pCache: A Proxy Cache for Peer-to-Peer Traffic

Mohamed Hefeeda, Cheng-Hsin Hsu, and Kianoosh Mokhtarian
School of Computing Science, Simon Fraser University, Surrey, BC, Canada

ABSTRACT

Peer-to-peer (P2P) applications generate a huge amount of traffic. We believe that caching of P2P traffic is a promising approach to mitigate the negative effects of P2P traffic. We have designed and implemented a proxy cache for P2P traffic, which we call pCache. pCache *transparently* intercepts and serves traffic from different P2P systems, while not affecting other Internet applications.

Categories and Subject Descriptors: C.2[Computer Systems Organization]: Computer-Communication Networks

General Terms: Design

1. INTRODUCTION

File-sharing using peer-to-peer (P2P) systems currently generate a huge amount of traffic. This huge traffic costs university campuses thousands of dollars every year. Internet service providers (ISPs) also suffer from P2P traffic, because it increases the load on their routers and links. To mitigate the negative effects of P2P traffic, several approaches have been proposed in the literature, such as designing locality-aware neighbor selection algorithms [1] and caching of P2P traffic [5]. We believe that caching of P2P traffic is a promising approach, because objects in P2P systems are mostly immutable and the traffic is highly repetitive [4]. In addition, caching does not require changing P2P protocols and can be deployed transparently from clients. Therefore, ISPs can readily deploy caching systems to reduce their costs. We have designed and implemented a proxy cache for P2P traffic, which we call pCache. pCache *transparently* intercepts and serves traffic from different P2P systems, while not affecting other Internet applications.

The proposed pCache is to be deployed by autonomous systems (ASes) or ISPs that are interested in reducing the burden of P2P traffic. pCache would be deployed at or near the gateway router of an AS. At a high-level, a client participating in a particular P2P network issues a request to download an object. This request is intercepted by pCache. If the requested object or parts of it are stored in the cache, they are served to the requesting client. This saves bandwidth on the external (expensive) links to the Internet. If a part of the requested object is not found in the cache, the request is forwarded to the P2P network. When the response comes back, pCache may store a copy of the object for future requests from other clients in its AS. Clients inside the AS as well as external clients are not aware of pCache, i.e., pCache is fully-transparent.

Copyright is held by the author/owner(s).
SIGCOMM'08, August 17–22, 2008, Seattle, Washington, USA.
ACM 978-1-60558-175-0/08/08.

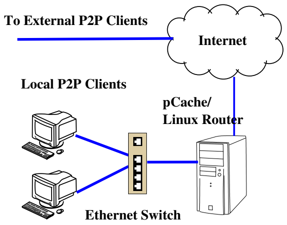

Figure 1: The setup to demonstrate pCache.

Several subtle issues need to be addressed while implementing pCache. For example, deciding whether a connection belongs to a P2P system is not as easy as in web caching systems, because many P2P systems use dynamic ports. In addition, clients in P2P systems issue requests for *segments* (byte ranges) of objects, not for entire objects. As P2P objects have large sizes [2, 5] and segment lengths are variable, new storage management schemes is needed.

2. SETUP OF THE DEMO

The setup of the demo is shown in Fig. 1. There is a Linux server configured to run as a router and as a cache for P2P traffic. Two client machines are connected to the server through an Ethernet switch. The TCP/IP stacks of the two machines are configured to use the Linux server as the default gateway router, but they do not know about the presence of pCache. To demo the operation of pCache, several download sessions using common BitTorrent and Gnutella clients will be issued from the client machines. These sessions will be *transparently* intercepted by pCache. Previously cached data will be served to the clients while other data will be transfered from external peers on the Internet to the local clients. All on-going P2P sessions will be shown through a web interface that we developed for pCache, which will show the detailed operations in real time and the statistical results over longer periods.

3. REFERENCES

[1] R. Bindal and et al. Improving traffic locality in BitTorrent via biased neighbor selection. In *Proc. of IEEE ICDCS 2006*, Lisboa, Portugal.

[2] K. Gummadi and et al. Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In *Proc. of ACM SOSP 2003*, Bolton Landing, NY.

[3] M. Hefeeda and et al. Transparent proxy caching of peer-to-peer traffic. Technical report, School of Computing Science, Simon Fraser University, April 2008. http://nsl.cs.sfu.ca/wiki/.

[4] N. Leibowitz and et al. Are file swapping networks cacheable? In *Proc. of WCW 2002*, Boulder, CO.

[5] O. Saleh and M. Hefeeda. Modeling and caching of peer-to-peer traffic. In *Proc. of IEEE ICNP 2006*, Santa Barbara, CA.