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.
Rockstar Games, Inc.

Petitioners

v.

Acceleration Bay, LLC
Alleged Patent Owner

DECLARATION OF SCOTT BENNETT, Ph.D.

9 March 2016

EXHIBIT

Ex. 1004

1. I, Scott Bennett, hereby declare as follows:

2. I am a retired academic librarian working as a Managing Partner of the firm Prior Art Documentation LLC at 711 South Race Street, Urbana, IL, 61801-4132. I was previously employed as follows:

University Librarian, **Yale University**, New Haven, CT., 1994-2001

Director, The Milton S. Eisenhower Library, **The Johns Hopkins University**, Baltimore, MD, 1989-1994

Assistant University Librarian for Collection Management, **Northwestern University**, Evanston, IL, 1981-1989

Instructor, Assistant, and Associate Professor of Library Administration**,** **University of Illinois at Urbana-Champaign**, Urbana, IL, 1974-1981

Assistant Professor of English, **University of Illinois at Urbana-Champaign**, 1967-1974

3. Over the course of my work as a librarian, professor of English, researcher, and author of nearly fifty scholarly papers and other publications, I have had extensive experience with cataloging records and online library management systems built around Machine-Readable Cataloging (MARC) standards. I also have substantial experience in authenticating printed documents

and establishing the date when they were accessible to ordinarily skilled researchers.

4. Exhibit A is my full resume. Further information about my firm is available at www.priorartdocumentation.com.

5. I have been retained by Winston & Strawn LLP to authenticate and establish the dates of public accessibility of certain documents in *inter partes* review proceedings for U.S. Patent Nos.6,829,634, 6,732,147, 6,910,069, and 6,920,497. For this service, I am being paid my usual hourly fee of $85/hour. My compensation in no way depends on the substance of my testimony or the outcome of this proceeding.

PRELIMINARIES

6. *Conference papers*. Conference papers enter the realm of public discourse at the time they are presented at the conference. The circumstances of such presentation vary substantially, especially as regards the prominence of the conference, the number of conference participants, and the organization of the conference. Formal publication of papers presented at conferences also varies widely. Sometimes, the papers are published before the conference so as to be available to conference participants. Sometimes, papers are published only after the conference, and in some cases only abstracts of the papers are published.

7. Because of this variability, detailed information about the conference may often be required to establish the public accessibility of conference papers. The availability of such information, especially for long past conferences, varies substantially.

8. *Library catalog records*. Libraries world-wide use the MARC format for catalog records; this machine readable format was developed at the Library of Congress in the 1960s. The MARC 040a Field identifies the library or other entity that created the original catalog record for a given document and transcribed it into machine readable form. The MARC 008 Field identifies the date when this first catalog record was entered on the file. This date persists in all subsequent uses of the first catalog record, although newly created records for the same document will show a new date.

9. WorldCat is the world's largest public online catalog, maintained by the Online Computer Library Center, Inc., or OCLC (see https://en.wikipedia.org/wiki/OCLC). Its records appear in many different catalogs, including the Statewide Illinois Library Catalog. WorldCat records use the MARC format, and the date a given catalog record was created (corresponding to the MARC 008 Field) appears in some detailed WorldCat records as the Date of Entry.

10. When a book has been cataloged, it will normally be made available to readers soon thereafter—normally within a few days or (at most) within a few weeks of cataloging.

11. The public availability of MARC-formatted catalog records and detailed WorldCat records showing the Date of Entry varies greatly.

12. *Publications in series*. A library typically creates a MARC catalog record for a series of publications, such as the proceedings of an annual conference, when the library receives its first issue. When the institution receives subsequent issues/volumes of the series, the issues/volumes are checked in (sometimes using a date stamp), added to the institution's holdings records, and made available very soon thereafter—normally within a few days of receipt or (at most) within a few weeks of receipt.

13. The initial serials record will often not reflect all of the subsequent changes in publication details (including minor variations in title, etc.).

14. When a library does not intend systematically to acquire a series, but adds individual volumes of a series to its collections, the library will typically treat each such volume as an individual book, or monograph. In this case, the 080 MARC field will record the date when the record for that individual volume, not the series, was created.

15. It is therefore sometimes possible to find both a series and a monograph library catalog record for the same publication.

16. *Periodical publications*. A library typically creates a MARC catalog record for a periodical publication when the library receives its first issue. When the institution receives subsequent issues/volumes of the periodical, the issues/volumes are checked in (often using a date stamp), added to the institution's holdings records, and made available very soon thereafter—normally within a few days of receipt or (at most) within a few weeks of receipt.

17. *Internet Archive.* The Internet Archive is a non-profit digital library founded in 1996 with the mission of providing universal access to all knowledge (see https://en.wikipedia.org/wiki/Internet_Archive).

18. The Internet Archive maintains an archive of webpages collected from the Internet using software called a crawler. Crawlers automatically create a snapshot of webpages as they existed at a certain point in time. The WayBack Machine is an application created by the Internet Archive to search its archive of Web pages and to represent, graphically, the date of each crawler capture.

19. The Internet Archive, now with about 50 petabytes of data, collects only Web material that is publicly available. This means that some sites are "not archived because they were password protected, blocked by robots.txt, or otherwise inaccessible to our automated systems. Site owners might have also

requested that their sites be excluded from the WayBack Machine" (see the WayBack Machine FAQ, https://archive.org/about/faqs.php#The_Wayback_Machine).

20. *Indexing*. An ordinarily skilled researcher will typically begin his or her search for relevant information in an index of periodical and other publications. Having found relevant material, the researcher will then obtain it online, look for it in libraries, or purchase it from the publisher, a bookstore, or other provider. Often, the date of a document's public accessibility will involve both indexing and library date information. Date information for indexing entries is, however, often unavailable. This is especially true for online indexes.

21. Indexing services, including Google Scholar, commonly also provide lists of publications that cite a given document. A citation of a document is evidence that that document was publicly available and in use by researchers as of the publication date of the citing document.

REVIEW OF INDIVIDUAL DOCUMENTS

22. **Document 1**. Shoubridge, Peter, and Dadej, Arek. "Hybrid Routing in Dynamic Networks" in *1997 IEEE International Conference on Communications: Toward the Knowledge Millennium. ICC '97, 8-12 June 1997 Montréal, Québec, Canada*. Vol. 3 (Piscataway, NJ: Institute of Electrical and Electronics Engineers, 1997), 1381-1386.

Authentication

23. Document 1 is a research paper given at an IEEE sponsored conference in Montréal in 1997 and published in the conference proceedings. Attachment 1a is a copy of Document 1 from the Northwestern University Library. Attachment 1b is a copy of that library's catalog record for the proceedings of the *1997 IEEE International Conference on Communications*. Document 1 is also available online. Attachment 1c is a copy of Document 1from the IEEE Xplore Digital Library, a common source of engineering literature.

Public accessibility

24. Document 1 entered the public realm of discourse when it was presented at the 1997 IEEE International Conference on Communications, held on 8-12 June 1997 in Montréal.

25. Attachment 1d is a copy of the Statewide Illinois Library Catalog monograph record for the proceedings of *1997 IEEE International Conference on Communications*. It shows the proceedings are held by 70 libraries world-wide. It shows a date of Entry of 24 September 1997.

26. Attachment 1e is a copy of the Statewide Illinois Library Catalog series record for the proceedings of *1997 IEEE International Conference on Communications*. It shows the proceedings are held by 65 libraries world-wide.

27. Attachment 1f is a copy of the IEEE Xplore Digital Library index record for Document 1. It has a digital object identified (DOI) suggesting the record was created in 1997.

28. Based on the evidence presented here—conference presentation and publication, library cataloging, and indexing—it is my opinion that Document 1 was accessible to an ordinarily skilled researcher at least by early October 1997.

29. **Document 2**. Shoubridge, Peter John. Adaptive Strategies for Routing in Dynamic Networks. Ph.D. Thesis, University of South Aukstralia, 1996.

Authentication

30. Document 2 is a Ph.D. thesis completed in 1996 at the University of South Australia and held there as archival material. Attachment 2a is a copy of the Statewide Illinois Library Catalog record for this thesis.

Public accessibility

31. Attachment 2a, a catalog record, shows that Document 2 is held at 1 library world-wide, as one would expect with a thesis. This catalog record has a date of Entry of 26 May 1998. The record has two subject headings—Telecommunication-Switching systems and Computer algorithms—that a person with skill in the art would use to find relevant material.

32. Based on the evidence presented here—a thesis completed in 1996 and cataloged in 1998—it is my opinion that Document 2 was accessible to an ordinarily skilled research at least by June 1998.

33. **Document 3**. Obraczka, Katia, et al. "A Tool of Massively Replicating Internet Archives: Design, Implementation, and Experience," in *Proceedings of the 16th International Conference on Distributed Computing Systems, 27-30 May 1996, Hong Kong* (New York, NY: IEEE, 1996), 657-664.

Authentication

34. Document 3 is a research paper given at an IEEE sponsored conference in 1996 and published in the conference proceedings. Attachment 3a is a copy of Document 3, along with the *Proceedings*' cover, title page, table of contents, and other preliminary matter, from the University of Illinois at Urbana-Champaign Library. Attachment 3b is a copy of that library's series catalog record for the *Proceedings of the 16th International Conference on Distributed Computing Systems,* showing the holding for the 1996 volume. Document 3 is also available online. Attachment 3c is a copy of Document from the IEEE Xplore Digital Library, a common source of engineering literature.

Public accessibility

35. Document 3 entered the public realm of discourse when it was presented at the 16th International Conference on Distributed Computing Systems, held on 27-30 May 1996 in Hong Kong

36. Attachment 3d is a copy of the Statewide Illinois Library Catalog monograph record for the proceedings of *1997 IEEE International Conference on Communications*. It shows the proceedings, considered as a monograph, are held by 61 libraries world-wide. It shows a date of Entry of 29 July 1996.

37. Attachment 3e is a copy of the Statewide Illinois Library Catalog series record for the proceedings of *1997 IEEE International Conference on Communications*. It shows the proceedings, considered as a series, are held by 133 libraries world-wide.

38. Attachment 3f is a copy of the IEEE Xplore Digital Library index record for Document 3. It has a digital object identified (DOI) suggesting the record was created in 1996.

39. Attachment 3g is a list of 8 publications, identified by Google Scholar, that cite Document 3. These documents were published between 1996 and 1999.

40. Based on the evidence presented here—conference presentation and publication, library cataloging, indexing, and citation—it is my opinion

that Document 3 was accessible to an ordinarily skilled researcher by mid-August 1996 and in actual use by researchers by 1996.

41. **Document 4**. Obraczka, Katia. Massively Replicating Services in Wide-Area Internetworks. Ph.D. Thesis, University of Southern California, 1994.

Authentication

42. Document 4 is a Ph.D. thesis completed in 1994 at the University of Southern California and held there as archival material. Attachment 4a is a copy of the Statewide Illinois Library Catalog record for this thesis. Document 4 is also available online. Attachment 4b is a copy of Document 4, available from ProQuest, a major full-text service. Document 4c is the ProQuest index record for Document 4.

Public accessibility

43. Attachment 4a, a catalog record, shows that Document 4 is held at 1 library world-wide, as one would expect with a thesis. This catalog record has a date of Entry of 4 June 1998.

44. Attachment 4b shows, on the cover page of the thesis, a December 1994 date for the thesis and a copyright date of 1995.

45. Attachment 4d is a list of 20 publications citing Document 4 identified by Google Scholar. These publications appeared between 1995 and 1999. Attachment 4e is a copy of the University of Twente Publications index record for one of these publications: Maria Eva M. Ljiding et al., "Object Distribution Networks for World-Wide Document Circulation," in the proceedings of the *3rd CYTED-RITOS International Workshop in Groupware. CRIWG 1997. October 1-3 1997, Madrid, Spain.*

46. Based on the evidence presented here—a thesis completed in 1994 and cataloged in 1998 and citations beginning in 1995—it is my opinion that Document 4 was accessible to and in actual use by ordinarily skilled researchers by 1995.

47. **Document 5**. Rufino, José, and Verissimo, Paulo. "A Study on the Inaccessibility Characteristics of ISO 8802/4 Token-Bus LANs," in *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 (Piscataway, NJ: IEEE Service Center, 1992), 0958-0967.

Authentication

48. Document 5 is a research paper given at an IEEE sponsored conference in 1992 and published in the conference proceedings. Attachment 5a is a copy of Document 3, along with the proceedings' title page, table of contents, and other preliminary matter, from the University of Illinois at Urbana-Champaign Library. Attachment 5b is a copy of that library's series catalog record for the

IEEE INFOCOM proceedings, showing the holdings for Volume 2. Document 5 is also available online. Attachment 5c is a copy of Document from the IEEE Xplore Digital Library, a common source of engineering literature.

Public accessibility

49. Document 5 entered the public realm of discourse when it was presented at IEEE INFOCOM '92: The Conference on Computer Communications held on 4-8 May 1992 in Florence, Italy. The Message from the Technical Chairs in the published proceedings (Attachment 5a) affirms that "since its establishment eleven years ago, the INFOCOM conference has served as a premier form for the exchange of technical ideas and research results in the area of computer communications." Almost 400 papers were submitted for peer review, and 176 were accepted for presentation at the conference in 44 sessions.

50. Attachment 5d is a copy of the Statewide Illinois Library Catalog record for *IEEE INFOCOM '92: The Conference on Computer Communications.* It shows the proceedings are held by 223 libraries world-wide.

51. Attachment 5e is a copy of the IEEE Xplore Digital Library index record for Document 5. It has a digital object identified (DOI) suggesting the record was created in 1992.

52. Attachment 5f is a list of 5 IEEE publications, identified by the IEEE Xplore Digital Library, that cite Document 5. These documents were published between 1995 and 1999.

53. Based on the evidence presented here—presentation at a prominent conference, publication, indexing, and citations—it is my opinion that Document 5 was accessible to an ordinarily skilled researcher by mid-year 1992 and in active use by researchers no later than 1995.

54. **Document 7.** Dénes, Tamás. "The 'Evolution' of Regular Graphs of Even Order by their Vertices" [in Hungarian]. *Matematikai Lapok*, 27, 3-4 (1976/1979): 365-377.

Authentication

55. Document 7 is a research paper published in a Hungarian mathematics journal. Attachment 7a is a copy of Document 7 from the University of Illinois at Urbana-Champaign Library. Attachment 7b is that library's catalog record for *Matematikai Lapok*, showing the holdings for Volume 27. Attachment 7c is the index record for Document 7 from MathSciNet, published by Mathematical Reviews, a common source for mathematical literature.

Public Accessibility

56. Attachment 7d is a copy of the Statewide Illinois Library Catalog record for *Matematikai Lapok*, showing this periodical is held by 97 libraries world-wide.

57. Attachment 7a shows that the issue of *Matematikai Lapok* has two date stamps. The first, 1 April 1980, is presumably the date when the periodical was received by the University of Illinois at Urbana-Champaign Library. The second date, 28 April 1980, is presumably the date it was added to the collections in the departmental Mathematics Library.

58. Based on the evidence presented here—periodical publication and library processing—it is my opinion that Document 7 was accessible to an ordinarily skilled researcher by May 1980.

59. NOTE: Document 7 has been translated from Hungarian to English. See Document 21, below, for my attestation regarding this translation.

60. **Document 8.** Toida, S. "Construction of Quartic Graphs." *Journal of Combinatorial Theory, Series B*, 16.2 (April 1974): 124-133.

Authentication

61. Document 8 is a research paper published in a scholarly journal. Attachment 8a is a copy of Document 8, with its journal cover, from the University of Illinois at Urbana-Champaign Library. Attachment 8b is a copy of that library's catalog record for the *Journal of Combinatorial Theory, Series B,* showing the

holding for Volume 16. Attachment 8c is the index record for Document 8 from MathSciNet, published by Mathematical Reviews, a common source for mathematical literature.

62. Document 8 is also available online. Attachment 8d is the index entry for Document 8 from ScienceDirect, a major index, abstract, and full-text service. Attachment 8e is a copy of Document 8 from ScienceDirect.

Public Accessibility

63. Attachment 8f is a copy of the Statewide Illinois Library Catalog record for the *Journal of Combinatorial Theory, Series B*, showing this journal is held by 463 libraries world-wide.

64. Attachment 8a shows a library processing date stamp of 15 April 1974.

65. Attachment 8g is a list of 9 publications citing Document 8 identified by Google Scholar. The earliest of these is by Francette Bories and Jean-Loup Jolivet, "Construction of 4-Regular Graphs," *Annals of Discrete Mathematics*, 17 (1983):99-118.

66. Based on the evidence presented here—publication in a widely held periodical, library processing, and citation—it is my opinion that Document 8 was accessible to an ordinarily skilled researcher by May 1974 and in actual use by researchers by no later than 1983.

67. **Document 9.** Bargen, Bradley and Donnelly, Peter. *Inside DirectX. In-Depth Techniques for Developing High-Performance Multimedia Applications*. Redmond, WA: Microsoft Press, 1998.

Authentication

68. Document 9 is a book published in 1998. Attachment 9a is a copy of the book's cover, title page, table of contents, other preliminary matter, and Chapter 1 from the Southern Illinois University Library. Attachment 9b is a copy of that library's catalog record for Document 9.

Public accessibility

69. Attachment 9c is a copy of the Statewide Illinois Library catalog record for Document 9, showing the book is held by 85 libraries world-wide. The date of Entry for this record is 22 September 1997.

70. Attachment 9d is a list of 2 publications citing Document 9 in 1998 identified by Google Scholar.

71. Based on the evidence presented here—book publication, library cataloging, and citations—it is my opinion that Document 9 was accessible to ordinarily skilled researchers by early October 1997 and in actual use by researchers in 1998.

72. **Document 10.** Bolding, Kevin, and Yost, William. "The Express Broadcast Network: A Network for Low-Latency Broadcast of Control Messages,"

in *ICAPP 95. IEEE First ICA ^{3}PP. IEEE First International Conference on Algorithms and Architectures for Parallel Processing. Brisbane, Australia, 19-21 April, 1995*, ed. V. L. Narasimhan, Vol. 1 (Piscataway, NJ: IEEE, 1995), 93-102.

Authentication

73. Document 10 is a paper presented at a conference in 1995 and published in the conference proceedings. Attachment 10a is a copy of Document 10, with the cover, title page, table of contents, and other preliminary matter of the conference proceedings, from the University of Illinois at Urbana-Champaign Library. Attachment 10b is a copy of that library's catalog record of the proceedings of the *IEEE First International Conference on Algorithms and Architectures for Parallel Processing*. Attachment 10c is the Scopus index record for Document 10. Scopus is a widely used index and abstract service in the sciences.

Public accessibility

74. Document 10 entered the public realm of discourse when it was presented at IEEE INFOCOM Conference on Computer Communications held on 4-8 May 1992 in Florence, Italy.

75. Attachment 10d is a copy of the Statewide Illinois Library Catalog monograph record for the proceedings of the *IEEE First International Conference on Algorithms and Architectures for Parallel Processing*. It shows that 3 libraries

world-wide hold these proceedings (treated as a monograph rather than a series). The date of Entry for this record is 27 November 1995.

76. Attachment 10e is a list of 3 publications citing Document 10 identified by Google Scholar. The publication not involving self-citation appeared in 1997.

77. It is my opinion based the information presented here—presentation at a conference and publication in the conference proceedings, library cataloging, indexing, and a citing document—that Document 10 was accessible to an ordinarily skilled researcher by mid-December 1995 and in actual use by researchers by 1997.

78. **Document 11.** Valiant, L. G. "Optimality of a Two-Phase Strategy for Routing in Interconnection Networks." *IEEE Transactions on Computers*, C-32.9 (September 1983): 861-863.

Authentication

79. Document 11 is a research paper published in 1983. Attachment 11a is a copy of Document 11 from the University of Illinois at Urbana-Champaign Library. Attachment 11b is a copy of that library's catalog record for *IEEE Transactions on Computers,* showing the holdings for Vol. 32. Attachment 11c is the IEEE Xplore Digital Library index record for Document 11.

Public Accessibility

80. Document 11d is a copy of the Statewide Illinois Library Catalog record for *IEEE Transactions on Computers*, showing that 661 libraries hold this periodical world-wide.

81. Attachment 11a shows a date stamp indicating that Document 11 was processed by the University of Illinois at Urbana-Champaign Library on 22 November 1983.

82. Attachment 11c, the IEEE Xplore Digital Library index record, has a digital object identifier (DOI) that suggests a 1983 date.

83. Attachment 11e is a list of 7 publications citing Document 11 identified in the IEEE Xplore Digital Library. These publications were issued between 1984 and 1995.

84. Based on the evidence presented here—publication in a widely held periodical, library processing, indexing, and citations—it is my opinion that Document 11 was accessible to an ordinarily skilled researcher by early December 1983 and in active use by researchers no later than 1984.

85. **Document 12.** Birman, Kenneth P., et al. "Bimodal Multicast." *ACM Transactions on Computer Systems*, 17.2 (May 1999): 41-88.

Authentication

86. Document 12 is a research paper published in 1999. Attachment 12a is a copy of Document 12, with its journal cover and table of contents, from the

University of Illinois at Urbana-Champaign Library. Attachment 12b is a copy of that library's catalog record for *ACM Transactions on Computer Systems*, showing the holdings for Volume 17.

87. Document 12 is also available online. Attachment 12c is a copy of Document 12 from the ACM Digital Library, a major source for computing literature. Attachment 12d is a copy of the ACM Digital Library index record for Document 12.

Public Accessibility

88. Attachment 12e is a copy of the Statewide Illinois Library Catalog record for *ACM Transactions on Computer Systems*, showing this journal is held by 634 libraries world-wide.

89. Attachment 12a shows a date stamp indicating that Document 12 was processed by the University of Illinois at Urbana-Champaign Library on 27 August 1999.

90. Attachment 12f is a list of 8 publications citing Document 12 identified by Google Scholar. All of these publications were issued in 1999. The earliest of these is by Suchitra Raman and Steven McCanne, "A Model, Analysis, and Protocol Framework for Soft State-Based Communication," in *SIGCOMM '99 Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, 29.4 (October 1999): 15-25.

91. Based on the evidence presented here—publication in a widely held periodical, library processing, and citations—it is my opinion that Document 12 was accessible to an ordinarily skilled researcher by early September 1999 and in actual use by researchers by no later than October 1999.

92. **Document 13.** Van Leeuwen, J. and Tan, R. B. "Interval Routing." *The Computer Journal*, 30.4 (August 1987): 298-307.

Authentication

93. Document 13 is a research paper published in 1987. Attachment 13a is a copy of Document 13, with its journal cover and table of contents, from the University of Illinois at Urbana-Champaign Library. Attachment 13b is a copy of that library's catalog record for *The Computer Journal*, showing the holdings for Volume 30.

94. Document 13 is also available online. Attachment 13c is a copy of Document 13 from Oxford Journals, the publisher of *The Computer Journal*. Attachment 13d is a copy of the Oxford Journals index record for Document 13.

Public Accessibility

95. Attachment 13e is a copy of the Statewide Illinois Library Catalog record for *The Computer Journal*, showing this journal is held by 597 libraries world-wide.

96. Attachment 13a shows a date stamp indicating that Document 13 was processed by the University of Illinois at Urbana-Champaign Library on 24 August 1987.

97. Attachment 13f is a copy of the first two pages of a list of 160 publications citing Document 13 identified by Google Scholar. All of these publications were issued between 1987 and 1998

98. Based on the evidence presented here—publication in a widely held periodical, library processing, and citations—it is my opinion that Document 12 was accessible to an ordinarily skilled researcher by early September 1987 and in active use by researchers as early as 1987.

99. **Document 14.** Vadapalli, P. "Two Different Families of Fixed Degree Regular Cayley Networks," in *Conference Proceedings of the 1995 IEEE Fourteenth Annual International Phoenix Conference on Computers and Communication, Scottsdale, Arizona, U.S.A., March 28-31 1995* (Piscataway, NJ: Institute of Electrical and Electronics Engineers, 1995), 263-269.

Authentication

100. Document 14 is a research paper given at a conference and published in the conference proceedings. Attachment 14a is a copy of Document 14, with its journal cover and table of contents, from the University of Illinois at Urbana-Champaign Library. Document 14b is a copy of the University of Illinois at

Urbana-Champaign Library series catalog record for the *Conference Proceedings of the 1995 IEEE Fourteenth Annual International Phoenix Conference on Computers and Communication,* showing the 1995 volume.

Public Accessibility

101. Document 14 entered the public realm of discourse when it was presented at the 1995 International Phoenix Conference of Computers and Communication, on 28-31 March 1995. The Conference Chair's Message (Attachment 1a) describes the conference as having a two-day technical program that "offers five parallel tracts . . . . In total, there will be 114 papers describing the latest results and development experiments including several special sessions and three panel sessions on topics of current interest."

102. Attachment 14c is a copy of the Statewide Illinois Library Catalog monograph record for the *Conference Proceedings of the 1995 IEEE Fourteenth Annual International Phoenix Conference on Computers and Communication,* showing this volume is held by 63 libraries worldwide. This record has an Entry date of 3 July 1995.

103. Attachment 14d is a copy of the IEEE Xplore Digital Library index record for Document 13. It has a digital object identifier (DOI) that suggests the record was created in 1995.

104. Document 14 was cited in a paper by Marcelo Moraes de Azevedo given at an IEEE 1996 symposium. Attachment 14e is the IEEE Xplore Digital Library record for Document 14 identifying the Moraes de Azevedo paper as a citing document.

105. Based on the evidence presented here—conference presentation, publication, library cataloging, indexing, and citation—it is my opinion that Document 14 was publicly accessible to an ordinarily skilled researcher by August 1995 and in active use by a researcher no later than 1996.

106. **Document 15**. Lee, Sunggu, and Shin, Kang G. "Interleaved All-to-All Reliable Broadcast on Meshes and Hypercubes." *Transactions on Parallel and Distributed Systems*, 5.5 (May 1994): 449-458

Authentication

107. Document 15 is a research paper published in 1994. Attachment 15a is a copy of Document 15, with its journal cover, from the University of Illinois at Urbana-Champaign Library. Attachment 15b is a copy of that library's catalog record for *Transactions on Parallel and Distributed Systems*, showing the holdings for Volume 5.

Public Accessibility

108. Attachment 15c is a copy of the Statewide Illinois Library Catalog record for *Transactions on Parallel and Distributed Systems*, showing this journal is held by 423 libraries world-wide.

109. Attachment 15a shows a date stamp indicating that Document 15 was processed by the University of Illinois at Urbana-Champaign Library on 17 May 1994.

110. Attachment 15d is the IEEE Xplore Digital Library index record for Document 15.

111. Attachment 15e is a list of 38 documents citing Document 15 identified by the IEEE Xplore Digital Library. Four of the non-patent documents (numbers 3-6) were published between 1996 and 1997.

112. Based on the evidence presented here—publication in a widely held periodical, library processing, indexing, and citations—it is my opinion that Document 15 was publicly available to ordinarily skilled researchers by June 1994 and in active use by researchers by at least 1996.

113. **Document 16.** Yoshikawa, Chad, et al. "Using Smart Clients to Build Scalable Services," in *Proceedings of the USENIX 1997 Annual Technical Conference. January 6-10 1997. Anaheim, California, USA* (Berkeley, CA: USENIX Association, 1997), 105-117.

Authentication

114. Document 16 is a research paper present at a conference and published in the conference proceedings. Attachment 16a is a copy of Document 16, along with the cover and contents pages of the *Proceedings*, from the Rutgers University Library. Attachment 16b is a copy of that library's catalog record for the *Proceedings of the USENIX 1997 Annual Technical Conference.*

Public Accessibility

115. Document 16 entered the public realm of discourse when it was presented on 8 January 1997 in a one-hour session featuring two papers at USENIX 1997 Conference held on 6-10 January 1997 in Anaheim, CA, as indicated in Attachment 16c, an Internet Archive capture of information about the USENIX 1997 Annual Technical Conference.

116. Attachment 16d is a copy of the Statewide Illinois Library Catalog record for the *Proceedings of the USENIX 1997 Annual Technical Conference*, showing this publication is held by 15 libraries world-wide.

117. Attachment 16e is a copy of the ACM Digital Library index record for Document 16.

118. Attachment 16f is a list of 51 documents citing Document 16 identified by the ACM Digital Library. Seven of these documents were published between 1997 and 1999. One of them is by Yatin Chawathe and Eric A. Brewer, "System Support for Scalable and Fault Tolerant Internet Services," in *Middleware*

'98. Proceedings of the IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing, September 1998, ed. Jochen Seitz (London: Springer, 1998), 71-88. Attachment 16g is the ACM Digital Library index record for the Chawathe paper, showing the citation to Document 16.

119. Based on the evidence presented here—conference presentation, publication in the conference proceedings, indexing, and citations—it is my opinion that Document 16 was accessible to an ordinarily skilled researcher in 1997 and in actual use by researches no later than September 1998.

120. **Document 17**. Naugle, Matthew. *Network Protocol Handbook*. McGraw-Hill Series on Computer Communications. New York, NY: McGraw-Hill, 1994.

Authentication

121. Document 17 is a book published late in 1993. Attachment 17a is a copy of the cover, title page, table of contents, preliminary matter, and Chapter 6 of Document 17 from the University of Illinois at Urbana-Champaign Library. Attachment 17b is that library's catalog record, in MARC format, for Document 17.

Public Accessibility

122. Attachment 17c is a copy of the Statewide Illinois Library Catalog record for Document 17, showing it is held by 268 libraries world-wide.

123. Attachment 17d is a copy of the United States Copyright Office registration information for Document 17. It shows that Document 17 was published on1 October 1993 and its copyright registered on 27 October 1993. Books published in the last quarter of the year often bear a printed publication date of the following year, as is the case with Document 17.

124. Attachment 17b, a library catalog record, shows in the 040 MARC Field that the catalog record for Document 17 was created at the Library of Congress (OCLC code DLC). The 008 MARC Field indicates this record was entered on 3 May 1993. Attachment 17a shows, on the verso of the title page, that this is a cataloging-in-publication record.

125. Attachment 17e is a copy of the list of 11 documents citing Document 17 identified by Google Scholar. These documents were published between 1994 and 1998.

126. Based on the evidence presented here—a widely held book, Copyright Office registration, catalog record, citations—it is my opinion that Document 17 was accessible by an ordinarily skilled researcher by no later than November 1993 and in actual use by researchers by 1994.

127. **Document 18.** Todd, Terence D. "The Token Grid Network." *IEEE/ACM Transactions on Networking*, 2.3 (June 1994): 279-287.

Authentication

128. Document 18 is a research paper published in in a periodical. Attachment 18a is a copy of Document 18 from the University of Illinois at Urbana-Champaign Library. Attachment 18b is a copy of that library's catalog record for the *IEEE/ACM Transactions on Networking*, showing the holdings for Volume 2.

Public Accessibility

129. Attachment 18c is a copy of the Statewide Illinois Library Catalog record for the *IEEE/ACM Transactions on Networking,* showing this periodical is held by 581 libraries world-wide.

130. Attachment 18a has a date stamp indicating the periodical containing Document 18 was processed by the University of Illinois at Urbana-Champaign Library on 7 September 1994.

131. Attachment 18d is the Scopus index record for Document 18.

132. Attachment 18e is a list of 12 documents citing Document 18 identified by Scopus. Five of these documents were published in 1996 and 1997.

133. **Based on the evidence presented here—publication in a widely held periodical, library processing, indexing, and citations—it is my opinion that Document 18 was accessible to an ordinarily skilled research by October 1994 and in actual use by researchers by 1996.**

134. **Document 19.** Damani, Om P. et al. "ONE-IP: techniques for hosting a service on a cluster of machines." *Computer Networks and ISDN Systems*, 29.8-13 (September 1997): 1019-1027.

Authentication

135. Document 19 is a research paper published in a periodical. Attachment 19a is a copy of Document 19 from the DePaul University Library. Attachment 19b is a copy of that library's catalog record for the *Computer Networks and ISDN Systems*, showing the library holdings for the print Volume 29.

Public Accessibility

136. Attachment 19c is a copy of the Statewide Illinois Library Catalog record for the *Computer Networks and ISDN Systems,* showing this periodical is held by 309 libraries world-wide.

137. Attachment 19a has a library date stamp indicated that this volume of the *Computer Networks and ISDN Systems* was received on 2 January 1998.

138. Attachment 19d is the Engineering Village index record for Document 19, with a 1997 copyright date.

139. Attachment 19e is a list of 4 documents citing Document 19 identified by Scopus. These documents were published in 1998 and 1999.

140. **Based on the evidence presented here—publication in a widely held periodical, indexing, and citations—it is my opinion that Document 19**

was accessible to an ordinarily skilled researcher by early 1998 and in actual use by researchers by 1998.

141. **Document 20**. Vahdat, Amin M. et al. "WebFS: A Global Cache Coherent File System." Available at https://users.cs.duke.edu/~vahdat/webfs/webfs.html .

Authentication

142. Document 20 is a research paper available at a Web site maintained by the Computer Science department at Duke University. Attachment 20a is a copy of Document 20 from this Web site.

Public Accessibility

143. Attachment 20b is a copy of the Internet Archive capture, of 29 January 1999, of an index page including an HTML link to Document 20. This HTML document is substantively the same as the Attachment 20a PDF.

144. Attachment 20c is a copy of the list of 12 documents citing Document 20 identified by Google Scholar. These documents were published between 1996 and 1998. Document 20d is a copy of the IEEE Xplore Digital Library record for one of these papers, by J. Carter, "Khazana: An Infrastructure for Building Distributed Services," in the *Proceedings of the 18th International Conference on Distributed Computing Systems, Amsterdam, 1998.*

145. Based on the evidence presented here—Web site publication, Internet Archive capture, and citations—it is my opinion that Document 20 was demonstrably accessible to an ordinarily skilled researcher by no later 29 January 1999 and in actual use by researchers by no later than 1998.

146. **Document 21.** Copy of a translation of Document 7, with an affidavit by Stephen Matthew Grimes.

147. Attachment 21a is a copy of an English translation of Document 7 by Stephen Grimes; it is provided by counsel. I have compared the copy of Document 7 included with this translation to Attachment 7a of this declaration, which is a copy of Document 7 from the University of Illinois at Urbana-Champaign Library. I find them to be the same document.

ATTACHMENTS

148. The following attachments are true and accurate representations of library material and online documents and records, as they are identified above. All attachments were created on 29 February – 7 March 2016, and all URLs referenced in this declaration were available 8 March 2016.

Attachment 1a: Copy of Document 1 from the Northwestern University Library

Attachment 1b: Northwestern University Library catalog record for the proceedings of *1997 IEEE International Conference on Communications*

Attachment 1c: Copy of Document 1 from the IEEE Xplore Digital Library

Attachment 1d: Statewide Illinois Library Catalog monograph record for the proceedings of *1997 IEEE International Conference on Communications*

Attachment 1e: Statewide Illinois Library Catalog series record for the proceedings of *1997 IEEE International Conference on Communications*

Attachment 1f: IEEE Xplore Digital Library index record for Document 1

Attachment 2a: Statewide Illinois Library Catalog record for Document 2

Attachment 3a: Copy of Document 1 from the University of Illinois at Urbana-Champaign Library

Attachment 3b: University of Illinois at Urbana-Champaign Library catalog record for the *Proceedings of the 16th International Conference on Distributed Computing Systems*

Attachment 3c: Copy of Document 3 from the IEEE Xplore Digital Library

Attachment 3d: Statewide Illinois Library Catalog monograph record for the *Proceedings of the 16th International Conference on Distributed Computing Systems*

Attachment 3e: Statewide Illinois Library Catalog series record for the *Proceedings of the 16th International Conference on Distributed Computing Systems*

Attachment 3f: IEEE Xplore Digital Library index record for Document 3

Attachment 3g: List of publications citing Document 3 identified by Google Scholar

Attachment 4a: Statewide Illinois Library Catalog record for Document 4

Attachment 4b: Copy of the Document 4 from ProQuest

Attachment 4c: ProQuest index record for Document 4

Attachment 4d: List of publications citing Document 4 identified by Google Scholar

Attachment 4e: University of Twente Publications index record for a publication citing Document 4.

Attachment 5a: Copy of Document 5 from the University of Illinois at Urbana-Champaign Library

Attachment 5b: University of Illinois at Urbana-Champaign Library catalog series record for *IEEE INFOCOM '92: The Conference on Computer Communications*

Attachment 5c: Copy of Document 5 from the IEEE Xplore Digital Library

Attachment 5d: Statewide Illinois Library Catalog series record for *IEEE INFOCOM '92: The Conference on Computer Communications*

Attachment 5e: IEEE Xplore Digital Library index record for Document 5

Attachment 5f: List of publications citing Document 5 identified by the IEEE Xplore Digital Library

Attachment 7a: Copy of Document 7 from the University of Illinois at Urbana-Champaign Library

Attachment 7b: University of Illinois at Urbana-Champaign Library catalog record for *Matematikai Lapok*

Attachment 7c: MathSciNet index record for Document 7

Attachment 7d: Statewide Illinois Library Catalog record for *Matematikai Lapok*

Attachment 8a: Copy of Document 8 from the University of Illinois at Urbana-Champaign Library

Attachment 8b: University of Illinois at Urbana-Champaign Library catalog record for the *Journal of Combinatorial Theory, Series B*

Attachment 8c: MathSciNet index record for Document 8

Attachment 8d: ScienceDirect index record for Document 8

Attachment 8e: Copy of Document 8 from ScienceDirect

Attachment 8f: Statewide Illinois Library Catalog record for the *Journal of Combinatorial Theory, Series B*

Attachment 8g: List of publications citing Document 8 identified by Google Scholar

Attachment 9a: Copy of Document 9 from the Southern Illinois University Library

Attachment 9b: Southern Illinois University Library catalog record for Document 9

Attachment 9c: Statewide Illinois Library Catalog record for Document 9

Attachment 9d: List of publications citing Document 9 identified by Google Scholar

Attachment 10a: Copy of Document 10 from the University of Illinois at Urbana-Champaign Library

Attachment 10b: University of Illinois at Urbana-Champaign Library catalog record of the proceedings of the *IEEE First International Conference on Algorithms and Architectures for Parallel Processing*

Attachment 10c: Scopus index record for Document 10

Attachment 10d: Statewide Illinois Library Catalog record for the proceedings of the *IEEE First International Conference on Algorithms and Architectures for Parallel Processing*

Attachment 10e: List of publications citing Document 10 identified by Google Scholar

Attachment 10f: ProQuest index record for a thesis citing Document 10

Attachment 11a: Copy of Document 11 from the University of Illinois at Urbana-Champaign Library

Attachment 11b: University of Illinois at Urbana-Champaign Library catalog record for *IEEE Transactions on Computers*

Attachment 11c: IEEE Xplore Digital Library index record for Document 11

Attachment 11d: Statewide Illinois Library Catalog record for *IEEE Transactions on Computers*

Attachment 11e: List of publications citing Document 11 identified by IEEE Xplore Digital Library

Attachment 12a: Copy of Document 12 from the University of Illinois at Urbana-Champaign Library

Attachment 12b: University of Illinois at Urbana-Champaign Library catalog record for *ACM Transactions on Computer Systems*

Attachment 12c: Copy of Document 12 from the ACM Digital Library

Attachment 12d: ACM Digital Library index record for Document 12

Attachment 12e: Statewide Illinois Library Catalog record for *ACM Transactions on Computer Systems*

Attachment 12f: List of publications citing Document 12 identified by Google Scholar

Attachment 13a: Copy of Document 13 from the University of Illinois at Urbana-Champaign Library

Attachment 13b: University of Illinois at Urbana-Champaign Library catalog record for *The Computer Journal*

Attachment 13c: Copy of Document 13 from Oxford Journals

Attachment 13d: Oxford Journals index record for Document 13

Attachment 13e: Statewide Illinois Library Catalog record for *The Computer Journal*

Attachment 13f: List of publications citing Document 13 identified by Google Scholar

Attachment 14a: Copy of Document 14, with cover, title page, table of contents, and preliminary matter, from the University of Illinois at Urbana-Champaign Library

Attachment 14b: University of Illinois at Urbana-Champaign Library catalog record for *Conference Proceedings of the 1995 IEEE Fourteenth Annual International Phoenix Conference on Computers and Communication*

Attachment 14c: Statewide Illinois Library Catalog record for *Conference Proceedings of the 1995 IEEE Fourteenth Annual International Phoenix Conference on Computers and Communication*

Attachment 14d: IEEE Xplore Digital Library index record for Document 14

Attachment 14e: Publication citing Document 14 identified by IEEE Xplore Digital Library

Attachment 15a: Copy of Document 15 from the University of Illinois at Urbana-Champaign Library

Attachment 15b: University of Illinois at Urbana-Champaign Library catalog record for IEEE *Transactions on Parallel and Distributed Systems*

Attachment 15c: Statewide Illinois Library Catalog record for *Transactions on Parallel and Distributed Systems*

Attachment 15d: IEEE Xplore Digital Library index record for Document 15

Attachment 15e: List of publications citing Document 15 identified by the IEEE Xplore Digital Library

Attachment 16a: Copy of Document 16 from the Rutgers University Library

Attachment 16b: Rutgers University Library catalog record for the *Proceedings of the USENIX 1997 Annual Technical Conference*

Attachment 16c: Internet Archive capture of a description of the USENIX 1997 Annual Technical Conference

Attachment 16d: Statewide Illinois Library Catalog record for the *Proceedings of the USENIX 1997 Annual Technical Conference*

Attachment 16e: ACM Digital Library index record for Document 16

Attachment 16f: List of publications citing Document 16 identified by the ACM Digital Library

Attachment 16g: ACM Digital Library index record for a publication citing Document 16

Attachment 17a: Copy of Document 17 from the University of Illinois at Urbana-Champaign Library

Attachment 17b: University of Illinois at Urbana-Champaign Library catalog record for Document 17

Attachment 17c: Statewide Illinois Library Catalog record for Document 17

Attachment 17d: United States Copyright Office registration information for Document 17

Attachment 17e: List of publications citing Document 17 identified by Google Scholar

Attachment 18a: Copy of Document 18 from the University of Illinois at Urbana-Champaign Library

Attachment 18b: University of Illinois at Urbana-Champaign Library catalog record for *Transactions on Networking*

Attachment 18c: Statewide Illinois Library Catalog record for *Transactions on Networking*

Attachment 18d: Scopus index record for Document 18

Attachment 18e: List of publications citing Document 18 identified by Scopus

Attachment 19a: Copy of Document 19 from the DePaul University Library

Attachment 19b: DePaul University Library catalog record for the print format of *Computer Networks and ISDN Systems*

Attachment 19c: Statewide Illinois Library Catalog record for *Computer Networks and ISDN Systems*

Attachment 19d: Engineering Village index record for Document 19

Attachment 19e: List of publications citing Document 19 identified by Scopus

Attachment 20a: Copy of Document 20 from a Duke University Web site

Attachment 20b: Internet Archive capture of an index page with a link to Document 20

Attachment 20c: List of publications citing Document 20 identified by Google Scholar

Attachment 20d: IEEE Xplore Digital Library record for a publication citing Document 20

Attachment 21a: Copy of an English translation of Document 7

ATTESTATION

149. This declaration and my opinions herein are made to the best of my knowledge and understanding, and based on the material available to me, at the time of signing this declaration. I declare under penalty of perjury under the laws of the United States of America that all statements made of my own knowledge are true and that all statements made on information and belief are believed to be true.

I understand that willful false statements and the like are punishable by fine or imprisonment, or both (18 U.S.C. § 1001).

9 March 2016

______________________ _______________________________

Date Name

EXHIBIT A: RESUME

SCOTT BENNETT
Yale University Librarian Emeritus

711 South Race
Urbana, Illinois 61801-4132
2scottb@prairienet.org
217-367-9896

EMPLOYMENT

Retired, 2001. Retirement activities include:

- Managing Partner in Prior Art Documentation Services, LLC, 2015-. This firm provides documentation services to patent attorneys; more information is available at http://www.priorartdocumentation.com
- Consultant on library space design, 2004- . This consulting practice is rooted in a research, publication, and public speaking program conducted since I retired from Yale University in 2001. I have served more than 50 colleges and universities in the United States and abroad with projects ranging in likely cost from under $50,000 to over $100 million. More information is available at http://www.libraryspaceplanning.com/
- Senior Advisor for the library program of the **Council of Independent Colleges**, 2001-2009
- Member of the Wartburg College Library Advisory Board, 2004-
- Visiting Professor, Graduate School of Library and Information Science, **University of Illinois at Urbana-Champaign**, Fall 2003

University Librarian, **Yale University**, 1994-2001

Director, The Milton S. Eisenhower Library, **The Johns Hopkins University**, Baltimore, Maryland, 1989-1994

Assistant University Librarian for Collection Management, **Northwestern University**, Evanston, Illinois, 1981-1989

Instructor, Assistant and Associate Professor of Library Administration**, University of Illinois at Urbana-Champaign**, 1974-1981

Assistant Professor of English, **University of Illinois at Urbana-Champaign**, 1967-1974

Woodrow Wilson Teaching Intern, **St. Paul's College**, Lawrenceville, Virginia, 1964-1965

EDUCATION

University of Illinois, M.S., 1976 (Library Science)
Indiana University, M.A., 1966; Ph.D., 1967 (English)
Oberlin College, A.B. magna cum laude, 1960 (English)

HONORS AND AWARDS

Morningside College (Sioux City, IA) Doctor of Humane Letters, 2010

American Council of Learned Societies Fellowship, 1978-1979; Honorary Visiting Research Fellow, Victorian Studies Centre**, University of Leicester**, 1979; **University of Illinois** Summer Faculty Fellowship, 1969

Indiana University Dissertation Year Fellowship and an **Oberlin College** Haskell Fellowship, 1966-1967; **Woodrow Wilson** National Fellow, 1960-1961

PROFESSIONAL ACTIVITIES

American Association for the Advancement of Science: Project on Intellectual Property and Electronic Publishing in Science, 1999-2001

American Association of University Professors: University of Illinois at Urbana-Champaign Chapter Secretary and President, 1975-1978; Illinois Conference Vice President and President, 1978-1984; national Council, 1982-1985, Committee F, 1982-1986, Assembly of State Conferences Executive Committee, 1983-1986, and Committee H, 1997-2001 ; Northwestern University Chapter Secretary/Treasurer, 1985-1986

Association of American Universities: Member of the Research Libraries Task Force on Intellectual Property Rights in an Electronic Environment, 1993-1994, 1995-1996

Association of Research Libraries: Member of the Preservation Committee, 1990-1993; member of the Information Policy Committee, 1993-1995; member of the Working Group on Copyright, 1994-2001; member of the Research Library Leadership and Management Committee, 1999-2001; member of the Board of Directors, 1998-2000

Carnegie Mellon University: Member of the University Libraries Advisory Board, 1994

Center for Research Libraries: Program Committee, 1998-2000

Johns Hopkins University Press: Ex-officio member of the Editorial Board, 1990-1994; Co-director of Project Muse, 1994

Library Administration and Management Association, Public Relations Section, Friends of the Library Committee, 1977-1978

Oberlin College: Member of the Library Visiting Committee, 1990, and of the Steering Committee for the library's capital campaign, 1992-1993; President of the Library Friends, 1992-1993, 2004-2005; member, Friends of the Library Council, 2003-

Research Society for Victorian Periodicals: Executive Board, 1971-1983; Co-chairperson of the Executive Committee on Serials Bibliography, 1976-1982; President, 1977-1982

A Selected Edition of W.D. Howells (one of several editions sponsored by the MLA Center for Editions of American Authors): Associate Textual Editor, 1965-1970; Center for Editions of American Authors panel of textual experts, 1968-1970

Victorian Studies: Editorial Assistant and Managing Editor, 1962-1964

Wartburg College: member, National Advisory Board for the Vogel Library, 2004-

Some other activities: Member of the **Illinois State Library** Statewide Library and Archival Preservation Advisory Panel; member of the **Illinois State Archives** Advisory Board; member of a committee advising the **Illinois Board of Higher Education** on the cooperative management of research collections; chair of a major collaborative research project conducted by the **Research Libraries Group** with support from Conoco, Inc.; active advisor on behalf of the **Illinois Conference AAUP** to faculty and administrators on academic freedom and tenure matters in northern Illinois.

Delegate to **Maryland Governor's Conference on Libraries and Information Service**; principal in initiating state-wide preservation planning in Maryland; principal in an effort to widen the use of mass deacidification for the preservation of library materials through cooperative action by the **Association of Research Libraries** and the **Committee on Institutional Cooperation**; co-instigator of a campus-wide information service for **Johns Hopkins University**; initiated efforts with the **Enoch Pratt Free Library** to provide information services to Baltimore's Empowerment Zones; speaker or panelist on academic publishing, copyright, scholarly communication, national and regional preservation planning, mass deacidification.

Consultant for the **University of British Columbia** (1995), **Princeton University** (1996), **Modern Language Association**, (1995, 1996), **Library of Congress** (1997), **Center for Jewish History** (1998, 2000-), **National Research Council** (1998); Board of Directors for the **Digital Library Federation**, 1996-2001; accreditation visiting team at **Brandeis University** (1997); mentor for **Northern Exposure to Leadership** (1997); instructor and mentor for ARL's **Leadership and Career Development Program** (1999-2000)

At the **Northwestern University Library**, led in the creation of a preservation department and in the renovation of the renovation, for preservation purposes, of the Deering Library book stacks.

At the **Milton S. Eisenhower Library**, led the refocusing and vitalization of client-centered services; strategic planning and organizational restructuring for the library; building renovation planning. Successfully completed a $5 million endowment campaign for the humanities collections and launched a $27 million capital campaign for the library.

At the **Yale University Library**, participated widely in campus-space planning, university budget planning, information technology development, and the promotion of effective teaching and learning; for the library has exercised leadership in space planning and renovation, retrospective conversion of the card catalog, preservation, organizational development, recruitment of minority librarians, intellectual property and copyright issues, scholarly communication, document delivery services among libraries, and instruction in the use of information resources. Oversaw approximately $70 million of library space renovation and construction. Was co-principal investigator for a grant to plan a digital archive for Elsevier Science.

Numerous to invitations speak at regional, national, and other professional meetings and at alumni meetings. Lectured and presented a series of seminars on library management at the **Yunnan University Library**, 2002. Participated in the 2005 International Roundtable for Library and Information Science sponsored by the **Kanazawa Institute of Technology** Library Center and the Council on Library and Information Resources.

PUBLICATIONS

“Putting Learning into Library Planning,” *portal: Libraries and the Academy*, 15, 2 (April 2015), 215-231.

“How librarians (and others!) love silos: Three stories from the field “ available at the Learning Spaces Collaborary Web site, http://www.pkallsc.org/

“Learning Behaviors and Learning Spaces,” *portal: Libraries and the Academy,* 11, 3 (July 2011), 765-789.

“Libraries and Learning: A History of Paradigm Change,” *portal: Libraries and the Academy,* 9, 2 (April 2009), 181-197. Judged as the best article published in the 2009 volume of *portal*.

“The Information or the Learning Commons: Which Will We Have?” *Journal of Academic Librarianship*, 34 (May 2008), 183-185. One of the ten most-cited articles published in JAL, 2007-2011.

“Designing for Uncertainty: Three Approaches,” *Journal of Academic Librarianship*, 33 (2007), 165–179.

“Campus Cultures Fostering Information Literacy,” *portal: Libraries and the Academy*, 7 (2007), 147-167. Included in Library Instruction Round Table Top Twenty library instruction articles published in 2007

“Designing for Uncertainty: Three Approaches,” *Journal of Academic Librarianship*, 33 (2007), 165–179.

“First Questions for Designing Higher Education Learning Spaces,” *Journal of Academic Librarianship*, 33 (2007), 14-26.

“The Choice for Learning,” *Journal of Academic Librarianship*, 32 (2006), 3-13.

With Richard A. O’Connor, “The Power of Place in Learning,” *Planning for Higher Education*, 33 (June-August 2005), 28-30

“Righting the Balance,” in *Library as Place: Rethinking Roles, Rethinking Space* (Washington, DC: Council on Library and Information Resources, 2005), pp. 10-24

Libraries Designed for Learning (Washington, DC: Council on Library and Information Resources, 2003)

“The Golden Age of Libraries,” in *Proceedings of the International Conference on Academic Librarianship in the New Millennium: Roles, Trends, and Global Collaboration*, ed. Haipeng Li (Kunming: Yunnan University Press, 2002), pp. 13-21. This is a slightly different version of the following item.

“The Golden Age of Libraries,” *Journal of Academic Librarianship*, 24 (2001), 256-258

“Second Chances. An address . . . at the annual dinner of the Friends of the Oberlin College Library November 13 1999,” Friends of the Oberlin College Library, February 2000

"Authors' Rights," *The Journal of Electronic Publishing* (December 1999), http://www.press.umich.edu/jep/05-02/bennett.html

"Information-Based Productivity," in *Technology and Scholarly Communication*, ed. Richard Ekman and Richard E. Quandt (Berkeley, 1999), pp. 73-94

"Just-In-Time Scholarly Monographs: or, Is There a Cavalry Bugle Call for Beleaguered Authors and Publishers?" *The Journal of Electronic Publishing* (September 1998), http://www.press.umich.edu/jep/04-01/bennett.html

"Re-engineering Scholarly Communication: Thoughts Addressed to Authors," *Scholarly Publishing*, 27 (1996), 185-196

"The Copyright Challenge: Strengthening the Public Interest in the Digital Age," *Library Journal*, 15 November 1994, pp. 34-37

"The Management of Intellectual Property," *Computers in Libraries*, 14 (May 1994), 18-20

"Repositioning University Presses in Scholarly Communication," *Journal of Scholarly Publishing*, 25 (1994), 243-248. Reprinted in *The Essential JSP. Critical Insights into the World of Scholarly Publishing. Volume 1: University Presses* (Toronto: University of Toronto Press, 2011), pp. 147-153

"Preservation and the Economic Investment Model," in *Preservation Research and Development. Round Table Proceedings, September 28-29, 1992*, ed. Carrie Beyer (Washington, D.C.: Library of Congress, 1993), pp. 17-18

"Copyright and Innovation in Electronic Publishing: A Commentary," *Journal of Academic Librarianship*, 19 (1993), 87-91; reprinted in condensed form in *Library Issues: Briefings for Faculty and Administrators*, 14 (September 1993)

with Nina Matheson, "Scholarly Articles: Valuable Commodities for Universities," *Chronicle of Higher Education*, 27 May 1992, pp. B1-B3

"Strategies for Increasing [Preservation] Productivity*," Minutes of the [119th] Meeting [of the Association of Research Libraries]* (Washington, D.C., 1992), pp. 39-40

"Management Issues: The Director's Perspective," and "Cooperative Approaches to Mass Deacidification: Mid-Atlantic Region," in *A Roundtable on Mass Deacidification*, ed. Peter G. Sparks (Washington, D.C.: Association of Research Libraries, 1992), pp. 15-18, 54-55

"The Boat that Must Stay Afloat: Academic Libraries in Hard Times," *Scholarly Publishing*, 23 (1992), 131-137

"Buying Time: An Alternative for the Preservation of Library Material," ACLS *Newsletter*, Second Series 3 (Summer, 1991), 10-11

"The Golden Stain of Time: Preserving Victorian Periodicals" in *Investigating Victorian Journalism*, ed. Laurel Brake, Alex Jones, and Lionel Madden (London: Macmillan, 1990), pp. 166-183

“Commentary on the Stephens and Haley Papers” in *Coordinating Cooperative Collection Development: A National Perspective*, an issue of *Resource Sharing and Information Networks*, 2 (1985), 199-201

“The Editorial Character and Readership of *The Penny Magazine*: An Analysis,” *Victorian Periodicals Review*, 17 (1984), 127-141

“Current Initiatives and Issues in Collection Management*,” Journal of Academic Librarianship*, 10 (1984), 257-261; reprinted in *Library Lit: The Best of 85*

“Revolutions in Thought: Serial Publication and the Mass Market for Reading” in *The Victorian Periodical Press: Samplings and Soundings*, ed. Joanne Shattock and Michael Wolff (Leicester: Leicester University Press, 1982), pp. 225-257

“Victorian Newspaper Advertising: Counting What Counts,” *Publishing History*, 8 (1980), 5-18

“Library Friends: A Theoretical History” in *Organizing the Library’s Support: Donors, Volunteers, Friends*, ed. D.W. Krummel, Allerton Park Institute Number 25 (Urbana: University of Illinois Graduate School of Library Science, 1980), pp. 23-32

“The Learned Professor: being a brief account of a scholar [Harris Francis Fletcher] who asked for the Moon, and got it,” *Non Solus*, 7 (1980), 5-12

“Prolegomenon to Serials Bibliography: A Report to the [Research] Society [for Victorian Periodicals],” *Victorian Periodicals Review*, 12 (1979), 3-15

“The Bibliographic Control of Victorian Periodicals” in *Victorian Periodicals: A Guide to Research*, ed. J. Don Vann and Rosemary T. VanArsdel (New York: Modern Language Association, 1978), pp. 21-51

“John Murray’s Family Library and the Cheapening of Books in Early Nineteenth Century Britain,” *Studies in Bibliography*, 29 (1976), 139-166. Reprinted in Stephen Colclough and Alexis Weedon, eds., *The History of the Book in the West: 1800-1914*, Vol. 4 (Farnham, Surrey: Ashgate, 2010), pp. 307-334.

with Robert Carringer, “Dreiser to Sandburg: Three Unpublished Letters,” *Library Chronicle*, 40 (1976), 252-256

“David Douglas and the British Publication of W. D. Howells’ Works,” *Studies in Bibliography*, 25 (1972), 107-124

as primary editor, W. D. Howells, *Indian Summer* (Bloomington: Indiana University Press, 1971)

“The Profession of Authorship: Some Problems for Descriptive Bibliography” in *Research Methods in Librarianship: Historical and Bibliographic Methods in Library Research*, ed. Rolland E. Stevens (Urbana: University of Illinois Graduate School of Library Science, 1971), pp. 74-85

edited with Ronald Gottesman, *Art and Error: Modern Textual Editing* (Bloomington: Indiana University Press, 1970)--also published in London by Methuen, 1970

"Catholic Emancipation, the *Quarterly Review*, and Britain's Constitutional Revolution," *Victorian Studies*, 12 (1969), 283-304

as textual editor, W. D. Howells, *The Altrurian Romances* (Bloomington: Indiana University Press, 1968); introduction and annotation by Clara and Rudolf Kirk

as associate textual editor, W. D. Howells, *Their Wedding Journey* (Bloomington: Indiana University Press, 1968); introduction by John Reeves

"A Concealed Printing in W. D. Howells," *Papers of the Bibliographic Society of America*, 61 (1967), 56-60

editor, *Non Solus*, A Publication of the University of Illinois Library Friends, 1974-1981

editor, Robert B. Downs Publication Fund, University of Illinois Library, 1975-1981

reviews, short articles, etc. in *Victorian Studies*, *Journal of English and German Philology*, *Victorian Periodicals Newsletter*, *Collection Management*, *Nineteenth-Century Literature*, *College & Research Libraries*, *Scholarly Publishing Today*, *ARL Newsletter*, *Serials Review*, *Library Issues*, *S[ociety for] S[cholarly] P[ublishing] Newsletter*, and *Victorian Britain: An Encyclopedia*

Attachment 1a: Copy of Document 1 from the Northwestern University Library

Northwestern University Interlibrary Loan

TN: 818262

Borrower: UIU

Lending String:
EEM,*INU,NJR,CLU,NRC,INT,JHE,CIN,FHM,CUV,
LHL,AUM,PUL

Patron:

Journal Title: 1997 IEEE International Conference on Communications : towards the knowledge millennium : ICC '97, 8-12 June 1997, Montréal, Québec, Canada, confere

Volume: 1997 (this has 3 volumes, from **Issue:**
Month/Year: 1997**Pages:** 1381-1386

Article Author: Shoubridge, Peter and Arek Dadej

Article Title: Hybrid Routing in Dynamic Networks; Also table of contents and cover page that shows acquisitions date

Imprint: [New York] : Institute of Electrical and Electronics Engineers ; Piscataway, NJ. : IEEE Service Center, ©1997.

Barcode:
35556027685031 35556027685023 3555602768 5015

Call #: 621.38 I22i 1997 3 vols

Location: Oak Grove Library Center

Charge
Maxcost: 35.00IFM

Shipping Address:
Interlibrary Borrowing
128 Library University of Illinois
1408 West Gregory Drive
Urbana, Illinois 61801
United States

Hybrid Routing in Dynamic Networks

Peter Shoubridge
Communications Division, Defence Science Technology Organisation
Salisbury, South Australia, 5108

Arek Dadej
Institute of Telecommunications Research, University of South Australia
The Levels, South Australia, 5095

Abstract— **In mobile radio communication networks the distribution of traffic loads and network topologies may vary from nearly static to very dynamic. This dynamic behaviour may vary both in space and in time. Since routing algorithms tend to be well suited to specific networking environments, it is very difficult to select a single routing algorithm that is most appropriate for a given network, if the network is subjected to varying degrees of dynamic behaviour. This paper proposes a routing strategy that smoothly adapts to changing network conditions by combining two distinct routing principles into a single hybrid routing procedure. The hybrid routing strategy exhibits a smooth change from shortest path routing to constrained flooding, as the behaviour of the network (or regions within) changes from quasi-static to very dynamic.**

I. Introduction

In addition to successfully forwarding user traffic between source and destination nodes in a communications network, the routing function generally seeks to optimise some performance criteria such as network utilisation or throughput, or perhaps average delay. 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.

For efficient routing, nodes require knowledge of network topology. Links may fail or become congested, nodes or users could be mobile, resulting in changes to source-destination paths. Information regarding these changes must be conveyed to network nodes so that appropriate routing can take place. This information exchange consumes network resources and presents a cost in maintaining up to date routing information in network nodes. 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. Inaccurate routing can result in lost or delayed traffic causing end-to-end retransmissions which further load the network. This additional load is itself an unwanted overhead which could be avoided with a better routing strategy.

Algorithms that can determine shortest paths between source and destination nodes are typically used in static or quasi-static networks. They require periods of network stability so routing tables can settle to reflect true shortest paths. The most efficient algorithm is that described by Dijkstra which requires complete knowledge of the network topology [1, p. 103]. This algorithm can be executed centrally or topology information can be broadcast to all nodes in a decentralised implementation [2]. Another algorithm, suited to distributed architectures, is a decentralised version of Ford and Fulkerson's algorithm which only requires information exchange between immediate neighbouring nodes [3, p. 268], [4, p. 130]. More recent algorithms derived from these offer significant improvements in performance [5], [6], [7].

Routing by flooding is often used in very dynamic networks where destination nodes are highly mobile or topological changes occur very frequently and therefore the network connectivity is uncertain. Flooding algorithms simply broadcast user traffic through a network ensuring that the destination will be reached because all paths to the destination are attempted. No routing tables or routing information exchange is required when using flooding algorithms, but flooding is very wasteful of network resources.

II. Routing Costs

Routing algorithms consume network resources in determining routes through a network to the required destination. In addition to overheads such as processing time for algorithm execution and memory to store routing tables, there is the consumption of transmission capacity that could otherwise be utilised for user traffic. Typically, 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.

Shortest path algorithms maintaining routing look up tables in each node contribute zero transmission cost when making a routing decision. However, they do consume resources every time the network changes by bringing all routing tables throughout the network up to date. If routing decisions are based on inaccurate information stored in routing tables, long delays or lost traffic can lead to retransmissions or reattempts caused by higher layer end-to-end protocols.

Flooding based routing procedures do not maintain routing tables and therefore do not require any table update procedures. As such they contribute zero update cost but do consume large amounts of network capacity in search of destinations each time a routing decision is required. Overheads resulting from routing decisions are generally independent from the frequency of changes occurring in networks.

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. This is indeed true if the shortest path algorithm is unable to maintain accurate routing tables due to high rates of change in link cost and network topology.

In considering the overall routing costs associated with routing procedures such as these, it is possible to identify con-

0-7803-3925-8/97 $10.00 ©1997 IEEE

ditions where frequency of change within a network requires the use of one particular algorithm over another. Furthermore, closely examining routing algorithm performance under dynamic conditions highlights the more dominant causes of routing cost. Strategies can then be conceived to improve performance by addressing these more dominating effects.

III. Routing Performance

Routing algorithm performance alone can be expressed in terms of convergence speed and routing table update message overhead. Further insight into overall routing procedure performance can be achieved by incorporating effects such as dynamic topologies, high network loads and end user retransmissions. Due to the difficulty in analysing adaptive distributed routing procedures [8, p. 318], [9, p. 211] and the desire to also include these additional conditions, computer simulation modelling has been used for the evaluation of routing strategies in this paper.

Transmission bearer capacity limits the total available resources within a network resulting in a threshold to average delay performance as network loads increase. As the average rate of user traffic load (γ) entering a network increases towards the saturation load (γ^*), overall average delay (T) departs from a "no load" delay with $T \to \infty$ at a traffic load of γ^* [8, p. 323]. If saturation load is approached and congestion is occurring, flow control mechanisms must be employed to relieve the congestion. A more efficient routing procedure will enable an increase in a given network's saturation load.

Using saturation load to assess a routing algorithm's performance is somewhat unrealistic because a network cannot be operated with infinite average delay. A more realistic measure is the maximum operating point, denoted γ', which is the maximum user traffic load entering the network before average delay departs from the "no load" delay and begins to grow rapidly. This is in effect just before the threshold "knee" of a delay-throughput curve. Maximum operating point γ' can be defined as the point where a line from the origin is tangent to the delay-throughput curve [10, p. 9]. Note that γ' as defined here is not true throughput but the maximum originating user traffic arrival rate. True throughput is required for the accurate comparison of routing procedure performance, and as described later this is derived from the simulation results.

A. Simulation model

An initial network model has been developed based on stationary stochastic processes evenly distributed across the network. This reduces bias and transient effects in simulation results. Consider the network model as a directed graph G with N nodes and L bidirectional links. Each node functions as a source of user traffic entering the network where traffic can be destined to all other nodes within the network. Routing algorithms are executed within these same nodes and update control messages are exchanged between neighbouring nodes for the purpose of maintaining routing tables. Routing is used to effectively provide a connectionless datagram service in forwarding user traffic through the network.

An adaptive distributed shortest path algorithm is selected for good performance when the network is static or quasi-static. This distributed algorithm offers higher survivability and if link cost functions reflect both connectivity and delay (node output queue lengths), minimum delay routing can be employed to enhance performance. However, incorrect congestion information leading to the choice of inferior routes, typically, has less serious consequences than incorrect topological information resulting in the possible choice of non-existent routes [11, p. 342]. A minimum hop shortest path algorithm has therefore been adopted where link costs are set to a constant value simply representing topology connectivity. The minimum hop algorithm reduces complexity in this investigation while still concentrating on the important issue of topological information update.

Routing algorithms are executed in each of the N nodes in G. Distance and routing tables are assumed initially configured with all known destinations and neighbouring nodes by some higher level topology control entity. In the actual simulation this occurs during initialisation.

Since the routing procedures are triggered by link cost change events, G has essentially fixed topology with links always existing between nodes and their respective neighbours. Topology change is modelled by having link quality changing between one of two possible states; a link is either up providing good error free transmission, or a link has failed providing no useful transmission at all. This simplified approach requires no topology control as a particular node's neighbours at any given time always belong to the same subset of nodes as originally defined in G.

The minimum hop routing procedure has been developed from the Jaffe-Moss distributed shortest path algorithm [6]. This algorithm has the advantage that it exhibits rapid convergence and is loop free. In addition, this algorithm can be used for more general least cost routing. To execute the algorithm as minimum hop, link costs are set to either the value 1 (link present) or ∞ (link failed). The Jaffe-Moss algorithm consists of two table update procedures. A link cost decrease causes an Independent Update Procedure (IUP) based on a decentralised version of Ford-Fulkerson's algorithm. Jaffe and Moss determined that if link costs decrease then algorithms such as Ford and Fulkerson's are inherently loop free. If a link cost increases, it is possible that loops may occur while the algorithm converges. To prevent the formation of loops a Coordinated Update Procedure (CUP) is used whenever link costs increase.

Routing table update messages used for the Independent and Coordinated Update Procedures are set at a fixed size of 8 octets (64 bits); this is considered sufficient to carry required information such as message type, source address, destination address, routing costs and other quality of service parameters. In the case of multiple new minimum costs occurring in distance tables, a random selection is made of nodes with the same distance to avoid any deterministic bias in the selection of a new next node.

Flood search routing has been selected for its robustness in dynamic networks and is modelled as constrained flooding, the most efficient way to flood an entire network [12]. 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. Constrained flooding uniquely identifies packets associated with a particular flood search by using sequence numbering. Nodes store sequence numbers of packets already flooded. If any packets revisit a node with the same sequence number, they are discarded instead of being further broadcast to neighbours. This technique ensures that all nodes are visited at least once and duplicated traffic is kept to a minimum throughout the network.

A 64 node network with connectivity of degree 4 is modelled 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. Transmission links function as a single server queueing system with service rate defined by packet size and link capacity. Link capacity is fixed at 16kbps to represent a narrowband radio link or perhaps a logical signalling channel. Infinite length queues are modelled to ensure that packet discards result solely from routing table uncertainty, loop formation, and link failure.

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, $\gamma = \sum_{n=1}^{N} \gamma_n$, is evenly distributed across all N nodes. User packet length is fixed and set to 128 octets (1024 bits). This is based on the default X.25 Packet Layer Protocol packet size [13, p. 354]. X.25 is typically used in wide area networks and modified versions have been proposed for packet radio networks [14], [15]. User packet size has been set larger than routing table update packet size to realistically represent data packets while effectively capturing the overheads caused by packet duplication when flooding.

Discarding of user packets due to routing table uncertainty creates a problem because the probability of successfully reaching a destination must now be incorporated into the analysis of results. Generally, in real systems lost traffic leads to reattempts or retransmissions by higher layer protocols. These retransmissions will inevitably place additional load on the network and be reflected as a transmission overhead cost. To include this behaviour and provide a more realistic model, a selective repeat Automatic Repeat reQuest (ARQ) protocol is modelled on an end-to-end basis between source-destination pairs with a retransmission timeout fixed at five seconds.

Destination nodes notify the source that a user packet has successfully been received by returning an acknowledgment packet. On receiving an acknowledgment, the source node cancels the retransmission timer for that outstanding user packet. A transport layer packet identifier is required for source nodes to uniquely identify originated packets being acknowledged. Transport layer retransmission acknowledgment packets are set to a packet size of 8 octets. This is to reflect the control functionality of this packet type as distinct to packets carrying user information. In a real system this information may be piggybacked on return user packets. The constrained flooding model does not utilise this acknowledgment procedure because of the high probability of successful transmission using this scheme.

Changes in network topology are evenly distributed across all links. The amount of change occurring in the network at any given time is denoted intensity of change, $\mathcal{I}$, and defined as the average number of link cost changes per second, per link. To reflect some stochastic behaviour in the simulation model, link cost change events are scheduled with event times derived from two negative exponential probability distributions. These distributions are characterised by two different mean interarrival times; one representing the mean interarrival time between link failures in the network and the other, average link down time. If a link failure event is scheduled to occur, a link (i,j) is randomly selected from the L possible links in G, using a uniform distribution. If the current state of the chosen link (i,j) is operational, then the link is failed. If the link (i,j) is already in a failed state, an alternative link is randomly selected repeating the process until an operational link is found. Both nodes i and j adjacent to the link are notified of the change in link cost. The link (i,j) is then placed in a failed state, queues are emptied and the link's restoration time is derived from the link restoration probability distribution and scheduled on a simulation event list.

The ratio of mean link failure interarrival time to mean link down time is regulated to ensure that on average only a given number of links are failed at any time. For the network simulation model, the average proportion of links failed at any time is 5%. This represents reasonable change to network topology without causing significant network partitioning. While the routing procedures can function in partitioned networks, gaining insight into the algorithm's behaviour is difficult. A link cost change event triggers routing procedures in nodes i and j to act upon the cost change for all affected destinations. 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. When a link fails or recovers, the cost change simultaneously affects both directions on a single bidirectional link.

B. Simulation results

Performance of the constrained flooding and minimum hop algorithms using a static network model ($\mathcal{I} = 0$) yielded the results shown in Figure 1. Overall average delay experienced by all user packets is presented as a function of the scaled originating user traffic arrival rate per node. Each point on the curve represents average results from 5 simulation runs using different random number seeds. Confidence intervals of 95% are calculated from the t distribution. Validation was achieved by thorough testing of algorithm execution, accuracy of routing tables and packet tracing. Further verification of the simulation model was obtained by comparison of simulation results with the results derived from an M/M/1 queueing model. For this comparison user packet size was derived from a negative exponential probability distribution and the selective repeat ARQ protocol was disabled. These

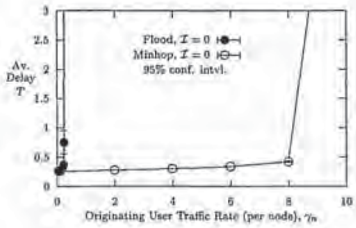


Fig. 1. Flooding and minimum hop in static network

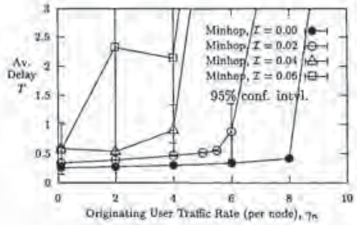


Fig. 2. Minimum hop with increasing change intensity

verification test results have not been reproduced here.

Duplicate packet overheads are the cause of reduced maximum operating point in constrained flooding. The high routing cost associated with flooding is due to searching the entire network every time a routing decision is required. The minimum hop algorithm requires no network resources for routing table updates in a static network, $\mathcal{I} = 0$. This results in a significantly higher maximum operating point. Characteristic threshold behaviour of both curves in Figure 1 is consistent with the behaviour of a network of M/M/1 queues.

Consider now performance of minimum hop routing when operating in a dynamic network environment. Figure 2 illustrates delay vs. throughput performance for intensity of change $\mathcal{I} = 0.0, 0.02, 0.04$ and 0.06. Performance degrades as the intensity of change increases. More network resources are consumed for maintaining the routing tables while changes in topology are being tracked. As routing table update costs increase with higher intensities of change, the network capacity available for user traffic is reduced.

If intensity of change is increased to $\mathcal{I} = 0.06$, the minimum hop algorithm no longer exhibits the expected threshold behaviour, see Figure 2. The increase in average delay and high variance in output statistic samples are due to the high rate of instances where next node entries are unknown, causing packet retransmissions. On average, many user packets are requiring retransmissions in order to successfully reach their respective destinations. The retransmission timeout delay is significantly affecting the average delay performance such that overall performance is being dictated by the performance of a higher layer "transport" protocol. The routing procedure is becoming ineffective under such dynamic conditions.

For constrained flooding, the maximum operating point remained the same, with $\gamma'_n = 0.2$, as $\mathcal{I}$ increased. As expected, constrained flooding is inefficient but independent from changes in network topology. 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. Unfortunately, flooding results in very low network utilisation and therefore better routing strategies need to be sought.

IV. Hybrid Routing

Many routing algorithms operate efficiently only for one class of networks (either static or very dynamic). The task of selecting a most appropriate routing algorithm can be very difficult because the dynamic nature of a given network can vary. It is possible to achieve improved performance over a wider range of network conditions by combining different routing procedures into a new hybrid routing procedure.

Somewhat similar in concept is a proposal to support user mobility in deployed circuit switched trunk networks [16]. With this scheme a new call request packet is forwarded using look up tables to a node servicing the destination user. If the destination user is no longer affiliated with that node, a flood search is initiated. Problems arise though with such an approach when networks exhibit dynamic topologies [17].

In observing the minimum hop routing procedure, it has been established that operation of the Jaffe-Moss algorithm in very dynamic networks is limited significantly by routing table uncertainty. While table update protocol overheads are substantial, they are less dominant. This observation opens an opportunity for significant improvements to the algorithm's performance by overcoming the problem of uncertainty in routing tables. As we will see from the results obtained, the proposed solution (simple local broadcast) leads to a routing procedure that exhibits a smooth change to flooding as intensity of change in the network increases.

A. Hybrid routing model

A hybrid routing strategy is now proposed where a local broadcast, derived from constrained flooding, is initiated if routing tables cannot provide a next node entry for forwarding user traffic. Under normal circumstances the Jaffe-Moss algorithm provides a next node entry for routing decisions. If this entry is unknown due to link failure, user traffic is broadcast on all outgoing links (except the incoming link). Subsequent nodes further downtree towards the destination may use minimum hop routing tables if a valid next node entry exists. All nodes remain involved in table update procedures.

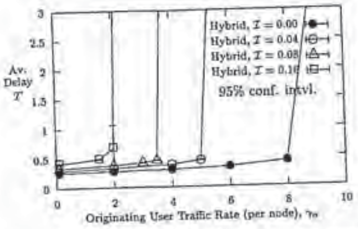


Fig. 3. Local broadcast hybrid routing with increasing change intensity

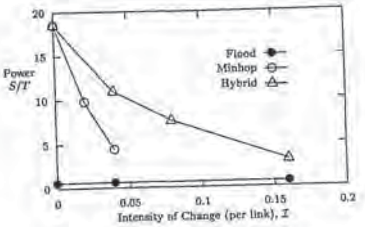


Fig. 4. Comparative performance over varying change intensity

Once a node initiating a CUP has been acknowledged and a valid next node determined, local broadcasting is no longer used. It is worth observing that the routing procedure defined here automatically adapts to changes in the network. Flooding occurs only for the parts of the network where routing tables are uncertain and only for the periods of time when the uncertainty persists.

Sequence numbering of packets is required in hybrid routing to manage the discarding of duplicated packets, as was done with constrained flooding. This use of sequence numbering also ensures that any packet reaching a destination does so along a loop free path because any packet with the same sequence number visiting a node more than once is discarded. While packets have a greater chance of reaching the destination due to multiple packets travelling along different paths, there is no guarantee that at least one packet will reach the destination. This is due to possible looping resulting in all packets being discarded. However, probability of reaching the destination is very high.

B. Performance of hybrid routing

Figure 3 demonstrates performance of the hybrid routing procedure for various intensities of change. Even with very high levels of change intensity, the hybrid scheme still exhibits threshold type behaviour. Throughput decreases as the intensity of change increases though due to the routing table update traffic and the flooded user traffic. Overall the effect of hybrid routing is a smooth transition from a shortest path routing procedure in static and quasi-static conditions towards a flooding procedure as the network becomes more dynamic. The efficiency of this scheme offers very significant improvements of throughput as compared to pure constrained flooding.

In using hybrid routing, network capacity is consumed by routing table update traffic and duplicate user and acknowledgment packets caused by local broadcasting. These overheads reduce available throughput for user traffic and represent the routing cost of this routing procedure. With minimum hop routing, overheads are dominated by routing table update traffic and user traffic retransmissions. Given that update traffic load will be the same for hybrid routing as it is for minimum hop routing, due to identical network and change conditions, it can be concluded that local broadcasting overheads are less than minimum hop retransmission overheads, resulting in the lower routing costs.

Comparing overall performance of the three routing procedures requires assessment of maximum operating points under varying dynamic conditions. True maximum operating point is a function of throughput (S) and delay (T), with the actual throughput calculated using Equ. (1). Maximum average throughput (scaled on a per node basis) is defined as the actual amount of traffic successfully delivered to destination nodes when the network is operating at its maximum operating point for a given intensity of change. Values for probability of successful transmission for node n, p_n, are measured by monitoring the proportion of packets received to those attempted to be sent. Routing power is defined as S/T and used to reflect the maximum operating point as a single parameter [10, p. 8].

$$S = \frac{\sum_{n=1}^{N} \gamma'_n p_n}{N} \qquad (1)$$

Results of routing power vs. intensity of change for constrained flooding, minimum hop and hybrid routing are plotted in Figure 4. Flooding demonstrates poor efficiency but is robust in dynamic topologies. The relative inefficiency of flooding can be attributed to the ratio of user packet to control packet size of 16:1. Minimum hop routing degrades until it essentially fails as indicated in Figure 2. Hybrid routing clearly offers the highest maximum operating throughput over the range of static through to dynamic network conditions. Operation with high throughput in very dynamic conditions provides an indication of the algorithm's robust nature.

Desirable properties of the Jaffe-Moss shortest path algorithm and constrained flooding are preserved in this hybrid routing procedure. Rapid convergence, minimal update traffic load and loop free routes to the destination are guaranteed through the use of sequence numbering, while local broadcasting also provides a high assurance of successful transmis sion under very dynamic conditions. Disadvantages include the extra complexity and storage required within network nodes to manage sequence numbering of user packets. Also, increased packet overhead to contain the sequence numbers may be a disadvantage. These disadvantages, however, are outweighed by the improved throughput/delay performance and adaptive properties of the algorithm.

V. Conclusions

In closely investigating the performance of minimum hop routing, it was found that routing table update traffic contributes only a proportion of total overhead traffic loads. A more dominant factor in degrading network performance is caused by user end-to-end retransmissions. As rates of link cost change within the network increase, so does the extent of routing table uncertainty. This leads to high levels of lost user traffic and reflects the algorithm's inability to track high rates of topology change.

Hybrid routing exploits flooding characteristics using a local broadcasting strategy to address the routing table uncertainty problem and provides improved overall performance. The focus of this strategy is to incorporate complimentary advantageous features of generalised least cost and flooding type algorithms into a more flexible routing strategy capable of adapting to changing operational conditions. The resulting hybrid scheme is relatively simple, robust and very efficient in dynamic networking environments.

References

[1] W-K. Chen, *Theory Of Nets: Flows In Networks*, John Wiley and Sons, 1990.

[2] J.M. McQuillan, I. Richer, and E.C. Rosen, "The new routing algorithm for the ARPANET," *IEEE Transactions on Communications*, vol. COM-28, no. 5, pp. 711–719, May 1980.

[3] M. Schwartz, *Telecommunication Networks: Protocols, Modeling and Analysis*, Addison-Wesley, 1988.

[4] L. Ford and D. Fulkerson, *Flows in Networks*, Princeton University Press, 1962.

[5] P.M. Merlin and A. Segall, "A fail safe distributed routing protocol," *IEEE Transactions on Communications*, vol. COM-27, no. 9, pp. 1280–1287, September 1979.

[6] J.M. Jaffe and F.H. Moss, "A responsive distributed routing algorithm for computer networks," *IEEE Transactions on Communications*, vol. COM-30, no. 7, pp. 1758–1762, July 1982.

[7] J.J. Garcia-Luna-Aceves, "Loop free routing using diffusing computations," *IEEE/ACM Transactions on Networking*, vol. 1, no. 1, pp. 130–141, February 1993.

[8] L. Kleinrock, *Queueing Systems Volume II: Computer Applications*, John Wiley and Sons, 1976.

[9] R.F. Garzia and M.R. Garzia, *Network Modeling, Simulation, and Analysis*, Marcel Dekker, Inc., 1990.

[10] L. Kleinrock, "Performance evaluation of distributed computer-communication systems," in *Queueing Theory and its Applications*, O.J. Boxma and R. Syski, Eds., pp. 1–57. Elsevier Science Publishers B.V. (North-Holland), 1988.

[11] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 1987.

[12] T.V. Vu and S.P. Risner, "Performance and cost of broadcast routing algorithms in the strategic defense system terrestrial network," in *IEEE MILCOM*, 1991, pp. 29.2.1–29.2.5.

[13] A.S. Tanenbaum, *Computer Networks*, Prentice-Hall International, second edition, 1989.

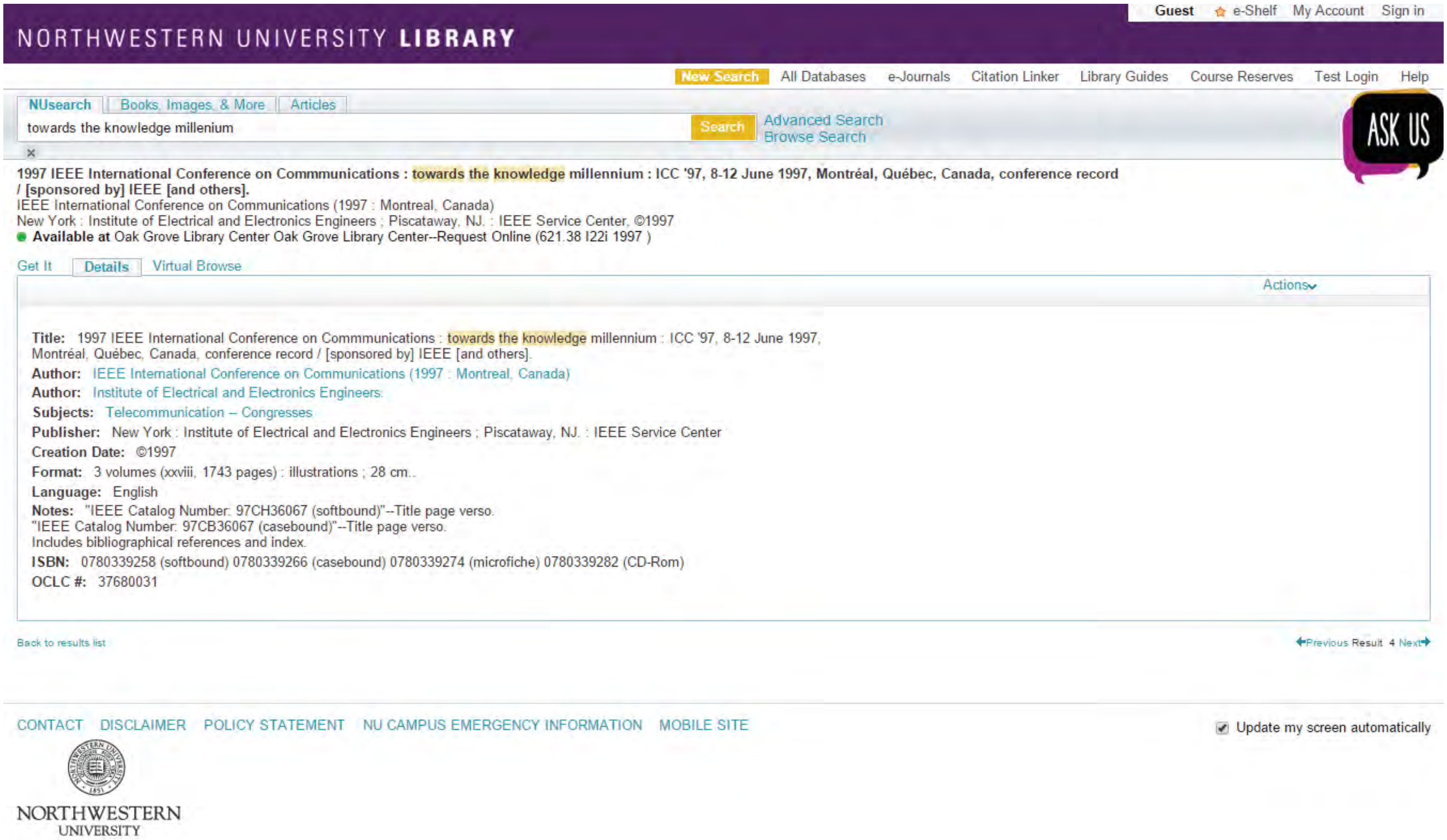
[14] B.H. Davies, S.J. Swain, and K. Watkins, "Efficient implementation of X.25 on narrowband packet radio systems," in *IEE Colloquium on "Military Communications - The Trend Towards Civil Standards", (Digest No. 55)*, April 1990, pp. 6/1–6/5.

[15] P.R. Karn, H.E. Price, and R.J. Diersing, "Packet radio in the amateur service," *IEEE Journal on Selected Areas in Communications*, vol. 3, no. 3, pp. 431–439, May 1985.

[16] V.O.K. Li and R-F Chang, "Proposed routing algorithms for the U.S. Army mobile subscriber equipment (MSE) network," in *IEEE MILCOM*, 1986, pp. 39.4.1–39.4.7.

[17] L.E. Miller, R.H. French, J.S. Lee, and D.J. Torrieri, "MSE routing algorithm comparison," in *IEEE MILCOM*, 1989, pp. 2.2.1–2.2.5.

Attachment 1b: Northwestern University Library catalog record for the proceedings of 1997 IEEE International Conference on Communications



towards the knowledge millenium

1997 IEEE International Conference on Commmunications : towards the knowledge millennium : ICC '97, 8-12 June 1997, Montréal, Québec, Canada, conference record / [sponsored by] IEEE [and others].
IEEE International Conference on Communications (1997 : Montreal, Canada)
New York : Institute of Electrical and Electronics Engineers ; Piscataway, NJ. : IEEE Service Center, ©1997
● **Available at** Oak Grove Library Center Oak Grove Library Center--Request Online (621.38 I22i 1997)

Title: 1997 IEEE International Conference on Commmunications : towards the knowledge millennium : ICC '97, 8-12 June 1997, Montréal, Québec, Canada, conference record / [sponsored by] IEEE [and others].

Author: IEEE International Conference on Communications (1997 : Montreal, Canada)

Author: Institute of Electrical and Electronics Engineers.

Subjects: Telecommunication -- Congresses

Publisher: New York : Institute of Electrical and Electronics Engineers ; Piscataway, NJ. : IEEE Service Center

Creation Date: ©1997

Format: 3 volumes (xxviii, 1743 pages) : illustrations ; 28 cm..

Language: English

Notes: "IEEE Catalog Number: 97CH36067 (softbound)"--Title page verso.
"IEEE Catalog Number: 97CB36067 (casebound)"--Title page verso.
Includes bibliographical references and index.

ISBN: 0780339258 (softbound) 0780339266 (casebound) 0780339274 (microfiche) 0780339282 (CD-Rom)

OCLC #: 37680031

Back to results list

Previous Result 4 Next

CONTACT DISCLAIMER POLICY STATEMENT NU CAMPUS EMERGENCY INFORMATION MOBILE SITE

☑ Update my screen automatically

NORTHWESTERN
UNIVERSITY

Hybrid Routing in Dynamic Networks

Peter Shoubridge
Communications Division, Defence Science Technology Organisation
Salisbury, South Australia, 5108

Arek Dadej
Institute of Telecommunications Research, University of South Australia
The Levels, South Australia, 5095

Abstract— **In mobile radio communication networks the distribution of traffic loads and network topologies may vary from nearly static to very dynamic. This dynamic behaviour may vary both in space and in time. Since routing algorithms tend to be well suited to specific networking environments, it is very difficult to select a single routing algorithm that is most appropriate for a given network, if the network is subjected to varying degrees of dynamic behaviour. This paper proposes a routing strategy that smoothly adapts to changing network conditions by combining two distinct routing principles into a single hybrid routing procedure. The hybrid routing strategy exhibits a smooth change from shortest path routing to constrained flooding, as the behaviour of the network (or regions within) changes from quasi-static to very dynamic.**

I. Introduction

In addition to successfully forwarding user traffic between source and destination nodes in a communications network, the routing function generally seeks to optimise some performance criteria such as network utilisation or throughput, or perhaps average delay. 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.

For efficient routing, nodes require knowledge of network topology. Links may fail or become congested, nodes or users could be mobile, resulting in changes to source-destination paths. Information regarding these changes must be conveyed to network nodes so that appropriate routing can take place. This information exchange consumes network resources and presents a cost in maintaining up to date routing information in network nodes. 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. Inaccurate routing can result in lost or delayed traffic causing end-to-end retransmissions which further load the network. This additional load is itself an unwanted overhead which could be avoided with a better routing strategy.

Algorithms that can determine shortest paths between source and destination nodes are typically used in static or quasi-static networks. They require periods of network stability so routing tables can settle to reflect true shortest paths. The most efficient algorithm is that described by Dijkstra which requires complete knowledge of the network topology [1, p. 103]. This algorithm can be executed centrally or topology information can be broadcast to all nodes in a decentralised implementation [2]. Another algorithm, suited to distributed architectures, is a decentralised version of Ford and Fulkerson's algorithm which only requires information exchange between immediate neighbouring nodes [3, p. 268], [4, p. 130]. More recent algorithms derived from these offer significant improvements in performance [5], [6], [7].

Routing by flooding is often used in very dynamic networks where destination nodes are highly mobile or topological changes occur very frequently and therefore the network connectivity is uncertain. Flooding algorithms simply broadcast user traffic through a network ensuring that the destination will be reached because all paths to the destination are attempted. No routing tables or routing information exchange is required when using flooding algorithms, but flooding is very wasteful of network resources.

II. Routing Costs

Routing algorithms consume network resources in determining routes through a network to the required destination. In addition to overheads such as processing time for algorithm execution and memory to store routing tables, there is the consumption of transmission capacity that could otherwise be utilised for user traffic. Typically, 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.

Shortest path algorithms maintaining routing look up tables in each node contribute zero transmission cost when making a routing decision. However, they do consume resources every time the network changes by bringing all routing tables throughout the network up to date. If routing decisions are based on inaccurate information stored in routing tables, long delays or lost traffic can lead to retransmissions or reattempts caused by higher layer end-to-end protocols.

Flooding based routing procedures do not maintain routing tables and therefore do not require any table update procedures. As such they contribute zero update cost but do consume large amounts of network capacity in search of destinations each time a routing decision is required. Overheads resulting from routing decisions are generally independent from the frequency of changes occurring in networks.

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. This is indeed true if the shortest path algorithm is unable to maintain accurate routing tables due to high rates of change in link cost and network topology.

In considering the overall routing costs associated with routing procedures such as these, it is possible to identify con-

0-7803-3925-8/97 $10.00 ©1997 IEEE

ditions where frequency of change within a network requires the use of one particular algorithm over another. Furthermore, closely examining routing algorithm performance under dynamic conditions highlights the more dominant causes of routing cost. Strategies can then be conceived to improve performance by addressing these more dominating effects.

III. Routing Performance

Routing algorithm performance alone can be expressed in terms of convergence speed and routing table update message overhead. Further insight into overall routing procedure performance can be achieved by incorporating effects such as dynamic topologies, high network loads and end user retransmissions. Due to the difficulty in analysing adaptive distributed routing procedures [8, p. 318], [9, p. 211] and the desire to also include these additional conditions, computer simulation modelling has been used for the evaluation of routing strategies in this paper.

Transmission bearer capacity limits the total available resources within a network resulting in a threshold to average delay performance as network loads increase. As the average rate of user traffic load (γ) entering a network increases towards the saturation load (γ^*), overall average delay (T) departs from a "no load" delay with $T \rightarrow \infty$ at a traffic load of γ^* [8, p. 323]. If saturation load is approached and congestion is occurring, flow control mechanisms must be employed to relieve the congestion. A more efficient routing procedure will enable an increase in a given network's saturation load.

Using saturation load to assess a routing algorithm's performance is somewhat unrealistic because a network cannot be operated with infinite average delay. A more realistic measure is the maximum operating point, denoted γ', which is the maximum user traffic load entering the network before average delay departs from the "no load" delay and begins to grow rapidly. This is in effect just before the threshold "knee" of a delay-throughput curve. Maximum operating point γ' can be defined as the point where a line from the origin is tangent to the delay-throughput curve [10, p. 9]. Note that γ' as defined here is not true throughput but the maximum originating user traffic arrival rate. True throughput is required for the accurate comparison of routing procedure performance, and as described later this is derived from the simulation results.

A. Simulation model

An initial network model has been developed based on stationary stochastic processes evenly distributed across the network. This reduces bias and transient effects in simulation results. Consider the network model as a directed graph G with N nodes and L bidirectional links. Each node functions as a source of user traffic entering the network where traffic can be destined to all other nodes within the network. Routing algorithms are executed within these same nodes and update control messages are exchanged between neighbouring nodes for the purpose of maintaining routing tables. Routing is used to effectively provide a connectionless datagram service in forwarding user traffic through the network.

An adaptive distributed shortest path algorithm is selected for good performance when the network is static or quasi-static. This distributed algorithm offers higher survivability and if link cost functions reflect both connectivity and delay (node output queue lengths), minimum delay routing can be employed to enhance performance. However, incorrect congestion information leading to the choice of inferior routes, typically, has less serious consequences than incorrect topological information resulting in the possible choice of non-existent routes [11, p. 342]. A minimum hop shortest path algorithm has therefore been adopted where link costs are set to a constant value simply representing topology connectivity. The minimum hop algorithm reduces complexity in this investigation while still concentrating on the important issue of topological information update.

Routing algorithms are executed in each of the N nodes in G. Distance and routing tables are assumed initially configured with all known destinations and neighbouring nodes by some higher level topology control entity. In the actual simulation this occurs during initialisation.

Since the routing procedures are triggered by link cost change events, G has essentially fixed topology with links always existing between nodes and their respective neighbours. Topology change is modelled by having link quality changing between one of two possible states; a link is either up providing good error free transmission, or a link has failed providing no useful transmission at all. This simplified approach requires no topology control as a particular node's neighbours at any given time always belong to the same subset of nodes as originally defined in G.

The minimum hop routing procedure has been developed from the Jaffe-Moss distributed shortest path algorithm [6]. This algorithm has the advantage that it exhibits rapid convergence and is loop free. In addition, this algorithm can be used for more general least cost routing. To execute the algorithm as minimum hop, link costs are set to either the value 1 (link present) or ∞ (link failed). The Jaffe-Moss algorithm consists of two table update procedures. A link cost decrease causes an Independent Update Procedure (IUP) based on a decentralised version of Ford-Fulkerson's algorithm. Jaffe and Moss determined that if link costs decrease then algorithms such as Ford and Fulkerson's are inherently loop free. If a link cost increases, it is possible that loops may occur while the algorithm converges. To prevent the formation of loops a Coordinated Update Procedure (CUP) is used whenever link costs increase.

Routing table update messages used for the Independent and Coordinated Update Procedures are set at a fixed size of 8 octets (64 bits); this is considered sufficient to carry required information such as message type, source address, destination address, routing costs and other quality of service parameters. In the case of multiple new minimum costs occurring in distance tables, a random selection is made of nodes with the same distance to avoid any deterministic bias in the selection of a new next node.

Flood search routing has been selected for its robustness in dynamic networks and is modelled as constrained flooding, the most efficient way to flood an entire network [12]. 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. Constrained flooding uniquely identifies packets associated with a particular flood search by using sequence numbering. Nodes store sequence numbers of packets already flooded. If any packets revisit a node with the same sequence number, they are discarded instead of being further broadcast to neighbours. This technique ensures that all nodes are visited at least once and duplicated traffic is kept to a minimum throughout the network.

A 64 node network with connectivity of degree 4 is modelled 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. Transmission links function as a single server queueing system with service rate defined by packet size and link capacity. Link capacity is fixed at 16kbps to represent a narrowband radio link or perhaps a logical signalling channel. Infinite length queues are modelled to ensure that packet discards result solely from routing table uncertainty, loop formation, and link failure.

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, $\gamma = \sum_{n=1}^{N} \gamma_n$, is evenly distributed across all N nodes. User packet length is fixed and set to 128 octets (1024 bits). This is based on the default X.25 Packet Layer Protocol packet size [13, p. 354]. X.25 is typically used in wide area networks and modified versions have been proposed for packet radio networks [14], [15]. User packet size has been set larger than routing table update packet size to realistically represent data packets while effectively capturing the overheads caused by packet duplication when flooding.

Discarding of user packets due to routing table uncertainty creates a problem because the probability of successfully reaching a destination must now be incorporated into the analysis of results. Generally, in real systems lost traffic leads to reattempts or retransmissions by higher layer protocols. These retransmissions will inevitably place additional load on the network and be reflected as a transmission overhead cost. To include this behaviour and provide a more realistic model, a selective repeat Automatic Repeat reQuest (ARQ) protocol is modelled on an end-to-end basis between source-destination pairs with a retransmission timeout fixed at five seconds.

Destination nodes notify the source that a user packet has successfully been received by returning an acknowledgment packet. On receiving an acknowledgment, the source node cancels the retransmission timer for that outstanding user packet. A transport layer packet identifier is required for source nodes to uniquely identify originated packets being acknowledged. Transport layer retransmission acknowledgment packets are set to a packet size of 8 octets. This is to reflect the control functionality of this packet type as distinct to packets carrying user information. In a real system this information may be piggybacked on return user packets. The constrained flooding model does not utilise this acknowledgment procedure because of the high probability of successful transmission using this scheme.

Changes in network topology are evenly distributed across all links. The amount of change occurring in the network at any given time is denoted intensity of change, $\mathcal{I}$, and defined as the average number of link cost changes per second, per link. To reflect some stochastic behaviour in the simulation model, link cost change events are scheduled with event times derived from two negative exponential probability distributions. These distributions are characterised by two different mean interarrival times; one representing the mean interarrival time between link failures in the network and the other, average link down time. If a link failure event is scheduled to occur, a link (i, j) is randomly selected from the L possible links in G, using a uniform distribution. If the current state of the chosen link (i, j) is operational, then the link is failed. If the link (i, j) is already in a failed state, an alternative link is randomly selected repeating the process until an operational link is found. Both nodes i and j adjacent to the link are notified of the change in link cost. The link (i, j) is then placed in a failed state, queues are emptied and the link's restoration time is derived from the link restoration probability distribution and scheduled on a simulation event list.

The ratio of mean link failure interarrival time to mean link down time is regulated to ensure that on average only a given number of links are failed at any time. For the network simulation model, the average proportion of links failed at any time is 5%. This represents reasonable change to network topology without causing significant network partitioning. While the routing procedures can function in partitioned networks, gaining insight into the algorithm's behaviour is difficult. A link cost change event triggers routing procedures in nodes i and j to act upon the cost change for all affected destinations. 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. When a link fails or recovers, the cost change simultaneously affects both directions on a single bidirectional link.

B. Simulation results

Performance of the constrained flooding and minimum hop algorithms using a static network model ($\mathcal{I} = 0$) yielded the results shown in Figure 1. Overall average delay experienced by all user packets is presented as a function of the scaled originating user traffic arrival rate per node. Each point on the curve represents average results from 5 simulation runs using different random number seeds. Confidence intervals of 95% are calculated from the t distribution. Validation was achieved by thorough testing of algorithm execution, accuracy of routing tables and packet tracing. Further verification of the simulation model was obtained by comparison of simulation results with the results derived from an M/M/1 queueing model. For this comparison user packet size was derived from a negative exponential probability distribution and the selective repeat ARQ protocol was disabled. These

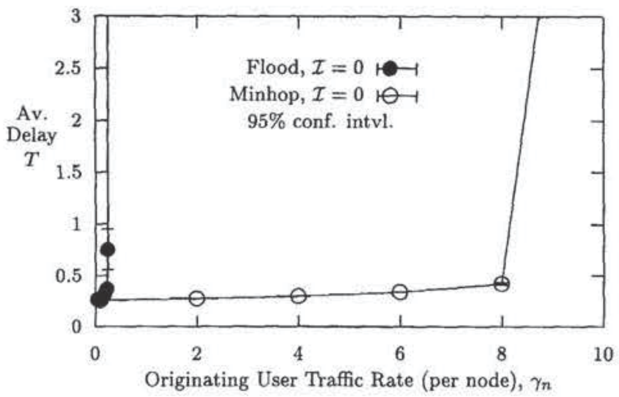


Fig. 1. Flooding and minimum hop in static network

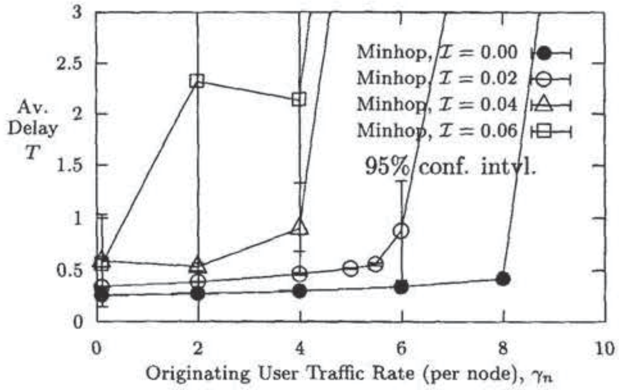


Fig. 2. Minimum hop with increasing change intensity

verification test results have not been reproduced here.

Duplicate packet overheads are the cause of reduced maximum operating point in constrained flooding. The high routing cost associated with flooding is due to searching the entire network every time a routing decision is required. The minimum hop algorithm requires no network resources for routing table updates in a static network, $\mathcal{I} = 0$. This results in a significantly higher maximum operating point. Characteristic threshold behaviour of both curves in Figure 1 is consistent with the behaviour of a network of M/M/1 queues.

Consider now performance of minimum hop routing when operating in a dynamic network environment. Figure 2 illustrates delay vs. throughput performance for intensity of change $\mathcal{I} = 0.0, 0.02, 0.04$ and 0.06. Performance degrades as the intensity of change increases. More network resources are consumed for maintaining the routing tables while changes in topology are being tracked. As routing table update costs increase with higher intensities of change, the network capacity available for user traffic is reduced.

If intensity of change is increased to $\mathcal{I} = 0.06$, the minimum hop algorithm no longer exhibits the expected threshold behaviour, see Figure 2. The increase in average delay and high variance in output statistic samples are due to the high rate of instances where next node entries are unknown, causing packet retransmissions. On average, many user packets are requiring retransmissions in order to successfully reach their respective destinations. The retransmission timeout delay is significantly affecting the average delay performance such that overall performance is being dictated by the performance of a higher layer "transport" protocol. The routing procedure is becoming ineffective under such dynamic conditions.

For constrained flooding, the maximum operating point remained the same, with $\gamma_n' = 0.2$, as $\mathcal{I}$ increased. As expected, constrained flooding is inefficient but independent from changes in network topology. 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. Unfortunately, flooding results in very low network utilisation and therefore better routing strategies need to be sought.

IV. Hybrid Routing

Many routing algorithms operate efficiently only for one class of networks (either static or very dynamic). The task of selecting a most appropriate routing algorithm can be very difficult because the dynamic nature of a given network can vary. It is possible to achieve improved performance over a wider range of network conditions by combining different routing procedures into a new hybrid routing procedure.

Somewhat similar in concept is a proposal to support user mobility in deployed circuit switched trunk networks [16]. With this scheme a new call request packet is forwarded using look up tables to a node servicing the destination user. If the destination user is no longer affiliated with that node, a flood search is initiated. Problems arise though with such an approach when networks exhibit dynamic topologies [17].

In observing the minimum hop routing procedure, it has been established that operation of the Jaffe-Moss algorithm in very dynamic networks is limited significantly by routing table uncertainty. While table update protocol overheads are substantial, they are less dominant. This observation opens an opportunity for significant improvements to the algorithm's performance by overcoming the problem of uncertainty in routing tables. As we will see from the results obtained, the proposed solution (simple local broadcast) leads to a routing procedure that exhibits a smooth change to flooding as intensity of change in the network increases.

A. Hybrid routing model

A hybrid routing strategy is now proposed where a local broadcast, derived from constrained flooding, is initiated if routing tables cannot provide a next node entry for forwarding user traffic. Under normal circumstances the Jaffe-Moss algorithm provides a next node entry for routing decisions. If this entry is unknown due to link failure, user traffic is broadcast on all outgoing links (except the incoming link). Subsequent nodes further downtree towards the destination may use minimum hop routing tables if a valid next node entry exists. All nodes remain involved in table update proce-

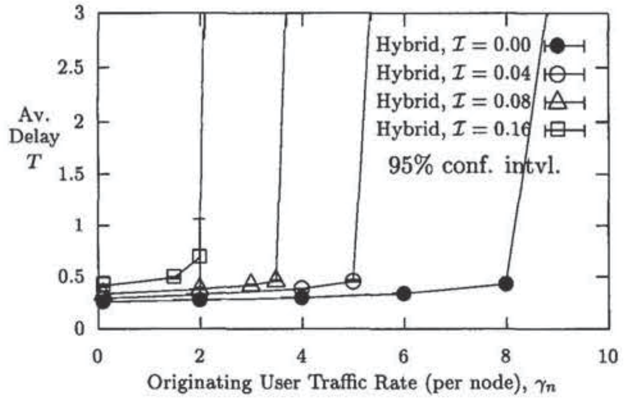


Fig. 3. Local broadcast hybrid routing with increasing change intensity

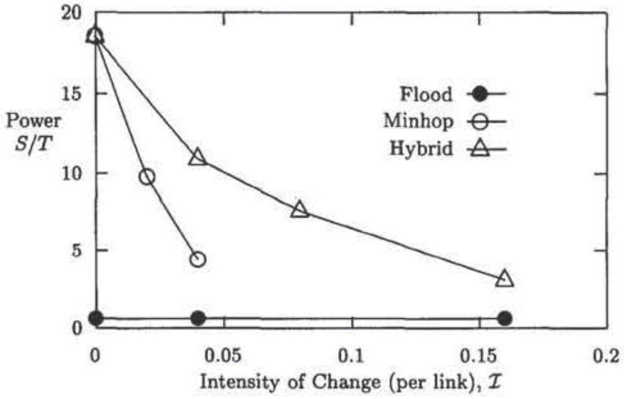


Fig. 4. Comparative performance over varying change intensity

dures. Once a node initiating a CUP has been acknowledged and a valid next node determined, local broadcasting is no longer used. It is worth observing that the routing procedure defined here automatically adapts to changes in the network. Flooding occurs only for the parts of the network where routing tables are uncertain and only for the periods of time when the uncertainty persists.

Sequence numbering of packets is required in hybrid routing to manage the discarding of duplicated packets, as was done with constrained flooding. This use of sequence numbering also ensures that any packet reaching a destination does so along a loop free path because any packet with the same sequence number visiting a node more than once is discarded. While packets have a greater chance of reaching the destination due to multiple packets travelling along different paths, there is no guarantee that at least one packet will reach the destination. This is due to possible looping resulting in all packets being discarded. However, probability of reaching the destination is very high.

B. Performance of hybrid routing

Figure 3 demonstrates performance of the hybrid routing procedure for various intensities of change. Even with very high levels of change intensity, the hybrid scheme still exhibits threshold type behaviour. Throughput decreases as the intensity of change increases though due to the routing table update traffic and the flooded user traffic. Overall the effect of hybrid routing is a smooth transition from a shortest path routing procedure in static and quasi-static conditions towards a flooding procedure as the network becomes more dynamic. The efficiency of this scheme offers very significant improvements of throughput as compared to pure constrained flooding.

In using hybrid routing, network capacity is consumed by routing table update traffic and duplicate user and acknowledgment packets caused by local broadcasting. These overheads reduce available throughput for user traffic and represent the routing cost of this routing procedure. With minimum hop routing, overheads are dominated by routing table update traffic and user traffic retransmissions. Given that update traffic load will be the same for hybrid routing as it is for minimum hop routing, due to identical network and change conditions, it can be concluded that local broadcasting overheads are less than minimum hop retransmission overheads, resulting in the lower routing costs.

Comparing overall performance of the three routing procedures requires assessment of maximum operating points under varying dynamic conditions. True maximum operating point is a function of throughput (S) and delay (T), with the actual throughput calculated using Equ. (1). Maximum average throughput (scaled on a per node basis) is defined as the actual amount of traffic successfully delivered to destination nodes when the network is operating at its maximum operating point for a given intensity of change. Values for probability of successful transmission for node n, p_n, are measured by monitoring the proportion of packets received to those attempted to be sent. Routing power is defined as S/T and used to reflect the maximum operating point as a single parameter [10, p. 8].

$$S = \frac{\sum_{n=1}^{N} \gamma'_n p_n}{N} \tag{1}$$

Results of routing power vs. intensity of change for constrained flooding, minimum hop and hybrid routing are plotted in Figure 4. Flooding demonstrates poor efficiency but is robust in dynamic topologies. The relative inefficiency of flooding can be attributed to the ratio of user packet to control packet size of 16:1. Minimum hop routing degrades until it essentially fails as indicated in Figure 2. Hybrid routing clearly offers the highest maximum operating throughput over the range of static through to dynamic network conditions. Operation with high throughput in very dynamic conditions provides an indication of the algorithm's robust nature.

Desirable properties of the Jaffe-Moss shortest path algorithm and constrained flooding are preserved in this hybrid routing procedure. Rapid convergence, minimal update traffic load and loop free routes to the destination are guaranteed through the use of sequence numbering, while local broadcasting also provides a high assurance of successful transmission under very dynamic conditions. Disadvantages include the extra complexity and storage required within network nodes to manage sequence numbering of user packets. Also, increased packet overhead to contain the sequence numbers may be a disadvantage. These disadvantages, however, are outweighed by the improved throughput/delay performance and adaptive properties of the algorithm.

V. Conclusions

In closely investigating the performance of minimum hop routing, it was found that routing table update traffic contributes only a proportion of total overhead traffic loads. A more dominant factor in degrading network performance is caused by user end-to-end retransmissions. As rates of link cost change within the network increase, so does the extent of routing table uncertainty. This leads to high levels of lost user traffic and reflects the algorithm's inability to track high rates of topology change.

Hybrid routing exploits flooding characteristics using a local broadcasting strategy to address the routing table uncertainty problem and provides improved overall performance. The focus of this strategy is to incorporate complimentary advantageous features of generalised least cost and flooding type algorithms into a more flexible routing strategy capable of adapting to changing operational conditions. The resulting hybrid scheme is relatively simple, robust and very efficient in dynamic networking environments.

References

[1] W-K. Chen, *Theory Of Nets: Flows In Networks*, John Wiley and Sons, 1990.

[2] J.M. McQuillan, I. Richer, and E.C. Rosen, "The new routing algorithm for the ARPANET," *IEEE Transactions on Communications*, vol. COM-28, no. 5, pp. 711–719, May 1980.

[3] M. Schwartz, *Telecommunication Networks: Protocols, Modeling and Analysis*, Addison-Wesley, 1988.

[4] L. Ford and D. Fulkerson, *Flows in Networks*, Princeton University Press, 1962.

[5] P.M. Merlin and A. Segall, "A fail safe distributed routing protocol," *IEEE Transactions on Communications*, vol. COM-27, no. 9, pp. 1280–1287, September 1979.

[6] J.M. Jaffe and F.H. Moss, "A responsive distributed routing algorithm for computer networks," *IEEE Transactions on Communications*, vol. COM-30, no. 7, pp. 1758–1762, July 1982.

[7] J.J. Garcia-Luna-Aceves, "Loop free routing using diffusing computations," *IEEE/ACM Transactions on Networking*, vol. 1, no. 1, pp. 130–141, February 1993.

[8] L. Kleinrock, *Queueing Systems Volume II: Computer Applications*, John Wiley and Sons, 1976.

[9] R.F. Garzia and M.R. Garzia, *Network Modeling, Simulation, and Analysis*, Marcel Dekker, Inc., 1990.

[10] L. Kleinrock, "Performance evaluation of distributed computer-communication systems," in *Queueing Theory and its Applications*, O.J. Boxma and R. Syski, Eds., pp. 1–57. Elsevier Science Publishers B.V. (North-Holland), 1988.

[11] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 1987.

[12] T.V. Vu and S.P. Risner, "Performance and cost of broadcast routing algorithms in the strategic defense system terrestrial network," in *IEEE MILCOM*, 1991, pp. 29.2.1–29.2.5.

[13] A.S. Tanenbaum, *Computer Networks*, Prentice-Hall International, second edition, 1989.

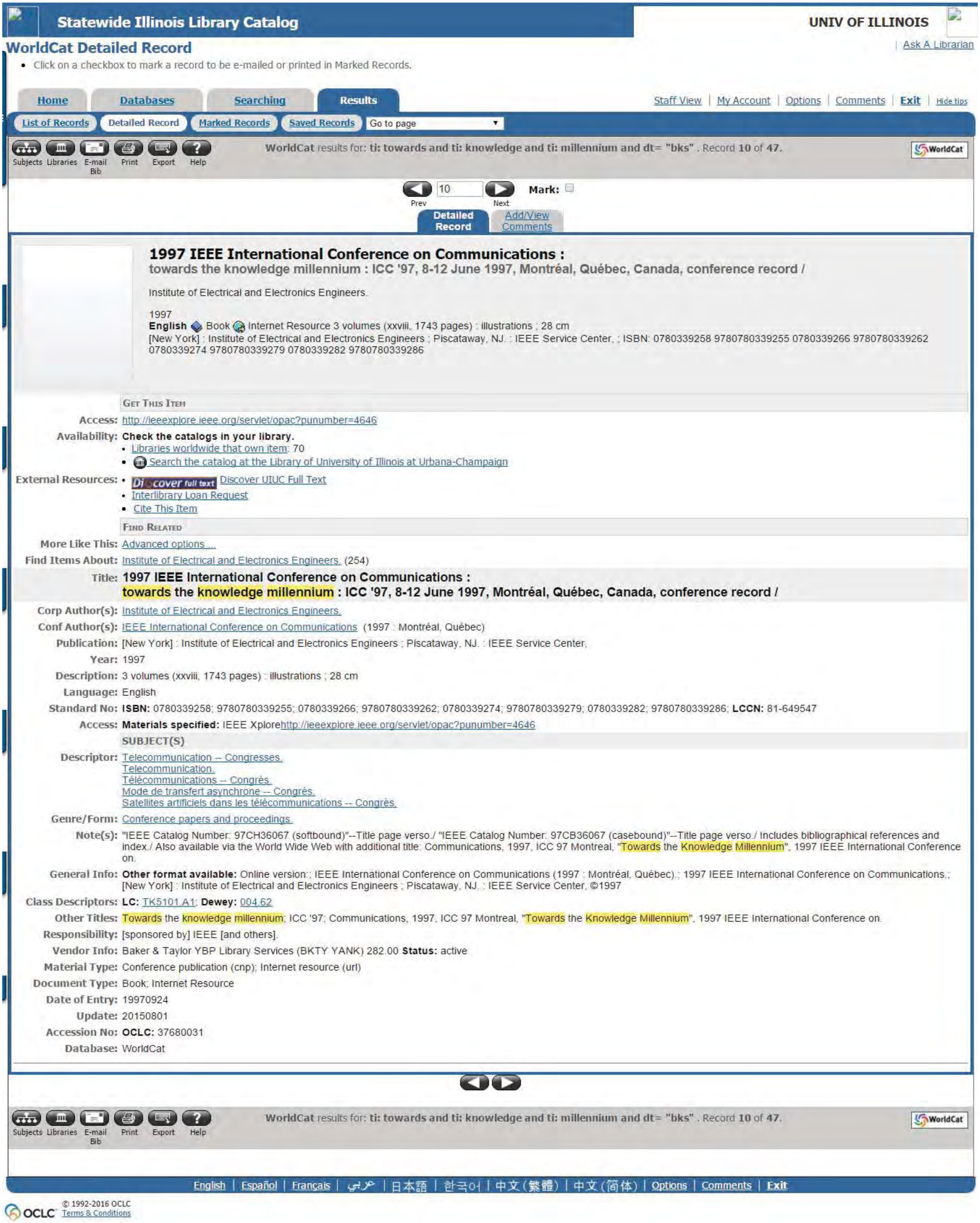
[14] B.H. Davies, S.J. Swain, and K. Watkins, "Efficient implementation of X.25 on narrowband packet radio systems," in *IEE Colloquium on "Military Communications - The Trend Towards Civil Standards", (Digest No. 55)*, April 1990, pp. 6/1–6/5.

[15] P.R. Karn, H.E. Price, and R.J. Diersing, "Packet radio in the amateur service," *IEEE Journal on Selected Areas in Communications*, vol. 3, no. 3, pp. 431–439, May 1985.

[16] V.O.K. Li and R-F Chang, "Proposed routing algorithms for the U.S. Army mobile subscriber equipment (MSE) network," in *IEEE MILCOM*, 1986, pp. 39.4.1–39.4.7.

[17] L.E. Miller, R.H. French, J.S. Lee, and D.J. Torrieri, "MSE routing algorithm comparison," in *IEEE MILCOM*, 1989, pp. 2.2.1–2.2.5.

Attachment 1d: Statewide Illinois Library Catalog monograph record for the proceedings of 1997 IEEE International Conference on Communications



WorldCat Detailed Record

- Click on a checkbox to mark a record to be e-mailed or printed in Marked Records.

WorldCat results for: **ti: towards and ti: knowledge and ti: millennium and dt= "bks"** . Record **10** of **47**.

10

1997 IEEE International Conference on Communications :

towards the knowledge millennium : ICC '97, 8-12 June 1997, Montréal, Québec, Canada, conference record /

Institute of Electrical and Electronics Engineers.

1997
English Book Internet Resource 3 volumes (xxviii, 1743 pages) : illustrations ; 28 cm
[New York] : Institute of Electrical and Electronics Engineers ; Piscataway, NJ. : IEEE Service Center, ; ISBN: 0780339258 9780780339255 0780339266 9780780339262 0780339274 9780780339279 0780339282 9780780339286

GET THIS ITEM

Access: http://ieeexplore.ieee.org/servlet/opac?punumber=4646

Availability: **Check the catalogs in your library.**
- Libraries worldwide that own item: 70
- Search the catalog at the Library of University of Illinois at Urbana-Champaign

External Resources:
- Discover full text Discover UIUC Full Text
- Interlibrary Loan Request
- Cite This Item

FIND RELATED

More Like This: Advanced options ...

Find Items About: Institute of Electrical and Electronics Engineers. (254)

Title: **1997 IEEE International Conference on Communications :
towards the knowledge millennium : ICC '97, 8-12 June 1997, Montréal, Québec, Canada, conference record /**

Corp Author(s): Institute of Electrical and Electronics Engineers.

Conf Author(s): IEEE International Conference on Communications (1997 : Montréal, Québec)

Publication: [New York] : Institute of Electrical and Electronics Engineers ; Piscataway, NJ. : IEEE Service Center,

Year: 1997

Description: 3 volumes (xxviii, 1743 pages) : illustrations ; 28 cm

Language: English

Standard No: **ISBN:** 0780339258; 9780780339255; 0780339266; 9780780339262; 0780339274; 9780780339279; 0780339282; 9780780339286; **LCCN:** 81-649547

Access: **Materials specified:** IEEE Xplorehttp://ieeexplore.ieee.org/servlet/opac?punumber=4646

SUBJECT(S)

Descriptor: Telecommunication -- Congresses.
Telecommunication.
Télécommunications -- Congrès.
Mode de transfert asynchrone -- Congrès.
Satellites artificiels dans les télécommunications -- Congrès.

Genre/Form: Conference papers and proceedings.

Note(s): "IEEE Catalog Number: 97CH36067 (softbound)"--Title page verso./ "IEEE Catalog Number: 97CB36067 (casebound)"--Title page verso./ Includes bibliographical references and index./ Also available via the World Wide Web with additional title: Communications, 1997, ICC 97 Montreal, "Towards the Knowledge Millennium", 1997 IEEE International Conference on.

General Info: **Other format available:** Online version:; IEEE International Conference on Communications (1997 : Montréal, Québec).; 1997 IEEE International Conference on Communications.; [New York] : Institute of Electrical and Electronics Engineers ; Piscataway, NJ. : IEEE Service Center, ©1997

Class Descriptors: **LC:** TK5101.A1; **Dewey:** 004.62

Other Titles: Towards the knowledge millennium; ICC '97; Communications, 1997, ICC 97 Montreal, "Towards the Knowledge Millennium", 1997 IEEE International Conference on.

Responsibility: [sponsored by] IEEE [and others].

Vendor Info: Baker & Taylor YBP Library Services (BKTY YANK) 282.00 **Status:** active

Material Type: Conference publication (cnp); Internet resource (url)

Document Type: Book; Internet Resource

Date of Entry: 19970924

Update: 20150801

Accession No: **OCLC:** 37680031

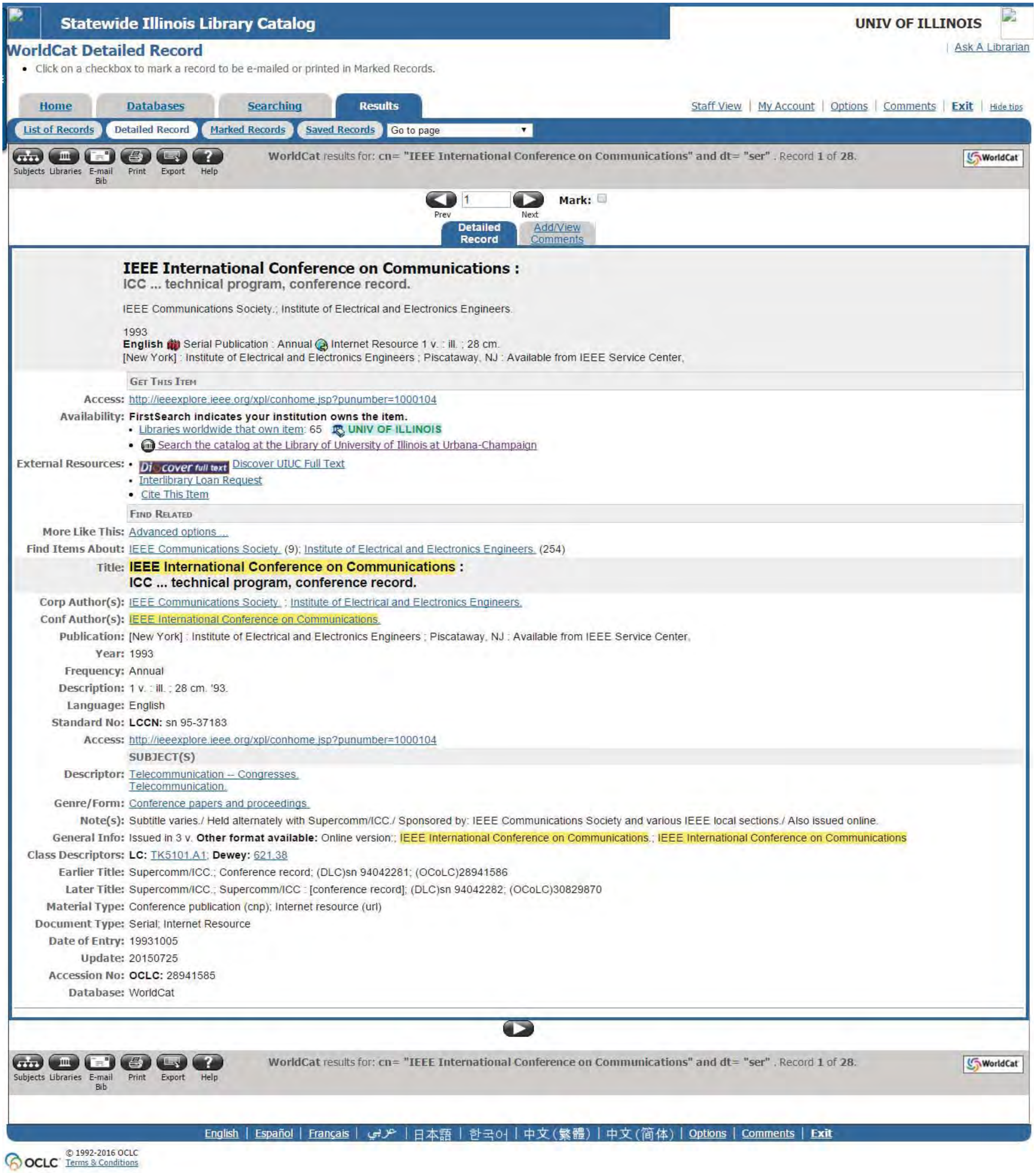
Database: WorldCat

WorldCat results for: **ti: towards and ti: knowledge and ti: millennium and dt= "bks"** . Record **10** of **47**.

English | Español | Français | عربي | 日本語 | 한국어 | 中文(繁體) | 中文(简体) | Options | Comments | Exit
OCLC

Attachment 1d: Statewide Illinois Library Catalog monograph record for the proceedings of 1997 IEEE International Conference on Communications

Attachment 1e: Statewide Illinois Library Catalog series record for the proceedings of 1997 IEEE International Conference on Communications



WorldCat Detailed Record

- Click on a checkbox to mark a record to be e-mailed or printed in Marked Records.

WorldCat results for: cn= "IEEE International Conference on Communications" and dt= "ser" . Record 1 of 28.

IEEE International Conference on Communications :
ICC ... technical program, conference record.

IEEE Communications Society.; Institute of Electrical and Electronics Engineers.

1993
English Serial Publication : Annual Internet Resource 1 v. : ill. ; 28 cm.
[New York] : Institute of Electrical and Electronics Engineers ; Piscataway, NJ : Available from IEEE Service Center,

GET THIS ITEM

Access: http://ieeexplore.ieee.org/xpl/conhome.jsp?punumber=1000104
Availability: **FirstSearch indicates your institution owns the item.**
- Libraries worldwide that own item: 65 UNIV OF ILLINOIS
- Search the catalog at the Library of University of Illinois at Urbana-Champaign

External Resources:
- Discover full text Discover UIUC Full Text
- Interlibrary Loan Request
- Cite This Item

FIND RELATED

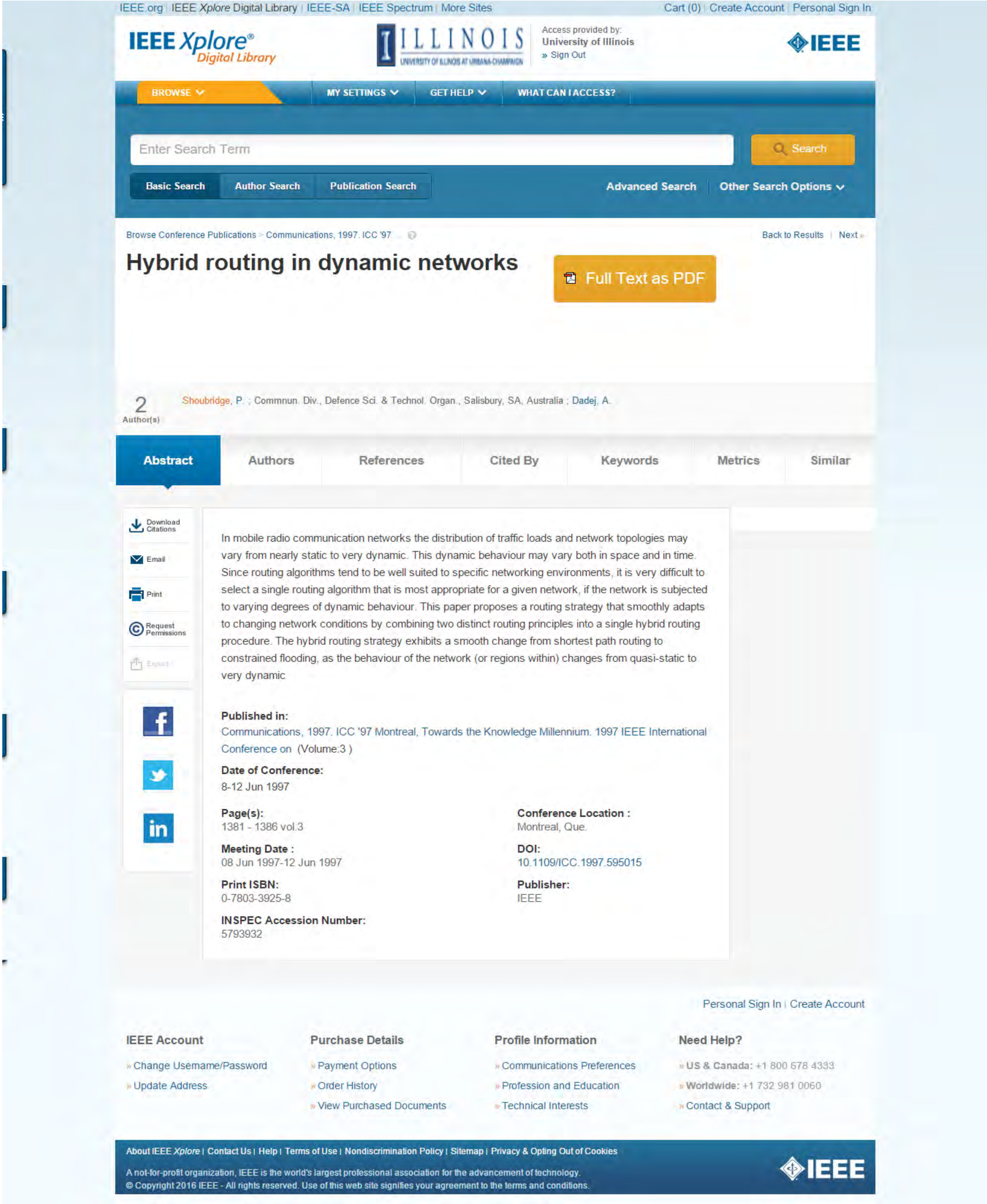
More Like This: Advanced options ...
Find Items About: IEEE Communications Society. (9); Institute of Electrical and Electronics Engineers. (254)
Title: **IEEE International Conference on Communications : ICC ... technical program, conference record.**
Corp Author(s): IEEE Communications Society. ; Institute of Electrical and Electronics Engineers.
Conf Author(s): IEEE International Conference on Communications.
Publication: [New York] : Institute of Electrical and Electronics Engineers ; Piscataway, NJ : Available from IEEE Service Center,
Year: 1993
Frequency: Annual
Description: 1 v. : ill. ; 28 cm. '93.
Language: English
Standard No: **LCCN:** sn 95-37183
Access: http://ieeexplore.ieee.org/xpl/conhome.jsp?punumber=1000104

SUBJECT(S)

Descriptor: Telecommunication -- Congresses.
Telecommunication.
Genre/Form: Conference papers and proceedings.
Note(s): Subtitle varies./ Held alternately with Supercomm/ICC./ Sponsored by: IEEE Communications Society and various IEEE local sections./ Also issued online.
General Info: Issued in 3 v. **Other format available:** Online version:; IEEE International Conference on Communications.; IEEE International Conference on Communications
Class Descriptors: **LC:** TK5101.A1; **Dewey:** 621.38
Earlier Title: Supercomm/ICC.; Conference record; (DLC)sn 94042281; (OCoLC)28941586
Later Title: Supercomm/ICC.; Supercomm/ICC : [conference record]; (DLC)sn 94042282; (OCoLC)30829870
Material Type: Conference publication (cnp); Internet resource (url)
Document Type: Serial; Internet Resource
Date of Entry: 19931005
Update: 20150725
Accession No: **OCLC:** 28941585
Database: WorldCat

WorldCat results for: cn= "IEEE International Conference on Communications" and dt= "ser" . Record 1 of 28.

Attachment 1f: IEEE Xplore Digital Library index record for Document 1


IEEE.org | IEEE Xplore Digital Library | IEEE-SA | IEEE Spectrum | More Sites
Cart (0) | Create Account | Personal Sign In
IEEE Xplore® Digital Library
ILLINOIS UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
Access provided by: University of Illinois » Sign Out
IEEE
BROWSE
MY SETTINGS
GET HELP
WHAT CAN I ACCESS?
Enter Search Term
Search
Basic Search
Author Search
Publication Search
Advanced Search
Other Search Options
Browse Conference Publications > Communications, 1997. ICC '97 ...
Back to Results | Next »
Full Text as PDF
Abstract
Authors
References
Cited By
Keywords
Metrics
Similar
Download Citations
Email
Print
Request Permissions
Export

Hybrid routing in dynamic networks

2 Author(s)

Shoubridge, P. ; Commun. Div., Defence Sci. & Technol. Organ., Salisbury, SA, Australia ; Dadej, A.

In mobile radio communication networks the distribution of traffic loads and network topologies may vary from nearly static to very dynamic. This dynamic behaviour may vary both in space and in time. Since routing algorithms tend to be well suited to specific networking environments, it is very difficult to select a single routing algorithm that is most appropriate for a given network, if the network is subjected to varying degrees of dynamic behaviour. This paper proposes a routing strategy that smoothly adapts to changing network conditions by combining two distinct routing principles into a single hybrid routing procedure. The hybrid routing strategy exhibits a smooth change from shortest path routing to constrained flooding, as the behaviour of the network (or regions within) changes from quasi-static to very dynamic

Published in:
Communications, 1997. ICC '97 Montreal, Towards the Knowledge Millennium. 1997 IEEE International Conference on (Volume:3)

Date of Conference:
8-12 Jun 1997

Page(s):
1381 - 1386 vol.3

Conference Location :
Montreal, Que.

Meeting Date :
08 Jun 1997-12 Jun 1997

DOI:
10.1109/ICC.1997.595015

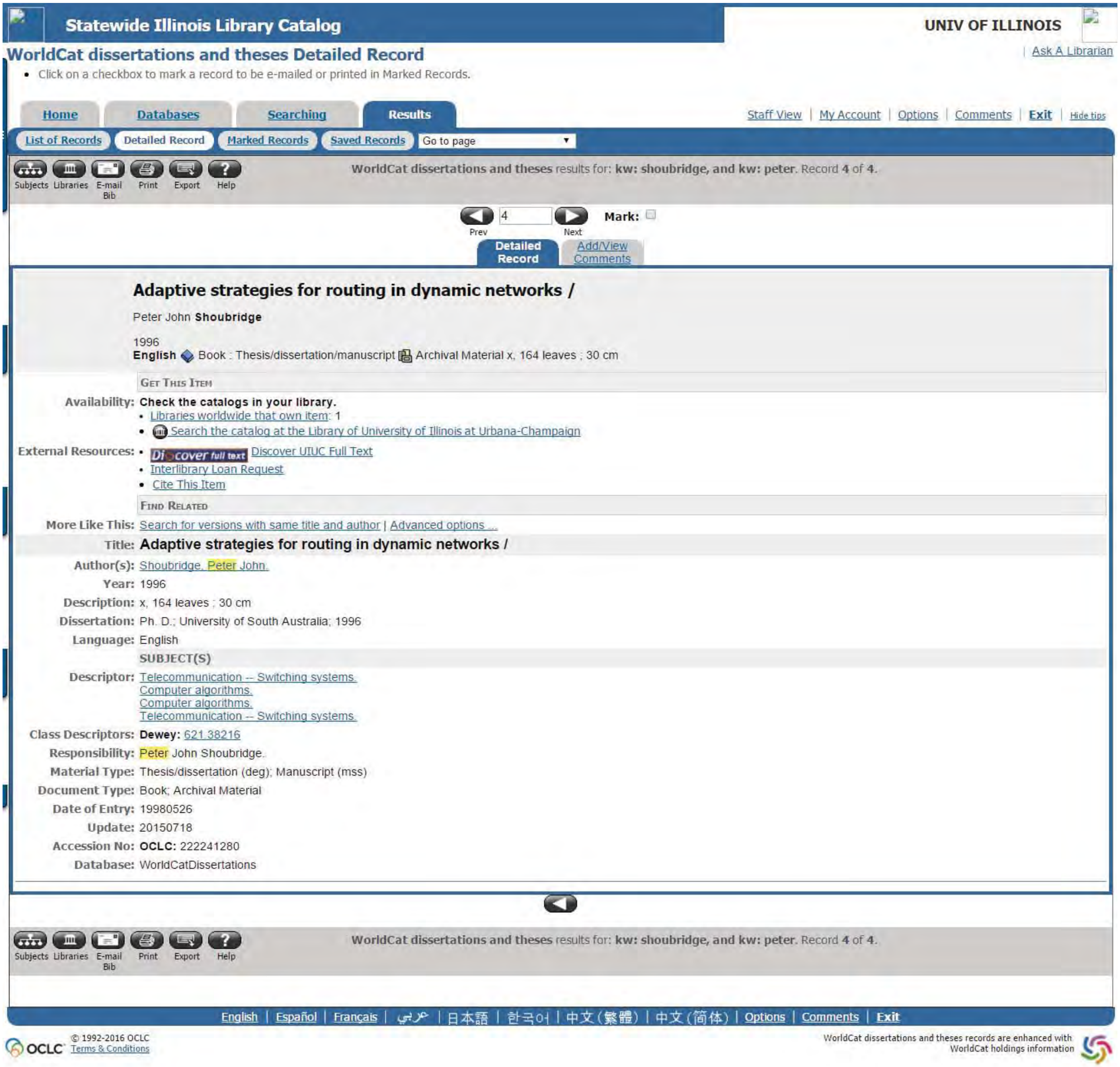
Print ISBN:
0-7803-3925-8

Publisher:
IEEE

INSPEC Accession Number:
5793932

Personal Sign In | Create Account
IEEE Account
» Change Username/Password
» Update Address
Purchase Details
» Payment Options
» Order History
» View Purchased Documents
Profile Information
» Communications Preferences
» Profession and Education
» Technical Interests
Need Help?
» US & Canada: +1 800 678 4333
» Worldwide: +1 732 981 0060
» Contact & Support
About IEEE Xplore | Contact Us | Help | Terms of Use | Nondiscrimination Policy | Sitemap | Privacy & Opting Out of Cookies

Attachment 2a: Statewide Illinois Library Catalog record for Document 2



WorldCat dissertations and theses Detailed Record

- Click on a checkbox to mark a record to be e-mailed or printed in Marked Records.

WorldCat dissertations and theses results for: **kw: shoubridge, and kw: peter**. Record **4** of **4**.

4 Mark: □

Adaptive strategies for routing in dynamic networks /

Peter John **Shoubridge**

1996
English Book : Thesis/dissertation/manuscript Archival Material x, 164 leaves ; 30 cm

GET THIS ITEM

Availability:	**Check the catalogs in your library.** • Libraries worldwide that own item: 1 • Search the catalog at the Library of University of Illinois at Urbana-Champaign
External Resources:	• Discover full text Discover UIUC Full Text • Interlibrary Loan Request • Cite This Item
	FIND RELATED
More Like This:	Search for versions with same title and author | Advanced options ...
Title:	**Adaptive strategies for routing in dynamic networks /**
Author(s):	Shoubridge, Peter John.
Year:	1996
Description:	x, 164 leaves ; 30 cm
Dissertation:	Ph. D.; University of South Australia; 1996
Language:	English
	SUBJECT(S)
Descriptor:	Telecommunication -- Switching systems. Computer algorithms. Computer algorithms. Telecommunication -- Switching systems.
Class Descriptors:	**Dewey:** 621.38216
Responsibility:	Peter John Shoubridge.
Material Type:	Thesis/dissertation (deg); Manuscript (mss)
Document Type:	Book; Archival Material
Date of Entry:	19980526
Update:	20150718
Accession No:	**OCLC:** 222241280
Database:	WorldCatDissertations

WorldCat dissertations and theses results for: **kw: shoubridge, and kw: peter**. Record **4** of **4**.

WorldCat dissertations and theses records are enhanced with WorldCat holdings information

1996 IEEE 16TH INTERNATIONAL CONFERENCE ON DISTRIBUTED COMPUTING SYSTEMS

Proceedings of the

16th International Conference on Distributed Computing Systems

May 27 – 30, 1996

Hong Kong

Sponsored by

The IEEE Computer Society Technical Committee on Distributed Processing

IEEE Computer Society Press
Los Alamitos, California
Washington • Brussels • Tokyo

IEEE Computer Society Press
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314

The papers in this book comprise the proceedings of the meeting mentioned on the cover and title page. They reflect the authors' opinions and, in the interests of timely dissemination, are published as presented and without change. Their inclusion in this publication does not necessarily constitute endorsement by the editors, the IEEE Computer Society Press, or the Institute of Electrical and Electronics Engineers, Inc.

IEEE Computer Society Press Order Number PR07398
ISBN 0-8186-7398-2
ISSN 1063-6927

IEEE Order Plan Catalog Number 96CB35954
IEEE Order Plan ISBN 0-8186-7399-0
Microfiche ISBN 0-8186-7400-8

Additional copies may be ordered from:

IEEE Computer Society Press
Customer Service Center
10662 Los Vaqueros Circle
P.O. Box 3014
Los Alamitos, CA 90720-1314
Tel: +1-714-821-8380
Fax: +1-714-821-4641
Email: cs.books@computer.org

IEEE Service Center
445 Hoes Lane
P.O. Box 1331
Piscataway, NJ 08855-1331
Tel: +1-908-981-1393
Fax: +1-908-981-9667
mis.custserv@computer.org

IEEE Computer Society
13, Avenue de l'Aquilon
B-1200 Brussels
BELGIUM
Tel: +32-2-770-2198
Fax: +32-2-770-8505
euro.ofc@computer.org

IEEE Computer Society
Ooshima Building
2-19-1 Minami-Aoyama
Minato-ku, Tokyo 107
JAPAN
Tel: +81-3-3408-3118
Fax: +81-3-3408-3553
tokyo.ofc@computer.org

Editorial production by Mary E. Kavanaugh
Cover by Joseph Daigle
Printed in the United States of America by KNI Inc.

The Institute of Electrical and Electronics Engineers, Inc.

Table of Contents

Keynote Address: "From HPCC to New Millennium Computing"

Speaker: T.Y. Feng – *Pennsylvania State University*

Session 1A: Fault-Tolerant Applications and Frameworks

Session 1B: Real-Time Synchronization and Scheduling

Session 1C: Distributed Shared Memory

Session 8B: Migration Strategies

Session 8C: The Web and the Internet

Panel Session 9A: "Impact of Technologies on Distributed Computing"

Panel Session 9B: "How Will Distributed Object Standards Impact the Distributed Computing Systems of the Future?"

Session 10A: Transactions

Session 10B: Mutual Exclusion

Message from the Program Co-Chairs

Due to the interest of a broad range of researchers from industry, academia, and government laboratories worldwide, ICDCS-16, as in ICDCS in previous years, brings together a diverse program of innovative research. Again this year, it is demonstrated that ICDCS is the premier conference for distributed computing.

Of 305 manuscripts received, 295 were sent forward for review and managed by 11 area Vice-Chairs; ten additional manuscripts were considered out of the scope of this conference or came after the deadline. The 11 areas and their Vice-Chairs were:

1. Distributed Operating Systems: Prof. Mukesh Singhal – *Ohio State University*
2. Distributed Databases and Information Systems: Prof. Tamer Ozsu – *University of Alberta*
3. Communication Protocols: Prof. Teruo Higashino – *Osaka University*
4. Distributed Real-Time Systems: Prof. Farnam Jahanian – *University of Michigan*
5. Languages, Tools and Software Engineering: Prof. Jeffrey Kramer – *Imperial College*
6. Computer Architecture and Interconnection: Prof. Lionel Ni – *Michigan State University*
7. Distributed Resource Management and Scheduling: Prof. Richard D. Schlichting – *University of Arizona*
8. Fault Tolerance, Availability and Security: Prof. Graham D. Parrington – *University of Newcastle upon Tyne*
9. Performance of Distributed Systems: Prof. Erol Gelenbe – *Duke University*
10. Mobile Computing: Dr. Hamid Ahmadi – *IBM T.J. Watson Research Center*
11. Distributed Algorithms and Applications: Prof. Nicola Santoro – *Carleton University*

They were also empowered to form the Program Committee by appointing members of their choosing. Each Vice-Chair was responsible for supervising the review of the papers belonging to one technical area and submitted reviews and a recommendation for disposition of each paper to the Program Co-Chairs. Additionally, each Vice-Chair could submit a nomination for the Best Paper award. In the final Program Committee meeting held in New Orleans (January 20 – 21, 1996), the technical merits of the selected papers were discussed, the scores of the recommended papers in the different areas were compared, the final selections were made, the session organization was planned, and five nominees for Best Paper were put forward.

Highlighting the truly international nature of the conference, the submissions originated from 29 countries, and from a total of 295 reviewed papers, 86 were selected for the final program:

USA	116	Japan	22	Hong Kong	20
France	19	Australia	17	Germany	13
UK	11	Taiwan	10	Korea	10
Canada	10	China	6	Switzerland	6
Italy	5	India	3	Netherlands	3
Poland	3	Portugal	3	Singapore	2
New Zealand	2	Norway	2	Belgium	2
Spain	2	Saudi Arabia	2	Isreal	1
Turkey	1	Syria	1	Greece	1
Mexico	1	Austria	1		
				Total:	**295**

Disposition of papers by topic	**Received**	**Accepted**
Distributed Operating Systems	31	9
Distributed Databases and Information Systems	29	8
Communication Protocols	42	11
Distributed Real-Time Systems	19	5
Languages, Tools and Software Engineering	26	8
Computer Architecture and Interconnection	20	9
Distributed Resource Management and Scheduling	21	4
Fault Tolerance, Availability and Security	24	9
Performance of Distributed Systems	16	5
Mobile Computing	25	7
Distributed Algorithms and Applications	42	11
Total:	**295**	**86**

We would like to thank Vincent Shen and the many volunteers for the tremendous effort they invested in organizing the conference and Benjamin Wah for handling the Best Paper award. We wish to thank the program participants including the speakers, panelists, and session chairs. Finally, and most of all, we thank you, the conference attendee, for considering this event to be worth your time and expense. If you acquire new insights, new ideas, or new motivation from our program, then we will consider the effort to be worthwhile.

It was our honor and pleasure to work with a team of dedicated people who made ICDCS-16 a reality.

Sam Chanson
Program Co-Chair
Hong Kong University of Science and Technology

Bill Buckles
Program Co-Chair
Tulane University

Proceedings of the International Conference on

Session 8C:
The Web and the Internet

Distributed Computing Systems – 1996

A Tool for Massively Replicating Internet Archives: Design, Implementation, and Experience

Katia Obraczka
University of Southern California
Information Science Institute
4676 Admiralty Way
Marina del Rey, CA 90292, USA
katia@isi.edu

Peter Danzig, Dante DeLucia, Erh-Yuan Tsai
University of Southern California
Computer Science Department
Los Angeles, CA 90089-0781
{danzig, dante, erhyuant}@usc.edu

Abstract

This paper reports the design, implementation, and performance of a scalable and efficient tool to replicate Internet information services. Our tool targets replication degrees of tens of thousands of weakly-consistent replicas scattered throughout the Internet's thousands of autonomously administered domains. The main goal of our replication tool is to make existing replication algorithms scale in today's exponentially-growing, autonomously-managed internetworks.

1. Introduction

Internet services provide large, rapidly evolving, highly accessed, autonomously managed information spaces. To achieve adequate performance, services such as WWW [1] will have to replicate their data in thousands of autonomous networks. As an example of a highly replicated service, take Internet news [5]. Although it manages a dynamic, flat, gigabyte database, it responds to queries in seconds. In contrast, popular WWW and FTP servers are constantly too overloaded to provide reasonable response time to users. As WWW, FTP and other Internet information services become more popular, their databases must be highly replicated for performance.

Existing replication mechanisms will not scale in today's exponentially growing, autonomously managed internets. We implemented a tool for efficient replication of Internet information services. We target replication degrees of thousands or even tens of thousands of weakly consistent replicas scattered throughout the Internet's thousands of administrative domains.

Our tool consists of two components, *mirror-d* and *flood-d*. Mirror-d propagates updates according to the logical topologies computed by flood-d. Flood-d aggregates replicas into multiple *replication groups* analogous to the way the Internet partitions itself into autonomous routing domains. Having multiple replication groups preserves autonomy, since administrative decisions of one group, such as its connectivity or when it should be split in two, do not affect other groups. Also, it insulates groups from topological rearrangements of their neighboring groups and from most of the network traffic associated with group membership.

For each replication group, flood-d automatically builds the logical topology over which updates travel. Unlike Lampson's Global Name Service (GNS) [6], flood-d's logical update topologies are not restricted to a Hamiltonian cycle. By automating the process of building update topologies among replicas of a service, flood-d offloads system administrators from having to make logical topology decisions.

We argue that efficient replication algorithms flood data between replicas. Note that the flooding scheme that we propose differs from network-level flooding as used by routing algorithms: flooding at the network level simply follows the network's physical topology and flood updates throughout all physical links of the network. Instead, the replicas flood data to their logical neighbor or peer replicas. Although the word "flooding" sounds inefficient, we claim that the application-level flooding scheme that we propose does use network bandwidth efficiently.

Flood-d employs a network topology estimator and a logical topology calculator. Every group member measures available bandwidth and propagation delay to the other

This research was funded in part by ARPA contract number DABT63-93-C-0052, AFOSR award number F49620-93-1-0082, NSF NYI award NCR-9457518, and NSF small-scale infrastructure grant number CDA-9216321. Katia Obraczka is currently supported by ARPA contract number J-FBI-95-204.

1063-6927/96 $5.00 © 1996 IEEE
Proceedings of the 16th ICDCS

group members. Based on these estimates, the logical topology calculator builds topologies for the group that are k-connected for resilience, have low total edge-cost for efficient use of the underlying network, and low diameter to limit update propagation delays.

Figure 1 illustrates the relationship between logical topologies and the underlying physical network. The left-hand figure shows three *overlapping* replication groups and their logical update topologies. The right-hand figure shows the physical network topology and the logical update topology built on top of it for the three replication groups in the left-hand figure.

1.1. What Current Algorithms Lack

As existing naming services and distributed file systems have demonstrated, the problem of replicating data that can be partitioned into autonomously managed subspaces has well-known solutions. Naming services such as the Domain Name Service (DNS) [8] and Grapevine organize their name space hierarchically according to well-defined administrative boundaries. They also use these administrative boundaries to partition their name space into several *domains*, which only need to be replicated in a handful of servers to meet adequate performance. In fact, according to the results we report in [3], over 85% of second level domains in the DNS hierarchy are replicated at most three times, while 100% of these domains use at most 7 replicas. The same study also shows that more than 90% of DNS's second-level domains store less than 1,000 entries. Because of the limited domain sizes and small number of replicas, DNS's primary-copy replication scheme performs quite adequately.

Similarly, distributed file systems organize their file space hierarchically, where intermediate nodes are directories and leaf nodes are files. Like LOCUS [11], Andrew-AFS [1] [4], and Coda [12], distributed file systems use locality of reference to partition their file space into directory subtrees. File servers replicate a subset of files in a directory subtree. Both LOCUS and Andrew provide read-only file replication, while Coda uses distributed updates to keep its writable replicas weakly consistent.

Because layered network protocols hide the network topology from application programs, replicas themselves cannot select their flooding peers to optimize use of the network. Both Grapevine and its commercial successor, the Clearinghouse [10] ignore network and update topology. GNS assumes the existence of a single administrator who hand-configures the topology over which updates travel. The GNS administrator places replicas in a Hamiltonian cycle, and reconfigures the ring when replicas are added or removed. As the number of replicas grows and replicas spread beyond single administrative boundaries, frequently reconfiguring the ring gets prohibitively expensive.

Netnews employs flooding to distribute updates among its thousands of replicas. Like GNS, NNTP site administrators hand-configure their logical flooding topology. Since obtaining current physical topology information is difficult in today's Internet, system administrators frequently confer with one another to plan changes in the logical flooding topology. They try to keep up with the dynamics of the underlying physical topology, specially as the Internet's scale and complexity increase.

The main contribution of the replication tool we built is to make GNS-like services scale in today's exponentially growing, autonomously managed internetworks. In this paper, we describe our experience designing, implementing, deploying and measuring our replication tool's performance.

2. Design

The Harvest resource discovery system [2] has been designed and implemented to solve the scalability and efficiency problems of early resource discovery services. For availability and response time, Harvest relies on massive replication of its servers. In particular, Harvest's directory, which stores information about all available servers, is expected to be highly accessed and must be massively replicated for adequate performance.

Although flood-d was designed to support Harvest servers' replication, it was built as an independent service to allow its use by other applications. Indeed, several existing Internet information services such as Network News, FTP archives, and WWW servers could use flood-d to propagate their updates more efficiently and timely, yet reducing the burden on system administrators.

The replication tool we implemented consists of two separate services: flood-d and mirror-d, whose name expresses its reliance on the well-known FTP-mirror package [7]. Flood-d estimates the underlying physical network load by measuring available bandwidth and round-trip time (RTT) between members of a replication group. Based on these estimates, flood-d computes a fault-tolerant, low-cost, low-diameter logical update topology for the group. Mirror-d, a weakly-consistent, replicated file archiver uses these logical topologies to propagate updates timely and efficiently.

To simplify replicating Harvest brokers, mirror-d uses a master copy replication scheme. All updates are performed at the master site, with replicas being read-only copies. Each replica has a version number that determines if a replica needs to be updated. Replicas pull data from neighbors rather than having a neighbor push data. This

[1] Andrew is the name of the research project at Carnegie-Mellon University. AFS is based on Andrew, and has become a product marketed and supported by Transarc Corporation.

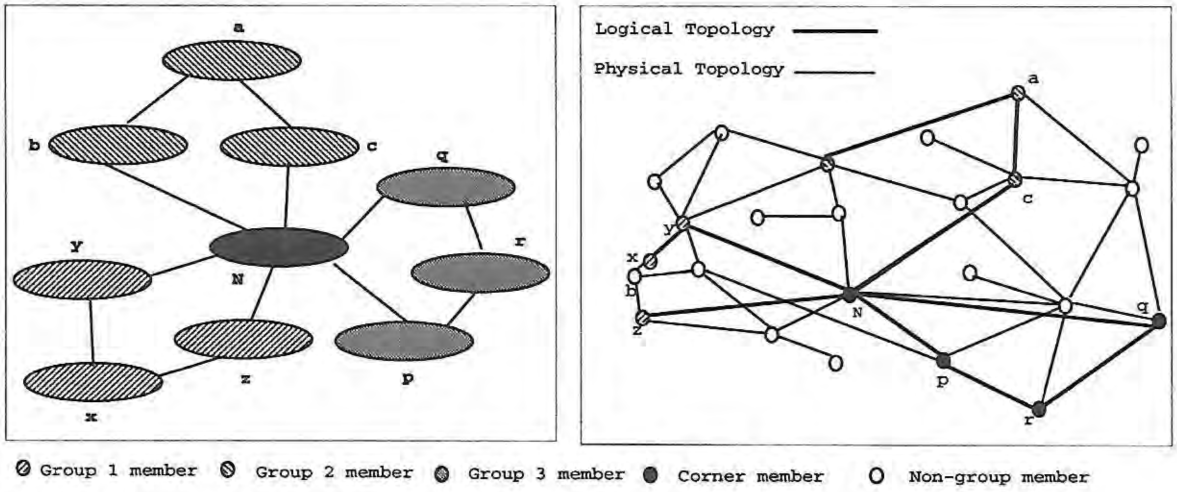


Figure 1. Replication groups showing logical versus physical topologies.

avoids the problem of multiple concurrent updates.

When a replica completes an update, it sends out a notification to its neighbors with the new version number. When a neighbor receives a notification it checks if its local version is out of date. If so, it then invokes mirror-d to update its local copy. Since it is relatively cheap to send out update notifications, mirror-ds can request them periodically to ensure that their local copies are up to date. This works well when replicas crash and lose update notifications, or are interrupted midway in an update process.

Another design decision was the use anonymous-FTP as the file propagation mechanism. While this complicates the installation of mirror-d, it has the advantage that most system administrators are familiar with anonymous-FTP. Additionally, anonymous-FTP is widely supported, and much work has been done to eliminate its security holes.

2.1. Consistency Between Groups

Consistency between replication groups is maintained as easily as it is between members of a group. Representative individual replicas, or *corner replicas* belong to multiple groups. Since replicas flood updates to neighbors in the logical topology, updates in one group make their way to all groups.

Although network node and link failures may result in network partitions and permanent node failures and group membership changes may introduce temporary inconsistencies, they are eventually resolved as long as flood-d topologies keep the nodes connected.

2.2. Updating Logical Topologies

Network nodes and links may fail temporarily, or may be permanently removed from service. 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 topology graph computed for the group.

Our replication tool uses flooding to propagate topology updates to all members of a replication group. Topology update messages carry a sequence number corresponding to the topology identifier, which replicas use to order topology updates, and detect duplicates. Topology update messages also contain the new group membership. When a replica receives a topology update, it floods the new topology according to the current topology before committing the new topology.

The topology update process generates additional traffic associated with propagating topology update messages to the participating replicas. The resulting overhead in terms of the total number of messages generated is proportional to the number of participating replicas, and the frequency with which topology updates occur. In a highly replicated service whose copies are spread throughout large internets, the topology update overhead may become prohibitively expensive. As the number of replicas increases, the overhead associated with maintaining group membership information, and estimating communication costs becomes excessive. Our hierarchical approach helps limit this overhead. Grouping replicas located physically close to one another restricts the scope of the changes of topology updates. It also limits the scope of the resulting topology updates to the local group, and therefore restricts topology update traffic on the more expensive, long-haul physical links.

The aggregate cost of topology estimation grows as $O(n^2)$ where n is the size of the group. On one hand, the more estimates collected, the more adaptive to network and server load changes the group is. On the other hand, the higher the estimation frequency, the greater the impact on network utilization. Section 5 presents estimation traffic measurements collected from replication groups of different sizes.

3. The Flood Daemon

A flood-d replica computes bandwidth and RTT estimates to other replicas in its group. A single designated site, the *group master*, generates topologies for the group. Flood-d handles group membership of one or more replication groups.

The first topology the group master generates after a new member joins will not be very good since the new member has not had time to perform bandwidth and RTT estimates between group members. The topology will get better as estimates improve.

Flood-d was written to be fast and robust. Consequently it is written as a Unix daemon that does not fork, but instead uses non-blocking I/O for all communication. The interface to flood-d is via an interface that can be queried with either 'telnet' or more conveniently with a WWW browser that speaks HTTP [1].

3.1. Membership and Multiple Groups

When a new site joins a group, it sends a join request to an existing group member. As soon as a replica receives a join request from a site, it adds the new site to the list of known sites and starts collecting network estimates for that site. The replica that receives the join request also floods it out to all replicas in the group. A site is not part of the the group until the master distributes a new topology that contains the site. This naturally gives the master control over group membership.

There is no protocol for leaving the group. Sites leave a replication group silently; after a pre-determined period of time, if a site has not been heard from, it is simply dropped from the group. This silence period is configurable and is currently set to 1 hour. Setting the silence period should take into account other group parameters such as the RTT time and bandwidth estimation periods, as well as the estimate reporting period.

A flood-d replica can be a member of more than one group. This is the case of Figure's 1's replica N.

3.2. Estimate Collection

A flood-d replica periodically performs RTT and bandwidth estimation between itself and other members of the group. To avoid synchronous group behavior, we add a random offset to the estimation frequency. Over time, a site builds estimates of RTT and available bandwidth to all other members of the group.

For RTT estimates, a replica sends a UDP packet containing a timestamp to a randomly-selected group member. When a flood-d receives such a packet it simply sends the packet back to the originator. The returned packet is then used to estimate the RTT between flood-ds. Similarly to RTT estimation, a flood-d replica estimates bandwidth by periodically choosing a random site and sending a block of data to that site. The default block size is 32 KBytes. Available bandwidth is defined as the

$$bandwidth = bytes\_sent / (time_{last\_byte} - time_{first\_byte})$$

The times at which the destination replica received the first and last bytes are given by $time_{first\_byte}$ and $time_{last\_byte}$ respectively. In order to build up a base of statistics more quickly, bandwidth is measured both when data is sent or received. Bandwidth estimation is performed whenever the data transfer meets or exceeds the bandwidth block size. While these means of collecting statistics are admittedly not very precise, they serve our purposes well enough. In Section 5, we report available bandwidth estimates using different data block sizes.

When computing the actual estimates to report to the group master, previous history is taken into account. This prevents adapting to transient changes. The damping effect is computed as follows:

$$\begin{aligned} &new\_estimate = \\ &\qquad \alpha * old\_estimate + (1 - \alpha) * current\_estimate \end{aligned}$$

where *old_estimate* is the previously reported estimate, and *current_estimate* is the estimate just measured. The damping rate α is set according to how much weight is given to past history. Currently we set α to 0.90.

To build up group estimates as quickly as possible, the periodic estimation algorithm "fast-starts" by performing more frequent estimates when flood-d first starts. Over time, the estimation process slows down to reduce the impact of bandwidth and RTT estimates on network utilization.

There is an obvious tradeoff between the ability of the system to adapt to network conditions and the amount of overhead incurred by bandwidth and RTT estimation. Large groups will need to perform more estimation than small groups. This difference is not linear since a n-replica group performs $O(n^2)$ estimates.

The end-to-end bandwidth and RTT estimates take into account the actual load on the servers involved. Measurements done at the network level might be more accurate in terms of the actual network statistics, but they do not reflect the actual delay and bandwidth seen by the application. For instance, the fact that a server tends to be heavily loaded is just as important as network congestion as far as applications are concerned.

The master collects estimates reported by group members into a cost matrix for the group, which is then used to compute the group's current logical update topology. Each entry C_ij in the cost matrix corresponds to the communication cost between nodes i and j, which is given by B_ij/D_ij, where B_ij and D_ij are the estimated bandwidth and RTT between i and j, respectively.

3.3. Topology Calculation

Flood-d generates logical update topologies by invoking a topology generator. The current topology generator uses as input the estimated cost matrix and the connectivity requirement k. It computes a minimum cost spanning tree with extra edges to provide redundancy in the event of link failure and to decrease the graph's diameter. The algorithm first computes a minimum spanning tree connecting all the nodes, and then, for each node whose degree d is less than the required connectivity k, adds the current cheapest edge until $d = k$. We are currently using $k = 2$.

The original topology generator [9] produced low-cost, low-diameter, k-connected topologies for a group using simulated annealing as its optimization technique. Our experiments demonstrated that this sophisticated algorithm only produced moderate reductions in total edge cost. Consequently, in practice we use a simpler, faster, less optimal algorithm.

Because the simulated annealing algorithm tries to lower both total edge cost and diameter, it may select more expensive links over cheaper ones. For instance, our testing environment contained several 28K PPP links. Frequently, a replica would attempt to replicate across a slow link, when there was another replica available via a local ethernet. We do not notice such phenomenon in the topologies generated by the minimum spanning tree algorithm.

4. The Mirror Daemon

A mirror-d replicated archive consists of a *master copy* and read-only replicas. Each replica contains the file system hierarchy being replicated and an associated version number. When the master site is updated, it issues a command to its mirror-d, which creates a new version number. It then sends out update notifications to neighbors according to the logical topology flood-d generates.

An update notification contains a version number, the name of the host on which the replica sending the update resides, information required to access the archive via FTP, and a template containing parameters to be used by the FTP-mirror package.

A read-only replica's mirror-d is responsible for determining if the replica's local copy is out of date. When a replica receives an update notification, it checks if the update's version is more recent than the local version. If it is, it places the notification in a notification set. After receiving several notifications, mirror-d selects from the notification set the update notification that came from the neighbor with the best bandwidth/delay metric according to the local flood-d. Using this update notification, mirror-d builds an FTP-mirror configuration file from the FTP-mirror template, and then starts an FTP-mirror process. If the mirror process succeeds, the local version is updated and and any redundant notifications are purged from the notification set. If the update fails, mirror-d selects another notification, and the FTP-mirror process repeats.

A mirror-d determines its neighbors by querying the local flood-d, and extracting a list of all neighbors and their corresponding logical link cost metrics. The fact that these neighbors might be in different replication groups is transparent to mirror-d. The mirror-d applications know nothing about the existing replication groups. Indeed, mirror-d could be easily hand-configured with pre-defined neighbors. One would lose the elegance of automatic topology calculation, but for some applications it might be useful that mirror-d be independent of flood-d.

Since a mirror-d will have several neighbors, it is often the case that it will receive update notifications from several of them. The mirror-d implementation never acts immediately on update notifications, but instead waits at least a minute to allow time for multiple update notification to arrive. It then orders the update notifications with the highest bandwidth, lowest delay neighbor first. This avoids the situation where a mirror-d will mirror over a 28K link, when it could do it over a local ethernet.

5. Performance Measurements

We have conducted preliminary performance measurement experiments with our replication tool. The goal of the first set of experiments is to evaluate the overhead flood-d generates, while the second experiment tries to assess the impact of the message size on the estimated available bandwidth.

5.1. Flood-d Overhead

Recall that replicas in a group periodically estimate RTT and available bandwidth to the other group members. For this experiment, the time between RTT and bandwidth estimation was set to 15 minutes and 1 hour, respectively. From time to time, replicas report their estimates, which the group master uses to compute a new logical topology

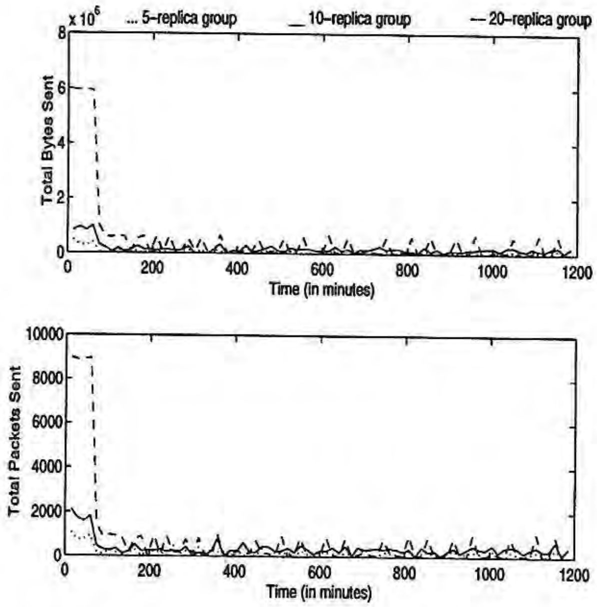


Figure 2. Total Estimation traffic for 5-, 10-, and 20-replica groups.

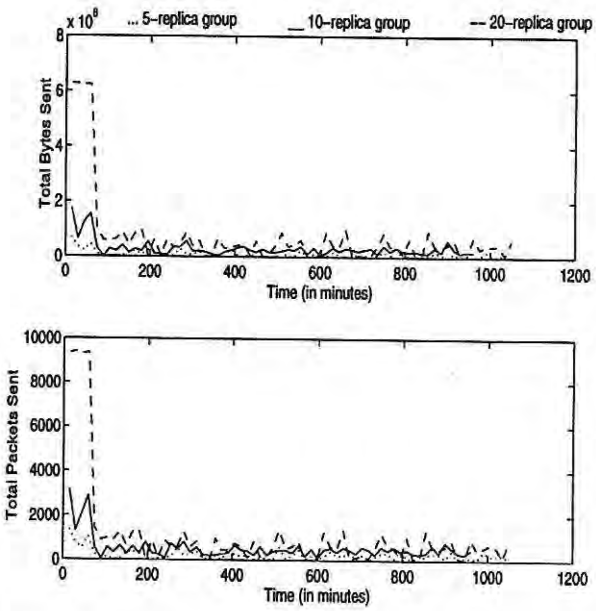


Figure 3. Total Estimation traffic for 5-, 10-, and 20-replica groups with dynamic membership.

for the group. The estimate reporting period for this experiment was set to 30 minutes. The master then floods the topology to all group members. The time between topology updates was set to 1 hour.

Replicas send RTT probes and RTT probe replies using UDP. For estimating bandwidth, sending estimates to the master, and propagating topology, they use TCP. Figure 2 shows the total overhead traffic in bytes and packets generated by a 5-, 10-, and 20-replica group. In these experiments, groups were created by having replicas join one by one. We recorded 20-hour traffic traces from replicas running locally in our laboratory. The measurements were taken every 15 minutes.

The reason we observe more traffic during the first 90 minutes is due to the fact that replicas "fast-start" when they start execution. In other words, when a replica starts, it performs estimates more frequently so that the replica's view of the group builds up faster. Notice that the amount of traffic stabilizes around a significantly lower value once the group membership becomes stable. For the 20-replica group, flood-d generates 6 MBytes of overhead traffic during approximately the first 90 minutes. From then on, flood-d's overhead traffic oscillates around 1 MByte.

Figure 3 shows how dynamic membership affects the overhead traffic flood-d generates. In this experiment, replicas join one by one in the beginning, and then, every hour a replica leaves and re-joins the group. We observe that the additional traffic generated when a replica re-joins the group is not considerable. The additional traffic generated is due to the fast-start of the re-joining replica.

5.2. Message Size and Available Bandwidth

For this experiment, we set up a wide-area, 6-replica group so that we could evaluate the impact of the bandwidth estimation message size on the estimated available bandwidth. Table 1 shows bandwidth estimates taken from a replica at USC, *ventura.usc.edu* to the other members of the group. Notice that for all sites the 16 KByte message size resulted in lower bandwidth estimates. This is probably due to the fact that the 16 KByte message was not large enough to open up the TCP window.

6. Experience

The building of flood-d and mirror-d has been a very enlightening experience. Much of the experience we gained was of the practical sort and we hope to pass some of it along to others who might be contemplating this type of work. The items below attempt to highlight some of the lessons we learned.

- Some of the problems encountered were directly related to the sheer volume of data and the large number of objects that had to be replicated. *Flood-d* was originally designed for small objects with relatively small updates (1 to 2 MBytes per update). Fortunately the

Replica Name	Available Bandwidth (Kbytes/sec)			
	16 Kbytes	32 Kbytes	48 KBytes	64 Kbytes
buzios.usc.edu	455.4	503.6	514.3	457.2
sunw0.cse.psu.edu	12.6	18.3	18.5	19.4
powell.cs.colorado.edu	9.2	17.3	27.7	16.8
casino.cchs.su.edu.au	1.8	2.0	1.4	1.5
aloe.cs.arizona.edu	7.2	9.4	20.0	8.7

Table 1. Bandwidth estimates collected at ventura.usc.edu using different message sizes.

system was designed to be modular with the ability to interface with a variety of components to handle data management. Consequently it was relatively easy to redesign the data management portion of the project and handle the archives required by Harvest.

- There needs to be a widely implemented standard for threads. Flood-d has been written entirely in C using standard UNIX system calls. This allows flood-d to use a very portable lightweight process and gives us great flexibility in dealing with network problems. A drawback is that is that non-blocking I/O is fiendishly difficult. The obvious solution to this is to use a threads package. Unfortunately, no portable threads packages were discovered that could deal with a wide variety of I/O on multiple platforms.

- Interpreters are nice for rapid prototyping. The mirror-d application is written in Perl. While Perl is not the most elegant scripting language, one can generate a portable prototype very quickly.

  On the down side, Perl rarely seems to be installed correctly, which one might construe as a portability problem. Fortunately, this is relatively simple to solve.

  A major drawback of Perl is the lack of precise memory management. Perl has a tendency to grow over time. Try as we might, we could not prevent Perl from growing. When writing a daemon process, this is very problematic. Perl tends to grow very large and eventually swamp the machine. The incredibly ugly hack solution to this is to have the Perl application restart itself periodically. Not elegant, but effective.

- FTP is not good for small file transfers. Ordinarily this is not a problem, but when transferring 16000 files of 500 bytes in size, the overhead is non-trivial. The obvious solution is to aggregate the smaller files into a much large files so that TCP can get through its slow-start phase and send data at reasonable rates.

- When moving files across multiple timezones with mirror site chosen dynamically, the issue of time becomes important. Some existing FTP daemons return file timestamps in local time, while others return time in GMT. Despite our best efforts, some host/software combinations refuse to return time in GMT. It would be helpful some notion of time were incorporated into anonymous FTP so that file timestamps were always returned in a standard timezone, or at least allow a user to discover the timezone utilized at the remote site.

- Memory is **NOT** cheap. Your application will at some point have to run on a "normal" machine. If one goes on the assumption that memory is cheap, perspective is quickly lost and memory utilization is not accounted for. If a machine has n MBytes of memory, it is always easy to use $2n$ MBytes of memory if one assumes memory is cheap. Is is best to assume that memory is a limited resource. The same can be said of disk space. Always aim for efficiency. Let someone else figure out how to use all that space, since you can always be sure that they will.

- Many assumptions are made about network connectivity. In our experience the only assumption one can make about internet connectivity is that the probability that any two sites are disconnected is non-zero. When working with a live internet, connectivity problems are rampant. At one period in time, network connectivity between USC and U Colorado Boulder network was lost with depressing regularity. When moving large amounts of data around (60 MBytes to 100 MBytes), these sort of problems can cause major headaches. It is very helpful to be able to start the transfer over from any point. When distributing these large archives, one cannot assume atomic updates, and must instead assume that update will be incremental.

- As for user interfaces, by far the best that we have found yet is HTTP. HTTP is an incredibly easy way to write a lightweight remote interface to a program. Once this has been done, it is a simple matter to find a WWW browser and use it as the front end. While HTTP may constrain the type of interaction with the application, it provides a good portable interface for many purposes.

- Care must be taken when generating data to be used for bandwidth estimation. Modern modems have the wonderful ability to automatically compress data. If textual data, or non-random data is used, the modem will in all likelihood do a reasonable compression job. We were getting reports of 6K per second over 28K dialup links when using non-random data, and 2k when using random data.

7. Conclusions and Future Work

In this paper we reported the design, implementation, and performance of a scalable and efficient replication tool for Internet information services. The primary goal of our replication tool is to make existing replication algorithms scale in today's exponentially growing, autonomously managed internetworks.

Our replication tool consists of two independent services: flood-d and mirror-d. Flood-d generates fault tolerant, low cost, low delay logical flooding topologies, which mirror-d uses to maintain weakly consistent FTP archives. Flood-d was designed as an independent service to allow its use by other Internet services, such as WWW servers, FTP archives, and an Internet cache hierarchy.

Our preliminary performance measurements indicate that flood-d's overhead behaves as expected, that is, it stabilizes over time. However, only after measuring the network and server resources flood-d's logical update topologies save, can we fully evaluate the benefits of using flood-d. This is the goal of the wide-area experiment we are currently setting up to replicate commonly accessed Internet archives. In this experiment we hope to demonstrate how flood-d's logical update topologies can reduce the mirror load on popular Internet archives, and at the same time, make more efficient use of the network, and reduce update propagation time.

In the longer term, we will investigate the use of IP multicast for bulk data distribution. Since IP multicast is a best effort protocol, a reliable transport mechanism must be built on top of it for applications that require reliable delivery. There are several efforts building reliable multicast transport protocol. We are focusing our efforts on building a flow control mechanism for bulk data distribution that can be used in conjunction with these protocols.

References

[1] T. Berners-Lee, R. Cailliau, J-F. Groff, and B. Pollermann. World-Wide Web: The information universe. *Electronic Networking: Research, Applications and Policy*, 1(2), Spring 1992.

[2] C.M. Bowman, P.B. Danzig, D.R. Hardy, U. Manber, and M.F. Schwartz. The Harvest information discovery and access system. Proceedings of the Second International World Wide Web Conference, October 1994.

[3] P.B. Danzig, K. Obraczka, and A. Kumar. An analysis of wide-area name server traffic: A study of the domain name system. *Proc. of the ACM SIGCOMM '92, Baltimore, Maryland*, August 1992.

[4] J. Howard, M. Kazar, S. Menees, D. Nichols, M. Satyanarayanan, R. Sidebotham, and M. West. Scale and performance in a distributed file system. *ACM Transactions on Computer Systems*, 6(1):51–81, February 1988.

[5] B. Kantor and P. Lapsley. Network news transfer protocol - a proposed standard for the stream-based transmission of news. Internet Request for Comments RFC 977, February 1986.

[6] B. Lampson. Designing a global name service. *Proceedings of the 5th. ACM Symposium on the Principles of Distributed Computing*, pages 1–10, August 1986.

[7] L. McLoughlin. The FTP-mirror software. Available from ftp://src.doc.ic.ac.uk/computing/archiving/mirror, January 1994.

[8] P. Mockapetris. Domain names - concepts and facilities. Internet Request for Comments RFC 1034, November 1987.

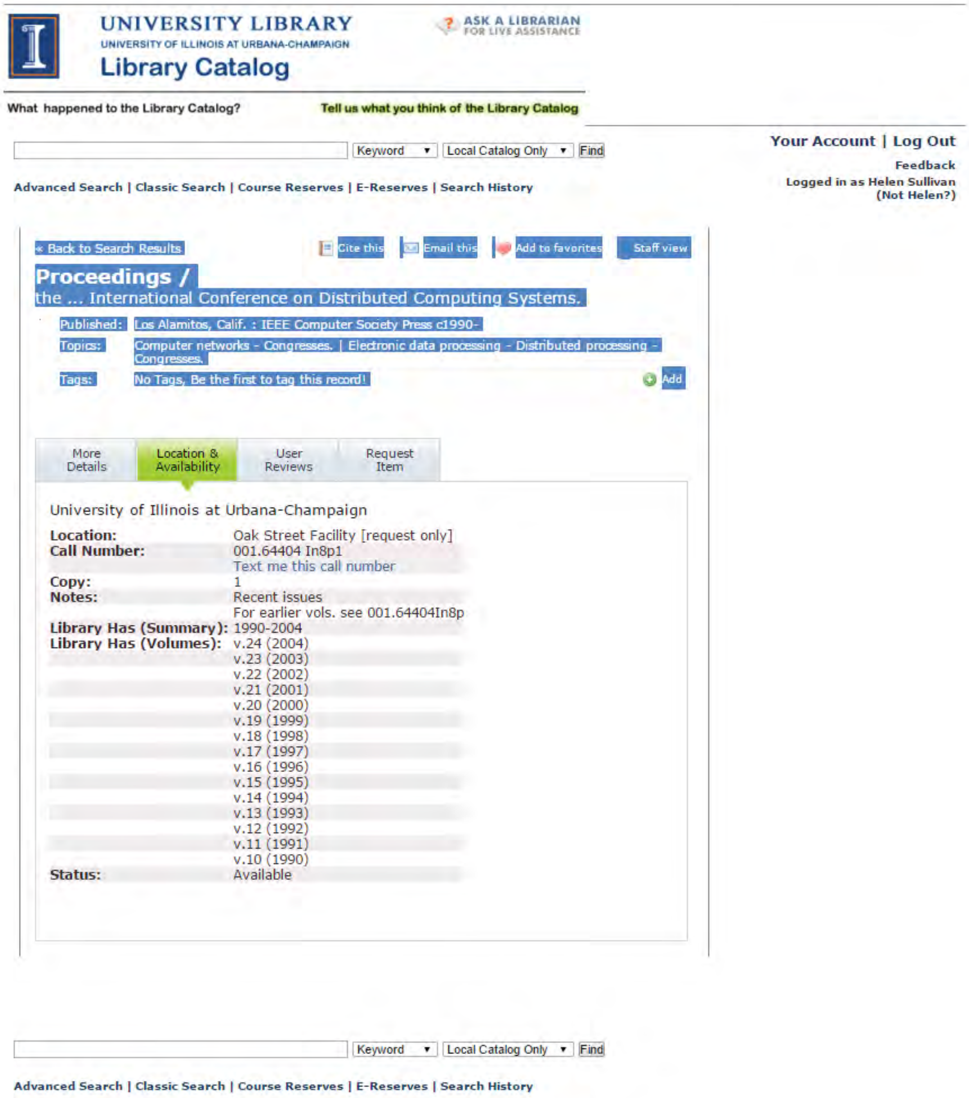
[9] K. Obraczka. *Massively Replicating Services in Wide-Area Internetworks*. PhD thesis, Computer Science Department, University of southern California, December 1994.

[10] D. Oppen and Y. Dalal. The Clearinghouse: A decentralized agent for locating named objects in a distributed environment. *ACM Transactions on Office Information Systems*, 1(3):230–253, July 1983.

[11] G. Popek, B. Walker, J. Chow, D. Edwards, C. Kline, and G. Rudisin ang G. Thiel. LOCUS: A network transparent, high reliability distributed system. *Proc. of the 8th. Symposium on Operating Systems Principles*, pages 169–177, December 1981.

[12] M. Satyanarayanan. Scalable, secure, and highly available distributed file access. *Computer Magazine*, 23(5):9–21, May 1990.

Attachment 3b: University of Illinois at Urbana-Champaign Library catalog record for the Proceedings of the 16th International Conference on Distributed Computing Systems



Logged in as Helen Sullivan
(Not Helen?)

Proceedings /
the ... International Conference on Distributed Computing Systems.

Published:	Los Alamitos, Calif. : IEEE Computer Society Press c1990-
Topics:	Computer networks -- Congresses. | Electronic data processing -- Distributed processing -- Congresses.
Tags:	No Tags, Be the first to tag this record!

University of Illinois at Urbana-Champaign

Location:	Oak Street Facility [request only]
Call Number:	001.64404 In8p1
Copy:	1
Notes:	Recent issues For earlier vols. see 001.64404In8p
Library Has (Summary):	1990-2004
Library Has (Volumes):	v.24 (2004) v.23 (2003) v.22 (2002) v.21 (2001) v.20 (2000) v.19 (1999) v.18 (1998) v.17 (1997) v.16 (1996) v.15 (1995) v.14 (1994) v.13 (1993) v.12 (1992) v.11 (1991) v.10 (1990)
Status:	Available

A Tool for Massively Replicating Internet Archives: Design, Implementation, and Experience

Katia Obraczka
University of Southern California
Information Science Institute
4676 Admiralty Way
Marina del Rey, CA 90292, USA
katia@isi.edu

Peter Danzig, Dante DeLucia, Erh-Yuan Tsai
University of Southern California
Computer Science Department
Los Angeles, CA 90089-0781
{danzig, dante, erhyuant}@usc.edu

Abstract

This paper reports the design, implementation, and performance of a scalable and efficient tool to replicate Internet information services. Our tool targets replication degrees of tens of thousands of weakly-consistent replicas scattered throughout the Internet's thousands of autonomously administered domains. The main goal of our replication tool is to make existing replication algorithms scale in today's exponentially-growing, autonomously-managed internetworks.

1. Introduction

Internet services provide large, rapidly evolving, highly accessed, autonomously managed information spaces. To achieve adequate performance, services such as WWW [1] will have to replicate their data in thousands of autonomous networks. As an example of a highly replicated service, take Internet news [5]. Although it manages a dynamic, flat, gigabyte database, it responds to queries in seconds. In contrast, popular WWW and FTP servers are constantly too overloaded to provide reasonable response time to users. As WWW, FTP and other Internet information services become more popular, their databases must be highly replicated for performance.

Existing replication mechanisms will not scale in today's exponentially growing, autonomously managed internets. We implemented a tool for efficient replication of Internet information services. We target replication degrees of thousands or even tens of thousands of weakly consistent replicas scattered throughout the Internet's thousands of administrative domains.

Our tool consists of two components, *mirror-d* and *flood-d*. Mirror-d propagates updates according to the logical topologies computed by flood-d. Flood-d aggregates replicas into multiple *replication groups* analogous to the way the Internet partitions itself into autonomous routing domains. Having multiple replication groups preserves autonomy, since administrative decisions of one group, such as its connectivity or when it should be split in two, do not affect other groups. Also, it insulates groups from topological rearrangements of their neighboring groups and from most of the network traffic associated with group membership.

For each replication group, flood-d automatically builds the logical topology over which updates travel. Unlike Lampson's Global Name Service (GNS) [6], flood-d's logical update topologies are not restricted to a Hamiltonian cycle. By automating the process of building update topologies among replicas of a service, flood-d offloads system administrators from having to make logical topology decisions.

We argue that efficient replication algorithms flood data between replicas. Note that the flooding scheme that we propose differs from network-level flooding as used by routing algorithms: flooding at the network level simply follows the network's physical topology and flood updates throughout all physical links of the network. Instead, the replicas flood data to their logical neighbor or peer replicas. Although the word "flooding" sounds inefficient, we claim that the application-level flooding scheme that we propose does use network bandwidth efficiently.

Flood-d employs a network topology estimator and a logical topology calculator. Every group member measures available bandwidth and propagation delay to the other

This research was funded in part by ARPA contract number DABT63-93-C-0052, AFOSR award number F49620-93-1-0082, NSF NYI award NCR-9457518, and NSF small-scale infrastructure grant number CDA-9216321. Katia Obraczka is currently supported by ARPA contract number J-FBI-95-204.

group members. Based on these estimates, the logical topology calculator builds topologies for the group that are *k*-connected for resilience, have low total edge-cost for efficient use of the underlying network, and low diameter to limit update propagation delays.

Figure 1 illustrates the relationship between logical topologies and the underlying physical network. The left-hand figure shows three *overlapping* replication groups and their logical update topologies. The right-hand figure shows the physical network topology and the logical update topology built on top of it for the three replication groups in the left-hand figure.

1.1. What Current Algorithms Lack

As existing naming services and distributed file systems have demonstrated, the problem of replicating data that can be partitioned into autonomously managed subspaces has well-known solutions. Naming services such as the Domain Name Service (DNS) [8] and Grapevine organize their name space hierarchically according to well-defined administrative boundaries. They also use these administrative boundaries to partition their name space into several *domains*, which only need to be replicated in a handful of servers to meet adequate performance. In fact, according to the results we report in [3], over 85% of second level domains in the DNS hierarchy are replicated at most three times, while 100% of these domains use at most 7 replicas. The same study also shows that more than 90% of DNS's second-level domains store less than 1,000 entries. Because of the limited domain sizes and small number of replicas, DNS's primary-copy replication scheme performs quite adequately.

Similarly, distributed file systems organize their file space hierarchically, where intermediate nodes are directories and leaf nodes are files. Like LOCUS [11], Andrew-AFS [1] [4], and Coda [12], distributed file systems use locality of reference to partition their file space into directory subtrees. File servers replicate a subset of files in a directory subtree. Both LOCUS and Andrew provide read-only file replication, while Coda uses distributed updates to keep its writable replicas weakly consistent.

Because layered network protocols hide the network topology from application programs, replicas themselves cannot select their flooding peers to optimize use of the network. Both Grapevine and its commercial successor, the Clearinghouse [10] ignore network and update topology. GNS assumes the existence of a single administrator who hand-configures the topology over which updates travel. The GNS administrator places replicas in a Hamiltonian cycle, and reconfigures the ring when replicas are added or removed. As the number of replicas grows and replicas spread beyond single administrative boundaries, frequently reconfiguring the ring gets prohibitively expensive.

Netnews employs flooding to distribute updates among its thousands of replicas. Like GNS, NNTP site administrators hand-configure their logical flooding topology. Since obtaining current physical topology information is difficult in today's Internet, system administrators frequently confer with one another to plan changes in the logical flooding topology. They try to keep up with the dynamics of the underlying physical topology, specially as the Internet's scale and complexity increase.

The main contribution of the replication tool we built is to make GNS-like services scale in today's exponentially growing, autonomously managed internetworks. In this paper, we describe our experience designing, implementing, deploying and measuring our replication tool's performance.

2. Design

The Harvest resource discovery system [2] has been designed and implemented to solve the scalability and efficiency problems of early resource discovery services. For availability and response time, Harvest relies on massive replication of its servers. In particular, Harvest's directory, which stores information about all available servers, is expected to be highly accessed and must be massively replicated for adequate performance.

Although flood-d was designed to support Harvest servers' replication, it was built as an independent service to allow its use by other applications. Indeed, several existing Internet information services such as Network News, FTP archives, and WWW servers could use flood-d to propagate their updates more efficiently and timely, yet reducing the burden on system administrators.

The replication tool we implemented consists of two separate services: flood-d and mirror-d, whose name expresses its reliance on the well-known FTP-mirror package [7]. Flood-d estimates the underlying physical network load by measuring available bandwidth and round-trip time (RTT) between members of a replication group. Based on these estimates, flood-d computes a fault-tolerant, low-cost, low-diameter logical update topology for the group. Mirror-d, a weakly-consistent, replicated file archiver uses these logical topologies to propagate updates timely and efficiently.

To simplify replicating Harvest brokers, mirror-d uses a master copy replication scheme. All updates are performed at the master site, with replicas being read-only copies. Each replica has a version number that determines if a replica needs to be updated. Replicas pull data from neighbors rather than having a neighbor push data. This

[1] Andrew is the name of the research project at Carnegie-Mellon University. AFS is based on Andrew, and has become a product marketed and supported by Transarc Corporation.

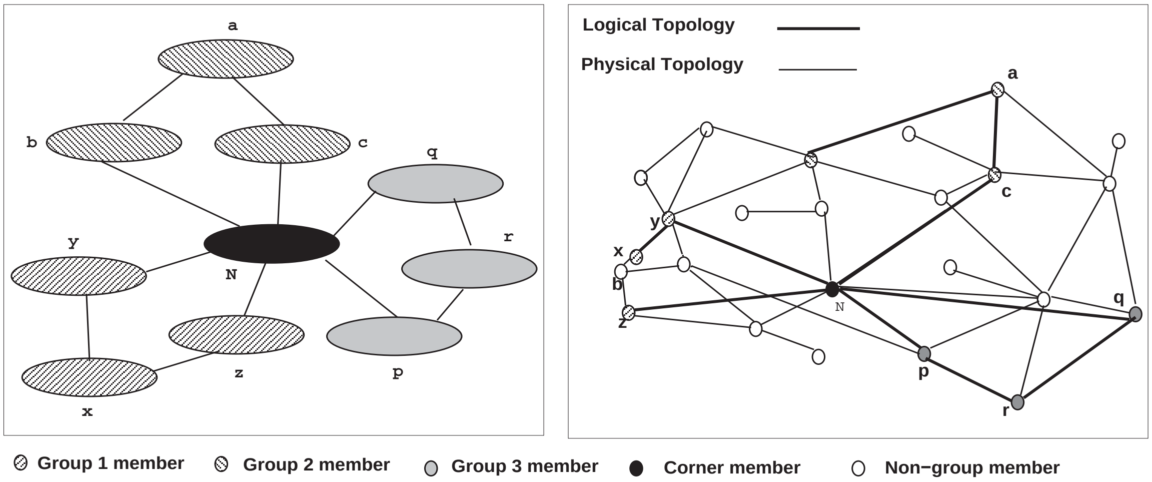


Figure 1. Replication groups showing logical versus physical topologies.

avoids the problem of multiple concurrent updates.

When a replica completes an update, it sends out a notification to its neighbors with the new version number. When a neighbor receives a notification it checks if its local version is out of date. If so, it then invokes mirror-d to update its local copy. Since it is relatively cheap to send out update notifications, mirror-ds can request them periodically to ensure that their local copies are up to date. This works well when replicas crash and lose update notifications, or are interrupted midway in an update process.

Another design decision was the use anonymous-FTP as the file propagation mechanism. While this complicates the installation of mirror-d, it has the advantage that most system administrators are familiar with anonymous-FTP. Additionally, anonymous-FTP is widely supported, and much work has been done to eliminate its security holes.

2.1. Consistency Between Groups

Consistency between replication groups is maintained as easily as it is between members of a group. Representative individual replicas, or *corner replicas* belong to multiple groups. Since replicas flood updates to neighbors in the logical topology, updates in one group make their way to all groups.

Although network node and link failures may result in network partitions and permanent node failures and group membership changes may introduce temporary inconsistencies, they are eventually resolved as long as flood-d topologies keep the nodes connected.

2.2. Updating Logical Topologies

Network nodes and links may fail temporarily, or may be permanently removed from service. 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 topology graph computed for the group.

Our replication tool uses flooding to propagate topology updates to all members of a replication group. Topology update messages carry a sequence number corresponding to the topology identifier, which replicas use to order topology updates, and detect duplicates. Topology update messages also contain the new group membership. When a replica receives a topology update, it floods the new topology according to the current topology before committing the new topology.

The topology update process generates additional traffic associated with propagating topology update messages to the participating replicas. The resulting overhead in terms of the total number of messages generated is proportional to the number of participating replicas, and the frequency with which topology updates occur. In a highly replicated service whose copies are spread throughout large internets, the topology update overhead may become prohibitively expensive. As the number of replicas increases, the overhead associated with maintaining group membership information, and estimating communication costs becomes excessive. Our hierarchical approach helps limit this overhead. Grouping replicas located physically close to one another restricts the scope of the changes of topology updates. It also limits the scope of the resulting topology updates to the local group, and therefore restricts topology update traffic on the more expensive, long-haul physical links.

The aggregate cost of topology estimation grows as $O(n^2)$ where n is the size of the group. On one hand, the more estimates collected, the more adaptive to network and server load changes the group is. On the other hand, the higher the estimation frequency, the greater the impact on network utilization. Section 5 presents estimation traffic measurements collected from replication groups of different sizes.

3. The Flood Daemon

A flood-d replica computes bandwidth and RTT estimates to other replicas in its group. A single designated site, the *group master*, generates topologies for the group. Flood-d handles group membership of one or more replication groups.

The first topology the group master generates after a new member joins will not be very good since the new member has not had time to perform bandwidth and RTT estimates between group members. The topology will get better as estimates improve.

Flood-d was written to be fast and robust. Consequently it is written as a Unix daemon that does not fork, but instead uses non-blocking I/O for all communication. The interface to flood-d is via an interface that can be queried with either 'telnet' or more conveniently with a WWW browser that speaks HTTP [1].

3.1. Membership and Multiple Groups

When a new site joins a group, it sends a join request to an existing group member. As soon as a replica receives a join request from a site, it adds the new site to the list of known sites and starts collecting network estimates for that site. The replica that receives the join request also floods it out to all replicas in the group. A site is not part of the the group until the master distributes a new topology that contains the site. This naturally gives the master control over group membership.

There is no protocol for leaving the group. Sites leave a replication group silently; after a pre-determined period of time, if a site has not been heard from, it is simply dropped from the group. This silence period is configurable and is currently set to 1 hour. Setting the silence period should take into account other group parameters such as the RTT time and bandwidth estimation periods, as well as the estimate reporting period.

A flood-d replica can be a member of more than one group. This is the case of Figure's 1's replica **N**.

3.2. Estimate Collection

A flood-d replica periodically performs RTT and bandwidth estimation between itself and other members of the group. To avoid synchronous group behavior, we add a random offset to the estimation frequency. Over time, a site builds estimates of RTT and available bandwidth to all other members of the group.

For RTT estimates, a replica sends a UDP packet containing a timestamp to a randomly-selected group member. When a flood-d receives such a packet it simply sends the packet back to the originator. The returned packet is then used to estimate the RTT between flood-ds. Similarly to RTT estimation, a flood-d replica estimates bandwidth by periodically choosing a random site and sending a block of data to that site. The default block size is 32 KBytes. Available bandwidth is defined as the

$$bandwidth = bytes\_sent/(time_{last\_byte} - time_{first\_byte})$$

The times at which the destination replica received the first and last bytes are given by $time_{first\_byte}$ and $time_{last\_byte}$ respectively. In order to build up a base of statistics more quickly, bandwidth is measured both when data is sent or received. Bandwidth estimation is performed whenever the data transfer meets or exceeds the bandwidth block size. While these means of collecting statistics are admittedly not very precise, they serve our purposes well enough. In Section 5, we report available bandwidth estimates using different data block sizes.

When computing the actual estimates to report to the group master, previous history is taken into account. This prevents adapting to transient changes. The damping effect is computed as follows:

$$\begin{aligned} &new\_estimate = \\ &\quad \alpha * old\_estimate + (1-\alpha) * current\_estimate \end{aligned}$$

where $old\_estimate$ is the previously reported estimate, and $current\_estimate$ is the estimate just measured. The damping rate α is set according to how much weight is given to past history. Currently we set α to 0.90.

To build up group estimates as quickly as possible, the periodic estimation algorithm "fast-starts" by performing more frequent estimates when flood-d first starts. Over time, the estimation process slows down to reduce the impact of bandwidth and RTT estimates on network utilization.

There is an obvious tradeoff between the ability of the system to adapt to network conditions and the amount of overhead incurred by bandwidth and RTT estimation. Large groups will need to perform more estimation than small groups. This difference is not linear since a n-replica group performs $O(n^2)$ estimates.

The end-to-end bandwidth and RTT estimates take into account the actual load on the servers involved. Measurements done at the network level might be more accurate in terms of the actual network statistics, but they do not reflect the actual delay and bandwidth seen by the application. For instance, the fact that a server tends to be heavily loaded is just as important as network congestion as far as applications are concerned.

The master collects estimates reported by group members into a cost matrix for the group, which is then used to compute the group's current logical update topology. Each entry C_ij in the cost matrix corresponds to the communication cost between nodes i and j, which is given by B_ij/D_ij, where B_ij and D_ij are the estimated bandwidth and RTT between i and j, respectively.

3.3. Topology Calculation

Flood-d generates logical update topologies by invoking a topology generator. The current topology generator uses as input the estimated cost matrix and the connectivity requirement k. It computes a minimum cost spanning tree with extra edges to provide redundancy in the event of link failure and to decrease the graph's diameter. The algorithm first computes a minimum spanning tree connecting all the nodes, and then, for each node whose degree d is less than the required connectivity k, adds the current cheapest edge until $d = k$. We are currently using $k = 2$.

The original topology generator [9] produced low-cost, low-diameter, k-connected topologies for a group using simulated annealing as its optimization technique. Our experiments demonstrated that this sophisticated algorithm only produced moderate reductions in total edge cost. Consequently, in practice we use a simpler, faster, less optimal algorithm.

Because the simulated annealing algorithm tries to lower both total edge cost and diameter, it may select more expensive links over cheaper ones. For instance, our testing environment contained several 28K PPP links. Frequently, a replica would attempt to replicate across a slow link, when there was another replica available via a local ethernet. We do not notice such phenomenon in the topologies generated by the minimum spanning tree algorithm.

4. The Mirror Daemon

A mirror-d replicated archive consists of a *master copy* and read-only replicas. Each replica contains the file system hierarchy being replicated and an associated version number. When the master site is updated, it issues a command to its mirror-d, which creates a new version number. It then sends out update notifications to neighbors according to the logical topology flood-d generates.

An update notification contains a version number, the name of the host on which the replica sending the update resides, information required to access the archive via FTP, and a template containing parameters to be used by the FTP-mirror package.

A read-only replica's mirror-d is responsible for determining if the replica's local copy is out of date. When a replica receives an update notification, it checks if the update's version is more recent than the local version. If it is, it places the notification in a notification set. After receiving several notifications, mirror-d selects from the notification set the update notification that came from the neighbor with the best bandwidth/delay metric according to the local flood-d. Using this update notification, mirror-d builds an FTP-mirror configuration file from the FTP-mirror template, and then starts an FTP-mirror process. If the mirror process succeeds, the local version is updated and and any redundant notifications are purged from the notification set. If the update fails, mirror-d selects another notification, and the FTP-mirror process repeats.

A mirror-d determines its neighbors by querying the local flood-d, and extracting a list of all neighbors and their corresponding logical link cost metrics. The fact that these neighbors might be in different replication groups is transparent to mirror-d. The mirror-d applications know nothing about the existing replication groups. Indeed, mirror-d could be easily hand-configured with pre-defined neighbors. One would lose the elegance of automatic topology calculation, but for some applications it might be useful that mirror-d be independent of flood-d.

Since a mirror-d will have several neighbors, it is often the case that it will receive update notifications from several of them. The mirror-d implementation never acts immediately on update notifications, but instead waits at least a minute to allow time for multiple update notification to arrive. It then orders the update notifications with the highest bandwidth, lowest delay neighbor first. This avoids the situation where a mirror-d will mirror over a 28K link, when it could do it over a local ethernet.

5. Performance Measurements

We have conducted preliminary performance measurement experiments with our replication tool. The goal of the first set of experiments is to evaluate the overhead flood-d generates, while the second experiment tries to assess the impact of the message size on the estimated available bandwidth.

5.1. Flood-d Overhead

Recall that replicas in a group periodically estimate RTT and available bandwidth to the other group members. For this experiment, the time between RTT and bandwidth estimation was set to 15 minutes and 1 hour, respectively. From time to time, replicas report their estimates, which the group master uses to compute a new logical topology

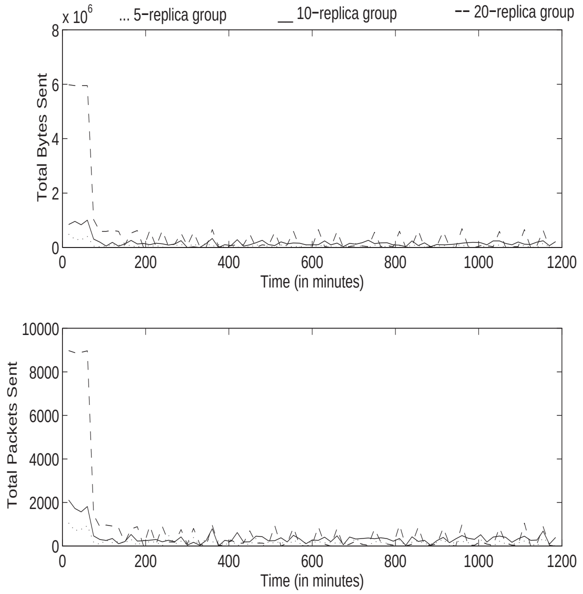


Figure 2. Total Estimation traffic for 5-, 10-, and 20-replica groups.

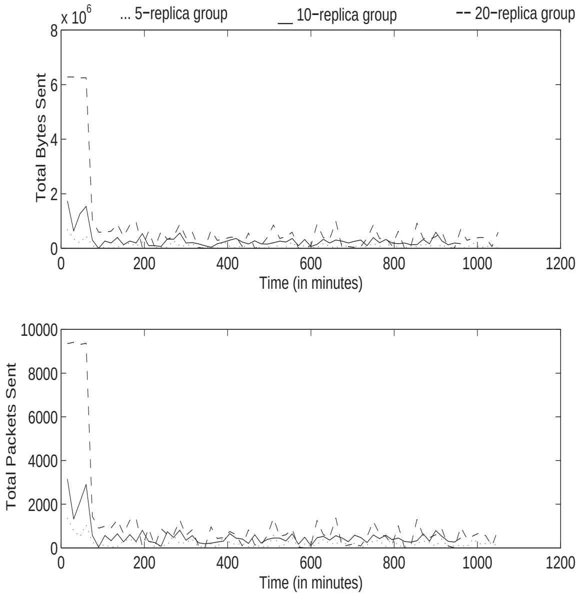


Figure 3. Total Estimation traffic for 5-, 10-, and 20-replica groups with dynamic membership.

for the group. The estimate reporting period for this experiment was set to 30 minutes. The master then floods the topology to all group members. The time between topology updates was set to 1 hour.

Replicas send RTT probes and RTT probe replies using UDP. For estimating bandwidth, sending estimates to the master, and propagating topology, they use TCP. Figure 2 shows the total overhead traffic in bytes and packets generated by a 5-, 10-, and 20-replica group. In these experiments, groups were created by having replicas join one by one. We recorded 20-hour traffic traces from replicas running locally in our laboratory. The measurements were taken every 15 minutes.

The reason we observe more traffic during the first 90 minutes is due to the fact that replicas “fast-start” when they start execution. In other words, when a replica starts, it performs estimates more frequently so that the replica’s view of the group builds up faster. Notice that the amount of traffic stabilizes around a significantly lower value once the group membership becomes stable. For the 20-replica group, flood-d generates 6 MBytes of overhead traffic during approximately the first 90 minutes. From then on, flood-d’s overhead traffic oscillates around 1 MByte.

Figure 3 shows how dynamic membership affects the overhead traffic flood-d generates. In this experiment, replicas join one by one in the beginning, and then, every hour a replica leaves and re-joins the group. We observe that the additional traffic generated when a replica re-joins the group is not considerable. The additional traffic generated is due to the fast-start of the re-joining replica.

5.2. Message Size and Available Bandwidth

For this experiment, we set up a wide-area, 6-replica group so that we could evaluate the impact of the bandwidth estimation message size on the estimated available bandwidth. Table 1 shows bandwidth estimates taken from a replica at USC, *ventura.usc.edu* to the other members of the group. Notice that for all sites the 16 KByte message size resulted in lower bandwidth estimates. This is probably due to the fact that the 16 KByte message was not large enough to open up the TCP window.

6. Experience

The building of flood-d and mirror-d has been a very enlightening experience. Much of the experience we gained was of the practical sort and we hope to pass some of it along to others who might be contemplating this type of work. The items below attempt to highlight some of the lessons we learned.

- Some of the problems encountered were directly related to the sheer volume of data and the large number of objects that had to be replicated. *Flood-d* was originally designed for small objects with relatively small updates (1 to 2 MBytes per update). Fortunately the

Replica Name	Available Bandwidth (Kbytes/sec)			
	16 Kbytes	**32 Kbytes**	**48 KBytes**	**64 Kbytes**
buzios.usc.edu	455.4	503.6	514.3	457.2
sunw0.cse.psu.edu	12.6	18.3	18.5	19.4
powell.cs.colorado.edu	9.2	17.3	27.7	16.8
casino.cchs.su.edu.au	1.8	2.0	1.4	1.5
aloe.cs.arizona.edu	7.2	9.4	20.0	8.7

Table 1. Bandwidth estimates collected at ventura.usc.edu using different message sizes.

system was designed to be modular with the ability to interface with a variety of components to handle data management. Consequently it was relatively easy to redesign the data management portion of the project and handle the archives required by Harvest.

- There needs to be a widely implemented standard for threads. Flood-d has been written entirely in C using standard UNIX system calls. This allows flood-d to use a very portable lightweight process and gives us great flexibility in dealing with network problems. A drawback is that is that non-blocking I/O is fiendishly difficult. The obvious solution to this is to use a threads package. Unfortunately, no portable threads packages were discovered that could deal with a wide variety of I/O on multiple platforms.

- Interpreters are nice for rapid prototyping. The mirror-d application is written in Perl. While Perl is not the most elegant scripting language, one can generate a portable prototype very quickly.

  On the down side, Perl rarely seems to be installed correctly, which one might construe as a portability problem. Fortunately, this is relatively simple to solve.

  A major drawback of Perl is the lack of precise memory management. Perl has a tendency to grow over time. Try as we might, we could not prevent Perl from growing. When writing a daemon process, this is very problematic. Perl tends to grow very large and eventually swamp the machine. The incredibly ugly hack solution to this is to have the Perl application restart itself periodically. Not elegant, but effective.

- FTP is not good for small file transfers. Ordinarily this is not a problem, but when transferring 16000 files of 500 bytes in size, the overhead is non-trivial. The obvious solution is to aggregate the smaller files into a much large files so that TCP can get through its slow-start phase and send data at reasonable rates.

- When moving files across multiple timezones with mirror site chosen dynamically, the issue of time becomes important. Some existing FTP daemons return file timestamps in local time, while others return time in GMT. Despite our best efforts, some host/software combinations refuse to return time in GMT. It would be helpful some notion of time were incorporated into anonymous FTP so that file timestamps were always returned in a standard timezone, or at least allow a user to discover the timezone utilized at the remote site.

- Memory is **NOT** cheap. Your application will at some point have to run on a “normal” machine. If one goes on the assumption that memory is cheap, perspective is quickly lost and memory utilization is not accounted for. If a machine has n MBytes of memory, it is always easy to use $2n$ MBytes of memory if one assumes memory is cheap. Is is best to assume that memory is a limited resource. The same can be said of disk space. Always aim for efficiency. Let someone else figure out how to use all that space, since you can always be sure that they will.

- Many assumptions are made about network connectivity. In our experience the only assumption one can make about internet connectivity is that the probability that any two sites are disconnected is non-zero. When working with a live internet, connectivity problems are rampant. At one period in time, network connectivity between USC and U Colorado Boulder network was lost with depressing regularity. When moving large amounts of data around (60 MBytes to 100 MBytes), these sort of problems can cause major headaches. It is very helpful to be able to start the transfer over from any point. When distributing these large archives, one cannot assume atomic updates, and must instead assume that update will be incremental.

- As for user interfaces, by far the best that we have found yet is HTTP. HTTP is an incredibly easy way to write a lightweight remote interface to a program. Once this has been done, it is a simple matter to find a WWW browser and use it as the front end. While HTTP may constrain the type of interaction with the application, it provides a good portable interface for many purposes.

- Care must be taken when generating data to be used for bandwidth estimation. Modern modems have the wonderful ability to automatically compress data. If textual data, or non-random data is used, the modem will in all likelihood do a reasonable compression job. We were getting reports of 6K per second over 28K dialup links when using non-random data, and 2k when using random data.

7. Conclusions and Future Work

In this paper we reported the design, implementation, and performance of a scalable and efficient replication tool for Internet information services. The primary goal of our replication tool is to make existing replication algorithms scale in today's exponentially growing, autonomously managed internetworks.

Our replication tool consists of two independent services: flood-d and mirror-d. Flood-d generates fault tolerant, low cost, low delay logical flooding topologies, which mirror-d uses to maintain weakly consistent FTP archives. Flood-d was designed as an independent service to allow its use by other Internet services, such as WWW servers, FTP archives, and an Internet cache hierarchy.

Our preliminary performance measurements indicate that flood-d's overhead behaves as expected, that is, it stabilizes over time. However, only after measuring the network and server resources flood-d's logical update topologies save, can we fully evaluate the benefits of using flood-d. This is the goal of the wide-area experiment we are currently setting up to replicate commonly accessed Internet archives. In this experiment we hope to demonstrate how flood-d's logical update topologies can reduce the mirror load on popular Internet archives, and at the same time, make more efficient use of the network, and reduce update propagation time.

In the longer term, we will investigate the use of IP multicast for bulk data distribution. Since IP multicast is a best effort protocol, a reliable transport mechanism must be built on top of it for applications that require reliable delivery. There are several efforts building reliable multicast transport protocol. We are focusing our efforts on building a flow control mechanism for bulk data distribution that can be used in conjunction with these protocols.

References

[1] T. Berners-Lee, R. Cailliau, J-F. Groff, and B. Pollermann. World-Wide Web: The information universe. *Electronic Networking: Research, Applications and Policy*, 1(2), Spring 1992.

[2] C.M. Bowman, P.B. Danzig, D.R. Hardy, U. Manber, and M.F. Schwartz. The Harvest information discovery and access system. Proceedings of the Second International World Wide Web Conference, October 1994.

[3] P.B. Danzig, K. Obraczka, and A. Kumar. An analysis of wide-area name server traffic: A study of the domain name system. *Proc. of the ACM SIGCOMM '92, Baltimore, Maryland*, August 1992.

[4] J. Howard, M. Kazar, S. Menees, D. Nichols, M. Satyanarayanan, R. Sidebotham, and M. West. Scale and performance in a distributed file system. *ACM Transactions on Computer Systems*, 6(1):51–81, February 1988.

[5] B. Kantor and P. Lapsley. Network news transfer protocol - a proposed standard for the stream-based transmission of news. Internet Request for Comments RFC 977, February 1986.

[6] B. Lampson. Designing a global name service. *Proceedings of the 5th. ACM Symposium on the Principles of Distributed Computing*, pages 1–10, August 1986.

[7] L. McLoughlin. The FTP-mirror software. Available from ftp://src.doc.ic.ac.uk/computing/archiving/mirror, January 1994.

[8] P. Mockapetris. Domain names - concepts and facilities. Internet Request for Comments RFC 1034, November 1987.

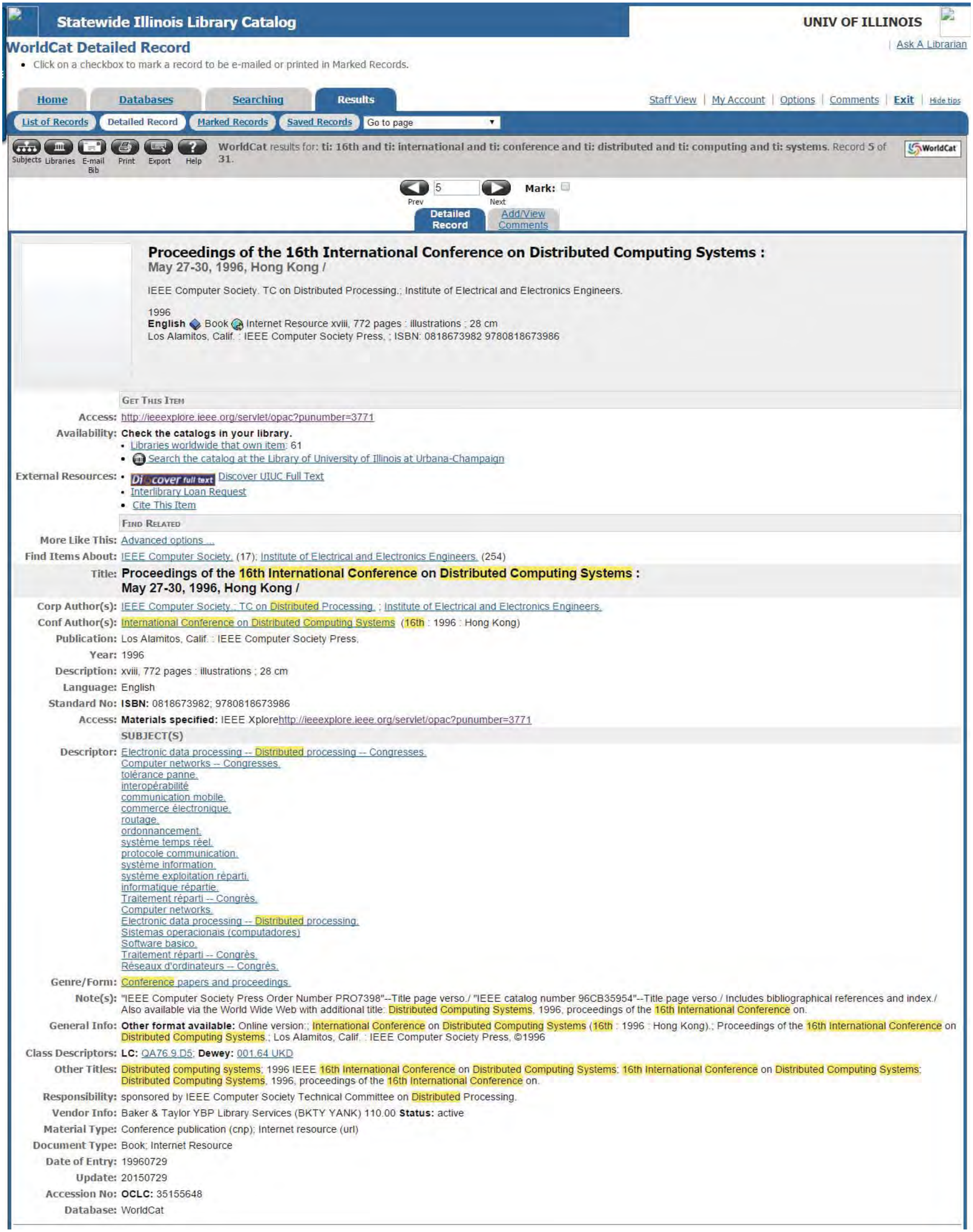
[9] K. Obraczka. *Massively Replicating Services in Wide-Area Internetworks*. PhD thesis, Computer Science Department, University of southern California, December 1994.

[10] D. Oppen and Y. Dalal. The Clearinghouse: A decentralized agent for locating named objects in a distributed environment. *ACM Transactions on Office Information Systems*, 1(3):230–253, July 1983.

[11] G. Popek, B. Walker, J. Chow, D. Edwards, C. Kline, and G. Rudisin ang G. Thiel. LOCUS: A network transparent, high reliability distributed system. *Proc. of the 8th. Symposium on Operating Systems Principles*, pages 169–177, December 1981.

[12] M. Satyanarayanan. Scalable, secure, and highly available distributed file access. *Computer Magazine*, 23(5):9–21, May 1990.

Attachment 3d: Statewide Illinois Library Catalog monograph record for the Proceedings of the 16th International Conference on Distributed Computing Systems



WorldCat Detailed Record

- Click on a checkbox to mark a record to be e-mailed or printed in Marked Records.

WorldCat results for: **ti: 16th and ti: international and ti: conference and ti: distributed and ti: computing and ti: systems**. Record 5 of 31.

Proceedings of the 16th International Conference on Distributed Computing Systems : May 27-30, 1996, Hong Kong /

IEEE Computer Society. TC on Distributed Processing.; Institute of Electrical and Electronics Engineers.

1996
English Book Internet Resource xviii, 772 pages : illustrations ; 28 cm
Los Alamitos, Calif. : IEEE Computer Society Press, ; ISBN: 0818673982 9780818673986

GET THIS ITEM

Access: http://ieeexplore.ieee.org/servlet/opac?punumber=3771

Availability: **Check the catalogs in your library.**
- Libraries worldwide that own item: 61
- Search the catalog at the Library of University of Illinois at Urbana-Champaign

External Resources:
- Discover full text Discover UIUC Full Text
- Interlibrary Loan Request
- Cite This Item

FIND RELATED

More Like This: Advanced options ...

Find Items About: IEEE Computer Society. (17); Institute of Electrical and Electronics Engineers. (254)

Title: **Proceedings of the 16th International Conference on Distributed Computing Systems : May 27-30, 1996, Hong Kong /**

Corp Author(s): IEEE Computer Society.; TC on Distributed Processing. ; Institute of Electrical and Electronics Engineers.

Conf Author(s): International Conference on Distributed Computing Systems (16th : 1996 : Hong Kong)

Publication: Los Alamitos, Calif. : IEEE Computer Society Press,

Year: 1996

Description: xviii, 772 pages : illustrations ; 28 cm

Language: English

Standard No: **ISBN:** 0818673982; 9780818673986

Access: **Materials specified:** IEEE Xplorehttp://ieeexplore.ieee.org/servlet/opac?punumber=3771

SUBJECT(S)

Descriptor: Electronic data processing -- Distributed processing -- Congresses.
Computer networks -- Congresses.
tolérance panne.
interopérabilité
communication mobile.
commerce électronique.
routage.
ordonnancement.
système temps réel.
protocole communication.
système information.
système exploitation réparti.
informatique répartie.
Traitement réparti -- Congrès.
Computer networks.
Electronic data processing -- Distributed processing.
Sistemas operacionais (computadores)
Software basico.
Traitement réparti -- Congrès.
Réseaux d'ordinateurs -- Congrès.

Genre/Form: Conference papers and proceedings.

Note(s): "IEEE Computer Society Press Order Number PRO7398"--Title page verso./ "IEEE catalog number 96CB35954"--Title page verso./ Includes bibliographical references and index./ Also available via the World Wide Web with additional title: Distributed Computing Systems, 1996, proceedings of the 16th International Conference on.

General Info: **Other format available:** Online version:; International Conference on Distributed Computing Systems (16th : 1996 : Hong Kong).; Proceedings of the 16th International Conference on Distributed Computing Systems.; Los Alamitos, Calif. : IEEE Computer Society Press, ©1996

Class Descriptors: **LC:** QA76.9.D5; **Dewey:** 001.64 UKD

Other Titles: Distributed computing systems; 1996 IEEE 16th International Conference on Distributed Computing Systems; 16th International Conference on Distributed Computing Systems; Distributed Computing Systems, 1996, proceedings of the 16th International Conference on.

Responsibility: sponsored by IEEE Computer Society Technical Committee on Distributed Processing.

Vendor Info: Baker & Taylor YBP Library Services (BKTY YANK) 110.00 **Status:** active

Material Type: Conference publication (cnp); Internet resource (url)

Document Type: Book; Internet Resource

Date of Entry: 19960729

Update: 20150729

Accession No: **OCLC:** 35155648

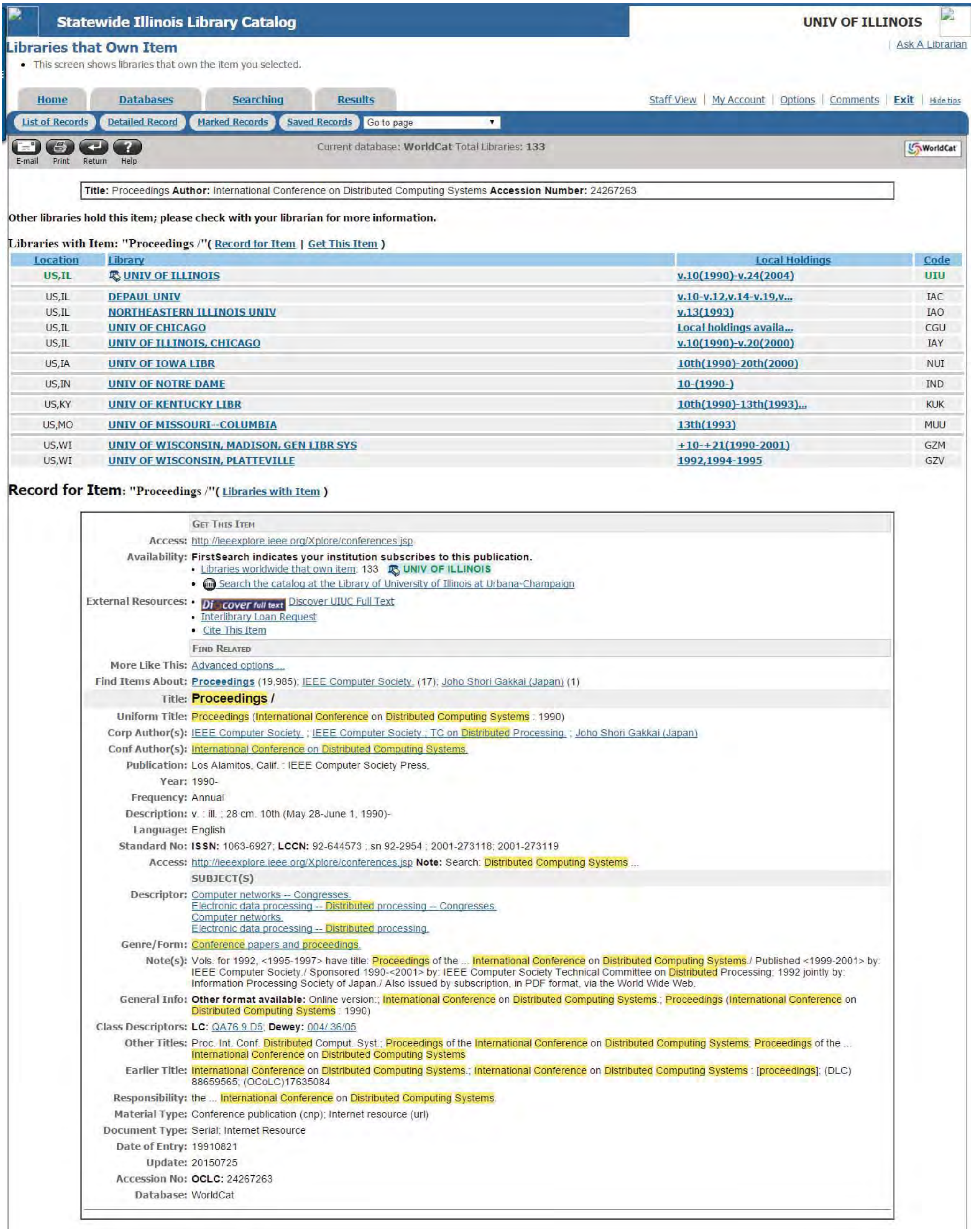
Database: WorldCat

Attachment 3d: Statewide Illinois Library Catalog monograph record for the Proceedings of the 16th International Conference on Distributed Computing Systems

Subjects
Libraries
E-mail Bib
Print
Export
Help
WorldCat
English | Español | Français | عربي | 日本語 | 한국어 | 中文(繁體) | 中文(简体) | Options | Comments | Exit
OCLC

WorldCat results for: **ti: 16th and ti: international and ti: conference and ti: distributed and ti: computing and ti: systems.** Record **5** of **31**.

Attachment 3e: Statewide Illinois Library Catalog series record for the Proceedings of the 16th International Conference on Distributed Computing Systems



Statewide Illinois Library Catalog

UNIV OF ILLINOIS

| Ask A Librarian

Libraries that Own Item

- This screen shows libraries that own the item you selected.

Home | Databases | Searching | Results

Staff View | My Account | Options | Comments | Exit | Hide tips

List of Records | Detailed Record | Marked Records | Saved Records | Go to page ▾

E-mail | Print | Return | Help

Current database: **WorldCat** Total Libraries: **133**

WorldCat

Title: Proceedings **Author:** International Conference on Distributed Computing Systems **Accession Number:** 24267263

Other libraries hold this item; please check with your librarian for more information.

Libraries with Item: "Proceedings /"(Record for Item | Get This Item)

Location	Library	Local Holdings	Code
US,IL	**UNIV OF ILLINOIS**	**v.10(1990)-v.24(2004)**	**UIU**
US,IL	DEPAUL UNIV	v.10-v.12,v.14-v.19,v...	IAC
US,IL	NORTHEASTERN ILLINOIS UNIV	v.13(1993)	IAO
US,IL	UNIV OF CHICAGO	Local holdings availa...	CGU
US,IL	UNIV OF ILLINOIS, CHICAGO	v.10(1990)-v.20(2000)	IAY
US,IA	UNIV OF IOWA LIBR	10th(1990)-20th(2000)	NUI
US,IN	UNIV OF NOTRE DAME	10-(1990-)	IND
US,KY	UNIV OF KENTUCKY LIBR	10th(1990)-13th(1993)...	KUK
US,MO	UNIV OF MISSOURI--COLUMBIA	13th(1993)	MUU
US,WI	UNIV OF WISCONSIN, MADISON, GEN LIBR SYS	+10-+21(1990-2001)	GZM
US,WI	UNIV OF WISCONSIN, PLATTEVILLE	1992,1994-1995	GZV

Record for Item: "Proceedings /"(Libraries with Item)

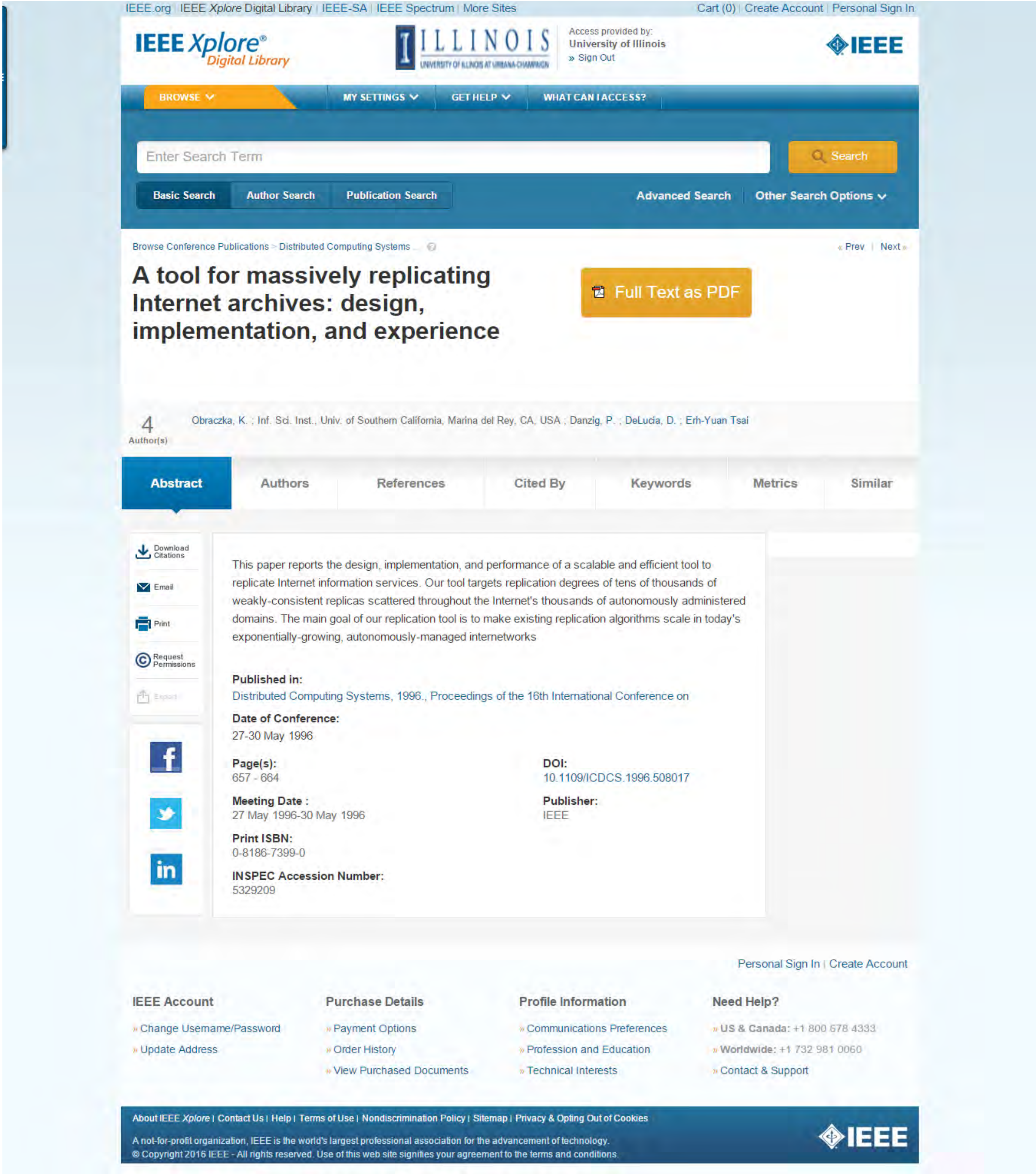
	GET THIS ITEM
Access:	http://ieeexplore.ieee.org/Xplore/conferences.jsp
Availability:	**FirstSearch indicates your institution subscribes to this publication.** • Libraries worldwide that own item: 133 UNIV OF ILLINOIS • Search the catalog at the Library of University of Illinois at Urbana-Champaign
External Resources:	• Discover full text Discover UIUC Full Text • Interlibrary Loan Request • Cite This Item
	FIND RELATED
More Like This:	Advanced options ...
Find Items About:	Proceedings (19,985); IEEE Computer Society. (17); Joho Shori Gakkai (Japan) (1)
Title:	**Proceedings /**
Uniform Title:	Proceedings (International Conference on Distributed Computing Systems : 1990)
Corp Author(s):	IEEE Computer Society. ; IEEE Computer Society ; TC on Distributed Processing. ; Joho Shori Gakkai (Japan)
Conf Author(s):	International Conference on Distributed Computing Systems.
Publication:	Los Alamitos, Calif. : IEEE Computer Society Press,
Year:	1990-
Frequency:	Annual
Description:	v. : ill. ; 28 cm. 10th (May 28-June 1, 1990)-
Language:	English
Standard No:	**ISSN:** 1063-6927; **LCCN:** 92-644573 ; sn 92-2954 ; 2001-273118; 2001-273119
Access:	http://ieeexplore.ieee.org/Xplore/conferences.jsp **Note:** Search: Distributed Computing Systems ...
	SUBJECT(S)
Descriptor:	Computer networks -- Congresses. Electronic data processing -- Distributed processing -- Congresses. Computer networks. Electronic data processing -- Distributed processing.
Genre/Form:	Conference papers and proceedings.
Note(s):	Vols. for 1992, <1995-1997> have title: Proceedings of the ... International Conference on Distributed Computing Systems./ Published <1999-2001> by: IEEE Computer Society./ Sponsored 1990-<2001> by: IEEE Computer Society Technical Committee on Distributed Processing; 1992 jointly by: Information Processing Society of Japan./ Also issued by subscription, in PDF format, via the World Wide Web.
General Info:	**Other format available:** Online version:; International Conference on Distributed Computing Systems.; Proceedings (International Conference on Distributed Computing Systems : 1990)
Class Descriptors:	**LC:** QA76.9.D5; **Dewey:** 004/.36/05
Other Titles:	Proc. Int. Conf. Distributed Comput. Syst.; Proceedings of the International Conference on Distributed Computing Systems; Proceedings of the ... International Conference on Distributed Computing Systems
Earlier Title:	International Conference on Distributed Computing Systems.; International Conference on Distributed Computing Systems : [proceedings]; (DLC) 88659565; (OCoLC)17635084
Responsibility:	the ... International Conference on Distributed Computing Systems.
Material Type:	Conference publication (cnp); Internet resource (url)
Document Type:	Serial; Internet Resource
Date of Entry:	19910821
Update:	20150725
Accession No:	**OCLC:** 24267263
Database:	WorldCat

Attachment 3e: Statewide Illinois Library Catalog series record for the Proceedings of the 16th International Conference on Distributed Computing Systems

E-mail
Print
Return
Help
WorldCat
English | Español | Français | عربي | 日本語 | 한국어 | 中文(繁體) | 中文(简体) | Options | Comments | Exit
OCLC

Current database: **WorldCat** Total Libraries: **133**

Attachment 3f: IEEE Xplore Digital Library index record for Document 3


IEEE.org | IEEE Xplore Digital Library | IEEE-SA | IEEE Spectrum | More Sites
Cart (0) | Create Account | Personal Sign In
IEEE Xplore® Digital Library
ILLINOIS UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
Access provided by:
University of Illinois
» Sign Out
IEEE
BROWSE
MY SETTINGS
GET HELP
WHAT CAN I ACCESS?
Enter Search Term
Search
Basic Search
Author Search
Publication Search
Advanced Search
Other Search Options
Browse Conference Publications > Distributed Computing Systems
« Prev | Next »
Full Text as PDF

A tool for massively replicating Internet archives: design, implementation, and experience

4 Author(s)

Obraczka, K. ; Inf. Sci. Inst., Univ. of Southern California, Marina del Rey, CA, USA ; Danzig, P. ; DeLucia, D. ; Erh-Yuan Tsai

This paper reports the design, implementation, and performance of a scalable and efficient tool to replicate Internet information services. Our tool targets replication degrees of tens of thousands of weakly-consistent replicas scattered throughout the Internet's thousands of autonomously administered domains. The main goal of our replication tool is to make existing replication algorithms scale in today's exponentially-growing, autonomously-managed internetworks

Published in:
Distributed Computing Systems, 1996., Proceedings of the 16th International Conference on

Date of Conference:
27-30 May 1996

Page(s):
657 - 664

DOI:
10.1109/ICDCS.1996.508017

Meeting Date :
27 May 1996-30 May 1996

Publisher:
IEEE

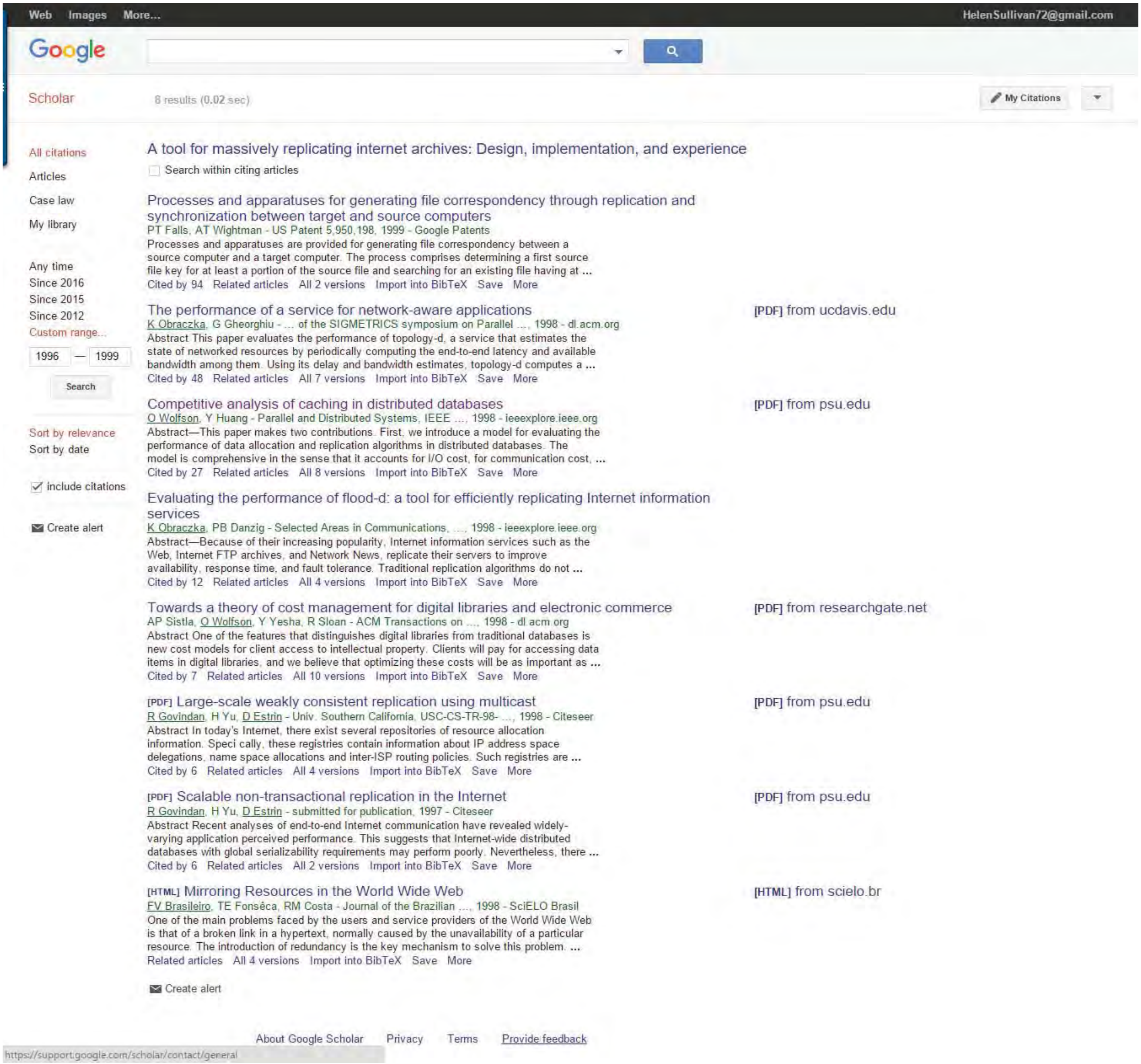
Print ISBN:
0-8186-7399-0

INSPEC Accession Number:
5329209

Personal Sign In | Create Account
IEEE Account
» Change Username/Password
» Update Address
Purchase Details
» Payment Options
» Order History
» View Purchased Documents
Profile Information
» Communications Preferences
» Profession and Education
» Technical Interests
Need Help?
» US & Canada: +1 800 678 4333
» Worldwide: +1 732 981 0060
» Contact & Support
About IEEE Xplore | Contact Us | Help | Terms of Use | Nondiscrimination Policy | Sitemap | Privacy & Opting Out of Cookies

A not-for-profit organization, IEEE is the world's largest professional association for the advancement of technology.

Attachment 3g: List of publications citing Document 3 identified by Google Scholar



A tool for massively replicating internet archives: Design, implementation, and experience

□ Search within citing articles

Processes and apparatuses for generating file correspondency through replication and synchronization between target and source computers
PT Falls, AT Wightman - US Patent 5,950,198, 1999 - Google Patents
Processes and apparatuses are provided for generating file correspondency between a source computer and a target computer. The process comprises determining a first source file key for at least a portion of the source file and searching for an existing file having at ...
Cited by 94 Related articles All 2 versions Import into BibTeX Save More

The performance of a service for network-aware applications — [PDF] from ucdavis.edu
K Obraczka, G Gheorghiu - ... of the SIGMETRICS symposium on Parallel ..., 1998 - dl.acm.org
Abstract This paper evaluates the performance of topology-d, a service that estimates the state of networked resources by periodically computing the end-to-end latency and available bandwidth among them. Using its delay and bandwidth estimates, topology-d computes a ...
Cited by 48 Related articles All 7 versions Import into BibTeX Save More

Competitive analysis of caching in distributed databases — [PDF] from psu.edu
O Wolfson, Y Huang - Parallel and Distributed Systems, IEEE ..., 1998 - ieeexplore.ieee.org
Abstract—This paper makes two contributions. First, we introduce a model for evaluating the performance of data allocation and replication algorithms in distributed databases. The model is comprehensive in the sense that it accounts for I/O cost, for communication cost, ...
Cited by 27 Related articles All 8 versions Import into BibTeX Save More

Evaluating the performance of flood-d: a tool for efficiently replicating Internet information services
K Obraczka, PB Danzig - Selected Areas in Communications, ..., 1998 - ieeexplore.ieee.org
Abstract—Because of their increasing popularity, Internet information services such as the Web, Internet FTP archives, and Network News, replicate their servers to improve availability, response time, and fault tolerance. Traditional replication algorithms do not ...
Cited by 12 Related articles All 4 versions Import into BibTeX Save More

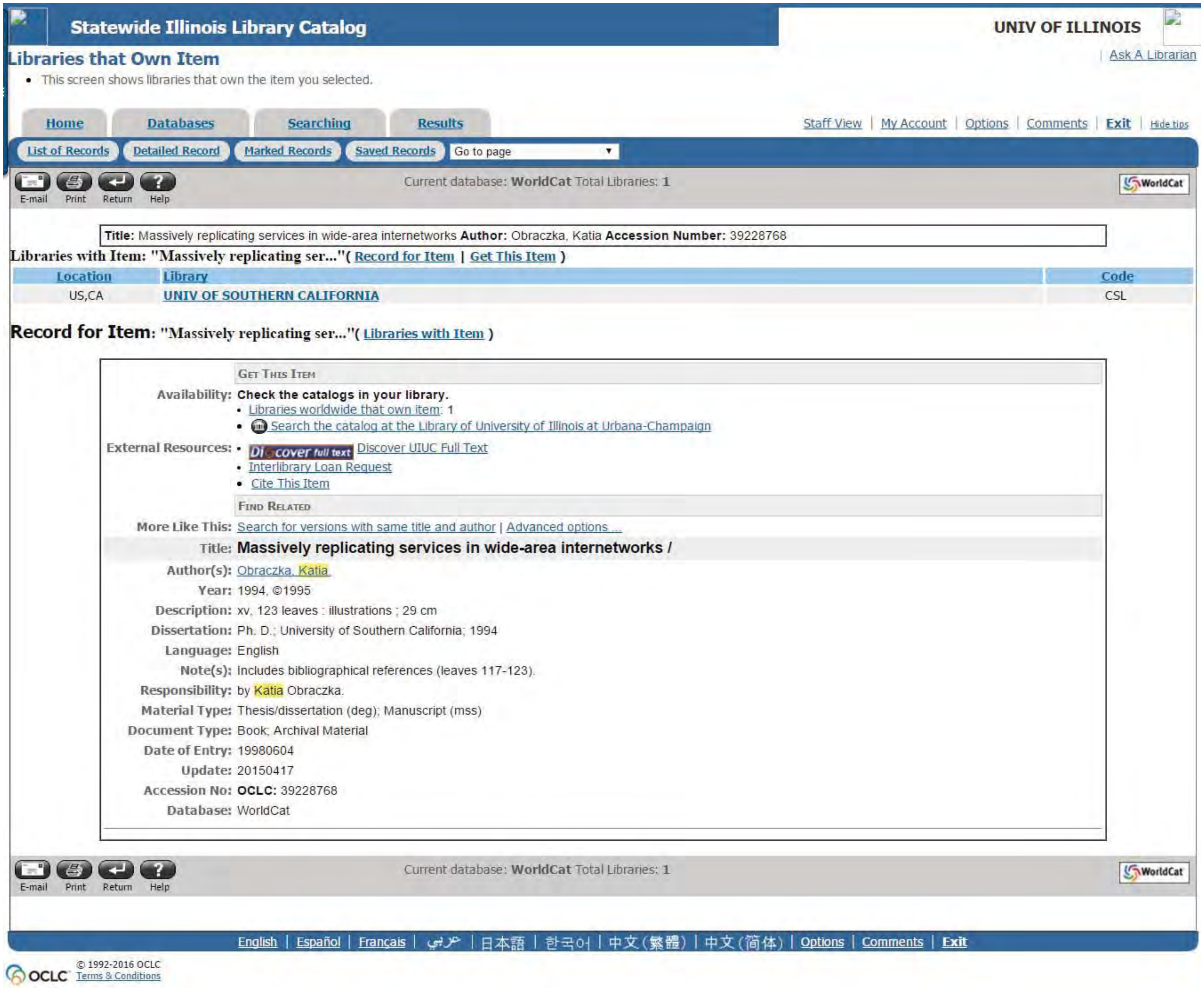
Towards a theory of cost management for digital libraries and electronic commerce — [PDF] from researchgate.net
AP Sistla, O Wolfson, Y Yesha, R Sloan - ACM Transactions on ..., 1998 - dl.acm.org
Abstract One of the features that distinguishes digital libraries from traditional databases is new cost models for client access to intellectual property. Clients will pay for accessing data items in digital libraries, and we believe that optimizing these costs will be as important as ...
Cited by 7 Related articles All 10 versions Import into BibTeX Save More

[PDF] Large-scale weakly consistent replication using multicast — [PDF] from psu.edu
R Govindan, H Yu, D Estrin - Univ. Southern California, USC-CS-TR-98- ..., 1998 - Citeseer
Abstract In today's Internet, there exist several repositories of resource allocation information. Speci cally, these registries contain information about IP address space delegations, name space allocations and inter-ISP routing policies. Such registries are ...
Cited by 6 Related articles All 4 versions Import into BibTeX Save More

[PDF] Scalable non-transactional replication in the Internet — [PDF] from psu.edu
R Govindan, H Yu, D Estrin - submitted for publication, 1997 - Citeseer
Abstract Recent analyses of end-to-end Internet communication have revealed widely-varying application perceived performance. This suggests that Internet-wide distributed databases with global serializability requirements may perform poorly. Nevertheless, there ...
Cited by 6 Related articles All 2 versions Import into BibTeX Save More

[HTML] Mirroring Resources in the World Wide Web — [HTML] from scielo.br
FV Brasileiro, TE Fonsêca, RM Costa - Journal of the Brazilian ..., 1998 - SciELO Brasil
One of the main problems faced by the users and service providers of the World Wide Web is that of a broken link in a hypertext, normally caused by the unavailability of a particular resource. The introduction of redundancy is the key mechanism to solve this problem. ...
Related articles All 4 versions Import into BibTeX Save More

Attachment 4a: Statewide Illinois Library Catalog record for Document 4


Statewide Illinois Library Catalog
UNIV OF ILLINOIS
| Ask A Librarian
Home
Databases
Searching
Results
Staff View | My Account | Options | Comments | Exit | Hide tips
List of Records
Detailed Record
Marked Records
Saved Records
Go to page
E-mail
Print
Return
Help
WorldCat
English | Español | Français | عربي | 日本語 | 한국어 | 中文(繁體) | 中文(简体) | Options | Comments | Exit
OCLC
© 1992-2016 OCLC
Terms & Conditions

Libraries that Own Item

- This screen shows libraries that own the item you selected.

Current database: **WorldCat** Total Libraries: **1**

Title: Massively replicating services in wide-area internetworks **Author:** Obraczka, Katia **Accession Number:** 39228768

Libraries with Item: "Massively replicating ser..."(Record for Item | Get This Item)

Location	Library	Code
US,CA	UNIV OF SOUTHERN CALIFORNIA	CSL

Record for Item: "Massively replicating ser..."(Libraries with Item)

	GET THIS ITEM
Availability:	**Check the catalogs in your library.** • Libraries worldwide that own item: 1 • Search the catalog at the Library of University of Illinois at Urbana-Champaign
External Resources:	• Discover full text Discover UIUC Full Text • Interlibrary Loan Request • Cite This Item
	FIND RELATED
More Like This:	Search for versions with same title and author | Advanced options ...
Title:	**Massively replicating services in wide-area internetworks /**
Author(s):	Obraczka, Katia
Year:	1994, ©1995
Description:	xv, 123 leaves : illustrations ; 29 cm
Dissertation:	Ph. D.; University of Southern California; 1994
Language:	English
Note(s):	Includes bibliographical references (leaves 117-123).
Responsibility:	by Katia Obraczka.
Material Type:	Thesis/dissertation (deg); Manuscript (mss)
Document Type:	Book; Archival Material
Date of Entry:	19980604
Update:	20150417
Accession No:	**OCLC:** 39228768
Database:	WorldCat

Current database: **WorldCat** Total Libraries: **1**

MASSIVELY REPLICATING SERVICES IN WIDE-AREA INTERNETWORKS

by

Katia Obraczka

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(Electrical Engineering)

December 1994

UNIVERSITY OF SOUTHERN CALIFORNIA
THE GRADUATE SCHOOL
UNIVERSITY PARK
LOS ANGELES, CALIFORNIA 90007

This dissertation, written by

Katia Obraczka

under the direction of her........ Dissertation Committee, and approved by all its members, has been presented to and accepted by The Graduate School, in partial fulfillment of requirements for the degree of

DOCTOR OF PHILOSOPHY

Alice C. Parker
....
Dean of Graduate Studies

Date December 7, 1994

DISSERTATION COMMITTEE

Peter B Danzig
.... *Chairperson*

....

....

Dedication

To my husband, grandparents,
parents, sister, and brothers.

Acknowledgments

This was the most difficult and at the same time the most enjoyable section to write. As I write it, I realize that it could well be the longest section in the dissertation, since I would like to acknowledge all the wonderful people I have met and interacted throughout my life. Unfortunately, for lack of space, I do not explicitly mention all their names.

First and foremost, I would like to express my sincere thanks to my advisor, Dr. Peter Danzig, for his guidance, encouragement, support, and friendship. I was very fortunate to have had the opportunity to work with him during the past 4 years. He has contributed greatly to this dissertation and my maturity as a researcher. I cannot imagine how a professor could be more dedicated to his students. He will always serve as a model to me.

I wish to thank the members of my qualifying and dissertation committees: Deborah Estrin, Clifford Neumann, Shahram Ghandeharizadeh, and John Silvester. I would especially like to thank Professors Estrin and Silvester for their helpful comments and discussions.

I would also like to acknowledge the Brazilian Education Ministry which provided me with a four-year graduate fellowship as a starting PhD student. This research was supported by the Advanced Research Projects Agency under contract number DABT63-93-C-0052.

Several people have assisted in this research. I would like to acknowledge Dante DeLucia for his implementation of *flood-d* and *mirror-d*. Kitinon Wangpattanamongkol has implemented the logical topology calculation algorithm, and Steve Miller has developed the simulation package I used to build my simulators.

I sincerely thank all the members, old and new, faculty and students, of the Network and Distributed Systems Laboratory at USC. I was fortunate enough to be a member of this friendly, enjoying, and stimulating research community. They

definitely made these years at USC a lot more fun. In particular, I would like to thank Prof. Rafael Saavedra, Gene Tsudik, Abhjit Khale, Lee Breslau, Steve Hotz, Doug Fang, Danny Mitzel, Ron Cocchi, Sugih Jamin, Shih-Hao Li, Jong-Suk Ahn, Daniel Zappala, Brenda Timmerman, John Noll, Kraig Meyer, Louie Ramos and Ari Medvinski. Finally, Gary Frenkel, whose memory provided inspiration to move on and face whatever challenges may appear.

I also wish to thank several friends, and fellow graduate students, whose friendship and support made the cultural chock effects resulting from going to graduate school in a foreign country a learning instead of a painful experience. I would especially like to thank Justine Gilman, Patricia Goldweic, Alfredo Weitzenfeld, Eve Schooler, Bob Felderman, and Steve Schrader.

I was also very fortunate to have a family that has always encouraged and supported me. I will always be grateful to my parents, Bertha and Jayme, who have always given me more than I could ever have asked for. My sister Sandra, and my brothers Marcelo, Ricardo, and Eduardo have always contributed and participated in whatever I set out to accomplish.

Finally, I would also like to thank my husband, Mendel, for the love, support, and encouragement which made it possible for me to overcome the obstacles in the life of a graduate student. He has put up with my fears and frustrations, and my finishing the PhD program is as much his accomplishment as it is mine.

Contents

List Of Tables

List Of Figures