DECADE H. Song
Internet-Draft N. Zong
Intended status: Standards Track Huawei
Expires: February 25, 2011 Y. Yang
R. Alimi
Yale University
August 24, 2010

DECoupled Application Data Enroute (DECADE) Problem Statement
draft-ietf-decade-problem-statement-00

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 total amount of 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 they do not allow users to manage access to content in the cache. For example, Content Providers cannot easily control access and resource usage policies to satisfy their own requirements. This document discusses the introduction of in-network storage for P2P applications, shows the need for a standard protocol for accessing this storage, and identifies the scope of this protocol. The accessing protocol can also be used by other applications with similar requirements.

Status of this Memo

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 Internet 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 backbone and transit links [I-D.weaver-alto-edge-caches].

P2P caches provide in-network storage and have been deployed in some networks. But 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 to effectively utilize in-network storage. In particular, P2P caches typically do not allow users to explicitly store content into in-network storage. Neither do they allow users to implement control over the content that have been placed into the in-network storage.

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.

With DECADE, P2P applications can still use their native protocols for signaling and data transport. However, they may use a standard protocol for data access incorporating in-network storage, and fall back to their native data transport protocols if in-network storage is not available or not used.

In essence, an open, standard protocol to access in-network storage provides an alternative mechanism for P2P application data access that is decoupled from P2P application control and signaling. This decoupling leads to many advantages, which is explained further in Section 4.

2. Terminology and Concepts

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

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.

IAP (In-network storage Access Protocol): a standard protocol that is spoken between P2P applications and in-network storage. The protocol may also be used between entities implementing the in-network storage service. IAP may be a new protocol or existing protocol with extensions.

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.

Content Publisher: An entity that originates content to consumers.

3. The Problems

The emergence of peer-to-peer (P2P) as a major type of network applications (esp. P2P file sharing and streaming apps) has led to substantial opportunities. The P2P paradigm can be utilized in designing highly scalable and robust applications at low cost, compared with traditional client-server paradigms. For example, CNN [CNN] reported that P2P streaming by Octoshape played a major role in its distribution of the historical inauguration address of President Obama. 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.

However, P2P applications also face substantial design challenges. A particular problem facing P2P applications is the substantial stress that they place on the network infrastructure. Also, lacking of infrastructure support can lead to unstable P2P application performance during peer churns and flash crowd. Below we elaborate on the problems in more 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 (ISP). Multiple measurements (e.g., [ipoque]) have shown that P2P traffic has become a major type of traffic on some networks. Furthermore, network-agnostic peering leads to unnecessary traversal across network domains or spanning the backbone of a network, leading to network inefficiency [I-D.ietf-alto-problem-statement].

Recently, the IETF Application Layer Traffic Optimization (ALTO) Working Group was formed to provide P2P applications with network information so that they can perform better-than-random initial peer selection [I-D.ietf-alto-problem-statement]. 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 pure P2P system. On the other hand, it is reported that P2P traffic is becoming the dominating traffic on the access networks of some networks, reaching as high as 50-60% at the down-links and 60-90% at the uplinks ([DCIA], [ICNP], [ipoque.P2P_survey.], [P2P_file_sharing]). 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 potential (relatively) slow upstream transmission. Similarly, the performance of real-time P2P applications may be affected by potential (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. For example, in [I-D.weaver-alto-edge-caches], the author demonstrates clearly the potential benefits of introducing in-network storage to improve network efficiency and thus reduce network infrastructure stress.

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 be evolving 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, high cost, and low reliability.

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

3.3. Ineffective integration of P2P applications and in-network storage

As P2P applications evolve, it becomes increasingly clear that they will need in-network resources to provide positive user experiences. For example, multiple P2P streaming systems seek additional in-network resources during 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.

A requirement by some P2P applications in using in-network infrastructural resources, however, is 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 their specific algorithms in selecting the peers that behave as the more stable, higher bandwidth sources. They continue to fine-tune such algorithms. In other words, in-network storage should export basic mechanisms and allow as much flexibility as possible to P2P applications to implement specific policies. 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. DECADE as an In-network Storage Capability

The objective of this working group is to design DECADE, an in-network storage protocol to address the problems discussed in the preceding section.

DECADE will provide access to storage and data transport services in the network to P2P applications to improve their efficiency and reduce the stress on the network infrastructure. Unlike the existing P2P caching architecture, DECADE is a standard interface for various P2P applications (both content publishers and end users) to access in-network storage. This decoupling of P2P data access from P2P application control and signaling reduces the complexity of in-network storage services. Furthermore, DECADE provides basic access mechanisms and allows P2P applications to implement flexible policies to create an ecosystem for application innovation and various business goals. Besides that, it also improves the availability of P2P contents because the in-network storage is always-on.

```
                       IAP
                  -----------+
                   |          |
                   |          v
            +---------------------+
            | In-network Storage |
            +---------------------+
              ^                ^
         IAP |            IAP |
   +-------------v-+          +-v--------------+
   | P2P           |          |  Content       |
   | application   |          |  publishers    |
   | clients       |          +----------------+
   +---------------+
    |            ^
    |            |
    +-------------+
    P2P application
    native protocol
```

Figure 1 Overview of protocol interaction between DECADE elements

Specifically, the main component of DECADE is the in-network storage access protocol (IAP), which is a standard, P2P-application-agnostic protocol for different P2P applications to access in-network storage. IAP defines a set of commands that P2P application elements can issue to in-network storage to store and retrieve data. IAP includes the following functionalities:

(1) Data access provides read/write by users (e.g., P2P application clients and content publishers) to the corresponding in-network storage and between entities implementing the in-network storage;

(2) Authorization implements access control to resources provided by in-network storage;

(3) Resource control allows users to implement application policies for the corresponding in-network storage.

Note that DECADE is independent of current IETF work on P2P, e.g. P2PSIP, ALTO, PPSP. For example, peers discovered by either RELOAD or ALTO can all use DECADE to share data.

The Peer to Peer Streaming Protocol effort in the IETF is investigating specification of signaling protocols (called PPSP protocols) for multiple types of 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 document [I-D.zhang-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. A DECADE service can be used for any application transparently to the DECADE 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. So 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. say because no CDN arrangement has been put in place between the Content Provider and the considered 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 considered network provider). In that case, the CDN nodes will participate as in-network PPSP Peers. The off-net PPSP Peers (ie 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. say because an arrangement has been put in place between the Content Provider and the considered network provider). In that case, the CDN nodes will participate as in-network PPSP Peers. The off-net PPSP Peers (ie 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 (using IAP protocol) by PPSP Peers (both off-net end-users and in-network CDN Nodes).

While DECADE will focus on P2P applications, the solution is expected to be applicable in other contexts with similar requirements.

4.1. Data access

P2P application clients use the protocol to read data from an in-network storage, store data to an in-network storage, or remove data from an in-network storage. The data could be of various types. Existing protocols will be used wherever possible and appropriate to support DECADE's requirements. In particular, data storage, retrieval, and management may be provided by an existing IETF protocols. The WG will not limit itself to a single data transport protocol since different protocols may have varying implementation costs and performance tradeoffs. However, to keep interoperability manageable, a small number of specific, targeted, data transport protocols will be identified and used. If a protocol is found to be suitable but does not fully meet the requirements, then the protocol may need to be extended. The following considerations should be taken into account. But it might be a trade-off when making decision.

- o The protocol(s) should support deployments with a very large number of users without substantial increase to operational complexity for the storage provider.

- o The protocol(s) should be easy for applications to integrate with when they want to use it for P2P applications (e.g. file-sharing or streaming) or other content distribution applications.

4.2. Authorization

DECADE ensures that access to the in-network storage is subject to authorization by the user owning the in-network storage service. The authorization can take into account the user trying to access, the content, the time period, etc.

4.3. Resource control

A user uses the protocol to manage the resources on in-network storage that can be used by other peers, e.g., the bandwidth or connections. The resource control policies could be based on individual remote peers or a whole application.

5. Usage Scenarios

Usage scenarios are described from two perspectives. First, we introduce high-level use cases showing how P2P applications may utilize in-network storage. Second, we show how in more detail how users exchange data using IAP.

5.1. BitTorrent

BitTorrent may be integrated with DECADE to be more network efficient and reduce the bandwidth consumed on ISP networks. When a BitTorrent client uploads a block to peers, the block traverses the last-mile uplink once for each peer. With DECADE, 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.

We now describe in more detail how BitTorrent can utilize DECADE. For illustration, we assume that both the BitTorrent client (A) and its peer (B) utilize in-network storage. When A requests a block, it indicates that it would like the block stored in its in-network storage and provides the necessary access control. Instead of sending the 'piece' message with the desired block, peer B replies with a 'redirect' message indicating that the content should be retrieved from its own in-network storage and provides the necessary access control. If the peer B had not previously stored the content in its in-network storage, it uploads the block. A downloads the block into its own in-network storage from B's in-network storage, and finally A itself retrieves the block.

Note that this requires extensions to the BitTorrent protocol. While there are multiple ways to do so, this example assumes the native BitTorrent 'request' message is extended to carry additional information and that a new 'redirect' message is added. Upload and download to/from in-network storage uses the standard IAP protocol.

This example has illustrated how utilizing DECADE can increase BitTorrent's network efficiency. First, notice that peer B does not utilize any uplink bandwidth if the block was already present in its in-network storage. Second, notice that the block is downloaded directly into A's in-network storage. When A wishes to share the block with another peer (say, peer C) that supports DECADE, it may upload directly from its in-network storage, again avoiding usage of the last-mile uplink.

This technique can be applied to other P2P applications as well. Since P2P applications use a standard for communicating with in-network storage, they no longer require in-network storage to explicitly support their protocol. P2P applications retain the ability to explicitly manage which content is placed in in-network storage, as well as access and resource control polices.

5.2. 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.

Consider a case where the Content Publisher owns an in-network storage account within ISP A. If there are multiple P2P peers within ISP A, the Content Publisher may utilize DECADE to distribute content to the peers.

First, the Content Publisher stores a block in the in-network storage, and then sends to each peer in ISP A the block's identifier and necessary access control. Second, each peer may then download from the Content Publisher's in-network storage.

In this example, the block is distributed in a more network efficient way (the content only traverses the ISP's interdomain link once), while the Content Publisher retains explicit control over access to the content placed in its own storage. The Content Publisher can remove content from its in-network storage when it is stale or needs to be replaced, and grant access and resources to only the desired peers.

Note that Content Publishers and individual peers can each use in-network storage. For example, after downloading content from the Content Publisher's in-network storage, peers may each utilize their own in-networks storage similar to the usage scenario in Section 5.1. This can have the benefit of increased network efficiency, while Content Publishers and peers still retain control over content placed in their own in-network storage.

5.3. Data Transfer Scenarios

The previous usage scenarios have utilized the ability for peers to transfer data by storing and retrieving from in-network storage. This section describes in further detail an example solution of how DECADE can provide this capability.

In this section, we consider the case of a user B (the receiver) requesting data from user A (the sender). We use Sa to denote User A's storage account, and Sb to denote User B's storage account. Each user independently decides if its in-network storage account is used, so there are four cases.

When a user indicates that it wishes to use its in-network storage, it provides an access control token the other user. The token is sent using the application's protocol. To simplify the illustration, we omit details of the access control from the detailed scenarios below.

5.3.1. Both Sender and Receiver Accounts are Used

This scenario is illustrated in Figure 2. B first requests an object from A using the application protocol indicating it wishes the object to be stored in Sb. A responds using the application protocol indicating that B should download the object from Sa. B sends a IAP request to Sb indicating that the object should be downloaded from Sa. Sb uses IAP to download from Sa, and finally, B downloads the object from Sb (also using IAP).

```
      +-------+ 4  IAP Get +-------+
      |  Sa   | <----------+  Sb   |
      +-------+  *********>+-------+
            5 data transfer  ^  *
                    3 IAP Get \  * 6 data transfer
             1 App request     \ \ /
   +-------+<------------------+-------+
   |User A |                   |User B |
   +-------+------------------>+-------+
             2 App response
```

Figure 2: Usage Scenario 1 (Sender and receiver Accounts used)

5.3.2. Only Sender’s Storage Account is Used

This scenario is illustrated in Figure 3. B requests an object from A using the application protocol. A responds using the application protocol indicating that B should download the object from Sa. Finally, B sends a IAP request to Sa to download the object.

```
                              +-------+
                              |  Sa   |
                              +-------+
                                ^  *
                       3 IAP Get  \  * 4 data transfer
                 1 App request     \ \/
       +-------+<-------------------+-------+
       |User A |                    |User B |
       +-------+------------------->+-------+
                 2 App response
```

Firgure 3: Usage Scenario 2 (Sender account used)

5.3.3. Only Receiver’s Storage Account is Used

This scenario is illustrated in Figure 4. B requests an object from A using the application protocol indicating that it wishes the object to be stored in Sb. A stores the object in Sb (using IAP), and responds to B (using the application protocol) that it should download from Sb. B uses IAP to download the object from Sb.

```
                        +---------+
                      > |   Sb    |
            3.       /  +---------+
         IAP Store /            ^  *
                 /    3.IAP Get   \  * 4 data transfer
               / 1.App request     \  \/
       +-------+<-------------------+-------+
       |User A |                    |User B |
       +-------+------------------->+-------+
                 2.App response
```

Figure 4: Usage Scenario 3 (Receiver account used)

5.3.4. No Storage Accounts are Used

This scenario is illustrated in Figure 5. In this scenario, the application protocol is used directly to send data. This scenario applies with one of the peers does not support IAP, or neither of the peers are using in-network storage.

```
              1.App request
 +-------+<------------------+-------+
 |User A |   ...   ...   ... |User B |
 +-------+******************>+-------+
              2.data transfer
```

Figure 5: Usage Scenario 4 (No storage accounts used)

6. Security Considerations

There are multiple security considerations. We focus on two in this section.

6.1. Denial of Service attack

Without access control or resource control, an attacker can try to consume a large portion of the in-network storage, or exhaust the connections of the in-network storage to commit a Denial of Service (DoS) attack. Thus, access control and resource control mechanisms are mandatory. A resource control mechanism should be used to allow a user to allocate the resource in its in-network storage account to be utilized by other clients.

6.2. Copyright and Legal issues

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

7. IANA Considerations

There is no IANA consideration with this document.

8. Acknowledgments

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

David Bryan

Francois Le Faucheur

Hongqiang Law.

Kar Ann Chew

Richard Woundy

Roni Even

Yunfei Zhang

Yu-shun Wang

Yingjie Gu

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

9.2. Informative References

[ipoque.com] "http://www.ipoque.com/resources/internet-studies/internet-study-2008_2009".

[I-D.ietf-alto-problem-statement] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", draft-ietf-alto-problem-statement-04 (work in progress), September 2009.

[I-D.weaver-alto-edge-caches] Weaver, N., "Peer to Peer Localization Services and Edge Caches", draft-weaver-alto-edge-caches-00 (work in progress), March 2009.

[I-D.zhang-ppsp-problem-statement] Zhang, Y., Zong, N., Camarillo, G., Seng, J., and Y. Yang, "Problem Statement of P2P Streaming Protocol (PPSP)", draft-zhang-ppsp-problem-statement-06 (work in progress), July 2010.

[RFC3720] Satran, J., Meth, K., Sapuntzakis, C., Chadalapaka, M., and E. Zeidner, "Internet Small Computer Systems Interface (iSCSI)", RFC 3720, April 2004.

[RFC5661] Shepler, S., Eisler, M., and D. Noveck, "Network File System (NFS) Version 4 Minor Version 1 Protocol", RFC 5661, January 2010.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[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.

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

Authors' Addresses

Haibin Song
Huawei
Baixia Road No. 91
Nanjing, Jiangsu Province 210001
P.R.China

Phone: +86-25-56622975
Email: melodysong@huawei.com

Ning Zong
Huawei
Baixia Road No. 91
Nanjing, Jiangsu Province 210001
P.R.China

Phone: +86-25-56622975
Email: zongning@huawei.com

Y. Richard Yang
Yale University

Email: yry@cs.yale.edu

Richard Alimi
Yale University

Email: richard.alimi@yale.edu