A Conceptual Framework for Increasing Physical Proximity in Unstructured Peer-To-Peer Networks

Peter Danielis, Stephan Kubisch, Harald Widiger,
Jens Schulz, Dirk Timmermann
University of Rostock
Institute of Applied Microelectronics and Computer Engineering
18051 Rostock, Germany
Tel./Fax: +49 (381) 498-7272 / -1187251
E-mail: {peter.danielis;dirk.timmermann}@uni-rostock.de
Web: http://www.imd.uni-rostock.de/networking

***Abstract*—Today, Peer-to-Peer (P2P) already represents 60 percent of Internet traffic. Although P2P users are a good source for revenue for Internet Service Providers (ISPs), the data volume caused by P2P poses a significant challenge to ISPs regarding traffic engineering. Because P2P routing is usually agnostic of the underlying topology, traffic engineering abilities of ISPs are inhibited. This problem is known as mismatching problem between the logical P2P overlay topology and the underlying physical network topology. To mitigate this problem and, e.g., to avoid traffic congestions, the concept for a new mechanism is proposed in this paper. P2P users are provided with accurate information on the hop counts to other peers to select close peers in *unstructured* P2P networks. This mechanism does neither require a modification of the construction algorithm for unstructured P2P networks nor create any communication overhead. Already in the access network, the original Time-To-Live (TTL) value of outgoing IP packets is copied and inserted as IP option into these packets by the network operator. At the traffic destination, the hop count for an IP packet is calculated as the difference between the copied TTL value and the TTL value of the IP header. Using the hop count, a relationship between the logical overlay and the physical network is established to do traffic engineering.**

***Index Terms*—Peer-to-Peer, Hop Count, Locality, Access Network, Traffic Engineering.**

I. Introduction

During the last years, several new flavors of private and commercial use of the Internet have developed. One of these flavors are Peer-to-Peer (P2P) networks, clients, tools, and communities. Today, Internet traffic is dominated by P2P data (up to 60 %), which is mainly caused by file sharing applications like eMule or BitTorrent [1], [2]. However, there are numerous other application areas for P2P, e.g., the collaborative software Microsoft Groove. Recently, P2P techniques have been applied to Internet Protocol Television (IPTV) as well as a workaround for IP multicast. Examples are Joost or Zattoo [3], [4].

On the one hand, ISPs benefit from the P2P hype through an increase of their operating income. This is due to the fact that P2P applications are one of the main reasons for Internet users to subscribe for a broadband connection [5]. On the other hand, the high P2P data volumes pose a significant traffic engineering challenge. Traffic engineering denotes the process of managing traffic flows through the network [6], [7]. This discrepancy between operating income and traffic engineering challenge puts network operators and Internet Service Providers (ISPs) into a difficult situation.

Other traffic like HTTP, which accounts for about 30 percent of the overall traffic [1], is choked down because the ISP network is overburdened by P2P data. The main reason is that routing within logical P2P networks does not take the underlying physical Internet topology into account [8]. Usually, an unstructured P2P network overlay—on which we focus in the paper—is constructed by choosing random peers [2]. Due to this arbitrary procedure, neighbourship on the P2P overlay does *not* indicate proximity on the underlying Internet topology at all. This problem is usually denoted as topology mismatching problem between P2P overlays and physical network infrastructures [9]. Thus, two communicating P2P neighbours can be physically far away from each other although the desired content would be available on a physically more proximate peer [10]. Communication with physically distant peers uses long data paths and routes, e.g., in terms of the hop count. This consumes more bandwidth, which is not efficient when the load of the network is already heavy and can therefore cause traffic congestions [7]. Congestions are neither good for ISPs nor for users. In contrast, communication with proximate peers reduces the path length, i.e. the hop count. Therefore, less bandwidth is consumed and congestions can be reduced. Consequently, ISPs and network operators should offer an additional service to P2P users in order to support traffic engineering and to mitigate the topology mismatching problem.

This paper addresses a conceptual framework to provide P2P users with the information on the hop counts (regarding the physical hops) to other P2P users. Using the hop count, close-by P2P users are chosen on the average such that a shorter and eventually even faster path to download desired content can be selected. By preferably establishing P2P traffic flows between proximate peers, the routing of P2P traffic is improved indirectly. However, a P2P user primarily wants to download desired content as fast as possible. He is usually not aware of or even not interested in the underlying transport mechanism. Thus, a user would not select the most proximate P2P user among all peers, which provide the desired content

with nearly the same upload capability. But by providing P2P nodes with their hop count to other P2P nodes, they do not have to determine hop count by themselves. They can then avail themselves of the knowledge of the ISPs. Thus, given the download rates to other P2P nodes by the P2P application, a P2P user is able to choose the smallest distance in terms of hop count although there is no obligation. Still, in this paper, we assume P2P users to be cooperative by selecting close-by peers unless they do not suffer from it regarding their performance. This way, the mechanism is transparent and, in the best case, beneficial for the user and the provider.

Currently, the number of hops has to be determined by sending additional packets, namely ICMP packets. However, repeatedly applying this approach to always have an up-to-date value does not scale for large numbers of peers as they do exist in a P2P network. Unnecessary traffic overhead is created [11]. A new mechanism is therefore proposed in this paper, which does not create any overhead to determine the number of hops between two peer systems. At the ingress points of the carriers' networks, the Time-To-Live (TTL) value of outgoing IP packets is inserted as IP option into IP packets by this mechanism. At the packet's destination, the hop count is calculated from the inserted TTL value and the received TTL value.

Briefly summarized, the main contributions of this paper are the following:

- Investigations are carried out on how to calculate hop count as a metrics for physical distance between peers.
- An innovative mechanism is proposed, which is used to provide hop count for P2P applications. This allows for the selection of proximate peers.

The remainder of this paper is organized as follows: Section II explains how to compute hop count from TTL. Section III introduces the concept for the new mechanism to insert information and specifies structure and type of an inserted option in detail. Section IV contains a comparison of the proposed mechanism to related work. Ongoing and future work is described in Section V. The paper concludes in Section VI.

II. Computing Hop Count

Since hop count information is not directly stored in the IP header, it is necessary to compute it. To calculate hop count, there are actually two methods as stated in [11]. One is the so-called active measurement; the other is denoted as passive measurement. For active measurement, ICMP ECHO packets are used. Although this method mostly gives an accurate hop count, applying it to many hosts in a P2P scenario is impractical because enormous traffic overhead is created. Thus, as measuring method in the envisaged P2P use case, it is not favourable to send ICMP packets. Contrary, passive measurement simply means subtracting the TTL of a received IP packet from its initial TTL value. This is ideal for computing hop counts of many hosts because no extra packets have to be sent. Consequently, this approach is chosen for calculating the hop count.

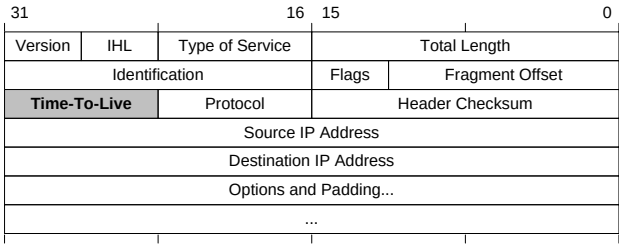

Fig. 1. IPv4 Header with Time-To-Live

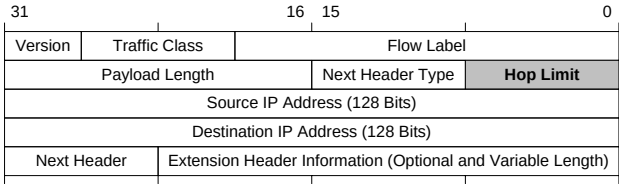

Fig. 2. IPv6 Header with Hop Limit

A. Time-To-Live (TTL) value in the IP header

The TTL value is the ninth byte of the IPv4 (further referred to as IP) header as shown in Figure 1.

As described in [12], the TTL specifies the maximum time a packet can exist on the network until it is discarded. A TTL value of, e.g., 64 allows a packet to be on the network for 64 seconds. The intention is to discard undeliverable packets if the TTL reaches 0 and to limit the maximum packet lifetime. Time is namely measured in seconds but actually the TTL is used as hop counter. Thus, every router that processes a packet must decrease the TTL by one even if it processes the packet in less than a second. Therefore, TTL can be used to compute hop count.

As the TTL value is usually used as hop count for physical hops with IPv4, for IPv6 this field has been renamed as Hop Limit (see Figure 2) [13].

B. Determination of the initial value for TTL

To calculate hop count from TTL, the initial TTL value of an outgoing IP packet is needed. Then, this value can be subtracted from the TTL value of the IP header at the packet's destination to get the hop count.

As shown in [14], due to the heterogeneity of the Internet, there is no unique initial TTL value. The initial TTL value depends on the operating system. In Table I, initial TTL values for different operating systems are given. TTLs may be different for TCP and UDP.

The question is now how to provide the initial TTL value without creating traffic overhead. This is where the new mechanism comes in by inserting the original TTL value as an IP option into every IP packet. This mechanism is described in the following section.

III. The New Mechanism In General

The development of the new mechanism was done totally decoupled from potential use cases like the one addressed in this paper. Originally, this mechanism called IPclip—Internet

TABLE I
INITIAL TTL VALUES OF DIFFERENT OPERATING SYSTEMS

Operating System	Initial TTL (TCP)	Initial TTL (UDP)
AIX	60	30
DEC Pathworks V5	30	30
FreeBSD 2.1R	64	64
HP/UX 9.0x	30	30
HP/UX 10.01	64	64
Irix 5.3	60	60
Irix 6.x	60	60
Linux	64	64
MacOS/MacTCP 2.0.x	60	60
OS/2 TCP/IP 3.0	64	64
OSF/1 V3.2A	60	30
Solaris 2.x	255	255
SunOS 4.1.3/4.1.4	60	60
Ultrix V4.1/V4.2A	60	30
VMS/Multinet	64	64
VMS/TCPware	60	64
VMS/Wollongong 1.1.1.1	128	30
VMS/UCX (latest rel.)	128	128
MS WfW	32	32
MS Windows 95	32	32
MS Windows NT 3.51	32	32
MS Windows NT 4.0	128	128
MS Windows XP	128	128

Protocol-Calling Line Identification Presentation—brings out a new and highly flexible solution that provides additional support for new services like VoIP emergency calls [15]. Thus, this section provides a brief overview of the general IPclip mechanism. For further detailed information on the general IPclip mechanism and its prototypic hardware realization, we refer to [15]. Reference [15] also illustrates other practical use cases for IPclip like emergency calls for VoIP.

The name IPclip is derived from the *Calling Line Identification Presentation*(CLIP) functionality in Integrated Services Digital Network (ISDN) telephone networks. CLIP is an optional feature submitting the caller's number to the telephonee and presenting this number on, e.g., a display. This way, the callee can identify the caller.

In case of packet-switched IP networks, the IP address of a user cannot be treated as equivalent to a fixed line telephone number. The reason is, as already mentioned in the introduction, that an IP address does not necessarily identify a distinct physical line. Furthermore, IP addresses *do not* allow any conclusions on the geographic location of a packet's origin. In contrast, fixed line telephone numbers *do* have a well-defined and known origin. The original idea and the name of the CLIP feature in classical ISDN telephone networks are thus adapted for packet-switched IP networks. From a technological point of view, IPclip is a completely novel mechanism and cannot be compared with the classic ISDN CLIP.

A. Why the Internet Protocol and what Kind of Information?

An Internet user and his actual geographic location can be identified with IPclip using a tuple of information consisting of the customer's current IP address and some additional information. As IP addresses do not suffice for explicit determination of the customer's location, reliable location information must be included in the additional information. Preferably, standardized data formats should be used for it in order to ensure global interoperability, which is essential in the Internet. Due to its global availability, the Global Positioning System (GPS) data format is used to encode geographic location information [16] at the moment. The sum of all additional information—in the following just specified as location information (LI)—is used for analysis, classification, or stimulation of further actions.

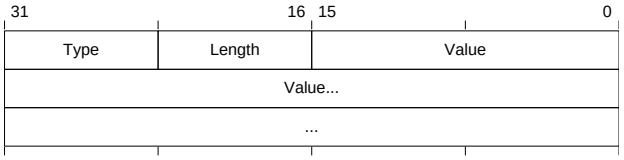

Fig. 3. TLV structure of an IP option

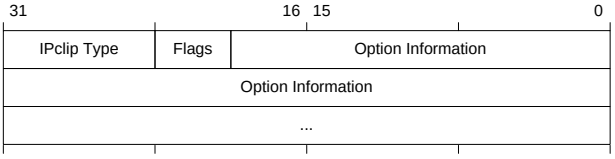

Fig. 4. Structure of an IPclip option inside the value field of an IP option

To provide this LI on a global scale, an optional data field is inserted into every IP packet. The reason is that IP is the central protocol in the Internet and the World Wide Web. IP provides end-to-end connectivity between users, service providers, and network nodes in general. Besides, structure and size of optional fields inside IP, so-called IP options, are standardized [12]. This way, the IPclip mechanism is a standard-compliant solution to deliver supplementary LI. Every IP-capable device can either analyze and processes IP options or ignore them. But in any case, devices must at least be able to parse and skip IP options for reasons of interoperability. Next to the feature of adding additional LI into packets, the whole mechanism can be configured to remove suchlike IP options. This may be necessary if Internet users do not want to receive or are not allowed to receive sensitive information about the geographic origin of IP traffic. In these cases, the use of IPclip is totally transparent. However, this depends on the particular application.

The new IP option shows the typical Type-Length-Value (TLV) structure as sketched in Figure 3.

The TLV structure must be understood by every IP-compliant network device. The type field is divided into a 3-bit field for various flags and a 5-bit IP option number. For prototyping, we have chosen 26 as option number for IPclip as it is not in use otherwise [17]. Length denotes the IP option length including type and length field. The value field of the new IP option contains the IPclip option. Figure 4 shows the structure of an IPclip option.

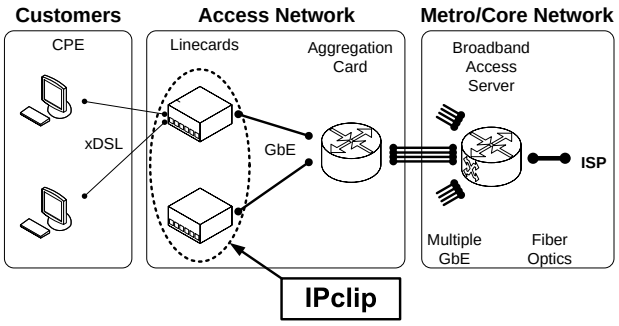

Fig. 5. Structure of a typical access network. The position of the IPclip functionality is highlighted.

The IPclip type field denotes the kind of information this IPclip option contains. For the P2P use case, 6 is chosen as IPclip type since preceding type values are already assigned to, e.g., GPS location information. The 4-bit status field contains flags, which are not used for the P2P use case. The option information field contains the actual information, which depends on the IPclip type. In our case, this is the initial TTL value of outgoing packets.

For a complete encoding, 8 bits for the initial TTL value are necessary. This leads to an option length of 8 bits plus 12 bits for IPclip type and flags resulting in 3 bytes for the IPclip option when complemented to an integer number of bytes by padding. For the complete TLV-structured IP option including IP option type and IP option length, 5 bytes are needed in total.

The addition of location information including its analysis and verification raises different important questions:

- Which is the place within the network infrastructure where the location information to be added is available?
- Which is the place within the network infrastructure where this location information can be added into IP packets?
- How can a trust relationship and a certain degree of credibility be described and how can it be ensured when analyzing and validating the additional information?

B. IPclip's Position within the Network Infrastructure

Network ingress—also known as access network—is the most reasonable place where LI can be added and verified. Access networks comprise the Customer Premises Equipment (CPE) as well as so-called access nodes like IP DSL Access Multiplexers (IP DSLAMs). Usually, access nodes consist of multiple linecards and an aggregation card. This structure is shown in Figure 5. While aggregation cards manage high-bandwidth interfaces towards the core network, linecards mainly concentrate high numbers of subscribers. Since the paper describes a conceptual framework, the generic term *access node* (AN) is used throughout the paper.

The inherent physical line information, e.g., the port number on the AN, can already be treated as some flavor of LI. Thus, our approach is based on the assumption that LI can be added either by the CPEs (only GPS location information) or by the IPclip mechanism in the ANs (GPS location information *and* access port number *and* access node ID). However, verification and validation of the LI and thereupon taken measures are solely done in the ANs. The reason for doing so is that CPEs are typically not considered as trustworthy network elements by network carriers and service providers. CPEs are usually not within the carriers' management domains. By contrast, ANs are part of the access network and thus within a carrier's management domain. A tuple of information available in ANs is used as precise LI to identify and locate an Internet user:

- the geographic location of the access node
- the access port number the user is connected to
- the access node ID

That is why the IPclip functionality is implemented in the ANs as highlighted in Figure 5.

TABLE II
DISTANCE CLASSIFICATION

Hop Count Distance	Description
0...1	Neighbour
2...4	Very close
5...9	Close
10...14	Medium
15...19	Far
>20	Very far

C. IPclip as a Mechanism to insert the Initial TTL Value

However, as mentioned above, IPclip is not limited to the scenario of adding location information to identify users by their geographic location. The IP option inserted by IPclip can take any value to support other applications. It is a generic container. For our purpose, the IPclip option shall contain the initial TTL value of outgoing IP packets. Then, at the packet's destination, hop count can be calculated to be used by P2P applications. The hop count value serves as an additional trigger for P2P applications to classify peers according to their physical distance to each other. Based on the classification of hop count in [18] (hop count is called IP path hops there), 6 different classes can be defined with regard to the number of hop counts between two peers (see Table II). Thereby, P2P client software can provide the user with striking information about the distance to other peers.

IV. COMPARISON WITH RELATED WORK

The mismatch between logical P2P overlay and the underlying physical topology becomes a serious obstacle for the development of P2P systems. Being aware of the mismatching problem, much scientific literature can be found, which addresses the locality problem in Distributed Hash Table (DHT)-based P2P networks, i.e. structured P2P networks [19], [20]. In structured P2P networks, all peers are organized into an identifier ring and an association between content and location where it is stored is given. Basically, there are three approaches, which have been proposed to exploit network proximity in DHTs. For a detailed discussion of the three approaches, the interested reader is referred to [21].

However, not only structured but also unstructured P2P networks suffer from the mismatching between the logical overlay and the underlying physical topology. Hence, in this paper we focus on selecting close-by peers in unstructured P2P networks. In contrast to structured P2P networks, peers in unstructured P2P networks are organized arbitrarily. There do exist many approaches (e.g., [22], [23] and [24]) to *construct* unstructured topology-aware overlay networks. These approaches improve performance significantly and avoid unnecessary traffic by exploiting network proximity. However, they require adding structure to unstructured P2P networks following physical network characteristics. Moreover, overhead is created for maintaining this structure. In constrast to this approach, we do not intervene with the *construction* of unstructured P2P networks. Instead, we use a new mechanism, which provides a trigger (the hop count) in every packet to be able to select proximate peers in randomly built, unstructured P2P networks. Thereby, no modification of the construction algorithm is necessary and no traffic overhead is created to determine the distance between peers.

V. Future Work

Ongoing and future work includes extensive simulations for a large set of unstructured P2P network configurations. Particularly, the P2P use case will be analyzed for the examples of BitTorrent and eMule (using the eDonkey2000 network), which are the largest P2P networks currently in use. Moreover, investigations are performed on how information other than hop count can be advantageous for P2P applications—especially when real time is an important constraint. For example, the latency between two peers could be determined using a GPS timestamp as additional information in packets. Currently, the framework is discussed for an IPv4 environs. But IPv6 will be the dominating protocol in the prospective Internet. Future work will thus focus on the adaptation of the new mechanism and its application for P2P use cases to IPv6 environments.

Furthermore, a hardware prototype is currently set up for an FPGA development board. The hardware module to insert the initial TTL value into IP packets can process IP traffic with wire speed.

VI. Conclusion

The paper proposed the concept for a new mechanism to provide P2P users with hop count for the selection of close-by peers in unstructured P2P networks like BitTorrent and eMule (using the eDonkey2000 network). By selecting proximate peers, physical proximity of P2P traffic can be increased. Therefore, less bandwidth is consumed, which avoids traffic congestions when the load of the network is already heavy. Thereof, ISPs benefit. The new mechanism does neither require a modification of the construction algorithm for unstructured P2P network nor create traffic overhead to determine the hop count. The hop count is calculated from the difference of the initial TTL value of a packet's IP header and the TTL value at the packet's destination. The initial TTL value is inserted as an IP option into every packet.

Acknowledgement

We would like to thank the Broadband Access Division of Nokia Siemens Networks in Greifswald, Germany for their inspiration and continued support in this project.

This work is partly granted by Nokia Siemens Networks as well as the 4th Priority Research Program on Information- and Communication Technologies, Mecklenburg-Vorpommern, Germany.

References

[1] CacheLogic Research, "Internet Protocol Breakdown 1993 - 2006," 2006.
[2] R. Steinmetz and K. Wehrle, *P2P Systems and Applications, Springer Lecture Notes in Computer Science*. Springer-Verlag Berlin Heidelberg, 2005.
[3] "Joost Free Online TV," 2007. [Online]. Available: http://www.joost.com/
[4] "Zattoo TV to Go," 2007. [Online]. Available: http://zattoo.com/
[5] T. Mennecke, "DSL Broadband Providers Perform Balancing Act," 2005. [Online]. Available: http://www.slyck.com/new.php?story=973
[6] J. Leyden, "P2P swamps broadband networks," 2002.
[7] X. Xiao and L. Ni, "Internet QoS: A Big Picture," vol. 13. IEEE Network Magazine, 1999, pp. 8–18.
[8] V. Aggarwal, S. Bender, A. Feldmann, and A. Wichmann, "Methodology for Estimating Network Distances of Gnutella Neighbors." GI Jahrestagung (2) 2004, 2004, pp. 219–223.
[9] H. Wan, N. Ishikawa, and J. Hjelm, "Autonomous Topology Optimization for Unstructured Peer-to-Peer Networks." IEEE Proceedings of the 11th International Conference on Parallel and Distributed Systems (ICPADS05), 2005, pp. 488–494.
[10] A. Rasti, D. Stutzbach, and R. Rejaie, "On the Long-term Evolution of the Two-Tier Gnutella Overlay." INFOCOM 2006, 2006.
[11] K. Fujii and S. Goto, "Correlation between Hop Count and Packet Transfer Time." Asia-Pacific Advanced Network (APAN) / IWS [Internet Workshop] 2000, 2000.
[12] Information Sciences Institute, University of Southern California, "Internet Protocol Specification," RFC 791, September 1981.
[13] S. Deering et al, "Internet Protocol, Version 6 (IPv6) Specification," RFC 2460, December 1998.
[14] Swiss Education & Research Network (SWITCH), "Default TTL Values in TCP/IP." [Online]. Available: http://secfr.nerim.net/docs/fingerprint/en/ttl_default.html
[15] H. Widiger, S. Kubisch, P. Danielis, J. Schulz, D. Timmermann, and T. B. D. Duchow, "Trust-by-Wire in Packet-switched Networks: Calling Line Identification Presentation for IP." Submitted at the Kaleidoscope Academic Conference Innovations In NGN, 2008.
[16] National Marine Electronics Association (NMEA), "NMEA 0183 Standard," January 2002.
[17] Internet Assigned Numbers Authority (IANA), "IP Option Numbers," February 2007.
[18] A. Iosup, P. Garbacki, J. Pouwelse, and D. Epema, "Correlating Topology and Path Characteristics of Overlay Networks and the Internet." Proceedings of the Sixth IEEE International Symposium on Cluster Computing and the Grid Workshops (CCGRIDW06), 2006.
[19] B. Zhao, Y. Duan, and L. Huang, "Brocade: Landmark Routing on Overlay Networks." 1st International Workshop on Peer-to-Peer Systems, 2002.
[20] N. J. A. Harvey, M. B. Jones, S. Saroiu, M. Theimer, and A. Wolman, "SkipNet: A Scalable Overlay Network with Practical Locality Properties." 4th USITS, 2003.
[21] M. Castro, P. Druschel, Y. Hu, and A. Rowstron, "Exploiting Network Proximity in Distributed Hash Tables." International Workshop on Future Directions in Distributed Computing (FuDiCo), 2002.
[22] H. Wang, Y. Zhu, and Y. Hu, "To Unify Structured and Unstructured P2P Systems." 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05), 2005.
[23] S. Merugu and E. Zegura, "Adding Structure to Unstructured Peer-to-Peer Networks: The Use of Small-World Graphs." Journal of Parallel and Distributed Computing, 2005.
[24] Y. Liu, X. Liu, L. Xiao, L.M.Ni, and X. Zhang, "Location-Aware Topology Matching in P2P Systems." INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies, 2004.