[·] Hermann De Meer, Kurt Tutschku and Phuoc Tran-Gia. Dynamic operation of peer-to-peer overlay networks. *Praxis der Informationsverarbeitung und Kommunikation (PIK)*, Volume 26, Number 2, pages 65–73, Dec. 2002.

See `http://www.net.fim.uni-passau.de/papers/DeMeer2002a` for full reference details (BibTeX, XML).

Dynamic Operation of Peer-to-Peer Overlays

H. deMeer and K. Tutschku

Abstract—**Virtual overlay networks, such as formed in 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 usage of shared resources. This paper examines the requirements of operating dynamic overlays, in particular, for peer-to-peer services. The analysis has been 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 infrastructue to implement a distributed control scheme for p2p overlays a so-called Application-Layer Active Networking platform has been chosen. Based on Application-Layer Active Networking, *Active Virtual Peers* are introduced as the main concept for dynamic operation and management of peer-to-peer overlay networks. Active Virtual Peers facilitate policy enforcement or performance management by means of self-organization, predominantly on the application layer with minimum interference on lower layers.**

I. INTRODUCTION

Peer-to-Peer (p2p) networks have become very popular recently amid the relentless spread of Napster and Gnutella music file sharing applications within an active user community. 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 run completely at the edges of a network. A lack of centralized control predictably leads to a huge amount of uncontrolled signaling traffic being generated and transmitted.

The current challenge is therefore to provide attractive p2p services, however, without compromising network services offered to concurrent applications and without sacrifying other user experiences in using network services. An effective management system for overlay network could have large benefits to a wider 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 a major obstacle to a wide-spread usage of these services. In this paper, we suggest a new concept for dynamic operation of p2p overlay networks. The approach applies Application-Layer Active Networking and introduces *Active Virtual Peers* (AVP).

II. A CHARACTERIZATION OF P2P OVERLAYS

Signaling messages are routed in the Gnutella overlay by using two simple principles [1]: *a) broadcast* to all neighbors, i.e. sent to all nodes with which the sender has open TCP connections, and *b)* responses are *back-propagated* in the overlay along the path taken by the triggering message.

An important feature of Gnutella p2p filesharing services as well as other p2p architectures 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. In Gnutella, Peers acquire new candidates for their overlay connections by sending periodically “Ping” messages to neighbors and by inspecting “Pong” responses. Nodes base their decision where to connect to in the network on their local information. The Gnutella protocol doesn’t provide any support for a coordinated organization of the signaling overlay. The Gnutella service forms an randomly structured overlay network.

While qualitative justification is straightforward, little has been known of quantitative results on the scale of dynamics in overlays and p2p applications. We therefore investigated in particular characterization of time scale and variability of the number of virtual overlay connections [2].

The variability of p2p overlays 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. Real-world measurements revealed a maintaining of 9.86 relations on average by a typical peer. Most importantly, however, the connectivity process revealed a very high variability in the number of simultaneously maintained p2p connections. If the connectivity of a peer is high, i.e., a peer maintains 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.

A correlation analysis led to a two-state model for Gnutella p2p overlay relations. In the first state, which is called the “short” state, peers establish only short-lived connections used to exchange host information. In the other mode, denoted as the “stable” state, peers establish a long-duration relations and exchange continuously signaling messages, mostly search requests. From the perspective of a user, the “stable” permits uninter-

H. DeMeer: Dept. of Electronic & Electrical Engineering, University College London, London, UK. H.DeMeer@ee.ucl.ac.uk .

K. Tutschku: Institute of Computer Science, Univ. of Würzburg, Würzburg, Germany. tutschku@informatik.uni-wuerzburg.de .

rupted operation of the p2p service.

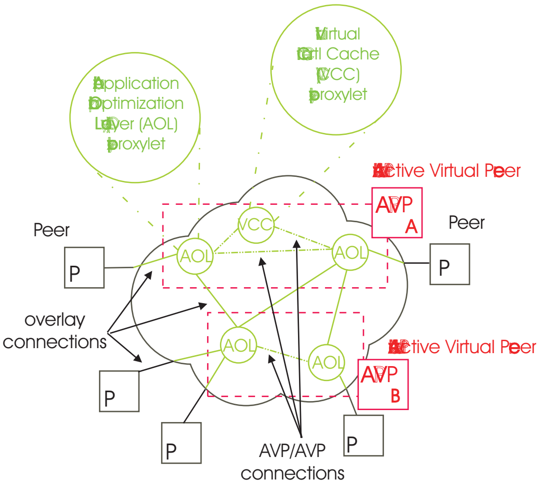

Fig. 1. The active virtual peer realm

III. MANAGING P2P OVERLAY WITH APPLICATION-LEVEL ACTIVE NETWORKS – THE "ACTIVE VIRTUAL PEER ARCHITECTURE"

A. *Management Objectives*

P2P overlay operation and control has to facilitate two objectives. First, p2p overlays should be operated in an application specific way. It becomes more and more apparent that application requirements can't be addressed purely by network layer functions, e.g., in a scalable and efficient way, whereas the requirements can well be dealt with on the application level. In consequence, this calls for a management architecture that has universal programmability on the application layer for performance control as well as for group management.

Second, p2p overlay control should be equipped with handles for 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.: active networks on application level [3], *ii)* automatic load-balancing on network elements on small times scales, and *iii)* the integration of self-organization and adaptiveness on application-level.

B. *ALAN architecture*

Active and Programmable Networks are being widely investigated as a possible vehicle for deploying new services into existing network infrastructures on demand [4], [5]. An alternative approach has been proposed in [3], where so-called Application-Level Active Networking (ALAN) is pursued. Very much like the Internet Protocol itself when it was originally introduced, ALAN has been based on an overlay technique. Active nodes, which operate on the application level, are strategically placed within the network. Active nodes allow a dynamic loading of code from special servers and the resulting services may interfere with data transport and control. A central service of ALAN is multi-metric application level routing, including active node discovery and state maintenance. Generic computing facilities, called Execution Environments for Proxylets (EEPs), are placed strategically inside 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. More generally, Application Layer Routing provides mechanisms for EEP discovery, application specific routing exchanges, service creation by deploying a web of proxylets across the physical infrastructure, and information routing by the proxylets once proxylets have been launched. So 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 forming topology regions by clustering. Clustering is achieved by the Self-Organising Application-Level Routing (SOAR) mechanisms, as described in more detail in [3], where protocols and implementation details are presented. SOAR can be seen as an essential technology for building self-organising network services on the application layer based on the ALAN infrastructure.

By using ALAN, the effectiveness of new services can be tested in "the wild" without compromising any existing network architectures. We are arguing, however, that application layer services should *predominantly* be provided on the application layer itself anyway, rather than involving lower layers, [6], [7] as claimed in the end-to-end arguments, even if that may lead to compromises in quality of service. Thus a use of overlay networks may provide sufficiently potent management schemes to achieve this aim. We believe that self-organization of application layer overlay networks may hold the key to the solution of that management task.

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 representative for a community of peers. Figure 1 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.

The AVP functions are arranged in horizontal layers as well as in vertical planes, cf. Figure 2. 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 transit 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 restriction or topology features. Topology characteristics such as number of overlay relations or 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 when ever and where ever 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. In addition, the VCC may offer control flow aggregation functions.

The lower layer of AVPs is denoted as the "Network Optimization Layer (NOL)". Its main task is the implemantation 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. The AVP architecture may support traffic engineering for standard IP routing protocols as well as for explicit QoS enabled mechanisms like MPLS.

Vertical planes

Orthogonally to the layering of 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 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.

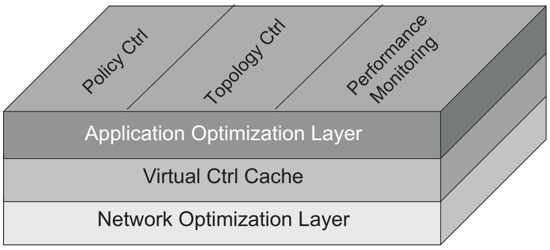

Fig. 2. Active Virtual Peer structure - Horizontal Layering and Vertical Planes

Policy control on application layer includes access restriction 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 include on the application level an auditing of the number of relayed and dropped messages, logging of message interarrival 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 service stability. Second, AVPs permit separation of and limited and controlled interference between network layer and application layer. Third, AVPs provide caching on application-level. Forth, AVPs enable and facilitate self-organization for dynamic operation of virtual overlay networks.

Additional information on dynamic overlay control using the Active Virtual Peer concept can be found at [8].

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] K. Tutschku and H. deMeer, "A Measurement Study on Signalling on Gnutella Overlay Networks," in *Proceedings of "Fachtagung - Kommunikation in Verteilten Systemen (KiVS) 2003", Leipzig, Germany*, February 2003.

[3] 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.

[4] A. Campbell, H. De Meer, M. Kounavis, K. Miki, J. Vicente, and D. Villela, "A survey of programmable networks," *Computer Communications Review*, vol. 29, no. 2, 1999.

[5] David L. Tennenhouse and David J. Wetherall, "Towards an active network architecture," *Computer Communication Review*, vol. 26, no. 2, 1996.

[6] Jerome H. Saltzer, David P. Reed, and David D. Clark, "End-to-end arguments in system design," *ACM Transactions on Computer Systems*, vol. 2, no. 4, pp. 277–288, Nov. 1984.

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

[8] "Homepage of UCL Networking and Distributed Systems Group: http://www.ee.ucl.ac.uk/netdist/," .