

A Performance Management Architecture for Peer-to-Peer Services based on Application-level Active Networks

H. deMeer

Dept. of Electronic & Electrical Eng.
University College London
Torrington Place
London WC1E 7JE, U.K.
H.DeMeer@ee.ucl.ac.uk

K. Tutschku

Institute of Computer Science
University of Würzburg
Am Hubland
D-97074 Würzburg, Germany.
tutschku@informatik.uni-wuerzburg.de

Abstract

We propose an application-level active networking-based architecture using modular “active proxylets” for managing p2p service. The approach combines application-level and network-level performance control.

1. Introduction

Peer-to-peer (p2p) services are gaining increasing popularity. Although they are intended for environments where a full management is not wanted or not feasible, in large scale networks more and more scenarios emerge where specialized performance management functions are required in order to preserve their functional integrity and maintain their advantages.

Instead of traditional routing, the highly distributed p2p services, like the Gnutella file sharing application, cf. [2], use broadcast mechanisms with simple metrics to relay messages. There doesn't exist a stable or a predictable connection between two arbitrary peers. In addition, due to their distributed architecture, p2p systems generate a reasonable amount of synchronization and control traffic in addition to the user data traffic. The volume of control traffic, in turn, mainly determines scalability and performance of p2p services, cf. [6]. Finally, the constantly changing set of participating peers reduces the stability of the p2p network and increases the complexity of locating the optimal resource.

Three options are available to facilitate resource and performance management for p2p services. The first one is to *enhance p2p protocols* by topology construction mechanisms, e.g. as implemented in the Sun supported project JXTA [5], or by sophisticated group multicasting mechanisms [3]. The second option is the use of the *accountability* concept, cf. Mojo Nation project [4]. The third option is the introduction of a transparent but separate control architecture which will provide control handles for p2p performance management functions.

This architectural variant may be of advantages particular in large scale environments, since it permits the system to start out as a pure p2p network

and then scales by introducing a hierarchy of “distinguished” peers. These peers can apply elaborate concepts like *self-organization* and *traffic aggregation* to increase p2p network stability and performance.

2. Managing P2P Services Performance with ALAN

To tackle the issue of p2p performance management in large scale environments, we propose an *Application-Level Active Networking (ALAN)* based architecture using modular *proxylets*, cf. [1]. The proxylets are denoted as “active peers” and may form a single or multiple “virtual active peers” for the other peers. The functions of the proxylet module are a) the management of the *Peer-to-Peer* relation, b) *Virtual Control Caching* capability, and c) implementing *Dynamic Traffic Engineering*. The proxylet is executed on an active networking element, cf. Figure 1.

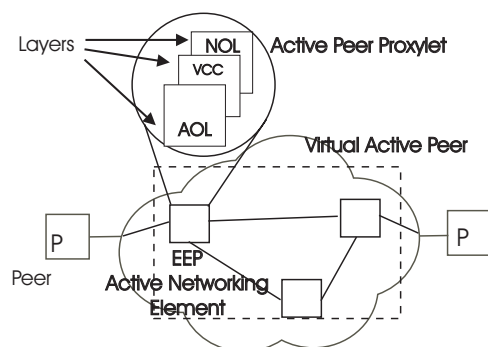


Figure 1: ALAN-based Performance Management Architecture for P2P Services

The proxylet functions are implemented as hierarchical layers. The upper layer is denoted as the “Application Optimization Layer (AOL)” and controls the peer-to-peer relation on application level. The AOL may apply application-level routing which can be optimized for different parameters such as privacy, policies or latency. In this layer, for example, control can be implemented by relaying messages only from peers with small response times. As a result of this control, the stability of the p2p system is increased since only connections between peers of predictable performance are allowed. In addition, the scalability is increased since the volume of the control traffic is reduced.

The middle layer is denoted as the “Virtual Control Cache (VCC)”. The VCC provides application-level aggregation capability and administrative control functions. If many peers request to connect to the same peer or to peers in the same group, the requests are combined and handled similarly. This feature is comparable to *Border Gateway Protocol (BGP)* routing aggregation, however it is performed on application level. Due to the application-level

aggregation the stability of the p2p system is increased. In addition, the aggregation provides better command on p2p performance objectives. Beside the aggregation also the differentiation of the requests can be facilitated in the VCC layer. In this way, administrative control functions can be implemented. The VCC may also implement smart multicasting, caching and replication capabilities.

The lower layer is denoted as the “Network Optimization Layer (NOL)”. It provides dynamic traffic engineering capabilities and maps the p2p traffic onto the network in an optimal way. The mapping is performed with respect to the performance control capabilities of the applied transport technology. The architecture can support legacy IPv4 transport mechanisms as well as future QoS enabled mechanisms like DiffServ or MPLS. The NOL layer enables the prediction of the performance of a connection between two arbitrary peers.

Self-organization

The “active peer” proxylets may facilitate self-organizing in p2p overlays. For example, the AOL layer monitors the response times for search queries of a p2p files sharing system within its area under control. The proxylets may accept only peers to their domain which meet the specified performance objectives on this metric. In this way, the proxylets form zones of equal *performance levels*. In addition, since the ALAN-based architecture allows proxylets to determine their location in the network by dynamic self-organization, the suggested control architecture can adapt itself to spatially varying performance conditions in the network.

The self-organization feature has the effect that the proposed p2p performance architecture is able to identify performance boundaries in the network. This feature can be used either in a hard way to implement application-level access control, or in a soft sense to inform the user about leaving its domain of equal performance.

References

- [1] A. Ghosh, M. Fry, and J. Crowcroft. An architecture for application layer routing. In H. Yasuda, editor, *Active Networks, LNCS 1942*, pages 71–86. Springer, 2000.
- [2] Gnutella Development Home Page. <http://gnutella.wego.com/>.
- [3] S. Lee, W. Su, and M. Gerla. On-demand multicast routing protocol in multihop wireless mobile networks. In *ACM/Baltzer Mobile Networks and Applications, Special Issue on Multipoint Communication in Wireless Mobile Networks, 2001, to appear*.
- [4] Mojo Nation - Home Page. <http://www.mojonation.net/>.
- [5] Project JXTA - Home Page. <http://www.jxta.org/>.
- [6] J. Ritter. Why Gnutella Can't Scale. No, Really. available at <http://www.darkridge.com/jpr5/doc/gnutella.html>, 2001.