DECADE
Internet-Draft
Intended status: Informational
Expires: August 11, 2012

H. Song
N. Zong
Huawei
Y. Yang
Yale University
R. Alimi
Google

February 8, 2012

# DECoupled Application Data Enroute (DECADE) Problem Statement
draft-ietf-decade-problem-statement-05

## Abstract

Peer-to-peer (P2P) applications have become widely used on the Internet today and make up a large portion of the traffic in many networks. In P2P applications, one technique for reducing the transit and uplink P2P traffic is to introduce storage capabilities within the network. Traditional caches (e.g., P2P and Web caches) provide such storage, but they are complex (due to explicitly supporting individual P2P application protocols and cache refresh mechanisms) and they do not allow users to manage access to content in the cache. For example, content providers wishing to use in-network storage cannot easily control cache access and resource usage policies to satisfy their own requirements. This document discusses the introduction of in-network storage for P2P applications, and shows the need for a standard protocol for accessing this storage.

## Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 11, 2012.

## Copyright Notice

Table of Contents

## 1. Introduction

P2P applications, including both P2P streaming and P2P filesharing applications, make up a large fraction of the traffic in many ISP networks today. One way to reduce bandwidth usage by P2P applications is to introduce storage capabilities in the networks. Allowing P2P applications to store and retrieve data from inside networks can reduce traffic on the last-mile uplink, as well as on backbone and transit links.

P2P caches provide in-network storage and have been deployed in some networks. However, the current P2P caching architecture poses challenges to both P2P cache vendors and P2P application developers. For P2P cache vendors, it is challenging to support a number of continuously evolving P2P application protocols, due to lack of documentation, ongoing protocol changes, and rapid introduction of new features by P2P applications. For P2P applications, closed P2P caching systems limit P2P applications from effectively utilizing in-network storage. In particular, P2P caches typically do not allow users to explicitly store content into in-network storage. They also do not allow users to implement control over the content that has been placed into the in-network storage.

P2P applications suffer decreased efficiency, and the network infrastructure suffers increased load because there is no standardized interface for accessing storage and data transport services in the Internet.

Both of these challenges can be effectively addressed by using an open, standard protocol to access in-network storage. P2P applications can store and retrieve content in the in-network storage, as well as control resources (e.g., bandwidth, connections) consumed by peers in a P2P application. As a simple example, a peer of a P2P application may upload to other peers through its in-network storage, saving its usage of last-mile uplink bandwidth.

In this document, we distinguish between two functional components of the native P2P application protocol: signaling and data access. Signaling includes operations such as handshaking and discovering peer and content availability. The data access component transfers content from one peer to another.

In essence, coupling of the signaling and data access makes in-network storage very complex to support various application services. However, these applications have common requirements for data access, making it possible to develop a standard protocol.





## 2. Terminology and Concepts

The following terms have special meaning in the definition of the in-network storage system.

in-network storage: A service inside a network that provides storage and bandwidth to network applications. In-network storage may reduce upload/transit/backbone traffic and improve network application performance.

P2P cache (Peer to Peer cache): A kind of in-network storage that understands the signaling and transport of specific P2P application protocols. It caches the content for those specific P2P applications in order to serve peers and reduce traffic on certain links.

## 3. The Problems

The emergence of peer-to-peer (P2P) as a major network application (especially P2P file sharing and streaming) has led to substantial opportunities. The P2P paradigm can be utilized to design highly scalable and robust applications at low cost, compared to the traditional client-server paradigm. For example, CNN reported that P2P streaming by Octoshape played a major role in its distribution of the historic inauguration address of President Obama[Octoshape]. PPLive, one of the largest P2P streaming vendors, is able to distribute large-scale, live streaming programs to more than 2 million users with only a handful of servers [PPLive].

However, P2P applications also face substantial design challenges. A particular problem facing P2P applications is the additional stress that they place on the network infrastructure. Furthermore, lack of infrastructure support can lead to unstable P2P application performance during peer churns and flash crowds, when a large group of users begin to retrieve the content during a short period of time. These problems are now discussed in further detail.

## 3.1. P2P infrastructural stress and inefficiency

A particular problem of the P2P paradigm is the stress that P2P application traffic places on the infrastructure of Internet service providers (ISPs). Multiple measurements (e.g., [Internet Study 2008/2009][Internet Study 2008-2009]) have shown that P2P traffic has become a major type of traffic on some networks. Furthermore, the inefficiency of network-agnostic peering (at the P2P transmission level) leads to unnecessary traversal across network domains or spanning the backbone of a network [RFC5693].





Using network information alone to construct more efficient P2P swarms is not sufficient to reduce P2P traffic in access networks, as the total access upload traffic is equal to the total access download traffic in a traditional P2P system. On the other hand, it is reported that P2P traffic is becoming the dominant traffic on the access networks of some networks, reaching as high as 50-60% on the downlinks and 60-90% on the uplinks ([DCIA], [ICNP], [ipoque.P2P_survey.], [P2P_file_sharing]). Consequently, it becomes increasingly important to reduce upload access traffic, in addition to cross-domain and backbone traffic.

The inefficiency is also represented when traffic is sent upstream as many times as there are remote peers interested in getting the corresponding information. For example, the P2P application transfer completion times remain affected by potentially (relatively) slow upstream transmission. Similarly, the performance of real-time P2P applications may be affected by potentially (relatively) higher upstream latencies.

## 3.2. P2P cache: a complex in-network storage

An effective technique to reduce P2P infrastructural stress and

inefficiency is to introduce in-network storage.

In the current Internet, in-network storage is introduced as P2P caches, either transparently or explicitly as a P2P peer. To provide service to a specific P2P application, the P2P cache server must support the specific signaling and transport protocols of the specific P2P application. This can lead to substantial complexity for the P2P Cache vendor.

First, there are many P2P applications on the Internet (e.g., BitTorrent, eMule, Flashget, and Thunder for file sharing; Abacast, Kontiki, Octoshape, PPLive, PPStream, and UUSee for P2P streaming). Consequently, a P2P cache vendor faces the challenge of supporting a large number of P2P application protocols, leading to product complexity and increased development cost.

Furthermore, a specific P2P application protocol may evolve continuously, to add new features or fix bugs. This forces a P2P cache vendor to continuously update to track the changes of the P2P application, leading to product complexity and increased costs.

Third, many P2P applications use proprietary protocols or support end-to-end encryption. This can render P2P caches ineffective.

Finally, a P2P cache is likely to be much better connected to end hosts than to remote peers. Without the ability to manage bandwidth





usage, the P2P cache may increase the volume of download traffic, which runs counter to the reduction of upload access traffic.

## 3.3. Ineffective integration of P2P applications

As P2P applications evolve, it has become increasingly clear that usage of in-network resources can improve user experience. For example, multiple P2P streaming systems seek additional in-network resources during a flash crowd, such as just before a major live streaming event. In asymmetric networks when the aggregated upload bandwidth of a channel cannot meet the download demand, a P2P application may seek additional in-network resources to maintain a stable system.

However, some P2P applications using in-network infrastructural

resources require flexibility in implementing resource allocation policies. A major competitive advantage of many successful P2P systems is their substantial expertise in how to most efficiently utilize peer and infrastructural resources. For example, many live P2P systems have specific algorithms to select those peers that behave as stable, higher-bandwidth sources. Similarly, the higher-bandwidth sources frequently use algorithms to chose to which peers the source should send content. Developers of these systems continue to fine-tune these algorithms over time.

To permit developers to evolve and fine-tune their algorithms and policies, the in-network storage should expose basic mechanisms and allow as much flexibility as possible to P2P applications. This conforms to the end-to-end systems principle and allows innovation and satisfaction of specific business goals. Existing techniques for P2P application in-network storage lack these capabilities.

## 4. Usage Scenarios

Usage scenarios are presented to illustrate the problems in both CDN and P2P scenarios.

### 4.1. BitTorrent

When a BitTorrent client A uploads a block to multiple peers, the block traverses the last-mile uplink once for each peer. And after that, the peer B who just received the block from A also needs to upload through its own last-mile uplink to others when sharing this block. This is not an efficient use of the last-mile uplink. With in-network storage server however, the BitTorrent client may upload the block to its in-network storage. Peers may retrieve the block from the in-network storage, reducing the amount of data on the last-





mile uplink. If supported by the in-network storage, a peer can also save the block in its own in-network storage while it is being retrieved; the block can then be uploaded from the in-network storage to other peers.

As previously discussed, BitTorrent or other P2P applications currently cannot explicitly manage which content is placed in the existing P2P caches, nor can they manage access and resource control

polices. Applications need to retain flexibility to control the content distribution policies and topology among peers.

## 4.2. Content Publisher

Content publishers may also utilize in-network storage. For example, consider a P2P live streaming application. A Content Publisher typically maintains a small number of sources, each of which distributes blocks in the current play buffer to a set of the P2P peers.

Some content publishers use another hybrid content distribution approach incorporating both P2P and CDN modes. As an example, Internet TV may be implemented as a hybrid CDN/P2P application by distributing content from central servers via a CDN, and also incorporating a P2P mode amongst endhosts and set-top boxes. In-network storage may be beneficial to hybrid CDN/P2P applications as well to support P2P distribution and to enable content publisher standard interfaces and controls.

However, there is no standard interface for different content publishers to access in-network storage. One streaming content publisher may need the existing in-network storage to support streaming signaling or such capability, such as transcoding capability, bitmap information, intelligent retransmission, etc, while a different content publisher may only need the in-network storage to distribute files. However it is reasonable that the application services are only supported by content publisher's original servers and clients, and intelligent data plane transport for those content publishers are supported by in-network storage.

A content publisher also benefits from a standard interface to access in-network storage servers provided by different providers. The standard interface must allow the content publisher to retain control over content placed in their own in-network storage, and grant access and resources only to the desired endhosts and peers.

In the hybrid CDN/P2P scenario, if only the endhosts can store content in the in-network storage server, the content must be downloaded and then uploaded over the last-mile access link before





another peer may retrieve it from a in-network storage server. Thus,

in this deployment scenario, it may be advantageous for a content publisher or CDN provider to store content in in-network storage servers.

## 5. Security Considerations

There are several security considerations to the in-network storage.

### 5.1. Denial of Service Attacks

An attacker can try to consume a large portion of the in-network storage, or exhaust the connections of the in-network storage through a Denial of Service (DoS) attack. Authentication, authorization and accounting mechanisms should be considered in the cross domain environment. Limitation of access from an administrative domain sets up barriers for content distribution.

### 5.2. Copyright and Legal Issues

Copyright and other laws may prevent the distribution of certain content in various localities. In-network storage operators may adopt system-wide ingress or egress filters to implement necessary policies for storing or retrieving content, and applications may apply DRM to the data stored in the network storage. However, the specification and implementation of such policies (e.g., filtering and DRM) is outside of the scope of this document.

### 5.3. Traffic Analysis

If the content is stored in the provider-based in-network storage, there may be a privacy risk that the provider can correlate the people who are accessing the same data object using the same object identity.

### 5.4. Modification of Information

The modification threat is the danger that some unauthorized entity may alter in-transit in-network storage access messages generated on behalf of an authorized principal in such a way as to effect unauthorized management operations, including falsifying the value of an object. See [RFC3414].

### 5.5. Masquerade

The masquerade threat is the danger that management operations may be attempted by assuming the identity of another user that has the

appropriate authorizations. See [RFC3414].

### 5.6. Disclosure

The disclosure threat is the danger of eavesdropping on the exchanges between in-network storage and application clients. Protecting against this threat may be required as a matter of application policy. See [RFC3414].

### 5.7. Message Stream Modification

The message stream modification threat is the danger that messages may be maliciously re-ordered, delayed or replayed to an extent which is greater than can occur through natural network system, in order to effect unauthorized management operations. See [RFC3414].

## 6. IANA Considerations

There are no IANA considerations in this document.

## 7. Acknowledgments

We would like to thank the following people for contributing to this document:

David Bryan

Kar Ann Chew

Roni Even

Lars Eggert

Yingjie Gu

Francois Le Faucheur

Hongqiang Liu

Tao Ma

Borje Ohlman

Akbar Rahman

Yu-shun Wang





Richard Woundy

Yunfei Zhang

## 8. Informative References

[Internet_Study_2008-2009]
"Internet Study 2008/2009", <http://www.ipoque.com/resources/internet-studies/internet-study-2008_2009>.

[RFC3414] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, December 2002.

[RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, October 2009.

[I-D.ietf-p2psip-base]
Jennings, C., Lowekamp, B., Rescorla, E., Baset, S., and H. Schulzrinne, "REsource LOcation And Discovery (RELOAD) Base Protocol", draft-ietf-p2psip-base-20 (work in progress), January 2012.

[DCIA] http://www.dcia.info, "Distributed Computing Industry Association".

[ipoque.P2P_survey.]
"Emerging Technologies Conference at MIT", Sept. 2007.

[P2P_file_sharing]
Parker, A., "The true picture of peer-to-peer filesharing", July 2004.

[Octoshape]
"http://www.octoshape.com/?page=company/about".

[PPLive] "http://www.synacast.com/products/".

[ICNP] Wu, H., "Challenges and opportunities of Internet developments in China, ICNP 2007 Keynote", Oct. 2007.

## Appendix A. Other Related Work in IETF

(To the RFC editor: Please remove this section and the related references in this section upon publication. The purpose of this





section is to give the IESG and RFC editor a better understanding of the current P2P related work in IETF and the relationship with DECADE WG.)

Note that DECADE WG's work is independent of current IETF work on P2P. The ALTO work is aimed for better peer selection and the RELOAD [I-D.ietf-p2psip-base] protocol is used for P2P overlay maintenance and resource discovery.

The Peer to Peer Streaming Protocol effort in the IETF is investigating the specification of signaling protocols (called the PPSP tracker protocol and peer protocol) for multiple entities (e.g. intelligent endpoints, caches, content distribution network nodes, and/or other edge devices) to participate in P2P streaming systems in both fixed and mobile Internet. As discussed in the PPSP problem statement, one important PPSP use case is the support of an in-network edge cache for P2P Streaming. However, this approach to providing in-network cache has different applicability, different objectives and different implications for the in-network cache operator. The goal of DECADE WG is to provide in-network storage service that can be used for any application transparently to the in-network storage operator: it can be used for any P2P Streaming application (whether it supports PPSP protocols or not), for any other P2P application, and for non P2P applications that simply want to benefit from in-network storage. With DECADE, the operator is providing a generic in-network storage service that can be used by any application without application involvement or awareness by the operator; in the PPSP cache use case, the cache operator is participating in the specific P2P streaming service.

DECADE and PPSP can both contribute independently, and (where appropriate) simultaneously, to making content available closer to

peers. Here are a number of example scenarios:

A given network supports DECADE in-network storage, and its CDN nodes do not participate as PPSP Peers for a given "stream" (e.g. because no CDN arrangement has been put in place between the content provider and the particular network provider). In that case, PPSP Peers will all be "off-net" but will be able to use DECADE in-network storage to exchange chunks.

A given network does not support DECADE in-network storage, and (some of) its CDN nodes participate as PPSP Peers for a given "stream" (e.g. say because an arrangement has been put in place between the content provider and the particular network provider). In that case, the CDN nodes will participate as in-network PPSP Peers. The off-net PPSP Peers (i.e., end users) will be able to get chunks from the in-network CDN nodes (using PPSP protocols





with the CDN nodes).

A given network supports DECADE in-network storage, and (some of) its CDN nodes participate as PPSP Peers for a given "stream" (e.g. because an arrangement has been put in place between the content provider and the particular network provider). In that case, the CDN nodes will participate as in-network PPSP Peers. The off-net PPSP Peers (i.e., end users) will be able to get chunks from the in-network CDN nodes (using PPSP protocols with the CDN nodes) as well as be able to get chunks / share chunks using DECADE in-network storage populated by PPSP Peers (both off-net end-users and in-network CDN Nodes).

PPSP and DECADE jointly provide P2P streaming service for heterogeneous networks including both fixed and mobile connections and enables the mobile nodes to use DECADE. In this case there may be some solutions that require more information in PPSP tracker protocol, e.g., the mobile node can indicate its DECADE in-network proxy to the PPSP tracker and the following requesting peer can finish data transfer with the DECADE proxy.

An ALTO (Application Layer Traffic Optimization) server provides P2P applications with network information so that they can perform better-than-random initial peer selection [RFC5693]. However, there are limitations on what ALTO can achieve alone. For example, network

information alone cannot reduce P2P traffic in access networks, as the total access upload traffic is equal to the total access download traffic in a traditional P2P system. Consequently, it becomes increasingly important to complement the ALTO effort and reduce upload access traffic, in addition to cross-domain and backbone traffic.

The IETF Low Extra Delay Background Transport (LEDBAT) Working Group is focusing on techniques that allow large amounts of data to be consistently transmitted without substantially affecting the delays experienced by other users and applications. It is expected that some P2P applications would start using such techniques, thereby somewhat alleviating the perceivable impact (at least on other applications) of their high volume traffic. However, such techniques may not be adopted by all P2P applications. Also, when adopted, these techniques do not remove all inefficiencies, such as those associated with traffic being sent upstream as many times as there are remote peers interested in getting the corresponding information. For example, the P2P application transfer completion times remain affected by potentially (relatively) slow upstream transmission. Similarly, the performance of real-time P2P applications may be affected by potentially (relatively) higher upstream latencies.





Authors' Addresses

Haibin Song
Huawei
101 Software Avenue, Yuhua District,
Nanjing, Jiangsu Province 210012
China

Phone: +86-25-56624792
Email: haibin.song@huawei.com

Ning Zong
Huawei
101 Software Avenue, Yuhua District,
Nanjing, Jiangsu Province 210012
China

Phone: +86-25-56624760
Email: zongning@huawei.com

Y. Richard Yang
Yale University

Email: yry@cs.yale.edu

Richard Alimi
Google

Email: ralimi@google.com