Stephen M. Grimes certification of English translation attaching English translation and original text of Tamás Dénes, "The 'Evolution' of Regular Graphs of Even Order by their Vertices," <i>Matematikai Lapok</i> , 27, 3-4 (1976/1979): 365-377 ("Denes")
Ex. 1017	English language translation from Exhibit 1016: Tamás Dénes, "The 'Evolution' of Regular Graphs of Even Order by their Vertices," <i>Matematikai Lapok</i> , 27, 3-4 (1976/1979): 365-377 ("Denes")
Ex. 1018	S. Toida, "Construction of Quartic Graphs," <i>Journal of Combinatorial Theory, Series B</i> , 16.2 (April 1974): 124-133 ("Toida")
Ex. 1019	T. Todd, "The Token Grid Network," <i>IEEE/ACM Transactions On Networking</i> , 2.3 (June 1994): 279-287 ("Todd")
Ex. 1020	Declaration of Peter John Shoubridge and, as Exhibit A, Peter J. Shoubridge, Adaptive Strategies for Routing in Dynamic Networks, Ph.D. Thesis (Univ. S. Austl., 1996) ("Shoubridge Thesis"), and as Exhibit B, Peter J. Shoubridge & Arek Dadej, Hybrid Routing in Dynamic Networks, in 3 IEEE INT'L CONF. ON COMMC'NS CONF. REC. 1381-86 (Montreal, 1997) ("Shoubridge").
Ex. 1021	U.S. Patent No. 5,802,285 to Hirviniemi ("Hirviniemi")
Ex. 1022	U.S. Patent No. 4,700,185 to Balph ("Balph")
Ex. 1023	U.S. Patent No. 6,603,742 to Steele <i>et al.</i> ("Steele")
Ex. 1024	U.S. Patent No. 6,732,147 File History Excerpt: November 5, 2003 Office Action
Ex. 1025	U.S. Patent No. 6,732,147 File History Excerpt: December 17, 2003 Amendment and Response
Ex. 1026	U.S. Patent No. 6,732,147 File History Excerpt: January 23, 2004 Notice of Allowance
Ex. 1027	L.G. Valiant, "Optimality of a Two-Phase Strategy for Routing in Interconnection Networks," <i>IEEE Transactions on Computers</i> , C-32.9 (September 1983): 861-863 ("Valiant")
Ex. 1028	U.S. Patent No. 6,490,247 to Gilbert <i>et al.</i> ("Gilbert")
Ex. 1029	U.S. Patent No. 6,122,277 to Garmire <i>et al.</i> ("Garmire")
Ex. 1030	U.S. Patent No. 5,181,017 to Frey <i>et al.</i> ("Frey")

Exhibit	Description
Ex. 1031	J. Van Leeuwen, <i>et al.</i> , “Interval Routing,” The Computer Journal, 30.4 (August 1987): 298-307 (“Van Leeuwen”).
<i>Ex. 1032-1100: Not Used</i>	
Ex. 1101	U.S. Patent No. 6,910,069 (“the ’069 patent”)
Ex. 1102	U.S. Patent No. 6,910,069 File History
Ex. 1103	U.S. Patent No. 6,910,069 File History Excerpt: May 17, 2004 Amendment

I, David R. Karger, hereby declare under penalty of perjury:

1. This declaration is submitted in support of petitions for *inter partes* review of U.S. Patent Nos. 6,732,147 (the '147 patent") and 6,910,069 ("the '069 patent"). For the reader's convenience, I address both the '147 and '069 patents in a single declaration because of the overlapping nature of the subject matter and prior art.

I. BACKGROUND AND QUALIFICATIONS

2. I have been retained on behalf of Petitioners and real parties in interest, Activision Blizzard, Inc., Electronic Arts Inc., Take-Two Interactive Software, Inc., 2K Sports, Inc., and Rockstar Games, Inc., to provide my opinions generally regarding the validity of claims 1-16 of U.S. Patent No. 6,732,147 (the "Challenged '147 Claims") and claims 1-17 of U.S. Patent No. 6,910,069 (the "Challenged '069 Claims").

3. Attached hereto as Appendix A is a true and correct copy of my Curriculum Vitae, and Faculty Personnel Record, which, collectively, describe my background and experience.

4. I earned a Bachelor's Degree in Computer Science and Physics from Harvard University in 1989, a certificate of advanced study in mathematics from Churchill College, Cambridge, England, in 1990, and a Ph.D. Degree in Computer Science from Stanford University in 1994.

5. I am currently a professor at the Massachusetts Institute of Technology where I am a member of the Computer Science and Artificial Intelligence Laboratory and the Electrical Engineering and Computer Science Department.

6. I have previously worked as a research scientist with Akamai Technologies (1998–2001), as a postdoctoral fellow at AT&T Bell Laboratories (1994–1995), as an intern at Xerox PARC (1991–1995), and have performed consulting work for Google, Microsoft, and Vanu.

7. My research interests include, *inter alia*, graph algorithms and their applications in communications, networking, and natural language processing. I have authored over two hundred publications in these and other areas.

8. My Ph.D. thesis on graph theory and graph algorithms received the 1994 ACM doctoral dissertation award and the Mathematical Programming Society's 1997 Tucker Prize. I also received the National Academy of Science's 2003 Award for Initiative in Research. See Appendix A for a representative list of other honors.

9. I am a fellow of the Association for Computing Machinery ("ACM"), and a member of the Society for Industrial and Applied Mathematics ("SIAM"). I am also a member of other professional associations. See Appendix A.

10. I have also served as a referee for the Journal of the ACM, the ACM Transactions on Information Systems, Mathematical Programming, the Journal of Algorithms, the SIAM Journal on Computing, Information Processing Letters, and the Journal of Computer and System Sciences, and chaired the 2009 International Semantic Web Conference in Chantilly, VA. In addition, I served on program committees for the 1996 ACM-SIAM Symposium on Discrete Algorithms, the 1996 and 1998 IEEE Symposium on Foundations of Computer Science, the 2002 International Peer to Peer Systems conference (IPTPS), the 2005 ACM Symposium on Theory of Computing, the 2007 and 2009 Conference on Innovative Database Systems Research (CIDR), the 2008 International Semantic Web Conference (ISWC), the 2010 Conference on Intelligent User Interfaces (IUI), the 2008 and 2010 Conference on Information Retrieval (SIGIR), the 2009 Visual Interface to the Social and Semantic Web, the Conference on Human-Computer Interaction (CHI) 2010, and the World Wide Web conference (WWW) in 2003, 2009, and 2010.

11. I have extensive experience in the technical areas of the '147 and '069 patents. For example, my work with colleagues at M.I.T. on distributed cache systems was the basis for the founding of Akamai Technologies, a well-known content distribution network, where I served as the first research scientist. I helped to develop network protocols for Akamai and was a named inventor on several

related patents. I also, together with a number of my colleagues at M.I.T., developed Chord, one of the four original distributed hash tables protocols, which address a fundamental problem in peer-to-peer networks (how to efficiently locate a node that stores particular data items). (See Stoica, Ion, *et al.* *Chord: A Scalable Peer-To-Peer Lookup Service For Internet Applications*, ACM SIGCOMM COMPUTER COMMUNICATION REVIEW, Vol. 31, No. 4 (2001), 149–160).

12. I have also performed services in several patent disputes as an independent technical expert and consultant, including non-litigation consulting on behalf of clients including Xerox PARC, NEC, IBM, Google, and Vanu. I served as an expert witness on computer and software-related cases before the United States Patent and Trademark Office and U.S. District Courts. I submitted expert declarations and was deposed in multiple IPR proceedings (IPR 2013-00292, 00293, 00294, 00295), which involved the computerized presentation of information from a variety of sources. I also submitted expert declarations in IPR 2014-00271 on behalf of Microsoft, and in *Inter Partes* Reexamination 95/000,685 on behalf of Apple. I also prepared an expert declaration addressing the proper construction of disputed claim terms on behalf of Microsoft in *SurfCast, Inc. v. Microsoft Corp.*, No. 2:12-cv-00333 (D. Me. filed Oct. 30, 2012). Finally, I was retained by Microsoft in several district court litigations, including: *Microsoft v. Alcatel-Lucent Enterprise*, No. 1:07-cv-00090, 2007 U.S. Dist. LEXIS 93048 (D.

Del. Dec. 18, 2008); *Microsoft v. Alcatel-Lucent*, No. 3:06-cv-02696 (S.D. Cal. Apr. 4, 2007); *Alcatel USA Sourcing, Inc. v. Microsoft Corp.*, No. 6:06-cv-00500, 2008 U.S. Dist. LEXIS 49615 (E.D. Tex. Dec. 17, 2008); *Alcatel USA Sourcing, Inc. v. Microsoft Corp.*, No. 6:06-cv-00499, 2008 U.S. Dist. LEXIS 64351 (E.D. Tex. Dec. 17, 2008); and *Lucent Technologies v. Microsoft Corp.*, 544 F. Supp. 2d 1080 (S.D. Cal 2008).

13. I have personal knowledge of the facts and opinions set forth in this declaration, and, if called upon to do so, I would testify competently thereto.

14. I am being compensated at a rate of \$770 per hour for my services. I have no financial interest in the outcome of this matter or on the pending litigations between Petitioners and Acceleration Bay LLC (“Acceleration Bay”) in federal court.

15. In developing my opinions below relating to the ’147 and ’069 patents, I have considered the materials cited herein, including those itemized in the “List of Exhibits” preceding this declaration.

II. LEVEL OF ORDINARY SKILL

16. I have been asked to provide my opinions regarding the knowledge and understanding of a person of ordinary skill in the art or “POSITA” as of at least July 31, 2000.

17. I understand that the factors considered in determining the ordinary level of skill in the art include: (i) the levels of education and experience of persons working in the field; (ii) the types of problems encountered in the field; and (iii) the sophistication of the technology. I understand that a POSITA is not a specific real individual, but rather a hypothetical individual having the qualities reflected by the factors above. This hypothetical person has knowledge of all prior art in the relevant field as if it were arranged on a workshop wall and takes from each reference what it would teach to a person having the skills of a POSITA.

18. In my opinion, the field of art relevant to the '147 and '069 patents is a combination of computer networking and graph theory, which are related disciplines.

19. In my opinion, a POSITA at the time of the alleged inventions claimed by the '147 and '069 patents would have a minimum of: (i) a bachelor's degree in computer science, computer engineering, applied mathematics, or a related field of study; and (ii) four or more years of industry experience relating to networking protocols and network topologies. Additional graduate education could substitute for professional experience, or significant experience in the field could substitute for formal education.

20. My opinions are based on my educational background, my commercial experience in the field of art, the technical training required to reduce

to practice the systems described in the '147 and '069 patents, the relevant prior art, my reading of the '147 and '069 patents and technical literature, and my experience consulting in many cases involving related technology. My opinion is also based on the numerous classes I have taught including undergraduate level computer science classes as identified on my *curriculum vitae* and my knowledge of the skills that such students possessed.

21. I understand that a POSITA is presumed to have knowledge of all relevant prior art. Therefore, a POSITA would have been familiar with each of the references cited herein and the full range of teachings they contain. Accordingly, with regards to the '147 patent, a POSITA reviewing Shoubridge would be familiar with the Denes and Rufino references, as discussed below, as well as the other references discussed in my declaration, and the full range of teachings they contain, at least because these prior art references address solutions to problems in networking. Additionally, with regards to the '069 patent, a POSITA reviewing Obraczka would be familiar with the Denes and Shoubridge references, as discussed below, as well as the other references discussed in my declaration, and the full range of teachings they contain, at least because these prior art references address solutions to problems in networking.

22. In July 2000, I would have qualified for or exceeded the level of skill required by the above definition, and I am in a position to opine on the

understanding of a POSITA as of that date. Unless otherwise noted, my statements and opinions below about the knowledge or understanding of a POSITA should be understood to refer to the knowledge or understanding of a POSITA as of July 31, 2000.

III. MATERIALS RELIED UPON

23. In reaching the conclusions described in this declaration, I have relied on the documents and materials cited in the “List of Exhibits” that precede my testimony in this declaration.

24. My opinions are also based upon my education, training, research, knowledge, and personal and professional experience.

IV. SUMMARY OF THE ’147 AND ’069 PATENTS

25. For the Board’s convenience, I address the ’147 and ’069 patents in this declaration because of the overlapping nature of the subject matter and prior art. The ’147 and ’069 patents both address maintenance of an m-regular non-complete network topology. The ’147 patent addresses removing a node from the topology. The ’069 patent addresses adding a node to the topology.

A. Overview Of The ’147 Patent

26. The ’147 patent, titled “Leaving a Broadcast Channel,” was filed on July 31, 2000, and issued on May 4, 2004. Ex. 1001.

27. The '147 patent describes a method for maintaining a non-complete m-regular topology when a node leaves a network. See, e.g., Ex. 1001, Abstract, cls. 1, 11.

28. The Challenged '147 Claims require that each participant is connected to the same number (m) of neighbors, so that the network is m-regular, where m is at least three. Ex. 1001, cls. 1, 11. Figure 1 of the '147 patent, for example, illustrates a graph where m is equal to 4 so that the graph is 4-regular. Ex. 1001 at Fig. 1, 2:46-47:

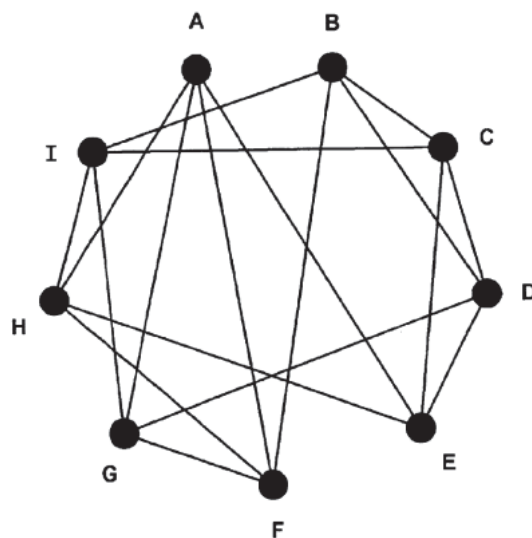


Figure 1 from the '147 Patent (Ex. 1001)

29. As explained in more detail below, the Challenged '147 Claims require that a computer broadcast a connection port search message on the broadcast channel to find another computer to which it can connect in order to maintain an m-regular graph. This requirement, viewed in the context of the '147 patent disclosure, makes it clear that the Challenged '147 Claims are directed to a

network that forms a “non-complete” graph—meaning at least two nodes in the network are not directly connected to each other. *See, e.g.*, Ex. 1001 at cls. 1, 6, 11. This is because a complete graph would continue to be regular when a computer disconnected, and would not need to perform any additional steps to maintain a regular graph.

30. Figure 5A illustrates one process of disconnecting a computer from the broadcast channel in order to maintain a non-complete m-regular graph. The '147 patent explains that “[w]hen computer H decides to disconnect, it sends its list of neighbors to each of its neighbors (computers A, E, F and I) and then disconnects from each of its neighbors. When computers A and I receive the message they establish a connection between them as indicated by the dashed line, and similarly for computers E and F.” Ex. 1001 at 9:20-26.

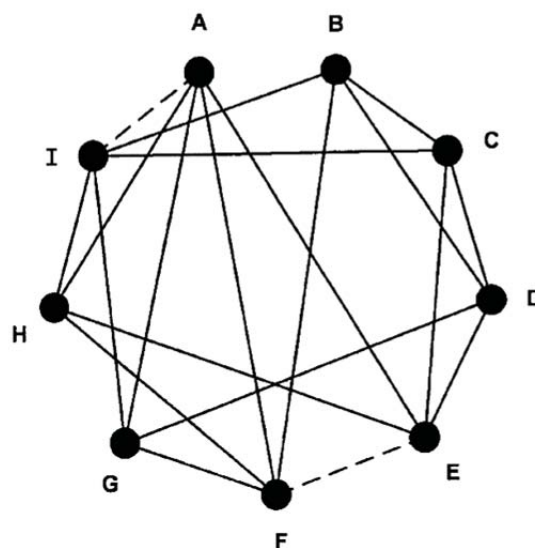


Figure 5A from the '147 Patent (Ex. 1001)

31. Figure 5B illustrates another process of disconnecting a computer from the broadcast channel in order to maintain a non-complete m-regular graph. The '147 patent explains that “[i]n this illustration, computer H has disconnected in an unplanned manner. When each of its neighbors, computers A, E, F, and I, recognizes the disconnection, each neighbor broadcasts a port connection request indicating that it needs to fill an empty port. As shown by the dashed lines, computers F and I and computers A and E respond to each other’s requests and establish a connection.” Ex. 1001 at 9:42-51.

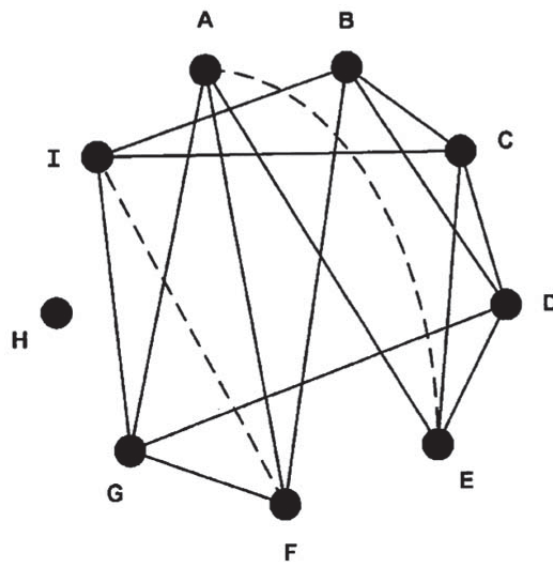


Figure 5B from the '147 Patent (Ex. 1001)

32. Notably, the '147 patent does not provide any generalized algorithm for maintaining a non-complete m-regular graph when a node disconnects. See generally Ex. 1001. The '147 patent only provides the above two examples. This

may be because algorithms for maintaining certain non-complete m-regular graphs when a node disconnects were well-known in the art by July 2000, as will be discussed below in Section V.

33. The independent claims are drawn to a node disconnection, and their dependents add commonplace features that would also have been well-known to a POSITA. Indeed, the combined teachings of Shoubridge in view of Denes and further in view of Rufino, renders obvious claims 1-16 of the '147 patent.

B. Overview Of The '147 Patent Prosecution History

34. The application that led to the '147 patent was filed July 31, 2000, and applicants did not claim priority to any earlier-filed application. Ex. 1001. I have reviewed the file history and believe this summary represents an accurate summary of the prosecution history for the patent.

35. The examiner issued a restriction requirement between: (Group I) original claims 1-8 drawn to determining the diameter of a broadcast channel; and (Group II) claims 9-30 disconnecting a computer from a broadcast channel. Ex. 1024 at 3. The applicant selected Group II. Ex. 1025 at 8.

36. The application was subject to a single office action where the examiner rejected original claims 9-30 as obvious in view of: (a) U.S. Patent No. 6,618,752 to Moore ("Moore") in view of U.S. Patent No. 6,353,599 to Bi ("Bi");

and (b) Moore in view of Bi in further view of “Graph Theory with Applications” by Bondy *et al.* (“Bondy”). Ex. 1024 at 5-11.

37. In response, applicants amended the preamble of each independent claim to indicate that the claim is directed to an m -regular graph where m is at least 3. Ex. 1025 at 4-6. According to applicants, Moore proposed a daisy chain arrangement where clients act as “mini-servers” to forward data to other clients. Ex. 1025 at 10. Applicants therefore argued that Moore did not disclose an m -regular graph where m is at least 3. Ex. 1025 at 11.

38. Applicants also added a limitation to independent claims 1¹ and 11² that the disconnecting computer sends a list of its neighbors to all of its neighbors; and that then the neighbors of the first computer can receive that list and can attempt to contact other computers on that list. Ex. 1025 at 4-6. Applicants added a similar limitation to claim 6³. *Id.* According to applicants, this technique is not disclosed in the Moore or Bondy references. Ex. 1025 at 12-13.

39. Subsequently, the examiner entered an Examiner’s Amendment which amended independent claims 9 and 23 to further reflect an explicit linking to the broadcast channel and its m -regular structure. Ex. 1026 at 4. The claims were

¹ Also referred to as claim 9 in Ex. 1025.

² Also referred to as claim 23 in Ex. 1025.

³ Also referred to as claim 15 in Ex. 1025.

then allowed. Ex. 1026 at 1. The examiner noted in his reasons for allowance that “[c]laims 9-12, 15-19, 22-25 and 28-30 are allowable over the prior art of record based on the argument set forth by the applicant in Paper 3 starting on page 10 as directed to the presently amended claims.” Ex. 1026 at 3.

C. Overview Of The '069 Patent

40. The '069 patent, titled “Joining a Broadcast Channel,” was filed on July 31, 2000, and issued on June 21, 2005. Ex. 1101.

41. The '069 patent describes a technique for maintaining an m-regular network topology when a participant is added to the network. *See, e.g.*, Ex. 1101, Abstract, cls. 1, 14.

42. Claims 1-17 of the '069 patent (the “Challenged '069 Claims”) require that each participant is connected to at least three or four other neighbors. Ex. 1101, cls. 1, 14. Figure 1 of the '069 patent, for example, “illustrates a graph that is 4-regular, and 4-connected.” Ex. 1101 at Fig. 1, 2:46-47. In the figure below, m is equal to 4.

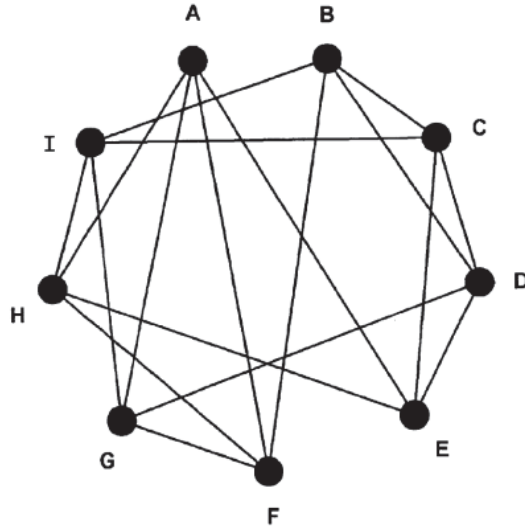


Figure 1 from the '069 Patent (Ex. 1101)

43. As explained in more detail below, the claims of the '069 patent require the disconnection of participants before connection to the incoming participant, which effectively requires that the network forms a “non-complete” graph—meaning at least two nodes in the network are not directly connected to each other. *See, e.g.*, Ex. 1101 at cls. 1 and 14.

44. Figures 3A-B of the '069 patent illustrate one process of connecting a new computer Z to the broadcast channel pursuant to the method disclosed in the '069 patent. Ex. 1101 at Figs. 3A-B, 2:51-52. In the method shown in these figures, new computer Z is added by breaking the connections between existing computer pairs (E, B) and (D, C) and then connecting computer Z to each of E, B, D and C. Ex. 1101 at 5:64 - 6:9.

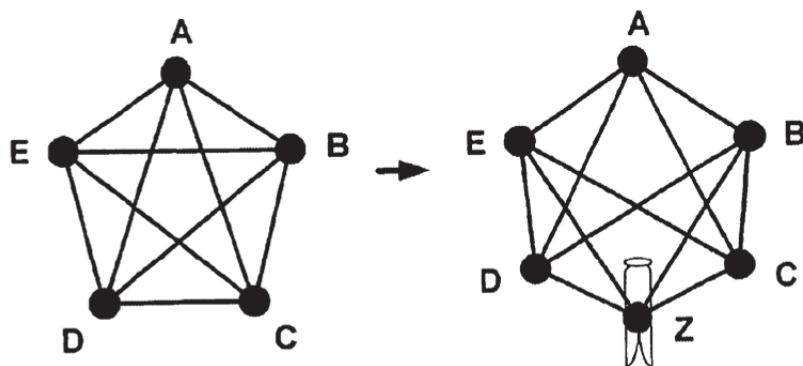


Figure 3A-B from the '069 Patent (Ex. 1101)

45. Notably, the '069 patent does not provide any generalized algorithm for maintaining a non-complete m -regular graph when adding a node. See generally Ex. 1101. The '069 patent only provides the above example. This may be because algorithms for maintaining certain non-complete m -regular graphs when adding a node were well-known in the art by July 2000, as will be discussed in Section V below.

46. Claim 1 is representative of the scope of the claims—its dependents add commonplace features that would also have been well-known to a POSITA. Independent claim 14 is similar in scope to claim 1 and its dependent claims similarly add commonplace features that would also have been well-known to a POSITA.

47. Indeed, the combined teachings of Obraczka in view of Denes and further in view of Shoubridge, renders obvious claims 1-3, 7-9, and 11-17. The

combination of Obraczka, Denes, Shoubridge, and Valiant renders obvious claims 4-6 and 10.

D. Overview Of The '069 Patent Prosecution History

48. The application that led to the '069 patent was filed July 31, 2000, and applicants did not claim priority to any earlier-filed application. Ex. 1101. I have reviewed the file history and believe this summary represents an accurate summary of the prosecution history for the patent.

49. In the first office action, the examiner issued a restriction requirement between: (Group I) claims 1-17, 32-40 drawn to method for adding a participant to a network; (Group II) claims 18-31 drawn to method for sending a connection edge search message for adding a participant to a network; and (Group III) claims 41-49 drawn to method for detecting neighbors with empty ports in a network. Ex. 1102 at 1185-89. The applicants selected Group I. *Id.* at 1195.

50. In the second office action, the Examiner rejected claims 1-17 and 32-40: (a) under 35 U.S.C. § 102(e) as being anticipated by U.S. Pat. No. 6,603,742 to Steele, *et al.* (“Steele”); (b) under 35 U.S.C. § 103(a) as obvious over Steele in view of Cho, *et al.*, “A Flood Routing Method for Data Networks,” ICICS '97 (“Cho”); under 35 U.S.C. § 103(a) as obvious over U.S. Pat. No. 6,490,247 to Gilbert *et al.* (“Gilbert”) in view of U.S. Pat. No. 6,533,020 to Hughes *et al.*

(“Hughes”); and under 35 U.S.C. § 103(a) as obvious over Gilbert in view of Hughes and further in view of Cho. Ex. 1102 at 1201-1217.

51. In response, applicants amended the preamble of the independent claims that are now numbered independent claims 1 and 14. Ex. 1103 at 4-6. Specifically, claim 1 was amended to require that the claimed method was “non-routing table based, non-switch based.” *Id.* Claim 14 was amended to require that the claimed method be “non-switch based.” *Id.* While independent claim 32 was also amended, this claim was later cancelled so it is not addressed in detail herein. Ex. 1102 at 1384.

52. According to applicants, their invention did not use a switch based method. Ex. 1103 at 12. In contrast, applicants argued that Gilbert disclosed a method of adding a node to a network using a switching mechanism. *Id.* at 16-17. The only example of “switch” that Gilbert appears to provide is one where all the nodes are connected at a common switch to make or break connections to other nodes, as illustrated in Figure 1 of Gilbert, provided below. *See also* Ex. 1028 at Figs. 5, 6, 8, and 10 (other figures that illustrate all nodes are connected at a common switch).

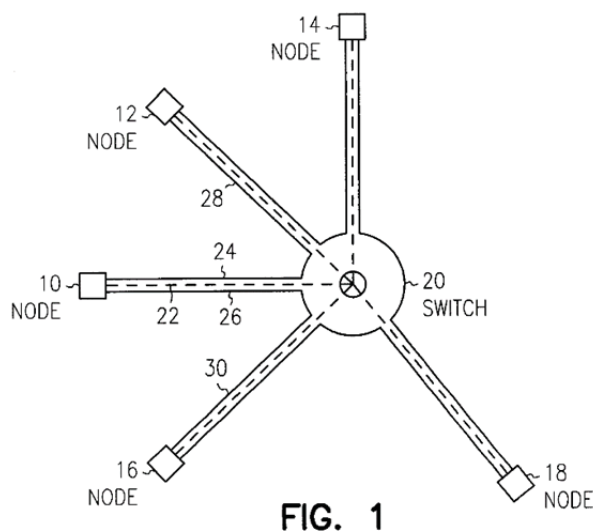


Figure 1 from Gilbert

53. Applicants also argued that their invention did not use routing tables. Ex. 1103 at 10-12. In contrast, applicants argued that Steele used an interim routing table to route traffic around the part of the network affected by the adding of a node. *Id.*

54. Applicants further added a limitation to claim 1 to require that “a seeking participant contacts a fully connected portal computer, which in turn sends an edge connection request to a number of randomly selected neighboring participants to which the seeking participant is to connect.” *Id.* at 10-12, 17-18. Claim 14 was similarly amended, albeit using “node” instead of “participant.” *Id.* According to applicants, Steele “fails to disclose a portal computer that directs the identification of viable neighboring participants to which the seeking participant is

to connect.” *Id.* at 11. A similar argument was made to distinguish the cited Gilbert reference. *Id.* at 17.

55. Applicants also emphasized that “the selection of the neighboring nodes is not random” in Gilbert since the selected nodes in Gilbert “are purposely adjacent to one another.” *Id.* Specifically, in Gilbert, the applicant said “[t]he portal node that is contacted provides information regarding a neighboring node that is adjacent to the seeking node; the selection of the neighboring node is not random.” *Id.* at 16. Applicants did not provide any further explanation of the meaning of “randomly.”

56. Subsequently, the examiner entered an Examiner’s Amendment which canceled claims 32-40. Ex. 1102 at 1384. The examiner noted in his reasons for allowance that claims 1-17 were allowed for “the reasons set forth by Applicant in Applicant’s response filed on May 10, 2004.” Ex. 1102 at 1386.

V. OVERVIEW OF THE TECHNICAL FIELD OF THE CLAIMED INVENTIONS AND BRIEF DISCUSSION OF SOME OF THE RELEVANT PRIOR ART

57. Graph theory is a branch of mathematics that involves the study of graphs, which are sets of vertices (often represented by points) connected by edges (represented by lines). Since long before the patent applications were filed, graph theory has been actively applied in a variety of industries and fields, including integrated circuit design, operations research (scheduling), and computer networks.

The hypothetical POSITA in July 2000 would have been familiar with the basic graph theory concepts discussed in this section.

58. Computer networks and their topologies are routinely represented using graph theory, and mathematical proofs or simulations are often developed to analyze and model the performance and reliability of a network. See, e.g., Ex. 1001 at 4:25-30 (“The broadcast technique overlays the underlying network system with a graph of point-to-point connections (*i.e.*, edges) between host computers (*i.e.*, nodes) through which the broadcast channel is implemented.”); Ex. 1101 at 4:25-30 (same); Ex. 1008 at 114 (“The various classes of networks are distinguished by certain topological properties of the graphs that represent them, like the degree of the nodes, or whether the graph is regular or not”); Ex. 1015 at 7 (“This paper presents a study of networks which are represented as linear graphs, and it is assumed the reader is familiar with elementary notions of graph theory.”); Ex. 1010 at 36 ¶ 2 (“We start this chapter by stating our topology computation problem as a graph theory problem.”).

59. Typically, each **node** on a graph (also known as a vertex) represents a participant (*e.g.*, a computer, router, or processor), and each **edge** (line) between two nodes represents a **connection**⁴ (also known as an arc or link) between them.

⁴ In this context, “connection” will generally refer to a *direct* connection between nodes. In graph theory, connectivity can also be used to refer to indirect

Ex. 1001 at 4:25-30; Ex. 1101 at 4:25-30; Ex. 1005 at 1382 ¶ 5. The term **neighbors** refers to nodes that are directly connected to each other. Ex. 1001 at 4:25-30; Ex. 1101 at 4:25-30; Ex. 1005 at 1382 ¶¶ 5, 8 and at 1383 ¶ 7.

60. For example, in **Fig. P1** below, there are seven nodes numbered 1-5, 7-8, which are connected by seven links. Node 2 is connected to its neighbors (nodes 1 and 3), but not to the other (nodes 4, 5, 7, and 8), with which it may only communicate indirectly (e.g., in this example, nodes 1 and 3 could potentially communicate via node 2, but nodes 1 and 3 themselves are not connected).

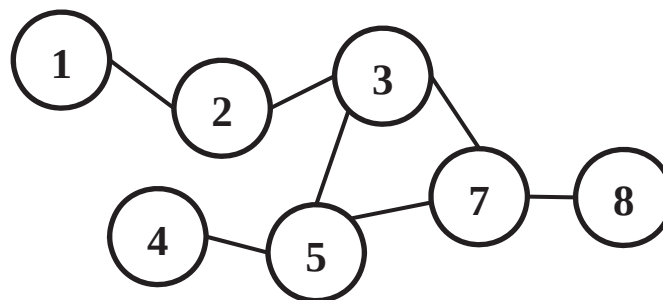


Fig. P1: Example model of a computer network as a graph

61. In graph theory, the **degree** of a node refers to the number of nodes it is connected to, in other words, the number of neighbors it has. Ex. 1001 at cl. 1; Ex. 1101 at cl. 1; Ex. 1008 at 118. A “regular” graph is one in which each node has the same degree. Ex. 1008 at 118; Ex. 1013 at 20 ¶ 7. An **m-regular** graph connectivity, e.g., via a path of directly connected intermediate nodes, as is meant when discussing “m-connectivity” as I do below. A POSITA would have inferred the intended meaning from context.

then, is one in which each node has degree m , in other words, a graph in which each node is connected to exactly m other nodes (*i.e.*, has exactly m neighbors). See Ex. 1001 at 4:40-43; Ex. 1101 at 4:40-43. For example, **Fig. P2** shows a 4-regular graph in which each of the six nodes is connected to exactly four other nodes:

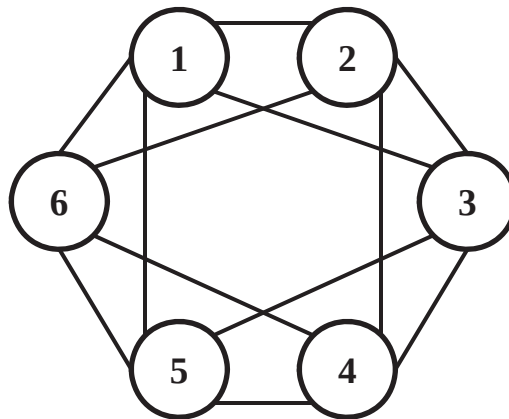


Fig. P2: Example of a 4-regular, non-complete graph

62. A **complete graph** is one that is fully connected, meaning that each node is connected to every other node, whereas in a **non-complete graph**, at least two nodes are not connected to each other. Ex. 1013 at 15, 20; see Ex. 1015 at 7. For example, the graph in **Fig. P2** above is non-complete, because node 1 is not connected to node 4. (Or alternatively, because node 2 is not connected to node 5, or because node 3 is not connected to node 6.)

63. Where N , the number of nodes in a graph, is at least two greater than m in an m -regular graph, the resultant graph will always be non-complete. See, *e.g.*, Ex. 1012 at cl. 1 (“wherein the number of participants is at least two greater

than m thus resulting in a non-complete graph.”). In a complete graph, each node has $N-1$ connections to other nodes, in other words, each node is connected to every other node. Ex. 1013 at 111. When N is at least two greater than m ($N \geq m + 2$), each node will always have at least one connection less than it needs for the graph to be complete. See Ex. 1012 at cl. 1. For example, the graph in **Fig. P2** above has six nodes ($N = 6$), and degree four ($m = 4$)—each node can only be connected to four other nodes, not five—making the graph incomplete.

64. Finally, **connectivity** can also refer to the minimum number of elements that must be removed from a graph to divide it into two or more separate parts, and is an important measure of the robustness of the graph. Ex. 1015 at 13–14, 15. Generally speaking, a graph is ***m-connected***, if it would require the removal of at least m nodes, or m links, to divide the graph into two separate parts. Ex. 1001 at 4:45-50; Ex. 1101 at 4:45-50; Ex. 1015 at 13-14, 15.

65. In graph theory, connectivity can refer to either ***node-connectivity*** or ***link-connectivity***. See Ex. 1015 (distinguishing node-connectivity from “line-connectivity,” which in the context of the reference is the same as link-connectivity). For example, the graph in **Fig. P3** below may be divided into separate sub-graphs that are not connected to each other (one comprising nodes 1, 2, 3, 4, and 5, and the other comprising nodes 7-9), by removing node 6. Alternatively, our example graph may be divided into two different sub-graphs

(one comprising nodes 1-8, and the other comprising just node 9), by removing the link between nodes 8 and 9:

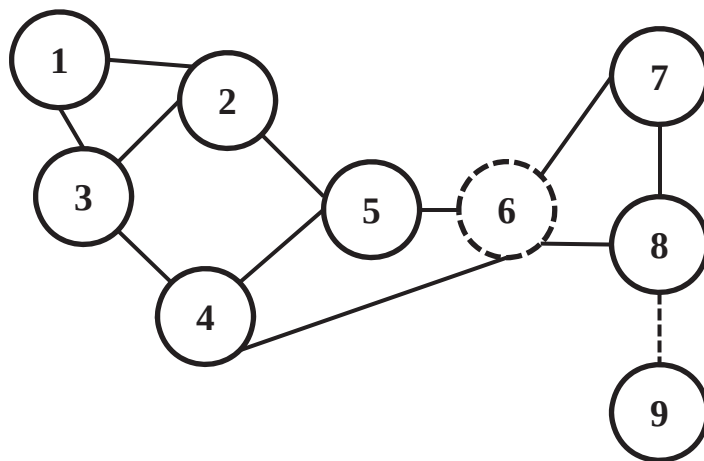


Fig. P3: Demonstrating the connectivity of a graph

66. However, “m-connected” as it relates to the claimed invention, according to the specification of the ’147 and ’069 patents, refers only to node-connectivity:

The graph used by the broadcast technique also has the property that it would take a failure of four computers to divide the graph into disjoint sub-graphs, that is two separate broadcast channels. This property is referred to as being 4-connected.

(Ex. 1001 at 4:45-50 (emphasis added); Ex. 1101 at 4:45-50 (same).)

67. Long before July 2000, a POSITA would have understood that the topology of a network could have a significant impact on the network’s characteristics, such as its performance, simplicity, scalability, and reliability. See Ex. 1015 at 6-7; Ex. 1014 at 6 § 52.4.1 to 12 § 52.4.5. Many types of network

topologies—including those based on non-complete, m-regular graphs—were well-known in the art, having been designed to achieve different balances of scalability, simplicity, reliability, and performance. *See, e.g.*, Ex. 1013 at 20.

68. In fact, the use of networks based on non-complete m-regular graphs was well known in the art. For example, and as discussed in more detail in Section IX below, Shoubridge (1997) discloses the use of broadcasting over m-regular, non-complete “torus” graphs, such as the graph in **Fig P4**, below. Similarly, a paper by Yogen Dalal (1977) discloses broadcasting over a non-complete, 4-regular network. *See* Ex. 1008 at 88-89, 157, 161; Ex. 1005 at 1382-1384. Broadcasting over a 4-regular “torus” network (shown below) is also disclosed in U.S. Patent No. 6,122,277 (Ex. 1029) (which incorporates the disclosure of U.S. Patent No. 5,181,017 (Ex. 1030) by reference, and which was filed August 19, 1997. *See* Ex. 1029 at 1:59-66, 5:29-43, 6:62; Ex. 1030 at Fig. 1 (reproduced as **Fig. P5**, below).

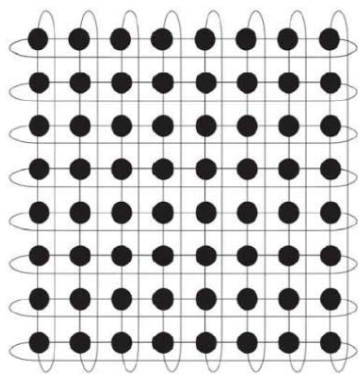


Figure 4.2: Network model - 64 nodes with connectivity of degree 4

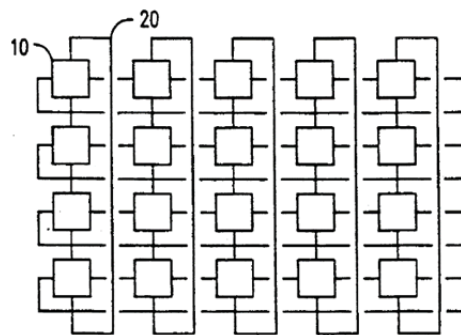


FIG. 1

**Fig P4: Ex. 1006 (Shoubridge Thesis),
Figure 4.2**

Fig P5: Ex. 1030, Figure 1

69. Long before July 2000, a POSITA would have understood the advantages of maintaining an m -regular non-complete topology when adding or subtracting a computer (*i.e.*, a participant or node) from the topology. *See, e.g.*, Ex. 1013 at 22 (“Regular graph: If some of the nodes in the network have less links than the others, it is likely for the network to have bottlenecks at these nodes. A regular graph has all of its edges uniformly distributed; therefore, reducing the probability of the occurrence of bottlenecks.”) Also, long before July 2000, a POSITA would have known the generalized formulas for maintaining an m -regular non-complete topology when adding or subtracting a node. For instance, Denes (Ex. 1017) and Toida (Ex. 1018), both published in the 1970s, disclose the same (or substantially similar) simple algorithms for maintaining an m -regular non-complete topology while adding or removing a node when m is an even number. Even the patents note that m must be an even number. Ex. 1001 at 14:52-56 (“In

the embodiment described above, each fully connected computer has four internal connections. The broadcast technique can be used with other numbers of internal connections. For example, each computer could have 6, 8, or any even number of internal connections.”); Ex. 1101 at 14:52-56 (same). Computer networks have added and dropped nodes in the way described by these formulas prior to July 2000. *See, e.g.*, Ex. 1023 at 12:26-32 (“When upgrading from 6 nodes to 7 nodes, the network administrator utilizes the interim routing table provided below, removes links 2-0 and 4-3, and then adds links 6-0, 6-2, 6-3, and 6-4.”)

70. To start, Denes discloses an operation called EC ($EC: E \rightarrow E'$; where $E' = E \setminus \{(p_i p_j)\} \cup \{(pp_i), (pp_j)\}$) to preserve a 2-regular structure of a network while adding a new node. Ex. 1017 at 365 ¶ 6. To do this, Denes discloses that the new graph (E') is the result of the disconnection of two neighbor nodes (*i.e.*, $\{(p_i p_j)\}$) from the original graph (E) which are then connected (noted by operation $\cup$) to a new node p ($\{(pp_i), (pp_j)\}$). *Id.* The result is a non-complete 2-regular graph (E'). *Id.*

71. Denes further discloses that the EC operation, which relates to a 2-regular graph, may be generalized to a k -regular graph. *Id.* The generalized operation is called ET ($ET: \Gamma_{n,k} \rightarrow \Gamma_n$; where $E' = E \setminus \{(p_1 p_2), (p_3 p_4), \dots, (p_{k-1} p_k)\} \cup \{(pp_1), (pp_2), \dots, (pp_k)\}$). *Id.* Denes discloses that the new graph (E') is the result of the disconnection of $k/2$ connected neighbor pairs ($\{(p_1 p_2), (p_3 p_4), \dots,$

$(p_{k-1}p_k)\})$ which are then connected (noted by operation $\cup$) to a new node p ($\{(pp_1), (pp_2), \dots, (pp_k)\}$). *Id.* The result is a non-complete k -regular graph. *Id.* Thus, *e.g.*, for a 4-regular graph, where $k = 4$, Denes teaches to disconnect $k/2$ neighbor pairs, or 2 neighbor pairs when $k = 4$. *Id.* Each of these 4 nodes that comprised the two neighbor pairs are then connected to the new node to maintain the 4-regular graph. *Id.*

72. The following paragraphs illustrate the above described operation of adding a node as disclosed by Denes as applied to Fig. 3A of the '069 patent. **Fig. P6** illustrates a fully connected 4-regular network where Node Z is to be added.

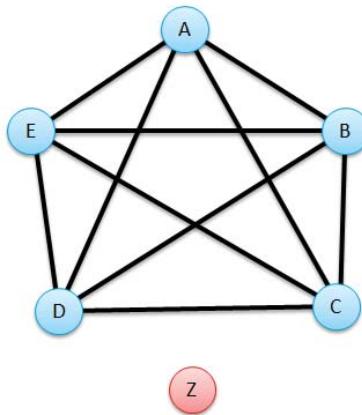


Fig. P6: fully connected 4-regular network where Node Z is to be added

73. Then, two connected neighbor pairs are selected. In a 4-regular network, $k = 4$, and the Denes ET operation divides k by two, and therefore, two neighbor pairs are selected. In **Fig. P7**, the two edges of the two neighbor pairs are selected, highlighted in red, pursuant to the formula disclosed in Denes. While I

have chosen pairs (B, E) and (D, C), Denes discloses that any two pairs may be chosen for this step.

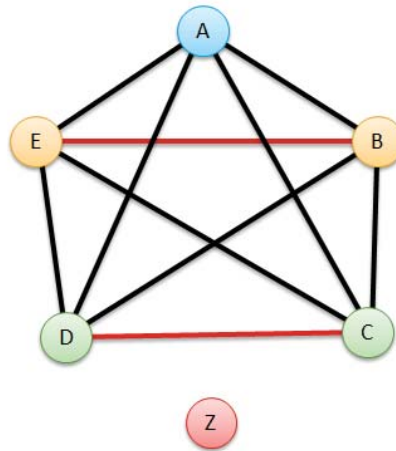


Fig. P7: $k/2$ neighbor pairs are selected; the two edges of the two neighbor pairs are highlighted in red

74. Then, Denes discloses disconnecting the edges (highlighted in red in ***Fig. P7***) connecting the neighbor pairs. The result of the operation is provided in ***Fig. P8***.

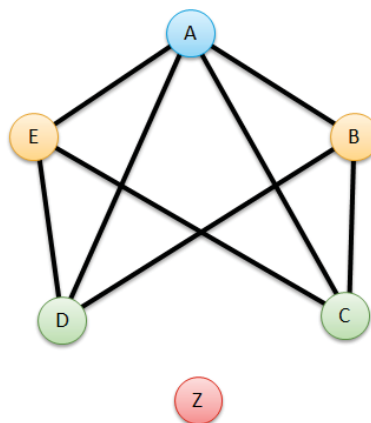


Fig. P8: two edges of the two neighbor pairs are removed

75. Finally, Denes discloses connecting Node Z to the disconnected neighbors (E, B, D, and C). The new edges are indicated in blue in **Fig. P9** below.

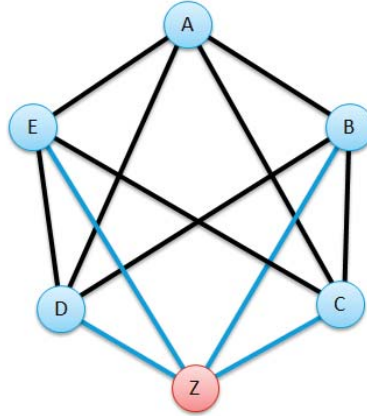


Fig. P9: Node Z is connected to Nodes E, B, D, and C.

76. Denes also discloses that a k -regular non-complete topology may be maintained when a node is removed from a network. For instance, Denes defines the aforementioned process in its description of the ET^{-1} operation, which is the inverse of the ET transformation. *Id.* at 2. The ET^{-1} operation is defined as: $ET^{-1}: \Gamma_{n,k} \rightarrow \Gamma_{n-1} = (P', E')$, $P' = P \setminus \{p\}$; $E' = E \setminus \{(pp_1), (pp_2), \dots, (pp_k) \cup \{(p_1p_2), (p_3p_4), \dots, (p_{k-1}p_k)\}$. *Id.* Denes discloses that the new graph (E') is the result of the disconnection of a node (p). *Id.* The edges connected to p ($\{(pp_1), (pp_2), \dots, (pp_k)\}$) are disconnected from its neighbor nodes. *Id.* Then the neighbor nodes are connected (noted by operation $\cup$) to one another ($\{(p_1p_2), (p_3p_4), \dots, (p_{k-1}p_k)\}$). *Id.* The result is a non-complete k -regular graph. *Id.*

77. The following paragraphs illustrate the above-described operation of removing a node as disclosed by Denes as applied to Fig. 1 of the '147 patent.

Fig. P10 illustrates a 4-regular network where Node H is to be removed.

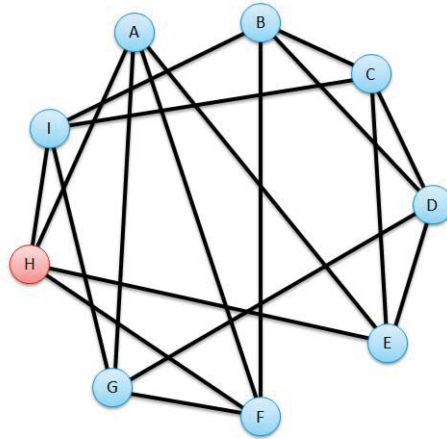


Fig. P10: illustrates a 4-regular network where Node H is to be removed

78. Then, Denes discloses that k neighbors (i.e., all neighbors) of node H are selected. In a 4-regular graph, four neighbors are selected, and the edges connected to the four neighbors are highlighted by the red lines below in **Fig. P11**.

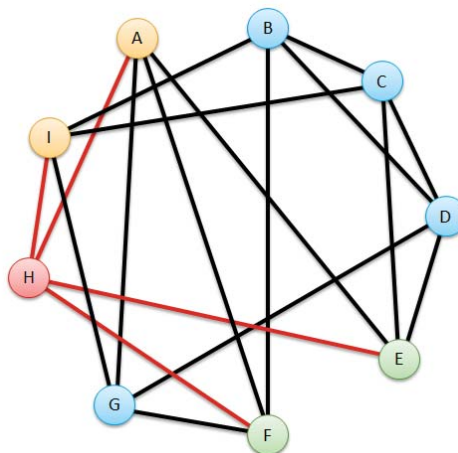


Fig. P11: the edges connected to the four neighbors are highlighted by the red lines

79. Then, Denes discloses disconnecting the highlighted red edges, which in **Fig. P11** are disconnected from node H. The result of this step of the operation is provided in **Fig. P12**.

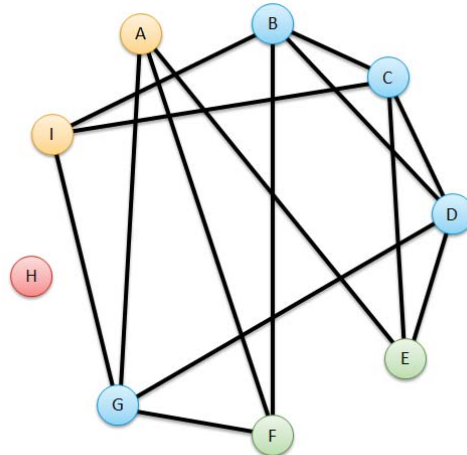


Fig. P12: the edges connected to the four neighbors are disconnected

80. Finally, Denes discloses that new connections are established between the neighbor nodes (A and I, E and F). The new edges are indicated in blue in **Fig. P13** below.

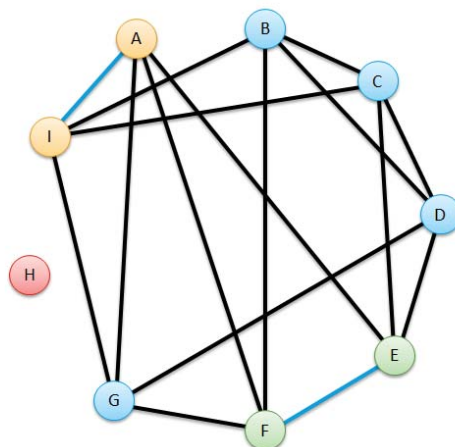


Figure P13: connecting neighbor nodes A and I, E and F together

81. Toida discloses the same algorithms as Denes. Toida discloses a method, called V-Type Reduction, for removing a node from an m-regular graph while maintaining regularity, where m is even and greater than or equal to 4. Ex. 1018 at 2-7. The V-Type Reduction method discussed in Toida is the same as Denes' ET^{-1} formula. Toida further discloses that the inverse operation of V-Type Reduction, called V-Type Expansion, can be used to add a node to an m-regular graph while maintaining m-regularity. *Id.* at 8. The V-Type Expansion formula is the same as Denes' ET formula. While Toida discloses the same methods, the analysis below will focus on Denes.

82. In my opinion, these examples and additional art demonstrate that the alleged inventions of the Challenged '147 and '069 Claims are unpatentable in view of the prior art (see Sections VIII and IX below).

VI. CLAIM CONSTRUCTION

83. I have been informed that for the purposes of this *inter partes* review, the standard for claim construction of terms within the claims of the patent is the "broadest reasonable construction" in light of the specification, which is different from the standard that applies in federal district court litigation.

84. For the purposes of this declaration, I have been asked to assume constructions for certain claim terms as presented below. I understand that these constructions are based on the citations from the specification identified below.

85. “**m-regular**” (’147 patent: cls. 1, 6, 11; ’069 patent: cl. 14) means “each node is connected to exactly m other nodes.” *See, e.g.*, Ex. 1001 at 4:40-50, 14:51-15:6; Ex. 1101 at 4:40-50, 14:51-15:6.

86. “**m-connected**” (’069 patent: cl. 14) means “dividing the network into two or more separate parts would require the removal of at least m nodes.” *See, e.g.*, 1101 at 4:40-50.

87. “**non-routing table based**” (’069 patent: cl. 1) means “does not use routing tables to carry out the steps of the claim.”

88. “**non-switch based method for adding [a participant/nodes] to a [network/graph]**” (’069 patent: cls. 1, 14) means “a common switch is not used for adding [a participant/nodes] to a [network/graph].”

89. For all terms not specifically listed above, I have been asked to assume that they have their plain and ordinary meaning and give them their broadest reasonable interpretation when considered in view of the text included in the specification (which includes the claims).

VII. LEGAL STANDARDS

90. I have been advised that if each and every element or step of a claim is disclosed within the “four corners” of a prior art reference, that claim is said to be “anticipated” by that single prior art reference and is invalid under 35 U.S.C. § 102 because the claimed invention is not, in fact, new or novel. I understand that

the standard for anticipation in an *inter partes* review proceeding is by a preponderance of the evidence.

91. I also have been advised that a prior art reference can disclose a claim feature if that feature is expressly described by that reference, or inherent from its disclosure. I understand that something is inherent from a prior art reference if the missing descriptive matter must necessarily be present, and it would be so recognized by a POSITA. I also understand that inherency cannot be established by probabilities or possibilities, and that the mere fact that something may result from a given set of circumstances is not sufficient to show inherency.

92. I also have been advised that a prior art document can disclose a claim feature, and anticipate a claimed invention, if that feature is described in another document that has been incorporated by reference. I understand that, to incorporate by reference, the host document must identify with detailed particularity what specific material it incorporates, and clearly indicate where that material is found in the incorporated document. I also understand that, in making the determination of the extent to which material is incorporated into a host document, the standard of a POSITA should be used to determine whether the host document describes the material to be incorporated by reference with sufficient particularity.

93. I understand that a claim may be invalid under § 103(a) if the subject matter described by the claim as a whole would have been obvious to a hypothetical POSITA in view of a prior art reference, or in view of a combination of references at the time the claimed invention was made. Therefore, I understand that obviousness is determined from the perspective of a hypothetical POSITA, and that the asserted claims of the patent should be read from the point of view of such a person at the time of claimed invention was made. I further understand that a hypothetical POSITA is assumed to know and to have all relevant prior art in the field of endeavor covered by the patent in suit and all analogous prior art. I understand that the standard for obviousness in an *inter partes* review proceeding is by a preponderance of the evidence.

94. I also understand that an analysis of whether a claimed invention would have been obvious should be considered in light of the scope and content of the prior art, the differences (if any) between the prior art and the claimed invention, and the level of ordinary skill in the pertinent art involved. I understand as well that a prior art reference should be viewed as a whole.

95. I understand that in considering whether a patent claim that covers a combination would have been obvious, I may assess: whether there are apparent reasons to combine known elements in the prior art in the manner claimed in view of interrelated teachings of multiple prior art references; the effects of demands

known to the design community or present in the marketplace; and/or the background knowledge possessed by a POSITA. I also understand that other principles may be relied on in evaluating whether a claimed invention would have been obvious, and that these principles include the following:

- A combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results;
- When a device or technology is available in one field of endeavor, design incentives and other market forces can prompt variations of it, either in the same field or in a different one, so that if a POSITA can implement a predictable variation, the variation is likely obvious;
- If a technique has been used to improve one device, and a POSITA would have recognized that it would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond his or her skill;
- An explicit or implicit teaching, suggestion, or motivation to combine two prior art references to form the claimed combination may demonstrate obviousness, but proof of obviousness does not depend on or require showing a teaching, suggestion, or motivation to combine, *see, e.g.*, Manual of Patent Examining Procedure § 2143(G);

- Market demand, rather than scientific literature, can drive design trends and may show obviousness;
- In determining whether the subject matter of a patent claim would have been obvious, neither the particular motivation or the avowed purpose of the named inventor controls;
- One of the ways in which a patent's subject can be proved obvious is by noting that there existed at the time of invention, a known problem for which there was an obvious solution encompassed by the patent's claims;
- Any need or problem known in the field of endeavor at the time of invention and addressed by the patent can provide a reason for combining the elements in the manner claimed;
- "Common sense" teaches that familiar items may have obvious uses beyond their primary purposes, and in many cases, a POSITA will be able to fit the teachings of multiple patents together like pieces of a puzzle;
- A POSITA is also a person of ordinary creativity, and is not an automaton;
- A patent claim can be proved obvious by showing that the claimed combination of elements was "obvious to try," particularly when there is a design need or market pressure to solve a problem and there are a finite number of identified, predictable solutions such that a POSITA would have

had good reason to pursue the known options within his or her technical grasp; and

- One should be cautious of using hindsight in evaluating whether a claimed invention would have been obvious.

96. I understand that, in making a determination as to whether or not the claimed invention would have been obvious to a person of ordinary skill, the Board may consider certain objective factors if they are present, such as: commercial success of product(s) practicing the claimed invention; long-felt but unsolved need; teaching away; unexpected results; copying; and praise by others in the field. These factors are generally referred to as “secondary considerations” or “objective indicia” of non-obviousness. I understand, however, that for such objective evidence to be relevant to the obviousness of a claim, there must be a causal relationship (called a “nexus”) between the claim and the evidence. I also understand that this nexus must be based on a novel element of the claim rather than something in the prior art. I also understand that there are instances when secondary considerations are unable to overcome primary evidence of obviousness (*e.g.*, motivation to combine with predictable results) that is sufficiently strong.

VIII. INVALIDITY OF THE '147 PATENT UNDER 35 U.S.C. § 103

97. In my opinion, the Challenged '147 Claims would have been obvious to a POSITA in view of the following prior art references or combinations of prior art references.

A. The Challenged Claims Of The '147 Patent

98. I have considered claims 1-16 of the '147 patent.

99. In the discussion below, I rely on Shoubridge as the primary reference, and Denes and Rufino as secondary references. On a few occasions, I refer to additional references. I have concluded that each of the Challenged '147 Claims is invalid based on the references described below.

100. My opinion is based on, among other things, the knowledge of a POSITA, as set forth in Section V above.

B. Overview Of Shoubridge

101. I understand that Shoubridge (Peter J. Shoubridge & Arek Dadej, “Hybrid Routing in Dynamic Networks,” IEEE International Conference on Communications, Montreal, 1997) is prior art. Ex. 1005; Ex. 1020 at 1 ¶¶ 3-5. This body of work is described in more detail in a 175-page Ph.D. thesis by the same individual, which I understand is also prior art. Peter John Shoubridge, “Adaptive Strategies For Routing In Dynamic Networks,” Ph.D. thesis, University of South Australia, 1996) (“Shoubridge Thesis”). Ex. 1006; Ex. 1020 at 2 ¶¶ 6-7.

For the avoidance of doubt, unless otherwise stated, my opinions are based on and refer to, the Shoubridge IEEE paper, rather than the Shoubridge Thesis.

102. Shoubridge discloses techniques for routing messages to participants in communications networks, including wireless computer networks, and draws extensively on work directed towards more general computer networking. *See* Ex. 1005 at 1381 ¶ 1; *id.* at References (citing numerous works on computer networking and routing in computer networks). Shoubridge models communications networks as graphs in which “[e]ach node functions as a source of user traffic entering the network.” Ex. 1005 at 1382 ¶ 5. Nodes with a connection between them are called neighbors. *Id.*

103. Shoubridge discloses a broadcasting protocol called “flooding” in which a node sends a message to all of its neighbors, and each neighbor, upon receiving a message for the first time, broadcasts the message to all of its neighbors except for the one from which it received the message. Ex. 1005 at 1382 ¶ 11 to 1383 ¶ 1.

104. Shoubridge also discusses the application of flooding in the context of routing a packet in a network from a source node to a destination node. Ex. 1005 at 1383 ¶¶ 3-5. In such a case, although a packet is ultimately only intended for a single destination node, it is nevertheless broadcast to all nodes. *See* Ex. 1005 at 1383 ¶ 1 (“This technique ensures that all nodes are visited at least once”); *id.*

at 1381 ¶ 4 (“Flooding algorithms simply broadcast user traffic through a network”). Because each packet in a network employing flooding is broadcast to all nodes in the network, flooding is appropriate not only for single-destination routing, but for broadcasting the same piece of information to all nodes in a network. See, e.g., Ex. 1005 at 1381 ¶ 3.

105. Shoubridge also demonstrates familiarity with the use of flooding for broadcasting information to all (rather than just one) nodes in the network as evidenced by its citation to Ex. 1007 (John M. McQuillan, *et al.*, “The New Routing Algorithm for the ARPANET,” IEEE Transactions Comms., Vol. 28, No. 5, 1980 (“McQuillan”). See Ex. 1005 at 1381 ¶ 3 (“This algorithm can be executed centrally or topology information can be broadcast to all nodes in a decentralized implementation [2]” (citing McQuillan)).

106. In McQuillan, flooding is the mechanism employed to broadcast updates throughout a network:

We considered different approaches for distributing the updates [8] and decided on “flooding,” in which each node sends each new update it receives on all its lines except the line on which the update was received. An important advantage of flooding is that the node sends the same message on all its lines, as opposed to creating separate messages on the different lines. These messages are short (no addressing information is required), so that the total overhead due to routing updates is much less than one percent. A final consideration

which favors flooding is that it is independent of the routing algorithm. This makes it a safe, reliable scheme.

Ex. 1007 at 5 ¶ 2. A POSITA reading Shoubridge thus would have understood that Shoubridge discloses that flooding is a mechanism to broadcast to all destinations in a network.

107. Shoubridge compares the characteristics of multiple routing protocols, including flooding, on a network topology that is a 64-node network in which each node has connections to 4 adjacent nodes—*i.e.*, an m -regular network, where $m = 4$. Ex. 1005 at 1383 ¶ 2 (“A 64 node network with connectivity of degree 4 is modeled as G . The network is a large regular graph forming a manhattan grid network that has been wrapped around itself as a torus to avoid edge effects.”) (emphasis added).

108. Shoubridge discloses a network topology that, in addition to being 4-regular, is also 4-connected (*i.e.*, m -connected for $m = 4$), meaning that it would take the failure of least 4 nodes to divide the network into two or more separate parts. See Ex. 1005 at 1383 ¶ 2; *see also*, Section V. In a “manhattan grid network” with “connectivity of degree 4” that “wrapped itself as a torus,” each node is connected to four other nodes in a grid format and the edge nodes are wrapped around to the opposite side (*i.e.*, left to right and top to bottom), forming a torus.

109. This “manhattan grid network” of Shoubridge is 4-connected. 4-connectivity can be proven using Menger’s Theorem, which states that a graph is 4-connected if one can construct 4 internally node-disjoint paths between any two vertices in the network (since in that case, removing one node only “breaks” one of the paths, which means that at least 4 nodes must be removed to break all 4 and disconnect the graph). The 4 vertex-disjoint paths in the grid can be defined straightforwardly as 4 paths that travel in 4 separate directions from the start node: one “up and then right,” one “right and then up,” one “left and then down,” one “down and then left,” producing 4 “L-shaped” paths that each take only a single turn. Minor variations are needed if the algorithm ends in the same row/column.

110. This network was graphically illustrated in Figure 4.2 of the Shoubridge Thesis, which is reproduced as **Fig. P4** below:

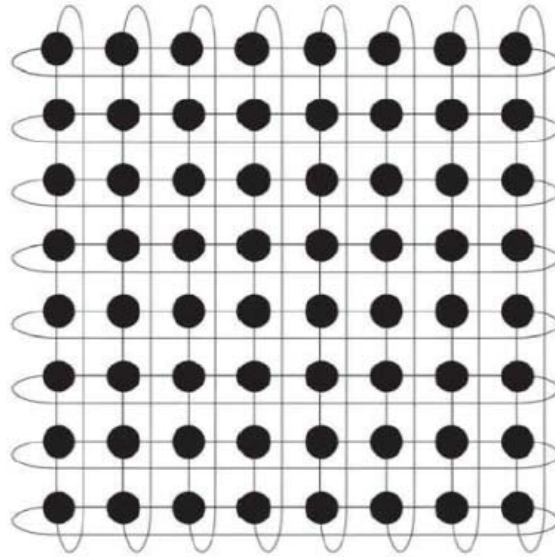


Figure 4.2: Network model - 64 nodes with connectivity of degree 4

Fig. P4: Ex. 1006, Shoubridge Thesis, Figure 4.2

111. Although a POSITA would have understood that a manhattan grid network could be represented with directional links, Ex. 1005 at 1382 ¶ 5 (“consider the network model as a directed graph G with N nodes and L bidirectional links”), Shoubridge expressly states that the links are bidirectional and that each bidirectional link corresponds to a single undirected graph edge. Ex. 1005 at 1383 ¶ 6 (“[t]he link (i,j) is then placed in a failed state”), 1383 ¶ 7 (“Note that a link (i, j) refers to a bidirectional link connecting two neighbouring nodes so there are two queueing systems associated with each bidirectional link.”).

C. Overview Of Denes

112. As discussed in Section V, Denes was published in the 1970s, and discloses methods of maintaining a non-complete m -regular topology when

removing a node. These methods are mathematical formulas that would have been well known to a POSITA by July 2000.

D. Overview Of Rufino

113. Rufino discloses a method for maintaining a ring network when a node is disconnected from the network. *See, e.g.*, Ex. 1011 at 0959 col. 2 ¶ 1 (“Access control is performed by a token passing protocol, which establishes a logical ring over the physical bus.”). A POSITA would understand that a ring network is a 2-regular non-complete graph topology. As a part of the method for maintaining a 2-regular non-complete graph, Rufino discloses that “[s]everal types of protocol data frames are exchanged between peer MAC entities, in order to provide logical ring housekeeping functions, essential for normal ring membership management, and for the preservation of ring integrity in the presence of faults.” *Id.* at 0960 col. 1 ¶¶ 3-4. These maintenance functions are supported by the following set of actions:

- Solicit new stations for logical ring entry.
- Find out the successor of a failed station (*who follows*).
- Solicit any station to respond at successor querying.
- Resolve contentions.
- Establish a new successor.
- Bid for token generation.
- Token passing.

Id.

114. A leaving station (first computer) in a token bus network leaves the logical ring in an orderly way by sending a *set_successor* frame (a disconnect message) to its predecessor station (second computer). Ex. 1011 at 0962 col. 2 ¶¶ 4-5 (“An orderly leave from a logical ring is only possible when that station holds the token. Station withdrawal is achieved through a ring patch between its predecessor and successor stations. For that purpose, the leaving station, before passing the token, issues a *set_successor* frame addressed to its predecessor and carrying the address of its future successor.”).

115. Rufino discloses that a *set_successor* frame (a disconnect message) sent by the leaving station to the predecessor station includes the address of the future successor. *Id.* One of ordinary skill in the art would understand that the future successor is the current successor station (neighbor) of the leaving station. Thus, Rufino discloses a disconnect message including a list of a single neighbor of the first computer. *Id.*

116. Rufino discloses a method for healing a disconnection in a token bus network when a station (second computer) in the token bus network is to leave the logical ring in an abrupt way from its predecessor station (first computer). Ex. 1011 at 0959 col. 1 ¶ 7 – col. 2 ¶ 1, 0962 col. 2 ¶ 9 – 0963 col. 1 ¶ 1 (“A station holding the token may transmit during an allowed period of time.

Afterwards, it should pass the token to its successor. If this station is failed, the token passing operation will not succeed. After the token pass-checking-period (one slot time) has elapsed a recovery strategy is tried.”).

117. Rufino discloses a predecessor station (first computer) attempting to send a token (a message) to the failed station (second computer). Ex. 1011 at 0962 col. 2 ¶ 9 – 0963 col. 1 ¶ 1 (“Under a single station failure assumption, the current token holder will receive, within this period, a response from the successor of the failed station. This establishes a new successor for the current token holder and ends the recovery procedures.”).

118. Rufino discloses that when a station cannot successfully send a message to its successor (because of an unplanned disconnect of the successor), the station broadcasts a connection port search message on the broadcast channel to find a third computer to which it can connect to maintain a 2-regular graph Ex. 1011 at 0960 col. 1 ¶¶ 3-4, 0962 col. 2 ¶ 9 – 0963 col. 1 ¶ 1. To do this, the station sends a *who follows* query (connection port search message) on the broadcast channel to find the successor (the neighbor) of the disconnected station. *Id.*

E. Combination Of The Teachings Of Shoubridge, Denes, And Rufino

119. A POSITA would have been motivated to combine the teachings of Shoubridge with the teachings of Denes and Rufino for several separate and independent reasons:

(a) Shoubridge Teaches The Use Of Its Network In A Dynamic Setting, And Denes And Rufino Address The Problem Of Dynamic Networks

120. As discussed above, a POSITA would have been motivated to combine the teachings of Shoubridge and Denes and Rufino because Shoubridge teaches that its system is used in a reliable dynamic network, and Denes and Rufino address how to maintain an m-regular non-complete topology when removing a node.

121. A POSITA would have been motivated to apply the well-known teachings of Denes and Rufino to the computer network taught by Shoubridge to improve its reliability.

122. Shoubridge discloses an exemplary 64-node, 4-regular network topology as an example of such a network environment. Ex. 1005 at 1384 (“It is reasonable to conclude that a large network similar to the one modeled, would require a flooding procedure if the network is to operate in a very dynamic, or potentially very dynamic environment.”); *id.* at 1383 (describing network model).

123. Denes discloses a straightforward mathematical principle to remove a node from a network to maintain a non-complete m-regular topology. Denes also discloses that it is straightforward to apply its formula for maintaining a 2-regular non-complete network to an m-regular non-complete network when m is even and exceeds 2. Denes also explicitly includes an example of a 4-regular non-complete network.

124. In view of Shoubridge and Denes, it is further straightforward to apply the messaging techniques disclosed in Rufino to coordinate the implementation of the method for removing a node disclosed in Denes to Shoubridge. For instance, it was well known to expand the messaging techniques of Rufino to grid topologies (where $m = 4$). *See, e.g.*, Ex. 1019 at 1 col. 2 ¶ 2 (“In this paper, a network referred to as the token grid is introduced. The token grid is a multidimensional extension of the token ring 121, where stations share access to a mesh formed by a set of overlapping rings. . . . Couplings between the rings are implemented by the user stations in a very simple fashion and under token control.”) (emphasis added).

125. For the reasons stated above, a POSITA, thus, would have been motivated to combine the teachings of Shoubridge, Denes, and Rufino to maintain a non-complete m-regular topology when removing a node.

**(b) Shoubridge Discloses A Non-Complete Regular Topology,
And Denes And Rufino Address The Problem Of
Maintaining A Non-Complete Regular Topology**

126. Shoubridge discloses a non-complete regular topology in a dynamic network. See Sections VIII.A-D. Denes and Rufino address the problem of maintaining a non-complete regular topology. *Id.* A POSITA would, therefore, have been aware of, and motivated to combine the teachings of all three references, because they were in the same technical field, and addressed the same technical problems.

127. For the reasons stated above, a POSITA thus would have been motivated to combine the teachings of Shoubridge, Denes, and Rufino to maintain a non-complete m-regular topology when removing a node.

F. Grounds 1-3: Invalidity Of Claims 1-16

(a) Claim 1

128. Preamble: To the extent the preamble is deemed to be a limitation of the claim, Shoubridge and Denes disclose the elements of the preamble, namely “[a] method of disconnecting a first computer from a second computer, the first computer and the second computer being connected to a broadcast channel, said broadcast channel forming an m-regular graph where m is at least 3.”

129. Shoubridge discloses a dynamic computer network in which “the distribution of traffic loads and network topologies may vary from nearly static to very dynamic. This dynamic behavior may vary both in space and in time.”

Ex. 1005 at 1381 Abstract. In a dynamic network, computers will fail, become congested, or enter and leave the network. Ex. 1005 at 1381 ¶ 3. (“Links may fail or become congested, nodes or users could be mobile, resulting in changes to source-destination paths.”).

130. Shoubridge also discloses networks in which each computer is connected to four (*i.e.*, at least three) neighbors in the broadcast channel. For example, Shoubridge discloses a 64-node network in which each participant is connected to four neighbors. Ex. 1005 at 1383 ¶ 2 (“A 64 node network with connectivity of degree 4 is modeled as G . The network is a large regular graph forming a manhattan grid network that has been wrapped around itself as a torus to avoid edge effects.”) (emphasis added).

131. Denes also discloses a method of disconnecting a first node from a second node, the first node and the second node being connected to a specific topology, said topology forming an m -regular graph where m is at least 3. See Section V above.

132. Claim 1(a): Shoubridge in view of Denes and in further view of Rufino, renders obvious “when the first computer decides to disconnect from the second computer, the first computer sends a disconnect message to the second computer, said disconnect message including a list of neighbors of the first computer.”

133. Denes discloses a method of disconnecting a first node from the second node. See Section V above. To do this, Denes discloses connecting the neighbors of the leaving node to each other. *Id.*

134. Rufino discloses a disconnect method where when the first computer decides to disconnect from the second computer, the first computer sends a disconnect message to the second computer. For example, Rufino discloses a method for a leaving station (first computer) in a token bus network to leave the logical ring in an orderly way by sending a *set_successor* frame (a disconnect message) to its predecessor station (second computer). Ex. 1011 at 0962 col. 2 ¶¶ 4-5 (“Stations may leave the logical ring either in an abrupt or orderly way. . . . An orderly leave from a logical ring is only possible when that station holds the token. Station withdrawal is achieved through a ring patch between its predecessor and successor stations. For that purpose, the leaving station before passing the token issues a *set_successor* frame, addressed to its predecessor, and carrying the address of its future successor.”)

135. Rufino discloses that a *set_successor* frame (a disconnect message) sent by the leaving station to the predecessor station includes the address of the future successor. Ex. 1011 at 0962 col. 2 ¶¶ 4-5. One of ordinary skill in the art would understand that the future successor is the current successor station (neighbor) of the leaving station. Thus, Rufino discloses the use of a disconnect

message including a list of neighbors of the first computer (a single neighbor in this example).

136. Therefore, it would have been obvious to modify Shoubridge in view of Denes in further view of Rufino so that when a node in the topology of Shoubridge is disconnected following the method taught in Denes, the disconnect message would include a list of neighbors of the first computer, as taught by Rufino.

137. Accordingly, a POSITA would have been motivated to combine the teachings of Shoubridge, Denes, and Rufino, which together disclose every limitation of this claim element (see Section VIII.E above).

138. Claim 1(b): Shoubridge in view of Denes in further view of Rufino renders obvious “when the second computer receives the disconnect message from the first computer, the second computer broadcasts a connection port search message on the broadcast channel to find a third computer to which it can connect in order to maintain an m-regular graph, said third computer being one of the neighbors on said list of neighbors.”

139. Denes discloses a method of removing nodes in order to maintain an m-regular graph. See Section V above. To do this, Denes discloses connecting the neighbors of the leaving node to each other. A POSITA would understand that the neighbors typically communicate with each other in order to connect as disclosed

in Denes. Indeed, it was and is commonplace for nodes in networks to exchange information. This information is routinely exchanged to facilitate communication in a network.

140. Furthermore, Rufino explicitly discloses broadcasting messages containing such information on the broadcast channel to maintain a 2-regular graph. Ex. 1011 at 0960 col. 1 ¶¶ 3-4 (“Several types of protocol data frames are exchanged between peer MAC entities, in order to provide logical ring housekeeping functions, essential for normal ring membership management, and for the preservation of ring integrity in the presence of faults.”) Rufino further discloses that the management functions include a variety of actions (such as finding out the successor of a failed station (*who_follows*), soliciting any station to respond at successor querying, resolving contentions, and establishing new successors). Ex. 1011 at 0960 col. 1 ¶¶ 3-4.

141. Rufino further discloses that when a station cannot successfully send a message to its successor (because of an unplanned disconnect of the successor), the station broadcasts a connection port search message on the broadcast channel to find a third computer to which it can connect to maintain a 2-regular graph. To do this, the station sends a *who follows* query (connection port search message) on the broadcast channel to find the successor (the neighbor) of the disconnected station. Ex. 1011 at 0963 col. 1 ¶ 1. (“During this *who follows* query the current token

holder waits for a *set_successor* frame, until the end of a three-slot-time period. Under a single station failure assumption, the current token holder will receive, within this period, a response from the successor of the failed station. This establishes a new successor for the current token holder and ends the recovery procedures.”)

142. As noted above, it would have been obvious to modify Shoubridge to use the technique disclosed in Denes to drop a node while maintaining the 4-regular graph, and this method of Denes involves connecting the former neighbors of the dropped node to each other. Rufino discloses the use of messaging when dropping a node from a network. It would have been obvious to further modify Shoubridge, when implementing the Denes node-dropping technique, to send a message including a list of multiple neighbors of the leaving station, as taught by Rufino. In particular, it would have been obvious to modify Shoubridge in view of Denes to include the orderly leave in Rufino, with the step of broadcasting a *who follows* query (connection port search message) on the broadcast channel. Broadcasting the *who follows* query allows the second computer to find a third computer to connect, the third computer being one of the multiple neighbors of the leaving station, to maintain the m-regular graph.

143. A POSITA would have been motivated to combine the teachings of Shoubridge, Denes, and Rufino for the reasons discussed above, which together disclose every limitation of claim 1 (*see* Section VIII.E above).

144. In my opinion, for at least the reasons discussed above, claim 1 is therefore rendered obvious under § 103 by Shoubridge in view of Denes in further view of Rufino.

(a) Claim 2

145. Shoubridge in view of Denes in further view of Rufino renders obvious the method of claim 2, which adds to claim 1 the additional limitation that “wherein the second computer receives a port connection message indicating that the third computer is proposing that the third computer and the second computer connect.”

146. Rufino discloses that if a station (first computer) leaves abruptly, while its predecessor station (second computer) has the token, then the token passing operation will not succeed and a recovery procedure is tried. Ex. 1011 at 0962 col. 2 ¶ 9-0963 col. 1 ¶ 1 (“During this *who follows* query the current token holder waits for a *set_successor* frame, until the end of a three-slot-time period. Under a single station failure assumption, the current token holder will receive, within this period, a response from the successor of the failed station. This establishes a new successor for the current token holder and ends the recovery

procedures.”). During this period, a response (port connection message) from the successor of the failed station (third computer) will be received by the second computer, and a connection between the second and third computer will be established. *Id.*

147. A POSITA would have been motivated to combine the teachings of Shoubridge, Denes, and Rufino, which together disclose every limitation of this claim (see Section VIII.E above). In particular, it would have been obvious to modify Shoubridge to implement the Denes node dropping technique, and to use the recovery procedure disclosed in Rufino, thereby achieving all of the elements of claim 2.

148. In my opinion, for at least the reasons discussed above, claim 2 is therefore rendered obvious under § 103 by Shoubridge in view of Denes in further view of Rufino.

(b) Claim 3

149. Shoubridge in view of Denes in further view of Rufino renders obvious the method of claim 3, which adds to claim 1 the additional requirement that “wherein the first computer disconnects from the second computer after sending the disconnect message.”

150. Rufino discloses a method for a leaving station (first computer) to disconnect from its predecessor station (second computer) by sending a

set_successor frame (a disconnect message). Ex. 1011 at 0962 col. 2 ¶¶ 4-5 (“An orderly leave from a logical ring is only possible when the station holds the token. Station withdrawal is achieved through a ring patch between its predecessor and successor stations. For that purpose, the leaving station before passing the token issues a *set_successor* frame, addressed to its predecessor, and carrying the address of its future successor.”).

151. A POSITA would have been motivated to combine the teachings of Shoubridge, Denes, and Rufino, which together disclose every limitation of this claim (see Section VIII.E above). In particular, it would have been obvious to modify Shoubridge to implement the Denes node dropping technique, and to use the recovery procedure disclosed in Rufino, thereby achieving all of the elements of claim 3.

152. In my opinion, for at least the reasons discussed above, claim 3 is therefore rendered obvious under § 103 by Shoubridge in view of Denes in further view of Rufino.

(c) Claims 4 And 5

153. Shoubridge in view of Denes in further view of Rufino renders obvious the additional features of claims 4-5, which include “wherein the broadcast channel is implemented using the Internet” (cl. 4) and “wherein the first computer and second computer are connected via a TCP/IP connection” (claim 5).

154. The knowledge of a POSITA when implementing the methods disclosed in Shoubridge, Denes, and Rufino would have included the use of the Internet and TCP/IP. The use of the Internet to broadcast messages in a network was ubiquitous by July 2000, and TCP/IP was the most common protocol used for Internet communications. In particular, Shoubridge discloses that its teachings are directed towards “nodes in a communications network.” Ex. 1005 at 1381 ¶ 1. A POSITA would have understood the recitation of a “communications network” to include networks such as the Internet and networks based on TCP/IP, which necessarily comprise TCP/IP connections that use TCP/IP protocol. In fact, the most common example of a “communications network” to a POSITA would have been the Internet itself, the dominant protocol of which was (and is) TCP/IP.

155. Accordingly, it would have been obvious to implement the connections taught in Shoubridge as Internet or TCP/IP connections based on the knowledge of a POSITA. A POSITA would have found it advantageous, obvious, and routine, in light of Shoubridge’s disclosure of “communications network[s],” to implement the disclosed networks using Internet and TCP/IP connections. It would have been clear to a POSITA that “Internet” and “TCP/IP connections” could be implemented on the network disclosed in Shoubridge, using techniques that were well-known in the art and that would have provided the expected, predictable results. A POSITA would have been motivated to do so for any

number of reasons that were also well known in the art, including, for example, ensuring the compatibility and usefulness of the disclosed communications network, given the popularity and mass reach of the Internet at the time.

156. A POSITA would have also been motivated to combine the teachings of Shoubridge, Denes, and Rufino, which together disclose every limitation of claims 4 and 5, as discussed in Section VIII.E above.

157. In my opinion, for at least the reasons discussed above, claims 4 and 5 are therefore rendered obvious under § 103 by Shoubridge in view of Denes in further view of Rufino.

158. If the Board believes that a specific reference is necessary for the additional elements of claims 4 and 5, then U.S. Patent No. 5,802,285 (“Hirviniemi”) can be combined with the above references to demonstrate the obviousness of these additional elements. Hirviniemi teaches that, by October 1997, it would have been advantageous, obvious, and routine, in light of Shoubridge’s disclosure of “communications network[s],” and Rufino’s network messaging techniques, to implement the disclosed networks using Internet and TCP/IP connections to connect a network. *See, e.g.*, Ex. 1021 at 5:2-3 (“A method of interfacing a transmission control protocol/internet protocol (TCP/IP) software designed for a local area network (LAN) to a wide area network (WAN)”). It would have been clear to a POSITA in view of Hirviniemi that “Internet” and

“TCP/IP connections” could be implemented on the network disclosed in Shoubridge and Rufino, using techniques that were well-known in the art and that would have provided the expected, predictable results. A POSITA would have been motivated to do so for any number of reasons that were also well known in the art, including, for example, ensuring the compatibility and usefulness of the disclosed communications network, given the popularity and mass reach of the Internet at the time.

159. A POSITA would have also been motivated to combine the teachings of Shoubridge, Denes, Rufino, and Hirviniemi, which together disclose every limitation of claims 4 and 5. *See also* Section VIII.E above.

160. In my opinion, for at least the reasons discussed above, claims 4 and 5 are also therefore rendered obvious under § 103 by Shoubridge in view of Denes and Rufino in further view of Hirviniemi.

(d) Claim 6

161. Preamble: To the extent the preamble is deemed to be a limitation of the claim, Shoubridge in combination with Denes discloses “[a] method for healing a disconnection of a first computer from a second computer, the computers being connected to a broadcast channel, said broadcast channel being an m -regular graph where m is at least 3.”

162. In particular, Shoubridge discloses a dynamic computer network including two computers connected to a broadcast channel, the broadcast channel forming an m -regular graph where m is at least 3, that operates when links fail and/or computers leave the network. See discussion of Shoubridge contained in claim 1 [Preamble].

163. In addition, Denes discloses a method of disconnecting a first node from a second node, the first node and the second node being connected to a specific topology, said topology forming an m -regular graph where m is at least 3. See Section V above.

164. Claim 6(a): Shoubridge in view of Denes in further view of Rufino renders obvious the first step of claim 6, *i.e.*, “attempting to send a message from the first computer to the second computer.”

165. Rufino discloses a dynamic computer network in which the removal of a computer from a network causes problems. Ex. 1011 at 0958 col. 2 ¶ 4 (“Uncontrolled omissions and partitions are a source of asynchrony and inconsistency. This is unacceptable for most systems, let alone real-time ones.”).

166. Rufino discloses a method for a station (second computer) in a token bus network to leave the logical ring in an abrupt way from its predecessor station (first computer). Ex. 1011 at 0959 col. 1 ¶ 7 - col. 2 ¶ 1 (“Access control is performed by a token passing protocol, which establishes a logical ring over the

physical bus. Access to the shared broadcast medium for data transmission is only granted to the station which currently holds the token.”); Ex. 1011 at 0962 col. 2 ¶¶ 4-5 (“Stations may leave the logical ring either in an abrupt or orderly way.”).

167. Rufino discloses a method for healing a disconnection of a first computer from a second computer. Rufino discloses that when a station cannot successfully send a message to its successor (because of an unplanned disconnect of the successor), the station broadcasts a connection port search message on the broadcast channel to find a third computer to which it can connect to maintain a 2-regular graph. To do this, the station sends a *who follows* query (connection port search message) on the broadcast channel to find the successor (the neighbor) of the disconnected station. Ex. 1011 at 0962 col. 2 ¶ 9 – 0963 col. 1 ¶ 1 (“During this *who follows* query the current token holder waits for a *set_successor* frame, until the end of a three-slot-time period. Under a single station failure assumption, the current token holder will receive, within this period, a response from the successor of the failed station. This establishes a new successor for the current token holder and ends the recovery procedures.”).

168. A POSITA would have been motivated to combine the teachings of Shoubridge, Denes, and Rufino, which together disclose the first step of claim 6 (see Section VIII.E above). In addition, as noted above, it would have been obvious to modify Shoubridge to employ the node removal technique disclosed in

Denes. Further, it would have been obvious to use the technique of Rufino to identify a failed port, namely “attempting to send a message from the first computer to the second computer” as claimed. Rufino discloses, as shown above, that an attempted transmission can be used to identify a failed node in the network.

169. Claim 6(b): Shoubridge in view of Denes in further view of Rufino renders obvious the second step of claim 6, *i.e.*, “when the attempt to send the message is unsuccessful, broadcasting from the first computer a connection port search message indicating that the first computer needs a connection.”

170. As described above, Rufino discloses a method for healing a disconnection of a first computer from a second computer. Rufino discloses that when a station cannot successfully send a message to its successor (because of an unplanned disconnect of the successor), the station broadcasts a connection port search message on the broadcast channel to find a third computer to which it can connect to maintain a 2-regular graph. To do this, the station sends a *who follows* query (connection port search message) on the broadcast channel to find the successor (the neighbor) of the disconnected station. Ex. 1011 at 0962 col. 2 ¶ 9 – 0963 col. 1 ¶ 1 (“During this *who follows* query the current token holder waits for a *set_successor* frame, until the end of a three-slot-time period.”). When applying this technique to Shoubridge employing the node removal technique of Denes in a 4-regular configuration, a POSITA would understand that the Rufino technique of

sending a query or broadcasting can be used to identify all four neighbors in the network, which Denes teaches must be found to return the network to a 4-regular configuration.

171. Claim 6(c): Shoubridge in view of Denes in further view of Rufino renders obvious “having a third computer not already connected to said first computer respond to said connection port search message in a manner as to maintain an m-regular graph.”

172. Denes discloses a method of connecting nodes in order to maintain an m-regular graph. *See* Section V above. To do this, Denes discloses connecting the neighbors of the leaving node to each other.

173. Rufino discloses broadcasting messages on the broadcast channel to maintain the integrity of the network, which in Rufino is a 2-regular graph. Ex. 1011 at 0960 col. 1 ¶¶ 3-4 (“Several types of protocol data frames are exchanged between peer MAC entities, in order to provide logical ring housekeeping functions, essential for normal ring membership management, and for the preservation of ring integrity in the presence of faults.”) Rufino discloses that the management functions include a variety of actions (such as finding out the successor of a failed station (*who_follows*), soliciting any station to respond at successor querying, resolving contentions, and establishing new successors).

174. Rufino discloses that when a station cannot successfully send a message to its successor (because of an unplanned disconnect of the successor), the station broadcasts a connection port search message on the broadcast channel to find a third computer to which it can connect to maintain a 2-regular graph. To do this, the station sends a *who follows* query (connection port search message) on the broadcast channel to find the successor (the neighbor) of the disconnected station. Ex. 1011 at 0962 col. 2 ¶ 9 – 0963 col. 1 ¶ 1 (“During this *who follows* query the current token holder waits for a *set_successor* frame, until the end of a three-slot-time period. Under a single station failure assumption, the current token holder will receive, within this period, a response from the successor of the failed station. This establishes a new successor for the current token holder and ends the recovery procedures.”).

175. Denes discloses a method for maintaining regularity of an m-regular graph when a node is removed from the graph. To do this, Denes connects the neighbors of the removed node to each other. Based on Denes, one of ordinary skill in the art would have understood that to maintain an m-regular network upon disconnection of a computer, it would have been obvious for the neighbors of the disconnected computer to connect to one of the other neighbors. Therefore, it would have been obvious to use the techniques of Rufino and Denes in connection with the 4-regular network disclosed in Shoubridge to have a third computer not

already connected to said first computer respond to said connection port search message in a manner as to maintain an m-regular graph.

176. A POSITA would have been motivated to combine the teachings of Shoubridge, Denes, and Rufino, which together disclose every limitation of this claim (see Section VIII.E above).

177. In my opinion, for at least the reasons discussed above, claim 6 is therefore rendered obvious under § 103 by Shoubridge in view of Denes in further view of Rufino.

(e) Claim 7

178. Shoubridge in view of Denes in further view of Rufino renders obvious the method of claim 7 which adds to claim 6 the requirement that “when a third computer receives the connection port search message and the third computer also needs a connection, sending a message from the third computer to the first computer proposing that the first computer and third computer connect.”

179. Rufino discloses that when the successor of the failed station (third computer), which needs a connection due to failure of its predecessor, receives the “who follows query” (port search message), then it will send a message (*set_successor* packet) to the predecessor of the failed station (first computer), and a connection between the first and third computer will be established. Ex. 1011 at 0962 col. 2 ¶ 9 – 0963 col. 1 ¶ 1 (“During this *who_follows* query the current token

holder waits for a *set_successor* frame, until the end of a three-slot-time period. Under a single station failure assumption, the current token holder will receive, within this period, a response from the successor of the failed station. This establishes a new successor for the current token holder and ends the recovery procedures.”).

180. As explained above, it would have been obvious to modify Shoubridge to employ the method of Denes to remove a node from a network. Denes teaches that, when removing a node in, *e.g.*, a 4-regular network, the nodes formerly connected to the removed node must connect. A POSITA would have considered it obvious to use the messaging protocols of Rufino in order for former neighbors of the removed node to locate each other and connect, thus rendering this claim obvious.

181. For the reasons addressed above, a POSITA would have been motivated to combine the teachings of Shoubridge, Denes, and Rufino, which together disclose every limitation of this claim (see Section VIII.E above).

182. In my opinion, for at least the reasons discussed above, claim 7 is therefore rendered obvious under § 103 by Shoubridge in view of Denes in further view of Rufino.

(f) Claim 8

183. Shoubridge in view of Denes and Rufino renders obvious the method of claim 8 which adds to claim 7 the requirement that “when the first computer receives the message proposing that the first computer and third computer connect, sending from the first computer to the third computer a message indicating that the first computer accepts the proposal to connect the first computer to the third computer.” Shoubridge in view of Denes and Rufino and further in view of U.S. Patent No. 4,700,185 (“Balph”) also renders obvious the method of claim 8.

184. As described above, Rufino teaches the connection method described in claim 7. Rufino also discloses this limitation of claim 8 because Rufino discloses in response to a *who follows* query, “the current token holder will receive . . . a response from the successor of the failed station. This establishes a new successor for the current token holder and ends the recovery procedure.” Ex. 1011 at 0962 col. 2 ¶ 9 – 0963 col. 1 ¶ 1.

185. Accordingly, a POSITA would have been motivated to combine the teachings of Shoubridge, Denes, and Rufino, which together disclose every limitation of this claim (*see also* Section VIII.E above).

186. In my opinion, for at least the reasons discussed above, claim 8 is therefore rendered obvious under § 103 by Shoubridge in view of Denes and Rufino.

187. To the extent that Rufino does not explicitly disclose this limitation, Balph teaches that, when two computers are connecting, it was old and well known to send messages between the two computers where a connection invitation is sent and accepted. Balph discloses that, in order to confirm the connection can be used, “protocols are used to determine how and when each station may transmit.” Ex. 1022 at 1:17-36. Balph discloses that when a first computer receives a message proposing that the first computer and a second computer connect, the first computer sends a message back to the second computer indicating that the first computer accepts the proposal to connect. *See* Ex. 1022 at cl. 1 (“a transmitting one of said stations for sending a request with response data frame to a predetermined receiving one of said stations on said network; said predetermined receiving station for responding to said transmitting station with a response data frame in response to said request with response data frame if said request with response mechanism thereof is enabled”). It would have been obvious to use the Balph messaging protocol when implementing the connection method disclosed in Rufino, as addressed above, in order to confirm the connection can be used. Accordingly, by combining the above “request with response” method of Balph with Rufino’s method discussed in claim 7, in particular using the “request with response” method for sending the connection-request message (set_successor packet) from the successor of the failed station (third computer), to the predecessor

of the failed station (first computer), then the first computer will send a response message back to the third computer indicating that it has received the connection-request message (and accepts the proposal). A POSITA would have been motivated to do so for any number of reasons that were also well known in the art, including, for example, maximizing the reliability and usefulness of the disclosed communications network. Indeed, any TCP connection would fulfill this claim because the TCP protocol requires that in order to start communication between two nodes, the first node must send a SYN packet, to which the receiving node must respond with a SYN-ACK packet, to which the originator must respond with a SYN packet.

188. Accordingly, a POSITA would have been motivated to combine the teachings of Shoubridge, Denes, Rufino, and Balph, which together disclose every limitation of this claim. *See also* Section VIII.E above.

189. In my opinion, for at least the reasons discussed above, claim 8 is therefore rendered obvious under § 103 by Shoubridge in view of Denes, Rufino, and Balph.

(g) Claim 9

190. Shoubridge in view of Denes and Rufino renders obvious the method of claim 9 which adds to claim 6 that “wherein each computer connected to the broadcast channel is connected to at least three other computers.” (See discussion

of claim 6 Preamble.) Because claim 9 includes substantively the same limitation as the preamble to claim 6, the same arguments presented above with respect to claim 6 apply to claim 9. Accordingly, in my opinion, for at least the reasons discussed above with respect to claim 6, claim 9 is therefore rendered obvious under § 103 by Shoubridge in view of Denes in further view of Rufino.

(h) Claim 10

191. Shoubridge in view of Denes in further view of Rufino renders obvious the method of claim 10 which adds to claim 6 “wherein the broadcasting includes sending the message to each computer to which the first computer is connected.”

192. As a threshold matter, claim 6 recites the sending of two different types of messages. It is unclear to me which message claim 10 is referring to. Regardless, my analysis remains the same since the addition of this limitation is trivial in light of the prior art.

193. Shoubridge discloses a network in which a participant uses a flooding algorithm to propagate a message to all of its neighbors and in which packets are broadcast to all participants. *See, e.g.*, Ex. 1005 at 1381 ¶ 1 (“In addition to successfully forwarding user traffic between source and destination nodes in a communications network, the routing function generally seek to optimize some performance criteria such as network utilization or throughput, or perhaps average

delay.”), 1382 ¶ 11 to 1383 ¶ 1 (“Any user packet transmitted from a node is copied and broadcast on all outgoing links. Intermediate transit nodes do not broadcast a packet on the same link that a packet was originally received on.”).

194. It would have been obvious to a POSITA to broadcast any message to each computer to which a first computer is connected because the well-known flooding technique results in messages being sent to the entire network, which, for instance, necessarily includes the computers connected to a computer originating a message.

195. Accordingly, a POSITA would have been motivated to combine the teachings of Shoubridge, Denes, and Rufino, which together disclose every limitation of this claim (see Section VIII.E above).

196. In my opinion, for at least the reasons discussed above, claim 10 is therefore rendered obvious under § 103 by Shoubridge in view of Denes in further view of Rufino.

(i) Claim 11

197. Preamble: To the extent the preamble is deemed to be a limitation of the claim, Shoubridge discloses the elements of the preamble, namely “[a] computer-readable medium containing instructions for controlling disconnecting of a computer from another computer, the computer and the other computer being

connected to a broadcast channel, said broadcast channel being an m-regular graph where m is at least 3.”

198. As discussed above in connection with claim 1 [Preamble], Shoubridge discloses controlling a dynamic network forming a broadcast channel, said broadcast channel forming an m-regular graph where m is at least 3. Denes discloses a method of disconnecting a first node from a second node, the first node and the second node being connected to a specific topology, said topology forming an m-regular graph where m is at least 3, as also discussed above in connection with the claim 1 [Preamble]. To the extent that Shoubridge and Denes do not specifically disclose storing instructions for implementing their disclosed techniques on a “computer readable medium” it was certainly well known to persons of ordinary skill for decades that software for implementing instructions in a network can be stored on a computer-readable medium.

199. Claim 11(a): Shoubridge in view of Denes in further view of Rufino renders obvious “a component that, when the computer decides to disconnect from the other computer, the computer sends a disconnect message to the other computer, said disconnect message including a list of neighbors of the computer.”

200. As discussed above in connection with claim 1(a), Denes discloses a method to disconnect a node from another node. Also as discussed above in connection with claim 1(a), Rufino discloses a component that, when the computer

decides to disconnect from the other computer, the computer sends a disconnect message to the other computer, said disconnect message including a list of neighbors of the computer seeking to disconnect.

201. The use of the term “component” in claim 11 refers generally to a collection of software instructions that carries out the steps recited in the elements of the claims. This is consistent with the specification of the ’147 patent, which states:

Each application program interfaces with a broadcaster component **602** for each broadcast channel to which it is connected. The broadcaster component may be implement as an object that is instantiated within the process space of the application program. Alternatively, the broadcaster component may execute as a separate process or thread from the application program. In one embodiment, the broadcaster component provides functions (*e.g.*, methods of class) that can be invoked by the application programs. The primary functions provided may include a connect function that an application program invokes passing an indication of the broadcast channel to which the application program wants to connect.

Ex. 1001 at 15:27-40.

202. A POSITA would have understood that the computer based-methods disclosed in Shoubridge and Rufino could, and ordinarily would, be implemented using software. Thus, the disclosures of Shoubridge and Rufino, when

implemented in software, would necessarily and inherently include “components” as used in claim 11.

203. Claim 11(b): Shoubridge in view of Denes in further view of Rufino renders obvious “a component that, when the computer receives a disconnect message from another computer, the computer broadcasts a connection port search message on the broadcast channel to find a computer to which it can connect in order to maintain an m-regular graph, said computer to which it can connect being one of the neighbors on said list of neighbors.”

204. As discussed above in connection with claim 1(b), Denes discloses connecting nodes in order to maintain an m-regular graph. Also as discussed above, Rufino discloses a component that, when the computer receives a disconnect message from another computer, the computer broadcasts a connection port search message on the broadcast channel to find a computer to which it can connect in order to maintain an m-regular graph, said computer to which it can connect being one of the neighbors on said list of neighbors.

205. A POSITA would have been motivated to combine the teachings of Shoubridge, Denes, and Rufino, which together disclose every limitation of claim 11 (see Section VIII.E above).

206. In my opinion, for at least the reasons discussed above, claim 11 is therefore rendered obvious under § 103 by Shoubridge in view of Denes in further view of Rufino.

(j) Claim 12

207. Shoubridge in view of Denes in further view of Rufino renders obvious the computer-readable medium of claim 12, which adds to claim 11 the additional limitation that “a component that, when the computer receives a connection port search message and the computer needs to connect to another computer, sends to the computer that sent the connection port search message a port connection message indicating that the computer is proposing that the computer that sent the connection port search message connect to the computer.”

208. Rufino discloses that when the successor of the failed station (third computer), which needs a connection due to failure of its predecessor, receives the “*who follows* query” (port search message), then it will send a message (*set_successor* packet) to the predecessor of the failed station (first computer), and a connection between the first and third computer will be established. Ex. 1011 at 0962 col. 2 ¶ 9 – 0963 col. 1 ¶ 1 (“During this *who_follows* query the current token holder waits for a *set_successor* frame, until the end of a three-slot-time period. Under a single station failure assumption, the current token holder will receive, within this period, a response from the successor of the failed station. This

establishes a new successor for the current token holder and ends the recovery procedures.”).

209. A POSITA would have been motivated to combine the teachings of Shoubridge, Denes, and Rufino, which together disclose every limitation of this claim (see Section VIII.E above). In particular, it would have been obvious to modify Shoubridge, as modified to implement the Denes node dropping technique, to use the recovery procedure disclosed in Rufino, thereby achieving all of the elements of claim 12.

210. In my opinion, for at least the reasons discussed above, claim 12 is therefore rendered obvious under § 103 by Shoubridge in view of Denes in further view of Rufino.

(k) Claim 13

211. Shoubridge in view of Denes and Rufino renders obvious claim 13, which adds to claim 12 the additional requirement that “a component that, when the computer receives a port connection message, connecting to the computer that sent the port connection message.” Shoubridge in view of Denes and Rufino further in view of Balph also renders obvious claim 13.

212. As described above, Rufino teaches the connection method described in claims 7 and 12. Rufino also discloses this limitation of claim 13 because Rufino discloses in response to a *who follows* query, “the current token holder will

receive . . . a response from the successor of the failed station. This establishes a new successor for the current token holder and ends the recovery procedure.” Ex. 1011 at 0962 col. 2 ¶ 9 – 0963 col. 1 ¶ 1.

213. A POSITA would have been motivated to combine the teachings of Shoubridge, Denes, and Rufino, which together disclose every limitation of this claim (*see* Section VIII.E above).

214. In my opinion, for at least the reasons discussed above, claim 13 is therefore rendered obvious under § 103 by Shoubridge in view of Denes in further view of Rufino.

215. To the extent that Rufino does not explicitly disclose this limitation, Balph teaches that, when two computers are connecting it was old and well known to send messages between the two computers where a connection invitation is sent and accepted. Balph discloses that, in order to confirm the connection can be used, “protocols are used to determine how and when each station may transmit.” Ex. 1022 at 1:17-36. Balph discloses that when a first computer receives a message proposing that the first computer and a second computer connect, the first computer sends a message back to the second computer indicating that the first computer accepts the proposal to connect. *See* Ex. 1022 at cl. 1 (“a transmitting one of said stations for sending a request with response data frame to a predetermined receiving one of said stations on said network; said predetermined

receiving station for responding to said transmitting station with a response data frame in response to said request with response data frame if said request with response mechanism thereof is enabled”). It would have been obvious to use the Balph messaging protocol when implementing the connection method disclosed in Rufino, as addressed above.

216. Accordingly, by combining the above “request with response” method of Balph with Rufino’s method discussed in claim 12, in particular using the “request with response” method for sending the connection-request message (set_successor packet) from the successor of the failed station (third computer), to the predecessor of the failed station (first computer), then the first computer will send a response message back to the third computer indicating that it has received the connection-request message (and accepts the proposal). A POSITA would have been motivated to do so for any number of reasons that were also well known in the art, including, for example, maximizing the reliability and usefulness of the disclosed communications network. Indeed, any TCP connection would fulfill this claim because the TCP protocol requires that in order to start communication between two nodes, the first node must send a SYN packet, to which the receiving node must respond with a SYN-ACK packet, to which the originator must respond with a SYN packet.

217. In addition, a POSITA would have been motivated to combine the teachings of Shoubridge, Denes, Rufino, and Balph, which together disclose every limitation of this claim. *See also* Section VIII.E above.

218. In my opinion, for at least the reasons discussed above, claim 13 is also therefore rendered obvious under § 103 by Shoubridge in view of Denes in further view of Rufino and Balph.

(I) Claims 14 And 16

219. Shoubridge in view of Denes in further view of Rufino renders obvious claims 14 and 16, which, respectively, add the following additional limitations to claim 11: “wherein the computers are connected via a TCP/IP connection” (cl. 14), and “wherein the broadcast channel is implemented using the Internet” (cl. 16).

220. As discussed above in connection with claims 4 and 5, Shoubridge discloses that its teachings are directed towards nodes in a communications network, and a POSITA would have understood the recitation of a communications network to include the Internet and networks based on TCP/IP. Also as discussed above in connection with claims 4 and 5, in the alternative, it would at a minimum have been obvious to implement the connections taught in Shoubridge as Internet or TCP/IP connections based on the knowledge of a POSITA. Further, as discussed above, if the Board believes that a specific reference is useful for the

additional elements of claims 14 and 16, Hirviniemi can be combined with the above references to demonstrate the obviousness of these additional elements.

221. A POSITA would have been motivated to combine the teachings of Shoubridge, Denes, Rufino, and/or Hirviniemi, which together disclose every limitation of this claim. *See also* Section VIII.E above.

222. In my opinion, for at least the reasons discussed above, claims 14 and 16 are therefore rendered obvious under § 103 by Shoubridge in view of Denes and Rufino and/or Hirviniemi.

(m) Claim 15

223. Shoubridge in view of Denes in further view of Rufino renders obvious claim 15, which adds to claim 11 “wherein the computers that are connected to the broadcast channel are peers.”

224. Shoubridge discloses a 64-node “manhattan grid network that has been wrapped around itself as a torus.” Ex. 1005 at 1383 ¶ 2. In a “torus” network like that disclosed by Shoubridge, there is no internal hierarchy or privileged participant and the failure of a single node does not disconnect or cause the breakdown of the network. All of the participants on such a network are peers. A POSITA would therefore have understood that the disclosed processors constitute peers connected in a peer-to-peer network, employing peer-to-peer communications.

225. Shoubridge further discloses a network in which “[e]ach node functions as a source of user traffic entering the network where traffic can be destined to all other nodes within the network,” and where “[t]he total load entering (and leaving) the network, $\gamma = \sum_{n=1}^N \gamma_n$, is evenly distributed across all N nodes.” Ex. 1005 at 1382-1383. A POSITA would therefore have understood that the disclosed processors constitute peers connected in a peer-to-peer network.

226. In the alternative, to the extent it is argued that this limitation is not expressly or inherently disclosed in Shoubridge, it would at a minimum have been obvious that the nodes of the network disclosed in Shoubridge could be implemented as peers and that the connections between those nodes would be peer-to-peer connections based on the knowledge of a POSITA. A POSITA would have found it advantageous, obvious, and routine, in light of Shoubridge’s disclosure that “[e]ach node functions as a source of user traffic entering the network where traffic can be destined to all other nodes within the network,” to implement the disclosed nodes as peers. *Id.* at 1, 2. It would have been clear to a POSITA that peer-to-peer connections could be implemented on the network disclosed in Shoubridge, using techniques that are well-known in the art and that would provide the expected, predictable results. A POSITA would have been motivated to do so, for any number of reasons that were also well known in the art, including, for example, improved reliability.

227. A POSITA would have been motivated to combine the teachings of Shoubridge, Denes, and Rufino, which together disclose every limitation of this claim (see Section VIII.E above).

228. In my opinion, for at least the reasons discussed above, claim 15 is therefore rendered obvious under § 103 by Shoubridge in view of Denes in further view of Rufino.

G. Ground 4: Invalidity Of Claims 1-16

229. The preceding section discussed three invalidity grounds. Specifically, I explained that: (Ground 1) Shoubridge, Denes, and Rufino renders obvious claims 1-16; (Ground 2) Shoubridge, Denes, Rufino, and Hirviniemi, in the alternative, renders obvious claims 4-5, 14, and 16; and (Ground 3) Shoubridge, Denes, Rufino, and Balph, in the alternative, renders obvious claims 8 and 13. As explained above, the claims addressed in Grounds 2 and 3 add trivial and common limitations that are disclosed in the prior art.

230. In this section I discuss Ground 4, which is a supplement to Ground 1. Ground 4 does not change my analysis for Grounds 2 and 3, which can easily depend from Ground 4 if the Board decides to institute trial on Ground 4 instead of Ground 1.

231. In Ground 1, I rely on Rufino as an example of a well-known messaging technique to accomplish the disconnection steps disclosed by Denes.

As discussed above, Rufino relates to a token ring, which is a 2-regular topology. While it would be within the knowledge of a POSITA to expand the messaging techniques from a 2-regular topology to a 4-regular topology, a specific reference is provided in this ground if the Board deems it necessary.

232. For example, Todd discloses a token grid network, which “is a two dimensional network structure arranged in R rows and C columns. An example for $R = C = 4$ is shown in Fig. 1.” Ex. 1019 at 1 col. 2 ¶ 4.

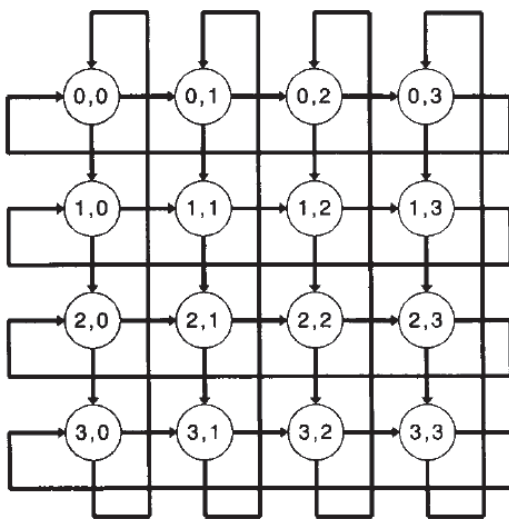


Fig. 1. The token grid (4×4 example).

233. Todd discloses that the reason it devised a token grid topology is because, in 2-regular systems, “the maximum throughput is severely restricted by the data rate of the shared channel.” *Id.* at 1 col. 1 ¶ 2. And “[i]n principle, this problem may be addressed using a network topology which permits multiple concurrent packet transmissions.” *Id.* In other words, Todd proposes adding additional connections in order to allow for better data transmission.

234. As discussed in Ground 1, a POSITA would have been motivated to apply the messaging techniques disclosed in Rufino to Shoubridge. While it would have been within a POSITA's knowledge of how to do so by July 2000, Todd (published in 1994) shows that the modification to Rufino can be done "in a very simple fashion" by overlapping token rings. *Id.* at 1 col. 2 ¶ 2.

235. Therefore, Ground 4 simply proposes an alternative ground for independent claims 1, 6, and 11 (which require *m* to be at least 3), and dependent claim 9 ("wherein each computer connected to the broadcast channel is connected to at least three other computers"). The inclusion of the other claims 2-5, 7-10, and 12-16 in this ground is merely because they depend from independent claims 1, 6, and 11, and my analysis above does not change with the inclusion of Todd in this ground.

IX. INVALIDITY OF THE '069 PATENT UNDER 35 U.S.C. § 103

236. In my opinion, the Challenged '147 Claims would have been obvious to a POSITA in view of the following prior art references or combinations of prior art references.

A. The Challenged Claims Of The '069 Patent

237. I have considered claims 1-17 of the '069 patent.

238. In the discussion below, I rely on Obraczka as the primary reference, and Denes and Shoubridge as secondary references. With regards to claims 4-6, 9,

and 10, I also refer to Valiant. I have concluded that each of the Challenged '069 Claims is invalid based on the references described below.

239. My opinion is based on, among other things, the knowledge of a POSITA, as set forth in Section V above.

B. Overview Of Obraczka

240. I have been asked to assume that Katia Obraczka, “Massively Replicating Services In Wide Area Internetworks” (Ph.D. Thesis, University of Southern California, December 1994) (“Obraczka”) is prior art. *See* Ex. 1010. Certain additional details are discussed in a subsequent paper by the same lead author, citing Obraczka, and which I have also been informed constitutes prior art. *See* Ex. 1009 (Katia Obraczka *et al.*, “A Tool for Massively Replicating Internet Archives: Design, Implementation, and Experience,” IEEE Proceedings of the 16th International Conference on Distributed Computing Systems, May 1996 (“Obraczka Paper”)). Unless otherwise stated, my opinions are based on and refer to Obraczka, rather than the Obraczka Paper.

241. Obraczka discloses an automated, scalable, and efficient replication tool for Internet information services. Ex. 1010 at 8 ¶ 5 – 10 ¶ 1. The proposed architecture builds a logical network topology over which nodes propagate updates to other nodes in their group. *E.g., id.* at xiv ¶ 4. Nodes (“sites” or “replicas”) compute and adopt a new logical k-connected topology every time they detect

changes in group membership and network topology in order to, *inter alia*, ensure the network is resilient, use the physical network efficiently, and restrict update propagation delays. *Id.*; *see also id.* at 20 (“Replicas may join or leave a replication group at any time. To join a group, a new replica copies a neighbor’s database, and floods its existence to the rest of the group. When a new member joins the group, a topology calculation is spawned. The resulting topology includes the new replica and is distributed to the group.”).

242. Obraczka’s method uses software-implemented techniques for distributing data and information (*e.g.*, updates, topology update messages, join request) to participants in communications networks (*id.*; Ex. 1010 at 15 ¶¶ 2-5), including wide-area networks and the Internet, and draws on work directed towards more general computer networking. *See, e.g., id.* at 1 ¶ 1-3.

243. Obraczka models communications networks as graphs in which each node (“site” or “replica”) “floods an update message to replicas that are its logical neighbors, according to the current logical topology.” Ex. 1010 at 19 ¶ 5; *see also* 14 ¶ 2 (“Since replicas flood updates to their neighbors in the logical topology, updates in one group make their way to all groups.”)

244. Flooding is an information broadcasting mechanism in which a node sends a message, update, or request to all of its neighbors, and each neighbor, upon receiving that piece of information for the first time, in turn broadcasts it to all of

its neighbors except for the one from which it received the message. Ex. 1010 at 5 ¶ 3 (“We also argue that efficient replication algorithms flood data between replicas . . . Although the word ‘flooding’ sounds inefficient, we claim that the application-level flooding scheme that we propose does use network bandwidth efficiently.”).

245. The “flood-d” process described in Obraczka performs two major functions: (i) estimating the network topology by gathering and reporting information regarding bandwidth and propagation delays and (ii) calculating a new, optimized, *k*-connected logical topology. Ex. 1010 at 19 ¶ 5 – 21 ¶ 3. Although any node running the flood-d program can manage both group membership and the logical network topology of the member nodes (*see, e.g.*, Ex. 1010 at 20 ¶ 4 (“Any flood-d replica can perform topology calculations.”)), one node is designated as the group master and is tasked with this management. Ex. 1010 at 20 ¶ 4.

246. Obraczka discloses that “a replica in a group is designated as the *group master* and is responsible for computing logical update topologies for the group.” *E.g.*, Ex. 1010 at 20; *see also id.* at 91 ¶ 3 (“The master is also responsible for supervising group membership.”). Indeed, in certain embodiments of Obraczka, “[a] process that wants to join the group sends a request to the master, who may accept or reject it.” *Id.* at 91 ¶ 3. Once a new node joins the network, the

group membership protocol will detect the addition. Ex. 1010 at 15 ¶ 2. (“Replicas may also join and leave a flooding group. The group membership protocol and physical topology estimation will eventually detect these changes, which will be reflected in the new k-connected graph the group computes.”) Then, a new logical network is created with the addition of the new node that is still k-connected. Ex. 1010 at 20 ¶ 6 (“Replicas may join or leave a replication group at any time. To join a group, a new replica copies a neighbor’s database, and floods its existence to the rest of the group. When a new member joins the group, a topology calculation is spawned. The resulting topology includes the new replica and is distributed to the group.”)

247. Based on the reported estimates and factoring in the join request from the new node, the group master sends out a topology update message, which is flooded through the network. *Id.* at 15 ¶ 2 – 16 ¶ 1. These topology update messages contain the new group membership and topology. *Id.* When a group member receives a topology update message, it forwards it on to its neighbors according to the current topology before committing to the proposed, updated topology. *Id.*

248. Obraczka characterizes the “topology computation problem as a graph theory problem.” *Id.* at 36 ¶ 2. Obraczka utilizes graph theory operations and principles to achieve its goal of building a logical update topology that is k-

connected for resilience, uses the physical network efficiently, and restricts update propagation delays. *Id.* at 36 ¶ 3. Obraczka further explains that “clearly” the shape of the logical topology “depends on the requirements of the problem being solved.” *Id.* at 45 ¶ 3.

249. In order to explore the problem to be solved more fully, Obraczka discusses two types of graphs: k -connected and k -connected regular graphs. *Id.* at 37 ¶ 2. As defined above in the context of being k -connected, Obraczka states: “[a] graph G is said to be k -connected if no removal of any $k - 1$ vertices [sic] together with all their incident edges disconnects G .” *Id.* As defined above in the context of being m -regular, Obraczka states: “[a] graph G is said to be a k -connected regular graph if all its vertices [sic] are exactly k -connected.” *Id.* Thus, Obraczka uses “ k -connected regular” to mean the same thing as an m -regular and exactly m -connected graph.

250. Obraczka also discusses, *inter alia*, two competing considerations: diameter and edge cost. The diameter of a graph is defined as the maximum shortest path between any two vertices. *Id.* at 37 ¶ 2. The diameter goes to the speed a message travels through the network – the larger the diameter, the longer it takes for a message to travel from one node to another, and vice versa. *Id.* at 45 ¶¶ 2-6. The edge cost is the cost of building or maintaining edges. *Id.* at 45 ¶¶ 2-6.

The larger the number of edges used, the more resources are consumed by the network. *Id.*

251. When new nodes seek to join a network, topologies are generated according to certain transformations. *Id.* at 43 ¶ 2. “These transformations consist of some combination of the basic add and delete edge operations.” *Id.* at 43 ¶ 2. These operations seek to obtain regularity: “[d]elete randomly chooses a pair of connected nodes with degree greater than the required connectivity and deletes the edge connecting them.” *Id.* at 43 ¶ 3 (emphasis added). “Add selects a pair of nodes that are not connected and adds an edge connecting them.” *Id.* For instance, “one of the transformations we use is Steiglitz’s *x-change* operation. It randomly selects a pair of connected nodes a and b. Then it selects another pair of connected nodes c and d such that c is not connected to a and d is not connected to b or vice versa.” *Id.* Then, a separate *X-change* operation is used to delete edges a-b and c-d, and adds edges a-c and b-d. *Id.* “The *x-change* operation has the property of not changing the degree of any node.” *Id.*

252. Obraczka studies the extremes of a graph that is k-connected regular and balances their costs and benefits. Ex. 1010 at 60 ¶ 4 – 61 ¶ 2. On the one hand, Obraczka studies a fully-connected topology and notes that it has a minimum diameter but maximum total edge cost. *Id.* Obraczka also studies a 2-connected regular topology and notes that it has a minimum total edge cost but a maximum

diameter. Ex. 1010 at 48 ¶¶ 3-4. Obraczka also studies 3-connected topologies. *Id.* at 48 ¶¶ 5-6.

C. Overview Of Denes

253. As discussed in Section V, Denes discloses methods of maintaining a non-complete m -regular topology when removing a node, where m is even. These methods are mathematical formulas that would have been well known to a POSITA by July 2000. I will explain below that Denes discloses the exact procedures claimed by the '069 patent of (a) identifying a pair of participants, (b) disconnecting the participants of the identified pair from each other, and (c) connecting each participant of the identified pair of participants to the seeking participant.

D. Overview Of Shoubridge

254. In addition to teaching flooding on an m -regular and m -connected network topology (see Section VIII.B), Shoubridge also studies a communication protocol that accommodates network dynamics and, in particular, the unavailability of routing tables at the nodes. *See, e.g.*, Ex. 1005 at 1381 ¶ 9 (“While shortest path algorithms are less costly to operate in a quasi-static network environment, flooding may actually consume less resources in a very dynamic network.”). Shoubridge identifies critical circumstances that arise due to network dynamics, such as when individual nodes lack a next node entry in their routing tables, and

specifically aims to address these situations. *Id.* at 1384 ¶ 9. In order to cope with these important situations when network behavior becomes very dynamic, despite certain trade-offs, Shoubridge employed a non-routing table based flooding protocol. *See, e.g.*, Ex. 1005 at 1384 ¶ 5 (“It is reasonable to conclude that a large network similar to the one modelled, would require a flooding procedure if the network is to operate in a very dynamic, or potentially very dynamic environment.”).

255. Alternatives to networks that used routing tables for data communications were known to a POSITA. For instance, Van Leeuwen explains that “a routing method is required in” a computer network to allow nodes to “communicate messages to each other.” Ex. 1031 at 298. “Normally this is provided by a routing table of size $O(N)$ at each node, which N is the number of nodes in the network.” *Id.* Yet, Van Leeuwen also recognizes that “routing can be achieved without the need for any routing tables at all, provided the nodes of the network are suitably labeled and the routing is restricted to a spanning tree.” *Id.* Van Leeuwen considers “generating optimum or near-optimum routing schemes” that do not rely on routing tables. *Id.* Van Leeuwen then provides certain proofs for a number of propositions regarding routing without using routing tables in any respect for networks with a number of different configurations.

E. Combination Of The Teachings Of Obraczka, Denes, And Shoubridge

256. A POSITA would have been motivated to combine the teachings of Obraczka with the teachings of Denes and Shoubridge for several separate and independent reasons:

(a) Obraczka And Denes Address The Problem Of Maintaining A Non-Complete M-Regular Topology When Nodes Are Added, And Shoubridge Discloses A Well-Known Topology

257. Obraczka, Denes, and Shoubridge all deal with the very specific topology claimed in the '069 patent – m-regular networks including when the network is incomplete and greater than 3. *See, e.g.*, Section IX.B-D. Additionally, Obraczka, Denes, and Shoubridge each recognizes that networks may dynamically expand or contract. *See, e.g.*, Section IX.B-D. A POSITA would therefore have been aware of, and motivated to combine the teachings of all three references, because they were in the same technical field, and addressed the same technical problems. For the reasons stated above, a POSITA thus would have been motivated to combine the teachings of Obraczka, Denes, and Shoubridge to ensure a non-complete m-regular topology when adding a node.

(b) Obraczka Teaches A Tool For Reconfiguring A Logical Topology, And Denes And Shoubridge Provide Additional Compatible Ways To Increase Reliability And Efficiency

258. A POSITA would have been motivated to combine the teachings of Obraczka, Denes, and Shoubridge because, as discussed above, Obraczka teaches a

tool for reconfiguring a logical topology, and Denes and Shoubridge provide additional compatible ways to increase scalability, reliability, and efficiency.

259. Obraczka discloses an automated, scalable, and efficient tool for reconfiguring the network topology of its participants. Ex. 1010 at 8 ¶ 5 – 10 ¶ 1. The proposed architecture builds a logical network topology over which nodes propagate updates to other nodes in their group. *Id.* at xiv ¶ 4. Nodes (“sites” or “replication groups”) compute and adopt a new logical k-connected topology every time they detect changes in group membership and network topology in order to, *inter alia*, ensure the network is resilient, use the physical network efficiently, and restrict update propagation delays. *Id.* at 20 ¶ 6. Obraczka discloses that any transformations may be used, and can consist of some combination of basic add and delete edge operations. *Id.* at 43 ¶ 2. After each transformation, Obraczka’s system checks to see whether a feasible topology is generated, and whether the current topology is satisfactory merely depends on the choice of the objective function of the problem to be solved. *Id.* at 45 ¶ 3.

260. Denes discloses a mathematical formula to add a node to a network to maintain a non-complete m-regular topology. It would be straightforward and obvious to apply the simple formula disclosed in Denes to Obraczka’s system. A POSITA would be motivated to do so for a number of reasons. For instance, it is generally appealing in systems to make incremental changes (such as the

incremental operation disclosed in Denes) because incremental changes minimize disruption to a network. Obraczka also uses a set of incremental modifications to build its logical topology. *See, e.g.*, Ex. 1010 at 43 ¶ 3. A POSITA considering Obraczka would naturally consider applying Denes' incremental change method to a network.

261. Shoubridge discloses an exemplary 64-node, 4-regular network topology as an example of a network environment. Ex. 1005 at 1383 (describing network model). In view of Obraczka, it is further straightforward to implement the well-known structure of a 4-regular network disclosed in Shoubridge. Because Obraczka addresses a tool for reconfiguring a logical topology (see Ex. 1010 at xiv ¶ 4), a POSITA would be motivated to use the 4-regular network disclosed in Shoubridge because a 4-regular network is well known to have certain characteristics, such as being resilient and efficient, for a large number of nodes (such as 64). A POSITA would have known to try the 4-regular network as well as other well-known topologies such as rings.

262. Additionally, as discussed in IX.B, Obraczka discloses that its goal is to create a k-connected network. If k is set to 4 in Obraczka's topology tool, then it would create a 4-regular network, such as the one disclosed in Shoubridge. This is because under a uniform cost metric, where each edge, e.g., costs 1, it is natural to create a k-regular topology because the cheapest topology to create is where

each node has k connections at each node. Obraczka's description of a "k-connected regular network" is consistent with this idea. See Ex. 1010 at 37 ¶ 2. Obraczka's description of choosing which edge to delete is also consistent with this idea. *Id.* at 43 ¶ 3 ("[d]elete randomly chooses a pair of connected nodes with degree greater than the required connectivity and deletes the edge connecting them.") (emphasis added).

263. If, for example, the underlying physical network happens to be a toroid when Obraczka's processes are run, then the minimum cost spanning tree and additional lowest cost edges that the topology calculator computes would likely result in a torus-shaped logical network. See Section IX.B. Wherever a torus is the underlying physical network, a toroid overlay network is likely the one that Obrackza's system will deploy. *Id.* With these considerations in mind, and since Obraczka addresses massively replicating services in wide-area internetworks (see Ex. 1010, Title), a POSITA would be motivated to use a structure that was well-known to be resilient, efficient, and suitable for a large numbers of participants (*i.e.*, scalable), as disclosed in Shoubridge. Ex. 1005 at 1381 ¶ 2 ("As networks increase in size, with more users demanding seamless connectivity, mobility and a high level of availability, the task of routing becomes an increasingly complex problem to network providers.").

264. In view of Shoubridge, it is further straightforward that Obraczka's flooding system does not use routing tables. *E.g.*, Ex. 1005 at 1381 ¶ 7 (“[f]looding based routing procedures do not maintain routing tables . . .”). For example, as network participants increase, “the task of routing becomes an increasingly complex problem to network providers.” Ex. 1005 at 1381 ¶ 2. As participants grow and the frequency of network changes increases, maintaining routing tables becomes prohibitively expensive, an obstacle to network scaling and efficiency. *Id.* at 1381 ¶ 3 (“As topology or traffic loads change more frequently, the network becomes dynamic and maintaining accurate routing information at individual nodes comes at a higher cost.”). Also, the commensurate risk and consequence of inadequate routing table maintenance threatens network reliability, supplying another motive to seek a substitution. *Id.* (“Inaccurate routing can result in lost or delayed traffic causing end-to-end retransmissions which further load the network.”). Accordingly, a POSITA would be motivated to avoid all of these downsides associated with routing table uncertainties by adopting flooding as a design choice. *See also, e.g., id.* at (regarding routing table maintenance, “This additional load is itself an unwanted overhead which could be avoided with a better routing strategy.”).

265. For the reasons stated above, a POSITA thus would have been motivated to combine the teachings of Obraczka, Denes, and Shoubridge to ensure a non-complete m-regular topology when adding a node.

F. Grounds 1-2: Invalidity Of Claims 1-17

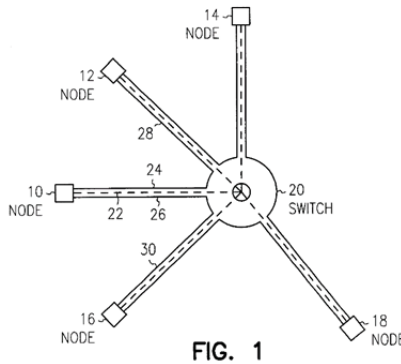
(a) Claim 1

266. Preamble: To the extent the preamble is deemed to be a limitation of the claim, Obraczka in view of Denes in further view of Shoubridge renders obvious “[a] computer-based, non-routing table based, non-switch based method for adding a participant to a network of participants, each participant being connected to three or more other participants.” Obraczka discloses a computer-based, non-switch based method for adding a participant to a network of participants, each participant being connected to two, three, and more other participants. Denes discloses a method for adding a participant to a network of participants, each participant being connected to three or more other participants. Shoubridge discloses a computer-based non-routing table based, non-switch based dynamic computer network, where each participant is connected to three or more other participants.

267. Obraczka discloses an automated, scalable, and efficient replication tool for replicating files and reconfiguring the network topology of its participants. Ex. 1010 at 8 ¶ 5 – 10 ¶ 1. The proposed tool builds a logical network topology

over which nodes propagate updates to other nodes in their group, which happens when a node joins a group. *E.g., id.* at xiv ¶ 4. Nodes (“sites” or “replicas”) compute and adopt a new logical k-connected topology every time they detect changes in group membership and network topology in order to, *inter alia*, ensure the network is resilient, use the physical network efficiently, and restrict update propagation delays. *Id.*; *see also id.* at 20 (“Replicas may join or leave a replication group at any time. To join a group, a new replica copies a neighbor’s database, and floods its existence to the rest of the group. When a new member joins the group, a topology calculation is spawned. The resulting topology includes the new replica and is distributed to the group.”).

268. Obraczka is further a non-switch based method for adding a participant because a common switch is not used for adding a participant to a network. A POSITA would readily understand that flooding is not used in such switch-based systems. This is consistent with how applicant defined a “switch-based” system in the prosecution history. The applicants distinguished their invention from Gilbert’s on the basis that Gilbert used a switch-based mechanism to add a node and their invention did not. Ex. 1103 at 16-17. The definition of “switch” that Gilbert appears to provide is that all the nodes use a common switch for adding a participant to a network, as illustrated in Figure 1 of Gilbert, provided below. *See also* Ex. 1028 at Figs. 5, 6, 8, and 10.



269. Obraczka discloses a network of participants where each participant is connected to two, three, and more other participants. Obraczka studies the extremes of a graph that is k -connected regular and balances their costs and benefits. Ex. 1010 at 60 ¶ 4 - 61 ¶ 2. On the one hand, Obraczka studies a fully connected topology and notes that it has a minimum diameter but maximum total edge cost. *Id.* Obraczka also studies a 2-connected regular topology and notes that it has a minimum total edge cost but a maximum diameter. Ex. 1010 at 48 ¶¶ 3-4. Obraczka also studies 3-connected topologies. *Id.* at 48 ¶¶ 5-6.

270. Denes discloses a method of adding a node while maintaining a non-complete m -regular graph that, as discussed below, is precisely the same as the claimed method. See Section V above. Denes further discloses a non-switch based graph structure that is 3-regular (*see, e.g.,* Fig. 2) because a common switch is not used for adding a node.

271. Shoubridge discloses a dynamic computer network in which “the distribution of traffic loads and network topologies may vary from nearly static to

very dynamic. This dynamic behavior may vary both in space and in time.” Ex. 1005 at 1381 Abstract. In a dynamic network, computers will fail, become congested, or enter and leave the network. Ex. 1005 at 1381 ¶ 3. (“Links may fail or become congested, nodes or users could be mobile, resulting in changes to source-destination paths.”)

272. Shoubridge discloses the use of a flooding algorithm to propagate a message to all of its neighbors and in which packets are broadcast to all participants. Shoubridge discloses that its “[f]looding based routing procedures do not maintain routing tables” Ex. 1005 at 1381 ¶ 7.

273. Shoubridge also discloses networks in which each computer is connected to four (*i.e.*, at least three) neighbors in the network of participants. For example, Shoubridge discloses a 64-node network in which each participant is connected to four neighbors. Ex. 1005 at 1383 ¶ 2 (“A 64 node network with connectivity of degree 4 is modeled as G . The network is a large regular graph forming a Manhattan grid network that has been wrapped around itself as a torus to avoid edge effects.”).

274. Claim 1(a): Obraczka in view of Denes in further view of Shoubridge renders obvious “identifying a pair of participants of the network that are connected wherein a seeking participant contacts a fully connected portal computer, which in turn sends an edge connection request to a number of randomly

selected neighboring participants to which the seeking participant is to connect.” Obraczka discloses that a seeking participant contacts a fully connected portal computer, which in turn sends an edge connection request to a number of randomly selected neighboring participants to which the seeking participant is to connect. Denes discloses the exact step of “identifying a pair of participants of the network that are connected.”

275. Obraczka discloses a “fully connected portal computer” that is in charge of adding a new participant to a group. Obraczka discloses that “a replica in a group is designated as the *group master* and is responsible for computing logical update topologies for the group.” *E.g.*, Ex. 1010 at 20; *see also id.* at 91 ¶ 3 (“The master is also responsible for supervising group membership.”). Indeed, in certain embodiments of Obraczka, “[a] process that wants to join the group sends a request to the master, who may accept or reject it.” *Id.* at 91 ¶ 3. Once a new node joins the network, the group membership protocol will detect the addition. Ex. 1010 at 15 ¶ 2. (“Replicas may also join and leave a flooding group. The group membership protocol and physical topology estimation will eventually detect these changes, which will be reflected in the new k-connected graph the group computes.”) Then, a new logical network is created with the addition of the new node that is still k-connected. Ex. 1010 at 20 ¶ 6 (“Replicas may join or leave a replication group at any time. To join a group, a new replica copies a neighbor’s

database, and floods its existence to the rest of the group. When a new member joins the group, a topology calculation is spawned. The resulting topology includes the new replica and is distributed to the group.”)

276. A POSITA would understand that the *group master* is fully connected because it is a part of the preexisting k-connected group. *Id.* In addition to being a part of the group, the group master is either “k-connected” or “k-connected regular.” See Ex. 1010 at 37. This is consistent with how the ’069 Patent defines “fully connected,” which states that “[a] computer that is . . . connected to four neighbors [in a 4-regular example] is in the ‘fully connected state’.” Ex. 1101 at 5:50-54.

277. Obraczka and Denes disclose exactly the claimed method of adding a node while maintaining a non-complete m-regular graph. See Section V above. Specifically, Denes teaches to identify and then disconnect two (*i.e.*, $k/2$ pairs where $k = 4$) neighbor pairs (p_1p_2), (p_3p_4), and then to connect new node p to each of p_1 , p_2 , p_3 , and p_4 . The resulting graph will continue to be 4-regular.

278. A POSITA would understand that the method of Denes can be readily applied to the system of Obraczka for adding new participants. Because Obraczka teaches that a *group master* is in charge of all additions of new members, it would be obvious to use the *group master* to implement the node adding method of Denes. This implementation would have been obvious involve the master

communicating the connection request (*i.e.*, the “edge connection request” of the claim) to the affected nodes, *i.e.*, p_1 , p_2 , p_3 , and p_4 . Indeed, a POSITA would have understood that it would not have been possible to implement the Denes connection method without such communications.

279. A POSITA would further understand that the methods of Obraczka and Denes readily teach choosing the neighboring participants in a “random” manner, as that term was used by applicants. During prosecution of the ’069 Patent, the applicants explained that their invention was different from the Gilbert reference because Gilbert disclosed that “[t]he portal node that is contacted provides information regarding a neighboring node that is adjacent to the seeking node; the selection of the neighboring node is not random.” Ex. 1103.

280. Obraczka discloses that when new nodes seek to join a network, topologies are generated according to certain random transformations. *Id.* at 43 ¶ 2. “These transformations consist of some combination of the basic add and delete edge operations.” *Id.* at 43 ¶ 2. Obraczka’s transformations seek to obtain regularity: “Delete randomly chooses a pair of connected nodes with degree greater than the required connectivity and deletes the edge connecting them.” *Id.* at 43 ¶ 3 (emphasis added). “Add selects a pair of nodes that are not connected and adds an edge connecting them.” *Id.* For instance, “one of the transformations we use is Steiglitz’s *x-change* operation. It randomly selects a pair of connected nodes a and

b. Then it selects another pair of connected nodes c and d such that c is not connected to a and d is not connected to b or vice versa.” *Id.* Thus, using randomness in the selection of elements of a network was certainly well known before the alleged invention.

281. Furthermore, as discussed in Section V, Denes is consistent with choosing neighbors in a random manner. Denes discloses a generalized formula for maintaining a non-complete regular graph when adding a node. This generalized algorithm can be applied to any set of neighboring nodes.

282. Because Obraczka teaches that a *group master* is in charge of all additions of new members, it would be obvious to use the *group master* to implement the node adding method of Denes. This implementation would have been obvious to involve the master communicating the connection request (*i.e.*, the “edge connection request” of the claim) to the affected nodes, *i.e.*, p₁, p₂, p₃, and p₄. Indeed, it would not be possible to implement Denes’ connection method without such communications. Therefore, it would have been obvious to choose the participant used in the adding method of Denes in a random manner.

283. Claim 1(b): Obraczka in view of Denes in further view of Shoubridge renders obvious “disconnecting the participants of the identified pair from each other.” Obraczka discloses disconnecting participants in order to connect a participant to a network of participants, and Denes discloses to disconnect the

participants of the identified pair in exactly the manner and order specified in this claim element.

284. Denes discloses a method of adding a node while maintaining a non-complete m-regular graph that is precisely the same as the claimed method. See Section V above. Specifically, as noted above, Denes explicitly discloses that the participants in the identified pairs, *e.g.*, (p_1, p_2) , (p_3, p_4) , in the 4-regular example, are disconnected from one another.

285. Claim 1(c): Obraczka in view of Denes in further view of Shoubridge renders obvious “connecting each participant of the identified pair of participants to the seeking participant.” Obraczka discloses connecting participants to a network of participants, and Denes discloses to connect each participant of the identified pair of participants to the seeking participant in exactly the manner and order specified in this element of the claim.

286. Denes discloses a method of adding a node while maintaining a non-complete m-regular graph that is precisely the same as the claimed method. See Section V above. Specifically, as noted above, Denes explicitly discloses that the participants in the identified pairs, *e.g.*, (p_1, p_2) , (p_3, p_4) , in the 4-regular example, are connected to the seeking participant p after they are disconnected from each other.

287. As discussed above in Section IX.E, a POSITA would have been motivated to combine the teachings of Obraczka, Denes, and Shoubridge.

288. Accordingly, in my opinion, for at least the reasons discussed above, claim 1 is therefore rendered obvious under § 103 by Obraczka in view of Denes and in further view of Shoubridge.

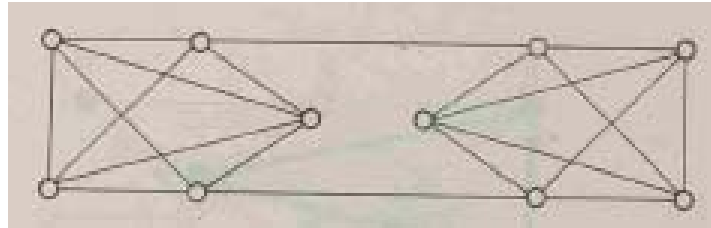
(b) Claim 2

289. Obraczka in view of Denes in further view of Shoubridge renders obvious the method of claim 2, which adds to claim 1 “wherein each participant is connected to 4 other participants.” Obraczka discloses the use of k-connected regular and fully connected regular systems, and Denes and Shoubridge disclose that k can be equal to four.

290. As noted above, Obraczka teaches “k-connected regular” and fully connected regular systems, which means that all vertices are exactly k-connected. *See* Ex. 1010 at 37 ¶ 2, 60 ¶ 4 – 61 ¶ 2. To the extent that Obraczka does not explicitly teach an example where $k = 4$, it would have been obvious to a POSITA to implement a 4-regular system in Obraczka.

291. Furthermore, Denes, in fact, does teach a specific example where $k = 4$. Denes discloses and illustrates graphs of degree 4 (also known as “4-regular”). *See, e.g.*, Ex. 1017 at 365 (generally discussing a “k-regular graph”), at 367 Fig. 2 (illustrating a k-regular graph where $k = 4$). As shown in Fig. 2 of

Denes, each node is connected to four other nodes in this non-complete graph example.



Ex. 1017: Fig. 2

292. Shoubridge also discloses a 4-regular network (see discussion of claim 1 above), in which each participant is connected to four other participants. Ex. 1005 at 1383 ¶ 2 (“A 64 node network with connectivity of degree 4 is modeled as G . The network is a large regular graph forming a manhattan grid network that has been wrapped around itself as a torus to avoid edge effects”). Because four is an even number, it follows that Shoubridge also discloses networks “wherein each participant is connected to an even number of other participants.”

293. Finally, as discussed above in Section IX.E, a POSITA would have been motivated to combine the teachings of Obraczka, Denes, and Shoubridge.

294. In my opinion, for at least the reasons discussed above, claim 2 is therefore rendered obvious under § 103 by Obraczka in view of Denes in further view of Shoubridge.

(c) Claim 3

295. Obraczka in view of Denes in further view of Shoubridge renders obvious the method of claim 3, which adds to claim 1 “wherein the identifying of a pair includes randomly selecting a pair of participants that are connected.” It is not clear how claim 3 further narrows claim 1. In any event, as discussed above in connection with claim 1(a), Denes discloses identifying a pair of participants that are connected to each other. Selecting these participants “randomly,” would have been obvious, particularly given the explicit teachings of Obraczka, as discussed above in connection with claim 1(a), of the random selection of nodes. Denes is consistent with selecting nodes randomly.

296. As discussed above in Section IX.E, a POSITA would have been motivated to combine the teachings of Obraczka, Denes, and Shoubridge.

297. In my opinion, for at least the reasons discussed above, claim 3 is therefore rendered obvious under § 103 by Obraczka in view of Denes in further view of Shoubridge.

(d) Claim 4

298. Obraczka in view of Denes and Shoubridge in further view of Valiant renders obvious the method of claim 4, which adds to claim 3 “wherein the randomly selecting of a pair includes sending a message through the network on a randomly selected path.” Obraczka discloses an automated, scalable, and efficient

replication tool for a distributed network (Ex. 1010 at 8 ¶ 5 – 10 ¶ 1), and Valiant discloses a technique for efficient communication of routing requests in distributed networks. *See, e.g.*, Ex. 1027 at 1 col. 1 ¶ 1 – col. 2 ¶ 2.

299. Valiant considers “the problem of realizing routing requests in packet switching networks that are sparse in connections and yet have to support heavy parallel traffic.” Ex. 1027 at 1 col. 1 ¶ 1. Valiant concludes that its technique of sending a message on a random path is an efficient technique for communicating routing requests in distributed networks. Ex. 1027 at 1 col. 2 ¶¶ 3-5. Valiant explains the “randomization” of a message’s path means there is “a fixed probability distribution (independent of the rest of the permutation) that specifies for each path from the source to the destination the probability that that path will be taken.” Ex. 1027 at 1 col. 1 ¶ 2 – col. 2 ¶ 1. Valiant discloses that its distributed algorithm consists “essentially of two consecutive phases. The first phase sends each packet to a random node in the network, while the second then forwards it to the desired destination.” Ex. 1027 at 1 col. 2 ¶ 3.

300. The use of random selection for sending messages was well-known by July 2000, and taught specifically in Valiant. A POSITA would understand that the method of Valiant can be readily applied to the system of Obraczka to improve performance. It would have been obvious to combine because Obraczka and Valiant’s teachings are in the same technical field and addressed the same

technical problems. For instance, Obraczka discloses a distributed network (Ex. 1010 at 8 ¶ 5 – 10 ¶ 1), and Valiant discloses a technique for optimal routing strategy in distributed networks.

301. Further, as discussed above in Section IX.E, a POSITA would have been motivated to combine the teachings of Obraczka, Denes, and Shoubridge.

302. In my opinion, for at least the reasons discussed above, claim 4 is therefore rendered obvious under § 103 by Obraczka in view of Denes in further view of Shoubridge and Valiant.

(e) Claim 5

303. Obraczka in view of Denes in further view of Shoubridge in further view of Valiant renders obvious the method of claim 5, which adds to claim 4 “wherein when a participant receives the message, the participant sends the message to a randomly selected participant to which it is connected.” As discussed in claim 4, Valiant discloses precisely the claimed step of receiving a message and then retransmitting it to a randomly selected participant. Ex. 1027 at 1 col. 1 ¶ 2 – 1 col. 2 ¶ 1 (“If randomization is used by the algorithm then obliviousness means that for each source-destination pair there is a fixed probability distribution (independent of the rest of the permutation) that specifies for each path from the source to the destination the probability that that path will be taken.”); *id.* at 1 col. 2 ¶ 3 (“The algorithms . . . consist essentially of two consecutive phases. The first

phase sends each packet to a random node in the network, while the second then forwards it to the desired destination.”). As discussed in claim 4, it would have been obvious to use the disclosed transmission method of Valiant in Obraczka’s system modified to implement Denes’ method of adding a participant.

304. Further, as discussed above in Section IX.E, a POSITA would have been motivated to combine the teachings of Obraczka, Denes, and Shoubridge.

305. In my opinion, for at least the reasons discussed above, claim 5 is therefore rendered obvious under § 103 by Obraczka in view of Denes in further view of Shoubridge and Valiant.

(f) Claim 6

306. Obraczka in view of Denes, Shoubridge, and Valiant renders obvious the method of claim 6, which adds to claim 4 “wherein the randomly selected path is proportional to the diameter of the network.” Obraczka defines what a diameter is and the impact it has on a network. A POSITA would understand that to complete a broadcast, a message must travel a path that is proportional to the diameter of the network.

307. Obraczka explains that the diameter of a graph is defined as the maximum shortest path between any two vertices. Ex. 1010 at 37 ¶ 2. The length of a diameter is proportional to the maximum time that it takes for a message to travel through the network – the larger the diameter, the longer it takes for a

message to travel from across the entire network. *Id.* at 45 ¶¶ 2-6. A POSITA would understand that the time to complete a broadcast is proportional to the diameter of the network because the diameter is the longest minimal distance between nodes in the network.

308. Alternatively, Valiant discloses a technique for efficient communication of routing requests in distributed networks. Valiant discloses that messages should be sent “twice as long as the diameter of the graph.” Ex. 1027 at Abstract. For the reasons discussed in claims 4 and 5, it would have been obvious to use the disclosed transmission method of Valiant in Obraczka’s system modified to implement Denes’ method of adding a participant.

309. Further, as discussed above in Section IX.E, a POSITA would have been motivated to combine the teachings of Obraczka, Denes, and Shoubridge.

310. In my opinion, for at least the reasons discussed above, claim 6 is therefore rendered obvious under § 103 by Obraczka in view of Denes, Shoubridge, and Valiant.

(g) Claim 7

311. Obraczka in view of Denes in further view of Shoubridge renders obvious the method of claim 7, which adds to claim 1 “wherein the participant to be added requests a portal computer to initiate the identifying of the pair of participants.” As discussed in the following paragraphs, Obraczka discloses the

participant to be added requests a portal computer to initiate the connection, and the teachings of Denes, when applied to Obraczka, indicate that the group master of Obraczka should initiate the identifying of the pair of participants.

312. As discussed above in connection with claim 1(a), Obraczka discloses a “group master” which corresponds to the “fully connected portal computer” that the new node contacts for connection requests. Ex. 1010 at 91 ¶ 3. Obraczka states that, at least in one embodiment, the participant who wants to join “sends a request to the master” and that the master is in charge of implementing the request. *Id.*

313. Also discussed above in claim 1(a), when the Obraczka system is implementing the method of Denes to add a node, the master computer will send the messages necessary to make the connections. It therefore would have been obvious to initiate the identifying of those connections as claimed.

314. As discussed above in Section IX.E, a POSITA would have been motivated to combine the teachings of Obraczka, Denes, and Shoubridge.

315. In my opinion, for at least the reasons discussed above, claim 7 is therefore rendered obvious under § 103 by Obraczka in view of Denes in further view of Shoubridge.

(h) Claim 8

316. Obraczka in view of Denes in further view of Shoubridge renders obvious the method of claim 8, which adds to claim 7 “wherein the initiating of the identifying of the pair of participants includes the portal computer sending a message to a connected participant requesting an edge connection.” As discussed in connection with claims 1(a) and 7, when the Obraczka system is implementing the method of Denes to add a node, the master computer will send the messages necessary to make the connections. This implementation would have been obvious to involve the master communicating the connection request (*i.e.*, the “edge connection request” of the claim) to the affected nodes, *i.e.*, p₁, p₂, p₃, and p₄. Indeed, a POSITA would have understood that it would not have been possible to implement the Denes connection method without such communications.

317. As discussed above in Section IX.E, a POSITA would have been motivated to combine the teachings of Obraczka, Denes, and Shoubridge.

318. In my opinion, for at least the reasons discussed above, claim 8 is therefore rendered obvious under § 103 by Obraczka in view of Denes in further view of Shoubridge.

(i) Claim 9

319. Obraczka in view of Denes in further view of Shoubridge renders obvious the method of claim 9, which adds to claim 8 “wherein the portal

computer indicates that the message is to travel a distance proportional to the diameter of the network and wherein the participant that receives the message after the message has traveled that distance is one of the identified pair of participants.” As discussed below, a POSITA would understand that Obraczka discloses the portal computer (*group master*) to indicate that a message is to travel a distance proportional to the diameter of the network, and Denes further specifies that the portal computer (*group master*) should initiate the identifying of the distinct and different pair of participants after the message has traveled that distance.

320. Obraczka teaches that the diameter of a graph is defined as the maximum shortest path between any two vertices. Ex. 1010 at 37 ¶ 2. The diameter goes to the speed a message travels through the network – the larger the diameter, the longer it takes for a message to travel from one node to another, and vice versa. *Id.* at 45 ¶¶ 2-6. A POSITA would understand that the time to complete a broadcast is proportional to the diameter of the network because the diameter is the longest minimal distance between any two nodes in the network.

321. Further, as discussed in claim 1(a), a POSITA would understand that the method of Denes can be readily applied to the system of Obraczka for adding new participants. Denes teach to identify and then disconnect two (*i.e.*, $k/2$ pairs where $k = 4$) neighbor pairs (p_1, p_2) , (p_3, p_4) , where the neighbor pairs are each different nodes. Because Obraczka teaches that a group master is in charge of all

additions of new members, it would be obvious to use the group master to implement the method disclosed in Denes. This implementation would have been obvious to involve the master communicating the connection request to the affected nodes, *i.e.*, p_1 , p_2 , p_3 , and p_4 .

322. As discussed above in Section IX.E, a POSITA would have been motivated to combine the teachings of Obraczka, Denes, and Shoubridge.

323. In my opinion, for at least the reasons discussed above, claim 9 is therefore rendered obvious under § 103 by Obraczka in view of Denes in further view of Shoubridge.

324. To the extent that this limitation is not disclosed by the combination of Obraczka, Denes, and Shoubridge, Valiant discloses this limitation. As discussed in claims 4-6, Valiant considers “the problem of realizing routing requests in packet switching networks that are sparse in connections and yet have to support heavy parallel traffic.” Ex. 1027 at 1 col. 1 ¶ 1. Valiant concludes that its technique of sending a message on a random path is an efficient technique for communicating routing requests in distributed networks. Ex. 1027 at 1 col. 2 ¶¶ 3-5. Valiant explains the “randomization” of a message’s path means there is “a fixed probability distribution (independent of the rest of the permutation) that specifies for each path from the source to the destination the probability that that path will be taken.” Ex. 1027 at col. 1 ¶ 2 – col. 2 ¶ 1. In order to arrive at the

randomly selected locations, Valiant discloses that its distributed algorithm consists “essentially of two consecutive phases. The first phase sends each packet to a random node in the network, while the second then forwards it to the desired destination.” Ex. 1027 at 1 col. 2 ¶ 3.

325. The use of random selection for sending messages was well-known by July 2000, and taught specifically in Valiant. A POSITA would understand that the method of Valiant can be readily applied to the system of Obraczka to improve performance. It would have been obvious to combine because Obraczka and Valiant’s teachings are in the same technical field and addressed the same technical problems. For instance, Obraczka discloses a distributed network (Ex. 1010 at 8 ¶ 5 – 10 ¶ 1), and Valiant discloses a technique for optimal routing strategy in distributed networks.

326. In my opinion, in the alternative, claim 9 is rendered obvious under § 103 by Obraczka in view of Denes, Shoubridge, and Valiant for at least the reasons discussed above.

(j) Claim 10

327. Obraczka in view of Denes in further view of Shoubridge and Valiant renders obvious the method of claim 10, which adds to claim 9 “wherein the certain distance is twice the diameter of the network.” As discussed above in

connection with claim 9, Valiant discloses when the certain distance is twice the diameter of the network. Ex. 1027 at Abstract.

328. Valiant provides one transmission method that would send a message on a random path that is twice the diameter of the network. Ex. 1027 at Abstract. Valiant discloses that it addresses the problem of realizing routing requests in networks that are sparse in connections and yet have to support heavy parallel traffic. Ex. 1027 at 1 col. 1 ¶ 1. When using Valiant's method for communicating the connection request message that is sent from the group master in Obraczka to affected nodes, the message identifying the neighbor pairs must travel a distance that is twice the diameter of the network. Therefore, it would have been obvious to broadcast the messages being implemented by the combination of Obraczka and Denes a distance of twice the diameter of the network, as taught by Valiant in order to obtain the benefits of such a broadcast as described by Valiant.

329. In my opinion, for at least the reasons discussed above, claim 10 is therefore rendered obvious under § 103 by Obraczka in view of Denes, Shoubridge, and Valiant.

(k) Claims 11 And 12

330. Obraczka in view of Denes in further view of Shoubridge renders obvious the methods of claims 11 and 12, which add to claim 1 "wherein the participants are connected via the Internet" (cl. 11), and "wherein the participants

are connected via TCP/IP connections” (cl. 12). Obraczka discloses the use of the Internet and TCP/IP.

331. Obraczka discloses its method being applicable to Internet communications. See Ex. 1010 at Abstract. Obraczka also discloses “IP multicasting,” which is an Internet Protocol. See Ex. 1010 at 109 ¶ 5 (“Finally, we evaluated the use of IP multicasting as flood-d’s primary update propagation mechanism.”). A POSITA would have understood the recitation of an “Internet Protocol” to include the Internet. The dominant protocol of the Internet was (and is) TCP/IP, which dates back to at least the mid-1970s.

332. As discussed above in Section IX.E, a POSITA would have been motivated to combine the teachings of Obraczka, Denes, and Shoubridge.

333. In my opinion, for at least the reasons discussed above, claims 11 and 12 are therefore rendered obvious under § 103 by Obraczka in view of Denes in further view of Shoubridge.

(I) Claim 13

334. Obraczka in view of Denes in further view of Shoubridge renders obvious the method of claim 13, which adds to claim 1 “wherein the participants are computer processes.” Obraczka and Shoubridge both disclose this limitation.

335. I note that the ’069 patent specification uses the terms “computer” and “computer process” synonymously. Ex. 1101 at 4:25-33 (“In one embodiment,

each computer is connected to four other computers, referred to as neighbors. (Actually, a process executing on a computer is connected to four other processes executing on this or four other computers.)”).

336. In any event, Obraczka discloses that a participant is a computer process. Obraczka uses software-implemented techniques for distributing data and information to participants in communication networks (Ex. 1010 at 15 ¶¶ 2-5), including wide-area networks and the Internet, and draws on work directed towards general computer networking. *See id.* at 1 ¶ 1-3. Notably, Obraczka discloses that its “replicas” and “sites” reside on computers. *E.g.*, Ex. 1010 at 4 ¶ 2 (“If replicas reside on mobile computers, these replicas will provide service even during disconnection.”).

337. Shoubridge also discloses the method of claim 14 wherein the participants are computer processes. For example, Shoubridge discloses that “[t]ypically, computer processing power and memory requirements within network nodes are of low relative cost when compared with transmission overhead requirements in networks using low to moderate link capacities.” Ex. 1005 at 1381 ¶ 51. A POSITA would have understood that the references to “computer processing power” and “memory” in Shoubridge to mean that the disclosed “participants” include computer processes.

338. As discussed above in Section IX.E, a POSITA would have been motivated to combine the teachings of Obraczka, Denes, and Shoubridge.

339. In my opinion, for at least the reasons discussed above, claim 13 is therefore rendered obvious under § 103 by Obraczka in view of Denes in further view of Shoubridge.

(m) Claim 14

340. Preamble: To the extent the preamble is deemed to be a limitation of the claim, Obraczka in view of Denes in further view of Shoubridge renders obvious “[a] computer-based, non-switch based method for adding nodes to a graph that is m-regular and m-connected to maintain the graph as m-regular, where m is four or greater.” As discussed in connection with the claim 1 [Preamble], the combination of Obraczka, Denes, and Shoubridge renders obvious a computer-based, non-switch based method for adding nodes to a graph that is m-regular to maintain the graph as m-regular, where m is four or greater. The combination of Obraczka and Shoubridge further renders obvious that the claimed method is applicable to a situation where the graph begins as m-connected.

341. As discussed in Section IX.B, Obraczka discusses two types of graphs: k-connected and k-connected regular graphs. *Id.* at 37 ¶ 2. Obraczka studies the extremes of a graph that is k-connected regular and balances their costs and benefits. Ex. 1010 at 60 ¶ 4-61 ¶ 2. On the one hand, Obraczka studies a fully

connected topology and notes that it has a minimum diameter but maximum total edge cost. *Id.* Obraczka also studies a 2-connected regular topology and notes that it has a minimum total edge cost but a maximum diameter. Ex. 1010 at 48 ¶¶ 3-4. Obraczka also studies 3-connected topologies. *Id.* at 48 ¶¶ 5-6.

342. While Obraczka does not seem to teach a specific example where a graph is 4-connected, it would have been obvious to a POSITA to implement a 4-connected system in Obraczka. The 4-regular network topology disclosed by Shoubridge is also 4-connected (*i.e.*, m -connected for $m = 4$), meaning that it would take the failure of at least four nodes to divide the network into two or more separate parts. See Ex. 1005 at 1383 ¶ 2. A “manhattan grid network” with “connectivity of degree 4” that is “wrapped around itself as a torus” is a 4-connected graph – each node is connected to four other nodes in a grid format and the edge nodes are wrapped around to the opposite side (*i.e.*, left to right and top to bottom) forming a torus. See also, Section IX.D, above (discussing connectedness of the disclosed topology in Shoubridge).

343. The topology disclosed by Shoubridge is thus both 4-regular and 4-connected, *i.e.*, m -regular and m -connected for $m=4$.

344. Claim 14(a): Obraczka in view of Denes in further view of Shoubridge renders obvious “identifying p pairs of nodes of the graph that are connected, where p is one half of m , wherein a seeking node contacts a fully

connected portal node, which in turn sends an edge connection request to a number of randomly selected neighboring nodes to which the seeking node is to connect.” As discussed in claim 1(a), Denes discloses identifying p pairs of nodes of the graph that are connected, where p is one half of m , and it would have been obvious to implement this method in Obraczka.

345. Claim 14(b): Obraczka in view of Denes in further view of Shoubridge renders obvious “disconnecting the nodes of each identified pair from each other.” As discussed in claim 1(b), Denes discloses disconnecting the nodes of each identified pair from each other, and it would have been obvious to implement this method in Obraczka.

346. Claim 14(c): Obraczka in view of Denes in further view of Shoubridge renders obvious “connecting each node of the identified pairs of nodes to the seeking node.” As discussed in claim 1(c), Denes discloses connecting each node of the identified pairs of nodes to the seeking node, and it would have been obvious to implement this method in Obraczka.

347. As discussed above in Section IX.E, a POSITA would have been motivated to combine the teachings of Obraczka in view of Denes and in further view of Shoubridge.

348. Accordingly, in my opinion, for at least the reasons discussed above, claim 14 is therefore rendered obvious under § 103 by Obraczka in view of Denes in further view of Shoubridge.

(n) Claim 15

349. Obraczka in view of Denes in further view of Shoubridge renders obvious the method of claim 15, which adds to claim 14 “wherein identifying of the p pair of nodes includes randomly selecting a pair of connected nodes.” As discussed in connection with the discussion of claim 13, Denes discloses identifying a pair of participants that are connected to each other. Selecting these participants would have been obvious, particularly given the explicit teachings in Obraczka and Denes, as discussed in connection with the discussion of claim 1, of the random selection of nodes.

(o) Claim 16

350. Obraczka in view of Denes in further view of Shoubridge renders obvious the method of claim 16, which adds to claim 14 “wherein the nodes are computers and the connections are point-to-point communications connections.” Obraczka and Shoubridge each disclose this element.

351. I note that the '069 patent specification appears to use the terms “computer” and “computer process” synonymously. Ex. 1101 at 4:25-33 (“In one embodiment, each computer is connected to four other computers, referred to as

neighbors. (Actually, a process executing on a computer is connected to four other processes executing on this or four other computers.)”).

352. In any event, Obraczka uses software-implemented techniques for distributing data and information to participants in communication networks (Ex. 1010 at 15 ¶¶ 2-5), including wide-area networks and the Internet, and draws on work directed towards general computer networking. *See id.* at 1 ¶ 1-3. Notably, Obraczka discloses that its “replicas” and “sites” reside on computers. *E.g.*, Ex. 1010 at 4 ¶ 2 (“If replicas reside on mobile computers, these replicas will provide service even during disconnection.”). These sites/replicas connect with one another using point-to-point communications connections. *See id.* at 20 ¶ 2 (“Since flood-d uses TCP for point-to-point update transmission”). Obraczka further discloses that its method is compatible with and uses existing IP multicasting, which also uses point-to-point communication connections. *Id.* at 88 ¶ 4 (“The MBONE uses virtual point-to-point links, or *tunnels*, to transmit multicast packets between DVMRP routers in different routing domains.”); *id.* at 100 ¶ 4-101 ¶ 1 (“Replicas recover from inconsistencies by getting missing updates from their peers through reliable point-to-point communication.”); *id.* at 102 ¶ 1 (“This scheme fits well with the IP multicasting is currently deployed on the Internet, where there are regions that can directly support multicast routing connected by point-to-point tunnels.”); *id.* at 114 ¶ 1 (“Like group replicas, corner

replicas use flood-d to periodically perform point-to-point consistency checks between themselves.”).

353. Shoubridge also discloses the method of claim 16 wherein the nodes are computers. For example, Shoubridge discloses that “[t]ypically, computer processing power and memory requirements within network nodes are of low relative cost when compared with transmission overhead requirements in networks using low to moderate link capacities.” Ex. 1005 at 1381 ¶ 5. A POSITA would have understood the references to “computer processing power” and “memory” in Shoubridge to mean that the disclosed “network nodes” include computers. Shoubridge discloses that these nodes use connections that are point-to-point communications connections. *See, e.g.*, Ex. 1005 at 1383 ¶ 3 (“User packets entering the network arrive with Poisson arrival rate γ_n at each node n . Each user packet entering the network at node n is randomly assigned a destination node d such that $d \in G$ and $d \neq n$. The total load entering (and leaving) the network . . . is evenly distributed across all N nodes.”).

354. As discussed above in Section IX.E, a POSITA would have been motivated to combine the teachings of Obraczka in view of Denes and in further view of Shoubridge.

355. Accordingly, in my opinion, for at least the reasons discussed above, claim 16 is therefore rendered obvious under § 103 by Obraczka in view of Denes in further view of Shoubridge.

(p) Claim 17

356. Obraczka in view of Denes in further view of Shoubridge renders obvious the method of claim 17, which adds to claim 14 “wherein m is even.” As discussed above in connection with claim 2, Obraczka discloses where m is equal to two (an even number); Denes and Shoubridge disclose where m is four (an even number). To the extent that Obraczka does not explicitly disclose a specific example of where $k = 4$, it would have been obvious to a POSITA to implement a 4-regular system in Obraczka, at least in view of the specific teaching of such a system in Shoubridge. As a POSITA would readily understand, four is an even number.

357. As discussed above in Section IX.E, a POSITA would have been motivated to combine the teachings of Obraczka in view of Denes and in further view of Shoubridge.

358. Accordingly, in my opinion, for at least the reasons discussed above, claim 17 is therefore rendered obvious under § 103 by Obraczka in view of Denes in further view of Shoubridge.

X. SECONDARY CONSIDERATIONS

359. At this time, I have not aware of any objective factors that the Board might consider in making a determination as to whether the Challenged '147 or '069 Claims would have been obvious to a person of ordinary skill in the art. I understand that if Acceleration Bay presents evidence of objective factors that it argues support a finding of non-obviousness, I may be asked to review such evidence and to formulate an opinion.

XI. CONCLUSION

360. A number of prior art references, including those discussed above and as provided in the "List of Exhibits," show that the various claim elements of the challenged claims of the '147 and '069 patents were well known and that the claims, taken as a whole, are invalid as obvious based on these references. A POSITA would have been familiar with the materials provided in the "List of Exhibits" and the full range of teachings they contain. To the extent any of the claim limitations are not explicitly disclosed in one reference, such limitations would have been well-known to a POSITA and combinable to produce predictable results. As such, in my opinion, the challenged claims would have been in the possession of or obvious to a POSITA from the disclosures in one or more of certain prior art references. In summary, a POSITA would be familiar with each of the commonplace claimed elements through multiple references (including those

cited herein) and would also easily be able to combine such elements into a working broadcasting network (including the network claimed in the '147 and '069 patents). In my opinion, all the prior art references discussed in this Declaration should be considered by the Board while determining invalidity of the '147 and '069 patents.

361. I reserve the right to supplement my opinions in the future to respond to any arguments that Acceleration Bay or its expert(s) may raise and to take into account new information as it becomes available to me.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001 and that such willful false statements may jeopardize the validity of the application or any patent issued thereon. If called to testify as to the truth of the matters stated herein, I could and would testify competently.

I declare under penalty of perjury that the foregoing is true and correct.

Executed on this 11th day of March 2016, in Boston, Massachusetts.

A handwritten signature in black ink, appearing to read "David Karger", written in a cursive style.

David R. Karger

APPENDIX A

DAVID R. KARGER

Work Address

MIT Computer Science and Artificial Intelligence Laboratory
32 Vassar St.
Cambridge, MA 02139
(617)-258-6167

Home Address

21 Craigie St.
Cambridge, MA 02138
(617)-491-9592
email: karger@mit.edu

Interests

Information management: personal information management, human computer interaction, databases, machine learning, social computing, semantic web

Data Structures and Algorithms; Graph Algorithms; Approximation Algorithms; Probabilistic Analysis. Applications in communication, networking, biology, natural language processing.

Experience

MASS. INSTITUTE OF TECHNOLOGY
January 1995–present
Professor. Cambridge, MA

AKAMAI TECHNOLOGIES
October 1998–April 2001
Research Scientist. Cambridge, MA

AT&T BELL LABORATORIES
September 1994–January 1995
Postdoctoral Fellow. Murray Hill, NJ

XEROX PALO ALTO RESEARCH CENTER
1991–1995
Consultant in Information Retrieval. Palo Alto, CA

STANFORD UNIVERSITY
1991–1994
Teaching assistant in theory of computation, data structures and algorithms, and graduate theory courses. Lectured, prepared assignments and exams, led sections. Stanford, CA

Education

STANFORD UNIVERSITY
1991–1995
Ph.D. in Computer Science. *Advisor:* Professor Rajeev Motwani Stanford, CA

CAMBRIDGE UNIVERSITY
1990
Certificate of Advanced Study in Mathematics. Cambridge, England

HARVARD UNIVERSITY
1985–89
B.A. *summa cum laude* in Computer Science and Physics. Cambridge, MA

- Dissertation** *Randomized Algorithms for Graph Optimization Problems.* Presents new algorithms for several graph problems. Gives presently best sequential and parallel algorithm for finding minimum cuts in undirected graphs. Presents random sampling approaches to speeding up graph problems such as minimum spanning tree and maximum flow. Gives approximation algorithms for network design problems.
- Honors**
- NAS award for Initiatives in Research, 2003
 - David and Lucille Packard Foundation Fellowship, 1997–2002
 - Alfred P. Sloane Foundation Fellowship, 1997–1999
 - Mathematical Programming Society Tucker Prize, 1997
 - National Science Foundation CAREER award, 1996-1999
 - ACM Doctoral Dissertation Award, 1994
 - Hertz Foundation Graduate Fellowship, 1993-94
 - National Science Foundation Graduate Fellowship, 1990-93
 - Churchill Foundation Fellowship for study at Cambridge University, 1989-90
 - Phi Beta Kappa, 1989
- Professional Activities**
- Referee for *Journal of the ACM*, *ACM Transactions on Information Systems*, *Mathematical Programming*, *Journal of Algorithms*, *SIAM Journal on Computing*, *Information Processing Letters*, *Journal of Computer and System Sciences*,
- Program Committees: 1996 ACM-SIAM Symposium on Discrete Algorithms, 1996 and 1998 IEEE Symposium on Foundations of Computer Science, 2002 International Peer to Peer Systems conference (IPTPS), 2005 ACM Symposium on Theory of Computing, 2007 and 2009 Conference on Innovative Database Systems Research (CIDR), 2008 International Semantic Web Conference (ISWC) , 2010 Conference on Intelligent User Interfaces (IUI), 2008 and 2010 Conference on Information Retrieval (SIGIR), 2009 Visual Interface to the Social and Semantic Web, Conference on Human-Computer Interaction (CHI) 2010, World Wide Web conference (WWW) 2003, 2009, 2010
- Chair: 2009 International Semantic Web Conference (ISWC) 2009
- Member SIAM, ACM, AMS
- ACM Fellow

MASSACHUSETTS INSTITUTE OF TECHNOLOGYSchool of Engineering Faculty Personnel Record

Date: January 25, 2012

Full Name: David Ron Karger

Department: Electrical Engineering
and Computer Science

1. Date of Birth: 1 May 1967

2. Citizenship: U.S.

3. Education:

<u>School</u>	<u>Degree</u>	<u>Date</u>
Harvard University	A.B.	1989
Cambridge University	Math Certificate	1990
Stanford University	Ph.D.	1994

4. Title of Ph.D. Thesis:

Random Sampling in Graph Optimization Problems

5. Principal Fields of Interest:

Analysis of Algorithms (graph algorithms, combinatorial optimization, randomization).
 Applications of Algorithms in Systems, Networking, and Artificial Intelligence
 Information Retrieval (text databases, digital libraries, filtering and routing).

6. Name and Rank of Other Department Faculty in the Same Field:

Erik Demaine, Assistant Professor
 Shafi Goldwasser, Professor
 Charles Leiserson, Professor
 Silvio Micali, Professor
 Ron Rivest, Professor
 Madhu Sudan, Associate Professor
 Piotr Indyk, Assistant Professor

7. Name and Rank of Faculty in Other Departments in the Same Field:

Bonnie Berger, Associate Professor (Mathematics)
 Michel Goemans, Associate Professor (Mathematics)
 F. Thomson Leighton, Professor (Mathematics)
 James Orlin, Professor (Sloane)
 Dan Spielman, Assistant Professor (Mathematics)
 Andreas Schulz, Associate Professor
 Santosh Vempala, Assistant Professor (Mathematics)

8. Non-M.I.T. Experience:

<u>Employer</u>	<u>Position</u>	<u>Beginning</u>	<u>Ending</u>
MIT Lincoln Laboratory	Summer Technical Staff	1985	1989
Harvard University	Teaching Fellow	Sep. 1987	May. 1989
Microsoft Corporation	Intern	Jun. 1990	Sep. 1990
Xerox PARC	Intern	Jun. 1991	Jun. 1993
AT&T	Postdoctoral Fellow	Sep. 1995	Jan. 1995
Akamai Technologies	Research Scientist	Jun. 1998	April 2001

9. History of M.I.T. Appointments:

<u>Rank</u>	<u>Beginning</u>	<u>Ending</u>
Assistant Professor	Jan. 1995	Jul. 1998
Esther & Harold E. Edgerton Career Development Associate Professor	1998	2001
Associate with Tenure	2001	Jun. 2004
Professor	Jul. 2004	present

10. Consulting Record:

<u>Firm</u>	<u>Beginning</u>	<u>Ending</u>
Xerox PARC	Jun. 1991	1999
NEC	1995	1998
IBM	1996	2000
Google	1999	2001
Vanu	2002	present

11. Department and Institute Committees, Other Assigned Duties:

<u>Activity</u>	<u>Beginning</u>	<u>Ending</u>
Undergraduate Counselor (Dept.)	Sep 1995	present
Doctoral Clean Slate	Dec 1995	Dec 1996
6-A Faculty Advisor for AIG	1996	1999
ACM and Sprowls thesis awards	June 1996/1998/2000	
Graduate Admissions	1997, 1998, 2001	
Graduate Recruitment	Jan 1997	present
Education	Jan 2003	present

12. Professional Service:

<u>Activity</u>	<u>Beginning</u>	<u>Ending</u>
Program Committee Member, 1995 DAGS Summer Institute on Electronic Publishing		1995
Program Committee Member, 1996 ACM-SIAM Symposium on Discrete Algorithms		1995
Program Committee Member, 1996 IEEE Symposium on Foundations of Computer Science		1996

Activity

Program Committee Member,

Workshop on Survey Methodology and Web Demographics

Program Committee Member,

1998 IEEE Symposium on Foundations of Computer Science

Program Committee Member,

1999 Workshop on Algorithm Engineering
and Experimentation

Program Committee Member,

1999 Workshop on Randomization and Approximation
Techniques in Computer Science

Program Committee Reviewer,

1999 SIGIR Conference on Information Retrieval

Judge, Tucker Prize

Mathematical Programming Society

Member at Large,

ACM SIGACT Executive Committee

Chair,

ACM SIGACT Electronic Publications Committee

Editorial Board

Journal of Information Retrieval (Kluwer Publishers)

Technical Advisor

MIT/HP DSpace digital library project

Program Committee Member,

1st International Workshop on Peer
to Peer Systems

Program Committee Member,

12th World Wide Web Conference (WWW 2003)

Program Committee Member,

ALENEX Workshop on Algorithm Engineering and Experimentation Program Committee Memb
Conference on Principles of Distributed Computing (PODC 2004)

Program Committee Member,

Second Workshop on Semantics in Peer-to-Peer and Grid Computing (SemPGRID04)

Program Committee Member,

Symposium on Theory of Computing (STOC 05)

Member,

DARPA Information Science and Technology Advisory Panel (ISAT)

Program Committee Member,

International Semantic Web Conference (ISWC06)

Activity

Program Committee Member,

Workshop on Semantic Web in Use at ISWC2006

Program Committee Member,

Semwiki2006 - From Wiki to Semantics. First workshop on semantic wikis, at 3rd Annual European Semantic Web Conference

Program Committee Member,

Conference on Innovative Database Research (CIDR07)

13. Awards Received:

<u>Award</u>	<u>Received</u>
Harvard College Scholarship	1988, 1989
A.B. <i>summa cum laude</i> , Harvard University	1989
Churchill Foundation Fellowship	1989
NSF Graduate Fellowship	1990
Hertz Foundation Graduate Fellowship	1993
ACM Doctoral Dissertation Award	1994
NSF CAREER faculty development award	1996
Alfred P. Sloane Foundation Fellowship	1997
Mathematical Programming Society Tucker Prize	1997
David and Lucille Packard Foundation Fellowship	1998
SIAM Outstanding Paper Prize	2000
Usenix Security Best Student Paper Award	2002
National Academy of Sciences Young Innovator Award	2003

14. Current Organization Membership:

Organization

ΦBK

ACM: Special Interest Group on Algorithms and Computation Theory
Special Interest Group on Information Retrieval

SIAM

AMS

15. Patents and Patent Applications Pending:

1. Douglas Cutting, David Karger, Jan Pedersen, and John Tukey, "Scatter-Gather: A Cluster-based Method and Apparatus for Browsing Large Document Collections," Patent No. 5,442,778, August 15, 1995
2. Douglas Cutting, David Karger, Jan Pedersen, "Method of Constant Interaction-Time Clustering Applied to Document Browsing,:" Patent No. 5,483,650, January 9 1996.
3. David Karger, Eric Lehman, Matt Levine, Daniel Lewin, Tom Leighton, Rina Panigrahy. "Method and Apparatus for Distributing Requests Among a Plurality of Resources." Patent No. 6,30,618, August 6, 2002.

4. David Karger, Eric Lehman, Matt Levine, Daniel Lewin, Tom Leighton, Rina Panigrahy. "Method for Caching Files in Networks." TLO case 7874S.
5. Andrew D. Berkheimer, William Bogstad, Rizwan Danidina, Ken Iwamoto, David R. Karger, Brian A. Kim, William Koffel, F. Thompson Leighton, Daniel Lewin, Lucas J. Matkins, Alexander Sherman, Adrian M. Soviani, David Wang, Zhi-Hui Xu, Yoav Yerushalmi. "Internet Content-Distribution Software." TLO Case 8197S.

Teaching Experience of David Ron Karger

1. Teaching Experience

Term	Subject Number	Title	Role	Course Type	Course evaluation survey given
ST95	6.001	Structure and Interpretation of Computer Programs	Recitation (2 sections)	Lecture	Yes
FT95	6.042	Mathematics for Computer Science	Co-in charge, lectures development	Lecture	Yes
ST96		parental release			
FT96	6.891	Randomized Algorithms	In charge, lectures development	Lecture	Yes
ST97	6.042	Mathematics for Computer Science	In charge, lectures	Lecture	Yes
FT97	6.854	Advanced Algorithms	In charge, lectures	Lecture	Yes
ST98	6.033	Computer Systems	Recitations (2 sections)	Design	Yes
FT98	6.856	Randomized Algorithms	In charge, lectures	Lecture	Yes
ST99		parental release			
FT99	6.854	Advanced Algorithms	In charge, lectures, development	Lecture	Yes
ST00	6.042	Mathematics for Computer Science	Co-in charge, lectures, development	Lecture	Yes
FT00	6.856	Randomized Algorithms	In charge, lectures	Lecture	Yes
ST01	6.042	Mathematics for Computer Science	Co-in charge, lectures development	Lecture	Yes
01-02		sabbatical			
FT02	6.856	Randomized Algorithms	In charge, lectures	Lecture	Yes
ST02		parental release			
FT03	6.854	Advanced Algorithms	co-in charge, lectures, development	Lecture	Yes

2. Teaching Evaluation Data

Term	Subject Number	Total students registered	Survey form used	Instructor Teaching Quality	Overall Course Quality
ST95	6.001	364	HKN	NA	5.6
FT95	6.042	120	HKN	NA	NA
ST96		parent release			
FT96	6.891	15	HKN	6.0	5.9
ST97	6.042	136	HKN	5.2	4.5
FT97	6.854	22	HKN	4.8	5.4
ST98	6.033	355	HKN	NA	5.6
FT98	6.856	40	HKN	5.7	5.6
ST99		parent release			
FT99	6.854	50	HKN	5.2	5.8
ST00	6.042	110	HKN	5.2	4.9
FT00	6.856	50	HKN	6.0	6.0
01-02		sabbatical			
FT02	6.856	37	HKN	6.2	6.0

1. Book Chapters

1. M. X. Goemans, D. R. Karger, and J. Kleinberg. “Randomized Algorithms”. In M. Dell’Amico, F. Maffioli, and S. Martello, editors, *Annotated Bibliographies in Combinatorial Optimization*. John Wiley & Sons, August 1997.
2. D. R. Karger, C. Stein, and J. Wein. “Scheduling Algorithms”. In M. J. Atallah, editor, *Algorithms and Theory of Computation Handbook*. CRC Press, 1998.
3. D. R. Karger. “Random Sampling in Graph Optimization Problems: A Survey”. In S. Rajasekaran, P. Pardalos, J. Reif, , and J. Rolim, editors, *Handbook on Randomization*. Kluwer Academic Press, 2001.

2. Papers in Refereed Journals

1. D. R. Karger, D. Koller, and S. J. Phillips. “Finding the Hidden Path: Time Bounds for All-Pairs Shortest Paths”. *SIAM Journal on Computing*, 22(6):1199–1217, December 1993. A preliminary version appeared in Proceedings of the 32nd Annual Symposium on the Foundations of Computer Science.
2. D. R. Karger, P. N. Klein, and R. E. Tarjan. “A Randomized Linear-Time Algorithm to Find Minimum Spanning Trees”. *Journal of the ACM*, 42(2):321–328, March 1995.
3. C. J. Alpert, T. C. Hu, J. H. Huang, A. B. Kahng, and D. Karger. “Prim-Dijkstra Tradeoffs for Improved Performance-Driven Routing Tree Design”. *IEEE Transactions on Computer Aided Design*, 14(7):890–895, July 1995.
4. D. R. Karger, S. Phillips, and E. Torng. “A Better Algorithm for an Ancient Scheduling Problem”. *Journal of Algorithms*, 20:400–430, March 1996. A preliminary version appeared in Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms.
5. D. R. Karger and C. Stein. “A New Approach to the Minimum Cut Problem”. *Journal of the ACM*, 43(4):601–640, July 1996. Preliminary portions appeared in SODA 1992 and STOC 1993.
6. A. Blum and D. R. Karger. “Improved approximation for graph coloring”. *Information Processing Letters*, 61(1):49–53, January 1997.
7. D. R. Karger and R. Motwani. “Derandomization Through Approximation: An $\mathcal{NC}$ Algorithm for Minimum Cuts”. *SIAM Journal on Computing*, 26(1):255–272, January 1997. A preliminary version appeared in Proceedings of the 25th ACM Symposium on Theory of Computing.
8. D. R. Karger, N. Nisan, and M. Parnas. “Fast Connected Components Algorithms for the EREW PRAM”. *SIAM Journal on Computing*, 28(3):1021–1034, 1999. A preliminary version appeared in Proceedings of the 4th Annual ACM-SIAM Symposium on Parallel Algorithms and Architectures.

9. D. R. Karger, G. D. S. Ramkumar, and R. Motwani. “On Approximating the Longest Path in a Graph”. *Algorithmica*, 18(1):82–98, May 1997. A preliminary version appeared in the 1993 Workshop on Algorithms and Data Structures.
10. P. Fizzano, D. R. Karger, C. Stein, and J. Wein. “Job Scheduling in Rings”. *Journal of Parallel and Distributed Computing*, 45(2):122–133, September 1997. A preliminary version appeared in Proceedings of the 6th Annual ACM-SIAM Symposium on Parallel Algorithms and Architectures.
11. D. R. Karger and D. Koller. “(De)randomized Construction of Small Sample Spaces in $\mathcal{NC}$ ”. *Journal of Computer and System Sciences*, 55(3):402–413, December 1997. Special issue of selected papers from Proceedings of the 35th Annual Symposium on the Foundations of Computer Science.
12. D. R. Karger. “Random Sampling and Greedy Sparsification in Matroid Optimization Problems”. *Mathematical Programming B*, 82(1–2):41–81, June 1998. A preliminary version appeared in Proceedings of the 34th Annual Symposium on the Foundations of Computer Science.
13. D. R. Karger, R. Motwani, and M. Sudan. “Approximate Graph Coloring by Semidefinite Programming”. *Journal of the ACM*, 45(2):246–265, March 1998. A preliminary version appeared in Proceedings of the 35th Annual Symposium on the Foundations of Computer Science.
14. D. R. Karger. “Random Sampling in Graph Optimization Problems: A Survey”. *Optima*, 58:1–11, 1998.
- * 15. D. Karger, A. Sherman, A. Berkheimer, B. Bogstad, R. Dhanidina, K. Iwamoto, B. Kim, L. Matkins, and Y. Yerushalmi. “Web caching with consistent hashing”. *Computer Networks*, 31(11–16):1203–1213, May 1999.
16. D. R. Karger. “A Randomized Fully Polynomial Approximation Scheme for the All Terminal Network Reliability Problem”. *SIAM Journal on Computing*, 29(2):492–514, 1999. A preliminary version appeared in Proceedings of the 27th ACM Symposium on Theory of Computing. A corrected version was published in SIAM Review. Winner, SIAM Outstanding Paper Prize, 2000.
17. D. R. Karger. “A Randomized Fully Polynomial Approximation Scheme for the All Terminal Network Reliability Problem”. *SIAM Review*, 43(3):499–522, 2001. A preliminary version appeared in Proceedings of the 27th ACM Symposium on Theory of Computing. This corrects a version published in SICOMP. Winner, SIAM Outstanding Paper Prize, 2000.
18. S. Arora, D. R. Karger, and M. Karpinski. “Polynomial Time Approximation Schemes for Dense Instances of $\mathcal{NP}$ -Hard Problems”. *Journal of Computer and System Sciences*, 58:193–210, 1999. Special issue of selected papers from Proceedings of the 27th ACM Symposium on Theory of Computing.

19. D. R. Karger. “Random Sampling in Cut, Flow, and Network Design Problems”. *Mathematics of Operations Research*, 24(2):383–413, May 1999. A preliminary version appeared in Proceedings of the 26th ACM Symposium on Theory of Computing.
20. D. R. Karger. “Minimum Cuts in Near-Linear Time”. *Journal of the ACM*, 47(1):46–76, January 2000. A preliminary version appeared in Proceedings of the 28th ACM Symposium on Theory of Computing.
- * 21. A. A. Benczúr and D. R. Karger. “Augmenting Undirected Edge Connectivity in $\tilde{O}(n^2)$ Time”. *Journal of Algorithms*, 37:2–36, 2000. Special issue of selected papers from Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms.

3. Papers to Appear in Refereed Journals

- * 1. I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan. “Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications”. *Transactions on Networking*, 2003. To appear.

4. Proceedings of Refereed Conferences

1. D. R. Karger, D. Koller, and S. J. Phillips. “Finding the Hidden Path: Time Bounds for All-Pairs Shortest Paths”. In *Proceedings of the 32nd Annual Symposium on the Foundations of Computer Science*, pages 560–568. IEEE, IEEE Computer Society Press, October 1991. Journal version appears in *SIAM Journal on Computing* 22(6).
2. D. Cutting, D. R. Karger, J. Pedersen, and J. W. Tukey. “Scatter/Gather: A Cluster-based Approach to Browsing Large Document Collections”. In *Proceedings of the 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 318–329, Copenhagen, Denmark, 1992.
3. D. R. Karger, N. Nisan, and M. Parnas. “Fast Connected Components Algorithms for the EREW PRAM”. In *Proceedings of the 4th Annual ACM-SIAM Symposium on Parallel Algorithms and Architectures*, pages 562–572. ACM, June 1992. Journal version appears in *SIAM Journal on Computing* 28(3).
4. D. R. Karger. “Global Min-cuts in $\mathcal{RNC}$ and Other Ramifications of a Simple Mincut Algorithm”. In *Proceedings of the 4th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 21–30. ACM-SIAM, January 1993. Journal version appears in *Journal of the ACM* 43(4).
5. D. R. Karger, G. D. S. Ramkumar, and R. Motwani. “On Approximating the Longest Path in a Graph”. In F. Dehne, editor, *WADS93: Algorithms and Data Structures : Third Workshop*, number 709 in *Lecture Notes in Computer Science*, pages 421–430. Springer-Verlag, 1993. Journal version appears in *Algorithmica* 18(1).
6. D. R. Karger and C. Stein. “An $\tilde{O}(n^2)$ Algorithm for Minimum Cuts”. In A. Aggarwal, editor, *Proceedings of the 25th ACM Symposium on Theory of Computing*, pages 757–765. ACM, ACM Press, May 1993. Journal version appears in *Journal of the ACM* 43(4).

7. D. R. Karger and R. Motwani. “Derandomization Through Approximation: An $\mathcal{NC}$ Algorithm for Minimum Cuts”. In A. Aggarwal, editor, *Proceedings of the 25th ACM Symposium on Theory of Computing*, pages 497–506. ACM, ACM Press, May 1993. Journal version appears in *SIAM Journal on Computing* 26(1).
8. D. Cutting, D. R. Karger, and J. Pedersen. “Constant Interaction-Time Scatter/Gather Browsing of Very Large Document Collections”. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 126–134, July 1993. Pittsburgh, PA.
9. D. R. Karger. “Random Sampling in Matroids, with Applications to Graph Connectivity and Minimum Spanning Trees”. In L. Guibas, editor, *Proceedings of the 34th Annual Symposium on the Foundations of Computer Science*, pages 84–93. IEEE, IEEE Computer Society Press, November 1993. Journal version appears in *Mathematical Programming B* 82(1–2).
10. D. R. Karger, S. Phillips, and E. Torng. “A Better Algorithm for an Ancient Scheduling Problem”. In D. D. Sleator, editor, *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 132–140. ACM-SIAM, January 1994. Journal version appears in *Journal of Algorithms* 20.
11. D. R. Karger. “Using Randomized Sparsification to Approximate Minimum Cuts”. In D. D. Sleator, editor, *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 424–432. ACM-SIAM, January 1994.
12. D. R. Karger. “Random Sampling in Cut, Flow, and Network Design Problems”. In *Proceedings of the 26th ACM Symposium on Theory of Computing*, pages 648–657. ACM, ACM Press, May 1994. Journal version appears in *Mathematics of Operation Research* 24(2), 1999.
13. P. Fizzano, D. R. Karger, C. Stein, and J. Wein. “Job Scheduling in Rings”. In *Proceedings of the 6th Annual ACM-SIAM Symposium on Parallel Algorithms and Architectures*, pages 210–219. ACM, June 1994. Journal version appears in *Journal of Parallel and Distributed Computing* 45(2).
14. D. R. Karger and D. Koller. “(De)Randomized Construction of Small Sample Spaces in $\mathcal{NC}$ ”. In S. Goldwasser, editor, *Proceedings of the 35th Annual Symposium on the Foundations of Computer Science*, pages 252–263. IEEE, IEEE Computer Society Press, November 1994. Journal version appears in *Journal of Computer and System Sciences* 55.
15. D. R. Karger, R. Motwani, and M. Sudan. “Approximate Graph Coloring by Semidefinite Programming”. In S. Goldwasser, editor, *Proceedings of the 35th Annual Symposium on the Foundations of Computer Science*, pages 2–13. IEEE, IEEE Computer Society Press, November 1994. Journal version appears in *Journal of the ACM* 45(2).
16. D. R. Karger and S. Plotkin. “Adding Multiple Cost Constraints to Combinatorial Optimization Problems, with Applications to Multicommodity Flows”. In *Proceedings*

- of the 27th ACM Symposium on Theory of Computing, pages 18–25. ACM, ACM Press, May 1995.
17. M. A. Hearst, D. R. Karger, and J. O. Pedersen. “Scatter/Gather as a Tool for Navigating Search Results”. In *Proceedings of the AAAI Fall Symposium on Knowledge Navigation*, 1995.
 18. S. Arora, D. R. Karger, and M. Karpinski. “Polynomial Time Approximation Schemes for Dense Instances of $\mathcal{NP}$ -Hard Problems”. In *Proceedings of the 27th ACM Symposium on Theory of Computing*, pages 284–293. ACM, ACM Press, May 1995. Journal version appears in *Journal of Computer and System Sciences*.
 19. D. R. Karger. “A Randomized Fully Polynomial Approximation Scheme for the All Terminal Network Reliability Problem”. In *Proceedings of the 27th ACM Symposium on Theory of Computing*, pages 11–17. ACM, ACM Press, May 1995. Journal version appears in *SIAM Journal on Computing* 29(2).
 - * 20. A. A. Benczúr and D. R. Karger. “Approximate s – t Min-Cuts in $\tilde{O}(n^2)$ Time”. In G. Miller, editor, *Proceedings of the 28th ACM Symposium on Theory of Computing*, pages 47–55. ACM, ACM Press, May 1996.
 21. D. R. Karger. “Minimum Cuts in Near-Linear Time”. In G. Miller, editor, *Proceedings of the 28th ACM Symposium on Theory of Computing*, pages 56–63. ACM, ACM Press, May 1996. Journal version appears in *Journal of the ACM* 47(1).
 - * 22. C. C. Chekuri, A. V. Goldberg, D. R. Karger, M. S. Levine, and C. Stein. “Experimental Study of Minimum Cut Algorithms”. In M. Saks, editor, *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 324–333. ACM-SIAM, January 1997.
 - * 23. D. R. Karger and R. P. Tai. “Implementing a Fully Polynomial Time Approximation Scheme for All Terminal Network Reliability”. In M. Saks, editor, *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 334–343. ACM-SIAM, January 1997.
 - * 24. D. R. Karger, E. Lehman, T. Leighton, M. Levine, D. Lewin, and R. Panigrahy. “Consistent Hashing and Random Trees: Distributed Caching protocols for Relieving Hot Spots on the World Wide Web”. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 654–663. ACM, ACM Press, May 1997.
 25. D. R. Karger. “Using Random Sampling to Find Maximum Flows in Uncapacitated Undirected Graphs”. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 240–249. ACM, ACM Press, May 1997.
 26. C. Young, D. S. Johnson, D. R. Karger, and M. D. Smith. “Near-Optimal Interprocedural Branch Alignment”. In *ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 183–193, Las Vegas, NV, June 1997. ACM.
 - * 27. D. Karger, A. Sherman, A. Berkheimer, B. Bogstad, R. Dhanidina, K. Iwamoto, B. Kim, L. Matkins, and Y. Yerushalmi. “Web Caching with Consistent Hashing”. In *Proceedings of the Eighth World-Wide Web Conference*, 1999.

28. S. Arora, M. Grigni, D. Karger, P. Klein, and A. Wolszryn. “A Polynomial-Time Approximation Scheme for Weighted Planar Graph TSP”. In H. Karloff, editor, *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 33–41. ACM-SIAM, January 1998.
29. D. R. Karger. “Better Random Sampling Algorithms for Flows in Undirected Graphs”. In H. Karloff, editor, *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 490–499. ACM-SIAM, January 1998.
- * 30. A. A. Benczúr and D. R. Karger. “Augmenting Undirected Edge Connectivity in $\tilde{O}(n^2)$ Time”. In H. Karloff, editor, *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 500–509. ACM-SIAM, January 1998. Journal version appears in *Journal of Algorithms* 37.
- * 31. D. R. Karger and M. Levine. “Finding Maximum Flows in Simple Undirected Graphs Seems Faster than Bipartite Matching”. In *Proceedings of the 29th ACM Symposium on Theory of Computing*, pages 69–78, New York, May 23–26 1998. ACM, ACM Press.
- * 32. D. W. Engels, D. R. Karger, S. G. Kolliopoulos, S. Sengupta, R. N. Uma, and J. Wein. “Techniques for Scheduling with Rejection”. In *European Symposium on Algorithms (Lecture Notes in Computer Science)*, volume 1461, page 490, 1998.
33. D. R. Karger, P. N. Klein, C. Stein, M. Thorup, and N. Young. “Optimal Rounding Algorithms for a Geometric Embedding of the Multiway Cut Problem”. In *Proceedings of the 30th ACM Symposium on Theory of Computing*, pages 668–677. ACM, ACM Press, May 1999.
34. F. Afrati, E. Bampis, C. Chekuri, D. Karger, C. Kenyon, S. Khanna, I. Milis, M. Queyranne, M. Skutella, C. Stein, and M. Sviridenko. “Approximation Schemes for Minimizing Average Weighted Completion Time with Release Dates”. In *Proceedings of the 32nd Annual Symposium on the Foundations of Computer Science*, pages 32–43. IEEE, IEEE Computer Society Press, October 1999.
- * 35. R. D. Iyer, D. R. Karger, H. Rahul, and M. Thorup. “An Experimental Study of Polylogarithmic Fully-Dynamic Connectivity Algorithms”. In *Proceedings of ALENEX00: Workshop on Algorithm Engineering and Experimentation*, January 2000. ALENEX00 special issue of the *Journal of Experimental Algorithmics*.
- * 36. E. Adar, D. R. Karger, and L. A. Stein. “Haystack: Per-User Information Environments”. In *Proceedings of the 8th International Conference on Information and Knowledge Management*, pages 413–422, 1999.
37. M. Thorup and D. R. Karger. “Dynamic Graph Algorithms with Applications”. In *Seventh Scandinavian Workshop on Algorithm Theory*, 2000.
- * 38. J. Li, J. Janotti, D. S. J. De Couto, D. R. Karger, and R. Morris. “A Scalable Location Service for Geographic Ad-Hoc Routing”. In *Proceedings of the 6th ACM International Conference on Mobile Computing and Networking (MobiCom 2000)*, pages 120–130, Boston, MA, August 2000.

- * 39. D. R. Karger and M. Minkoff. “Building Routing Trees with Incomplete Global Knowledge”. In *Proceedings of the 33rd Annual Symposium on the Foundations of Computer Science*, pages 613–623. IEEE, IEEE Computer Society Press, November 2000.
- * 40. D. W. Engels, J. Feldman, D. R. Karger, and M. Ruhl. “Scheduling Trees with Communication and Precedence Delays”. In S. R. Kosaraju, editor, *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*. ACM-SIAM, January 2001.
- * 41. D. R. Karger and N. Srebro. “Learning Markov Networks: Maximum Bounded Tree-width Graphs”. In S. R. Kosaraju, editor, *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 392–401. ACM-SIAM, January 2001.
- * 42. F. Dabek, E. Brunskill, M. F. Kaashoek, D. Karger, R. Morris, I. Stoica, and H. Balakrishnan. “Building Peer-to-Peer Systems With Chord, a Distributed Lookup Service”. In *Proceedings of the 8th Workshop on Hot Topics in Operating Systems (HotOS-VIII)*, Schloss Elmau, Germany, May 2001. IEEE Computer Society.
- * 43. I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. “Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications”. In *Proceedings of the ACM SIGCOMM ’01 Conference*, pages 149–160, San Diego, California, August 2001.
- * 44. F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica. “Wide-area cooperative storage with CFS”. In *Proceedings of the ACM 2001 Symposium on Operating System Principles*, October 2001.
- * 45. D. R. Karger and M. Ruhl. “Finding Nearest Neighbors in Growth Restricted Metrics”. In *Proceedings of the 33rd ACM Symposium on Theory of Computing*, pages 741–750. ACM, ACM Press, May 2002.
- * 46. D. R. Karger and M. S. Levine. “Random Sampling from Residual Graphs”. In *Proceedings of the 33rd ACM Symposium on Theory of Computing*, pages 63–66. ACM, ACM Press, May 2002.
- 47. E. W. Boyer, K. Shih, D. R. Karger, L. Quang, and P. Case. “Internet Surveillance of Pro-drug Websites. I. Incidence of Club Drug Reporting Over a One-Year Period.”. *Journal of Toxicology: Clinical Toxicology*, 39:536, 2001. (abstract).
- 48. E. W. Boyer, K. Shih, D. R. Karger, L. Quang, and P. Case. “Internet Surveillance of Pro-drug Websites. II. Identification of Emerging Drug Use Trends.”. *Journal of Toxicology: Clinical Toxicology*, 39:537, 2001. (abstract).
- 49. E. W. Boyer, K. Shih, D. R. Karger, L. Quang, and P. Case. “Internet Surveillance of Pro-drug Websites. III. Identification of Emerging Drugs of Abuse.”. *Journal of Toxicology: Clinical Toxicology*, 39:537, 2001. (abstract).
- * 50. J. Feldman and D. R. Karger. “Decoding Turbo-Like Codes via Linear Programming”. In *Proceedings of the 35th Annual Symposium on the Foundations of Computer Science*. IEEE, IEEE Computer Society Press, November 2002.
- * 51. D. Libben-Nowell, H. Balakrishnan, and D. R. Karger. “Observations on the Dynamic Evolution of Peer to Peer Systems”. In *Proceedings of the First International Workshop on Peer-to-Peer Systems*, Cambridge, MA, March 2002.

- * 52. D. Libben-Nowell, H. Balakrishnan, and D. R. Karger. “Analysis of the Evolution of Peer to Peer Systems”. In *ACM Symposium on Principles of Distributed Computing*, pages 233–242, Monterey, CA, July 2002.
- * 53. N. Feamster, M. Balazinska, G. Harfst, H. Balakrishnan, and D. Karger. “Infrant: Circumventing Web Censorship and Surveillance”. In *Proceedings of the 11th USENIX Security Symposium*, pages 247–262, San Fransisco, CA, August 2002. Best Student Paper Award.
- * 54. D. Huynh, D. Karger, and D. Quan. “Haystack: A Platform for Creating, Organizing, and Visualizing Information Using RDF”. In *Semantic Web Workshop at WWW2002*, Hawaii, May 2002.
- * 55. J. Feldman, D. R. Karger, and M. Wainright. “Linear Programming Based Decoding of Turbo-Like Codes and its Relation to Iterative Approaches”. In *Allerton Conference on Communication, Control, and Computing*, 2002.
- 56. M. F. Kaashoek and D. R. Karger. “Koorde: A Simple Degree-Optimal Distributed Hash Table”. In *2nd International Workshop on Peer to Peer Systems*, LNCS Hot Topics, Berkeley, CA, January 2003. Springer.
- * 57. J. Li, B. T. Loo, J. Hellerstein, F. Kaashoek, D. R. Karger, and R. Morris. “On the Feasibility of Peer-to-Peer Web Indexing and Search”. In *2nd International Workshop on Peer to Peer Systems*, LNCS Hot Topics, Berkeley, CA, January 2003. Springer.
- * 58. J. Feldman, D. Karger, and M. Wainwright. “Using Linear Programming to Decode Linear Codes”. In *37th annual Conference on Information Sciences and Systems (CISS)*, Baltimore, MD, March 2003.
- * 59. D. Quan, D. Huynh, and D. R. Karger. “RDF Authoring Environments for End Users”. In *International Workshop on Semantic Web Foundations and Application Technologies (SWFAT)*, 2003.
- * 60. T. Ho, D. R. Karger, M. Médard, and R. Koetter. “Network Coding from a Network Flow Perspective”. In *IEEE International Symposium on Information Theory*, Yokohama, Japan, June 2003. IEEE.
- * 61. T. Ho, R. Koetter, M. Médard, D. R. Karger, and M. Efron. “The Benefits of Network Coding over Routing in a Randomized Setting”. In *IEEE International Symposium on Information Theory*, Yokohama, Japan, June 2003. IEEE.
- * 62. J. Teevan and D. R. Karger. “Empirical Development of an Exponential Probabilistic Model for Text Retrieval: Using Textual Analysis to Build a Better Model”. In *26th International ACM SIGIR Conference*, Toronto, July 2003. ACM SIGIR, ACM.
- * 63. J. D. M. Rennie, L. Shih, J. Teevan, and D. R. Karger. “Tackling the Poor Assumptions of Naive Bayes Text Classifiers”. In *The Twentieth International Conference on Machine Learning*, Washington, DC, August 2003. To appear.
- * 64. D. Quan, K. Bakshi, D. Huynh, and D. R. Karger. “User Interfaces for Supporting Multiple Categorization.”. In *INTERACT: 9th IFIP International Conference on Human*

Computer Interaction, Zurich, September 2003. International Federation for Information Processing.

- * 65. A. Blum, S. Chawla, D. R. Karger, T. Lane, A. Meyerson, and M. Minkoff. “Approximation Algorithms for Orienteering and Discounted-Reward TSP”. In *Proceedings of the 36th Annual Symposium on the Foundations of Computer Science*. IEEE, IEEE Computer Society Press, October 2003. To appear.
- 66. D. Quan, D. Huynh, D. R. Karger, and R. Miller. “User Interface Continuations”. In *The 16th Annual Symposium on User Interface Software and Technology*, Vancouver, BC, November 2003. ACM. To appear.
- * 67. D. Quan, D. Huynh, and D. R. Karger. “Haystack: A Platform for Authoring End User Semantic Web Applications”. In *2nd International Semantic Web Conference*, Sanibel Island, FL, October 2003.

5. Other Major Publications
6. Internal Memoranda and Progress Reports (items not listed above)

7. Invited Lectures

1. “Approximate Graph Coloring by Semidefinite Programming,” 15th International Symposium on Mathematical Programming, Ann Arbor, Michigan, 1994.
2. “Polynomial Time Approximation Schemes for Dense Instances of $\mathcal{NP}$ -hard Problems” and “A Fully Polynomial Approximation Scheme for the All-Terminal Network Reliability Problem,” The Hebrew University Computer Science Theory Seminar, June 26, 1995. Jerusalem, Israel.
3. “Random Sampling in Graph Optimization Problems,”
MIT Operations Research Center and Decision Sciences Center Seminar Series, February 29, 1996.
12th Maryland Theory Day, March 22 1996.
IBM Almaden Research Labs, August 1996.
Stanford University, August 1996.
Carnegie-Mellon University, March 13, 1996.
4. “A Fully Polynomial Time Approximation Scheme for the All Terminal Network Reliability Problem,” Carnegie-Mellon University, March 15, 1996, Pittsburgh, PA. INFORMS, November 1996, Atlanta, GA.
5. “Consistent Hashing and Random Trees: Tools for Relieving Hot Spots on the World Wide Web.” Princeton Computer Science Colloquium, Princeton University, Princeton, NJ. December 17, 1998.
6. “Random Sampling Algorithms for Computing Cuts and Flows in Graphs: A Survey.” Princeton theory lunch, Princeton University, Princeton, NJ. December 17, 1998.
7. “Information Retrieval: Challenges in Interactive-time Manipulation of Massive Text Collections.”
DIMACS special session on massive data sets at the meeting of the American Mathematical Society, January 8, 1997. San Diego, CA.
Annual meeting of the American Association for the Advancement of Science, January 2000.
8. “Randomized Algorithms for Graph Optimization Problems,” invited minicourse at the 2000 meeting of the Dutch Operations Research Society, Lunteren, Netherlands.
9. Panel presentation, NSF workshop on Challenges for Theoretical Compute Science at 2000 Symposium on Theory of Computing.
10. “Peer to Peer Routing Algorithms.” Hebrew University, Jerusalem, Israel. January 2002.
11. “Text Search in Peer to Peer Systems”, First international symposium on peer to peer systems, Cambridge, MA, March 2002.

12. “Randomized Algorithms for Graph Optimization Problems,” invited minicourse at the 2002 ISMP Conference on Integer Programming and Combinatorial Optimization, Cambridge, MA May 2002.
13. “Algorithmic Tools Applied to Problems in Machine Learning and Inference,” invited tutorial at 2003 NIPS Neural Information Processing Symposium, Dec 2003.
14. “Haystack: Putting Users Back in Control of their Information Organizations,” invited presentation to the ACM Greater Boston Chapter (GBC/ACM), May 2005.

Theses Supervised by David Ron Karger

Summary

	<u>Total</u>	<u>Completed</u>	<u>In Progress</u>
S.B.	2	2	0
S.M.	12	10	2
M.Eng.	18	16	2
<u>Doctoral</u>			
As Supervisor	8	4	4
As Reader	16	15	1

S.B. Theses

Martha Greenberg, *Dynamic Hierarchical Caching for the World Wide Web*. Jan, 1997.

Luke Matkins, *Re-architecting the Haystack Information Retrieval System*. Jun, 2000.

S.M. Theses

Matt Levine, *An Experimental Evaluation of Minimum Cut Algorithms*. June, 1997.

Rina Panigrahy, *Relieving Hot Spots on the World Wide Web*. June, 1997

Daniel Lewin, *Consistent Hashing and Random Trees*. June, 1998

Raj Iyer, *Boosting Algorithms for Machine Learning*. June, 1999.

Maria Minkoff, *The Prize Collecting Steiner Tree Problem*. June, 2000.

Jon Feldman, *Steiner Network Problems with a Fixed Number of Terminals*. June, 2000.

Nati Srebro, *Markov Random Fields and Maximum Weight Hypertrees*. September 2000.

April Rasala, *Graph Coloring for Optical Networks*. June 2001.

Jaime Teevan, *Probabilistic Models in Information Retrieval*. June 2002.

David Huynh, *A Semantic User Interface for Haystack*. June 2003.

Vineet Sinha, *Navigating Semistructured Metadata*. Expected August 2003.

Karun Bakshi, *A Semantically Active Messaging Framework*. Expected January 2004.

M.Eng. Theses

Ray P. Tai. *Implementing an Approximation Scheme for All Terminal Network Reliability*.
June, 1996.

Jack Fu. *The Generalized Document Summarizer*. June, 1997.

Mehul Shah. *ReferralNet: A Resource Location System Guided by Personal Relations*. June, 1997.

Eytan Adar. *Integrating a Database System into the Haystack Information Retrieval Engine*. June, 1998.

Mark Asdoorian *Data Manipulation Services in the Haystack IR System*. June, 1998.

Ilya Lisansky, *Data Model Management in the Haystack Information Retrieval System*. June 1999.

Aidan Low, *A User Interface for the Haystack Information Retrieval System*. June 1999.

Amit Khetan. *Code Layout and Quadratic Assignment Problems*. June 1999.

Dennis Ruhl. *Implementing Maximum Flow Algorithms for Unit Capacity Graphs*. June 1999.

Wendy Chien, *Relevance Feedback for the Haystack Information Retrieval System*. June 2000.

Svetlana Shnitser, *Integrating Relational Database Operations in the Haystack Information Retrieval System*. June 2000.

Adam Holt. *A Document Scanning System for the Haystack Information Retrieval System*. June 2000.

David Zych, *High Level Architecture of the Haystack Information Retrieval System*. June 2001.

Kenneth McCracken, *Robust Transactional Storage for the Haystack Information Retrieval System*. June 2001.

Dominic Daleo, *Versioning in the Haystack Retrieval System*, 2002.

Mark Rosen. *A Machine Learning Email Classifier*. February 2003.

Jon Derryberry. Expected August 2003.

Andrew Hogue. Expected January 2004.

Doctoral Theses, Supervisor

Matt Levine, *Theory and Implementation of Flow Algorithms*. Jul 2002.

Jonathan Feldman. *Decoding Error-Correcting Codes via Linear Programming*, June 2003.

Dennis Quan. *Designing End-User Information Environments Based on Semistructured Data Models*, June 2003.

Maria Minkoff. *Approximation Algorithms for Combinatorial Optimization Under Uncertainty*. August 2003.

Matthias Ruhl. *Efficient Algorithms for New Computation Models*. August 2003.

Lawrence Shih. *Machine Learning for Text Recommender Systems*. November 2003.

Nicole Immerlica. *Computing with Strategic Agents*. May 2005.

Jaime Teevan. Proposal in progress.

Doctoral Theses, Reader

Jon Kleinberg. *Approximation Algorithms for Disjoint Path Problems*, June 1996

David Wilson, *Exact Sampling with Markov Chains*. June 1996 (Math)

Igal Galperin, *On Consulting a Set of Experts and Searching*, August 1996

Andras Benczúr. *Cut Structure and Randomized Algorithms in Edge Connectivity Problems*, March 1997 (Math)

Matthew Andrews, *Approximation Algorithms for Scheduling and Load Balancing Problems*. June, 1997 (Math)

Parry Husbands, *The Parallel Problem Server*, June 1999.

Bien Velez-Rivera, *Computing Interactive Query Hierarchies Using Query Lookahead*. June 1999.

Matt Welborn, *Flexible Signal Processing Algorithms for Wireless Communications*. June 2000.

Daniel Engels, *Scheduling for Hardware-Software Partitioning in Embedded Hardware Design*. June, 2000.

Kenney Ng, *Subword-based Approaches for Spoken Document Retrieval*. June, 2000.

Tracey Ho, *Networking from a Network Coding Perspective*. June, 2004

Ching Law, *Distributed Algorithms for Dynamic Topology Construction and Their Applications*. September 2004.

David Liben-Nowell, *Algorithms for Small Worlds Networks*. May 2005.

Vahab Mirrokni. *Algorithms for Strategic Agents*. May 2005.

Dwaine Clarke. May 2005.

Introduction.

My interests cover two broad areas—algorithms and information retrieval.

My work in algorithms focuses on improving algorithms for combinatorial optimization problems—particularly those involving graphs (networks), such as maximum flows, minimum cuts, minimum spanning trees, graph coloring and partitioning. While I use all available tools to attack these problems, the one I have found most effective is *randomization*. Letting algorithms make random choices as they execute often leads to solutions that are faster and simpler than those limited to a single deterministic behavior. My work has led to substantial improvements in the algorithms for some of the most fundamental graph optimization problems. I believe that a great deal remains to be done in this area and continue to work actively in it.

I also explore opportunities to apply our rich store of algorithmic knowledge to problems outside the theoretical community. Theoreticians have often been accused of navel-gazing, defining problems of interest only to the theory community. I hope to fight this image. My recent work includes applications to caching algorithms for the world wide web, block layout and instruction scheduling for compilers, routing in mobile networks, signal processing, and machine learning. In all cases, I took problems defined by non-theoreticians and found ways to apply theoretical techniques to solve them. I have pursued the web work further by consulting at Akamai technologies, which aims to put web caching algorithms into practice. Conversely, I have been exploring experimental approaches to theoretical work, in which large-scale computation is used to explore the space of potential solutions to an algorithmic problem, thus leading to new theory. In my teaching as well (as discussed below) I have tried to reach out to non-theoreticians in my graduate theory classes.

While my background is mainly in theory, I have maintained a side interest in information retrieval that has been growing stronger in recent years. My interest focuses on the problem of extracting useful information from large text collections. I have been particularly interested in how IR systems can adapt to the needs, background, and habits of their individual users. Most past work on IR has had the library as a model, positing a massive repository which is visited by anonymous users when they have a specific information need. I aim to study bookshelves—repositories that exist per-person, and are constantly involved in the user's interaction with information. I am also exploring ways to let individuals seek out useful information in others' personal collections, just as people now turn to their colleagues when looking for information they do not have. This work requires me to combine concepts from three distinct problem domains. Thinking about flexible data representations (different people structure their information differently) requires an understanding of Databases and Knowledge Representation. Thinking about user interfaces (how to watch a user and find out what they want, and how to present information in a way that works best for that user) requires insight from the field of Human-Computer Interaction. Finally, understanding how to adapt, based on gathered data, to the specific preferences of the user requires methods from the field of Machine Learning.

Algorithmic Theory

My work in pure algorithms addresses the area of combinatorial optimization. Optimization, studied in various forms in applied mathematics, operations research, and computer science, involves the general study of finding good solutions to problems involving certain constraints and objective functions. A classic example would be finding the minimum of some complex multivariate function on the positive real numbers, using techniques such as gradient descent. Combinatorial optimization focuses instead on problems where the constraints and objective have interesting discrete combinatorial structure. For example, we might seek the shortest path between two points in a network—the problem is posed over a network, a discrete structure, rather than the real numbers, and possible solutions are sets of links that form a path. Combinatorial structure can often be exploited to get solution algorithms that are more efficient (in terms of time to solve or memory used) or higher quality (in value of solution) than those based on general purpose mathematical programming.

A particular combinatorial object on which I focus is *graphs*. The graph is one of the most common structures in computer science, modeling among other things roads, communication and transportation networks, electrical circuits, relationships between individuals, corporate hierarchies, hypertext collections, tournaments, resource allocations, project plans, database and program dependencies, and parallel architectures. The shortest path problem studied above has natural applications in many of these domains. Other graph optimization problems I have studied include maximum flow (the problem of finding the maximum amount of material that can be shipped, subject to capacity constraints, from one point to another in a network), minimum cut (finding the cheapest way to “disconnect” two elements in a network by removing edges), graph coloring (the problem of partitioning elements into a small number of groups such that no “conflicting” elements are in the same group), and scheduling (arranging a set of jobs on machines with different capabilities so that they complete as quickly as possible).

My major contribution in the area of combinatorial optimization has been to demonstrate novel applications of *randomization*, and in particular *random sampling*, as a solution technique for numerous optimization problems. The representative random sample is a central concept of statistics. It is often possible to gather a great deal of information about a large population by examining a small sample randomly drawn from it. This approach has obvious advantages in reducing the investigator’s work, both in gathering and in analyzing the data.

This idea has applications in optimization. Given an optimization problem, it may be possible to generate a small representative subproblem by random sampling (perhaps the most natural sample from a graph is a random subset of its edges). Intuitively, such a subproblem should form a microcosm of the larger problem. One can examine the subproblem and use it to glean information about solutions to the original problem. Since the subproblem is small, we can spend proportionally more time examining it than we would spend examining the original problem.

Using these ideas, I have given dramatically improved algorithms for many of the core problems in combinatorial optimization, including maximum flow, minimum cut, graph coloring, and minimum spanning trees. These problems have been studied for decades, and have recently been characterized by small, incremental improvements. My work yielded completely different ways of tackling these problems. The new techniques yielded improved running times that were decidedly

not incremental.

Applications of Algorithms

While continuing my work in pure theory, I have used my time at MIT to explore applications of algorithmic thinking to problems in many other domains of computer science. This has given me an opportunity to collaborate with many of my colleagues and evangelize the value of algorithmic techniques.

Perhaps the single largest application area I have explored has been in computer networks. My interest in this area began during collaborations with Professor Tom Leighton, as we studied the problem of coping with “flash crowds” on the world-wide web, when huge numbers of people all try to access the same web site, causing overload and failure. The load balancing techniques we developed in tackling this problem, based on randomization, were patented and some time later, used as core technology in the founding of Akamai Technologies, in which I participated. Several years later, closely related ideas played a central role in research on peer to peer (P2P) systems. P2P systems attempt to aggregate and share the resources of large numbers of uncoordinated machines. For example, the millions of gigabytes of empty space on disks in individual’s desktops could be very valuable for backing up critical information in many copies over geographically distributed areas, providing substantial redundancy. To perform such aggregation, it is necessary to decide which data should be placed on which machines. A centralized directory can do so, but creating a centralized directory loses the benefits of redundancy and geographic distribution that are offered by a P2P system. Instead, in collaboration with Professors Kaashoek, Morris, and Balakrishnan among others, we developed techniques to let the many nodes efficiently cooperate in deciding where data should be placed, even as individual nodes connect to and disconnect from the system. We continue to pursue a substantial research program in this area, involving tens of faculty and graduate students at several institutions.

A second area in which I have found interesting applications of theory is artificial intelligence. Modern artificial intelligence makes substantial use of probabilistic models, using them to incorporate observations and make predictions about future events. The structure of these probabilistic models is important—when their structure is “nice” it is possible to make predictions efficiently. We were able to cast the problem of finding nice models into the language of combinatorial optimization and derive new techniques. A paper based on this work won the best student paper award at AAAI, the primary artificial intelligence conference. More recently, we have applied combinatorial optimization techniques to Markov decision processes, another core model within artificial intelligence. And with Josh Tenenbaum (professor in BCS), I’ve demonstrated applications of data structures (originally developed in our peer to peer networking framework) to support fast data classification.

Yet another domain to which we’ve applied algorithmic techniques is coding theory. Coding theory is concerned with intelligently replicating or otherwise encoding information so that it can be reliably transmitted even in the presence of noise in the communication channel. While coding theory is as deeply mathematical as algorithms, it uses a completely different set of techniques; we have shown that it is quite fruitful to apply ideas from combinatorial optimization to coding. With one recently graduated student I demonstrated applications of linear programming, a core

technique in combinatorial optimization, to the decoding of turbo codes very powerful and widely used family of codes. In other work, with Muriel Medard, I've been studying codes specifically developed to transmit information over networks to multiple receivers, and shown connections to ideas from maximum flows and randomized maximum matchings. Lastly, with Hari Balakrishnan, I've applied codes developed in the theoretical computer science community to the problem of helping citizens of unfree regimes defeat, in an undetectable way, the internet censorship they often face.

Information Retrieval.

A chronic present-day mystery is how to deal with the vast amounts of information available at our digital fingertips. We have reached a situation in which the information we need almost always exists, but remains unavailable to us because we cannot find it. This problem has driven the field of information retrieval for decades.

Current information retrieval systems have a "one size fits all" flavor. A web search engine such as Google gives the same search results to a query whether the seeker is a 5 years old novice or a 50 year old professional.

I postulate that much better performance can be achieved by a system that takes full account of the varying backgrounds, skills, habits and preferences of individual users. But to accomplish this, a number of problems must be surmounted.

1. Different users understand and manipulate information in different ways. In order to adapt properly, a personalized IR system needs an information model and user interface sufficiently rich to represent whatever the user considers important.
2. In order to deliver individualized performance, we need to learn about the user. But users are always unwilling (due to lack of time) and often unable (due to lack of expertise or understanding) to describe their backgrounds, skills, or interests. Thus, an adaptive information retrieval system must gather information by passively observing the user's interactions with information.
3. Given these observations, an adaptive system must develop a model of the user that captures their particular interests in and ways of dealing with information. And based on this model, the system must properly skew itself to preferentially seek out information that is useful to that user, and to present it to that user in the way that makes the most sense to them.

To tackle these problems we are building the Haystack system. The Haystack Project aims to create a community of "haystacks": desktop-integrated personal information repositories which archive not only base content but also user-specific meta-information, enabling them to adapt to the particular needs of their users. Haystack consists of three major elements, paralleling the problems outlined above:

Data repository. Haystack stores information using RDF, a general purpose semistructured data model developed by the world wide web consortium. Thanks to the model's flexibility, Haystack can store any type of information that its user considers important, from the authors and titles of a document, to the color of its binding, to the food someone was eating

when they last read it. The machine readability of the RDF representation makes it possible to build autonomous agents to help manage the users repository. We have begun exploring this aspect of the system in collaboration with the web consortium, under the auspices of DARPA's semantic web project, and have constructed a prototype system.

User Interface. The semistructured data model is almost too powerful, in that it imposes no real restriction on what kind of information might be found in a user's repository. This makes it quite challenging to present the information that is there to the user in a meaningful fashion. We propose to develop a user interface to this information that is as customizable as the information itself. Just as a user can choose the kind of information they store, they will be able to choose the way that information is presented to them. As these users are non-experts, they cannot be expected to "program" a user interface. Instead, we must let them customize their views "by example", using manipulations (such as dragging and dropping interface elements) with which they are already familiar. Since information is "typed" (e.g., there are documents, people, and tasks), the system can generalize a user's manipulation of one piece of information and apply it to other information of the same type.

Machine Learning. Haystack aims to use observations of past information access to improve future information access. For example, by seeing what news items a person spends time reading, Haystack will begin to predict what kinds of news interests that user. Haystack can use this prediction both to filter incoming news traffic to pass only information of interest, and to proactively seek out and present to the user news items that would otherwise have gone unnoticed. Similarly, after watching a user perform a web search, and seeing which proposed search results were ultimately used, Haystack will be able to post-process future search results so as to bring to the fore those results that are preferred by its particular user. In addition to the traditional approach of learning based on the textual content of documents, Haystack will make use of its rich semistructured data model to learn based on other important attributes such as the other of the document, its age, its relation to other documents, and whatever other attributes have been recorded by the user. Ultimately, simply by observing a users activities (what files they are viewing at a given time, what applications they are running, who they are communicating with), Haystack may be able to place at their fingertips all the other information necessary for them to complete their current task.

Teaching.

My undergraduate teaching experience has comprised terms as a recitation leader for 6.001 and 6.033 and three terms lecturing 6.042. 6.001 and 6.033 are relatively stable and served mainly as learning experiences for me; in 6.042 I helped Tom Leighton and Nancy Lynch overhaul the course, modifying the both content and the way it is being taught. In the graduate area, I have worked on two courses. 6.856 (Randomized Algorithms) was developed from scratch, while 6.854 (Advanced Algorithms) was substantially modified.

I would like to round out my "teaching education" by teaching recitations in 6.034 at some point. However, I will focus my energies on the theory offerings. My major goal is to convey the

ideas of these courses—mathematical modeling of problems and rigorous analysis of solutions—to the broad student population. I want this approach to be a part of every student's toolkit, rather than an esoteric methodology that he sees briefly and then forgets. The same holds at the graduate level. Algorithms is broadly useful, so I try to teach theory courses that are accessible (though challenging) to all of our first year graduate students in computer science. In both advanced courses, I have produced a curriculum which is challenging, but is specifically intended to target non-theoreticians—they leave with a tool-box that will help them devise algorithmic solutions to the problems they encounter in practice. The algorithm implementation projects which make up the final project in my course offer frequent demonstrations of the benefits of this kind of crossover, as students use what they've learned in the class to tackle problems in a broad variety of fields.

In the long term I would like to carry this idea further, developing a course surveys the broad range of theoretical computer science activity and simultaneously addresses its applications in practice. I also hope to develop a course on information retrieval.

1. Jon Feldman and Matthias Ruhl. "The Directed Steiner Network Problem is Tractable for a Constant Number of Terminals." IEEE Symposium on Foundations of Computer Science, 1999.