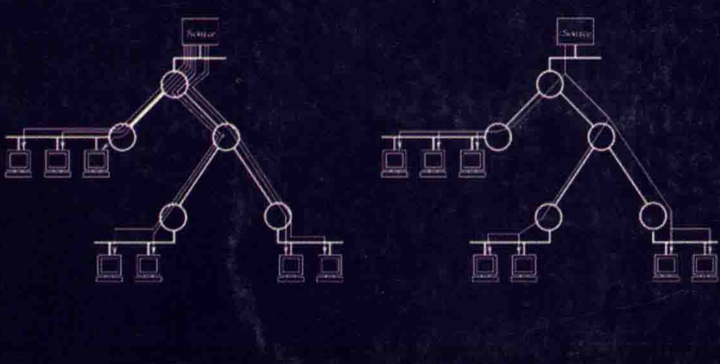
Tutorial

LNCS 2497

Enrico Gregori
Giuseppe Anastasi
Stefano Basagni (Eds.)

Advanced Lectures on Networking

NETWORKING 2002 Tutorials



Springer

Enrico Gregori Giuseppe Anastasi
Stefano Basagni (Eds.)

Advanced Lectures on Networking

NETWORKING 2002 Tutorials

Springer

Volume Editors

Enrico Gregori
CNR-IIT Institute
Via G. Moruzzi, 1, 56124 Pisa, Italy
E-mail: enrico.gregori@iit.cnr.it

Giuseppe Anastasi
University of Pisa, Department of Information Engineering
Via Diotisalvi 2, 56122 Pisa, Italy
E-mail: g.anastasi@iet.unipi.it

Stefano Basagni
Northeastern University, DANA 312
Department of Electrical and Computer Engineering
360 Huntington Ave., Boston, MA 02115, USA
E-mail: basagni@ece.neu.edu

Cataloging-in-Publication Data applied for

A catalog record for this book is available from the Library of Congress

Bibliographic information published by Die Deutsche Bibliothek
Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliographie; detailed bibliographic data is available in the Internet at <http://dnd.ddb.de>.

CR Subject Classification (1998): C.2, H.3.4-7, H.4, H.5.1

ISSN 0302-9743
ISBN 3-540-00165-4 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Printed in Germany

Typesetting: Camera-ready by author, data conversion by PTP-Berlin, Stefan Sossna e.K.
Printed on acid-free paper SPIN: 10870724 06/3142 5 4 3 2 1 0

Lecture Notes in Computer Science 2497
Edited by G. Goos, J. Hartmanis, and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Preface

This book of proceedings brings together papers on the topics of the tutorials given at the second IFIP-TC6 networking conference "Networking 2002", held in Pisa, Italy, on May 19–24, 2002.

The aim of this book is to provide overviews and surveys on prevailing topics in the field of networking, to sum up and complement what the speakers conveyed in their tutorials. Despite the vast spectrum of topics in networking, we were pleased to receive proposals from leading researchers in areas of great interest. The papers collected in this book attest to the great variety of topics and the richness of current research and trends in areas that span from mobile computing and wireless networking to network security and optical networks.

The reader unfamiliar with the specific topic of a tutorial will find the papers presented in this book useful as pointers to the basics on each topic. Researchers in the area of a given tutorial will find here the latest trends and achievements on the tutorial topic.

The general chair and the tutorial co-chairs would like to thank here all the tutorial speakers, whose effort and cooperation made this "extra" book possible.

May 2002

Enrico Gregori
Giuseppe Anastasi
Stefano Basagni

Organizers

Sponsoring Institutions

Networking 2002 Organization Committee

Conference Executive Committee

General Chair:
Enrico Gregori, National Research Council, Italy

General Vice-Chair:
Ioannis Stavrakakis, University of Athens, Greece

Technical Program Chair:
Marco Conti, National Research Council, Italy

Special Track Chair for Networking Technologies, Services and Protocols:
Andrew T. Campbell, Columbia University, USA

Special Track Chair for Performance of Computer and Communication Networks:
Moshe Zukerman, University of Melbourne, Australia

Special Track Chair for Mobile and Wireless Communications:
Guy Omidyar, National University of Singapore

Tutorial Program Co-Chairs:
Giuseppe Anastasi, University of Pisa, Italy
Stefano Basagni, Northeastern University, USA

Workshop Chairs:

Workshop 1 — *Web Engineering*
Ludmilla Cherkasova, Hewlett-Packard Labs, USA
Fabio Panzieri, University of Bologna, Italy

Workshop 2 — *Peer-to-Peer Computing*
Gianpaolo Cugola, Politecnico di Milano, Italy
Gian Pietro Picco, Politecnico di Milano, Italy

Workshop 3 — *IP over WDM*
Giancarlo Prati, Scuola Superiore S. Anna, Italy
Piero Castoldi, Scuola Superiore S. Anna, Italy

Invited Speaker Chair:
Fabrizio Davide, PhD, Telecom Italia SpA, Italy

Organization Chair:
Stefano Giordano, University of Pisa, Italy

Publicity Chair:
Silvia Giordano, Federal Inst. of Technology Lausanne (EPFL), Switzerland
Laura Feeney, SICS, Sweden

Steering Committee Chair:
Harry Perros, North Carolina State University, USA

Steering Committee Members:
Augusto Casaca, IST/INESC, Portugal
S.K. Das, The University of Texas at Arlington, USA
Erol Gelenbe, University of Central Florida, USA
Harry Perros, NCSU, USA (Chair)
Guy Pujolle, University of Paris 6, France
Harry Rudin, Switzerland
Jan Slavik, TESTCOM, Czech Republic
Hideaki Takagi, University of Tsukuba, Japan
Samir Thome, ENST, France
Adam Wolisz, TU-Berlin, Germany

Electronic Submission:
Alessandro Urpi, University of Pisa, Italy

Web Designer:
Patrizia Andronico, IAT–CNR, Italy

Local Organizing Committee:
Renzo Beltrame, CNUCE–CNR, Italy
Raffaele Bruno, CNUCE–CNR, Italy
Willy Lapenna, CNUCE–CNR, Italy
Gaia Maselli, CNUCE–CNR, Italy
Renata Bandelloni, CNUCE–CNR, Italy

Table of Contents

for some number of clients; the end system is therefore a single bottleneck for performance and reliability. Server sites may make use of a number of techniques to increase reliability and performance, such as replication, load balancing and request routing; at some point along the evolution of this thinking, it is a natural step to include the client's resources in the system – the mutual benefits in a large system, especially given the spare resources of today's clients, are clear.

Thus peer-to-peer systems emerge out of client-server systems by removing the asymmetry in rôles: a client is also a server, and allows access to its resources by other systems.

A claim sometimes made about peer-to-peer systems is that they no longer have *any* distinguished node, and thus are highly fault tolerant and have very good performance and scaling properties. We will see that this claim has some truth to it, although there are plenty of peer-to-peer systems that have some level of distinguished nodes, and also plenty of peer-to-peer systems that have performance limitations. In fact, the fault tolerance claims are hardly born out at all in the early instantiations of the peer-to-peer movement. Initial availability figures in Napster[1], Gnutella[2] and Freenet[3] do not compare favourably with even the most humble of web sites!

However, second and later generation systems do indeed provide the claimed functionality and performance gains, and we will see in Pastry[40], Chord[11] and CAN[12] very promising results, and even more recent work building applications and services over these systems shows great potential gains[16].

At the same time as peer-to-peer, a number of network researchers have been frustrated in their attempts to deliver new network services within the context of traditional telecommunications or Internet networks[2]. Instead, researchers have built experimental infrastructures by constructing *overlay* systems. An overlay may be a simple as a collection of static IP in IP tunnels, or as complex as a full dynamic VPN ("virtual private network"). Some of these systems are in use in the Active Networks research community. Others are more straightforward in their motivation, such as the GRID communities' requirements for more robust Internet connectivity!

The classical distributed systems community would claim that many of these ideas were present in the early work on fault tolerant systems in the 1970s. For example the Xerox Network System's name service, *Grapevine*[43] included many of the same traits[44] as systems mentioned above and discussed below. Other systems that could easily be construed as architecturally true peer-to-peer systems include Net News (NNTP is certainly not client-server) and the Web's Inter-cache protocol, ICP. The Domain Name System also includes Zone Transfers and other mechanisms which are not part of its normal client-server resolver behaviour.

Also, IP itself is built out of a set of autonomous routers running a peer-to-peer protocol (for example, the routing protocols OSPF and BGP are peer-

[2] New Internet network level services such as IP QoS in the form of integrated services, and differentiated services, as well as novel service models such as multicast and mobility have proved notoriously hard to build and deploy in their native forms.

to-peer, and certainly not client-server, and are designed that way for exactly the reasons given above for peer-to-peer); not only that, but IP was originally an overlay service , implemented above other layered communications system: the PSTN, ARPANET and X.25 circuit switched networks. Indeed, this overlay model keeps re-emerging as network operators deploy faster switched infrastructures such as Frame Relay, ATM and WDM and PONS (Pure Optical Networked Systems) core networks. Most of the peer-to-peer work that we discuss in this document is positioned at the Application Layer, as illustrated in Figure 1.

One can look at differences and similarities between classical client-server and modern peer-to-peer systems on another axis: statefulness. Despite successes with stateless servers, many Web servers use cookies and other mechanisms (Web Services) to keep state over various transactions with a client. Peer-to-peer systems may keep track of each other although the implications of the privacy/anonymity models we discuss below mitigate this.

Yet another viewpoint from which one can dissect these systems is that of the use of intermediaries. In the Web (and client-server file systems such as NFS and AFS) we use caches to improve average latency and to reduce networking load. Peer-to-peer systems (particularly Freenet[3]) offer intermediary services very naturally, although again, anonymity (and locality) may not necessarily be preserved.

As an aside, we would say that since this topic seems to recur every few years, there appears to be something compelling and unavoidable about peer-to-peer and overlays.

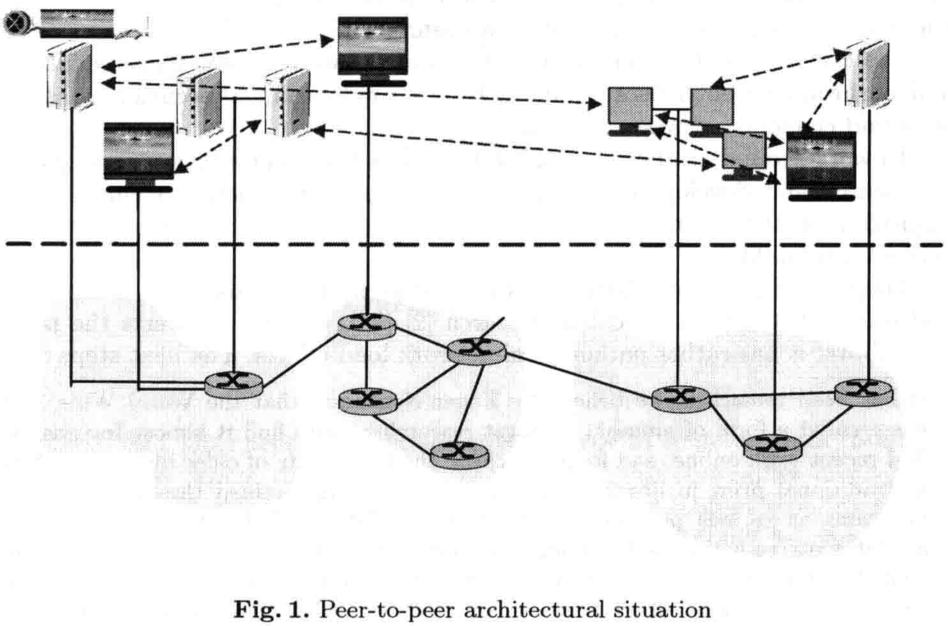


Fig. 1. Peer-to-peer architectural situation

In the next section we take a further look at both this recent, and longer term history in a little more detail.

1.2 History

While we could talk about the 1970s and 1980s work on autonomous distributed systems, or the use of ideas from those in other areas such as IP routing, here we choose mainly to look at the recent peer-to-peer work. [3] Lest readers think *we* are not aware of the longer history of peer-to-peer protocols, we cite the work of 1215[46].

Peer-to-peer "year zero" can effectively be set to Napster[1]. Napster was used heavily to share music content. This is a political hot-potato – the music industry was (and still is at the time of writing) very slow to make content available through the Internet, perhaps due to lack of perception of the market, perhaps due to lack of efficient rights management, and content tracking and protection mechanisms. This seems commercially naïve, given that a large part of the customer base were early network adopters with low tolerance to profiteering. Indeed, the media industry as a whole has preferred to pursue legal mechanisms to protect its legacy hard copy business than to pursue a lower cost (but not necessarily lower margin or profit) soft copy distribution business – the fear of copying would be plausible if it were not for the fact that this risk already exists, so the lack of availability of online music services until fairly recently is all the more remarkable[4].

Napster was not just a system architecture. It also became a company which provided a central directory. This directory is where file listings are uploaded by peers, and where lookups go to. The true behaviour is that of alternating client-server rôles. This is illustrated in Figures 2, 3, 4 and 5.

The central directory server exhibits problems which are not just reliability and performance bottlenecks, but are also a single point of security, political, legal and economic attacks.

To work around all these problems, there has been an evolutionary sequence of research and development of next generation peer-to-peer systems, largely improving on the lookup mechanism, but then moving on to better and better content availability.

The first of these was Gnutella, which dispensed with the central directory and replaced it with a flood based search [2] [35]. While this averts the problems above, it has rather pathological network load effects. The next steps were

[3] It has been remarked, we believe by Roger Needham, that the World Wide Web has created a form of amnesia amongst researchers who find it almost too easy to find recent work online, and forget to check the large body of older literature offline in traditional print in libraries. Luckily, this is being rectified through concerted programs at various publishers, notably the ACM and IEEE to scan and make available online entire back catalogs. However, one can see several instances of the lack of historical perspective today in papers which cite nothing before WWW year zero, submitted to recent conferences in which the authors have been involved.

[4] One might observe that the best way to ensure that kids do something is to ban it!

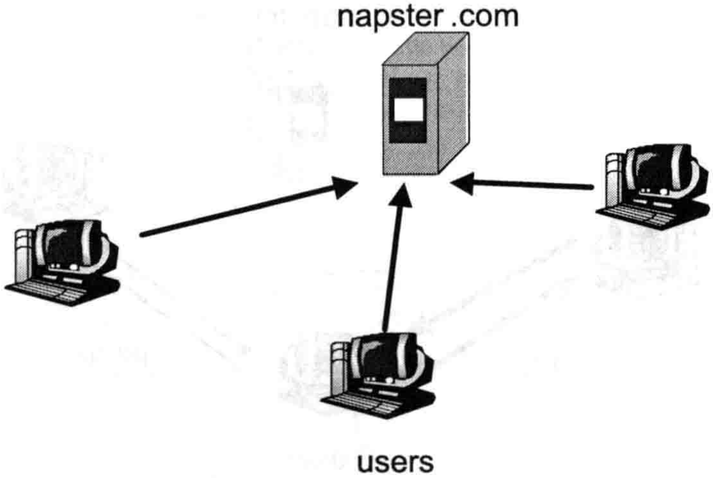


Fig. 2. Napster Central Directory Use, Directory Upload

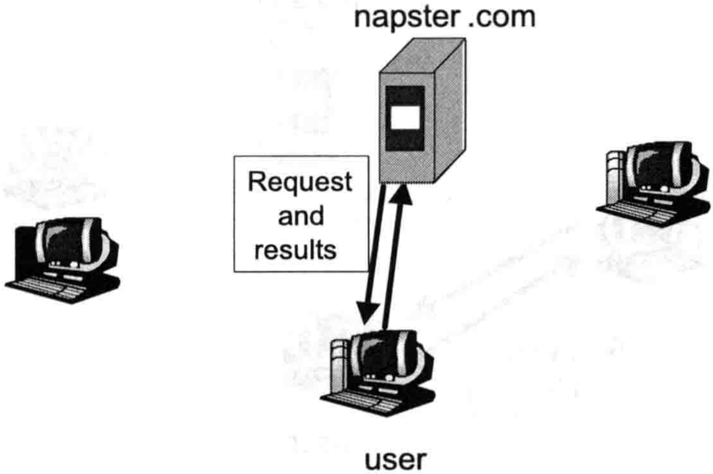


Fig. 3. Napster Central Directory Use, Item Lookup

Freenet[3] and Morpheus[5] which add mechanisms to *route* requests to an appropriate node where the content is *likely* to be. The techniques involve two separate mechanisms: the first makes use of hashing on content keys to derive a node number. This is routed to nodes which are where content keys are stored; the second part of the mechanism is to carry out demand caching. The last phase of Freenet that is particularly interesting in that content is encrypted so that servers (oops, sorry, *peers*) do not even *know* what content they store. This is

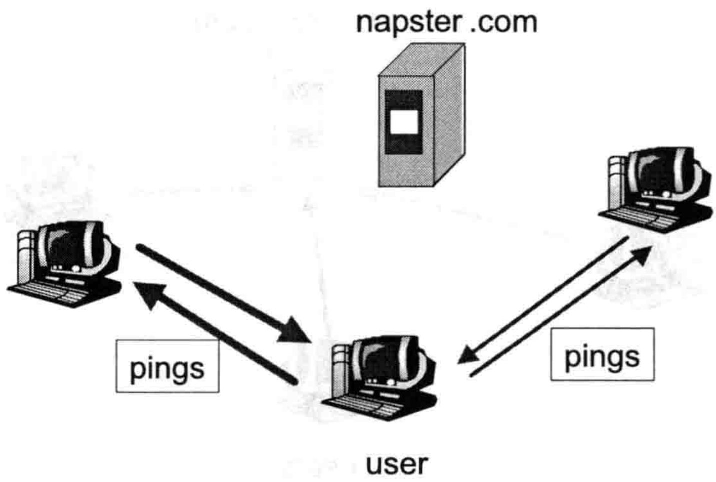


Fig. 4. Napster Peer-to-Peer phase, Peer Selection

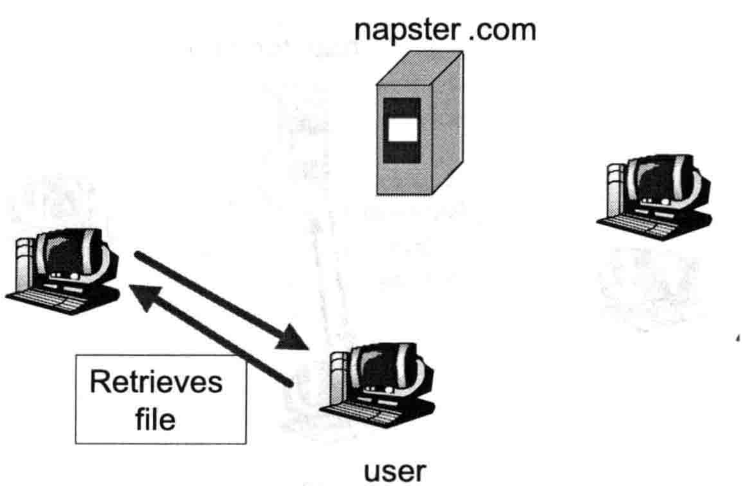


Fig. 5. Napster Peer-to-Peer phase, Content Retrieval

similar to the Eternity[45] file system which predates most peer-to-peer systems by a few years.

A web based service that has many of the same characteristics as Eternity is the Publius system[24], which we mention for interest.

The commonality of the use of Java in some research projects, combined with the observation that there are enough common components in peer-to-peer development activities, has led to some common toolkits for future work – one such is JXTA[4] from Sun.

Overlays. As the ease of development and deployment of peer-to-peer became clearer to the network research community, the use of what one might call "multi-hop" applications, or more normally *overlays*, has started to take off in a big way.

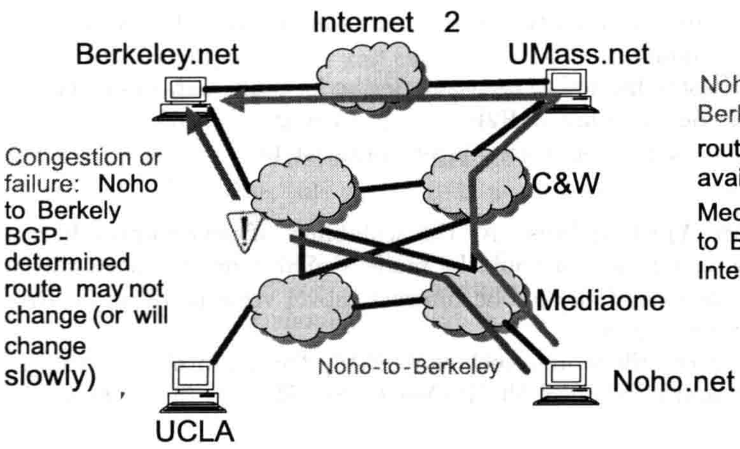


Fig. 6. RON

Balakrishnan *et. al.* at MIT have developed the Resilient Overlay Network system[6,33]. This idea is illustrated in Figure 6, where a number of sites collaborate to find a longer path at the IP "level", which has better properties (such as throughput or loss) at the application level, by dynamically routing via a set of dynamically created tunnels. In parallel, Turner *et. al.* at Washington developed a similar approach for multicast[30].

The difficulties in deployment of native IP multicast have led several groups, most notably the CMU group have developed an "End System only Multicast" system which constructs distribution trees for applications by running a routing algorithm for a degree-constrained multicast tree without including any intermediate IP routers, but with fairly limited negative impact on the efficiency of the distribution tree [7,8]. Further work used this to implement multimedia conferencing systems and demonstrated the workability and usability of the scheme [9]. A group at Stanford also used this approach for streaming media [28].

Other researchers have used the approach for anycast [14], and server selection [15].

Next Generation. Dissatisfied with the poor resilience and network load caused by super-nodes and flooding, respectively, a number of researchers have been working on distributing the directory for peer-to-peer storage systems.

The main approach is to implement a distributed hash table. There are then a number of design choices as to how the keys are distributed amongst nodes, and how requests are routed to the appropriate node(s).

The principle schemes are Pastry[40], Chord[11], CAN[12] and Tapestry[10].

Chord uses a clever *fingertable* to reduce the size and failure probability of the directory. Pastry uses a similar scheme but with network locality hints. CAN uses several hashes, to map into a multi-dimensional node id space. Tapestry creates an inverted index of nodes.

The success of Pastry has led to its use in Scribe for multicast above the level of a peer-to-peer service (overlay on P2P!)[22]. In turn, the creators of CAN have also investigated its use for application layer multicast[13].

Next Generation Applications. As the scalability of peer-to-peer became apparent, and the resources that could be made available on the large number of systems on Intranets and the public Internet, novel versions of applications on P2P have been developed.

These include object/file stores, such as ASF[16], Oceanstore[18], PAST[29], and databases[31], and even novel VOIP (Voice Over IP) routing systems such as Bazooka[32].

Measurement and Security. There is an interesting tension in measurement work in peer-to-peer systems. Due to their open nature, one can introduce instrumented nodes very easily. Due to the anonymity of some of the systems, it can be hard to ascertain the real behaviour of the whole system at a higher level[19].

Some of the more interesting work has centred on the combined behaviour of users and the peer-to-peer system itself. The seminal paper "Free riding on Gnutella"[20] shows that even mutual benefits do not stop people behaving selfishly.

Other aspects of peer-to-peer that have garnered study are security and collective behaviour [23,25].

Modelling Work. Alongside measurement, we usually find modelling, and peer-to-peer networking is no different from telecommunications or the Internet in this regard.

Examples of interesting models include papers on the power law of organisation of the system and the small world model of user, content and peer connectivity [21,26,27,34,36].

Industry and other interest. Peer-to-peer has attracted a lot of industry interest [37,38,39].

Possibly the most interesting work to date is the Microsoft Farsite project[42]. Building a massive, highly available, distributed file system out of (massively) unreliable components obviously has its attractions for this organisation.