NOTICE: This is the author's version of a work accepted for publication. Changes resulting from the publishing process, including peer review, editing, corrections, structural formatting and other quality control mechanisms, may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. The final publication is available at www.degruyter.com.

# Dynamic Operation of Peer-to-Peer Overlay Networks

H. deMeer, K. Tutschku, and P. Tran-Gia

*Abstract*— **Virtual overlay networks, such as virtual private networks or peer-to-peer services, can be seen as a new paradigm for providing multi-service networks. Virtual overlay networks may offer customized services to a specified community while providing a high degree of flexibility in the usage of shared resources. This paper examines the requirements of operating dynamic overlays, in particular, for peer-to-peer services. The analysis is based on extensive measurement studies performed on the global Gnutella network during operation. The obtained results indicate limitations in scalability of native p2p overlays, suggesting the need of a control scheme for efficiency reasons. As an enabling infrastructure to implement a distributed control scheme for p2p overlays a so-called Application-Layer Active Networking (ALAN) platform has been chosen. Based on Application-Layer Active Networking, *Active Virtual Peers (AVP)* are introduced as the main concept for dynamic operation and management of peer-to-peer overlay networks. AVPs facilitate policy enforcement or performance management by means of self-organization, predominantly on the application layer with minimum interference on lower layers.**

**Keywords:** Peer-to-Peer services, overlay systems, Gnutella, measurement, traffic management.

## I. INTRODUCTION

Peer-to-Peer (p2p) networks have become very popular recently amid the relentless spread of Gnutella [1], Kazaa [2], and eDonkey [3] file sharing applications. Remarkably, only very little support was needed to make these distributed services operable on a large scale in very little time. One of the main reasons for the noted success is due to the fact that p2p networks operate as overlays. Overlays work without specific network or transport support and can be completely run at the edge of the network. While p2p overlay networks do implement a certain type of group communication structures, they do not suffer from same the deployment difficulties as multicast did in the past. But ease in deployment came at a cost: A lack of central servers, or of any central control for that matter, predictably led to a huge amount of uncontrolled signaling traffic being generated and transmitted.

The current challenge is to provide attractive p2p services without compromising network services offered to concurrent applications. An effective management system for overlay networks could have large benefits to a wide range of network applications that may go far beyond improving usage of the popular p2p services. It would be applicable to content delivery networks or other many-to-many communication services that need Quality-of-Service (QoS) support, effectively removing the need to implement QoS provisioning on the network layer, which has been the major obstacle to a wide-spread usage of these services.

The remainder of the paper is organized as follows. Section II discusses key requirements of future network applications forming virtual overlays. Section III describes a special representative of overlay networks: the Gnutella file sharing service. Section IV provides a measurement-based analysis of signaling in Gnutella. This is followed by Section V which discusses the impact of variable overlays on future traffic management. Section VI introduces a new concept for dynamic operation of p2p overlay networks. The approach applies Application-Layer Active Networking and introduces *Active Virtual Peers* (AVP). Section VII describes an on-going field trial of the suggested AVP concept. Section VIII discusses the context of the measurement study at hand and outlines briefly related concepts for operating p2p overlay networks- Section IX concludes the paper.

## II. FUTURE DISTRIBUTED AND VARIABLE APPLICATION ARCHITECTURE

### A. Requirements

The capabilities of new application-oriented protocols such as *SOAP (Simple Object Access Protocol)* [4] or recent p2p services protocols such as Gnutella [1] indicate that future network applications will be highly decentralized and more loosely connected than those based on the traditional client/server concept. The new class of applications will offer services at the network's edge. The services are anticipated to be offered within application-specific *overlays.* The overlay network is largely independent from

---

H. deMeer is with the Department of Electronic & Electrical Engineering, University College London, London, Great Britain. E-mail: H.DeMeer@ee.ucl.ac.uk .

K. Tutschku and P. Tran-Gia are with the Institute of Computer Science, University of Würzburg, Würzburg, Germany. E-mail: [tutschku,trangia]@informatik.uni-wuerzburg.de .

the physical one. In the overlay, nodes will have significant autonomy and symmetric roles. An overlay node may be client, server, and router at the same time. Therefore, they are denoted as *peers*. Overlay membership is granted spontaneously and nodes may as well suddenly disappear.

These features lead to multiple new challenges in network operation. The expected difficulties are the high dynamics in overlay topologies and the dispersion of traffic sources throughout the network, i.e. high volume traffic flows may appear on short time-scales at almost any location at the network's edge. The *scale of dynamics* of p2p service overlays will prohibit, or at least complicate, the application of traditional traffic engineering techniques.

### *B. Virtual Overlay Networks*

Overlay networks may offer customized services and virtualization of resources to a specified community, providing some form of flexibility in usage of shared resources. Examples of virtual overlay networks are virtual private networks (VPNs), content delivery networks (CDNs), or many of the popular p2p services. VPN systems are well suited for enterprise networks since they provide basic, well defined, and well managed transmission services. In this context, however, short time-scale network variability of overlays is usually not supported as servers are statically located. Therefore, those overlays are denoted as *static overlays*, in contrast to p2p overlays for which high variability is a predominant characteristic as shown next.

## III. THE GNUTELLA FILESHARING SERVICE

The Gnutella service is an entirely distributed file sharing application [5]. The Gnutella hosts are denoted as servents *(SERVer + cliENT)* and act simultaneously as servers and clients. In addition, they are responsible for routing the *signaling traffic*. This traffic spreads information used to preserve network integrity and is needed to locate information. The downloads are performed outside of the overlay by a direct peer-to-peer connection between servents.

The Gnutella protocol defines two categories of signaling messages: *a) Overlay Membership:* To discover additional hosts on the Gnutella network, servents use a "Ping/Pong" protocol. A servent issues a "Ping" message[1] to actively probe the network. A servent receiving a "Ping" is supposed to respond with a "Pong" message, containing the IP address and the amount of data it is sharing on the network. A "Ping" message can be answered with multiple "Pong" messages from multiple servents. *b) Searching Information:* A piece of information is located in Gnutella via "Query" and "QueryHit" messages. A "Query" contains mainly the search criteria. A servent receiving a "Query" descriptor responds with a "QueryHit" message if a local match is found. A "QueryHit" message contains mainly information to identify the replying servent in the IP address space as well as in the Gnutella domain. Once a servent receives a "QueryHit" response, a user may trigger a download. An HTTP connection containing a GET request is directly established between the servents.

In order to join the Gnutella signaling overlay, a new servent connects to one of numerous well-known hosts that are always available, e.g. `router.limewire.com`. After having been connected successfully, servents send messages to interact with each other. Gnutella servents know only about servents which are directly connected to them. Other nodes are invisible unless they announce themselves. A node may maintain multiple simultaneous connections to other servents in the overlay. The maximal number of simultaneous connections can usually be configured by the user.

Signaling messages are routed in the overlay by using two simple principles: *a)* they are *broadcasted* to all neighbors, and *b)* responses are *back-propagated* in the overlay along the path taken by the triggering message.

An important feature of the Gnutella p2p filesharing service is that peers may join or leave the signaling overlay arbitrarily. To preserve network integrity, servents have to maintain multiple simultaneous connections. New overlay connections have to be initiated as soon as old ones terminate. Peers acquire new candidates for their overlay connections by sending "Ping" messages to neighbors and inspecting "Pong" responses. Nodes base their decision where to connect to in the network purely on their local information. The Gnutella protocol doesn't provide any support for a coordinated organization of the signaling overlay. The Gnutella service forms a randomly structured overlay network.

## IV. OVERLAY MEASUREMENTS

While the qualitative justification is straightforward, little is known of quantitative results on the scale of dynamics in overlays and p2p applications. In particular, time scale and variability of the number of virtual overlay connections have to be characterized.

### *A. Measurement Set-Up*

To analyze the signaling traffic in the Gnutella overlay, we modified a publicly available Gnutella command line application `Gnut` Version 0.4.24 [6] to record all signal-

---

[1] The Gnutella "Ping" message should not be mistaken for the ICMP echo request message often colloquially also denoted as "Ping".

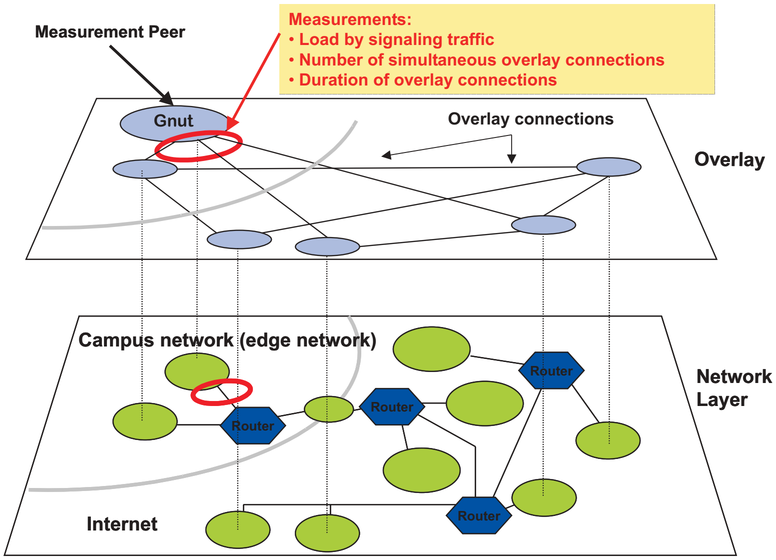

Fig. 1. Overlay measurements

ing packets with time stamp, payload size, and source and destination IP address. The `Gnut` application was executed on an Linux-based PC. The PC was located within the campus network of the University of Würzburg and was connected to the departmental network via FastEthernet. The measurement campaign was carried out in March 2002. The measurement duration was 60 hours. Figure 1 shows the layout of the measurements in a simplified manner. The measurements were performed on the edge network connection of the peer at overlay level.

### B. Traffic Load

Figure 2 shows the sum of all signaling traffic load observed by the measurement peer. The depicted load is the total load of all simultaneous overlay connections. Figure 2 depicts solely traffic load generated by Gnutella search requests (“Query”), search replies (“QueryHit”), host queries (“Pings”), and host announcements (“Pongs”). No download traffic contributed to the load shown in Figure 2 and the load is averaged on 10sec intervals.

The observed average of signaling load was $0.274$Mbps in the $10$sec-intervals. The maximum was $50.9$Mbps. Figure 2 was truncated to the range of $10$Mbps in order to focus on the most relevant part of the graph. The figure depicts also the $95\%$ percentile of the load, which is $1.03$Mbps. In total, more than $7.06$Gbyte of signaling traffic was transmitted over the measurement peer during the 60 hours. That data volume is equivalent to ten Video-CDs with $700$Mbyte each and is without any immediate benefit to the user. The high overhead is due to the use of broadcast mechanisms in the original Gnutella protocol. The Gnutella overlay network is flooded with signaling information. Moreover, the better the peer’s connection to the overlay, the faster messages are forwarded to it. A control of the traffic load is difficult. With the exception of a TTL (Time-To-Life) field, the Gnutella protocol doesn’t contain any mechanisms to control the signaling. Traffic control has to be implemented locally and independently on each peer. Newer Gnutella implementations, e.g. `Limewire` [7], fight the high signaling load by the introduction of *superpeers* which aggregate signaling information. Such an aggregation, however, is left to implementation and configuration of the peer.

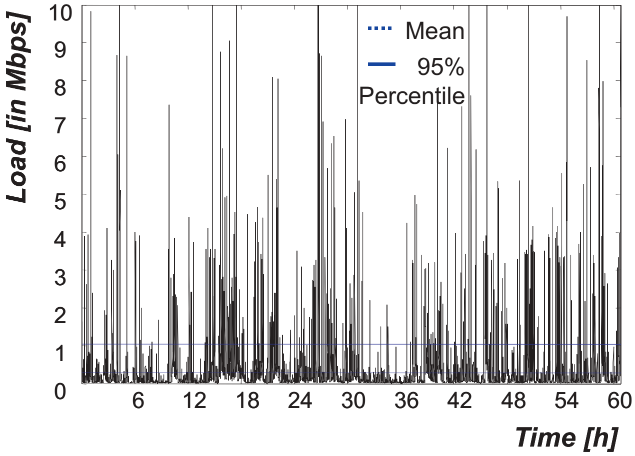

Fig. 2. Sum of signaling traffic load

### C. Overlay Variability

The variability in the p2p overlay can be characterized by two factors: *a)* the number of simultaneous overlay relations maintained by a peer and *b)* the duration of maintaining these relations. The term “relation duration” as used in this paper denotes the time between the first and the last instance of information exchange between the measurement peer and a particular other peer. A peer is identified by its IP address and the TCP port number used by the Gnutella application running on this peer. A peer-to-peer relation may therefore last several physical connections between some peers. The term “connection”, however, is not fully appropriate in the context of p2p services. Many signaling messages may be exchanged between some peers, while the same peers may repeatedly join or leave the overlay.

#### *Number of simultaneous overlay relations*

A peer tries to maintain a given number of relations. The number can be configured by the user and is fixed here at 20 relations for the measurement peer. If, for instance, a peer maintains less relations than configured, it picks out an arbitrary host announcement and tries to establish a new relation to this host.

Figure 3 depicts the number of simultaneous p2p relations maintained by the measurement peer during the measurement period. Although the peer was configured to keep up with 20 relations, it maintained only an average of 9.86 relations. Most importantly, however, the connectivity process reveals a very high variability and is far from being constant.[2]

If the connectivity of a peer is high, i.e., a peer maintains a high number of simultaneous overlay relations, many signaling messages will be forwarded to it. If bandwidth is not sufficiently available an overload situation is caused in the physical network. If the connectivity of a peer is low, i.e., a peer maintains a small number of relations, then a peer might not receive enough signaling information to discover new hosts and new resources. In an extreme case, a peer might drop out of the overlay network and has to be re-connected to a well-known peer. That may cause a severe disruption of the service. This characteristic suggests the existence of an optimal level of connectivity. But rather than consistently maintaining an optimal level of connectivity, connectivity fluctuates widely in unmanaged p2p environments.

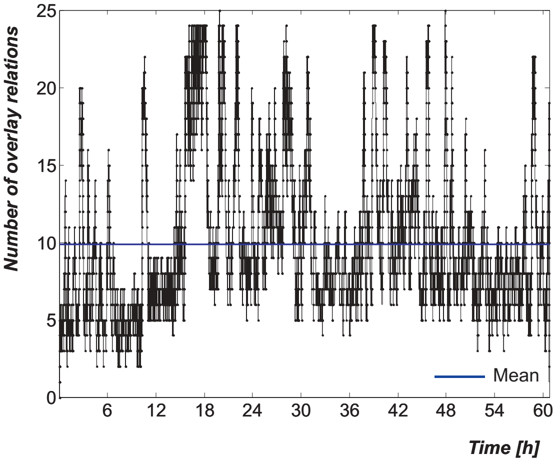

Fig. 3. Number of simultaneous overlay relations

The high variability in the number of simultaneous overlay relations indicates that management might be needed in order to maintain the optimal point of operation for a peer under given performance and reliability constraints.

#### *Distribution of the relation time*

The distribution of the relation times of peers is the second factor in the variability of the overlay.

The measurement peer exchanged signaling messages with 5320 distinct peers and monitored the duration of the relation times. Figure 4 depicts the histogram of the relation durations observed by the measurement peer. The mean relation holding time is $4.05 \cdot 10^2$sec. The histogram shows also the $90\%$-interval of the measurements, which is between $1.28 \cdot 10^{-1}$sec and $1.28 \cdot 10^3$sec, and which reaches over four orders of magnitudes in the time scale. The median is at $6.88 \cdot 10^1$sec. In addition, Figure 4 reveals clearly a distribution with two modes. The separating minimum is located at about $10$sec. This behavior indicates that the p2p relation is governed by multiple states.

From the user’s point of view, the participation in a p2p overlay is fruitful when peers receive sufficient content information. That way, peers might use and contribute resources to the p2p community in a valuable way. Hence the volume of *incoming signaling traffic* was examined in greater detail. The traffic was correlated with the relation times.

Figure 7 shows the correlation of the relation times for the complete amount of incoming signaling data transmitted during existence of a p2p relation. Each point in the diagram stands for a single relation. The abscissa denotes the relation duration and the ordinate represents the transmitted amount of signaling data in this relation.

Figure 7 depicts the $90\%$ interval for the relation duration (vertical lines) and the traffic volume (horizontal

---

[2] Figure 3 shows also that the measurement peer occasionally maintained more than 20 connections. This is an implementation feature of the `Gnut` client.

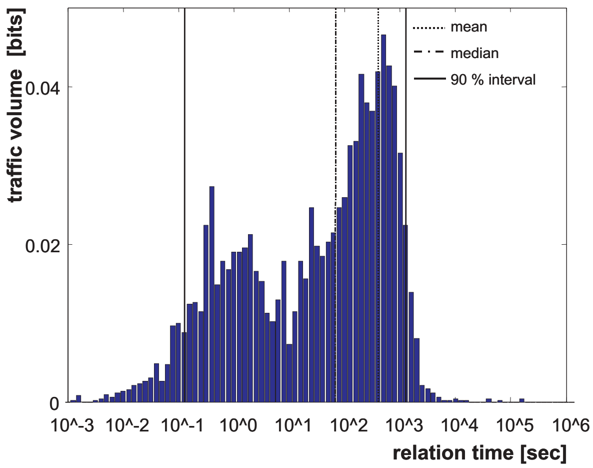

Fig. 4. Relation times observed by the measurement peer - all relations

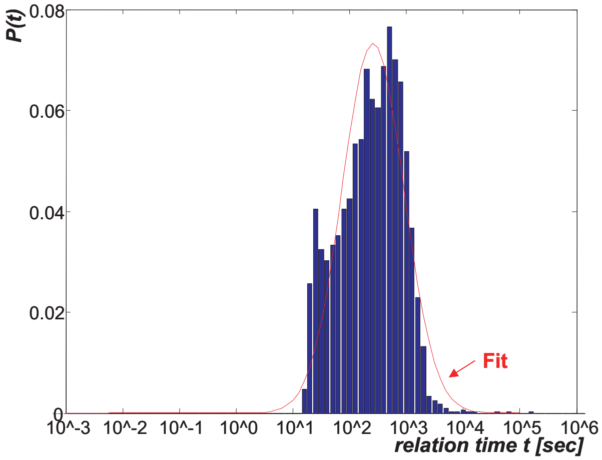

Fig. 6. Relation times: second category ($\geq 1.95 \cdot 10^1$sec and $\geq 9.12 \cdot 10^3$bits )

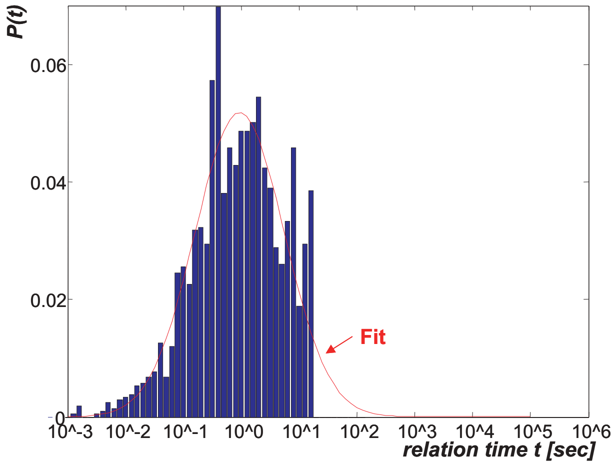

Fig. 5. Relation times: first category ($< 1.95 \cdot 10^1$sec or $< 9.12 \cdot 10^3$bits)

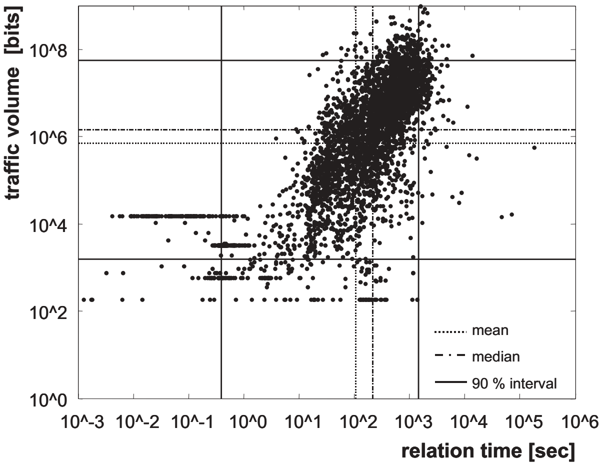

Fig. 7. Correlation of relation times and amount of incoming signaling data: all information (content and host)

lines). The average and the median are indicated by dotted and dashed lines. The figure allows the identification of two discriminating values which correspond to the lower bound of the interval. For the relation times a separating value of $3.90 \cdot 10^{-1}$sec and for the signaling traffic volume a total of $1.54 \cdot 10^4$ bits are indicated. The lower bounds of the 90% interval on the time axis and the volume axis describe the range of beneficial overlay relation. During this relations either sufficient host or search information is exchanged between peers. The separating values become even more evident when the correlation is performed for the individual Gnutella protocol entities. Figure 8 depicts the correlation for "Query" packets, i.e., for file search requests only. The lower bounds of the 90% interval for relation duration and signaling volume are at $1.95 \cdot 10^1$sec and $9.12 \cdot 10^3$ bits. A similar behavior is also visible for "QueryHit" packets, cf. Figure 9.

In contrast to the behavior of "Query" and "QueryHit" packets, the correlation analysis for "Ping" packets, i.e., host query packets, shows that a considerable number of p2p relations exist which have a duration of less than 10sec. In this case the transmitted amount of signaling information is small, see Figure 11, and typically less than $10^4$ bits. This is also the case for "Pong" packets, i.e., host announcement information, see Figure 10.

### D. Statistical Model

Based on the correlation analysis the histogram of relation times was reassessed. The p2p relation times are filtered and divided into two disjoint categories. The categories are determined by the lower bounds of the 90%

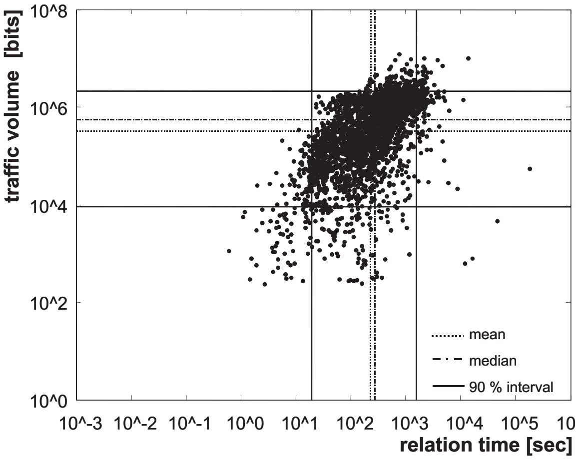

Fig. 8. Correlation: Query packets only

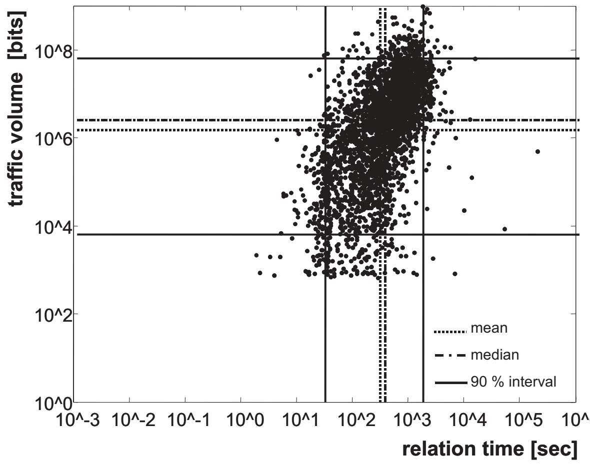

Fig. 9. Correlation: QueryHit packets only

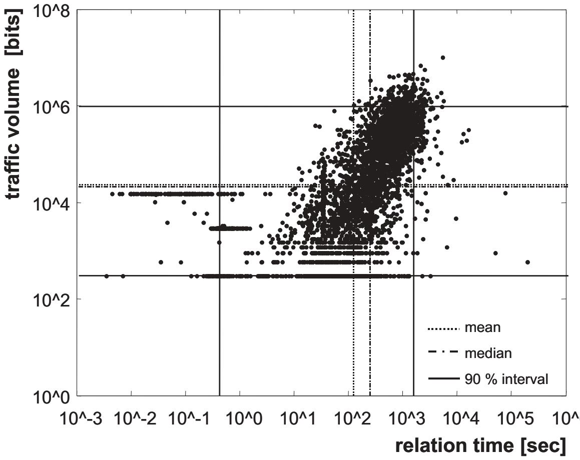

Fig. 10. Correlation: Pong packets only

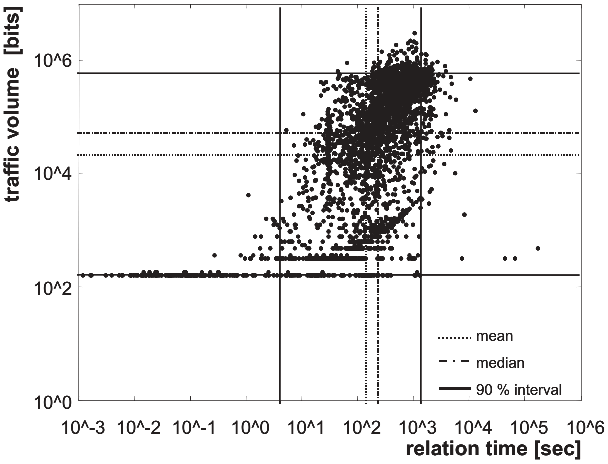

Fig. 11. Correlation: Ping packets only

interval of the relation times and traffic volume for queries (see Figure 8). The first category contains overlay relations which last less then $1.95 \cdot 10^1$sec and have less than $9.12 \cdot 10^3$ bits signaling volume, see Figure 5, and contains $39.0\%$ of the relations (2077 relations out of the total of 5320). The second category has relation times greater than $1.95 \cdot 10^1$sec and a traffic volume of more than $9.12 \cdot 10^3$ bits, see Figure 6. The category comprises $61.0\%$ of the relations (3243 out of 5320). In both categories the shape of the histogram of relation times may be approximated by a normal probability distribution. Since the abscissa in Figures 4 to 6 are of logarithmic scale, it is indicated that relation duration in the two classes are distributed according to the *log-normal* distribution function:

$$f(x) = \frac{1}{x\sqrt{2\pi\sigma^2}} e^{\frac{-(\ln x - \mu)^2}{2\sigma^2}}. \tag{1}$$

The fitted distributions are added to the histograms of Figures 5 and 6. Visual inspection of these figures shows that the fit is remarkably well. This indicates in this case that the log-normal distribution appears as a valid statistical model for states which conduct the set-up of overlay relations. Table I provides the values (measured and fitted) for the mean and the variance of the relation durations in the two categories.

### *Significance for P2P Overlays*

The correlation analysis leads to a two-state model for Gnutella p2p overlay relations. In the first state, which is called the “short” state, a peer establishes only short-lived connections to other peers. The involved peers exchange some signaling information, typically only host information, and the relation is terminated immediately. In the other state a peer establishes a long-duration relation

|  | Mean |  | Variance |  |
|---|---|---|---|---|
| Category / State | Measurement | Fit | Measurement | Fit |
| 1 / “short” | 3.01 sec | 1.85 sec | $1.98 \cdot 10^1$ sec$^2$ | 9.99 sec$^2$ |
| 2 / “stable” | $5.93 \cdot 10^2$ sec | $3.63 \cdot 10^2$ sec | $1.24 \cdot 10^7$ sec$^2$ | $1.29 \cdot 10^5$ sec$^2$ |

TABLE I
MEASURED AND FITTED PARAMETERS FOR RELATION DURATION

and continuously exchanges signaling messages, mostly search requests. From the perspective of a user, this state can be called “stable” state, since it permits uninterrupted operation of the p2p service.

The characteristic of log-normal distributed p2p relations indicates that the majority of peers in the stable phase resides in the system for the same average amount of time. A significant number of peers, however, may stay longer than the average. This result supports the view that few peers are more powerful and durable than other peers [8].

## V. POSSIBLE IMPACT OF THE SCALE OF DYNAMICS ON TRAFFIC MANAGEMENT

The observed scales in the Gnutella signaling overlay indicate that traffic management for future Internet will require new approaches. It can be expected that the *scale of dynamics* of future application-specific service overlays will be in the range of the variability of today’s p2p overlays. As the measurements of the variability of the Gnutella overlay have shown, cf. Section IV-C, this will be on a time-scale in the order of tens of minutes. This characteristic prevents the application of today’s traffic engineering techniques, such as Traffic Load Flow Optimization or Multi-Hour dimensioning (see Annex 6 of [9]). In addition, the group communication support by future applications will additionally lead to the dispersion of traffic sources throughout the network and may also cause difficulties in estimating traffic flows on small time scales.

The new services offered by future Internet will be built on node autonomy and on symmetric roles of networked nodes. The application specific overlays may contend for network capacities [10]. In contrast, current IP Quality-of-Service (QoS) design favors the differentiation of traffic, e.g., by explicit use of ToS (Type-of-Service) bits to select QoS, to avoid congestion. To provide attractive overlay services, future services will have to include self-organization mechanisms on application layer. The mechanisms should be able to observe overlay load and be adaptive on small timescales.

An absence of any traffic engineering, as currently observed with many p2p overlays, will lead to a reduction of the service quality of these services. The service performs well for users with high bandwidth access, i.e., they perceive a high throughput for downloads. On the down side, a large amount of signaling traffic is also forwarded to these peers and has to be handled there. Considerable bandwidth is consumed without getting immediate benefit. It is anticipated that a management architecture will be needed that can handle specific granularities in time as well as in space to enable dynamic and adaptive operation of future virtual overlay networks.

## VI. MANAGING P2P OVERLAY WITH APPLICATION-LEVEL ACTIVE NETWORKS – THE “ACTIVE VIRTUAL PEER ARCHITECTURE”

### A. *Management Objectives*

It becomes increasingly clear that p2p application requirements can’t be addressed purely by network layer functions in a scalable and efficient way, whereas the requirements can be dealt well on the application layer. In consequence, this calls for a management architecture that has a universal programmability on the application layer for performance control as well as for group management.

Furthermore, p2p overlay control should be equipped with handles for the adaptivity to different scales of dynamics to overcome limitations of conventional static traffic engineering. The suggested solution is based on: *i)* a flexible infrastructure, e.g., an ALAN-based approach [11], *ii)* automatic load-balancing in the network on small times scales, and *iii)* the integration of self-organization and adaptiveness on application level.

### B. *ALAN architecture*

A promising Active Networking concept has been proposed in [11], where so-called Application-Level Active Networking (ALAN) is pursued. In the ALAN framework, generic computing facilities, called Execution Environments for Proxylets (EEPs), are placed strategically in networks. Active code elements are deployed on demand using a URL-mechanism and are executed on the EEPs. Proximity measures and other metrics are used to choose appropriate EEPs for launching proxylets sensibly and establishing an application specific overlay topology.

A central service of ALAN is multi-metric applica-

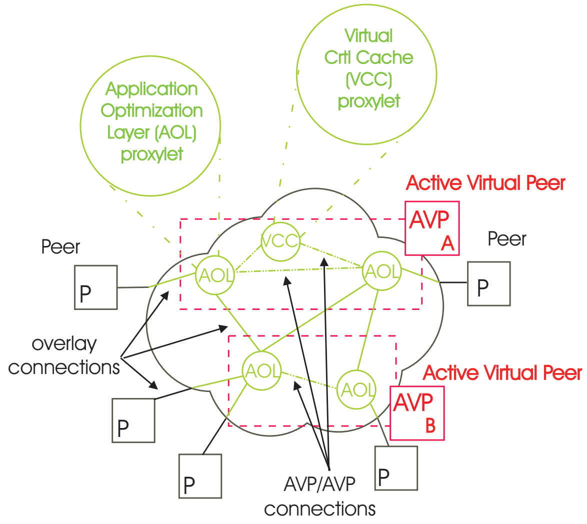

Fig. 12. The active virtual peer realm

tion level routing. Application Layer Routing entails a whole range of elements, reaching from self-configuring distributed EEP discovery to building up an application-specific connectivity mesh and topology maps and, finally, to dynamically form topology regions by clustering.

By using ALAN, the effectiveness of new services can be tested in "the wild" without compromising any existing network architectures

### C. The Active Virtual Peer Concept

The main contribution for dynamic p2p overlay operation rests on the introduction of *Active Virtual Peers (AVPs)*. Each AVP acts like a single, ordinary peer. An AVP, however, is thought to be a representative for a community of peers. Figure 12 depicts the ALAN-based AVP realm. Two Active Virtual Peers, marked by dashed boxes and letters "A" and "B", are located within an Internet cloud. Multiple ordinary peers, denoted by "P", maintain p2p overlay connections to the AVPs. The AVPs impose control on the overlay connection as well as they maintain overlay connections to each other.

#### Active Virtual Peers

The AVP functions are arranged in horizontal layers as well as in vertical planes, cf. Figure 13. The horizontal layers correspond to the layers on which an AVP imposes control. The vertical separation describes the functional planes of AVPs.

#### Horizontal layering

The upper layer of AVPs is called the "Application Optimization Layer (AOL)". It controls and optimizes the peer-to-peer relation on application level. The AOL may apply *application-level routing* in conjunction with policies similar to rules used for *Inter-Domain Policy Routing*. The policies implemented so far by an AOL are *access restrictions*. The AOL applies also routing policies using the *(virtual) peer state* or the *(virtual overlay) link state*. Forwarding is based on peer load and overlay link characteristics such as drop rate, throughput, or delay.

In addition, an AOL allows for *active overlay topology control* which is accomplished in two ways. The Active Virtual Peer may initiate, accept, or terminate overlay relations based on access restrictions or topology features. Topology characteristics such as the number of overlay relations or the characteristic path length can be enforced or may govern the overlay structure. Furthermore, the AOL layer makes use of ALAN control mechanisms for implementing self-organization features. The AOL can initiate and execute AVP modules whenever and whereever needed. The virtual overlay structure may adapt itself to varying demand and traffic patterns by launching new overlay relations and new virtual peers.

The middle layer of an AVP is denoted as the "Virtual Control Cache (VCC)". The VCC provides content caching on application level similar to conventional proxies.

The lower layer of AVPs is denoted as the "Network Optimization Layer (NOL)". Its main task is the implementation of dynamic traffic engineering capabilities which map p2p traffic onto the physical network in an optimized way. The mapping is performed with respect to the performance control capabilities of the applied transport technology.

#### Vertical planes

Orthogonally to the layering of the service levels, an AVP exhibits a vertical separation into three functional planes: a) *topology control*, b) *policy control*, and c) *performance monitoring*.

Topology control on application level comprises explicit initiation and termination of overlay connections and

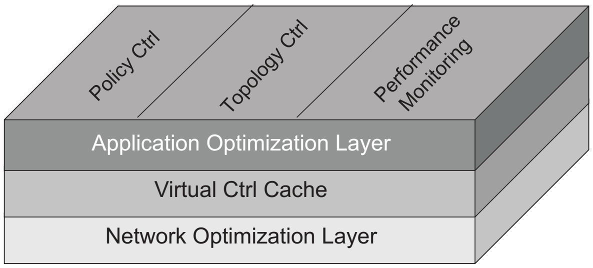

Fig. 13. Active Virtual Peer structure - Horizontal Layering and Vertical Planes

AVPs. On network level, however, topology control is limited. Only traffic engineering and traffic control functions are applied. This is for scalability, efficiency, and flexibility reasons.

Policy control on application layer includes access restrictions on a peer or peer group basis for content and p2p control information. The VCC may implement policies by localization or aggregation of messages. In addition, coordinated caching strategies between the AVP modules might be applied. The network optimization layer may enforce policies on the traffic volume allowed to be transmitted.

Performance monitoring capabilities of AVPs on the application level include an auditing of the number of relayed and dropped messages, logging of message inter-arrival times, monitoring of application response times, and active collection of topological information, such as the degree of the connectivity of a peer. This data provides information on robustness of the overlay and is used for controlling the overlay and the application layer routing decisions. On the network level, a proxylet can monitor the round trip delay, link error rate, or throughput.

### AVP benefits

AVPs provide four main benefits. First, they allow for on-demand resource aggregation on application-level. This improves the stability of the service. Second, AVPs permit separation and controlled interference between network layer and application layer. Third, AVPs provide caching on application-level. Fourth, AVPs enable and facilitate self-organization for dynamic operation of virtual overlay networks.

## VII. FIELD TRIAL

The suggested ALAN-enabled AVP architecture is currently implemented and prepared for use in a field trial at the University College London, the University of Würzburg, and BTExact. The scenario of this trial is to control the Gnutella service between a university's dormitory and the global Gnutella domain, cf. Figure 14. Two control objectives are demonstrated in the trial, *access control* and *self-organization.*

The trial consists of one AVP, consisting of two AOL proxylets and one VCC proxylet. The first AOL proxylet (AOL 1) is located at the student dormitory, the second AOL proxylet (AOL 2) and the VCC proxylet are placed at the Computer Science Department. The AOL proxylets control three types of p2p communication relationships, see Figure 14: *a)* to ordinary peers in the student dormitory domain - marked by [a], *b)* intra-AVP connections - marked by [b], and c) to Gnutella peers in the global domain - marked by [c].

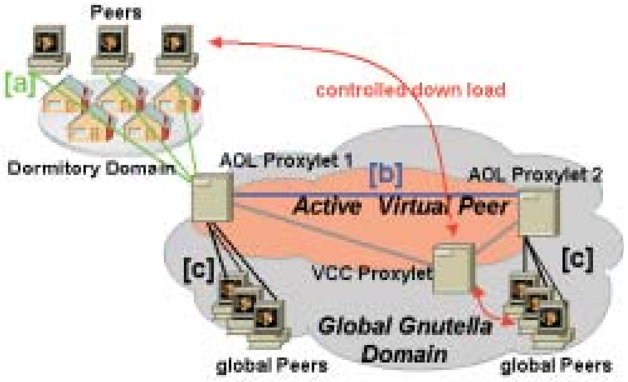

Fig. 14. Student dormitory scenario

AOL 1 enforces access control on application level by changing the originator address information of Gnutella messages and by aggregation and blocking of messages emitted from the student residence domain. AOL 1 and AOL 2 implement the AVP's self-organization feature by applying application-specific routing and coordinated overlay connection initiation. In this way the AVP separates the student residence domain from the global Gnutella sphere while balancing the load between the overlay connections it maintains.

## VIII. RELATED WORK

Diverse measurement studies of Gnutella have been performed recently. The measurements of Adar et al. [8] have revealed the strong asymmetry between content providers and content consumers. Nearly 70% of the Gnutella users share no files, and nearly 50% of all responses are returned by the top 1% of sharing hosts. The measurements of Saroiu et al. [12] characterized the end-user hosts participating in Gnutella. They provided values and distributions for bottleneck bandwidth, IP-level latencies, how often hosts connect and disconnect from the overlay. Jovanovic et al. [13] investigated the connectivity and the degree of cooperation between peers. Their results provided evidence that the degree distribution of the Gnutella network topology follows a *Power-law*. The measurements and experiments of Vaucher et al. [14] demonstrated that the composition of the community changes quite rapidly.

The measurements presented in this paper complement the ones reported in the previous publications. The focus of the investigations discussed here, is on the variability and time scales in the overlay of the Gnutella service. This aspect has not yet been properly addressed.

A recent investigation of p2p architectures has shown that the topology of p2p overlays has significant impact on the service performance [15]. Despite that observation, only few measurement studies of p2p architectures have been carried out which characterize the typical dynamics of peers that choose to participate in an overlay connection [12]. Quantitative results on the performance management capabilities of p2p architectures are limited and more research is required [16].

An approach for dynamic p2p service operation is the use of Routing Indices [17]. Routing Indices allow peers to forward messages and queries to neighbors that are more likely to have answers. This approach, however, makes use of a given overlay topology regardless whether the topology is suitable or not, while our approach supports control of dynamic topology creation.

## IX. CONCLUSION

In this paper we have presented a measurement study on signaling in Gnutella overlay networks. Both signaling load and the scale of variability in the existence of p2p overlay connections have been investigated. The variability of the overlay was characterized by two factors: a) the number of simultaneous overlay connections maintained by a peer and b) the duration of maintaining these connections. We presented a first simple statistical model of the process of maintaining overlay connections and provided estimates on some model parameters. It has been validated by measurements that p2p services are prone to highly variable connectivity patterns and traffic load profiles.

Today's peer-to-peer networks exhibit signaling characteristics that are anticipated to be typical for future Internet services. We expect a new management architecture to be needed for future applications that can handle specific granularities in time as well as in space to enable dynamic and adaptive operation of future virtual overlay networks.

According to the end-to-end argument [18], application layer services should predominantly be provided on the application layer itself rather than on lower layers. The scales in p2p overlay dynamics call for adaptive and dynamic overlay operation and control. The concept of *Active Virtual Peers* has been introduced to specifically address operation and control on scales of time and space that are typical for p2p overlay networks. It has been practically demonstrated in a real-world trial at our university campus network how application-level routing and dynamic topology creation can be used to control p2p overlays and how these methods may increase p2p overlay efficiency. Further studies are needed, however, to validate the effectiveness of our approach on larger p2p overlays, possibly with the inclusion of globally distributed partners. As a first step towards a more global deployment of AVPs, extensive simulation studies are currently under way to show the effectiveness of our topology management and p2p operation scheme.

**Acknowledgement:** This work has been funded in part by BTExact within the Alpine Project.

## REFERENCES

[1] Anonymous, "The Gnutella Protocol Specification v0.4," available at http://dss.clip2.com, Clip2 Distributed Search Solutions, 2001.

[2] Information available at http://www.kazaa.com , 2002.

[3] Information available at http://www.edonkey2000.com/ , 2002.

[4] W3C, "Simple Object Access Protocol (SOAP) 1.1," Available at http://www.w3.org/tr/soap/, W3C, 2000.

[5] Anonymous, "The Gnutella protocol," available at http://gnutelladev.wego.com, GnutellaDev, 2001.

[6] Anonymous, "Gnut - console Gnutella client for Linux and Windows," available at http://www.gnutelliums.com/linux_unix/gnut/, 2001.

[7] Anonymous, "LimeWire Version 2.7," available at http://www.limewire.com/, Lime Wire LCC, 2002.

[8] Eytan Adar and Bernardo A. Huberman, "Free riding on gnutella," Research report, Xerox Palo Alto Research Center, October 2000.

[9] G. Ash, "Traffic Engineering & QoS Methods for IP-, ATM-, & TDM-Based Multiservice Networks," Internet draft: <draft-ietf-tewg-qos-routing-04.txt> available at www.ietf.org, IETF, 2001.

[10] D. Clark, K. Sollins, J. Wroclawski, and R. Braden, "Tussle in cyberspace: Defining tomorrow's internet," in *Proceedings of the ACM SIGCOMM Conference 2002, Pittsburgh, 19-23 August, 2002.*

[11] A. Ghosh, M. Fry, and J. Crowcroft, "An architecture for application layer routing," in *Active Networks, LNCS 1942*, H. Yasuda, Ed. 2000, pp. 71–86, Springer.

[12] S. Saroiu, P. Gummadi, and S. Gribble, "A measurement study of peer-to-peer file sharing systems," in *SPIE Multimedia Computing and Networking (MMCN2002), San Jose, CA, Jan 18-25, 2002.*

[13] M. Jovanovic, F. Annexstein, and K. Berman, "Scalability issues in large peer-to-peer networks - a case study of gnutella," Technical report, University of Cincinnati, 2001.

[14] J. Vaucher, P. Kropf, G. Babin, and T. Jouve, "Experimenting with gnutella communities," in *Distributed Communities on the Web (DCW 2002), Sydney, Australia, Apr. 2002.*

[15] L. Adamic, R. Lukose, A. Puniyani, and B. Huberman., "Search in Power-Law Networks," available at www.parc.xerox.com/istl/groups/iea/papers/plsearch/, 2001.

[16] A. Oram, "Research Possibilities in Peer-to-Peer Networking - Speech delivered at the Virtual Internet2 Member Meeting, Oct. 2-5 2001, available at http://www.internet2.edu/," .

[17] A. Crespo and H. Garcia-Molina, "Routing indices for peer-to-peer systems," in *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria.*, July 2002.

[18] D. Reed, J. Saltzer, and D. Clark, "Commentaries on active networking and end-to-end arguments," *IEEE Network*, vol. 12, no. 3, 1998.