ALTO
Internet-Draft
Intended status: BCP
Expires: January 13, 2011

Y. Gu
Huawei
R. Alimi
Yale University
R. Even
Huawei

July 12, 2010

ALTO Information Redistribution
draft-gu-alto-redistribution-03

Abstract

The ALTO protocol proposes several mechanisms to increase scalability. One of the proposed mechanisms is ALTO information redistribution. This document concretely defines ALTO Information Redistribution, indicates suggested extensions to the ALTO Protocol to support redistribution, and shows how redistribution could be used in practice.

Table of Contents

1. Introduction

When providing an ALTO service, Network Providers offer information to clients with the goal of helping P2P applications to perform better peer selection and improving network efficiency. The ALTO Service becomes a distribution point of network information for ALTO Clients within its network. A Network Provider may deploy an ALTO Service using techniques such as load balancing and caching to handle a large number of subscribers. In this document, we discuss an additional mechanism to distribute ALTO information to ALTO Clients: ALTO Information Redistribution. Consider a scenario where a large number (e.g., millions) of users start their P2P live streaming clients just before the start of a popular event. Each client requests ALTO information directly from the ALTO Service within their ISP if it has not yet been retrieved or needs to be refreshed. For certain content (e.g., content with broad interest), even the number of streaming clients within a single ISP may be extremely large. For example, tens of millions of people watched the United States Presidential Inauguration in Jan. 2009 via the Internet through such sites as CNN. During the 2009 Spring Festival Evening in China, an audience of about 24 million watched the program on the Internet. In such a case, an ISP's ALTO Service may not be able to handle the load and provide ALTO information directly to each client.

Many mechanisms, such as load balancing, can be used to increase scalability of an ALTO Service. [I-D.penno-alto-protocol] proposes ALTO Information Redistribution as one technique. Using ALTO Information Redistribution, ALTO Clients may distribute ALTO information to each other instead of requesting directly from the ALTO Server. For example, a P2P infrastructure can be used to distribute ALTO information to a large number of ALTO Clients with small load on the ALTO Server.

This document serves three primary purposes:

1) Defines basic requirements for redistributing ALTO information, and considerations for developing a redistribution scheme.

2) Propose extensions to the ALTO Protocol to support ALTO Information redistribution to meet the defined requirements.

3) Provide use cases showing how redistribution may be applied in practice.

We envision that multiple redistribution schemes are possible, and the design may depend on the particular setting, such as scalability requirements and existing application protocols. Thus, standardization of a redistribution scheme for all kinds of scenario

is not an object. This BCP intends to extract the fundamental for real-world practice, and to provide use cases for most common scenario of ALTO Redistribution.

Note that certain design changes during the development of the ALTO Protocol may affect ALTO information redistribution. This document will be updated to track the progress of the ALTO Protocol.

2. Terminologies 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 document uses terms defined in [I-D.ietf-alto-problem-statement]and [I-D.penno-alto-protocol].

3. Redistribution Framework

Before defining requirements for ALTO Information Redistribution, we first give a concrete model for ALTO Information Redistribution that we use in this document:

1. A set of ALTO Servers {S1, S2, S3, ...} are deployed, each possibly serving a different network region.

2. A set of ALTO Clients are running. We assume that each ALTO Client can be mapped to one of the available ALTO Servers by the ALTO discovery process [I-D.song-alto-server-discovery]. Let Clients(Si) denote the set of ALTO Clients mapped to ALTO Server Si.

3. An ALTO Client wishes to obtain a particular set of information from the ALTO Server. The desired information is fully specified by: (1) the ALTO Server (hostname and port), and (2) the query and input parameters that would be sent to the ALTO Server. See Section 6 for a discussion of behavior when criteria other than the input parameters are used by an ALTO Server to generate a response.

4. An ALTO Client may obtain the information either directly from its ALTO Server Si, or it may obtain the information from a member of Clients(Si).

5. For a particular query to ALTO Server Si, at least one member of Clients(Si) directly fetches from Si. The remaining members of

Clients(Si) obtain the information from some other member in Clients(Si).

3.1. Basic Requirements

A redistribution scheme should satisfy some basic requirements in order to successfully redistribute an ALTO Server's response to a set of ALTO Client.

1. The ALTO Client should be able to identify the desired redistributed data based only on the ALTO Server hostname and port, and the input parameters.

2. The ALTO Client should be able to check the validity of the information once it is retrieved. That is, the ALTO Client should be able to determine if the retrieved information was indeed generated by its ALTO Server, , and is generated based on the particular input parameters.

3.2. ALTO Information Types

In general, it should be possible to redistribute the response from a particular ALTO Server that does not depend on anything except query input parameters. However, redistribution may only be worthwhile for queries that are made by a large number of ALTO Clients. In the context of [I-D.penno-alto-protocol], we provide an example list of information types that may be commonly used across many ALTO Clients, and hence benefit from redistribution. The example list the most obvious examples to our best knowledge, and there might be other information types that are suitable for redistribution.

- Server Capability which indicates the capabilities implemented by an ALTO Server.

- Full Network Map which lists the PIDs and IP prefixes that are contained within each PID.

- Full Path Cost Map among all PIDs, which indicates the Path Cost between each pair of PIDs.

[I-D.penno-alto-protocol]also specifies certain queries that may not benefit from redistribution. For example, if a peer requests ordinal Path Costs (i.e., a ranked list) for a set of individual endpoint addresses (i.e., Resource Providers), the response may not be useful to other ALTO Clients. This is because other ALTO Clients may be running different applications or have a different set of available peers (e.g., participating in a different swarm, or be in contact with a different set of peers within the same swarm).

3.3. Redistribution Scheme Design

While this document does not fully specify a particular redistribution scheme, we provide a sampling of decisions that should be considered when designing an implementing a redistribution scheme. This list can be used as a guide for implementers when designing a redistribution scheme for a particular setting. Considerations for a redistribution scheme include:

- Who places the ALTO Information onto overlay, and how the ALTO Information is published on the overlay
- How could peers decide whether to get ALTO Information from ALTO Server or to get from Overlay?
- What method is used for peers to locate redistributed ALTO information?
- What protocol is used for retrieving redistributed ALTO information?
- How could peers verify the Redistributed Information it obtained.
- How to update the Redistributed Information on time and who is responsible for updating.
- What naming scheme is used to specify the ALTO Server hostname, port, and input parameters in the protocol for locating redistributed ALTO information? For example, the naming scheme could define how to compute the 'key' in a DHT.
- How is the redistributed information encoded? Note that the original response from the ALTO Service may be reformatted (e.g., compressed) for redistribution. Note that if this approach is taken and ALTO Clients still wish to verify received information, ALTO Clients should be able to reconstruct the ALTO Service's original response (e.g., via decompression). If a lossy transformation is used (e.g., filtering), ALTO Clients may not be able to verify the received information.

4. ALTO Redistribution Solutions Analysis

In this section, we present our considerations on Redistribution Implementation. In the previous section, several questions have been enumerated, and the answer to the first question is the baseline of ALTO Redistribution. There are two distinct ways to place the ALTO Information into the overlay: Redistribution Proxy PUSH the

Information into the overlay or components of the overlay PULL the Information into the Overlay.

The main difference betwen PUSH and PULL is as follows.

- In PUSH, the Redistribution Proxy can publish the updated redistribution Information into the overlay. Since the Redistribution Proxy can be much intelligent than a normal peer, e.g. Redistribution Proxy has overall understanding of the overlay, it can make decision on how to distribute the Information in order to utilize the overlay much efficient. For example, integration with DECADE, distribution via a CDN (many P2P apps are integrating CDNs now). We will add more use case for PUSH in our later version.

- While In PULL, peers individually pull/request updated ALTO information from an overlay or ALTO server. Considering the general size of a redistributable information, there might be an out-of-control flooding through the overlay.

In the following text, we introduces some consideration of both PUSH and PULL, and conclude other features of PUSH and PULL.

4.1. PUSH Information into Overlay

In PUSH method, we introduce a new terminology, Redistribution Proxy.

A Redistribution Proxy can be a Tracker in Tracker-based P2P Overlay or a supernode either in Tracker-based or Trackerless P2P Overlay. The duty of Redistribution Proxy to accecpt the redistributable information from ALTO Server and push the information into the overlay. There are two options on obtaining redistributable inforamtion from ALTO Server. One is Redistribution Proxy request and ALTO Server answers or ALTO Server positively update the information to the Redistribution Proxy. Since the number of Redistribution Proxy is much fewer than the total number of ALTO Clients, so the latter option does not occupy much connection resources on ALTO Server. In our draft, we introduce the PUSH based on the latter option, which is ALTO Server pushes the uptaed information to Redistribution Proxy positivly.

4.1.1. General Requirements for PUSH

To make PUSH feasible, the following requirements must be satisfied.

- Redistribution Proxy MUST has public IP Addresse, so that both ALTO server and Peers can connect with it without NAT issue.

- ALTO server MUST know Redistribution proxy's IP Address.

- Redistribution Proxy MUST be reliable.

4.1.2. Tracker Acts as Redistribution Proxy

If Tracker is deployed as Redistribution Proxy, it has two choices.

Tracker can publish the ALTO Information into overlay, then peers can search and obtain the Information through Overlay.

Or Tracker can store the Information on itself, and answer peers' particular request based on the Information. For those information that is not redistributable, peers can request directly from ALTO server or ask Tracker to request from ALTO server and answer.

In both case, an indication is necessary for telling peers where to get ALTO Service. For example, peers should know that for those redistributable type of Information, peers should first request from Tracker, and for those unredistributable, they should request from ALTO Sever. Which can be configured by application. The definition of indication configuration is out of the scope of ALTO.

4.1.3. Supernode Acts as Redistribution Proxy

If Supernode is deployed as Redistribution Proxy, it has two choices as well.

Supernode store the Information and answer peers request. In this case, peers should learn Supernode's IP Address in advance.

Or Supernode just publish the Information into the overlay and then peers search and get the Information through the overlay.

In both case, an indication is necessary as well.

4.1.4. Advantages of PUSH

The advantages of PUSH is obvious.

1 Intelligence. Refer to the text that explain the main difference between PUSH and PULL.

2 Scalability. PUSH can significantly reduce the request directly sent to ALTO Server, which can reduce the load on ALTO Server.

3 Quick updating. Once the ALTO Information is updated in Server, ALTO server can notify the Redistribution Proxy with new Information.

4 Good Consistency, because redistributed Informaiton can be updated in time.

5 Never expiration, because the redistributed Information is always as new as those in the server.

6 Safe. Redistribution Proxy is reliable.

7 Low latency, because peers do not need to search and locate the redistribution Information on the overlay, which is extremely meaningful for time sensitive application, e.g. Live streaming.

4.2. PULL Inforamtion into Overlay

There could be Tracker or peers who PULL the ALTO Information into the overlay.

If it's peer that pull ALTO information into the overlay, several aspects should be considered.

1 Where to request for ALTO Information first, ALTO server or overlay? This can be configured by application, e.g. always request redistributable type of ALTO Information through Overlay first. If appliction configuration recommends peer to request from ALTO Server first, this won't be any good to ALTO Redistribution.

2 Publishing mechanism. Should peer store the Information on itself and publish the resource into the overlay, or should peer store the Information on the corresponding responsible peer on the overlay? However, this won't make much difference.

3 How to authenticate the Information? We have introduce a security method with Signature, which is now in ALTP Protocol, while the question is where to get the Public Key.

4.2.1. PULL Use Cases

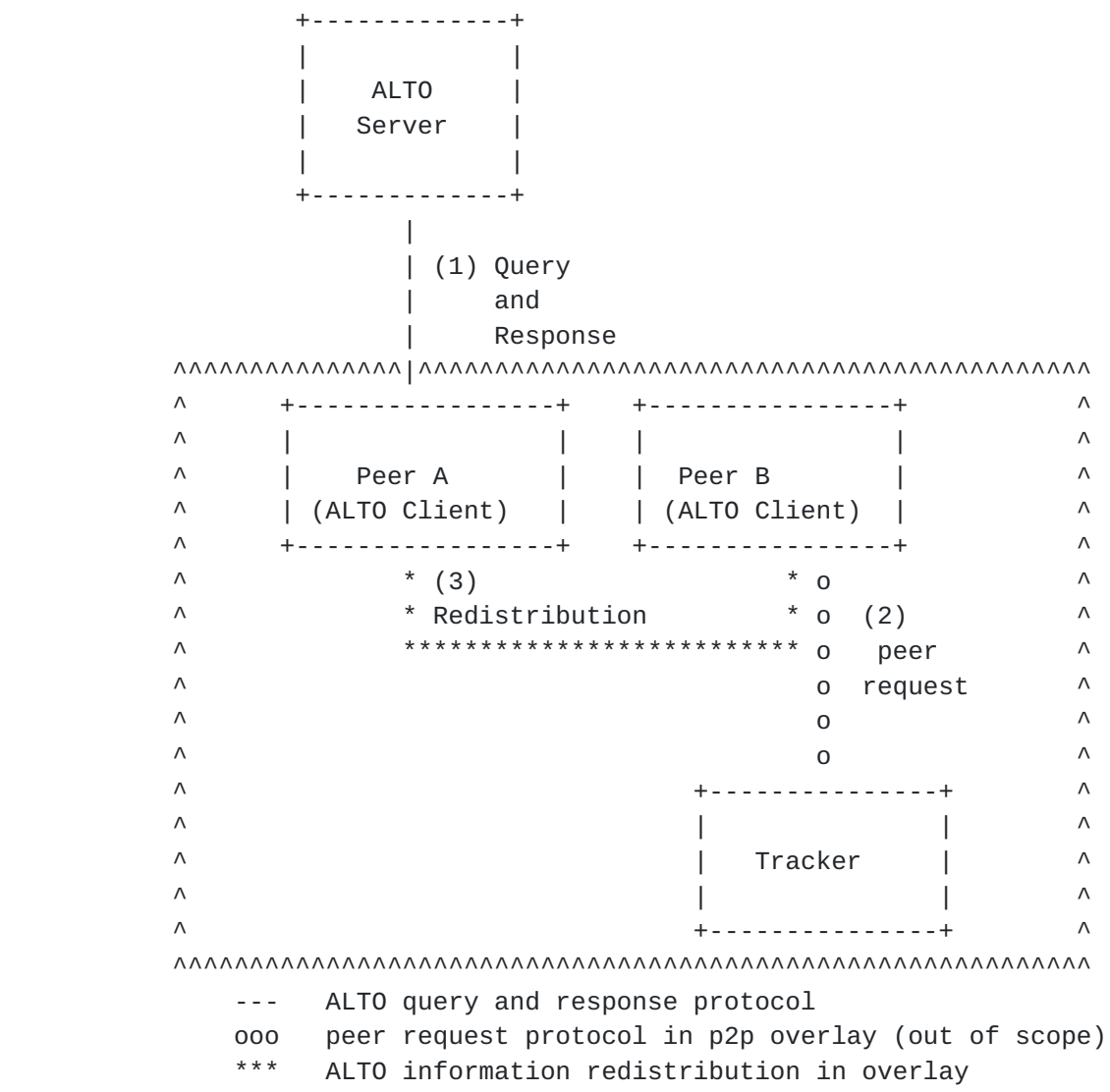
The architecture of a particular P2P application will affect the redistribution mechanism. Generally speaking, there are two kinds of P2P applications: trackerless, and those using a tracker. In Tracker-based applications, a resource directory is maintained on a tracker, and peers contact the tracker to learn about new peers. In a Trackerless overlay, peers are organized through a particular

algorithm, e.g. DHT, and they publish or find resources by routing through the overlay.

4.2.1.1. Tracker-based redistribution

1. The Tracker finds the ALTO server on behalf of a peer, queries the necessary ALTO information and replies to the peer with the ALTO information as well as the candidate list. [I-D.kiesel-alto-3pdisc] describes several methods by which Tracker can find the right ALTO server. Note that the Tracker might omit the 'more preferred' peers when making the original selection. However, the ALTO Information can be applied to peers learned from other sources (e.g., Peer Exchange and/or DHT).

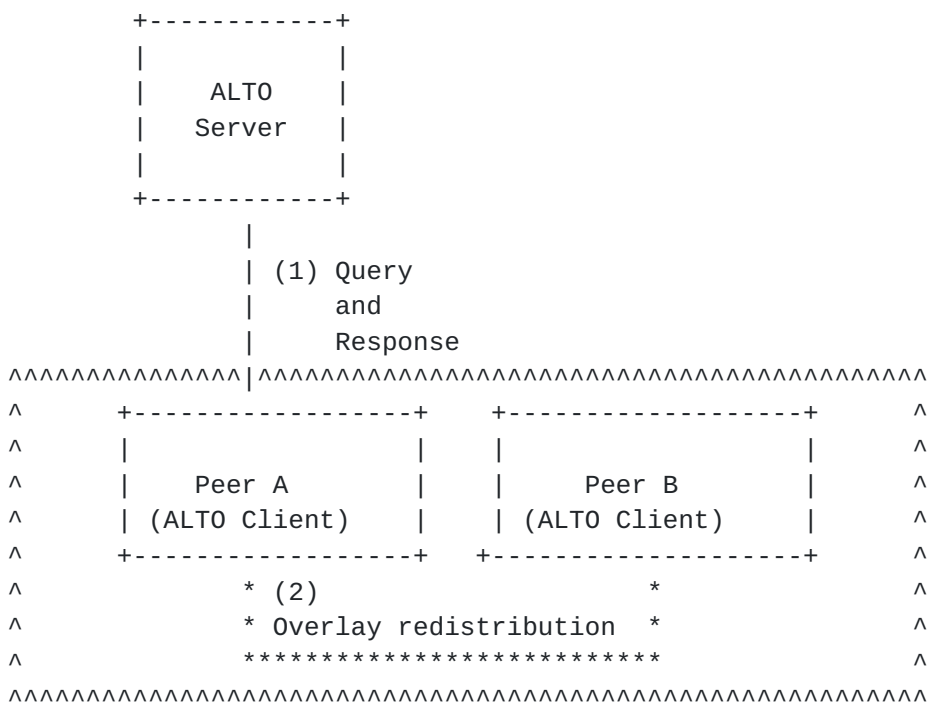
2. A peer asks for a Resource and Tracker replies without any ALTO information. The peer queries the ALTO server for ALTO information, and selects peers. In order to help lessen the burden on ALTO server, as well as to help other peers who want the same ALTO information, the peer publishes the ALTO information on the Tracker (if the Tracker allows this behavior). Peers may then distribute the ALTO information just as any other Resource. The method introduced here can be regard as a complementary process to (1).



Information redistribution among peers in Tracker-based P2P Application

4.2.1.2. Trackerless redistribution

In a Trackerless overlay, peers obtain the ALTO information, then publish it via a P2P protocol (e.g., in a DHT). Peers can also locate and retrieve ALTO information through the protocol.



Information redistribution among peers in Trackerless P2P Overlay

4.2.1.2.1. Lookup in DHT

When searching for a piece of data in a DHT, a node constructs an identifier for the desired data. We propose here a simple naming scheme that can be used to lookup ALTO Information in a DHT. This is provided as a suggestion for an implementation technique, and is not a requirement on redistribution implementations employing a DHT.

Also note that the ALTO Information does not need to be included directly in the DHT. A mechanism such as the Distributed Tracker implemented in Vuze [http://www.vuze.com] could be used to locate an ALTO Client that in turn provides access to the ALTO Information.

This naming scheme allows redistribution of ALTO information requested using HTTP GET requests in [I-D.penno-alto-protocol] Since REST-style URLs are used, input parameters are included directly in the URL (along with ALTO Server hostname and port). Thus, we compute a hash of the form:

```
hash("alto:REQUEST_URL")
```

hash("alto:REQUEST_URL")

where REQUEST_URL is the full HTTP URL that would have been used to request ALTO information from the ALTO Server directly. The resulting string can be used as the lookup key in the DHT.

The following simple examples show how the scheme applies to the Information Types in Section 3.2:

1) Server Capability

hash("alto:http://alto.example.com:80/capability").

2) Full Network Map.

hash("alto:http://alto.example.com:80/prop/pid/map").

3) Full Path Cost among all PIDs

hash("alto:http://alto.example.com:80/cost/pid/map").

5. Acknowledgments

The authors would like to give special thanks to Jan Seedorf and many others for the illuminative discussion in the mailing list. The authors also thank David Bryan for providing comments on preliminary versions of the draft.

6. References

6.1. Normative References

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

[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.

6.2. Informative References

[I-D.penno-alto-protocol] Penno, R. and Y. Yang, "ALTO Protocol", draft-penno-alto-protocol-04 (work in progress),

October 2009.

[I-D.kiesel-alto-3pdisc]
Kiesel, S., Tomsu, M., Schwan, N., Scharf, M., and M. Stiemerling, "Third-party ALTO server discovery", draft-kiesel-alto-3pdisc-03 (work in progress), July 2010.

[I-D.song-alto-server-discovery]
Yongchao, S., Tomsu, M., Garcia, G., Wang, Y., and V. Avila, "ALTO Service Discovery", draft-song-alto-server-discovery-03 (work in progress), July 2010.

Authors' Addresses

Gu Yingjie
Huawei
Baixia Road No. 91
Nanjing, Jiangsu Province 210001
P.R.China

Phone: +86-25-84565868
Fax: +86-25-84565888
Email: guyingjie@huawei.com

Richard Alimi
Yale University

Email: richard.alimi@yale.edu

Roni Even
Huawei

Email: ron.even.tlv@gmail.com