Network Working Group
Internet Draft
Intended status: Informational

Greg Bernstein (Grotto)
Young Lee (Huawei)

June 28, 2011

Use Cases for High Bandwidth Query and Control of Core Networks

draft-bernstein-alto-large-bandwidth-cases-00.txt

Status of this Memo

Copyright Notice

Abstract

This draft describes two generic use-cases that illustrate application layer traffic optimization concepts applied to high bandwidth core networks. For the purposes here high bandwidth will mean bandwidth that is significant with respect to the capacity of a wavelength in a wavelength division multiplexed optical transport system, e.g., 10-40Gbps or more. For each of these generic use cases, we present a generic optimization problem, look at the type of information needed (query interface) to perform the optimization, investigate a reservation interface to request network resources, and also consider enhanced availability and recovery scenarios.

Table of Contents

1. Introduction

Cloud Computing, network applications, software as a service (SaaS), Platform as a service (PaaS), and Infrastructure as a Service (IaaS), are just a few of the terms used to describe situations where multiple computation entities interact with one another across a network. When the communication resources consumed by these interacting entities is significant compared with link or network

capacity then opportunities may exist for more efficient utilization of available computation and network resources if both computation and network stratums cooperate in some way. The application layer traffic optimization (ALTO) working group is tackling the similar problem of "better-than-random peer selection" for distributed applications based on peer to peer (P2P) or client server architectures [16]. In addition, such optimization is important in content distribution networks (CDNs) as illustrated in [17].

General multi-protocol label switching (GMPLS) [18] can and is being applied to various core networking technologies such as SONET/SDH [19] and wavelength division multiplexing (WDM) [20]. GMPLS provides dynamic network topology and resource information, and the capability to dynamically allocation resources (provision label switched paths). Furthermore, the path computation element (PCE) [21] provides for traffic engineered path optimization.

However, neither GMPLS nor PCE provide interfaces that are appropriate for an application layer entity to use for the following reasons:

- GMPLS routing exposes full network topology information which tends to be proprietary to a carrier or require specialized knowledge and techniques to make use of, e.g., the routing and wavelength assignment (RWA) problem in WDM networks [20].

- Core networks typically consist of two or more layers, while applications are typically only know about the IP layer and above. Hence applications would not be able to make direct use of PCE capabilities.

- GMPLS signaling interfaces are defined for either peer GMPLS nodes or via a user network interface (UNI) [22]. Neither of these is appropriate for direct use by an application entity.

In this paper we discuss two general use-cases that can generate core network flows with significant bandwidth and may vary significantly over time. The "cross stratum optimization" problems generated by these use cases are discussed. Finally, we look at interfaces between the application and network "stratums" that can enable overall optimization.

1.1. Computing Clouds, Data Centers, and End Systems

While the definition of cloud computing or compute clouds is somewhat nebulous (or "foggy" if you will) [1], the physical instantiation of compute resources with network connectivity is very real and bounded by physical and logical constraints. For the purposes of this paper

we will call any network connected compute resources a data center if its network connectivity is significant compared either to the bandwidth of an individual WDM wavelength or with respect to the network links in which it is located. Hence we include in our definition very large data centers that feature multiple fiber access and consume more than 10MW of power [2], moderate to large content distribution network (CDN) installations located in or near major internet exchange points [3], medium sized business centers, etc...

We will refer to those computational entities that don't meet our bandwidth criteria for a data center as an "end system".

2. End System Aggregate Networking

In this section we consider the fundamental use case of end systems communicating with data centers as shown in Figure 1. In this figure the "clients" are end systems with relatively small access bandwidth compared to a WDM wavelength, e.g., under 100Mbps. We show these clients roughly partitioned into three network related regions ("A", "B", and "C"). Given a particular network application, in a static network application situation, each client in a region would be associated with a particular data center.

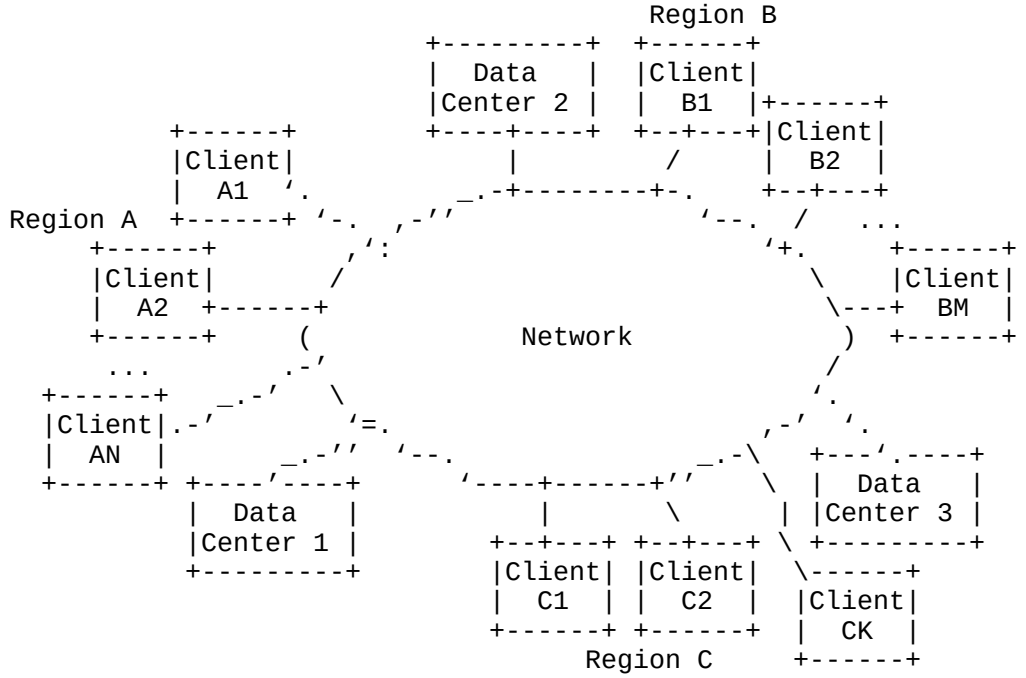


Figure 1. End system to data center communications.

2.1. Aggregated Bandwidth Scaling

One of the simplest examples where the aggregation of end system bandwidth can quickly become significant to the "network" is for video on demand (VoD) streaming services. Unlike a live streaming service where IP or lower layer multicast techniques can be generally applied, in VoD the transmissions are unique between the data center and clients. For regular quality VoD we'll use an estimate of 1.5Mbps per stream (assuming H.264 coding), for HD VoD we'll use an estimate of 10Mbps per stream. To fill up a 10Gbps capacity optical wavelength requires either 6,666 or 1,000 clients for regular or high definition respectively. Note that special multicasting techniques such as those discussed in [4] and peer assistance techniques such as provided in some commercial systems [5] can reduce the overall network bandwidth requirements.

With current high speed internet deployment such numbers of clients are easily achieved; in addition demand for VoD services can vary significantly over time, e.g., new video releases, inclement weather (increases number of viewers), etc...

2.2. Cross Stratum Optimization Example

In an ideal world both data centers and networks would have unlimited capacity, however in actuality both can have constraints and possibly varying marginal costs that vary with load or time of day. For example suppose that in Figure 1 that Data Center 3 has been primarily serving VoD to region "C" but that it has, at a particular period in time, run out of computation capacity to serve all the client requests coming from region "C". At this point we have a fundamental cross stratum optimization (CSO) problem. We want to see if we can accommodate additional client request from region "C" by using a different data center than the fully utilized data center #3. To answer this questions we need to know (a) available capacity on other data centers to meet a request, (b) the marginal (incremental) cost of servicing the request on a particular data center with spare capacity, (c) the ability of the network to provide bandwidth between region "C" to a data center, and (d) the incremental cost of bandwidth from region "C" to a data center.

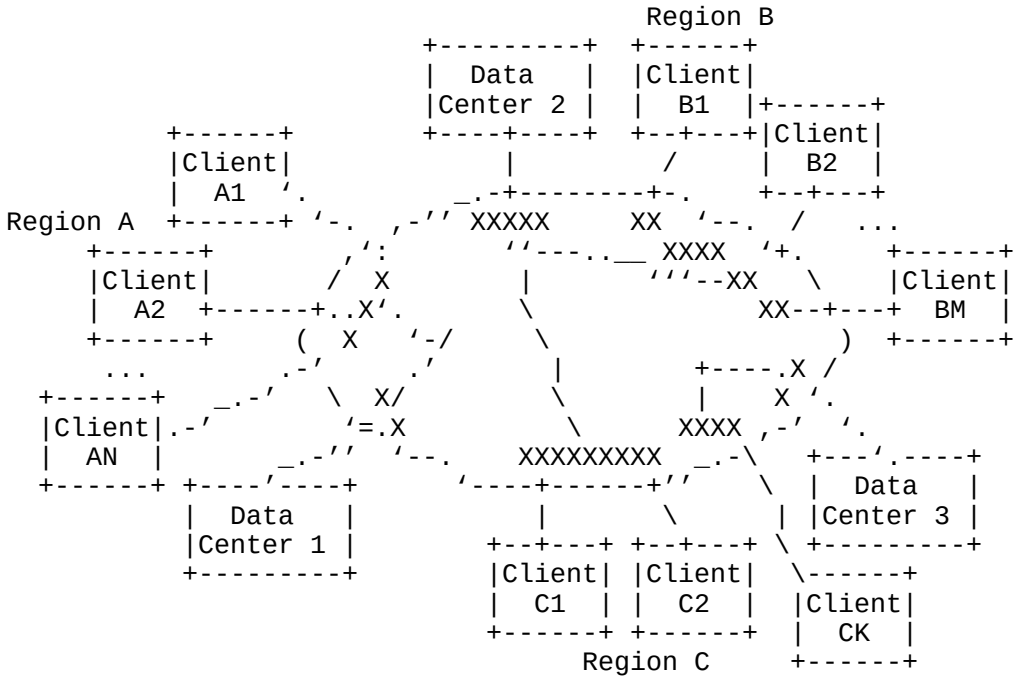


Figure 2. Aggregated flows between end systems and data centers.

In Figure 2 we show a possible result of solving the previously mentioned CSO problem. Here we show the additional client requests from region "C" being serviced by data center #2 across the network. Figure 2 also illustrates the possibility of setting up "express" routes across the network at the MPLS level or below. Such techniques, known as "optical grooming" or "optical bypass" [6], [7] at the optical layer, can result in significant equipment and power savings for the network by "bypassing" higher level routers and switches.

2.3. Data Center and Network Faults and Recovery

Data center failures, whether partial or complete, can have a major impact on revenues in the VoD example previously described. If there is excess capacity in other data centers within the network associated with the same application then clients could be redirected to those other centers if the network has the capacity. Moreover, MPLS and GMPLS controlled networks have the ability to reroute traffic very quickly while preserving QoS. As with general network recovery techniques [8] various combinations of pre-planning and "on

the fly" approaches can be used to tradeoff between recovery time and excess network capacity needed for recovery.

In the case of network failures there is the potential for clients to be redirected to other data centers to avoid failed or over utilized links.

2.4. Cross Stratum Control Interfaces

Two types of load balancing techniques are currently utilized in cloud computing. The first is load balancing within a data center and is sometimes referred to as local load balancing. Here one is concerned with distributing requests to appropriate machines (or virtual machines) in a pool based on the current machine utilization. The second type of load balancing is known as global load balancing and is used to assign clients to a particular data center out of a choice of more than one within the network and is our concern here. A number of commercial vendors offer both local and global load balancing products (F5, Brocade, Coyote Point Systems). Currently global load balancing systems have very little knowledge of the underlying network. To make better assignments of clients to data centers many of these systems use geographic information based on IP addresses [9]. Hence we see that current systems are attempting to perform cross stratum optimization albeit with very coarse network information. A more elaborate interface for CSO in the client aggregation case would be:

1. A Network Query Interface - Where the global load balancer can inquire as to the bandwidth availability between "client regions" and data centers.

2. A Network Resource Reservation Interface - Where the global load balancer can make explicit requests for bandwidth between client regions and data centers.

3. A Fault Recovery Interface - For the global load balancer to make requests for expedited bulk rerouting of client traffic from one data center to another.

The network query interface can be considered a superset of the functionality proposed from the ALTO (application layer traffic optimization) servers being standardized in [10]. Note that in the network query and reservation interfaces it would be worthwhile to consider both current resources and resources at a future time, i.e., scheduled resources. Although scheduled reservations are not supported directly by technologies such as MPLS and GMPLS they can be considered in network planning and provisioning systems. For example, a VoD provider knows ahead of time when the latest "blockbuster" film

will be available via its service and can make estimates based on historical data on the bandwidth that it will need to deal with the subsequent demand.

3. Data Center to Data Center Networking

There are a number of motivations for data center to data center communications: on demand capacity expansion ("cloud bursting") [11], cooperative exchanges between business partners, offsite data backup, "rent before building"[12], etc... In Figure 3 we show an example where a number of businesses each with an "internal data center" contracts with a large external data center for additional computational (which may include storage) capacity. The data centers may connect to each other via IP transit type services or more typically via some type of Ethernet virtual private line or LAN service.

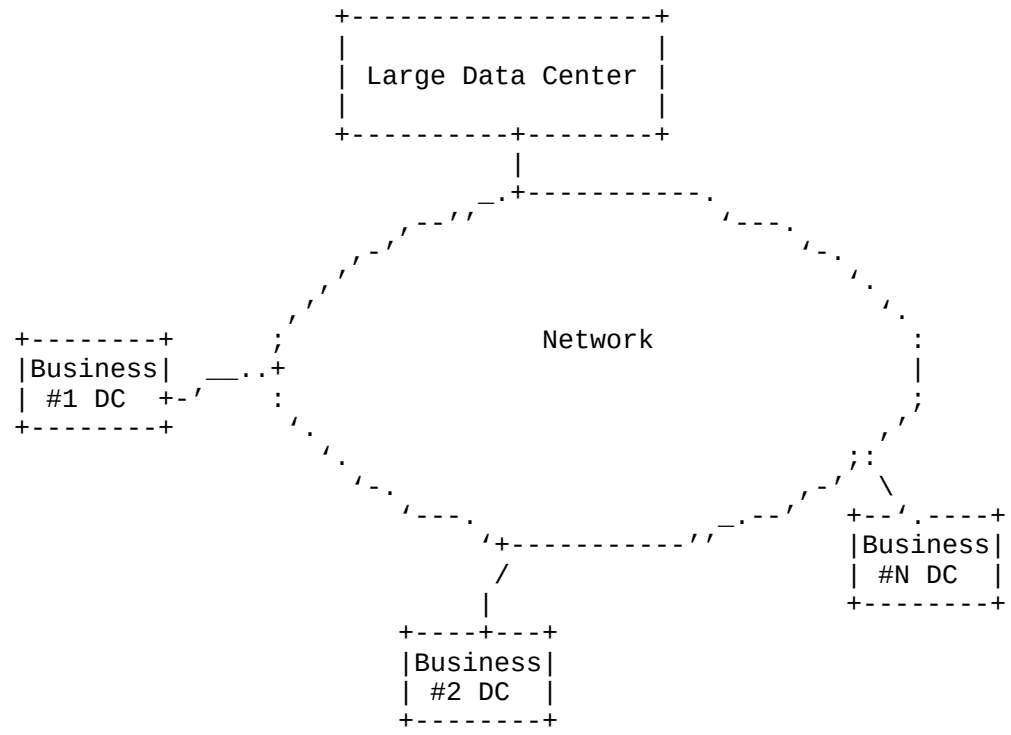


Figure 3. Basic data center to data center networking.

3.1. Cross Stratum Optimization Examples

In the DC-to-DC example of Figure 3 we can have computational constraints/limits at both local and remote data centers; fixed and marginal computational costs at local and remote data centers; and network bandwidth costs and constraints between data centers. Note that computing costs could vary by the time of day along with the cost of power and demand. Some cloud providers such as Amazon [13] have quite sophisticated compute pricing models including: reserved, on demand, and spot (auction) variants.

In addition, to possibly dynamically changing pricing, traffic loads between data centers can be quite dynamic. In addition, data movement between data centers is another source of large network usage variation. Such peaks can be due to scheduled daily or weekly offsite data backup, bulk VM migration to a new data center, periodic virtual machine migration [14], etc...

3.2. Network and Data Center Faults and Reliability

For networked applications that require high levels of reliability/availability the network diagram of Figure 4 could be enhanced with redundant business locations and external data centers as shown in Figure 4. For example cell phone subscriber databases and financial transactions generally require what is called geographic database replication [15] and results in extra communication between sites supporting high availability. For example if business #1 in Figure 4 required a highly available database related service then there would be an additional communication flows from the data center "1a" to data center "1b". Furthermore, if business #1 has outsourced some of its computation and storage needs to independent data center X then for resilience it may want/need to replicate (hot-hot redundancy) this information at independent data center Y.

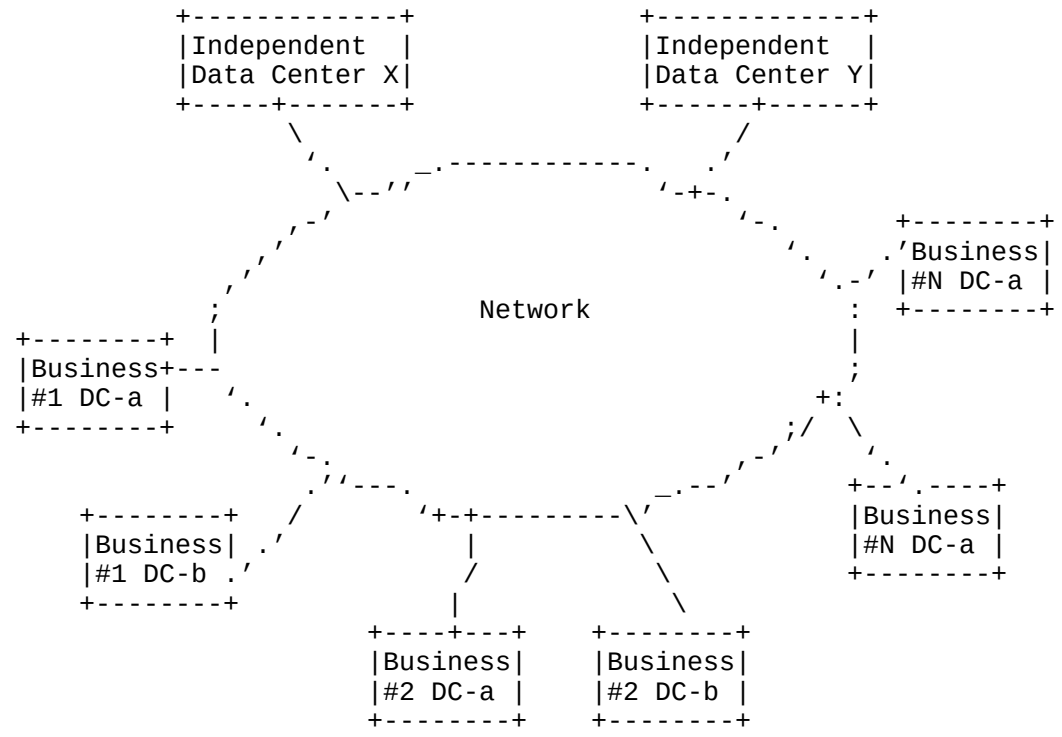


Figure 4. Data center to data center networking with redundancy.

3.3. Cross Stratum Control Interfaces

Similar to the end system aggregation case we can decompose cross stratum interfaces into three general types: (a) network query, (b) network reservation, and (c) recovery. However for DC-to-DC interfaces we are interested in network resources between data centers rather than between "client regions" and data centers.

For network resource queries we may be concerned with (a) current bandwidth availability, (b) bandwidth availability at a future time, or (c) bandwidth for a bulk data transfer of a given amount that must take place within a given time window. A network reservation interface with both current and advanced reservation capability would complement the query interface.

A simple recovery interface for data center based faults could be based on unused backup paths between data centers that are reserved

but not activated unless a request is received from the application stratum that recovery action is requested.

4. Conclusion

In this draft we have discussed two generic use cases that motivate the usefulness of general interfaces for cross stratum optimization in the network core. In our first use case network resource usage became significant due to the aggregation of many individually unique client demands. While in the second use case where data centers were communicating with each other bandwidth usage was already significant enough to warrant the use of private line/LAN type of network services.

Both use cases result in optimization problems that trade off computational versus network costs and constraints. Both featured scenarios where advanced reservation, on demand, and recovery type service interfaces could prove beneficial. Many concepts from recent standardization work at the IETF [10] such as location identifiers, and endpoint properties could be reused in defining such interfaces.

5. Security Considerations

TBD

6. IANA Considerations

This informational document does not make any requests for IANA action.

7. References

7.1. Informative References

[1] M. Armbrust et al., "A view of cloud computing," Communications of the ACM, vol. 53, p. 50-58, Apr. 2010.

[2] "Location Information | DuPont Fabros Technology." (Online). Available: http://www.dft.com/data-centers/location-information.

[3] "Amazon CloudFront." (Online). Available: http://aws.amazon.com/cloudfront/.

[4] K. A. Hua and S. Sheu, "Skyscraper broadcasting: a new broadcasting scheme for metropolitan video-on-demand systems," in Proceedings of the ACM SIGCOMM ’97 conference on Applications, technologies, architectures, and protocols for computer communication, Cannes, France, 1997, pp. 89-100.

[5] "Adobe Flash Media Server 4.0 * Building peer-assisted networking applications." (Online). Available: http://help.adobe.com/en_US/flashmediaserver/devguide/WSa4cb07693d123884520b86f312a354ba36d-8000.html.

[6] Rudra Dutta and George N. Rouskas, "Traffic grooming in WDM networks: Past and future," IEEE Network, vol. 16, no. 6, pp. 46 -56, 2002.

[7] Keyao Zhu and B. Mukherjee, "Traffic grooming in an optical WDM mesh network," Selected Areas in Communications, IEEE Journal on, vol. 20, no. 1, pp. 122-133, 2002.

[8] G. Bernstein, B. Rajagopalan, and D. Saha, Optical Network Control: Architecture, Protocols, and Standards. Addison-Wesley Professional, 2003.

[9] "Our IP Geolocation Products | Quova, Inc." (Online). Available: http://www.quova.com/what/products/.

[10] "draft-ietf-alto-reqs-09." (Online). Available: http://datatracker.ietf.org/doc/draft-ietf-alto-reqs/.

[11] "Cloud Computing’s Tipping Point -- InformationWeek." (Online). Available: http://www.informationweek.com/news/government/cloud-saas/229401691.

[12] "Lessons From FarmVille: How Zynga Uses The Cloud -- InformationWeek." (Online). Available: http://www.informationweek.com/news/global-cio/interviews/229402805#.

[13] "Amazon EC2 Pricing." (Online). Available: http://aws.amazon.com/ec2/pricing/.

[14] Dynamic Workload Balancing with EMC VPLEX and Ciena Networking. EMC, 2010.

[15] "MySQL.:: MySQL Cluster Features." (Online). Available: http://www.mysql.com/products/cluster/features.html#geo.

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

[17] B. Niven-Jenkins (Ed.), G. Watson, N. Bitar, J. Medved, S. Previdi, "Use Cases for ALTO within CDNs", work in progress, draft-jenkins-alto-cdn-use-cases.

[18] E. Mannie, Ed., "GMPLS Framework Generalized Multi-Protocol Label Switching (GMPLS) Architecture" RFC 3945, October 2004.

[19] G. Bernstein, E. Mannie, V. Sharma, E. Gray, "Framework for Generalized Multi-Protocol Label Switching (GMPLS)-based Control of Synchronous Digital Hierarchy/Synchronous Optical Networking (SDH/SONET) Networks", RFC 4257, December 2005.

[20] Y. Lee, Ed., G. Bernstein, Ed., W. Imajuku, "WSON Framework Framework for GMPLS and Path Computation Element (PCE) Control of Wavelength Switched Optical Networks (WSONs)", RFC6163, April 2011.

[21] A. Farrel, J.-P. Vasseur, J. Ash, "PCE Framework A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.

[22] G. Swallow, J. Drake, H. Ishimatsu, Y. Rekhter, "Generalized Multiprotocol Label Switching (GMPLS) User-Network Interface (UNI): Resource ReserVation Protocol-Traffic Engineering(RSVP-TE) Support for the Overlay Model" RFC 4208, October 2005.

Author's Addresses

Greg M. Bernstein
Grotto Networking
Fremont California, USA
Phone: (510) 573-2237
Email: gregb@grotto-networking.com

Young Lee
Huawei Technologies
1700 Alma Drive, Suite 500
Plano, TX 75075
USA
Phone: (972) 509-5599
Email: ylee@huawei.com

Intellectual Property Statement

Disclaimer of Validity

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.

ALTO Working Group
Internet-Draft
Intended status: Informational
Expires: September 5, 2011

Z. Dulinski
Jagiellonian University
P. Wydrych
R. Stankiewicz
P. Cholda
M. Kantor
AGH University of Science and Technology
March 4, 2011

Inter-ALTO Communication Problem Statement
draft-dulinski-alto-inter-problem-statement-00

Abstract

This draft considers an approach to the optimization of the traffic generated by the overlay (peer-to-peer) applications, where some information on inter-AS (Autonomous System) paths is obtained with the usage of dedicated communication scheme known as inter-ALTO communication.

The large amount of network traffic generated by overlay applications requires effective management. This traffic traverses inter-AS links and thus generates substantial costs for the operators and poses problems with overload on the external and internal links. The traffic is not time-stable as the peers connect and disconnect very often. Additionally, when the overlay traffic is observed on inter-AS links, it can be seen that sources and destinations change in a very short period of time. The ALTO (Application-Layer Traffic Optimization) service provides the information that enables more efficient management of the overlay traffic. Such applications can use the information to perform better-than-random peer selection. The ALTO servers are responsible for a pre-selection procedure; the final selection is done by overlay clients and then the ALTO protocol conveys network information to applications. To be credible, this information should be as close to real network situation as possible. However, some types of data are not hold by an operator, but they should be gained on the basis of the additional information exchange with other AS operators. This document presents rationale for the need of introduction of the inter-ALTO communication.

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 September 5, 2011.

Copyright Notice

Table of Contents

1. Introduction

This document describes the rationale for a communication to be used between ALTO servers located in different autonomous systems (AS). Such an inter-ALTO communication extends the ALTO service [RFC5693]capabilities and provides additional information on remote peers, i.e., peers located in other ASes. To make the consideration more clear we distinguish local AS and remote ASes. Local AS is the one from which perspective we describe the communication. Local peers are located in the local AS and are served by a local ALTO server. On the contrary, all other peers are located in remote ASes. Those peers are referred to as remote and are served by remote ALTO server. This basic terminology adheres to majority of considerations in this document.

The motivation for the ALTO service as discussed in the ALTO problem statement [RFC5693] focuses on the overlay traffic optimization based on information gathered from the Internet Service Provider (ISP) domain, i.e., an Autonomous System (AS). Due to the suggested approach, information on the AS internal topology and some routing information obtained from the global Internet (the BGP routing tables) may be used for the peer selection procedures. The data transfer cost can be also incorporated. However, there are some parameters which can be used for the better peer selection mechanism, but they are not available in the local AS and must be obtained from outside, i.e., from a remote AS. For example, the BGP routing information available in the AS identifies only the upstream traffic. The information about the downstream traffic is not present or is incomplete and the full BGP information for this traffic could be obtained from the remote AS containing the subnetwork where the peer sending downstream traffic is attached. In order to obtain such data, we propose the inter-ALTO communication.

It is assumed that the ALTO servers are deployed in the local and remote ASes. The ALTO server located in the client AS can request desired information from the ALTO server which is located in the remote AS. Each server is managed by a respective ISP. Each ISP decides what type of information can be exposed to the requesting party. The ALTO server responds with the type of information that was previously agreed to exchange. Each local ALTO server has to discover the address of the remote ALTO server before starting the communication. The discovery procedure may use the IP addresses of remote peers for the establishment of IP addresses of remote ALTO servers. The detailed analysis of this functionality is out of scope of this document.

The information delivered by remote ALTO servers may be used by a local ALTO server to perform advanced rating/ranking procedure of

peers. The general idea is as follows.

1. A peer receives a list of other peers from a tracker, i.e., a list of potential candidates to communicate with.

2. A peer uses the ALTO protocol [I-D.ietf-alto-protocol] to send the list of peers to its local ALTO server.

3. Local ALTO server obtains additional information on remote peers by communicating respective remote ALTO servers. If sufficient information is available locally and the inter-ALTO communication is not needed, this step is omitted.

4. Using ISP specific policies and values of parameters associated with remote peers the local ALTO server performs rating/ranking procedure.

5. Sorted/rated list of peers is sent back to the peer with usage of the ALTO protocol.

The requirements, syntax and detailed operation of the inter-ALTO communication as well as the rating/ranking procedure is out of scope of this document.

2. Definitions

In the scope of this document we use all the definitions specified in the Section 2 of ALTO problem statement [RFC5693]. Moreover, the following terms have the special meaning.

Local Peer: A peer which belongs to the same Autonomous System to which the ALTO client belongs.

Remote Peer: A peer which belongs to another Autonomous System than the one to which the ALTO client belongs.

Local AS: The Autonomous System to which the ALTO client belongs.

Remote AS: An Autonomous System to which a remote peer belongs.

Local ALTO Server: The ALTO server serving the ALTO client and the local peers.

Remote ALTO Server: An ALTO server serving remote peers.

3. The Problem and Motivation

ALTO server optimization capabilities are limited by the fact that they use information available locally only. It can be shown that more information on remote peers, a routing path, or remote ISP preferences would be useful. The data from remote ASes obtained by the external interface as shown in Figure 1 of the ALTO protocol draft [I-D.ietf-alto-protocol] may have a substantial significance for the management of overlay traffic (e.g. with respect to peer rating, ranking, or the choice of the best peers). The suggested approach to deliver these types of information is defined in the inter-ALTO communication discussed in this document.

The ALTO service may also be used by the client-server applications, supporting the best choice of the mirror servers. If some mirror servers are in other ASes than the client's AS, some information about the network conditions in the remote ASes may be delivered only by the ALTO servers localized in these ASes. Both clients and mirror servers may communicate with their local ALTO servers for delivering traffic optimization parameters. As an ALTO client communicates only with its local ALTO server, the information from remote ASes is accessible only via inter-ALTO communication.

The ALTO-based traffic optimization may be also used in the context of the Content Delivery Networks (CDNs) [I-D.penno-alto-cdn]. Penno et al. discuss how the ALTO service can be used in the existing and future CDNs. They consider the case when the CDN nodes are in multiple administrative domains. In that case the inter-ALTO communication is used.

The basic ALTO architecture presented in Figure 1 of the ALTO protocol draft [I-D.ietf-alto-protocol] defines the external interface used to communicate with other information sources like remote ALTO servers. The ALTO Protocol draft, however, does not define what information should be exchanged between ALTO servers to optimize the traffic. The inter-ALTO communication proposed by the current document implements the external interface as defined by the ALTO protocol draft.

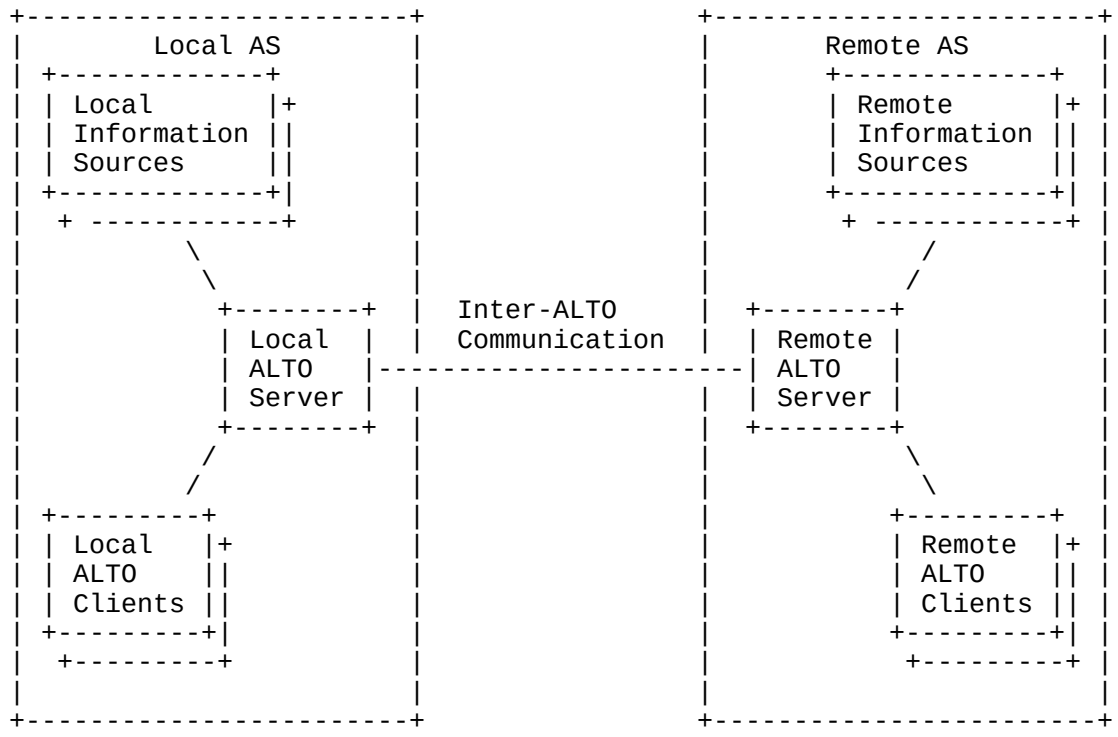


Figure 1: Inter-ALTO communication architecture.

The architecture of the Inter-ALTO communication is shown in Figure 1. Both ALTO servers gather the information from their information sources like routing protocols, provisioning policies, or dynamic network information sources. The local ALTO server needs to communicate with a remote ALTO server to obtain information which is available only at the entities in the remote AS.

In particular, the following key aspects motivate the proposal of the inter-ALTO communication.

- Route asymmetry.
- Different types of business relations.
- Congestion avoidance.
- Proximity awareness (distance to the remote AS), e.g.:
  - number of inter-AS hops;

    * delay (RTT).

- Remote ISP's preference.
- Coordination of ISPs' policies.
- Outdated information.

3.1. Route Asymmetry

The communication between two ASes does not need to follow the same path in the upstream and downstream direction. It was shown that about 29% of paths between AS pairs in the Internet are fully symmetric, i.e., upstream and downstream traffic follows exactly the same path [ICC.optimal]. In 51% of cases the number of inter-AS hops is different for the upstream and downstream direction. Additionally, in 50.5% of all path pairs a neighbor AS for upstream and downstream paths are different.

The ALTO server can obtain routing information locally (e.g. from BGP routers) and can determine the upstream path. Information about the downstream path is usually not easily available. Some additional routing information can be obtained from Looking Glass Servers, but not all ASes provide them. The inter-ALTO communication provides the ability to exchange the relevant information between ALTO servers. Especially, the downstream path can be reliably determined using the information provided by remote ALTO server. In the light of route asymmetry in the Internet such information appears to be necessary for a better optimization of a peer rating/ranking algorithm, as assumption that the inter-AS routes follow symmetrical paths can give not only sub-optimal, but misleading and, in effect, harmful results.

3.2. Different Types of Business Relations

Two basic business relations between ISPs may be distinguished.

When two ISPs agree to exchange the traffic without any charge, such a relation is called peering. The inter-domain link between the respective ASes is also called a peering link. Usually, there is no charge if the difference between traffic volumes passing such a link in different directions does not exceed a previously agreed limit.

The other case occurs when one ISP serves as a network provider to another ISP (e.g. relation between tier 2 and tier 3 ISPs). In such a case one ISP (acting as a customer) has to pay the other ISP (acting as provider) for the traffic sent over the inter-AS link connecting them. The real monetary cost of the traffic volume exchanged on such a link depends on agreements between ISPs. In

general, some links may be considered as cheaper or more expensive.

AS may be connected to many other ASes with various agreements. The cost of the inter-AS traffic transfer may differ depending on which neighbor AS the path passes. For this reason an ISP may prefer that its own customers exchange data with remote peers located in such ASes that the path directed to them passes cheaper links. The ALTO server may sort peers taking into account these criteria. To receive almost complete information on routing paths to and from different remote domains the information provided by remote ALTO server using inter-AS communication may be helpful.

3.3. Congestion Avoidance

A peer rating/ranking procedure may also take into account the congestions on inter-AS links. An ISP is able to monitor queues on its inter-domain links and assign metrics indicating the buffer occupancy or bandwidth utilization. These metrics can express percentage use of buffers or bandwidth on a particular inter-AS link. If one inter-domain link is congested it is desirable to promote peers reachable through lightly loaded links. Again, information provided by the remote ALTO server would support such optimization. The aim of the inter-ALTO communication is not to replace the existing congestion avoidance mechanisms. The idea is to support the present mechanism by the exchange of parameters describing the load on the inter-AS links.

3.4. Proximity Awareness

For a set of reasons (e.g. the performance of an application) the ALTO server may suggest its customers to connect to remote peers located in its proximity. The simplest measure of proximity is the number of inter-AS hops. As indicated above, due to the route asymmetry, the number of hops may significantly differ between the upstream and downstream paths. Such information for the downstream path may be provided by the remote ALTO server. A more advanced metric of proximity can be found in the delay that can be approximated by exchanging messages between ALTO servers. The ALTO servers can be equipped with an application-layer ping functionality which only operates between ALTO servers. By exchanging special packets prepared by the ALTO servers, these servers can estimate delay and packet loss.

3.5. Remote ISP’s Preference

If two ISPs agree on a cooperation, the remote ALTO server may provide its preference parameters (remote preference parameters) indicating which peers are better from the point of view of the

remote ISP. For instance, the AS in which the remote ALTO server is located may possess two subnetworks connected to the operator's core network by distinct links. It may happen that a connection to one of the subnetworks is cheaper than the other. The remote operator may prefer connections through cheaper link, so peers located in the subnetwork transferring data via this cheaper link are preferred.

The remote preference parameter may be also used when a remote ISP wants to suggest peers which are connected to the Internet through access links of higher capacity. This way, the remote ALTO server, without exposing the exact values of access link bandwidth, may indicate peers with higher throughput. The remote preference parameters have only local meaning, i.e., their values are comparable for peers located in the same AS only.

If a remote ISP does not want to reveal numerical values of network parameters related to its peers (such information might be considered as confidential) the remote ALTO server may perform a rating/ranking procedure and assign priority parameter to its peers. The rating/ ranking criteria may remain hidden for the requesting local ALTO server.

3.6. Coordination of ISPs' Policies

Operators may have an incentive to coordinate their efforts in order to decrease transfer costs on inter-AS links or improve quality experienced by peers, i.e., coordinate their peer rating/ranking strategies. This way, operators may avoid contradictory strategies resulting in inefficiency of rating/ranking algorithms. Operators may agree to promote each other's peers.

For example, it may happen that operator A wanting to decrease traffic on one of its links discourages its own peers from communicating with peers located in operator B's domain. On the other hand, operator B would consider peers located in a domain of operator A as very attractive for its own peers. As a result, rating/ranking procedures performed by respective ALTO servers give contradictory results what may decrease the effectiveness of these procedures. To avoid such a situation, the inter-ALTO communication is needed.

Another example of a usefulness of coordination of policies is clustering of ASes. Recent studies [IJNM.unfairness] have shown that locality promotion might be ineffective or even harmful if used in AS with small number of peers. A proposed solution is to create a cluster of two or more ASes. Then ALTO servers serving different ASes in the cluster treat all peers located in the cluster as if they were in a single AS. In other words, from a point of view of

locality promotion algorithm all peers located in the cluster are local, regardless of their home AS.

3.7. Sensitivity of Topology Information

The minimum information that the remote AS provides to the local ALTO server via the inter-ALTO communication may be the number of inter-AS hops and the number of the local AS's neighbor in the downstream path (the full downstream AS_PATH may be not exchanged). Such information does not reveal any sensitive information neither on the ISP internal topology details nor remote AS connections with other ASes, but does provide basic and very useful information for the local ALTO server.

3.8. Outdated Information

It is expected that some information (parameters) from routing protocols that will be used in the rating/ranking procedures may outdate. Also information related to the network performance is constantly changing. Therefore, the information obtained from the remote AS requires updates. This updates may be generated on request (pull mechanism), on event base schema or periodically (push mechanism). The inter-ALTO communication should be equipped with mechanisms for updates. The need for the present information describing network conditions and some routing parameters are arguments for introducing specific protocol for the communication between ALTO servers.

4. Usage of the Mechanisms Offered by the ALTO Protocol

The basic ALTO protocol architecture allows an ALTO server to communicate with a third party through the external interface. The inter-ALTO communication may use some functionalities offered by the ALTO protocol [I-D.ietf-alto-protocol].

Server Information Service: This service defined by the ALTO protocol may be extended in order to provide information about server's ability to cooperate with other ALTO servers. Thanks to this service, the other ALTO servers may acquire the information about available parameters and their definitions. These parameters may be used by cooperating ALTO servers for the peer rating/ranking procedures. The access for this service may be restricted. Some information may be accessible only by the privileged ALTO servers after the successful authentication.

ALTO Information Services: These services has been defined to provide the query information services for ALTO clients. All the information delivered by these services has local meaning. This information is related to the locally defined parameters describing a particular ISP's network. Some part of this information managed by a remote ALTO server may be useful for the requesting local ALTO server. The requesting ALTO server obtains this information via inter-ALTO communication. After receiving the response, the local ALTO server has to perform some calculations, scaling, merging, or adaptation of the received parameters. In this way the local ALTO server may conform to both its internal network topology and measurements, and the external ones. However, it should be stressed that the ALTO Information Services is designed for communication between ALTO clients and servers, not for the inter-ALTO communication.

Network Map: This structure is defined by the ISP and reflects the internal structure of the ISP network. This structure has only a local meaning and, generally, it is not unique for all entities within the Internet.
A particular network map can be used by different operators. The requesting ALTO server usually has to perform some prediction of the external topology on its own. The ALTO server has to apply its own rules and definitions. The PIDs, defined in the remote ALTO server, have to be mapped on the PID structure defined in the local AS.

Cost Map: This structure also has the local meaning. The local ALTO server may receive the network map and the cost map from a remote ALTO server. These costs may require recalculation in order to unify the cost measures in the local AS. After these operations, if it is needed, the rating/ranking procedure can be performed.

5. Security Considerations

The communication between ALTO servers requires authentication and authorization procedures. In some cases it may require establishment of the secured tunnels between the partner ALTO servers. The minimum security requirements for the inter-ALTO communication is out of scope of this document.

The inter-ALTO communication allows ALTO servers to exchange any parameters which improve the performance of the overlay traffic, or, generally, allows them to manage overlay traffic. In order to achieve this results a group ISP may exchange sensitive data, the exchanged parameters may be confidential. They should not be

accessible by a third party, e.g., some other ISPs or peers.

An ISP may have its own policy how organize the overlay traffic and this policy may use a number of parameters during the evaluation procedure. The policy result may be delivered to peers in many ways. It can take the form of a sorted peer list without any parameters, a sorted list with some parameters which are derived from the parameters exchanged in the inter-ALTO communication, or raw exchanged parameters. ISPs may have an incentive not to expose these parameters in the raw form to peers. The mentioned sensitive parameters require applying a higher level of the security procedures.

In order to keep the exchanged parameters confidential it may be reasonable to keep the communications between peers and ALTO server from communication among ALTO servers by the protocol differentiation separated. Different security procedures may be easier to manage if these communication procedures take the form of two distinct protocols. This protocol separation allows to define mechanisms which are specific for the inter-ALTO communication only. The protocol should not allow to use this mechanism by overlay peers. The set of procedures for the inter-ALTO communication is expected to be separated from the client ALTO communication and this can be achieved by distinct protocols.

6. IANA Considerations

This document has no actions for IANA.

7. Acknowledgements

This draft evolved from draft-dulinski-alto-inter-alto-protocol-00. The authors would like to thank all authors of the Inter-ALTO communication protocol draft for their contributions.

This work has been partially supported by the EU through the ICT FP7 Project SmoothIT (FP7-2007-ICT-216259).

8. Informative References

[I-D.ietf-alto-protocol]
Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-06 (work in progress), October 2010.

[I-D.penno-alto-cdn]
Penno, R., Raghunath, S., Medved, J., Alimi, R., Yang, R., and S. Previdi, "ALTO and Content Delivery Networks", draft-penno-alto-cdn-02 (work in progress), October 2010.

[ICC.optimal]
Dulinski, Z., Kantor, M., Krzysztofek, W., Stankiewicz, R., and P. Cholda, "Optimal Choice of Peers based on BGP Information", Proceedings of 2010 IEEE International Conference on Communications (ICC), May 2010.

[IJNM.unfairness]
Lehrieder, F., Oechsner, S., Hossfeld, T., Staehle, D., Despotovic, Z., Kellerer, W., and M. Michel, "Mitigating unfairness in locality-aware peer-to-peer networks", International Journal of Network Management, Volume 21, Issue 1, pp. 3-20, January/February 2011.

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

Authors' Addresses

Zbigniew Dulinski
Jagiellonian University
ul. Reymonta 4
Krakow 30-059
Poland

Phone: +48 12 663 5571
Fax: +48 12 633 4079
Email: dulinski@th.if.uj.edu.pl

Piotr Wydrych
AGH University of Science and Technology
al. Mickiewicza 30
Krakow 30-059
Poland

Phone: +48 12 617 4817
Fax: +48 12 634 2372
Email: piotr.wydrych@agh.edu.pl

Rafal Stankiewicz
AGH University of Science and Technology
al. Mickiewicza 30
Krakow 30-059
Poland

Phone: +48 12 617 4036
Fax: +48 12 634 2372
Email: rstankie@agh.edu.pl

Piotr Cholda
AGH University of Science and Technology
al. Mickiewicza 30
Krakow 30-059
Poland

Phone: +48 12 617 4036
Fax: +48 12 634 2372
Email: piotr.cholda@agh.edu.pl

Miroslaw Kantor
AGH University of Science and Technology
al. Mickiewicza 30
Krakow 30-059
Poland

Phone: +48 12 617 2852
Fax: +48 12 634 2372
Email: kantor@kt.agh.edu.pl

ALTO Working Group
Internet-Draft
Intended status: Informational
Expires: January 12, 2012

Z. Dulinski
Jagiellonian University
P. Wydrych
R. Stankiewicz
AGH University of Science and Technology
July 11, 2011

Inter-ALTO Communication Problem Statement
draft-dulinski-alto-inter-problem-statement-01

Abstract

This draft considers an approach to the optimization of the traffic generated by the overlay (peer-to-peer) applications, where some information on inter-AS (Autonomous System) paths is obtained with the usage of dedicated communication scheme known as inter-ALTO communication.

The large amount of network traffic generated by overlay applications requires effective management. This traffic traverses inter-AS links and thus generates substantial costs for the operators and poses problems with overload on the external and internal links. The traffic is not time-stable as the peers connect and disconnect very often. Additionally, when the overlay traffic is observed on inter-AS links, it can be seen that sources and destinations change in a very short period of time. The ALTO (Application-Layer Traffic Optimization) service provides the information that enables more efficient management of the overlay traffic. Such applications can use the information to perform better-than-random peer selection. The ALTO servers are responsible for a pre-selection procedure; the final selection is done by overlay clients and then the ALTO protocol conveys network information to applications. To be credible, this information should be as close to real network situation as possible. However, some types of data are not hold by an operator, but they should be gained on the basis of the additional information exchange with other AS operators. This document presents rationale for the need of introduction of the inter-ALTO communication.

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 January 12, 2012.

Table of Contents

1. Introduction

This document describes the rationale for a communication to be used between ALTO servers located in different autonomous systems (AS). Such an inter-ALTO communication extends the ALTO service [RFC5693]capabilities and provides additional information on remote peers, i.e., peers located in other ASes. To make the consideration more clear we distinguish local AS and remote ASes. Local AS is the one from which perspective we describe the communication. Local peers are located in the local AS and are served by a local ALTO server. On the contrary, all other peers are located in remote ASes. Those peers are referred to as remote and are served by remote ALTO server. This basic terminology adheres to majority of considerations in this document.

The motivation for the ALTO service as discussed in the ALTO problem statement [RFC5693] focuses on the overlay traffic optimization based on information gathered from the Internet Service Provider (ISP) domain, i.e., an Autonomous System (AS). Due to the suggested approach, information on the AS internal topology and some routing information obtained from the global Internet (the BGP routing tables) may be used for the peer selection procedures. The data transfer cost can be also incorporated. However, there are some parameters which can be used for the better peer selection mechanism, but they are not available in the local AS and must be obtained from outside, i.e., from a remote AS. For example, the BGP routing information available in the AS identifies only the upstream traffic. The information about the downstream traffic is not present or is incomplete and the full BGP information for this traffic could be obtained from the remote AS containing the subnetwork where the peer sending downstream traffic is attached. In order to obtain such data, we propose the inter-ALTO communication.

It is assumed that the ALTO servers are deployed in the local and remote ASes. The ALTO server located in the client AS can request desired information from the ALTO server which is located in the remote AS. Each server is managed by a respective ISP. Each ISP decides what type of information can be exposed to the requesting party. The ALTO server responds with the type of information that was previously agreed to exchange. Each local ALTO server has to discover the address of the remote ALTO server before starting the communication. The discovery procedure may use the IP addresses of remote peers for the establishment of IP addresses of remote ALTO servers. The detailed analysis of this functionality is out of scope of this document.

The information delivered by remote ALTO servers may be used by a local ALTO server to perform advanced rating/ranking procedure of

peers. The general idea is as follows.

1. A peer receives a list of other peers from a tracker, i.e., a list of potential candidates to communicate with.

2. A peer uses the ALTO protocol [I-D.ietf-alto-protocol] to send the list of peers to its local ALTO server.

3. Local ALTO server obtains additional information on remote peers by communicating respective remote ALTO servers. If sufficient information is available locally and the inter-ALTO communication is not needed, this step is omitted.

4. Using ISP specific policies and values of parameters associated with remote peers the local ALTO server performs rating/ranking procedure.

5. Sorted/rated list of peers is sent back to the peer with usage of the ALTO protocol.

The requirements, syntax and detailed operation of the inter-ALTO communication as well as the rating/ranking procedure is out of scope of this document.

2. Definitions

In the scope of this document we use all the definitions specified in the Section 2 of ALTO problem statement [RFC5693]. Moreover, the following terms have the special meaning.

Local Peer: A peer which belongs to the same Autonomous System to which the ALTO client belongs.

Remote Peer: A peer which belongs to another Autonomous System than the one to which the ALTO client belongs.

Local AS: The Autonomous System to which the ALTO client belongs.

Remote AS: An Autonomous System to which a remote peer belongs.

Local ALTO Server: The ALTO server serving the ALTO client and the local peers.

Remote ALTO Server: An ALTO server serving remote peers.

3. The Problem and Motivation

ALTO server optimization capabilities are limited by the fact that they use information available locally only. It can be shown that more information on remote peers, a routing path, or remote ISP preferences would be useful. The data from remote ASes obtained by the external interface as shown in Figure 1 of the ALTO protocol draft [I-D.ietf-alto-protocol] may have a substantial significance for the management of overlay traffic (e.g. with respect to peer rating, ranking, or the choice of the best peers). The suggested approach to deliver these types of information is defined in the inter-ALTO communication discussed in this document.

The ALTO service may also be used by the client-server applications, supporting the best choice of the mirror servers. If some mirror servers are in other ASes than the client's AS, some information about the network conditions in the remote ASes may be delivered only by the ALTO servers localized in these ASes. Both clients and mirror servers may communicate with their local ALTO servers for delivering traffic optimization parameters. As an ALTO client communicates only with its local ALTO server, the information from remote ASes is accessible only via inter-ALTO communication.

The ALTO-based traffic optimization may be also used in the context of the Content Delivery Networks (CDNs) [I-D.jenkins-alto-cdn-use-cases]. The draft by Niven-Jenkins et al. shows how CDNs may benefit from the information provided by the ALTO protocol. Local information, however, may be not sufficient to optimize the request routing process. The information gathered from ALTO servers in other domains may be needed.

The basic ALTO architecture presented in Figure 1 of the ALTO protocol draft [I-D.ietf-alto-protocol] defines the external interface used to communicate with other information sources like remote ALTO servers. The ALTO Protocol draft, however, does not define what information should be exchanged between ALTO servers to optimize the traffic. The inter-ALTO communication proposed by the current document implements the external interface as defined by the ALTO protocol draft.

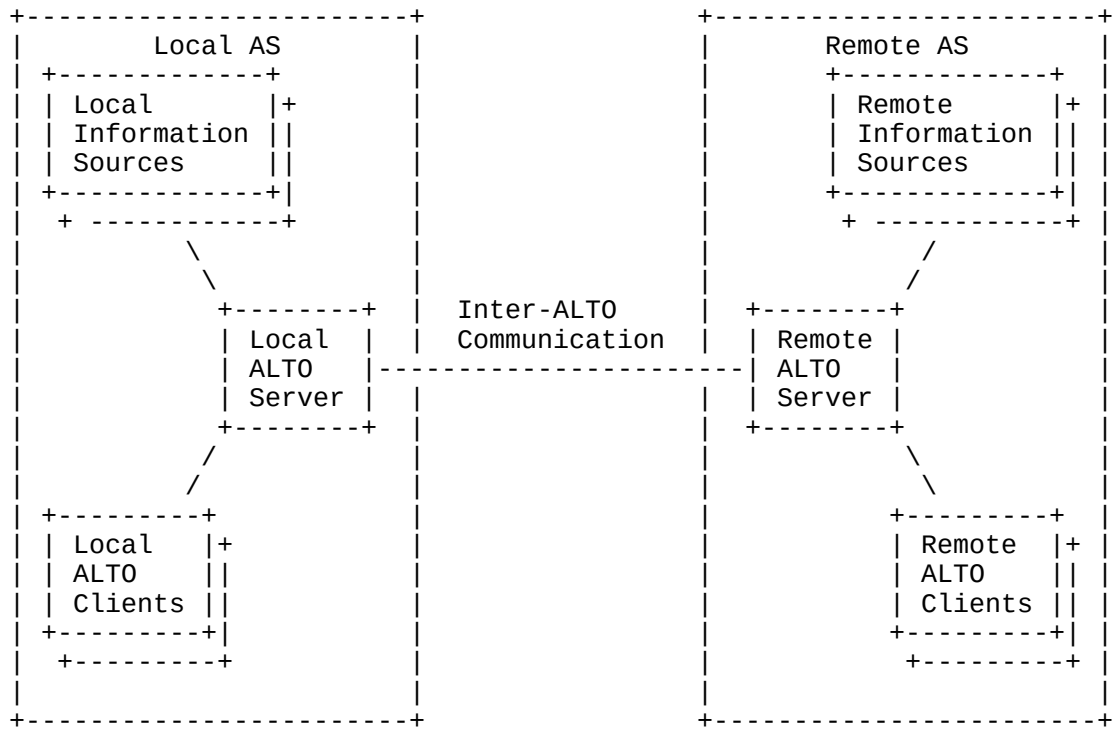


Figure 1: Inter-ALTO communication architecture.

The architecture of the Inter-ALTO communication is shown in Figure 1. Both ALTO servers gather the information from their information sources like routing protocols, provisioning policies, or dynamic network information sources. The local ALTO server needs to communicate with a remote ALTO server to obtain information which is available only at the entities in the remote AS.

In particular, the following key aspects motivate the proposal of the inter-ALTO communication.

- Route asymmetry.
- Different types of business relations.
- Congestion avoidance.
- Proximity awareness (distance to the remote AS), e.g.:
  - number of inter-AS hops;

    * delay (RTT).

- Remote ISP's preference.
- Coordination of ISPs' policies.
- Outdated information.

3.1. Route Asymmetry

The communication between two ASes does not need to follow the same path in the upstream and downstream direction. It was shown that about 29% of paths between AS pairs in the Internet are fully symmetric, i.e., upstream and downstream traffic follows exactly the same path [ICC.optimal]. In 51% of cases the number of inter-AS hops is different for the upstream and downstream direction. Additionally, in 50.5% of all path pairs a neighbor AS for upstream and downstream paths are different.

The ALTO server can obtain routing information locally (e.g. from BGP routers) and can determine the upstream path. Information about the downstream path is usually not easily available. Some additional routing information can be obtained from Looking Glass Servers, but not all ASes provide them. The inter-ALTO communication provides the ability to exchange the relevant information between ALTO servers. Especially, the downstream path can be reliably determined using the information provided by remote ALTO server. In the light of route asymmetry in the Internet such information appears to be necessary for a better optimization of a peer rating/ranking algorithm, as assumption that the inter-AS routes follow symmetrical paths can give not only sub-optimal, but misleading and, in effect, harmful results.

3.2. Many ASes within One ISP

An ISP may possess a complex topology network composed of many autonomous systems. Current ALTO specification allows for deployment of independent ALTO servers in each AS. In such a case the overlay traffic management performed by the ALTO server is restricted to a single AS since cost maps have a local meaning. An ISP operating a multi-AS network may be interested in managing the traffic in the whole administrative domain in a consistent and coordinated manner. The information possessed by a single ALTO server is insufficient. To obtain a complete knowledge on the multi-AS network a communication between ALTO servers is needed. As a result, local cost maps originating from different autonomous systems may be coordinated. A uniform cost map reflecting the whole network structure may be created and distributed between ALTO servers.

3.3. Different Types of Business Relations

Two basic business relations between ISPs may be distinguished.

When two ISPs agree to exchange the traffic without any charge, such a relation is called peering. The inter-domain link between the respective ASes is also called a peering link. Usually, there is no charge if the difference between traffic volumes passing such a link in different directions does not exceed a previously agreed limit.

The other case occurs when one ISP serves as a network provider to another ISP (e.g. relation between tier 2 and tier 3 ISPs). In such a case one ISP (acting as a customer) has to pay the other ISP (acting as provider) for the traffic sent over the inter-AS link connecting them. The real monetary cost of the traffic volume exchanged on such a link depends on agreements between ISPs. In general, some links may be considered as cheaper or more expensive.

AS may be connected to many other ASes with various agreements. The cost of the inter-AS traffic transfer may differ depending on which neighbor AS the path passes. For this reason an ISP may prefer that its own customers exchange data with remote peers located in such ASes that the path directed to them passes cheaper links. The ALTO server may sort peers taking into account these criteria. To receive almost complete information on routing paths to and from different remote domains the information provided by remote ALTO server using inter-AS communication may be helpful.

3.4. Congestion Avoidance

A peer rating/ranking procedure may also take into account the congestions on inter-AS links. An ISP is able to monitor queues on its inter-domain links and assign metrics indicating the buffer occupancy or bandwidth utilization. These metrics can express percentage use of buffers or bandwidth on a particular inter-AS link. If one inter-domain link is congested it is desirable to promote peers reachable through lightly loaded links. Again, information provided by the remote ALTO server would support such optimization. The aim of the inter-ALTO communication is not to replace the existing congestion avoidance mechanisms. The idea is to support the present mechanism by the exchange of parameters describing the load on the inter-AS links.

3.5. Proximity Awareness

For a set of reasons (e.g. the performance of an application) the ALTO server may suggest its customers to connect to remote peers located in its proximity. The simplest measure of proximity is the

number of inter-AS hops. As indicated above, due to the route asymmetry, the number of hops may significantly differ between the upstream and downstream paths. Such information for the downstream path may be provided by the remote ALTO server. A more advanced metric of proximity can be found in the delay that can be approximated by exchanging messages between ALTO servers. The ALTO servers can be equipped with an application-layer ping functionality which only operates between ALTO servers. By exchanging special packets prepared by the ALTO servers, these servers can estimate delay and packet loss.

3.6. Remote ISP's Preference

If two ISPs agree on a cooperation, the remote ALTO server may provide its preference parameters (remote preference parameters) indicating which peers are better from the point of view of the remote ISP. For instance, the AS in which the remote ALTO server is located may possess two subnetworks connected to the operator's core network by distinct links. It may happen that a connection to one of the subnetworks is cheaper than the other. The remote operator may prefer connections through cheaper link, so peers located in the subnetwork transferring data via this cheaper link are preferred.

The remote preference parameter may be also used when a remote ISP wants to suggest peers which are connected to the Internet through access links of higher capacity. This way, the remote ALTO server, without exposing the exact values of access link bandwidth, may indicate peers with higher throughput. The remote preference parameters have only local meaning, i.e., their values are comparable for peers located in the same AS only.

If a remote ISP does not want to reveal numerical values of network parameters related to its peers (such information might be considered as confidential) the remote ALTO server may perform a rating/ranking procedure and assign priority parameter to its peers. The rating/ ranking criteria may remain hidden for the requesting local ALTO server.

3.7. Coordination of ISPs' Policies

Operators may have an incentive to coordinate their efforts in order to decrease transfer costs on inter-AS links or improve quality experienced by peers, i.e., coordinate their peer rating/ranking strategies. This way, operators may avoid contradictory strategies resulting in inefficiency of rating/ranking algorithms. Operators may agree to promote each other's peers.

For example, it may happen that operator A wanting to decrease

traffic on one of its links discourages its own peers from communicating with peers located in operator B's domain. On the other hand, operator B would consider peers located in a domain of operator A as very attractive for its own peers. As a result, rating/ranking procedures performed by respective ALTO servers give contradictory results what may decrease the effectiveness of these procedures. To avoid such a situation, the inter-ALTO communication is needed.

Another example of a usefulness of coordination of policies is clustering of ASes. Recent studies [IJNM.unfairness] have shown that locality promotion might be ineffective or even harmful if used in AS with small number of peers. A proposed solution is to create a cluster of two or more ASes. Then ALTO servers serving different ASes in the cluster treat all peers located in the cluster as if they were in a single AS. In other words, from a point of view of locality promotion algorithm all peers located in the cluster are local, regardless of their home AS.

3.8. Sensitivity of Topology Information

The minimum information that the remote AS provides to the local ALTO server via the inter-ALTO communication may be the number of inter-AS hops and the number of the local AS's neighbor in the downstream path (the full downstream AS_PATH may be not exchanged). Such information does not reveal any sensitive information neither on the ISP internal topology details nor remote AS connections with other ASes, but does provide basic and very useful information for the local ALTO server.

3.9. Outdated Information

It is expected that some information (parameters) from routing protocols that will be used in the rating/ranking procedures may outdate. Also information related to the network performance is constantly changing. Therefore, the information obtained from the remote AS requires updates. This updates may be generated on request (pull mechanism), on event base schema or periodically (push mechanism). The inter-ALTO communication should be equipped with mechanisms for updates. The need for the present information describing network conditions and some routing parameters are arguments for introducing specific protocol for the communication between ALTO servers.

3.10. Mobile Networks

The inter-ALTO communication may be very useful for mobile network operators and content providers serving mobile clients. An ALTO server may recognize mobile clients and properly assign them to PIDs.

Some information about the mobile network resources gathered from mobile network nodes located in different networks should be exchanged between operators for better then random peer selection. ALTO servers should posses information which allows to make proper peer selection, taking into account, e.g., the mobile network load (including the load in the radio access network and in the circuit- and packet-switched domains).

After collecting the load information, the ALTO server may assign priorities. These priorities may exemplify the load in some parts of the radio access network. Via the inter-ALTO communication, the priorities may be passed to the other operator's networks where other clients are located. Relying on this information, the ALTO server may optimize the connections between clients.

3.11. Route Aggregation

The BGP protocol allows the aggregation of specific routes into one route. In such a case the aggregate route is advertised. The full path is either lost completely or the AS set information is available. In the latter case only the set of ASes behind the aggregating router is known but the detailed information about the routing path, including AS sequence and AS-hop count, is lost. From the overlay traffic optimization point of view the knowledge on ASes located behind aggregating router and the number as well as sequence of inter-AS hops may be useful, e.g., because of route asymmetry problem described earlier (Section 3.1). The solution for this problem is information exchange between ALTO servers located in ASes ahead and behind the router aggregating routes.

4. Usage of the Mechanisms Offered by the ALTO Protocol

The basic ALTO protocol architecture allows an ALTO server to communicate with a third party through the external interface. The inter-ALTO communication may use some functionalities offered by the ALTO protocol [I-D.ietf-alto-protocol].

Server Information Service: This service defined by the ALTO protocol may be extended in order to provide information about server's ability to cooperate with other ALTO servers. Thanks to this service, the other ALTO servers may acquire the information about available parameters and their definitions. These parameters may be used by cooperating ALTO servers for the peer rating/ranking procedures. The access for this service may be restricted. Some information may be accessible only by the privileged ALTO servers after the successful authentication.

ALTO Information Services: These services has been defined to provide the query information services for ALTO clients. All the information delivered by these services has local meaning. This information is related to the locally defined parameters describing a particular ISP's network. Some part of this information managed by a remote ALTO server may be useful for the requesting local ALTO server. The requesting ALTO server obtains this information via inter-ALTO communication. After receiving the response, the local ALTO server has to perform some calculations, scaling, merging, or adaptation of the received parameters. In this way the local ALTO server may conform to both its internal network topology and measurements, and the external ones. However, it should be stressed that the ALTO Information Services is designed for communication between ALTO clients and servers, not for the inter-ALTO communication.

Network Map: This structure is defined by the ISP and reflects the internal structure of the ISP network. This structure has only a local meaning and, generally, it is not unique for all entities within the Internet.
A particular network map can be used by different operators. The requesting ALTO server usually has to perform some prediction of the external topology on its own. The ALTO server has to apply its own rules and definitions. The PIDs, defined in the remote ALTO server, have to be mapped on the PID structure defined in the local AS.

Cost Map: This structure also has the local meaning. The local ALTO server may receive the network map and the cost map from a remote ALTO server. These costs may require recalculation in order to unify the cost measures in the local AS. After these operations, if it is needed, the rating/ranking procedure can be performed.

5. Security Considerations

The communication between ALTO servers requires authentication and authorization procedures. In some cases it may require establishment of the secured tunnels between the partner ALTO servers. The minimum security requirements for the inter-ALTO communication is out of scope of this document.

The inter-ALTO communication allows ALTO servers to exchange any parameters which improve the performance of the overlay traffic, or, generally, allows them to manage overlay traffic. In order to achieve this results a group ISP may exchange sensitive data, the exchanged parameters may be confidential. They should not be

accessible by a third party, e.g., some other ISPs or peers.

An ISP may have its own policy how organize the overlay traffic and this policy may use a number of parameters during the evaluation procedure. The policy result may be delivered to peers in many ways. It can take the form of a sorted peer list without any parameters, a sorted list with some parameters which are derived from the parameters exchanged in the inter-ALTO communication, or raw exchanged parameters. ISPs may have an incentive not to expose these parameters in the raw form to peers. The mentioned sensitive parameters require applying a higher level of the security procedures.

In order to keep the exchanged parameters confidential it may be reasonable to keep the communications between peers and ALTO server from communication among ALTO servers by the protocol differentiation separated. Different security procedures may be easier to manage if these communication procedures take the form of two distinct protocols. This protocol separation allows to define mechanisms which are specific for the inter-ALTO communication only. The protocol should not allow to use this mechanism by overlay peers. The set of procedures for the inter-ALTO communication is expected to be separated from the client ALTO communication and this can be achieved by distinct protocols.

6. IANA Considerations

This document has no actions for IANA.

7. Acknowledgements

The authors would like to thank following people for the valuable discussions (in alphabetical order):

- Piotr Cholda (AGH University of Science and Technology)
- Miroslaw Kantor (AGH University of Science and Technology)
- Jan Medved (Juniper)
- Stefano Previdi (Cisco)
- Robert Varga (Pantheon)

This work has been partially supported by the EU through the ICT FP7 Project SmoothIT (FP7-2007-ICT-216259).

8. References

8.1. Normative References

[I-D.ietf-alto-protocol]
Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-09 (work in progress), June 2011.

[ICC.optimal]
Dulinski, Z., Kantor, M., Krzysztofek, W., Stankiewicz, R., and P. Cholda, "Optimal Choice of Peers based on BGP Information", Proceedings of 2010 IEEE International Conference on Communications (ICC), May 2010.

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

8.2. Informative References

[I-D.jenkins-alto-cdn-use-cases]
Niven-Jenkins, B., Watson, G., Bitar, N., Medved, J., and S. Previdi, "Use Cases for ALTO within CDNs", draft-jenkins-alto-cdn-use-cases-01 (work in progress), June 2011.

[IJNM.unfairness]
Lehrieder, F., Oechsner, S., Hossfeld, T., Staehle, D., Despotovic, Z., Kellerer, W., and M. Michel, "Mitigating unfairness in locality-aware peer-to-peer networks", International Journal of Network Management, Volume 21, Issue 1, pp. 3-20, January/February 2011.

Authors' Addresses

Zbigniew Dulinski
Jagiellonian University
ul. Reymonta 4
Krakow 30-059
Poland

Phone: +48 12 663 5571
Fax: +48 12 633 4079
Email: dulinski@th.if.uj.edu.pl

Piotr Wydrych
AGH University of Science and Technology
al. Mickiewicza 30
Krakow 30-059
Poland

Phone: +48 12 617 4817
Fax: +48 12 634 2372
Email: piotr.wydrych@agh.edu.pl

Rafal Stankiewicz
AGH University of Science and Technology
al. Mickiewicza 30
Krakow 30-059
Poland

Phone: +48 12 617 4036
Fax: +48 12 634 2372
Email: rstankie@agh.edu.pl

ALTO
Internet-Draft
Intended status: Informational
Expires: September 15, 2011

M. Stiemerling
NEC Europe Ltd.
S. Kiesel
University of Stuttgart
March 14, 2011

ALTO Deployment Considerations
draft-ietf-alto-deployments-01

Abstract

Many Internet applications are used to access resources, such as pieces of information or server processes, which are available in several equivalent replicas on different hosts. This includes, but is not limited to, peer-to-peer file sharing applications. The goal of Application-Layer Traffic Optimization (ALTO) is to provide guidance to these applications, which have to select one or several hosts from a set of candidates, that are able to provide a desired resource. The protocol is under specification in the ALTO working group. This memo discusses deployment related issues of ALTO for peer-to-peer and CDNs, some preliminary security considerations, and also initial guidance for application designers using ALTO.

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 September 15, 2011.

Table of Contents

1. Introduction

Many Internet applications are used to access resources, such as pieces of information or server processes, which are available in several equivalent replicas on different hosts. This includes, but is not limited to, peer-to-peer file sharing applications and Content Delivery Networks (CDNs). The goal of Application-Layer Traffic Optimization (ALTO) is to provide guidance to applications, which have to select one or several hosts from a set of candidates, that are able to provide a desired resource. The basic ideas of ALTO are described in the problem space of ALTO is described in [RFC5693] and the set of requirements is discussed in [I-D.ietf-alto-reqs].

However, there are no considerations about what operational issues are to be expected once ALTO will be deployed. This includes, but is not limited to, location of the ALTO server, imposed load to the ALTO server, or from whom the queries are performed.

Comments and discussions about this memo should be directed to the ALTO working group: alto@ietf.org.

2. General Considerations

The ALTO protocol is a client/server protocol, operating between a number of ALTO clients and an ALTO server, as sketched in Figure 1. The ALTO working groups defines the ALTO protocol [I-D.ietf-alto-protocol].

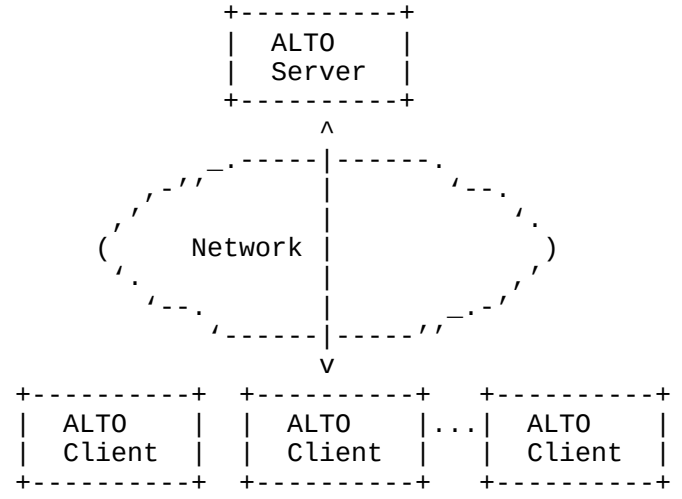


Figure 1: Network Overview of ALTO Protocol

2.1. General Placement of ALTO

The ALTO server and ALTO clients can be situated at various entities in a network deployment. The first differentiation is whether the ALTO client is located on the actual host that runs the application, as shown in Figure 2, (e.g., peer-to-peer filesharing application) or if the ALTO client is located on resource directory, as shown in Figure 3 (e.g., a tracker in peer-to-peer filesharing).

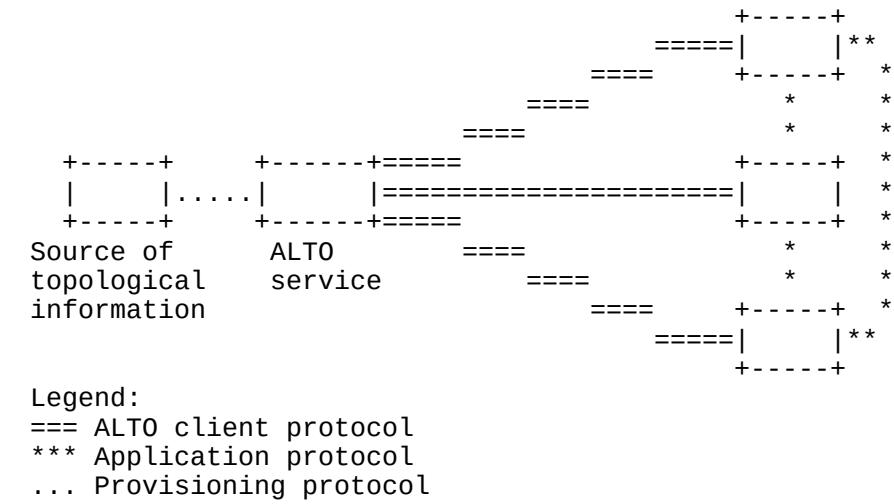


Figure 2: Overview of protocol interaction between ALTO elements,scenario without tracker

Figure 2 shows the operational model for applications that do not use a tracker, such as, edonky, or in if the tracker should be the querying party. This use case also holds true for CDNs. The ALTO server can also be queried by CDNs to get a guidance about where the a particular client accessing data in the CDN is exactly located in the ISP's network.

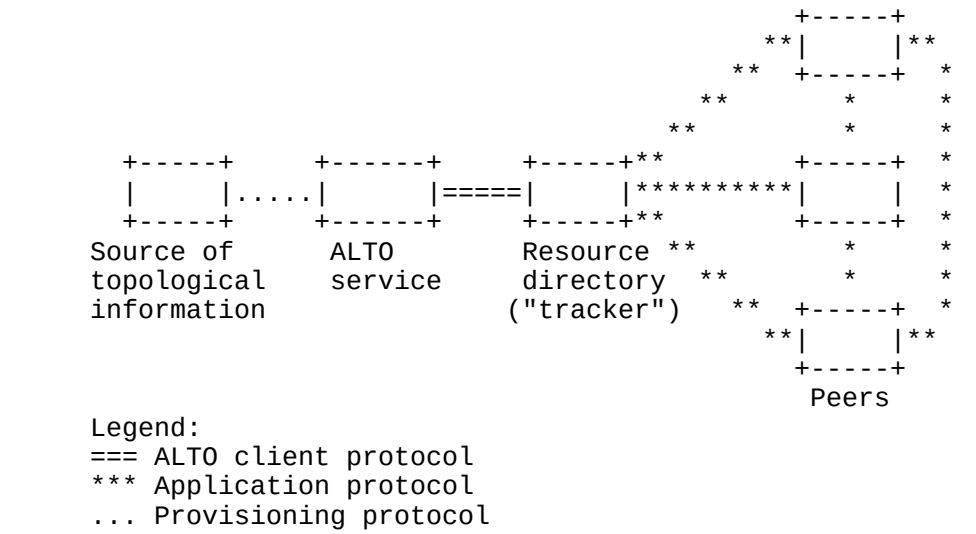


Figure 3: Overview of protocol interaction between ALTO elements, scenario with tracker

However, Figure 3 does not denote where the ALTO elements are actually located, i.e., if the tracker and the ALTO server are in the same ISP's domain, or if the tracker and the ALTO server are managed/ owned/located in different domains. The latter is the typical use case, e.g., taking Pirate Bay as example that serves Bittorrent peers world-wide.

2.2. Relationship between ALTO and Applications

ALTO is intended to be used by a wide-range of applications. However, any application using ALTO must also work if no ALTO servers can be found or if no responses to ALTO queries are received, e.g., due to connectivity problems or overload situation (see also [I-D.ietf-alto-reqs]). (Editor's note: better text needed here!)

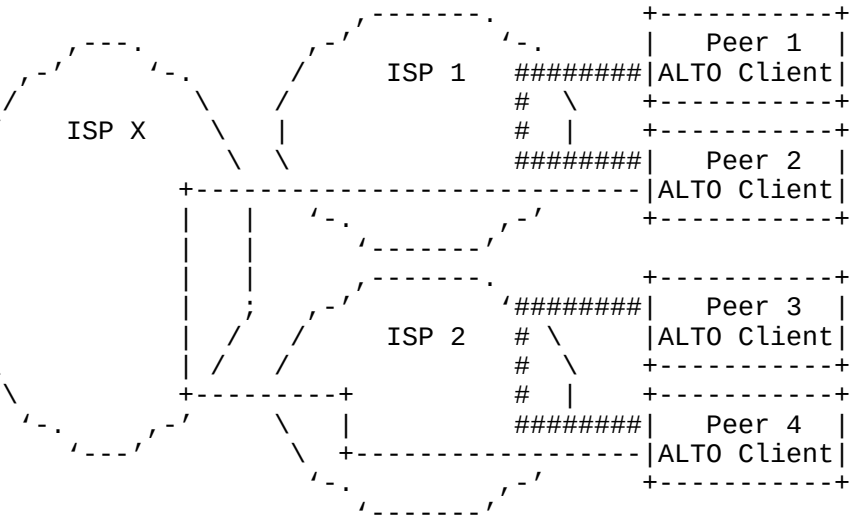
2.3. Provided Guidance

ALTO gives guidance to applications on what IP addresses or IP prefixes, and such which hosts are to be preferred according to the operator of the ALTO server. The general assumption of the ALTO WG is that a network operator would always express to prefer hosts in its own network while hosts located outside its own network are to be avoided (are undesired to be considered by the applications). This might be applicable in some cases but may not be applicable in the general case. The ALTO protocol gives only the means to let the ALTO server operator to express is preference, whatever this preference

is. This section explores this space.

2.3.1. Keeping Traffic Local in Network

ALTO guidance can be used to let applications prefer other peers within the same network operator's network instead of randomly connecting to other peers which are located in another operator's network. Figure 4 shows such a scenario where peers prefer peers in the same network (e.g., Peer 1 and Peer 2 in ISP1 and Peer 3 and Peer 4 in ISP2).



Legend:
preferred "connections"
--- non-preferred "connections"

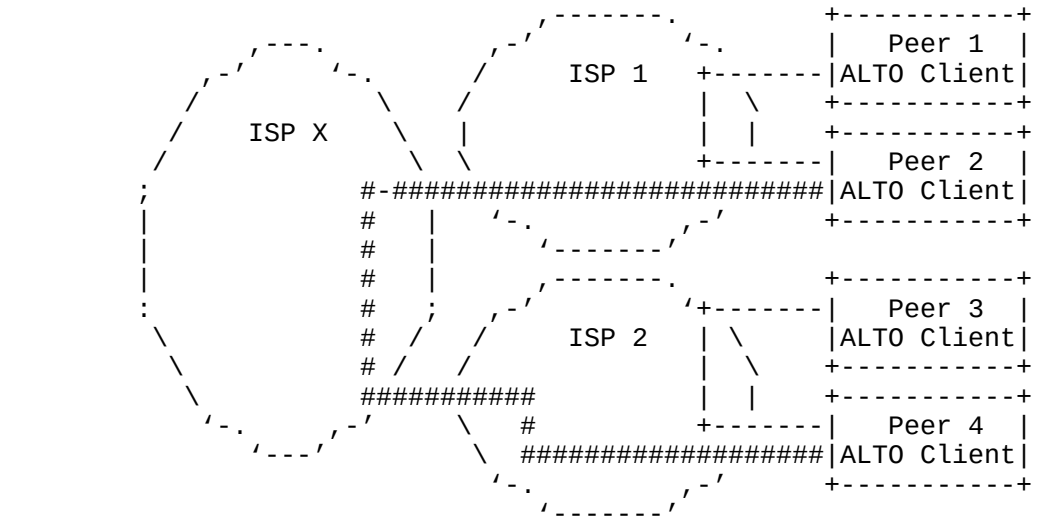
Figure 4: ALTO Traffic Network Localization

TBD: Describes limits of this approach (e.g., traffic localization guidance is of less use if the peers cannot upload); describe how maps would look like.

2.3.2. Off-Loading Traffic from Network

Another scenario where the use of ALTO can be beneficial is in mobile broadband networks, e.g., CDMA200 or UMTS, but where the network operator may have the desire to guide peers in its own network to use peers in remote networks. One reason can be that the wireless network is not made for the load cause by, e.g., peer-to-peer

applications, and the operator has the need that peers fetch their data from remote peers in other parts of the Internet.



Legend:
=== preferred "connections"
--- non-preferred "connections"

Figure 5: ALTO Traffic Network De-Localization

Figure 5 shows the result of such a guidance process where Peer 2 prefers a connection with Peer4 instead of Peer 1, as shown in Figure 4.

TBD: Limits of this approach in general and with respect to p2p. describe how maps would look like.

2.3.3. Intra-Network Localization/Bottleneck Off-Loading

The above sections described the results of the ALTO guidance on an inter-network level. However, ALTO can also be used to guide peers on which internal peers are to be preferred. For instance, to guide Peers on a remote network side to prefer to connect to each other, instead of crossing a bottleneck link, a backhaul link to connect the side to the network core. Figure 6 shows such a scenario where Peer 1 and Peer 2 are located in Net 2 of ISP1 and connect via a low capacity link to the core (Net 1) of the same ISP1. Peer1 and Peer 2 would both exchange their data with remote peers, probably clogging the bottleneck link.

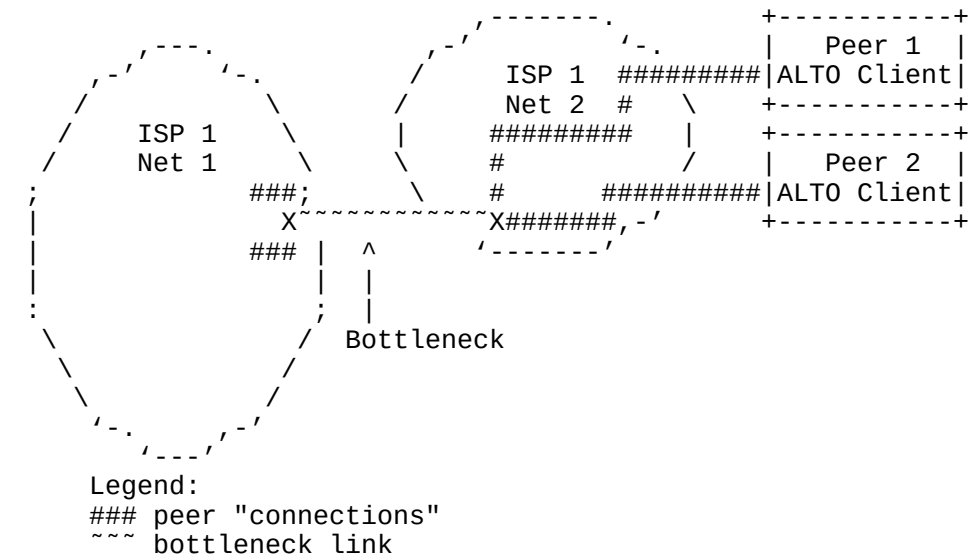


Figure 6: Without Intra-Network ALTO Traffic Localization

The operator can guide the peers in such a situation to try first local peers in the same network islands, avoiding or at least lowering the effect on the bottleneck link, as shown in Figure 7.

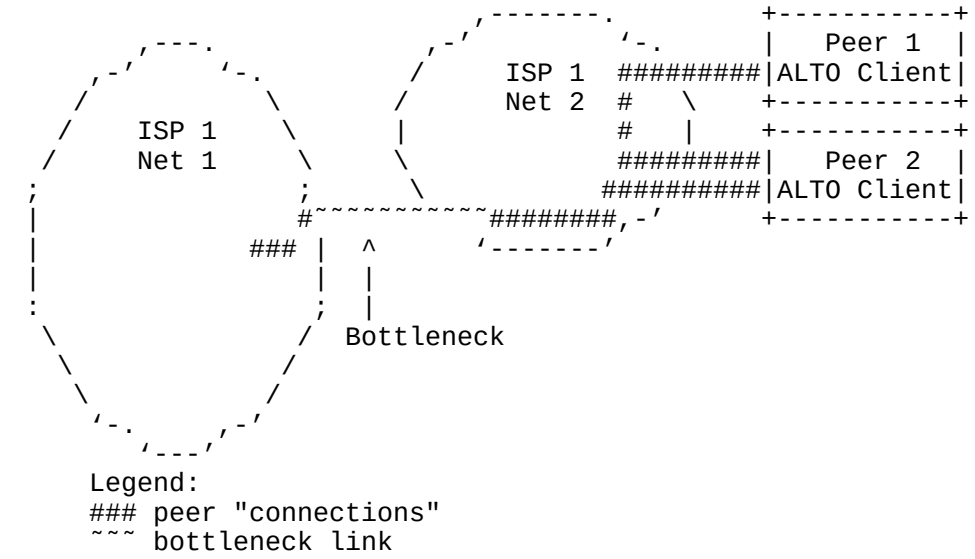


Figure 7: With Intra-Network ALTO Traffic Localization

TBD: describe how maps would look like.

2.4. Provisiong ALTO Maps

This section will describe how ALTO maps in the protocol can be populated before using them.

3. Using ALTO for P2P

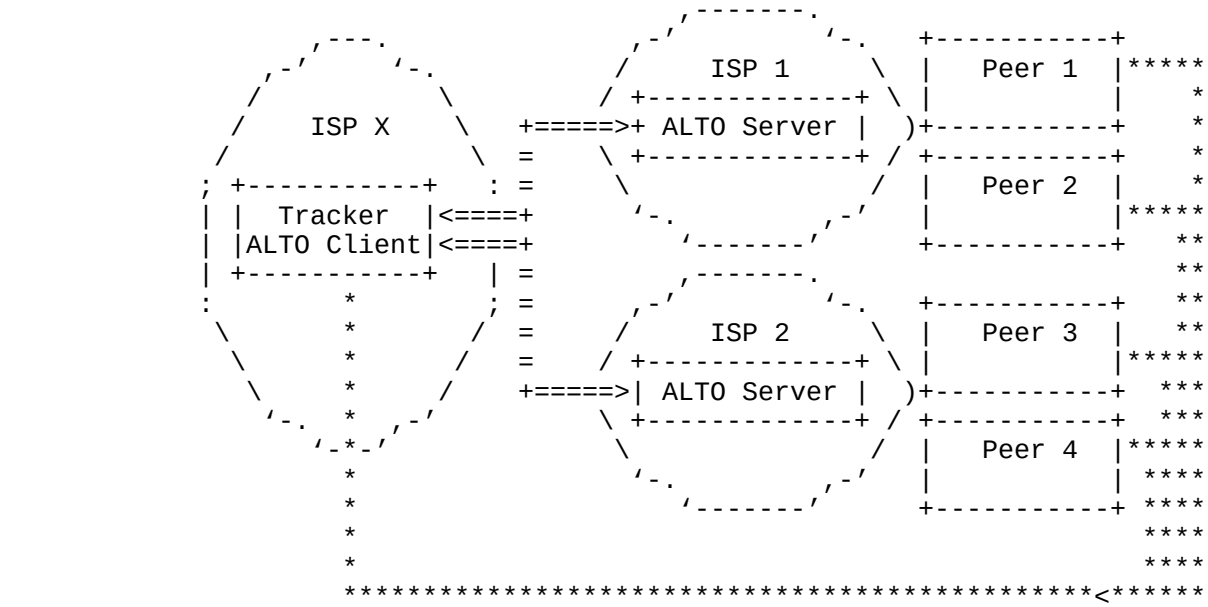


Figure 8: Global tracker accessing ALTO server at various ISPs

Figure 8 depicts a tracker-based system, where the tracker embeds the ALTO client. The tracker itself is hosted and operated by an entity different than the ISP hosting and operating the ALTO server. Initially, the tracker has to look-up the ALTO server in charge for each peer where it receives a ALTO query for. Therefore, the ALTO server has to discover the handling ALTO server, as described in [I-D.kiesel-alto-3pdisc]. However, the peers do not have any way to query the server themselves. This setting allows to give the peers a better selection of candidate peers for their operation at an initial time, but does not consider peers learned through direct peer-to-peer knowledge exchange, AKA peer exchange in various peer-to-peer protocols.

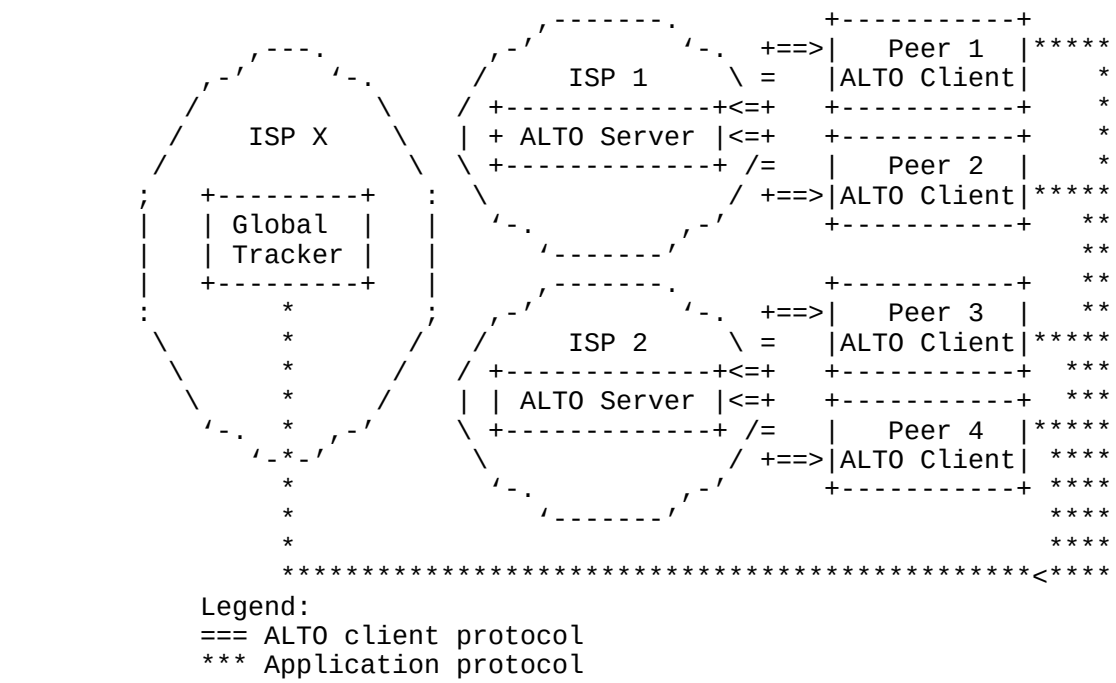


Figure 9: Global Tracker - Local ALTO Servers

The scenario in Figure 9 lets the peers directly communicate with their ISP's ALTO server (i.e., ALTO client embedded in the peers), giving thus the peers the most control on which information they query for, as they can integrate information received from trackers and through direct peer-to-peer knowledge exchange.

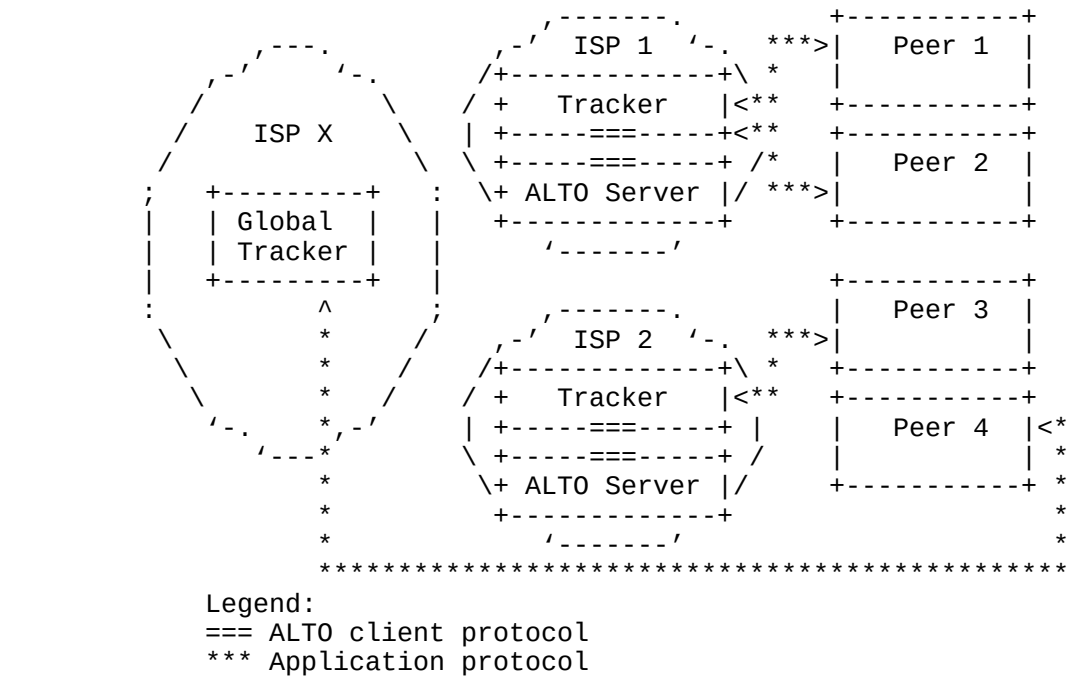


Figure 10: P4P approach with local tracker and local ALTO server

There are some attempts to let ISP's to deploy their own trackers, as shown in Figure 10. In this case, the client has no chance to get guidance from the ALTO server, other than talking to the ISP's tracker. However, the peers would have still chance the contact other trackers, deployed by entities other than the peer's ISP.

Figure 10 and Figure 8 ostensibly take peers the possibility to directly query the ALTO server, if the communication with the ALTO server is not permitted for any reason. However, considering the plethora of different applications of ALTO, e.g., multiple tracker and non-tracker based P2P systems and or applications searching for relays, it seems to be beneficial for all participants to let the peers directly query the ALTO server. The peers are also the single point having all operational knowledge to decide whether to use the ALTO guidance and how to use the ALTO guidance. This is a preference for the scenario depicted in Figure Figure 9.

3.1. Using ALTO for Tracker-based Peer-to-Peer Applications

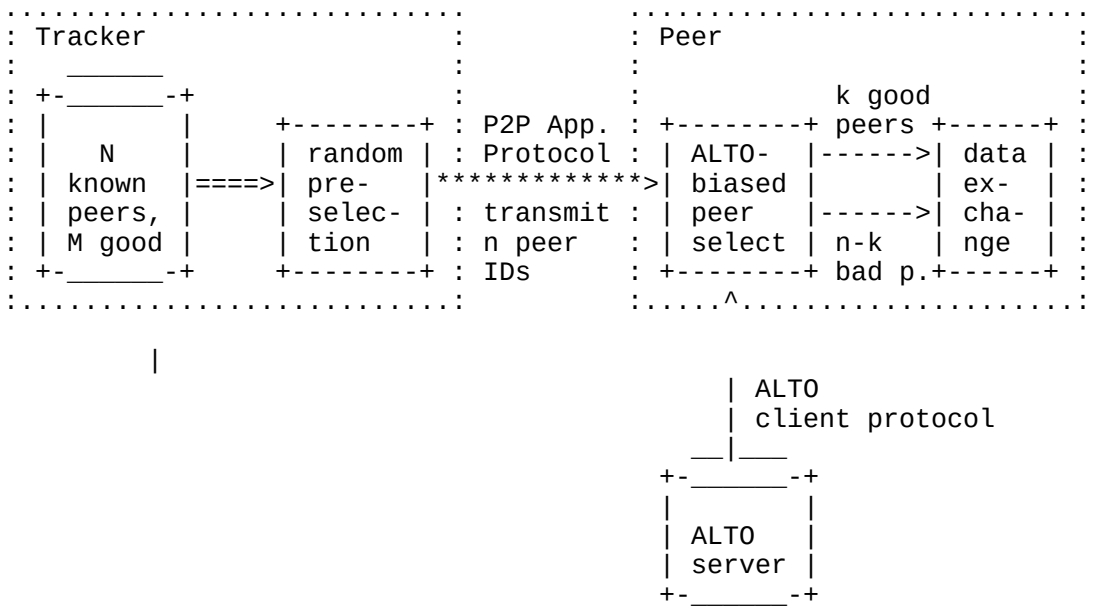


Figure 11: Tracker-based P2P Application with random peer preselection

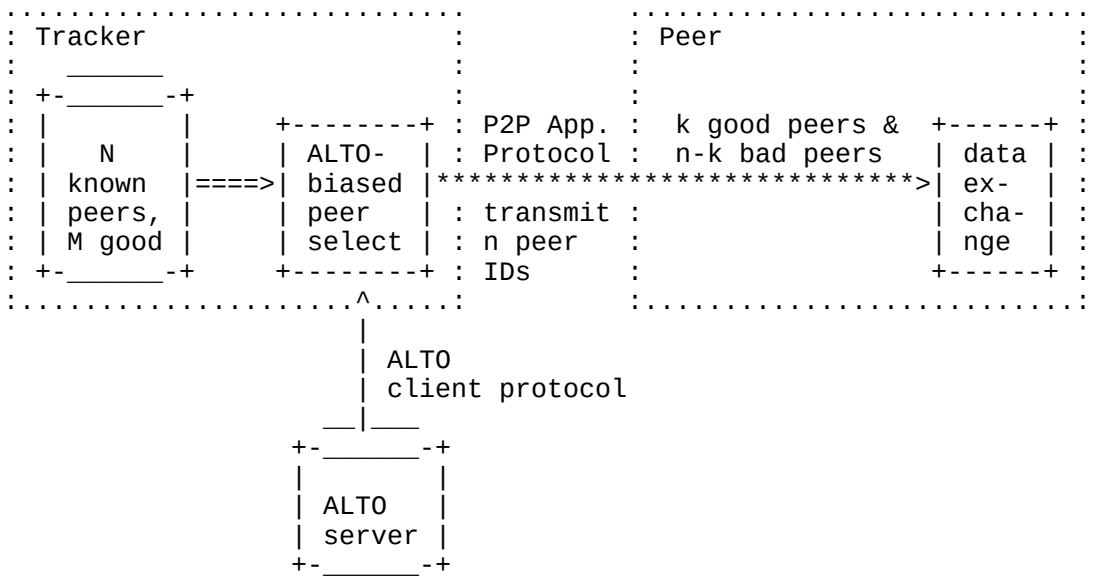


Figure 12: Tracker-based P2P Application with ALTO client in tracker

TBD: explain why Figure 12 usually will yield better results wrt. peer selection than Figure 11.

3.2. Expectations of ALTO

This section hints to some recent experiments conducted with ALTO-like deployments in Internet Service Provider (ISP) network's. NTT performed tests with their HINT server implementation and dummy nodes to gain insight on how an ALTO-like service influence a peer-to-peer systems [I-D.kamei-p2p-experiments-japan]. The results of an early experiment conducted in the Comcast network are documented here[RFC5632]

4. Using ALTO for CDNs

Section 2 discussed the placement and usage of ALTO for P2P systems, but not beyond. This section discuss the usage of ALTO for Content Delivery Networks (CDNs). CDNs are used to bring a service (e.g., a web page, videos, etc) closer to the location of the user - where close refers to shorten the distance between the client and the server in the IP topology. CDNs use several techniques to decide which server is closest to a client requesting a service. One common way to do so, is relying on the DNS system, but there are many other ways, see [RFC3568].

The general issue for CDNs, independent of DNS or HTTP Redirect based approaches (see, for instance, [I-D.penno-alto-cdn]), is that the CDN logic has to match the client’s IP address with the closest CDN cache. This matching is not trivial, for instance, in DNS based approaches, where the IP address of the DNS original requester is unknown (see [I-D.vandergaast-edns-client-ip] for a discussion of this and a solution approach).

5. Advanced Features

5.1. Cascading ALTO Servers

The main assumptions of ALTO seems to be each ISP operates its own ALTO server independently, irrespectively of the ISP's situation. This may true for most envisioned deployments of ALTO but there are certain deployments that may have different settings. Figure 13 shows such setting, were for example, a university network is connected to two upstream providers. ISP2 if the national research network and ISP1 is a commercial upstream provider to this university network. The university, as well as ISP1, are operating their own ALTO server. The ALTO clients, located on the peers will contact the ALTO server located at the university.

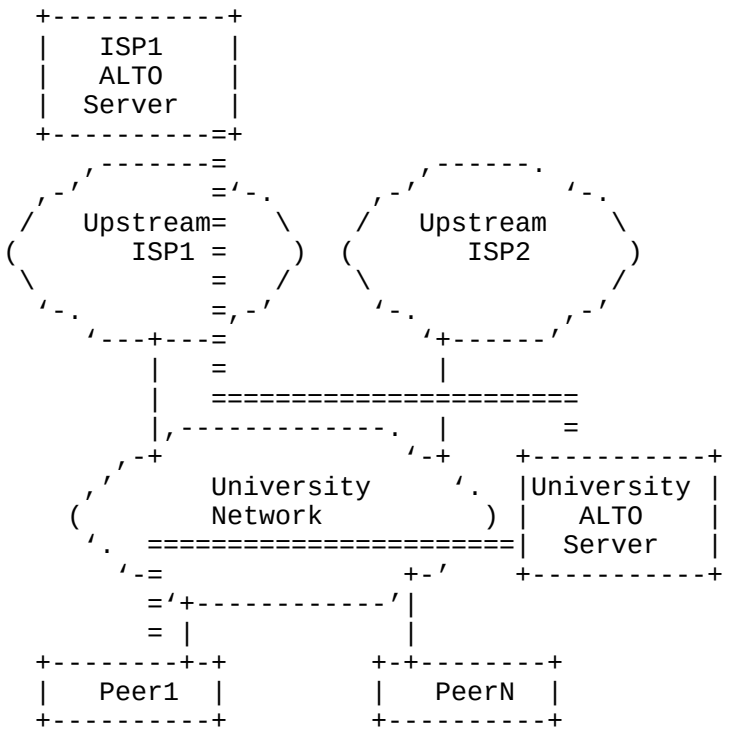


Figure 13: Cascaded ALTO Server

In this setting all "destinations" useful for the peers within ISP2 are free-of-charge for the peers located in the university network (i.e., they are preferred in the rating of the ALTO server). However, all traffic that is not towards ISP2 will be handled by the

ISP1 upstream provider. Therefore, the ALTO server at the university has also to include the guidance given by the ISP1 ALTO server in its replies to the ALTO clients. This can be called cascaded ALTO servers.

5.2. ALTO for IPv4 and IPv6

TBD

5.3. Monitoring ALTO

TBD

6. Known Limitations of ALTO

This section describes some known limitations of ALTO in general or specific mechanisms in ALTO.

6.1. Limitations of Map-based Approaches

The specification of the ALTO protocol [I-D.ietf-alto-protocol] uses, amongst others mechanism, so-called network maps. The network map approach uses Host Group Descriptors that group one or multiple subnetworks (i.e., IP prefixes) to a single Host Group Descriptor. A set of IP prefixes is called partition and the associated Host Group Descriptor is called partition ID. The "costs" between the various partition IDs is stored in a second map, the cost map. Map-based approaches are chosen as they lower the signaling load on the server, as the maps have only to be retrieved if they are changed.

The main assumption for map-based approaches is that the information provided in these maps is static for a longer period of time, where this period of time refers to days, but not hours or even minutes. This assumption is fine, as long as the network operator does not change any parameter, e.g., routing within the network and to the upstream peers, IP address assignment stays stable (and thus the mapping to the partitions). However, there are several cases where this assumption is not valid, as:

1. ISPs reallocate IPv4 subnets from time to time;

2. ISPs reallocate IPv4 subnets on short notice;

3. IP prefix blocks may be assigned to a single DSLAM which serves a variety of access networks.

For 1): ISPs reallocate IPv4 subnets within their infrastructure from time to time, partly to ensure the efficient usage of IPv4 addresses (a scarce resource), and partly to enable efficient route tables within their network routers. The frequency of these "renumbering events" depend on the growth in number of subscribers and the availability of address space within the ISP. As a result, a subscriber’s household device could retain an IPv4 address for as short as a few minutes, or for months at a time or even longer.

> Some folks have suggested that ISPs providing ALTO services could sub-divide their subscribers’ devices into different IPv4 subnets (or certain IPv4 address ranges) based on the purchased service tier, as well as based on the location in the network topology. The problem is that this sub-allocation of IPv4 subnets tends to decrease the efficiency of IPv4 address allocation. A growing ISP

that needs to maintain high efficiency of IPv4 address utilization may be reluctant to jeopardize their future acquisition of IPv4 address space.

However, this is not an issue for map-based approaches if changes are applied in the order of days.

For 2): ISPs can use techniques, such as ODAP (XXX) that allow the reallocation of IP prefixes on very short notice, i.e., within minutes. An IP prefix that has no IP address assignment to a host anymore can be reallocate to areas where there is currently a high demand for IP addresses.

For 3): In DSL-based access networks, IP prefixes are assigned to DSLAMs which are the first IP-hop in the access-network between the CPE and the Internet. The access-network between CPE and DSLAM (called aggregation network) can have varying characteristics (and thus associated costs), but still using the same IP prefix. For instance one IP addresses IP11 out of a IP prefix IP1 can be assigned to a VDSL (e.g., 2 MBit/s uplink) access-line while the subsequent IP address IP12 is assigned to a slow ADSL line (e.g., 128 kbit/s uplink). These IP addresses are assigned on a first come first served basis, i.e., the a single IP address out of the same IP prefix can change its associated costs quite fast. This may not be an issue with respect to the used upstream provider (thus the cross ISP traffic) but depending on the capacity of the aggregation-network this may raise to an issue.

6.2. Limitiations of Non-Map-based Approaches

The specification of the ALTO protocol [I-D.ietf-alto-protocol] uses, amongst others mechanism, a mechanism called Endpoint Cost Service. ALTO clients can ask guidance for specific IP addresses to the ALTO server. However, asking for IP addresses, asking with long lists of IP addresses, and asking quite frequent may overload the ALTO server. The server has to rank each received IP address which causes load at the server. This may be amplified by the fact that not only a single ALTO client is asking for guidance, but a larger number of them.

Caching of IP addresses at the ALTO client or the usage of the H12 approach [I-D.kiesel-alto-h12] in conjunction with caching may lower the query load on the ALTO server.

6.3. General Challenges

An ALTO server stores information about preferences (e.g., a list of preferred autonomous systems, IP ranges, etc) and ALTO clients can retrieve these preferences. However, there are basically two

different approaches on where the preferences are actually processed:

1. The ALTO server has a list of preferences and clients can retrieve this list via the ALTO protocol. This preference list can be partially updated by the server. The actual processing of the data is done on the client and thus there is no data of the client's operation revealed to the ALTO server .

2. The ALTO server has a list of preferences or preferences calculated during runtime and the ALTO client is sending information of its operation (e.g., a list of IP addresses) to the server. The server is using this operational information to determine its preferences and returns these preferences (e.g., a sorted list of the IP addresses) back to the ALTO client.

Approach 1 (we call it H1) has the advantage (seen from the client) that all operational information stays within the client and is not revealed to the provider of the server. On the other hand, does approach 1 require that the provider of the ALTO server, i.e., the network operator, reveals information about its network structure (e.g., AS numbers, IP ranges, topology information in general) to the ALTO client.

Approach 2 (we call it H2) has the advantage (seen from the operator) that all operational information stays with the ALTO server and is not revealed to the ALTO client. On the other hand, does approach 2 require that the clients send their operational information to the server.

Both approaches have their pros and cons and are extensively discussed on the ALTO mailing list. But there is basically a dilemma: Approach 1 is seen as the only working solution by peer-to-peer software vendors and approach 2 is seen as the only working by the network operators. But neither the software vendors nor the operators seem to willing to change their position. However, there is the need to get both sides on board, to come to a solution.

7. Extensions to the ALTO Protocol

7.1. Host Group Descriptors

Host group descriptors are used in the ALTO client protocol to describe the location of a host in the network topology. The ALTO client protocol specification defines a basic set of host group descriptor types, which have to be supported by all implementations, and an extension procedure for adding new descriptor types . The following list gives an overview on further host group descriptor types that have been proposed in the past, or which are in use by ALTO-related prototype implementations. This list is not intended as normative text. Instead, the only purpose of the following list is to document the descriptor types that have been proposed so far, and to solicit further feedback and discussion:

- Autonomous System (AS) number
- Protocol-specific group identifiers, which expand to a set of IP address ranges (CIDR) and/or AS numbers. In one specific solution proposal, these are called Partition ID (PID).

7.2. Rating Criteria

Rating criteria are used in the ALTO client protocol to express topology- or connectivity-related properties, which are evaluated in order to generate the ALTO guidance. The ALTO client protocol specification defines a basic set of rating criteria, which have to be supported by all implementations, and an extension procedure for adding new criteria . The following list gives an overview on further rating criteria that have been proposed in the past, or which are in use by ALTO-related prototype implementations. This list is not intended as normative text. Instead, the only purpose of the following list is to document the rating criteria that have been proposed so far, and to solicit further feedback and discussion:

7.2.1. Distance-related Rating Criteria

- Relative topological distance: relative means that a larger numerical value means greater distance, but it is up to the ALTO service how to compute the values, and the ALTO client will not be informed about the nature of the information. One way of generating this kind of information MAY be counting AS hops, but when querying this parameter, the ALTO client MUST NOT assume that the numbers actually are AS hops.
- Absolute topological distance, expressed in the number of traversed autonomous systems (AS).

- Absolute topological distance, expressed in the number of router hops (i.e., how much the TTL value of an IP packet will be decreased during transit).

- Absolute physical distance, based on knowledge of the approximate geolocation (continent, country) of an IP address.

7.2.2. Charging-related Rating Criteria

- Traffic volume caps, in case the Internet access of the resource consumer is not charged by "flat rate". For each candidate resource provider, the ALTO service could indicate the amount of data that may be transferred from/to this resource provider until a given point in time, and how much of this amount has already been consumed. Furthermore, it would have to be indicated how excess traffic would be handled (e.g., blocked, throttled, or charged separately at an indicated price). The interaction of several applications running on a host, out of which some use this criterion while others don't, as well as the evaluation of this criterion in resource directories, which issue ALTO queries on behalf of other peers, are for further study.

7.2.3. Performance-related Rating Criteria

The following rating criteria are subject to the remarks below.

- The minimum achievable throughput between the resource consumer and the candidate resource provider, which is considered useful by the application (only in ALTO queries), or

- An arbitrary upper bound for the throughput from/to the candidate resource provider (only in ALTO responses). This may be, but is not necessarily the provisioned access bandwidth of the candidate resource provider.

- The maximum round-trip time (RTT) between resource consumer and the candidate resource provider, which is acceptable for the application for useful communication with the candidate resource provider (only in ALTO queries), or

- An arbitrary lower bound for the RTT between resource consumer and the candidate resource provider (only in ALTO responses). This may be, for example, based on measurements of the propagation delay in a completely unloaded network.

The ALTO client MUST be aware, that with high probability, the actual performance values differ significantly from these upper and lower bounds. In particular, an ALTO client MUST NOT consider the "upper

bound for throughput" parameter as a permission to send data at the indicated rate without using congestion control mechanisms.

The discrepancies are due to various reasons, including, but not limited to the facts that

- the ALTO service is not an admission control system
- the ALTO service may not know the instantaneous congestion status of the network
- the ALTO service may not know all link bandwidths, i.e., where the bottleneck really is, and there may be shared bottlenecks
- the ALTO service may not know whether the candidate peer itself is overloaded
- the ALTO service may not know whether the candidate peer throttles the bandwidth it devotes for the considered application
- the ALTO service may not know whether the candidate peer will throttle the data it sends to us (e.g., because of some fairness algorithm, such as tit-for-tat)

Because of these inaccuracies and the lack of complete, instantaneous state information, which are inherent to the ALTO service, the application must use other mechanisms (such as passive measurements on actual data transmissions) to assess the currently achievable throughput, and it MUST use appropriate congestion control mechanisms in order to avoid a congestion collapse. Nevertheless, these rating criteria may provide a useful shortcut for quickly excluding candidate resource providers from such probing, if it is known in advance that connectivity is in any case worse than what is considered the minimum useful value by the respective application.

7.2.4. Inappropriate Rating Criteria

Rating criteria that SHOULD NOT be defined for and used by the ALTO service include:

- Performance metrics that are closely related to the instantaneous congestion status. The definition of alternate approaches for congestion control is explicitly out of the scope of ALTO. Instead, other appropriate means, such as using TCP based transport, have to be used to avoid congestion.

8. API between ALTO Client and Application

This sections gives some informational guidance on how the interface between the actual application using the ALTO guidance and the ALTO client can look like.

This is still TBD.

9. Security Considerations

The ALTO protocol itself, as well as, the ALTO client and server raise new security issues beyond the one mentioned in [I-D.ietf-alto-protocol] and issues related to message transport over the Internet. For instance, Denial of Service (DoS) is of interest for the ALTO server and also for the ALTO client. A server can get overloaded if too many TCP requests hit the server, or if the query load of the server surpasses the maximum computing capacity. An ALTO client can get overloaded if the responses from the sever are, either intentionally or due to an implementation mistake, too large to be handled by that particular client.

9.1. Information Leakage from the ALTO Server

The ALTO server will be provisioned with information about the owning ISP's network and very likely also with information about neighboring ISPs. This information (e.g., network topology, business relations, etc) is consider to be confidential to the ISP and must not be revealed.

The ALTO server will naturally reveal parts of that information in small doses to peers, as the guidance given will depend on the above mentioned information. This is seen beneficial for both parties, i.e., the ISP's and the peer's. However, there is the chance that one or multiple peers are querying an ALTO server with the goal to gather information about network topology or any other data considered confidential or at least sensitive. It is unclear whether this is a real technical security risk or whether this is more a perceived security risk.

9.2. ALTO Server Access

Depending on the use case of ALTO, several access restrictions to an ALTO server may or may not apply. For an ALTO server that is solely accessible by peers from the ISP network (as shown in Figure 9), for instance, the source IP address can be used to grant only access from that ISP network to the server. This will "limit" the number of peers able to attack the server to the user's of the ISP (however, including botnet computers).

On the other hand, if the ALTO server has to be accessible by parties not located in the ISP's network (see Figure Figure 8), e.g., by a third-party tracker or by a CDN system outside the ISP's network, the access restrictions have to be more loose. In the extreme case, i.e., no access restrictions, each and every host in the Internet can access the ALTO server. This might no the intention of the ISP, as the server is not only subject to more possible attacks, but also on

the load imposed to the server, i.e., possibly more ALTO clients to serve and thus more work load.

9.3. Faking ALTO Guidance

It has not yet been investigated how a faked or wrong ALTO guidance by an ALTO server can impact the operation of the network and also the peers.

Here is a list of examples how the ALTO guidance could be faked and what possible consequences may arise:

Sorting An attacker could change to sorting order of the ALTO guidance (given that the order is of importance, otherwise the ranking mechanism is of interest), i.e., declaring peers located outside the ISP as peers to be preferred. This will not pose a big risk to the network or peers, as it would mimic the "regular" peer operation without traffic localization, apart from the communication/processing overhead for ALTO. However, it could mean that ALTO is reaching the opposite goal of shuffling more data across ISP boundaries, incurring more costs for the ISP.

Preference of a single peer A single IP address (thus a peer) could be marked as to be preferred all over other peers. This peer can be located within the local ISP or also in other parts of the Internet (e.g., a web server). This could lead to the case that quite a number of peers to trying to contact this IP address, possibly causing a Denial of Service (DoS) attack.

This section is solely giving a first shot on security issues related to ALTO deployments.

10. Conclusion

This is the first version of the deployment considerations and for sure the considerations are yet incomplete and imprecise.

11. References

11.1. Normative References

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

[RFC3568] Barbir, A., Cain, B., Nair, R., and O. Spatscheck, "Known Content Network (CN) Request-Routing Mechanisms", RFC 3568, July 2003.

11.2. Informative References

[I-D.ietf-alto-protocol]
Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-06 (work in progress), October 2010.

[I-D.ietf-alto-reqs]
Previdi, S., Stiemerling, M., Woundy, R., and Y. Yang, "Application-Layer Traffic Optimization (ALTO) Requirements", draft-ietf-alto-reqs-08 (work in progress), March 2011.

[I-D.kamei-p2p-experiments-japan]
Kamei, S., Momose, T., and T. Inoue, "ALTO-Like Activities and Experiments in P2P Network Experiment Council", draft-kamei-p2p-experiments-japan-05 (work in progress), March 2011.

[I-D.kiesel-alto-3pdisc]
Kiesel, S., Tomsu, M., Schwan, N., Scharf, M., and M. Stiemerling, "ALTO Server Discovery Protocol", draft-kiesel-alto-3pdisc-04 (work in progress), October 2010.

[I-D.kiesel-alto-h12]
Kiesel, S. and M. Stiemerling, "ALTO H12", draft-kiesel-alto-h12-02 (work in progress), March 2010.

[I-D.penno-alto-cdn]
Penno, R., Raghunath, S., Medved, J., Alimi, R., Yang, R., and S. Previdi, "ALTO and Content Delivery Networks", draft-penno-alto-cdn-02 (work in progress), October 2010.

[I-D.vandergaast-edns-client-ip]
Contavalli, C., Gaast, W., Leach, S., and D. Rodden, "Client IP information in DNS requests",

draft-vandergaast-edns-client-ip-01 (work in progress), May 2010.

[RFC5632] Griffiths, C., Livingood, J., Popkin, L., Woundy, R., and Y. Yang, "Comcast's ISP Experiences in a Proactive Network Provider Participation for P2P (P4P) Technical Trial", RFC 5632, September 2009.

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

Appendix A. Acknowledgments

Martin Stiemerling is partially supported by the NAPA-WINE project (Network-Aware P2P-TV Application over Wise Networks, http://www.napa-wine.org), a research project supported by the European Commission under its 7th Framework Program (contract no. 214412). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the NAPA-WINE project or the European Commission.

Authors' Addresses

Martin Stiemerling
NEC Laboratories Europe
Kurfuerstenanlage 36
Heidelberg 69115
Germany

Phone: +49 6221 4342 113
Fax: +49 6221 4342 155
Email: martin.stiemerling@neclab.eu
URI: http://ietf.stiemerling.org

Sebastian Kiesel
University of Stuttgart, Computing Center
Allmandring 30
Stuttgart 70550
Germany

Email: ietf-alto@skiesel.de

ALTO
Internet-Draft
Intended status: Informational
Expires: January 12, 2012

M. Stiemerling
NEC Europe Ltd.
S. Kiesel
University of Stuttgart
July 11, 2011

ALTO Deployment Considerations
draft-ietf-alto-deployments-02

Abstract

Many Internet applications are used to access resources, such as pieces of information or server processes, which are available in several equivalent replicas on different hosts. This includes, but is not limited to, peer-to-peer file sharing applications. The goal of Application-Layer Traffic Optimization (ALTO) is to provide guidance to these applications, which have to select one or several hosts from a set of candidates, that are able to provide a desired resource. The protocol is under specification in the ALTO working group. This memo discusses deployment related issues of ALTO for peer-to-peer and CDNs, some preliminary security considerations, and also initial guidance for application designers using ALTO.

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 January 12, 2012.

Table of Contents

1. Introduction

Many Internet applications are used to access resources, such as pieces of information or server processes, which are available in several equivalent replicas on different hosts. This includes, but is not limited to, peer-to-peer file sharing applications and Content Delivery Networks (CDNs). The goal of Application-Layer Traffic Optimization (ALTO) is to provide guidance to applications, which have to select one or several hosts from a set of candidates, that are able to provide a desired resource. The basic ideas of ALTO are described in the problem space of ALTO is described in [RFC5693] and the set of requirements is discussed in [I-D.ietf-alto-reqs].

However, there are no considerations about what operational issues are to be expected once ALTO will be deployed. This includes, but is not limited to, location of the ALTO server, imposed load to the ALTO server, or from whom the queries are performed.

Comments and discussions about this memo should be directed to the ALTO working group: alto@ietf.org.

2. General Considerations

The ALTO protocol is a client/server protocol, operating between a number of ALTO clients and an ALTO server, as sketched in Figure 1. The ALTO working groups defines the ALTO protocol [I-D.ietf-alto-protocol].

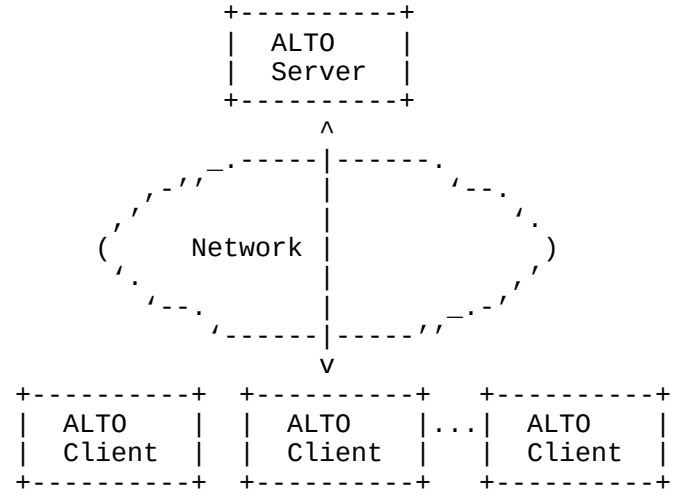


Figure 1: Network Overview of ALTO Protocol

2.1. General Placement of ALTO

The ALTO server and ALTO clients can be situated at various entities in a network deployment. The first differentiation is whether the ALTO client is located on the actual host that runs the application, as shown in Figure 2, (e.g., peer-to-peer filesharing application) or if the ALTO client is located on resource directory, as shown in Figure 3 (e.g., a tracker in peer-to-peer filesharing).

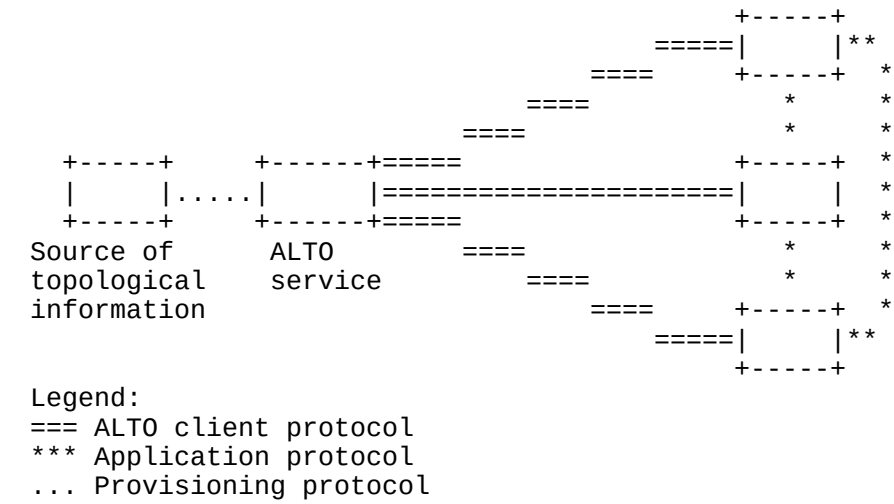


Figure 2: Overview of protocol interaction between ALTO elements,scenario without tracker

Figure 2 shows the operational model for applications that do not use a tracker, such as, edonky, or in if the tracker should be the querying party. This use case also holds true for CDNs. The ALTO server can also be queried by CDNs to get a guidance about where the a particular client accessing data in the CDN is exactly located in the ISP's network.

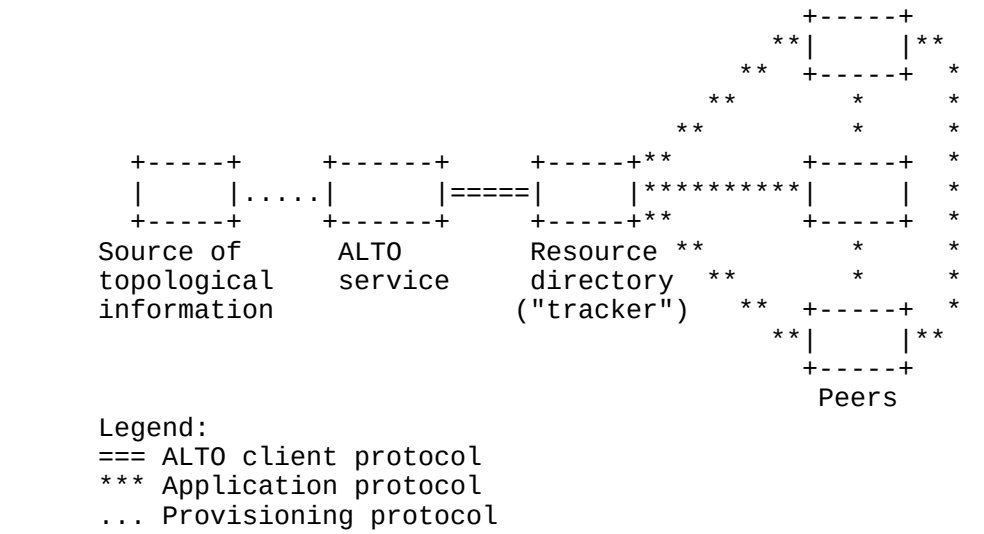


Figure 3: Overview of protocol interaction between ALTO elements, scenario with tracker

However, Figure 3 does not denote where the ALTO elements are actually located, i.e., if the tracker and the ALTO server are in the same ISP's domain, or if the tracker and the ALTO server are managed/ owned/located in different domains. The latter is the typical use case, e.g., taking Pirate Bay as example that serves Bittorrent peers world-wide.

2.2. Relationship between ALTO and Applications

ALTO is intended to be used by a wide-range of applications. However, any application using ALTO must also work if no ALTO servers can be found or if no responses to ALTO queries are received, e.g., due to connectivity problems or overload situation (see also [I-D.ietf-alto-reqs]). (Editor's note: better text needed here!)

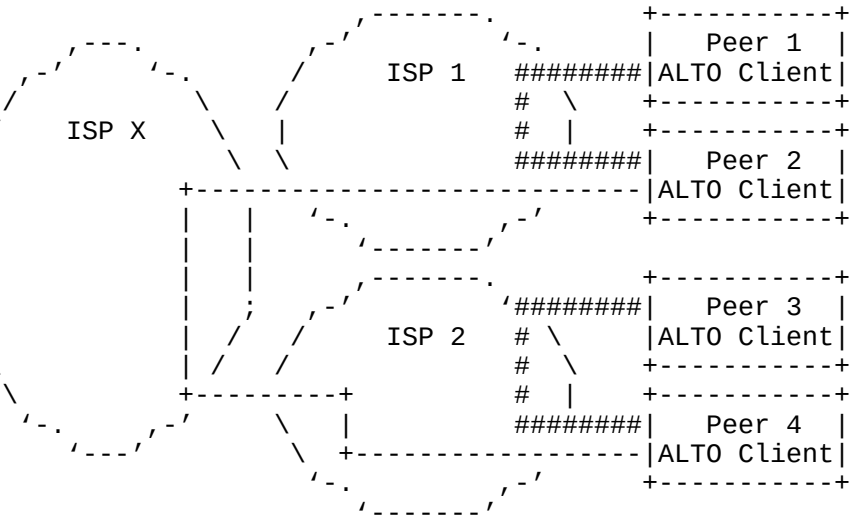
2.3. Provided Guidance

ALTO gives guidance to applications on what IP addresses or IP prefixes, and such which hosts are to be preferred according to the operator of the ALTO server. The general assumption of the ALTO WG is that a network operator would always express to prefer hosts in its own network while hosts located outside its own network are to be avoided (are undesired to be considered by the applications). This might be applicable in some cases but may not be applicable in the general case. The ALTO protocol gives only the means to let the ALTO server operator to express is preference, whatever this preference

is. This section explores this space.

2.3.1. Keeping Traffic Local in Network

ALTO guidance can be used to let applications prefer other peers within the same network operator's network instead of randomly connecting to other peers which are located in another operator's network. Figure 4 shows such a scenario where peers prefer peers in the same network (e.g., Peer 1 and Peer 2 in ISP1 and Peer 3 and Peer 4 in ISP2).



Legend:
preferred "connections"
--- non-preferred "connections"

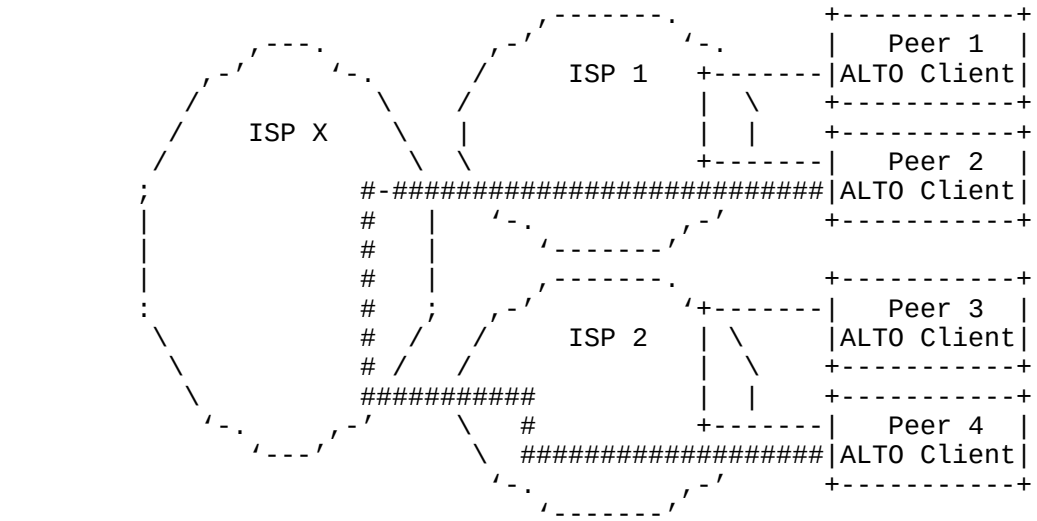
Figure 4: ALTO Traffic Network Localization

TBD: Describes limits of this approach (e.g., traffic localization guidance is of less use if the peers cannot upload); describe how maps would look like.

2.3.2. Off-Loading Traffic from Network

Another scenario where the use of ALTO can be beneficial is in mobile broadband networks, e.g., CDMA200 or UMTS, but where the network operator may have the desire to guide peers in its own network to use peers in remote networks. One reason can be that the wireless network is not made for the load cause by, e.g., peer-to-peer

applications, and the operator has the need that peers fetch their data from remote peers in other parts of the Internet.



Legend:
=== preferred "connections"
--- non-preferred "connections"

Figure 5: ALTO Traffic Network De-Localization

Figure 5 shows the result of such a guidance process where Peer 2 prefers a connection with Peer4 instead of Peer 1, as shown in Figure 4.

TBD: Limits of this approach in general and with respect to p2p. describe how maps would look like.

2.3.3. Intra-Network Localization/Bottleneck Off-Loading

The above sections described the results of the ALTO guidance on an inter-network level. However, ALTO can also be used to guide peers on which internal peers are to be preferred. For instance, to guide Peers on a remote network side to prefer to connect to each other, instead of crossing a bottleneck link, a backhaul link to connect the side to the network core. Figure 6 shows such a scenario where Peer 1 and Peer 2 are located in Net 2 of ISP1 and connect via a low capacity link to the core (Net 1) of the same ISP1. Peer1 and Peer 2 would both exchange their data with remote peers, probably clogging the bottleneck link.

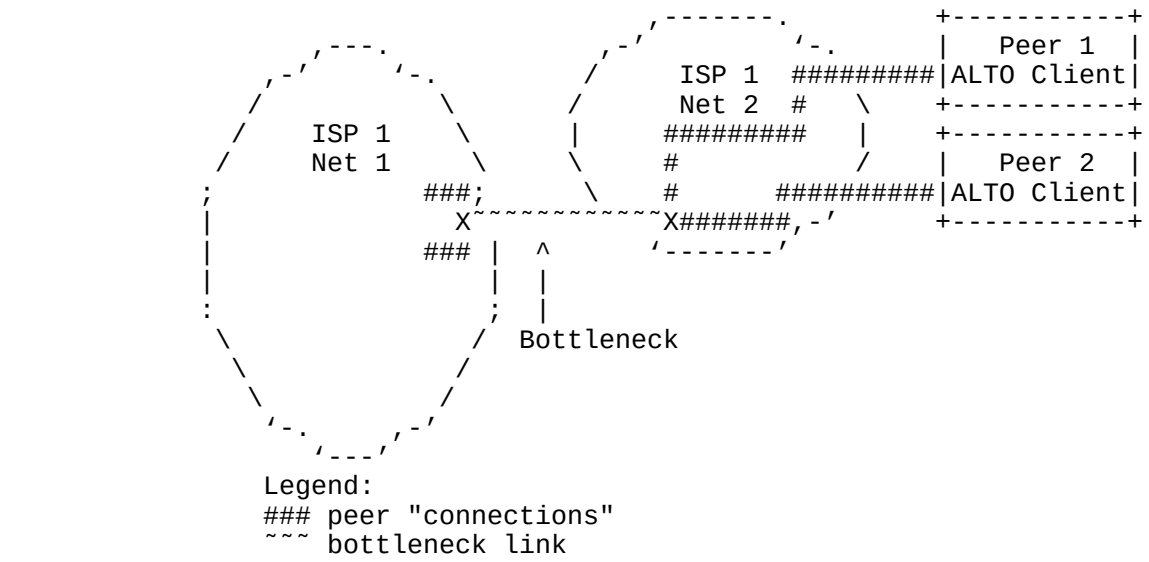


Figure 6: Without Intra-Network ALTO Traffic Localization

The operator can guide the peers in such a situation to try first local peers in the same network islands, avoiding or at least lowering the effect on the bottleneck link, as shown in Figure 7.

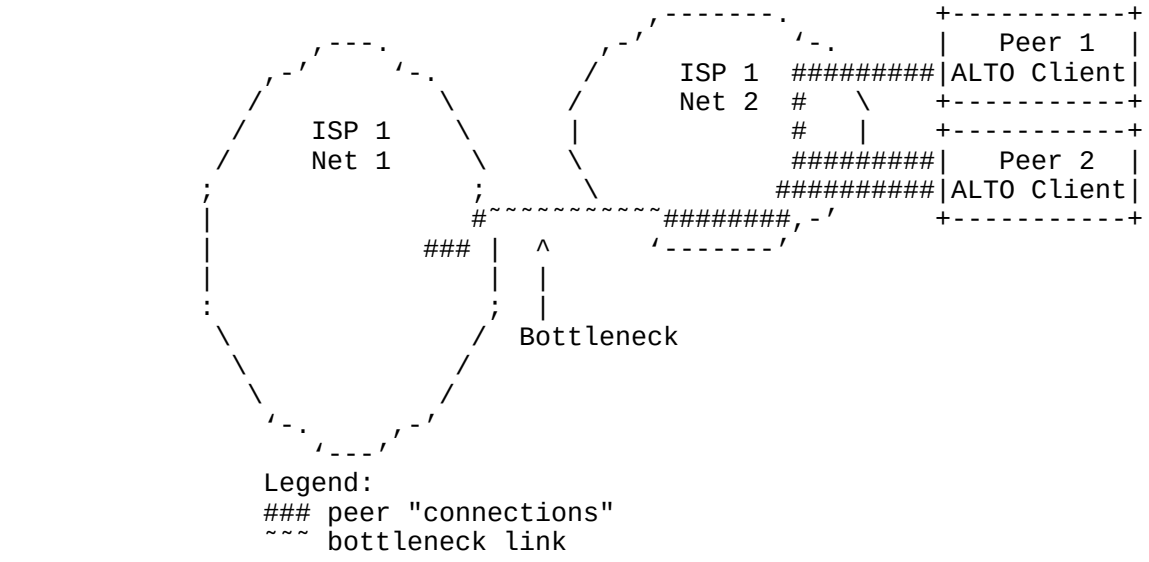


Figure 7: With Intra-Network ALTO Traffic Localization

TBD: describe how maps would look like.

2.4. Provisiong ALTO Maps

This section will describe how ALTO maps in the protocol can be populated before using them.

3. Deployment Considerations by ISPs

The Internet is a large network constituted of multiple networks worldwide. Numerous of these networks are built by telecom operators or network operators (named ISP in this memo), and these networks provide network connectivity, such as cable networks, 3G and so on. As well as some of networks are built by universities or big organizations themselves, and these networks are used to provide connectivity for research and work. The essence of Internet is its connectivity and sharing capability. However, ISPs emphasize network's manageability and controllability, because ISPs provide public network access service for most person and families, they need to manage, to control and to audit the traffic. Thus, it's important for ISPs to understand the requirement of optimizing traffic, and how to deploy ALTO service in these manageability and controllability networks.

3.1. Requirement for Traffic Optimization by ISPs

All networks of ISPs are connected to each other through peering points. From view of business mode, the inter-network settlement is needed in traffic exchanging between these ISP's networks. The current settlement can be costly. So to save these cost, the simple and basic method is to decrease the traffic exchange across the peering points and keep the traffic in own network area.

For some large ISPs, their whole network is layered. The upper layer network includes one or several backbone networks, and the lower layer network includes multiple access networks. These access networks are connected to backbone networks, and the exchange traffic with others through backbone network. In this kind of layered network, the bandwidth of backbone network is important and may be scarce. Traffic should be limited to the access networks, so to decrease the usage of backbone as far as possible.

Compared to fixed networks, mobile networks have some special characters, including small link bandwidth, high cost, limited radio frequency resource, and terminal battery. In mobile network, the usage of wireless link should be decreased as far as possible and be high-efficient. For example, in the case of a P2P service, the clients in the fixed network should decrease the data transport from the clients in the mobile networks, as well as the clients in the mobile networks should prefer the data transmission from the clients in the fixed networks.

3.2. Considerations for ISPs

3.2.1. Very small ISPs with simple Network Structure

For very small ISPs, the traffic optimizing problem they focus is that how to decrease the traffic exchanging with other ISPs, because of high settlement costs. To use the ALTO service to optimize traffic, small ISPs can define two optimization areas: one is their own network; the other is all outer networks connected with their network. The cost map can be defined like this: the cost of link between clients of inner ISP's networks is lower than from clients of outer ISP's networks to clients of inner ISP's networks. So the client of this ISP will prefer to require data from the clients in the same ISP with high priority.

One example is given as below in Figure 8. ISP A is one small ISP, only having one access network. In ALTO service deploying, we can define ISP A to be one optimization area, named as PID1, and define other networks to be the other optimization area, named as PID2. C1 is denoted as the link cost in inner ISP A. C2 is denoted as the link cost from PID2 to PID1. We define the cost map as:

C1<C2

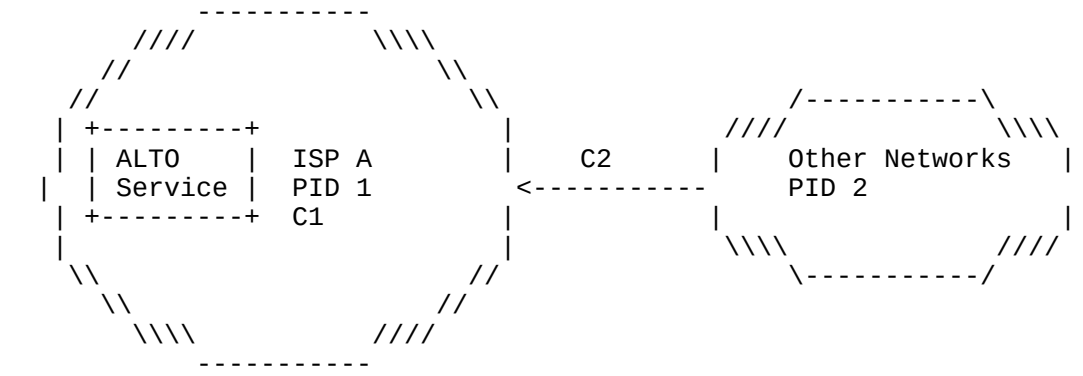


Figure 8: ALTO deployment in small ISPs

3.2.2. Large ISPs with layered fixed Network Structure

For large ISPs with layered fixed network structure, the traffic optimizing problems they focus will include that: using backbone network by high-efficiency, adjusting traffic balance in different access networks according to traffic conditions and management policies, and considering settlement cost with other ISPs. So in

ALTO service deploying to this kind of large ISP, first the optimization area can be defined according to real network condition. For example, each access network can be defined to be one optimization area. Then cost can be defined according to the optimizing requirement by ISPs. There is one example described below and also shown in Figure 9.

In this example, ISP A has one backbone network and three access networks, named as AN A, AN B, and AN C. A P2P application is used in this example. For the traffic optimization, the first requirement is to decrease the P2P traffic of backbone network in inner ISP A; and the second requirement is to decrease the P2P traffic to outer ISPs. Always, the second requirement is prior to the first one. Also, we assume that the settlement rate with ISP B is lower than with other ISPs. Then ISP A can deploy ALTO service to meet the need of traffic optimization. We will give the detail example of ALTO service definition and configuration according to requirements above.

In inner network of ISP A, we can define each access network to be one optimization area, and assign one PID to every access network, such as PID1, PID2, and PID 3. Because of different settlement with different outer ISPs, we define ISP B to be one optimization area, and assign PID 4 to it, as well as define all other networks to be one optimization area and PID 5.

We assign cost names (C1, C2, C3, C4, C5, C6, C7) as the figure below. C1 is denoted as the link cost in inner AN A, the same as C2 and C3. C4 is denoted as the link cost from PID 1 to PID 2, the same as C5. C6 is denoted as the link cost from the ISP B to ISP A. C7 is denoted as the link cost from other networks to ISP A.

According to discussion of the first requirement and the second requirement above, the relationship of these costs will be defined as: (C1, C2, C3) < (C4, C5) < (C6) < (C7)

This is one very simple example above, in which we do not consider the different link type of access network. In deploying ALTO service in real network, we must consider more real network conditions and requirements. One real example is described in greater detail in [I-D.lee-alto-chinatelecom-trial].

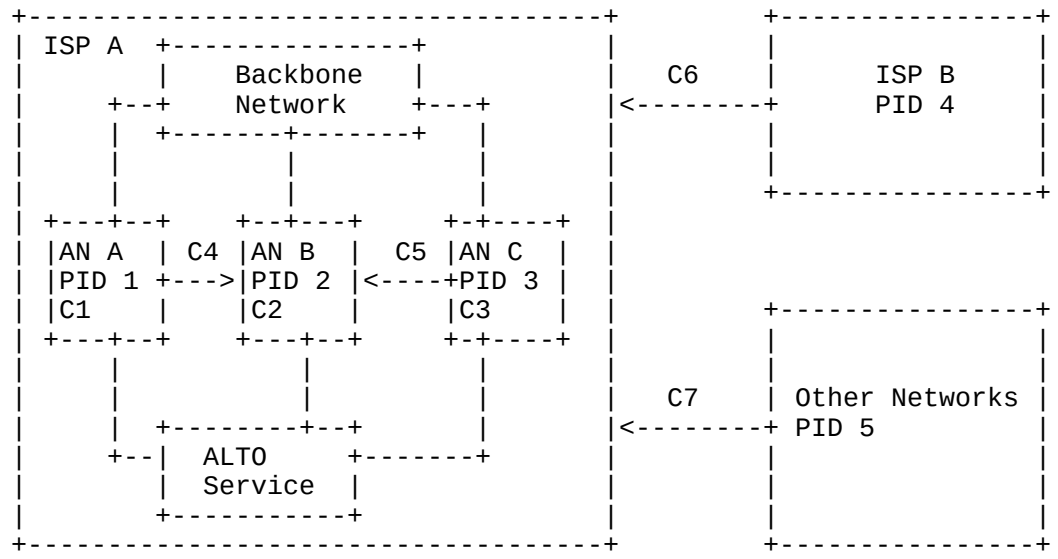


Figure 9: ALTO deployment in large ISPs with layered fixed network structures

3.2.3. ISPs with Mobile Network

For ISPs with mobile network and fixed network, the traffic optimizing problems they focus will be optimizing the mobile traffic, except problems on last hop section. Wireless radio frequency resource is scarce and costly in mobile network. The requirement of traffic optimization in mobile network is mainly decreasing the usage of radio resource. The ALTO service can be deployed to meet these needs.

For example in one ISP A as below in Figure 10, there is one mobile network is connected to backbone network. In this kind of network structure, mobile network can be defined as one optimization area, and assigned PID 1. We also define other PID and cost as figure below.

To decrease the usage of wireless link, the relationship of these costs will be defined to:

From view of mobile network:(C4 < C1). This means that, the clients in mobile network requiring data resource from clients of the other access networks is prior to clients of mobile network. This policy can decrease the usage of wireless link and power consumption in terminal.

From view of AN A:(C2 < C6, C5 = maximum cost). This means that, to other optimization area, requiring data from mobile network should be avoided.

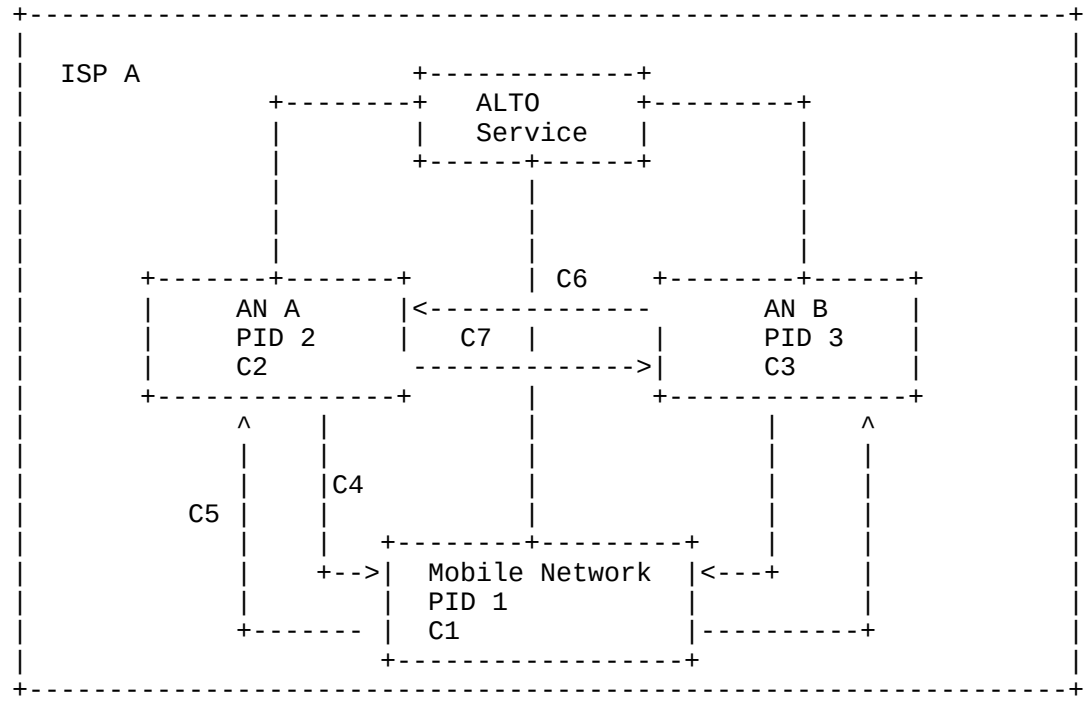


Figure 10: ALTO deployment in ISPs with mobile network

4. Using ALTO for P2P

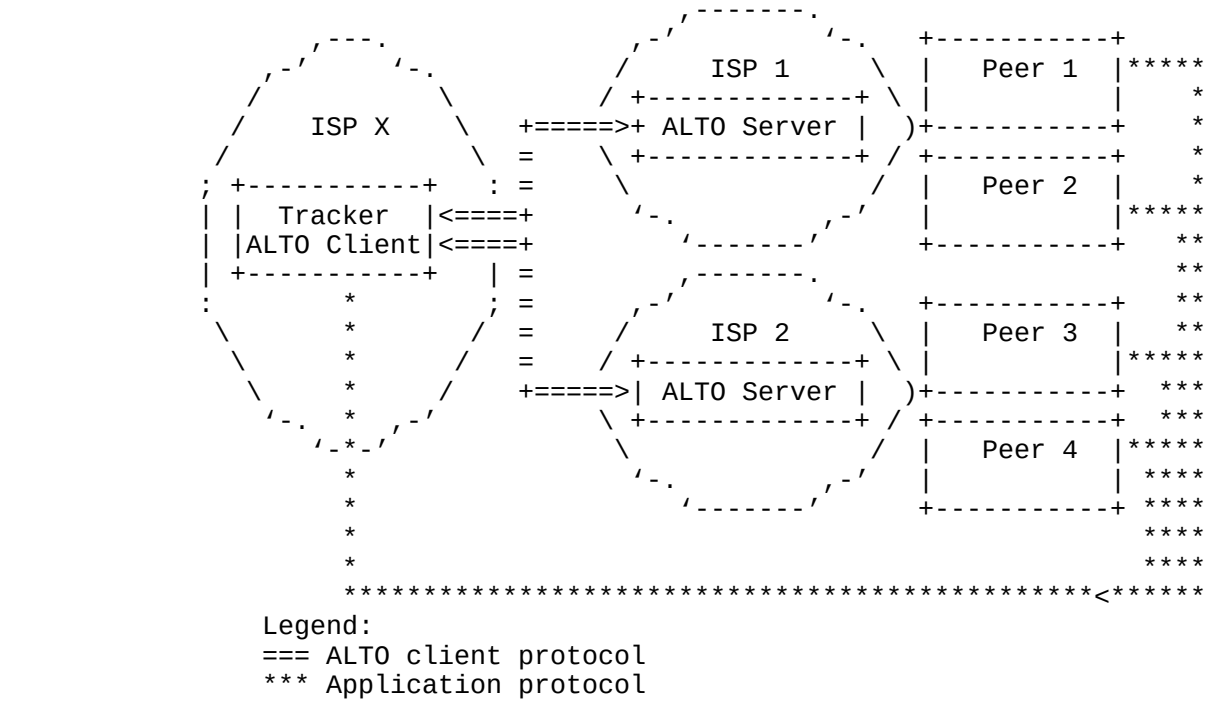


Figure 11: Global tracker accessing ALTO server at various ISPs

Figure 11 depicts a tracker-based system, where the tracker embeds the ALTO client. The tracker itself is hosted and operated by an entity different than the ISP hosting and operating the ALTO server. Initially, the tracker has to look-up the ALTO server in charge for each peer where it receives a ALTO query for. Therefore, the ALTO server has to discover the handling ALTO server, as described in [I-D.kiesel-alto-3pdisc]. However, the peers do not have any way to query the server themselves. This setting allows to give the peers a better selection of candidate peers for their operation at an initial time, but does not consider peers learned through direct peer-to-peer knowledge exchange, AKA peer exchange in various peer-to-peer protocols.

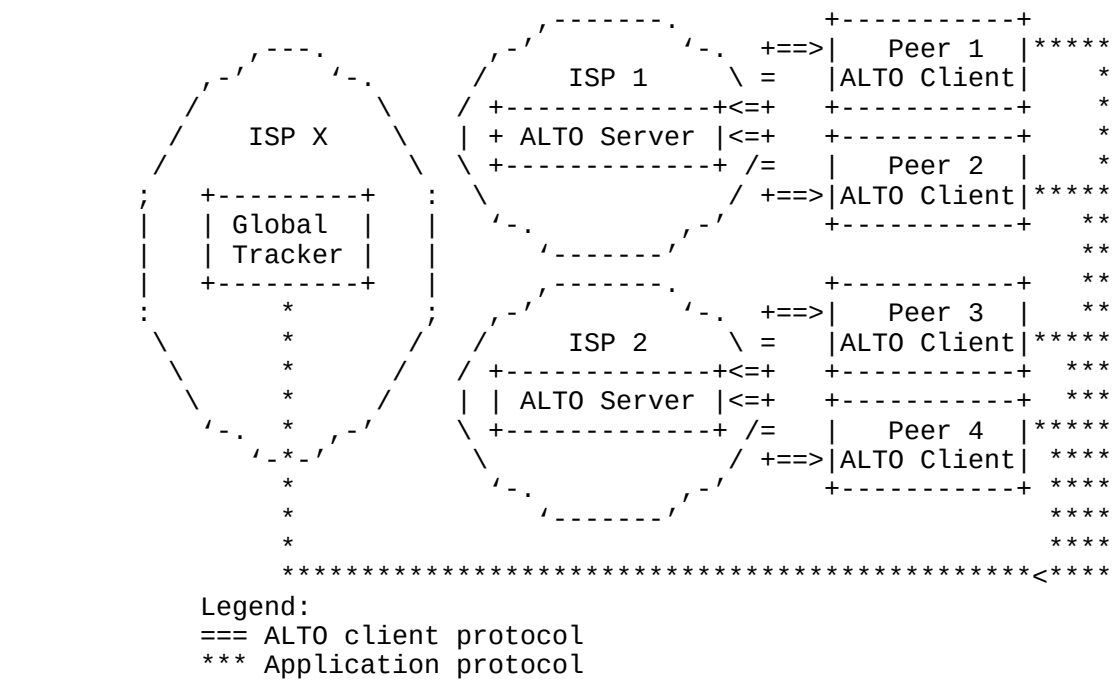


Figure 12: Global Tracker - Local ALTO Servers

The scenario in Figure 12 lets the peers directly communicate with their ISP's ALTO server (i.e., ALTO client embedded in the peers), giving thus the peers the most control on which information they query for, as they can integrate information received from trackers and through direct peer-to-peer knowledge exchange.

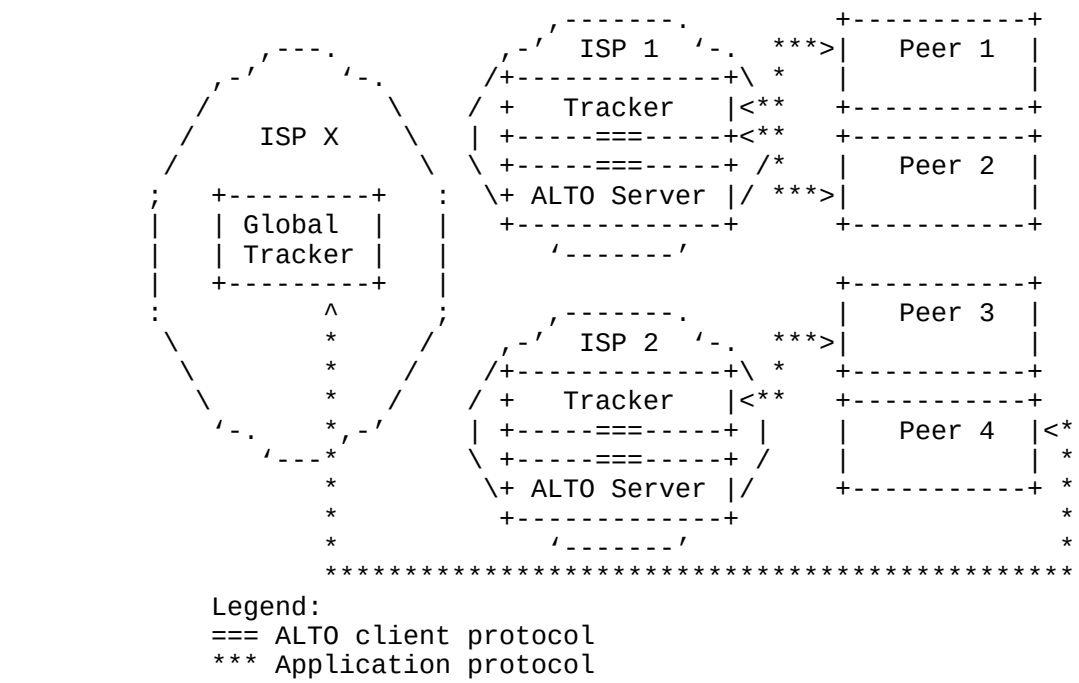


Figure 13: P4P approach with local tracker and local ALTO server

There are some attempts to let ISP's to deploy their own trackers, as shown in Figure 13. In this case, the client has no chance to get guidance from the ALTO server, other than talking to the ISP's tracker. However, the peers would have still chance the contact other trackers, deployed by entities other than the peer's ISP.

Figure 13 and Figure 11 ostensibly take peers the possibility to directly query the ALTO server, if the communication with the ALTO server is not permitted for any reason. However, considering the plethora of different applications of ALTO, e.g., multiple tracker and non-tracker based P2P systems and or applications searching for relays, it seems to be beneficial for all participants to let the peers directly query the ALTO server. The peers are also the single point having all operational knowledge to decide whether to use the ALTO guidance and how to use the ALTO guidance. This is a preference for the scenario depicted in Figure Figure 12.

4.1. Using ALTO for Tracker-based Peer-to-Peer Applications

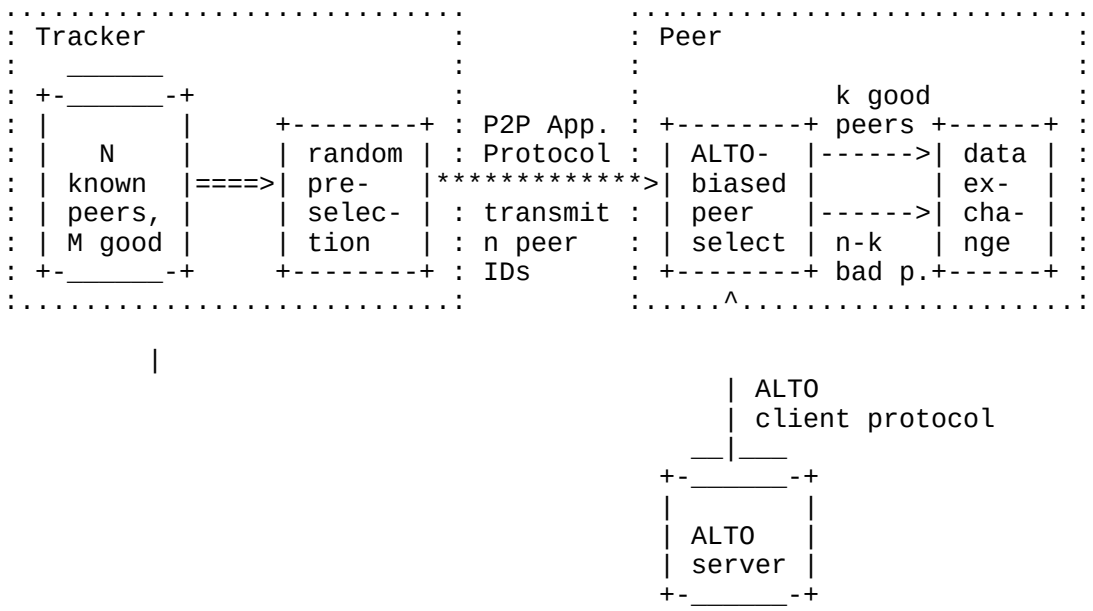


Figure 14: Tracker-based P2P Application with random peer preselection

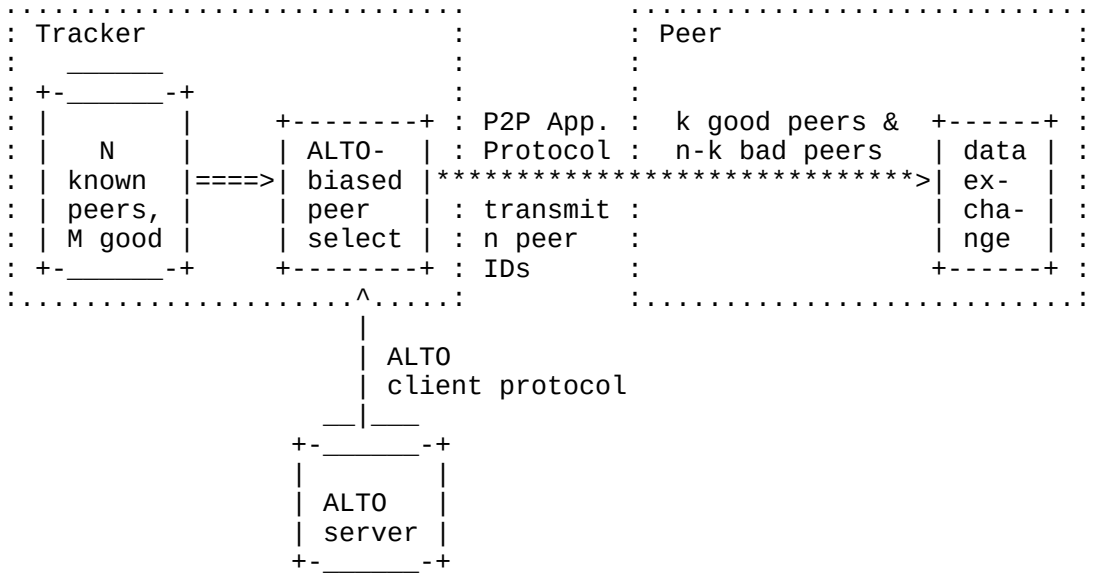


Figure 15: Tracker-based P2P Application with ALTO client in tracker

TBD: explain why Figure 15 usually will yield better results wrt. peer selection than Figure 14.

4.2. Expectations of ALTO

This section hints to some recent experiments conducted with ALTO-like deployments in Internet Service Provider (ISP) network's. NTT performed tests with their HINT server implementation and dummy nodes to gain insight on how an ALTO-like service influence a peer-to-peer systems [I-D.kamei-p2p-experiments-japan]. The results of an early experiment conducted in the Comcast network are documented here[RFC5632]

5. Using ALTO for CDNs

Section 2 discussed the placement and usage of ALTO for P2P systems, but not beyond. This section discuss the usage of ALTO for Content Delivery Networks (CDNs). CDNs are used to bring a service (e.g., a web page, videos, etc) closer to the location of the user - where close refers to shorten the distance between the client and the server in the IP topology. CDNs use several techniques to decide which server is closest to a client requesting a service. One common way to do so, is relying on the DNS system, but there are many other ways, see [RFC3568].

The general issue for CDNs, independent of DNS or HTTP Redirect based approaches (see, for instance, [I-D.penno-alto-cdn]), is that the CDN logic has to match the client’s IP address with the closest CDN cache. This matching is not trivial, for instance, in DNS based approaches, where the IP address of the DNS original requester is unknown (see [I-D.vandergaast-edns-client-ip] for a discussion of this and a solution approach).

6. Advanced Features

6.1. Cascading ALTO Servers

The main assumptions of ALTO seems to be each ISP operates its own ALTO server independently, irrespectively of the ISP's situation. This may true for most envisioned deployments of ALTO but there are certain deployments that may have different settings. Figure 16 shows such setting, were for example, a university network is connected to two upstream providers. ISP2 if the national research network and ISP1 is a commercial upstream provider to this university network. The university, as well as ISP1, are operating their own ALTO server. The ALTO clients, located on the peers will contact the ALTO server located at the university.

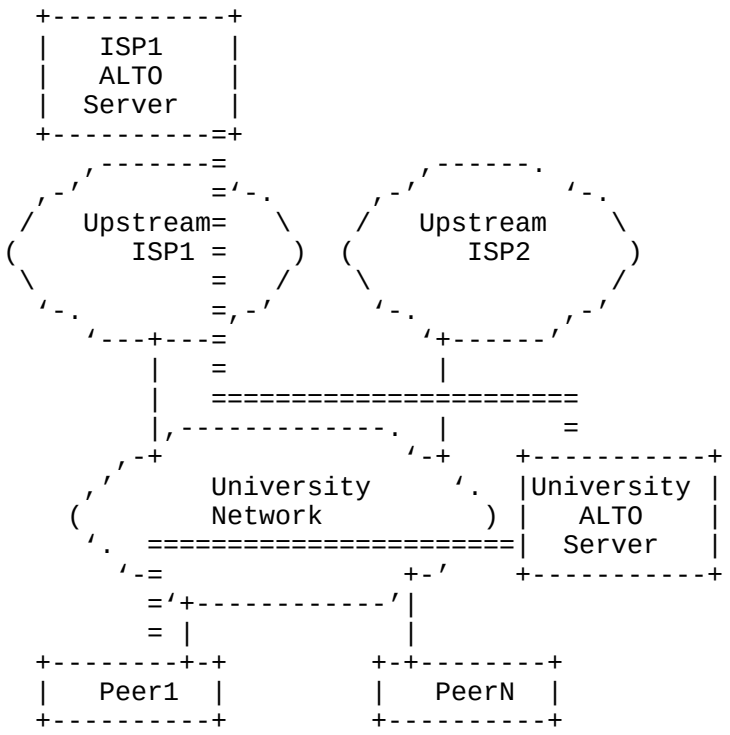


Figure 16: Cascaded ALTO Server

In this setting all "destinations" useful for the peers within ISP2 are free-of-charge for the peers located in the university network (i.e., they are preferred in the rating of the ALTO server). However, all traffic that is not towards ISP2 will be handled by the

ISP1 upstream provider. Therefore, the ALTO server at the university has also to include the guidance given by the ISP1 ALTO server in its replies to the ALTO clients. This can be called cascaded ALTO servers.

6.2. ALTO for IPv4 and IPv6

TBD

6.3. Monitoring ALTO

In addition to providing configuration, an ISP providing ALTO may want to deploy a monitoring infrastructure to assess the benefits of ALTO and adjust its ALTO configuration according to the results of the monitoring.

To construct an effective monitoring infrastructure, the ISP should (1) define the performance metrics to be monitored; (2) and identify and deploy data sources to collect data to compute the performance metrics. We discuss both below.

[Editor’s note: Is there a relationship to the IPPM working group at the IETF?]

6.3.1. Monitoring Metrics Definition

- Inter-domain ALTO-Integrated Application Traffic (Network metric): This metric includes total cross domain traffic generated by applications that utilize ALTO guidance. This metric evaluates the impacts of ALTO on the inbound and outbound traffic of a domain.

- Total Inter-domain Traffic (Network metric): This is similar to the preceding but focuses on all of the traffic, ALTO aware or not. One possibility is that some of the reduction of interdomain traffic by ALTO aware applications may (XXX missing words?). This metric is always used with the preceding and the following metrics.

- Intra-domain ALTO-Integrated Application Traffic (Network metric). (XXX description missing)

- Network hop count (Network metric): This metric provides the average number of hops that traffic traverses inside a domain. ALTO may reduce not only traffic volume but also the hops. The metric can also indirectly reflect some application performance (e.g., latency).

- Application download rate (Application metric): This metric measures application performance directly. Download means inbound traffic to one user. Global average means the average value of all users' download rates in one or more domains.

- Application Client type audit(Application metric): this metric gives the audit of client types in ALTO service. The current types include fixed network client and mobile network client.

6.3.2. Monitoring Data Sources

The preceding metrics are derived from data sources. We identify three data sources.

1. Application Log Server: Many application systems deploy Log Servers to collect data.

2. P2P Clients: Some P2P applications may not have Log Servers. When available, P2P client logs can provide data. This is for P2P application

3. OAM: Many ISPs deploy OAM systems to monitor IP layer traffic. An OAM provides traffic monitoring of every network device in its management area. It provides data such as link physical bandwidth and traffic volumes.

6.3.3. Monitoring Structure

As discussed in the preceding section, some data sources are from ISP while some others are from application. When there is a collaboration agreement between the ISP and an application, there can be an integrated monitoring system as shown in the figure below. In particular, an application developer may deploy Monitor Clients to communicate with Monitor Server of the ISP to transmit raw data from the Log Server or P2P clients of the application to the ISP.

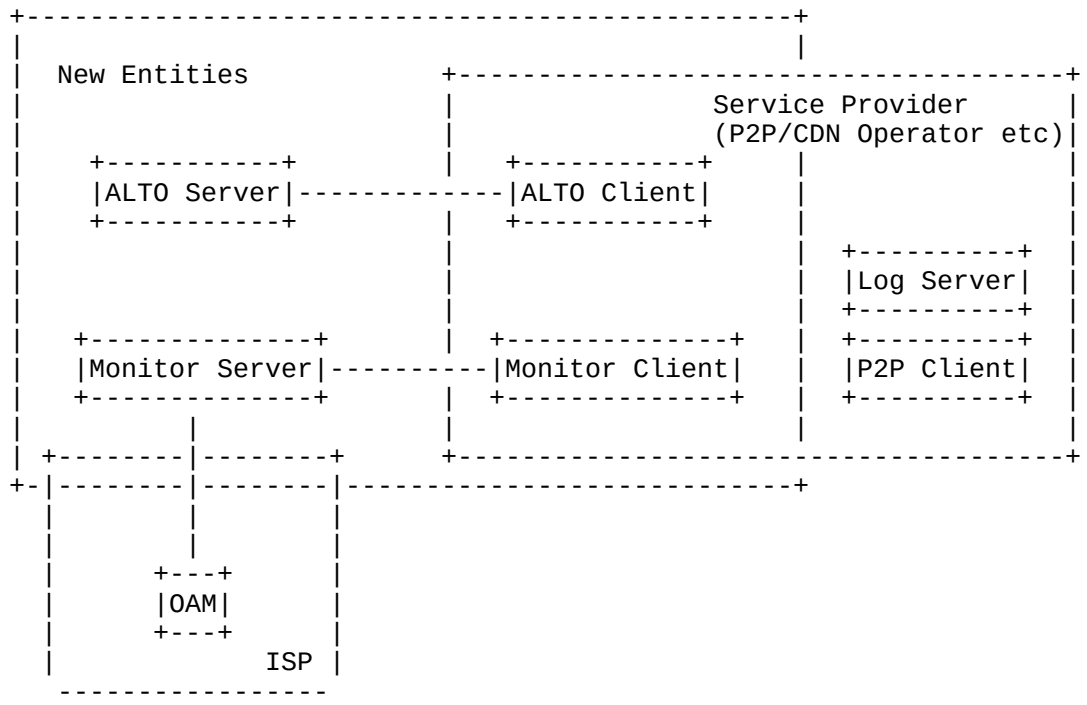


Figure 17: Monitoring Structure

7. Known Limitations of ALTO

This section describes some known limitations of ALTO in general or specific mechanisms in ALTO.

7.1. Limitations of Map-based Approaches

The specification of the ALTO protocol [I-D.ietf-alto-protocol] uses, amongst others mechanism, so-called network maps. The network map approach uses Host Group Descriptors that group one or multiple subnetworks (i.e., IP prefixes) to a single Host Group Descriptor. A set of IP prefixes is called partition and the associated Host Group Descriptor is called partition ID. The "costs" between the various partition IDs is stored in a second map, the cost map. Map-based approaches are chosen as they lower the signaling load on the server, as the maps have only to be retrieved if they are changed.

The main assumption for map-based approaches is that the information provided in these maps is static for a longer period of time, where this period of time refers to days, but not hours or even minutes. This assumption is fine, as long as the network operator does not change any parameter, e.g., routing within the network and to the upstream peers, IP address assignment stays stable (and thus the mapping to the partitions). However, there are several cases where this assumption is not valid, as:

1. ISPs reallocate IPv4 subnets from time to time;

2. ISPs reallocate IPv4 subnets on short notice;

3. IP prefix blocks may be assigned to a single DSLAM which serves a variety of access networks.

For 1): ISPs reallocate IPv4 subnets within their infrastructure from time to time, partly to ensure the efficient usage of IPv4 addresses (a scarce resource), and partly to enable efficient route tables within their network routers. The frequency of these "renumbering events" depend on the growth in number of subscribers and the availability of address space within the ISP. As a result, a subscriber’s household device could retain an IPv4 address for as short as a few minutes, or for months at a time or even longer.

> Some folks have suggested that ISPs providing ALTO services could sub-divide their subscribers’ devices into different IPv4 subnets (or certain IPv4 address ranges) based on the purchased service tier, as well as based on the location in the network topology. The problem is that this sub-allocation of IPv4 subnets tends to decrease the efficiency of IPv4 address allocation. A growing ISP

that needs to maintain high efficiency of IPv4 address utilization may be reluctant to jeopardize their future acquisition of IPv4 address space.

However, this is not an issue for map-based approaches if changes are applied in the order of days.

For 2): ISPs can use techniques, such as ODAP (XXX) that allow the reallocation of IP prefixes on very short notice, i.e., within minutes. An IP prefix that has no IP address assignment to a host anymore can be reallocate to areas where there is currently a high demand for IP addresses.

For 3): In DSL-based access networks, IP prefixes are assigned to DSLAMs which are the first IP-hop in the access-network between the CPE and the Internet. The access-network between CPE and DSLAM (called aggregation network) can have varying characteristics (and thus associated costs), but still using the same IP prefix. For instance one IP addresses IP11 out of a IP prefix IP1 can be assigned to a VDSL (e.g., 2 MBit/s uplink) access-line while the subsequent IP address IP12 is assigned to a slow ADSL line (e.g., 128 kbit/s uplink). These IP addresses are assigned on a first come first served basis, i.e., the a single IP address out of the same IP prefix can change its associated costs quite fast. This may not be an issue with respect to the used upstream provider (thus the cross ISP traffic) but depending on the capacity of the aggregation-network this may raise to an issue.

7.2. Limitiations of Non-Map-based Approaches

The specification of the ALTO protocol [I-D.ietf-alto-protocol] uses, amongst others mechanism, a mechanism called Endpoint Cost Service. ALTO clients can ask guidance for specific IP addresses to the ALTO server. However, asking for IP addresses, asking with long lists of IP addresses, and asking quite frequent may overload the ALTO server. The server has to rank each received IP address which causes load at the server. This may be amplified by the fact that not only a single ALTO client is asking for guidance, but a larger number of them.

Caching of IP addresses at the ALTO client or the usage of the H12 approach [I-D.kiesel-alto-h12] in conjunction with caching may lower the query load on the ALTO server.

7.3. General Challenges

An ALTO server stores information about preferences (e.g., a list of preferred autonomous systems, IP ranges, etc) and ALTO clients can retrieve these preferences. However, there are basically two

different approaches on where the preferences are actually processed:

1. The ALTO server has a list of preferences and clients can retrieve this list via the ALTO protocol. This preference list can be partially updated by the server. The actual processing of the data is done on the client and thus there is no data of the client's operation revealed to the ALTO server .

2. The ALTO server has a list of preferences or preferences calculated during runtime and the ALTO client is sending information of its operation (e.g., a list of IP addresses) to the server. The server is using this operational information to determine its preferences and returns these preferences (e.g., a sorted list of the IP addresses) back to the ALTO client.

Approach 1 (we call it H1) has the advantage (seen from the client) that all operational information stays within the client and is not revealed to the provider of the server. On the other hand, does approach 1 require that the provider of the ALTO server, i.e., the network operator, reveals information about its network structure (e.g., AS numbers, IP ranges, topology information in general) to the ALTO client.

Approach 2 (we call it H2) has the advantage (seen from the operator) that all operational information stays with the ALTO server and is not revealed to the ALTO client. On the other hand, does approach 2 require that the clients send their operational information to the server.

Both approaches have their pros and cons and are extensively discussed on the ALTO mailing list. But there is basically a dilemma: Approach 1 is seen as the only working solution by peer-to-peer software vendors and approach 2 is seen as the only working by the network operators. But neither the software vendors nor the operators seem to willing to change their position. However, there is the need to get both sides on board, to come to a solution.

8. Extensions to the ALTO Protocol

8.1. Host Group Descriptors

Host group descriptors are used in the ALTO client protocol to describe the location of a host in the network topology. The ALTO client protocol specification defines a basic set of host group descriptor types, which have to be supported by all implementations, and an extension procedure for adding new descriptor types . The following list gives an overview on further host group descriptor types that have been proposed in the past, or which are in use by ALTO-related prototype implementations. This list is not intended as normative text. Instead, the only purpose of the following list is to document the descriptor types that have been proposed so far, and to solicit further feedback and discussion:

- Autonomous System (AS) number
- Protocol-specific group identifiers, which expand to a set of IP address ranges (CIDR) and/or AS numbers. In one specific solution proposal, these are called Partition ID (PID).

8.2. Rating Criteria

Rating criteria are used in the ALTO client protocol to express topology- or connectivity-related properties, which are evaluated in order to generate the ALTO guidance. The ALTO client protocol specification defines a basic set of rating criteria, which have to be supported by all implementations, and an extension procedure for adding new criteria . The following list gives an overview on further rating criteria that have been proposed in the past, or which are in use by ALTO-related prototype implementations. This list is not intended as normative text. Instead, the only purpose of the following list is to document the rating criteria that have been proposed so far, and to solicit further feedback and discussion:

8.2.1. Distance-related Rating Criteria

- Relative topological distance: relative means that a larger numerical value means greater distance, but it is up to the ALTO service how to compute the values, and the ALTO client will not be informed about the nature of the information. One way of generating this kind of information MAY be counting AS hops, but when querying this parameter, the ALTO client MUST NOT assume that the numbers actually are AS hops.
- Absolute topological distance, expressed in the number of traversed autonomous systems (AS).

- Absolute topological distance, expressed in the number of router hops (i.e., how much the TTL value of an IP packet will be decreased during transit).

- Absolute physical distance, based on knowledge of the approximate geolocation (continent, country) of an IP address.

8.2.2. Charging-related Rating Criteria

- Traffic volume caps, in case the Internet access of the resource consumer is not charged by "flat rate". For each candidate resource provider, the ALTO service could indicate the amount of data that may be transferred from/to this resource provider until a given point in time, and how much of this amount has already been consumed. Furthermore, it would have to be indicated how excess traffic would be handled (e.g., blocked, throttled, or charged separately at an indicated price). The interaction of several applications running on a host, out of which some use this criterion while others don't, as well as the evaluation of this criterion in resource directories, which issue ALTO queries on behalf of other peers, are for further study.

8.2.3. Performance-related Rating Criteria

The following rating criteria are subject to the remarks below.

- The minimum achievable throughput between the resource consumer and the candidate resource provider, which is considered useful by the application (only in ALTO queries), or

- An arbitrary upper bound for the throughput from/to the candidate resource provider (only in ALTO responses). This may be, but is not necessarily the provisioned access bandwidth of the candidate resource provider.

- The maximum round-trip time (RTT) between resource consumer and the candidate resource provider, which is acceptable for the application for useful communication with the candidate resource provider (only in ALTO queries), or

- An arbitrary lower bound for the RTT between resource consumer and the candidate resource provider (only in ALTO responses). This may be, for example, based on measurements of the propagation delay in a completely unloaded network.

The ALTO client MUST be aware, that with high probability, the actual performance values differ significantly from these upper and lower bounds. In particular, an ALTO client MUST NOT consider the "upper

bound for throughput" parameter as a permission to send data at the indicated rate without using congestion control mechanisms.

The discrepancies are due to various reasons, including, but not limited to the facts that

- the ALTO service is not an admission control system
- the ALTO service may not know the instantaneous congestion status of the network
- the ALTO service may not know all link bandwidths, i.e., where the bottleneck really is, and there may be shared bottlenecks
- the ALTO service may not know whether the candidate peer itself is overloaded
- the ALTO service may not know whether the candidate peer throttles the bandwidth it devotes for the considered application
- the ALTO service may not know whether the candidate peer will throttle the data it sends to us (e.g., because of some fairness algorithm, such as tit-for-tat)

Because of these inaccuracies and the lack of complete, instantaneous state information, which are inherent to the ALTO service, the application must use other mechanisms (such as passive measurements on actual data transmissions) to assess the currently achievable throughput, and it MUST use appropriate congestion control mechanisms in order to avoid a congestion collapse. Nevertheless, these rating criteria may provide a useful shortcut for quickly excluding candidate resource providers from such probing, if it is known in advance that connectivity is in any case worse than what is considered the minimum useful value by the respective application.

8.2.4. Inappropriate Rating Criteria

Rating criteria that SHOULD NOT be defined for and used by the ALTO service include:

- Performance metrics that are closely related to the instantaneous congestion status. The definition of alternate approaches for congestion control is explicitly out of the scope of ALTO. Instead, other appropriate means, such as using TCP based transport, have to be used to avoid congestion.

9. API between ALTO Client and Application

This sections gives some informational guidance on how the interface between the actual application using the ALTO guidance and the ALTO client can look like.

This is still TBD.

10. Security Considerations

The ALTO protocol itself, as well as, the ALTO client and server raise new security issues beyond the one mentioned in [I-D.ietf-alto-protocol] and issues related to message transport over the Internet. For instance, Denial of Service (DoS) is of interest for the ALTO server and also for the ALTO client. A server can get overloaded if too many TCP requests hit the server, or if the query load of the server surpasses the maximum computing capacity. An ALTO client can get overloaded if the responses from the sever are, either intentionally or due to an implementation mistake, too large to be handled by that particular client.

10.1. Information Leakage from the ALTO Server

The ALTO server will be provisioned with information about the owning ISP's network and very likely also with information about neighboring ISPs. This information (e.g., network topology, business relations, etc) is consider to be confidential to the ISP and must not be revealed.

The ALTO server will naturally reveal parts of that information in small doses to peers, as the guidance given will depend on the above mentioned information. This is seen beneficial for both parties, i.e., the ISP's and the peer's. However, there is the chance that one or multiple peers are querying an ALTO server with the goal to gather information about network topology or any other data considered confidential or at least sensitive. It is unclear whether this is a real technical security risk or whether this is more a perceived security risk.

10.2. ALTO Server Access

Depending on the use case of ALTO, several access restrictions to an ALTO server may or may not apply. For an ALTO server that is solely accessible by peers from the ISP network (as shown in Figure 12), for instance, the source IP address can be used to grant only access from that ISP network to the server. This will "limit" the number of peers able to attack the server to the user's of the ISP (however, including botnet computers).

On the other hand, if the ALTO server has to be accessible by parties not located in the ISP's network (see Figure Figure 11), e.g., by a third-party tracker or by a CDN system outside the ISP's network, the access restrictions have to be more loose. In the extreme case, i.e., no access restrictions, each and every host in the Internet can access the ALTO server. This might no the intention of the ISP, as the server is not only subject to more possible attacks, but also on

the load imposed to the server, i.e., possibly more ALTO clients to serve and thus more work load.

10.3. Faking ALTO Guidance

It has not yet been investigated how a faked or wrong ALTO guidance by an ALTO server can impact the operation of the network and also the peers.

Here is a list of examples how the ALTO guidance could be faked and what possible consequences may arise:

Sorting An attacker could change to sorting order of the ALTO guidance (given that the order is of importance, otherwise the ranking mechanism is of interest), i.e., declaring peers located outside the ISP as peers to be preferred. This will not pose a big risk to the network or peers, as it would mimic the "regular" peer operation without traffic localization, apart from the communication/processing overhead for ALTO. However, it could mean that ALTO is reaching the opposite goal of shuffling more data across ISP boundaries, incurring more costs for the ISP.

Preference of a single peer A single IP address (thus a peer) could be marked as to be preferred all over other peers. This peer can be located within the local ISP or also in other parts of the Internet (e.g., a web server). This could lead to the case that quite a number of peers to trying to contact this IP address, possibly causing a Denial of Service (DoS) attack.

This section is solely giving a first shot on security issues related to ALTO deployments.

11. Conclusion

This is the first version of the deployment considerations and for sure the considerations are yet incomplete and imprecise.

12. References

12.1. Normative References

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

[RFC3568] Barbir, A., Cain, B., Nair, R., and O. Spatscheck, "Known Content Network (CN) Request-Routing Mechanisms", RFC 3568, July 2003.

12.2. Informative References

[I-D.ietf-alto-protocol]
Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-09 (work in progress), June 2011.

[I-D.ietf-alto-reqs]
Kiesel, S., Previdi, S., Stiemerling, M., Woundy, R., and Y. Yang, "Application-Layer Traffic Optimization (ALTO) Requirements", draft-ietf-alto-reqs-10 (work in progress), June 2011.

[I-D.kamei-p2p-experiments-japan]
Kamei, S., Momose, T., and T. Inoue, "ALTO-Like Activities and Experiments in P2P Network Experiment Council", draft-kamei-p2p-experiments-japan-05 (work in progress), March 2011.

[I-D.kiesel-alto-3pdisc]
Kiesel, S., Stiemerling, M., Schwan, N., Scharf, M., Tomsu, M., and S. Yongchao, "ALTO Server Discovery Protocol", draft-kiesel-alto-3pdisc-05 (work in progress), March 2011.

[I-D.kiesel-alto-h12]
Kiesel, S. and M. Stiemerling, "ALTO H12", draft-kiesel-alto-h12-02 (work in progress), March 2010.

[I-D.lee-alto-chinatelecom-trial]
Li, K. and G. Jian, "ALTO and DECADE service trial within China Telecom", draft-lee-alto-chinatelecom-trial-02 (work in progress), April 2011.

[I-D.penno-alto-cdn]
Penno, R., Medved, J., Alimi, R., Yang, R., and S. Previdi, "ALTO and Content Delivery Networks", draft-penno-alto-cdn-03 (work in progress), March 2011.

[I-D.vandergaast-edns-client-ip]
Contavalli, C., Gaast, W., Leach, S., and D. Rodden, "Client IP information in DNS requests", draft-vandergaast-edns-client-ip-01 (work in progress), May 2010.

[RFC5632] Griffiths, C., Livingood, J., Popkin, L., Woundy, R., and Y. Yang, "Comcast's ISP Experiences in a Proactive Network Provider Participation for P2P (P4P) Technical Trial", RFC 5632, September 2009.

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

Appendix A. Acknowledgments

Xianghui Sun, Lee Kai, and Richard Yang contributed Section 3 and Section 6.3

Martin Stiemerling is partially supported by the COAST project (COntent Aware Searching, retrieval and sTreaming, http://www.coast-fp7.eu), a research project supported by the European Commission under its 7th Framework Program (contract no. 248036). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the COAST project or the European Commission.

Authors' Addresses

Martin Stiemerling
NEC Laboratories Europe
Kurfuerstenanlage 36
Heidelberg 69115
Germany

Phone: +49 6221 4342 113
Fax: +49 6221 4342 155
Email: martin.stiemerling@neclab.eu
URI: http://ietf.stiemerling.org

Sebastian Kiesel
University of Stuttgart, Computing Center
Allmandring 30
Stuttgart 70550
Germany

Email: ietf-alto@skiesel.de

ALTO WG
Internet-Draft
Intended status: Standards Track
Expires: December 29, 2011

R. Alimi, Ed.
Google
R. Penno, Ed.
Juniper Networks
Y. Yang, Ed.
Yale University
June 27, 2011

ALTO Protocol
draft-ietf-alto-protocol-09.txt

Abstract

Networking applications today already have access to a great amount of Inter-Provider network topology information. For example, views of the Internet routing table are easily available at looking glass servers and entirely practical to be downloaded by clients. What is missing is knowledge of the underlying network topology from the ISP or Content Provider (henceforth referred as Provider) point of view. In other words, what a Provider prefers in terms of traffic optimization -- and a way to distribute it.

The ALTO Service provides information such as preferences of network resources with the goal of modifying network resource consumption patterns while maintaining or improving application performance. This document describes a protocol implementing the ALTO Service. While such service would primarily be provided by the network (i.e., the ISP), content providers and third parties could also operate this service. Applications that could use this service are those that have a choice in connection endpoints. Examples of such applications are peer-to-peer (P2P) and content delivery networks.

Requirements Language

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 RFC 2119 [RFC2119].

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on December 29, 2011.

Copyright Notice

Table of Contents

1. Introduction

1.1. Background and Problem Statement

Today, network information available to applications is mostly from the view of endhosts. There is no clear mechanism to convey information about the network's preferences to applications. By leveraging better network-provided information, applications have the potential to become more network-efficient (e.g., reduce network resource consumption) and achieve better application performance (e.g., accelerated download rate). The ALTO Service intends to provide a simple way to convey network information to applications.

The goal of this document is to specify a simple and unified protocol that meets the ALTO requirements [I-D.ietf-alto-reqs] while providing a migration path for Internet Service Providers (ISP), Content Providers, and clients that have deployed protocols with similar intentions (see below). This document is a work in progress and will be updated with further developments.

1.2. Design History and Merged Proposals

The protocol specified here consists of contributions from

- P4P [I-D.p4p-framework], [P4P-SIGCOMM08], [I-D.wang-alto-p4p-specification];
- ALTO Info-Export [I-D.shalunov-alto-infoexport];
- Query/Response [I-D.saumitra-alto-queryresponse], [I-D.saumitra-alto-multi-ps];
- ATTP [ATTP];
- Proxidor [I-D.akonjang-alto-proxidor].

See Appendix A for a list of people that have contributed significantly to this effort and the projects and proposals listed above.

1.3. Solution Benefits

The ALTO Service offers many benefits to both end-users (consumers of the service) and Internet Service Providers (providers of the service).

1.3.1. Service Providers

The ALTO Service enables ISPs to influence the peer selection process in distributed applications in order to increase locality of traffic, improve user-experience, amongst others. It also helps ISPs to efficiently manage traffic that traverses more expensive links such as transit and backup links, thus allowing a better provisioning of the networking infrastructure.

1.3.2. Applications

Applications that use the ALTO Service can benefit in multiple ways. For example, they may no longer need to infer topology information, and some applications can reduce reliance on measuring path performance metrics themselves. They can take advantage of the ISP's knowledge to avoid bottlenecks and boost performance.

An example type of application is a Peer-to-Peer overlay where peer selection can be improved by including ALTO information in the selection process.

2. Architecture

Two key design objectives of the ALTO Protocol are simplicity and extensibility. At the same time, it introduces additional techniques to address potential scalability and privacy issues. This section first introduces the terminology, and then defines the ALTO architecture and the ALTO Protocol's place in the overall architecture.

2.1. Terminology

We use the following terms defined in [RFC5693]: Application, Overlay Network, Peer, Resource, Resource Identifier, Resource Provider, Resource Consumer, Resource Directory, Transport Address, Host Location Attribute, ALTO Service, ALTO Server, ALTO Client, ALTO Query, ALTO Reply, ALTO Transaction, Local Traffic, Peering Traffic, Transit Traffic.

We also use the following additional terms: Endpoint Address, Autonomous System Number (ASN), and Network Location.

2.1.1. Endpoint Address

An endpoint address represents the communication address of an endpoint. An endpoint address can be network-attachment based (IP address) or network-attachment agnostic. Common forms of endpoint

addresses include IP address, MAC address, overlay ID, and phone number.

Each Endpoint Address has an associated Address Type, which indicates both its syntax and semantics.

2.1.2. ASN

An Autonomous System Number.

2.1.3. Network Location

Network Location is a generic term denoting a single endpoint or group of endpoints.

2.1.4. ALTO Information

ALTO Information is a generic term referring to the network information sent by an ALTO Server.

2.1.5. ALTO Information Base

Internal representation of the ALTO Information maintained by the ALTO Server. Note that the structure of this internal representation is not defined by this document.

2.2. ALTO Service and Protocol Scope

An ALTO Server conveys the network information from the perspective of a network region; the ALTO Server presents its "my-Internet View" of the network region. In particular, an ALTO Server defines network Endpoints (and aggregations thereof) and generic costs amongst them, both from the network region's own perspective. A network region in this context can be an Autonomous System, an ISP, or perhaps a smaller region or set of ISPs; the details depend on the deployment scenario and discovery mechanism.

To better understand the ALTO Service and the role of the ALTO Protocol, we show in Figure 1 the overall system architecture. In this architecture, an ALTO Server prepares ALTO Information; an ALTO Client uses ALTO Service Discovery to identify an appropriate ALTO Server; and the ALTO Client requests available ALTO Information from the ALTO Server using the ALTO Protocol.

The ALTO Information provided by the ALTO Server can be updated dynamically based on network conditions, or can be seen as a policy which is updated at a larger time-scale.

More specifically, the ALTO Information provided by an ALTO Server may be influenced (at the operator's discretion) by other systems. Examples include (but are not limited to) static network configuration databases, dynamic network information, routing protocols, provisioning policies, and interfaces to outside parties. These components are shown in the figure for completeness but outside the scope of this specification.

Note that it may also be possible for ALTO Servers to exchange network information with other ALTO Servers (either within the same administrative domain or another administrative domain with the consent of both parties) in order to adjust exported ALTO Information. Such a protocol is also outside the scope of this specification.

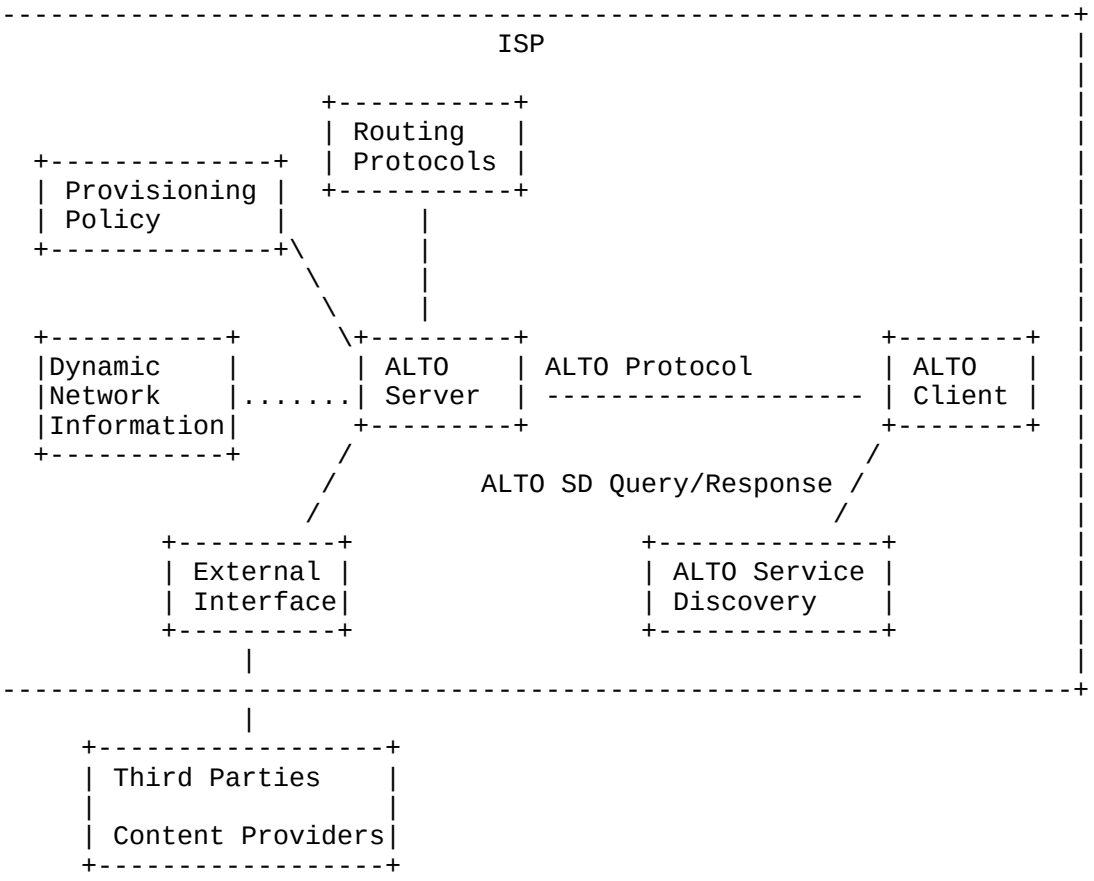


Figure 1: Basic ALTO Architecture.

3. Protocol Structure

The ALTO Protocol uses a simple extensible framework to convey network information. In the general framework, the ALTO protocol will convey properties on both Network Locations and the paths between Network Locations.

In this document, we focus on a particular Endpoint property to denote the location of an endpoint, and provider-defined costs for paths between pairs of Network Locations.

The ALTO Protocol is built on a common transport protocol, messaging structure and encoding, and transaction model. The protocol is subdivided into services of related functionality. ALTO-Core provides the Server Information Service and the Map Service to provide ALTO Information. Other ALTO Information services can provide additional functionality. There are three such services defined in this document: the Map Filtering Service, Endpoint Property Service, and Endpoint Cost Service. Additional services may be defined in companion documents. Note that functionality offered in different services are not totally non-overlapping (e.g., the Map Service and Map Filtering Service).

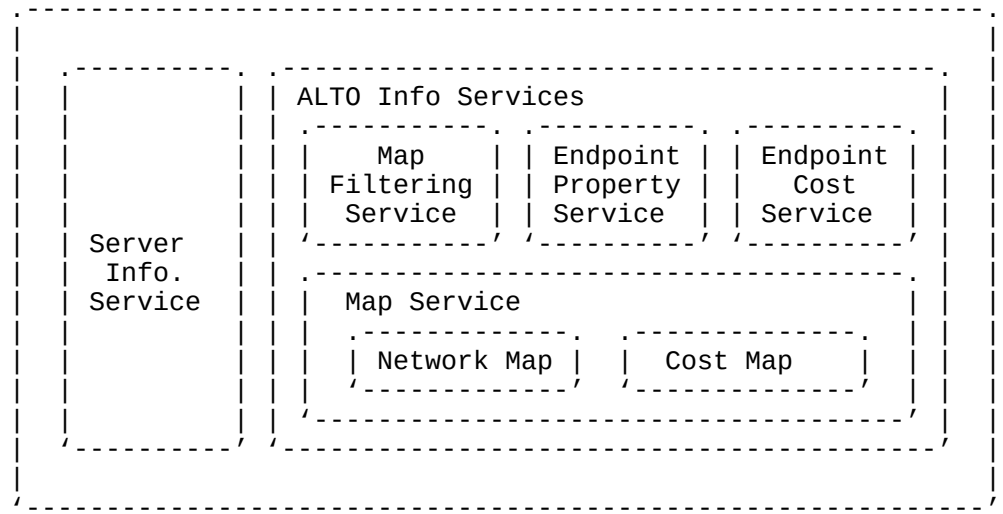


Figure 2: ALTO Protocol Structure

3.1. Server Information Service

The Server Information Service lists the details on the information that can be provided by an ALTO Server and perhaps other ALTO Servers maintained by the network provider. The configuration includes, for example, details about the operations and cost metrics supported by the ALTO Server and other related ALTO Servers that may be usable by an ALTO Client.

3.2. ALTO Information Services

Multiple, distinct services are defined to allow ALTO Clients to query ALTO Information from an ALTO Server. The ALTO Server internally maintains an ALTO Information Base that encodes the network provider's preferences. The ALTO Information Base encodes the Network Locations defined by the ALTO Server (and their corresponding properties), as well as the provider-defined costs between pairs of Network Locations.

3.2.1. Map Service

The Map Service provides batch information to ALTO Clients in the form of Network Map and Cost Map. The Network Map (See Section 4) provides the full set of Network Location groupings defined by the ALTO Server and the endpoints contained with each grouping. The Cost Map (see Section 5) provides costs between the defined groupings.

These two maps can be thought of (and implemented as) as simple files with appropriate encoding provided by the ALTO Server.

3.2.2. Map Filtering Service

Resource constrained ALTO Clients may benefit from query results being filtered at the ALTO Server. This avoids an ALTO Client spending network bandwidth or CPU collecting results and performing client-side filtering. The Map Filtering Service allows ALTO Clients to query for the ALTO Server Network Map and Cost Map based on additional parameters.

3.2.3. Endpoint Property Service

This service allows ALTO Clients to look up properties for individual endpoints. An example endpoint property is its Network Location (its grouping defined by the ALTO Server) or connectivity type (e.g., ADSL, Cable, or FTTH).

3.2.4. Endpoint Cost Service

Some ALTO Clients may also benefit from querying for costs and rankings based on endpoints. The Endpoint Cost Service allows an ALTO Server to return either numerical costs or ordinal costs (rankings) directly amongst Endpoints.

4. Network Map

In reality, many endpoints are very close to one another in terms of network connectivity, for example, endpoints on the same site of an enterprise. By treating a group of endpoints together as a single entity in ALTO, we can achieve much greater scalability without losing critical information.

The Network Location endpoint property allows an ALTO Server to group endpoints together to indicate their proximity. The resulting set of groupings is called the ALTO Network Map.

The definition of proximity varies depending on the granularity of the ALTO information configured by the provider. In one deployment, endpoints on the same subnet may be considered close; while in another deployment, endpoints connected to the same PoP may be considered close.

As used in this document, the Network Map refers to the syntax and semantics of the information distributed by the ALTO Server. This document does not discuss the internal representation of this data structure within the ALTO Server.

4.1. PID

Each group of Endpoints is identified by a provider-defined Network Location identifier called a PID. There can be many different ways of grouping the endpoints and assigning PIDs.

A PID is an identifier that provides an indirect and network-agnostic way to specify an aggregation of network endpoints that may be treated similarly, based on network topology, type, or other properties. For example, a PID may be defined by the ALTO service provider to denote a subnet, a set of subnets, a metropolitan area, a PoP, an autonomous system, or a set of autonomous systems. Aggregation of endpoints into PIDs can indicate proximity and can improve scalability. In particular, network preferences (costs) may be specified between PIDs, allowing cost information to be more compactly represented and updated at a faster time scale than the network aggregations themselves.

Using PIDs, the Network Map may also be used to communicate simple preferences with only minimal information from the Cost Map. For example, an ISP may prefer that endpoints associated with the same PoP (Point-of-Presence) in a P2P application communicate locally instead of communicating with endpoints in other PoPs. The ISP may aggregate endhosts within a PoP into a single PID in the Network Map. The Cost Map may be encoded to indicate that peering within the same PID is preferred; for example, cost(PID_i, PID_i) == c* and cost(PID_i, PID_j) > c* for i != j. Section 5 provides further details about Cost Map structure.

4.2. Endpoint Addresses

Communicating endpoints may have many types of addresses, such as IP addresses, MAC addresses, or overlay IDs. The current specification only considers IP addresses.

4.2.1. IP Addresses

The endpoints aggregated into a PID are denoted by a list of IP prefixes. When either an ALTO Client or ALTO Server needs to determine which PID in a Network Map contains a particular IP address, longest-prefix matching MUST be used.

A Network Map MUST define a PID for each possible address in the IP address space for all of the address types contained in the map. A RECOMMENDED way to satisfy this property is to define a PID that contains the 0.0.0.0/0 prefix for IPv4 or ::/0 (for IPv6).

Each endpoint MUST map into exactly one PID. Since longest-prefix matching is used to map an endpoint to a PID, this can be accomplished by ensuring that no two PIDs contain an identical IP prefix.

4.3. Example Network Map

Figure 3 illustrates an example Network Map. PIDs are used to identify network-agnostic aggregations.

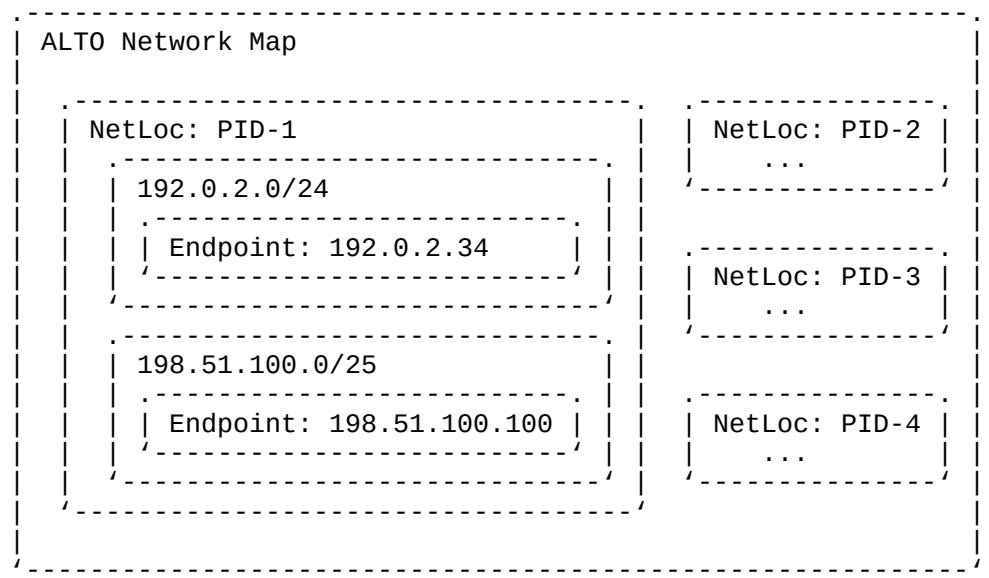


Figure 3: Example Network Map

5. Cost Map

An ALTO Server indicates preferences amongst network locations in the form of Path Costs. Path Costs are generic costs and can be internally computed by a network provider according to its own needs.

An ALTO Cost Map defines Path Costs pairwise amongst sets of source and destination Network Locations.

One advantage of separating ALTO information into a Network Map and a Cost Map is that the two components can be updated at different time scales. For example, Network Maps may be stable for a longer time while Cost Maps may be updated to reflect dynamic network conditions.

As used in this document, the Cost Map refers to the syntax and semantics of the information distributed by the ALTO Server. This document does not discuss the internal representation of this data structure within the ALTO Server.

5.1. Cost Attributes

Path Costs have attributes:

- Type: identifies what the costs represent;

- Mode: identifies how the costs should be interpreted.

Certain queries for Cost Maps allow the ALTO Client to indicate the desired Type and Mode.

5.1.1. Cost Type

The Type attribute indicates what the cost represents. For example, an ALTO Server could define costs representing air-miles, hop-counts, or generic routing costs.

Cost types are indicated in protocol messages as strings.

5.1.1.1. Cost Type: routingcost

An ALTO Server MUST define the 'routingcost' Cost Type.

This Cost Type conveys a generic measure for the cost of routing traffic from a source to a destination. Lower values indicate a higher preference for traffic to be sent from a source to a destination.

Note that an ISP may internally compute routing cost using any method it chooses (e.g., air-miles or hop-count) as long as it conforms to these semantics.

5.1.2. Cost Mode

The Mode attribute indicates how costs should be interpreted. Specifically, the Mode attribute indicates whether returned costs should be interpreted as numerical values or ordinal rankings.

It is important to communicate such information to ALTO Clients, as certain operations may not be valid on certain costs returned by an ALTO Server. For example, it is possible for an ALTO Server to return a set of IP addresses with costs indicating a ranking of the IP addresses. Arithmetic operations, such as summation, that would make sense for numerical values, do not make sense for ordinal rankings. ALTO Clients may handle such costs differently.

Cost Modes are indicated in protocol messages as strings.

An ALTO Server MUST support at least one of 'numerical' and 'ordinal' costs. ALTO Clients SHOULD be cognizant of operations when a desired cost mode is not supported. For example, an ALTO Client desiring numerical costs may adjust behavior if only the ordinal Cost Mode is

available. Alternatively, an ALTO Client desiring ordinal costs may construct ordinal costs given numerical values if only the numerical Cost Mode is available.

5.1.2.1. Cost Mode: numerical

This Cost Mode is indicated by the string 'numerical'. This mode indicates that it is safe to perform numerical operations (e.g. summation) on the returned costs.

5.1.2.2. Cost Mode: ordinal

This Cost Mode is indicated by the string 'ordinal'. This mode indicates that the costs values to a set of Destination Network Locations from a particular Source Network Location are a ranking, with lower values indicating a higher preference. The values are non-negative integers. Ordinal cost values from a particular Source Network Location to a set of Destination Network Locations need not be unique nor contiguous. In particular, from the perspective of a particular Source Network Location, two Destination Network Locations may have an identical rank (ordinal cost value). This document does not specify any behavior by an ALTO Client in this case; an ALTO Client may decide to break ties by random selection, other application knowledge, or some other means.

It is important to note that the values in the Cost Map provided with the ordinal Cost Mode are not necessarily the actual cost known to the ALTO Server.

5.2. Cost Map Structure

A query for a Cost Map either explicitly or implicitly includes a list of Source Network Locations and a list of Destination Network Locations. (Recall that a Network Location can be an endpoint address or a PID.)

Specifically, assume that a query has a list of multiple Source Network Locations, say [Src_1, Src_2, ..., Src_m], and a list of multiple Destination Network Locations, say [Dst_1, Dst_2, ..., Dst_n].

The ALTO Server will return the Path Cost for each communicating pair (i.e., Src_1 -> Dst_1, ..., Src_1 -> Dst_n, ..., Src_m -> Dst_1, ..., Src_m -> Dst_n). We refer to this structure as a Cost Map.

If the Cost Mode is 'ordinal', the Path Cost of each communicating pair is relative to the m*n entries.

5.3. Network Map and Cost Map Dependency

If a Cost Map contains PIDs in the list of Source Network Locations or the list of Destination Network Locations, the Path Costs are generated based on a particular Network Map (which defines the PIDs). Version Tags are introduced to ensure that ALTO Clients are able to use consistent information even though the information is provided in two maps.

A Version Tag is an opaque string associated with a Network Map maintained by the ALTO Server. When the Network Map changes, the Version Tag MUST also be changed. (Thus, the Version Tag is defined similarly to HTTP's Entity Tags; see Section 3.11 of [RFC2616].) Possibilities for generating a Version Tag include the last-modified timestamp for the Network Map, or a hash of its contents.

A Network Map distributed by the ALTO Server includes its Version Tag. A Cost Map referring to PIDs also includes the Version Tag of the Network Map on which it is based.

6. Protocol Design Overview

The ALTO Protocol design uses a REST-ful design with the goal of leveraging current HTTP [RFC2616] implementations and infrastructure. The REST-ful design supports flexible deployment strategies and provides extensibility. ALTO requests and responses are encoded with JSON [RFC4627].

6.1. Benefits

Benefits enabled by these design choices include easier understanding and debugging, mature libraries, tools, infrastructure, and caching and redistribution of ALTO information for increased scalability.

6.1.1. Existing Infrastructure

HTTP is a natural choice for integration with existing applications and infrastructure. In particular, the ALTO Protocol design leverages:

- the huge installed base of infrastructure, including HTTP caches,
- mature software implementations,
- the fact that many P2P clients already have an embedded HTTP client, and

- o authentication and encryption mechanisms in HTTP and SSL/TLS.

6.1.2. ALTO Information Reuse and Redistribution

ALTO information may be useful to a large number of applications and users. For example, an identical Network Map may be used by all ALTO Clients querying a particular ALTO Server. At the same time, distributing ALTO information must be efficient and not become a bottleneck.

Beyond integration with existing HTTP caching infrastructure, ALTO information may also be cached or redistributed using application-dependent mechanisms, such as P2P DHTs or P2P file-sharing. This document does not define particular mechanisms for such redistribution, but it does define the primitives (e.g., digital signatures) needed to support such a mechanism. See [I-D.gu-alto-redistribution] for further discussion.

Note that if caching or redistribution is used, the response message may be returned from another (possibly third-party) entity. Reuse and Redistribution is further discussed in Section 12.4. Protocol support for redistribution is specified in Section 8.

6.2. Protocol Design

The ALTO Protocol uses a REST-ful design. There are two primary components to this design:

- o Information Resources: Each service provides network information as a set of resources, which are distinguished by their media types [RFC2046]. An ALTO Client may construct an HTTP request for a particular resource (including any parameters, if necessary), and an ALTO Server returns the requested resource in an HTTP response.

- o Information Resource Directory: An ALTO Server provides to ALTO Clients a list of available resources and the URI at which each is provided. This document refers to this list as the Information Resource Directory. This directory is the single entry point to an ALTO Service. ALTO Clients consult the directory to determine the services provided by an ALTO Server.

7. Protocol Specification

This section first specifies general client and server processing, followed by a detailed specification for each ALTO Information Resource.

7.1. Notation

This document uses an adaptation of the C-style struct notation to define the required and optional members of JSON objects. Unless explicitly noted, each member of a struct is REQUIRED.

The types 'JSONString', 'JSONNumber', 'JSONBool' indicate the JSON string, number, and boolean types, respectively.

Note that no standard, machine-readable interface definition or schema is provided. Extension documents may document these as necessary.

7.2. Basic Operation

The ALTO Protocol employs standard HTTP [RFC2616]. It is used for discovering available Information Resources at an ALTO Server and retrieving Information Resources. ALTO Clients and ALTO Servers use HTTP requests and responses carrying ALTO-specific content with encoding as specified in this document, and MUST be compliant with [RFC2616].

7.2.1. Discovering Information Resources

To discover available resources, an ALTO Client may request the Information Resource Directory, which an ALTO Server provides at the URI found by the ALTO Discovery protocol.

Informally, an Information Resource Directory enumerates URIs at which an ALTO Server offers Information Resources. Each entry in the directory indicates a URI at which an ALTO Server accepts requests, and returns either the requested Information Resource or an Information Resource Directory that references additional Information Resources. See Section 7.6 for a detailed specification.

7.2.2. Requesting Information Resources

Through the retrieved Information Resource Directories, an ALTO Client can determine whether an ALTO Server supports the desired Information Resource, and if it is supported, the URI at which it is available.

Where possible, the ALTO Protocol uses the HTTP GET method to request resources. However, some ALTO services provide Information Resources that are the function of one or more input parameters. Input parameters are encoded in the HTTP request's entity body, and the request uses the HTTP POST method.

Note that it is possible for an ALTO Server to employ caching for the response to a POST request. This can be accomplished by returning an HTTP 303 status code ("See Other") indicating to the ALTO Client that the resulting Cost Map is available via a GET request to an alternate URL (which may be cached).

When requesting an ALTO Information Resource that requires input parameters specified in a HTTP POST request, an ALTO Client MUST set the Content-Type HTTP header to the media type corresponding to the format of the supplied input parameters.

7.2.3. Response

Upon receiving a request, an ALTO server either returns the requested resource, provides the ALTO Client an Information Resource Directory indicating how to reach the desired resource, or returns an error.

The type of response MUST be indicated by the media type attached to the response (the Content-Type HTTP header). If an ALTO Client receives an Information Resource Directory, it can consult the received directory to determine if any of the offered URIs contain the desired Information Resource.

The generic encoding for an Information Resource is specified in Section 7.3.

Errors are indicated via either ALTO-level error codes, or via HTTP status codes; see Section 7.4.

7.2.4. Client Behavior

7.2.4.1. Using Information Resources

This specification does not indicate any required actions taken by ALTO Clients upon successfully receiving an Information Resource from an ALTO Server. Although ALTO Clients are suggested to interpret the received ALTO Information and adapt application behavior, ALTO Clients are not required to do so.

7.2.4.2. Error Conditions

If an ALTO Client does not successfully receive a desired Information Resource from a particular ALTO Server, it can either choose another server (if one is available) or fall back to a default behavior (e.g., perform peer selection without the use of ALTO information). An ALTO Client may also retry the request at a later time.

7.2.5. Authentication and Encryption

An ALTO Server MAY support SSL/TLS to implement server and/or client authentication, as well as encryption. See [RFC6125] for considerations regarding verification of server identity.

7.2.6. HTTP Cookies

If cookies are included in an HTTP request received by an ALTO Server, they MUST be ignored.

7.2.7. Parsing

This document only details object members used by this specification. Extensions may include additional members within JSON objects defined in this document. ALTO implementations MUST ignore such unknown fields when processing ALTO messages.

7.3. Information Resource

An Information Resource is an HTTP entity body received by an ALTO Server that encodes the ALTO Information desired by an ALTO Client.

This document specifies multiple Information Resources that can be provided by an ALTO Server. Each Information Resource has certain attributes associated with it, indicating its data format, the input parameters it supports, and format of the input parameters.

7.3.1. Capabilities

An ALTO Server may advertise to an ALTO Client that it supports certain capabilities in requests for an Information Resource. For example, if an ALTO Server allows requests for a Cost Map to include constraints, it may advertise that it supports this capability.

7.3.2. Input Parameters Media Type

An ALTO Server may allow an ALTO Client to supply input parameters when requesting certain Information Resources. The format of the input parameters (i.e., as contained in the entity body of the HTTP POST request) is indicated by the media type [RFC2046].

7.3.3. Media Type

The media type [RFC2046] uniquely indicates the data format of the Information Resource as returned by an ALTO Server in the HTTP entity body.

7.3.4. Encoding

Though each Information Resource may have a distinct syntax, they are designed to have a common structure containing generic ALTO-layer metadata about the resource, as well as data itself.

An Information Resource has a single top-level JSON object of type InfoResourceEntity:

```
object {
  InfoResourceMetaData   meta;
  [InfoResourceDataType] data;
} InfoResourceEntity;
```

with members:

meta meta-information pertaining to the Information Resource

data the data contained in the Information Resource

7.3.4.1. Meta Information

Meta information is encoded as a JSON object with type InfoResourceMetaData:

```
object {
  InfoResourceRedistDesc redistribution;    [OPTIONAL]
} InfoResourceMetaData;
```

with members:

redistribution Additional data for use in Information Resources that may be redistributed amongst ALTO Clients. See Section 8.

7.3.4.2. ALTO Information

The "data" member of the InfoResourceEntity encodes the resource-specific data; the structure of this member is detailed later in this section for each particular Information Resource.

7.3.4.3. Signature

An ALTO Server MAY additionally supply a signature asserting that it generated a particular response. See Section 8.2.2.

7.3.4.4. Example

The following is an example of the encoding for an Information Resource:

```
HTTP/1.1 200 OK
Content-Length: [TODO]
Content-Type: application/alto-costmap+json

{
  "meta" : {
    "redistribution" : { ... }
  },
  "data" : {
    ...
  }
}
```

7.4. ALTO Errors

If there is an error processing a request, an ALTO Server SHOULD return additional ALTO-layer information, if it is available, in the form of an ALTO Error Resource encoded in the HTTP response's entity body.

If no ALTO-layer information is available, an ALTO Server may omit an ALTO Error resource from the response. An appropriate HTTP status code MUST be set.

It is important to note that the HTTP Status Code and ALTO Error Code have distinct roles. An ALTO Error Code provides detailed information about the why a particular request for an ALTO Resource was not successful. The HTTP status code indicates to HTTP processing elements (e.g., intermediaries and clients) how the response should be treated.

7.4.1. Media Type

The media type for an Error Resource is "application/alto-error+json".

7.4.2. Resource Format

An Error Resource has the format:

```
object {
  JSONString code;
  JSONString reason;    [OPTIONAL]
} ErrorResourceEntity;
```

where:

code An ALTO Error Code defined in Table 1

reason A (free-form) human-readable explanation of the particular error

7.4.3. Error Codes

This document defines ALTO Error Codes to support the error conditions needed for purposes of this document. Additional status codes may be defined in companion or extension documents.

The HTTP status codes corresponding to each ALTO Error Code are defined to provide correct behavior with HTTP intermediaries and clients. When an ALTO Server returns a particular ALTO Error Code, it MUST indicate one of the corresponding HTTP status codes in Table 1in the HTTP response.

If multiple errors are present in a single request (e.g., a request uses a JSONString when a JSONInteger is expected and a required field is missing), then the ALTO Server MUST return exactly one of the detected errors. However, the reported error is implementation defined, since specifying a particular order for message processing encroaches needlessly on implementation technique.

ALTO Error Code	HTTP Status Code(s)	Description
E_SYNTAX	400	Parsing error in request (including identifiers)
E_JSON_FIELD_MISSING	400	Required field missing
E_JSON_VALUE_TYPE	400	JSON Value of unexpected type
E_INVALID_COST_MODE	400	Invalid cost mode
E_INVALID_COST_TYPE	400	Invalid cost type

| E_INVALID_PROPERTY_TYPE | 400 | Invalid property type |
+-------------------------+-------------+---------------------------+

Table 1: Defined ALTO Error Codes

7.5. ALTO Types

This section details the format for particular data values used in the ALTO Protocol.

7.5.1. PID Name

A PID Name is encoded as a US-ASCII string. The string MUST be no more than 64 characters, and MUST NOT contain characters other than alphanumeric characters or the '.' separator. The '.' separator is reserved for future use and MUST NOT be used unless specifically indicated by a companion or extension document.

The type 'PIDName' is used in this document to indicate a string of this format.

7.5.2. Endpoints

This section defines formats used to encode addresses for Endpoints. In a case that multiple textual representations encode the same Endpoint address or prefix (within the guidelines outlined in this document), the ALTO Protocol does not require ALTO Clients or ALTO Servers to use a particular textual representation, nor does it require that ALTO Servers reply to requests using the same textual representation used by requesting ALTO Clients. ALTO Clients must be cognizant of this.

7.5.2.1. Address Type

Address Types are encoded as US-ASCII strings consisting of only alphanumeric characters. This document defines the address type "ipv4" to refer to IPv4 addresses, and "ipv6" to refer to IPv6 addresses. Extension documents may define additional Address Types.

The type 'AddressType' is used in this document to indicate a string of this format.

7.5.2.2. Endpoint Address

Endpoint Addresses are encoded as US-ASCII strings. The exact characters and format depend on the type of endpoint address.

The type 'EndpointAddr' is used in this document to indicate a string

of this format.

7.5.2.2.1. IPv4

IPv4 Endpoint Addresses are encoded as specified by the 'IPv4address' rule in Section 3.2.2 of [RFC3986].

7.5.2.2.2. IPv6

IPv6 Endpoint Addresses are encoded as specified in Section 4 of [RFC5952].

7.5.2.2.3. Typed Endpoint Addresses

When an Endpoint Address is used, an ALTO implementation must be able to determine its type. For this purpose, the ALTO Protocol allows endpoint addresses to also explicitly indicate their type.

Typed Endpoint Addresses are encoded as US-ASCII strings of the format 'AddressType:EndpointAddr' (with the ':' character as a separator). The type 'TypedEndpointAddr' is used to indicate a string of this format.

7.5.2.3. Endpoint Prefixes

For efficiency, it is useful to denote a set of Endpoint Addresses using a special notation (if one exists). This specification makes use of the prefix notations for both IPv4 and IPv6 for this purpose.

Endpoint Prefixes are encoded as US-ASCII strings. The exact characters and format depend on the type of endpoint address.

The type 'EndpointPrefix' is used in this document to indicate a string of this format.

7.5.2.3.1. IPv4

IPv4 Endpoint Prefixes are encoded as specified in Section 3.1 of [RFC4632].

7.5.2.3.2. IPv6

IPv6 Endpoint Prefixes are encoded as specified in Section 7 of [RFC5952].

7.5.2.4. Endpoint Address Group

The ALTO Protocol includes messages that specify potentially large sets of endpoint addresses. Endpoint Address Groups provide a more efficient way to encode such sets, even when the set contains endpoint addresses of different types.

An Endpoint Address Group is defined as:

```
object {
  EndpointPrefix [AddressType]<0..*>;
  ...
} EndpointAddrGroup;
```

In particular, an Endpoint Address Group is a JSON object with the name of each member being the string corresponding to the address type, and the member's corresponding value being a list of prefixes of addresses of that type.

The following is an example with both IPv4 and IPv6 endpoint addresses:

```
{
  "ipv4": [
    "192.0.2.0/24",
    "198.51.100.0/25"
  ],
  "ipv6": [
    "2001:db8:0:1::/64",
    "2001:db8:0:2::/64"
  ]
}
```

7.5.3. Cost Mode

A Cost Mode is encoded as a US-ASCII string. The string MUST either have the value 'numerical' or 'ordinal'.

The type 'CostMode' is used in this document to indicate a string of this format.

7.5.4. Cost Type

A Cost Type is encoded as a US-ASCII string. The string MUST be no more than 32 characters, and MUST NOT contain characters other than alphanumeric characters, the hyphen ('-'), or the ':' separator.

Identifiers prefixed with 'priv:' are reserved for Private Use [RFC5226]. Identifiers prefixed with 'exp:' are reserved for Experimental use. All other identifiers appearing in an HTTP request or response with an 'application/alto-*' media type MUST be registered in the ALTO Cost Types registry Section 11.2.

The type 'CostType' is used in this document to indicate a string of this format.

7.5.5. Endpoint Property

An Endpoint Property is encoded as a US-ASCII string. The string MUST be no more than 32 characters, and MUST NOT contain characters other than alphanumeric characters, the hyphen ('-'), or the ':' separator.

Identifiers prefixed with 'priv:' are reserved for Private Use [RFC5226]. Identifiers prefixed with 'exp:' are reserved for Experimental use. All other identifiers appearing in an HTTP request or response with an 'application/alto-*' media type MUST be registered in the ALTO Endpoint Property registry Section 11.3.

The type 'EndpointProperty' is used in this document to indicate a string of this format.

7.6. Information Resource Directory

An Information Resource Directory indicates to ALTO Clients which Information Resources are made available by an ALTO Server.

Since resource selection happens after consumption of the Information Resource Directory, the format of the Information Resource Directory is designed to be simple with the intention of future ALTO Protocol versions maintaining backwards compatibility. Future extensions or versions of the ALTO Protocol SHOULD be accomplished by extending existing media types or adding new media types, but retaining the same format for the Information Resource Directory.

An ALTO Server MUST make an Information Resource Directory available via the HTTP GET method to a URI discoverable by an ALTO Client. Discovery of this URI is out of scope of this document, but could be accomplished by manual configuration or by returning the URI of an

Information Resource Directory from the ALTO Discovery Protocol [I-D.ietf-alto-server-discovery].

7.6.1. Media Type

The media type is "application/alto-directory+json".

7.6.2. Encoding

An Information Resource Directory is a JSON object of type InfoResourceDirectory:

```
object {
  ...
} Capabilities;

object {
  JSONString   uri;
  JSONString   media-types<1..*>;
  JSONString   accepts<0..*>;        [OPTIONAL]
  Capabilities capabilities;         [OPTIONAL]
} ResourceEntry;

object {
  ResourceEntry resources<0..*>;
} InfoResourceDirectory;
```

where the "resources" array indicates a list of Information Resources provided by an ALTO Server. Note that the list of available resources is enclosed in a JSON object for extensibility; future protocol versions may specify additional members in the InfoResourceDirectory object.

Each entry MUST indicate a URI that either directly provides the indicated Information Resource, or responds to a HTTP OPTIONS request which provides an Information Resource Directory with entries of additional Information Resources.

If an ALTO Client makes a GET or POST request to a URI that does not directly provide an indicated Information Resource, the ALTO Server MUST either reply with an HTTP 300 status code ("Multiple Choices") and an Information Resource Directory in the HTTP response’s entity body, or indicate an appropriate HTTP status code. Note that in general, it is preferred that ALTO Clients use HTTP OPTIONS requests to discover additional Information Resources.

A URI that directly provides an Information Resource MAY also respond to HTTP OPTIONS requests, but it is not required to do so (in which case, it MUST respond with HTTP 405 status code ("Method Not Allowed"). This allows certain Information Resources to be configured as static files with minimal configuration on some HTTP servers.

Each entry in the directory specifies:

uri A URI at which the ALTO Server provides one or more Information Resources, or an Information Resource Directory indicating additional Information Resources.

media-types The list of all media types of Information Resources (see Section 7.3.3) available via GET or POST requests to the corresponding URI or URIs discoverable via the URI.

accepts The list of all media types of input parameters (see Section 7.3.2) accepted by POST requests to the corresponding URI or URIs discoverable via the URI. If this member is not present, it MUST be assumed to be an empty array.

capabilities A JSON Object enumerating capabilities of an ALTO Server in providing the Information Resource at the corresponding URI and Information Resources discoverable via the URI. If this member is not present, it MUST be assumed to be an empty array. If a capability for one of the offered Information Resources is not explicitly listed here, an ALTO Client may either issue an OPTIONS HTTP request to the corresponding URI to determine if the capability is supported, or assume its default value.

If an entry has an empty list for "accepts", then the corresponding URI MUST support GET requests. If an entry has a non-empty list for "accepts", then the corresponding URI MUST support POST requests. If an ALTO Server wishes to support both GET and POST on a single URI, it MUST specify two entries in the Information Resource Directory.

7.6.3. Example

The following is an example Information Resource Directory returned by an ALTO Server. In this example, the ALTO Server provides additional Network and Cost Maps via a separate subdomain, "custom.alto.example.com". The maps available via this subdomain are Filtered Network and Cost Maps as well as pre-generated maps for the "hopcount" and "routingcost" Cost Types in the "ordinal" Cost Mode.

An ALTO Client can discover the maps available by "custom.alto.example.com" by successfully performing an OPTIONS

request to "http://custom.alto.example.com/maps".

```
GET /directory HTTP/1.1
Host: alto.example.com
Accept: application/alto-directory+json,application/alto-error+json
```

```
HTTP/1.1 200 OK
Content-Length: [TODO]
Content-Type: application/alto-directory+json

{
  "resources" : [
    {
      "uri" : "http://alto.example.com/serverinfo",
      "media-types" : [ "application/alto-serverinfo+json" ]
    }, {
      "uri" : "http://alto.example.com/networkmap",
      "media-types" : [ "application/alto-networkmap+json" ]
    }, {
      "uri" : "http://alto.example.com/costmap/num/routingcost",
      "media-types" : [ "application/alto-costmap+json" ],
      "capabilities" : {
        "cost-modes" : [ "numerical" ],
        "cost-types" : [ "routingcost" ]
      }
    }, {
      "uri" : "http://alto.example.com/costmap/num/hopcount",
      "media-types" : [ "application/alto-costmap+json" ],
      "capabilities" : {
        "cost-modes" : [ "numerical" ],
        "cost-types" : [ "hopcount" ]
      }
    }, {
      "uri" : "http://custom.alto.example.com/maps",
      "media-types" : [
        "application/alto-networkmap+json",
        "application/alto-costmap+json"
      ],
      "accepts" : [
        "application/alto-networkmapfilter+json",
        "application/alto-costmapfilter+json"
      ]
    }, {
      "uri" : "http://alto.example.com/endpointprop/lookup",
```

```
        "media-types" : [ "application/alto-endpointprop+json" ],
        "accepts" : [ "application/alto-endpointpropparams+json" ],
        "capabilities" : {
          "prop-types" : [ "pid" ]
        }
      }, {
        "uri" : "http://alto.example.com/endpointcost/lookup",
        "media-types" : [ "application/alto-endpointcost+json" ],
        "accepts" : [ "application/alto-endpointcostparams+json" ],
        "capabilities" : {
          "cost-constraints" : true,
          "cost-modes" : [ "ordinal", "numerical" ],
          "cost-types" : [ "routingcost", "hopcount" ]
        }
      }
    ]
  }
```

```
  OPTIONS /maps HTTP/1.1
  Host: custom.alto.example.com
  Accept: application/alto-directory+json,application/alto-error+json
```

```
HTTP/1.1 200 OK
Content-Length: [TODO]
Content-Type: application/alto-directory+json

{
  "resources" : [
    {
      "uri" : "http://custom.alto.example.com/networkmap/filtered",
      "media-types" : [ "application/alto-networkmap+json" ],
      "accepts" : [ "application/alto-networkmapfilter+json" ]
    }, {
      "uri" : "http://custom.alto.example.com/costmap/filtered",
      "media-types" : [ "application/alto-costmap+json" ],
      "accepts" : [ "application/alto-costmapfilter+json" ],
      "capabilities" : {
        "cost-constraints" : true,
        "cost-modes" : [ "ordinal", "numerical" ],
        "cost-types" : [ "routingcost", "hopcount" ]
      }
    }, {
      "uri" : "http://custom.alto.example.com/ord/routingcost",
      "media-types" : [ "application/alto-costmap+json" ],
      "capabilities" : {
        "cost-modes" : [ "ordinal" ],
        "cost-types" : [ "routingcost" ]
      }
    }, {
      "uri" : "http://custom.alto.example.com/ord/hopcount",
      "media-types" : [ "application/alto-costmap+json" ],
      "capabilities" : {
        "cost-modes" : [ "ordinal" ],
        "cost-types" : [ "hopcount" ]
      }
    }
  ]
}
```

7.6.4. Usage Considerations

7.6.4.1. ALTO Client

This document specifies no requirements or constraints on ALTO Clients with regards to how they process an Information Resource Directory to identify the URI corresponding to a desired Information Resource. However, some advice is provided for implementors.

It is possible that multiple entries in the directory match a desired

Information Resource. For instance, in the example in Section 7.6.3, a full Cost Map with "numerical" Cost Mode and "routingcost" Cost Type could be retrieved via a GET request to "http://alto.example.com/costmap/num/routingcost", or via a POST request to "http://custom.alto.example.com/costmap/filtered".

In general, it is preferred for ALTO Clients to use GET requests where appropriate, since it is more likely for responses to be cacheable.

7.6.4.2. ALTO Server

This document indicates that an ALTO Server may or may not provide the Information Resources specified in the Map Filtering Service. If these resources are not provided, it is indicated to an ALTO Client by the absence of a Network Map or Cost Map with any media types listed under "accepts".

7.7. Information Resources

This section documents the individual Information Resources defined in the ALTO Protocol.

7.7.1. Server Information Service

The Server Information Service provides generic information about an ALTO Server.

7.7.1.1. Server Info

This Information Resource MUST be provided by an ALTO Server.

7.7.1.1.1. Media Type

The media type is "application/alto-serverinfo+json".

7.7.1.1.2. HTTP Method

This resource is requested using the HTTP GET method.

7.7.1.1.3. Input Parameters

None.

7.7.1.1.4. Capabilities

None.

7.7.1.1.5. Response

The returned InfoResourceEntity object has "data" member of type InfoResourceServerInfo:

```
object {
  JSONString   service-id;            [OPTIONAL]
  JSONString   certificates<0..*>;    [OPTIONAL]
} InfoResourceServerInfo;
```

which has members:

service-id UUID [RFC4122] indicating an one or more ALTO Servers serving equivalent ALTO Information.

certificates List of PEM-encoded X.509 certificates used by the ALTO Server in the signing of responses.

If an ALTO Server has the possibility of marking any response as redistributable, the ’service-id’ and ’certificates’ fields are REQUIRED instead of OPTIONAL. See Section 8&$160;for detailed specification.

7.7.1.1.6. Example

```
GET /serverinfo HTTP/1.1
Host: alto.example.com
Accept: application/alto-serverinfo+json,application/alto-error+json
```

```
HTTP/1.1 200 OK
Content-Length: [TODO]
Content-Type: application/alto-serverinfo+json

{
  "meta" : {},
  "data" : {
    "service-id" : "c89ca72f-dead-41b5-9e2b-b65455ace1ee",
    "certificates" : [ ... ]
  }
}
```

7.7.2. Map Service

The Map Service provides batch information to ALTO Clients in the form of two types of maps: a Network Map and Cost Map.

7.7.2.1. Network Map

The Network Map Information Resource lists for each PID, the network locations (endpoints) within the PID. It MUST be provided by an ALTO Server.

7.7.2.1.1. Media Type

The media type is "application/alto-networkmap+json".

7.7.2.1.2. HTTP Method

This resource is requested using the HTTP GET method.

7.7.2.1.3. Input Parameters

None.

7.7.2.1.4. Capabilities

None.

7.7.2.1.5. Response

The returned InfoResourceEntity object "data" member of type InfoResourceNetworkMap:

```
object {
  EndpointAddrGroup [pidname]<0..*>;
  ...
} NetworkMapData;

object {
  JSONString     map-vtag;
  NetworkMapData map;
} InfoResourceNetworkMap;
```

with members:

map-vtag The Version Tag (Section 5.3) of the Network Map.

map The Network Map data itself.

NetworkMapData is a JSON object with each member representing a single PID and its associated set of endpoint addresses. A member's name is a string of type PIDName.

The returned Network Map MUST include all PIDs known to the ALTO Server.

7.7.2.1.6. Example

```
GET /networkmap HTTP/1.1
Host: alto.example.com
Accept: application/alto-networkmap+json,application/alto-error+json
```

```
HTTP/1.1 200 OK
Content-Length: [TODO]
Content-Type: application/alto-networkmap+json

{
  "meta" : {},
  "data" : {
    "map-vtag" : "1266506139",
    "map" : {
      "PID1" : {
        "ipv4" : [
          "192.0.2.0/24",
          "198.51.100.0/25"
        ]
      },
      "PID2" : {
        "ipv4" : [
          "198.51.100.128/25"
        ]
      },
      "PID3" : {
        "ipv4" : [
          "0.0.0.0/0"
        ],
        "ipv6" : [
          "::/0"
        ]
      }
    }
  }
}
```

7.7.2.2. Cost Map

The Cost Map resource lists the Path Cost for each pair of source/ destination PID defined by the ALTO Server for a given Cost Type and Cost Mode. This resource MUST be provided for at least the 'routingcost' Cost Type and 'numerical' Cost Mode.

Note that since this resource, an unfiltered Cost Map requested by an HTTP GET, does not indicate the desired Cost Mode or Cost Type as input parameters, an ALTO Server MUST indicate in an Information Resource Directory a unfiltered Cost Map Information Resource by specifying the capabilities (Section 7.7.2.2.4) with "cost-types" and "cost-modes" members each having a single element. This technique will allow an ALTO Client to determine a URI for an unfiltered Cost Map of the desired Cost Mode and Cost Type.

7.7.2.2.1. Media Type

The media type is "application/alto-costmap+json".

7.7.2.2.2. HTTP Method

This resource is requested using the HTTP GET method.

7.7.2.2.3. Input Parameters

None.

7.7.2.2.4. Capabilities

This resource may be defined for across multiple Cost Types and Cost Modes. The capabilities of an ALTO Server URI providing this resource are defined by a JSON Object of type CostMapCapability:

```
object {
  CostMode cost-modes<0..*>;
  CostType cost-types<0..*>;
} CostMapCapability;
```

with members:

cost-modes The Cost Modes (Section 5.1.2) supported by the corresponding URI. If not present, this member MUST be interpreted as an empty array.

cost-types The Cost Types (Section 5.1.1) supported by the corresponding URI. If not present, this member MUST be interpreted as an empty array.

An ALTO Server MUST support all of the Cost Types listed here for each of the listed Cost Modes. Note that an ALTO Server may provide multiple Cost Map Information Resources, each with different capabilities.

7.7.2.2.5. Response

The returned InfoResourceEntity object has "data" member of type InfoResourceCostMap:

```
object DstCosts {
  JSONNumber [PIDName];
  ...
};

object {
  DstCosts [PIDName]<0..*>;
  ...
} CostMapData;

object {
  CostMode    cost-mode;
  CostType    cost-type;
  JSONString  map-vtag;
  CostMapData map;
} InfoResourceCostMap;
```

with members:

cost-mode Cost Mode (Section 5.1.2) used in the Cost Map.

cost-type Cost Type (Section 5.1.1) used in the Cost Map.

map-vtag The Version Tag (Section 5.3) of the Network Map used to generate the Cost Map.

map The Cost Map data itself.

CostMapData is a JSON object with each member representing a single Source PID; the name for a member is the PIDName string identifying the corresponding Source PID. For each Source PID, a DstCosts object denotes the associated cost to a set of destination PIDs (Section 5.2); the name for each member in the object is the PIDName string identifying the corresponding Destination PID. DstCosts MUST have a single member for each Destination PID in the map.

The returned Cost Map MUST include Path Costs for each pair of Source PID and Destination PID known to the ALTO Server.

7.7.2.2.6. Example

```
GET /costmap/num/routingcost HTTP/1.1
Host: alto.example.com
Accept: application/alto-costmap+json,application/alto-error+json
```

```
HTTP/1.1 200 OK
Content-Length: [TODO]
Content-Type: application/alto-costmap+json

{
  "meta" : {},
  "data" : {
    "cost-mode" : "numerical",
    "cost-type" : "routingcost",
    "map-vtag"  : "1266506139",
    "map" : {
      "PID1": { "PID1": 1,  "PID2": 5,  "PID3": 10 },
      "PID2": { "PID1": 5,  "PID2": 1,  "PID3": 15 },
      "PID3": { "PID1": 20, "PID2": 15, "PID3": 1  }
    }
  }
}
```

7.7.3. Map Filtering Service

The Map Filtering Service allows ALTO Clients to specify filtering criteria to return a subset of the full maps available in the Map Service.

7.7.3.1. Filtered Network Map

A Filtered Network Map is a Network Map Information Resource (Section 7.7.2.1) for which an ALTO Client may supply a list of PIDs to be included. A Filtered Network Map MAY be provided by an ALTO Server.

7.7.3.1.1. Media Type

See Section 7.7.2.1.1.

7.7.3.1.2. HTTP Method

This resource is requested using the HTTP POST method.

7.7.3.1.3. Input Parameters

Input parameters are supplied in the entity body of the POST request. This document specifies the input parameters with a data format indicated by the media type "application/alto-networkmapfilter+json", which is a JSON Object of type ReqFilteredNetworkMap, where:

```
object {
  PIDName pids<0..*>;
} ReqFilteredNetworkMap;
```

with members:

pids Specifies list of PIDs to be included in the returned Filtered Network Map. If the list of PIDs is empty, the ALTO Server MUST interpret the list as if it contained a list of all currently-defined PIDs. The ALTO Server MUST interpret entries appearing multiple times as if they appeared only once.

7.7.3.1.4. Capabilities

None.

7.7.3.1.5. Response

See Section 7.7.2.1.5 for the format.

The ALTO Server MUST only include PIDs in the response that were specified (implicitly or explicitly) in the request. If the input parameters contain a PID name that is not currently defined by the ALTO Server, the ALTO Server MUST behave as if the PID did not appear in the input parameters.

7.7.3.1.6. Example

```
POST /networkmap/filtered HTTP/1.1
Host: custom.alto.example.com
Content-Length: [TODO]
Content-Type: application/alto-networkmapfilter+json
Accept: application/alto-networkmap+json,application/alto-error+json

{
  "pids": [ "PID1", "PID2" ]
}


HTTP/1.1 200 OK
Content-Length: [TODO]
Content-Type: application/alto-networkmap+json

{
  "meta" : {},
  "data" : {
    "map-vtag" : "1266506139",
    "map" : {
      "PID1" : {
        "ipv4" : [
          "192.0.2.0/24",
          "198.51.100.0/24"
        ]
      },
      "PID2" : {
        "ipv4": [
          "198.51.100.128/24"
        ]
      }
    }
  }
}
```

7.7.3.2. Filtered Cost Map

A Filtered Cost Map is a Cost Map Information Resource (Section 7.7.2.2) for which an ALTO Client may supply additional parameters limiting the scope of the resulting Cost Map. A Filtered Cost Map MAY be provided by an ALTO Server.

7.7.3.2.1. Media Type

See Section 7.7.2.2.1.

7.7.3.2.2. HTTP Method

This resource is requested using the HTTP POST method.

7.7.3.2.3. Input Parameters

Input parameters are supplied in the entity body of the POST request. This document specifies the input parameters with a data format indicated by the media type "application/alto-costmapfilter+json", which is a JSON Object of type ReqFilteredCostMap, where:

```
object {
  PIDName srcs<0..*>;
  PIDName dsts<0..*>;
} PIDFilter;

object {
  CostMode   cost-mode;
  CostType   cost-type;
  JSONString constraints<0..*>;   [OPTIONAL]
  PIDFilter  pids;                [OPTIONAL]
} ReqFilteredCostMap;
```

with members:

cost-type The Cost Type (Section 5.1.1) for the returned costs. This MUST be one of the supported Cost Types indicated in this resource's capabilities (Section 7.7.3.2.4).

cost-mode The Cost Mode (Section 5.1.2) for the returned costs. This MUST be one of the supported Cost Modes indicated in this resource's capabilities (Section 7.7.3.2.4).

constraints Defines a list of additional constraints on which elements of the Cost Map are returned. This parameter MUST NOT be specified if this resource's capabilities (Section 7.7.3.2.4) indicate that constraint support is not available. A constraint contains two entities separated by whitespace: (1) an operator either 'gt' for greater than or 'lt' for less than (2) a target numerical cost. The numerical cost is a number that MUST be defined in the same units as the Cost Type indicated by the cost-type parameter. ALTO Servers SHOULD use at least IEEE 754 double-

precision floating point [IEEE.754.2008] to store the numerical cost, and SHOULD perform internal computations using double-precision floating-point arithmetic. If multiple 'constraint' parameters are specified, they are interpreted as being related to each other with a logical AND.

pids A list of Source PIDs and a list of Destination PIDs for which Path Costs are to be returned. If a list is empty, the ALTO Server MUST interpret it as the full set of currently-defined PIDs. The ALTO Server MUST interpret entries appearing in a list multiple times as if they appeared only once. If the "pids" member is not present, both lists MUST be interpreted by the ALTO Server as containing the full set of currently-defined PIDs.

7.7.3.2.4. Capabilities

The URI providing this resource supports all capabilities documented in Section 7.7.2.2.4 (with identical semantics), plus additional capabilities. In particular, the capabilities are defined by a JSON object of type FilteredCostMapCapability:

```
object {
  CostMode cost-modes<0..*>;
  CostType cost-types<0..*>;
  JSONBool cost-constraints;
} FilteredCostMapCapability;
```

with members:

cost-modes See Section 7.7.2.2.4.

cost-types See Section 7.7.2.2.4.

cost-constraints If true, then the ALTO Server allows cost constraints to be included in requests to the corresponding URI. If not present, this member MUST be interpreted as if it specified false.

7.7.3.2.5. Response

See Section 7.7.2.2.5 for the format.

The returned Cost Map MUST NOT contain any source/destination pair that was not indicated (implicitly or explicitly) in the input parameters. If the input parameters contain a PID name that is not currently defined by the ALTO Server, the ALTO Server MUST behave as

if the PID did not appear in the input parameters.

If any constraints are specified, Source/Destination pairs that do for which the Path Costs do not meet the constraints MUST NOT be included in the returned Cost Map. If no constraints were specified, then all Path Costs are assumed to meet the constraints.

7.7.3.2.6. Example

```
POST /costmap/filtered HTTP/1.1
Host: custom.alto.example.com
Content-Type: application/alto-costmapfilter+json
Accept: application/alto-costmap+json,application/alto-error+json

{
  "cost-mode" : "numerical",
  "cost-type" : "routingcost",
  "pids" : {
    "srcs" : [ "PID1" ],
    "dsts" : [ "PID1", "PID2", "PID3" ]
  }
}
```

```
HTTP/1.1 200 OK
Content-Length: [TODO]
Content-Type: application/alto-costmap+json

{
  "meta" : {},
  "data" : {
    "cost-mode" : "numerical",
    "cost-type" : "routingcost",
    "map-vtag" : "1266506139",
    "map" : {
      "PID1": { "PID1": 0,  "PID2": 1,  "PID3": 2 }
    }
  }
}
```

7.7.4. Endpoint Property Service

The Endpoint Property Service provides information about Endpoint properties to ALTO Clients.

7.7.4.1. Endpoint Property

The Endpoint Property resource provides information about properties for individual endpoints. It MAY be provided by an ALTO Server. If an ALTO Server provides the Endpoint Property resource, it MUST provide and define at least the 'pid' property for each Endpoint.

7.7.4.1.1. Media Type

The media type is "application/alto-endpointprop+json".

7.7.4.1.2. HTTP Method

This resource is requested using the HTTP POST method.

7.7.4.1.3. Input Parameters

Input parameters are supplied in the entity body of the POST request. This document specifies the data format of input parameteres with the media type "application/alto-endpointpropparams+json", which is a JSON Object of type ReqEndpointProp:

```
object {
  EndpointProperty  properties<1..*>;
  TypedEndpointAddr endpoints<1..*>;
} ReqEndpointProp;
```

with members:

properties List of endpoint properties to returned for each endpoint. Each specified property MUST be included in the list of supported properties indicated by this resource's capabilities (Section 7.7.4.1.4). The ALTO Server MUST interpret entries appearing multiple times as if they appeared only once.

endpoints List of endpoint addresses for which the specified properties are to be returned. The ALTO Server MUST interpret entries appearing multiple times as if they appeared only once.

7.7.4.1.4. Capabilities

This resource may be defined across multiple types of endpoint properties. The capabilities of an ALTO Server URI providing Endpoint Properties are defined by a JSON Object of type EndpointPropertyCapability:

```
object {
  EndpointProperty prop-types<0..*>;
} EndpointPropertyCapability;
```

with members:

prop-types The Endpoint Property Types (Section 3.2.3) supported by the corresponding URI. If not present, this member MUST be interpreted as an empty array.

7.7.4.1.5. Response

The returned InfoResourceEntity object has "data" member of type InfoResourceEndpointProperty, where:

```
object {
  JSONString [EndpointProperty];
  ...
} EndpointProps;

object {
  EndpointProps  [TypedEndpointAddr]<0..*>;
  ...
} InfoResourceEndpointProperty;
```

InfoResourceEndpointProperty has one member for each endpoint indicated in the input parameters (with the name being the endpoint encoded as a TypedEndpointAddr). The requested properties for each endpoint are encoded in a corresponding EndpointProps object, which encodes one name/value pair for each requested property, where the property names are encoded as strings of type EndpointProperty and the property values encoded as JSON Strings.

The ALTO Server returns the value for each of the requested endpoint properties for each of the endpoints listed in the input parameters.

If the ALTO Server does not define a requested property for a particular endpoint, then it MUST omit it from the response for only that endpoint.

7.7.4.1.6. Example

```
POST /endpointprop/lookup HTTP/1.1
Host: alto.example.com
Content-Length: [TODO]
Content-Type: application/alto-endpointpropparams+json
Accept: application/alto-endpointprop+json,application/alto-error+json

{
  "properties" : [ "pid" ],
  "endpoints" : [ "ipv4:192.0.2.34", "ipv4:203.0.113.129" ]
}
```

```
HTTP/1.1 200 OK
Content-Length: [TODO]
Content-Type: application/alto-endpointprop+json

{
  "meta" : {},
  "data": {
    "ipv4:192.0.2.34"    : { "pid": "PID1" },
    "ipv4:203.0.113.129" : { "pid": "PID3" }
  }
}
```

7.7.5. Endpoint Cost Service

The Endpoint Cost Service provides information about costs between individual endpoints.

In particular, this service allows lists of Endpoint prefixes (and addresses, as a special case) to be ranked (ordered) by an ALTO Server.

7.7.5.1. Endpoint Cost

The Endpoint Cost resource provides information about costs between individual endpoints. It MAY be provided by an ALTO Server. If it is provided.

It is important to note that although this resource allows an ALTO Server to reveal costs between individual endpoints, an ALTO Server is not required to do so. A simple alternative would be to compute the cost between two endpoints as the cost between the PIDs corresponding to the endpoints. See Section 12.1 for additional

details.

7.7.5.1.1. Media Type

The media type is "application/alto-endpointcost+json".

7.7.5.1.2. HTTP Method

This resource is requested using the HTTP POST method.

7.7.5.1.3. Input Parameters

Input parameters are supplied in the entity body of the POST request. This document specifies input parameters with a data format indicated by media type "application/alto-endpointcostparams+json", which is a JSON Object of type ReqEndpointCostMap:

```
object {
  TypedEndpointAddr srcs<0..*>;    [OPTIONAL]
  TypedEndpointAddr dsts<1..*>;
} EndpointFilter;

object {
  CostMode          cost-mode;
  CostType          cost-type;
  JSONString        constraints;   [OPTIONAL]
  EndpointFilter    endpoints;
} ReqEndpointCostMap;
```

with members:

cost-mode The Cost Mode (Section 5.1.2) to use for returned costs. This MUST be one of the Cost Modes indicated in this resource's capabilities (Section 7.7.5.1.4).

cost-type The Cost Type (Section 5.1.1) to use for returned costs. This MUST be one of the Cost Types indicated in this resource's capabilities (Section 7.7.5.1.4).

constraints Defined equivalently to the "constraints" input parameter of a Filtered Cost Map (see Section 7.7.3.2).

endpoints A list of Source Endpoints and Destination Endpoints for which Path Costs are to be returned. If the list of Source Endpoints is empty (or not included), the ALTO Server MUST interpret it as if it contained the Endpoint Address corresponding

to the client IP address from the incoming connection (see Section 10.3 for discussion and considerations regarding this mode). The list of destination Endpoints MUST NOT be empty. The ALTO Server MUST interpret entries appearing multiple times in a list as if they appeared only once.

7.7.5.1.4. Capabilities

See Section 7.7.3.2.4.

7.7.5.1.5. Response

The returned InfoResourceEntity object has "data" member equal to InfoResourceEndpointCostMap, where:

```
object EndpointDstCosts {
  JSONNumber [TypedEndpointAddr];
  ...
};

object {
  EndpointDstCosts [TypedEndpointAddr]<0..*>;
  ...
} EndpointCostMapData;

object {
  CostMode            cost-mode;
  CostType            cost-type;
  EndpointCostMapData map;
} InfoResourceEndpointCostMap;
```

InfoResourceEndpointCostMap has members:

cost-mode The Cost Mode used in the returned Cost Map.

cost-type The Cost Type used in the returned Cost Map.

map The Endpoint Cost Map data itself.

EndpointCostMapData is a JSON object with each member representing a single Source Endpoint specified in the input parameters; the name for a member is the TypedEndpointAddr string identifying the corresponding Source Endpoint. For each Source Endpoint, a EndpointDstCosts object denotes the associated cost to each Destination Endpoint specified in the input parameters; the name for each member in the object is the TypedEndpointAddr string identifying

the corresponding Destination Endpoint.

7.7.5.1.6. Example

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: [TODO]
Content-Type: application/alto-endpointcostparams+json
Accept: application/alto-endpointcost+json,application/alto-error+json

{
  "cost-mode" : "ordinal",
  "cost-type" : "routingcost",
  "endpoints" : {
    "srcs": [ "ipv4:192.0.2.2" ],
    "dsts": [
      "ipv4:192.0.2.89",
      "ipv4:198.51.100.34",
      "ipv4:203.0.113.45"
    ]
  }
}
```

```
HTTP/1.1 200 OK
Content-Length: [TODO]
Content-Type: application/alto-endpointcost+json

{
  "meta" : {},
  "data" : {
    "cost-mode" : "ordinal",
    "cost-type" : "routingcost",
    "map" : {
      "ipv4:192.0.2.2": {
        "ipv4:192.0.2.89"    : 1,
        "ipv4:198.51.100.34" : 2,
        "ipv4:203.0.113.45"  : 3
      }
    }
  }
}
```

8. Redistributable Responses

This section defines how an ALTO Server enables certain Information Resources to be redistributed by ALTO Clients. Concepts are first introduced, followed by the protocol specification.

8.1. Concepts

8.1.1. Service ID

The Service ID is a UUID that identifies a set of ALTO Servers that would provide semantically-identical Information Resources for any request for any ALTO Client. Each ALTO Server within such a set is configured with an identical Service ID.

If a pair of ALTO Servers would provide an identical Information Resource (same information sources, configuration, internal computations, update timescales, etc) in response to any particular ALTO Client request, then the pair of ALTO Servers MAY have the same Service ID. If this condition is not true, the pair of ALTO Servers MUST have a different Service ID.

8.1.1.1. Rationale

For scalability and fault tolerance, multiple ALTO Servers may be deployed to serve equivalent ALTO Information. In such a scenario, Information Resources from any such redundant server should be seen as equivalent for the purposes of redistribution. For example, if two ALTO Servers A and B are deployed by the service provider to distribute equivalent ALTO Information, then clients contacting Server A should be able to redistribute Information Resources to clients contacting Server B.

To accomplish this behavior, ALTO Clients must be able to determine that Server A and Server B serve identical ALTO Information. One technique would be to rely on the ALTO Server's DNS name. However, such an approach would mandate that all ALTO Servers resolved by a particular DNS name would need to provide equivalent ALTO information, which may be unnecessarily restrictive. Another technique would be to rely on the server's IP address. However, this suffers similar problems as the DNS name in deployment scenarios using IP Anycast.

To avoid such restrictions, the ALTO Protocol allows an ALTO Service Provider to explicitly denote ALTO Servers that provide equivalent ALTO Information by giving them identical Service IDs. Service IDs decouple the identification of equivalent ALTO Servers from the discovery process.

8.1.1.2. Server Information Resource

If an ALTO Server generates redistributable responses, the Server Information resource's 'service-id' field MUST be set to the ALTO Server's Service ID.

8.1.1.3. Configuration

To help prevent ALTO Servers from mistakenly claiming to distribute equivalent ALTO Information, ALTO Server implementations SHOULD by default generate a new UUID at installation time or startup if one has not explicitly been configured.

8.1.2. Expiration Time

Information Resources marked as redistributable should indicate a time after which the information is considered stale and should be refreshed from the ALTO Server (or possibly another ALTO Client).

If an expiration time is present, the ALTO Server SHOULD ensure that it is reasonably consistent with the expiration time that would be computed by HTTP header fields. This specification makes no recommendation on which expiration time takes precedence, but implementers should be cognizant that HTTP intermediaries will obey only the HTTP header fields.

8.1.3. Signature

Information Resources marked as redistributable include a signature used to assert that the ALTO Server Provider generated the ALTO Information.

8.1.3.1. Rationale

Verification of the signature requires the ALTO Client to retrieve the ALTO Server's public key. To reduce requirements on the underlying transport (i.e., requiring SSL/TLS), an ALTO Client retrieves the public key as part of an X.509 certificate from the ALTO Server's Server Information resource.

8.1.3.2. Certificates

8.1.3.2.1. Local Certificate

The ALTO Server's public key is encoded within an X.509 certificate. The corresponding private key MUST be used to sign redistributable responses. This certificate is termed the Local Certificate for an ALTO Server.

8.1.3.2.2. Certificate Chain

To ease key provisioning, the ALTO Protocol is designed such that each ALTO Server with an identical Service ID may have a unique private key (and hence certificate).

The ALTO Service Provider may configure a certificate chain at each such ALTO Server. The Local Certificate for a single ALTO Server is the bottom-most certificate in the chain. The Certificate Chains of each ALTO Server with an identical Service ID MUST share a common Root Certificate.

Note that there are two simple deployment scenarios:

- One-Level Certificate Chain (Local Certificate Only): In this deployment scenario, each ALTO Server with an identical Service ID may provisioned with an identical Local Certificate.

- Two-Level Certificate Chain: In this deployment scenario, a Root Certificate is maintained for a set of ALTO Servers with the same Service ID. A unique Local Certificate signed by this CA is provisioned to each ALTO Server.

There are advantages to using a Certificate Chain instead of deploying the same Local Certificate to each ALTO Server. Specifically, it avoids storage of the CA's private key at ALTO Servers. It is possible to revoke and re-issue a key to a single ALTO Server.

8.1.3.2.3. Server Information Resource

If an ALTO Server generates redistributable responses, the Server Information resource's 'certificates' field MUST be populated with the ALTO Server's full certificate chain. The first element MUST be the ALTO Server's Local Certificate, followed by the remaining Certificate Chain in ascending order to the Root Certificate.

8.1.3.3. Signature Verification

ALTO Clients SHOULD verify the signature on any ALTO information received via redistribution before adjusting application behavior based on it.

An ALTO Client SHOULD cache its ALTO Server's Service ID and corresponding Certificate Chain included in the Server Information resource. Recall that the last certificate in this chain is the Root Certificate. The retrieval of the Service ID and certificates SHOULD be secured using HTTPS with proper validation of the server endpoint

of the SSL/TLS connection [RFC6125].

An Information Resource received via redistribution from Service ID S is declared valid if an ALTO Client can construct a transitive certificate chain from the certificate (public key) used to sign the Information Resource to the Root Certificate corresponding to Service ID S obtained by the ALTO Client in a Server Information resource.

To properly construct the chain and complete this validation, an ALTO Client may need to request additional certificates from other ALTO Clients. A simple mechanism is to request the certificate chain from the ALTO Client that received the Information Resource. Note that these additional received certificates may be cached locally by an ALTO Client.

ALTO Clients SHOULD verify Information Resources received via redistribution.

8.1.3.4. Redistribution by ALTO Clients

ALTO Clients SHOULD pass the ALTO Server Certificate, Signature, and Signature Algorithm along with the Information Resource. The mechanism for redistributing such information is not specified by the ALTO Protocol, but one possibility is to add additional messages or fields to the application's native protocol.

8.2. Protocol

An ALTO Server MAY indicate that a response is suitable for redistribution by including the "redistribution" member in the RspMetaData JSON object of an Information Resource. This additional member, called the Response Redistribution Descriptor, has type InfoResourceRedistDesc:

```
object {
  JSONString service-id;
  JSONString request-uri;
  JSONValue  request-body;
  JSONString media-type;
  JSONString expires;
} InfoResourceRedistDesc;
```

The fields encoded in the Response Redistribution Descriptor allows an ALTO Client receiving redistributed ALTO Information to understand the context of the query (the ALTO Service generating the response and any input parameters) and to interpret the results.

Information about ALTO Client performing the request and any HTTP

Headers passed in the request are not included in the Response Redistribution Descriptor. If any such information or headers influence the response generated by the ALTO Server, the response SHOULD NOT be indicated as redistributable.

8.2.1. Response Redistribution Descriptor Fields

This section defines the fields of the Response Redistribution Descriptor.

8.2.1.1. Service ID

The 'service-id' member is REQUIRED and MUST have a value equal to the ALTO Server's Service ID.

8.2.1.2. Request URI

The 'request-uri' member is REQUIRED and MUST specify the HTTP Request-URI that was passed in the HTTP request.

8.2.1.3. Request Body

If the HTTP request's entity body was non-empty, the 'request-body' member MUST specify full JSON value passed in the HTTP request's entity body (note that whitespace may differ, as long as the JSON Value is identical). If the HTTP request was empty, then the 'request-body' MUST NOT be included.

8.2.1.4. Response Media Type

The 'media-type' member is REQUIRED and MUST specify the same HTTP Content-Type that is used in the HTTP response.

8.2.1.5. Expiration Time

The 'expires' element is RECOMMENDED and, if present, MUST specify a time in UTC formatted according to [RFC3339].

8.2.2. Signature

The Hash Algorithm, Signature Algorithm, and Signature are included as either HTTP Headers or Trailers. Headers may be useful if Information Resources are pre-generated, while Trailers may be useful if Information Resources are dynamically generated (e.g., to avoid buffering large responses in memory while the hash value is computed).

The following HTTP Headers (the ALTO Server MAY specify them as HTTP

Trailers instead) MUST be used to encode the Signature parameters for redistributable Information Resources:

```
ALTO-HashAlgorithm: <HashAlgorithm>
ALTO-SignatureAlgorithm: <SignatureAlgorithm>
ALTO-SignatureDigest: <Signature>
```

where <HashAlgorithm> and <SignatureAlgorithm> are an integer values from the IANA TLS HashAlgorithm and SignatureAlgorithm registries, and <Signature> is the corresponding Base64-encoded signature.

9. Use Cases

The sections below depict typical use cases.

9.1. ALTO Client Embedded in P2P Tracker

Many P2P currently-deployed P2P systems use a Tracker to manage swarms and perform peer selection. P2P trackers may currently use a variety of information to perform peer selection to meet application-specific goals. By acting as an ALTO Client, an P2P tracker can use ALTO information as an additional information source to enable more network-efficient traffic patterns and improve application performance.

A particular requirement of many P2P trackers is that they must handle a large number of P2P clients. A P2P tracker can obtain and locally store ALTO information (the Network Map and Cost Map) from the ISPs containing the P2P clients, and benefit from the same aggregation of network locations done by ALTO Servers.

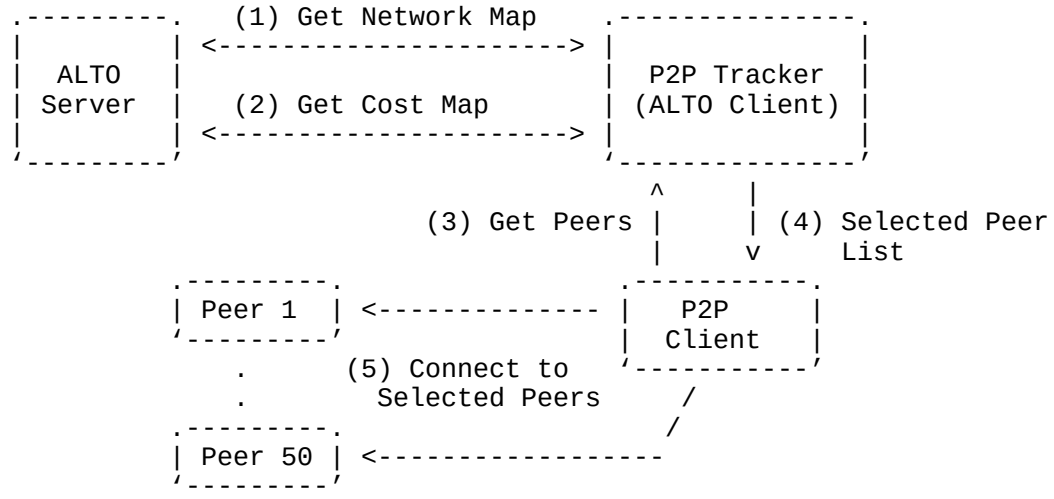


Figure 4: ALTO Client Embedded in P2P Tracker

Figure 4 shows an example use case where a P2P tracker is an ALTO Client and applies ALTO information when selecting peers for its P2P clients. The example proceeds as follows:

1. The P2P Tracker requests the Network Map covering all PIDs from the ALTO Server using the Network Map query. The Network Map includes the IP prefixes contained in each PID, allowing the P2P tracker to locally map P2P clients into a PIDs.

2. The P2P Tracker requests the Cost Map amongst all PIDs from the ALTO Server.

3. A P2P Client joins the swarm, and requests a peer list from the P2P Tracker.

4. The P2P Tracker returns a peer list to the P2P client. The returned peer list is computed based on the Network Map and Cost Map returned by the ALTO Server, and possibly other information sources. Note that it is possible that a tracker may use only the Network Map to implement hierarchical peer selection by preferring peers within the same PID and ISP.

5. The P2P Client connects to the selected peers.

Note that the P2P tracker may provide peer lists to P2P clients distributed across multiple ISPs. In such a case, the P2P tracker may communicate with multiple ALTO Servers.

9.2. ALTO Client Embedded in P2P Client: Numerical Costs

P2P clients may also utilize ALTO information themselves when selecting from available peers. It is important to note that not all P2P systems use a P2P tracker for peer discovery and selection. Furthermore, even when a P2P tracker is used, the P2P clients may rely on other sources, such as peer exchange and DHTs, to discover peers.

When an P2P Client uses ALTO information, it typically queries only the ALTO Server servicing its own ISP. The my-Internet view provided by its ISP's ALTO Server can include preferences to all potential peers.

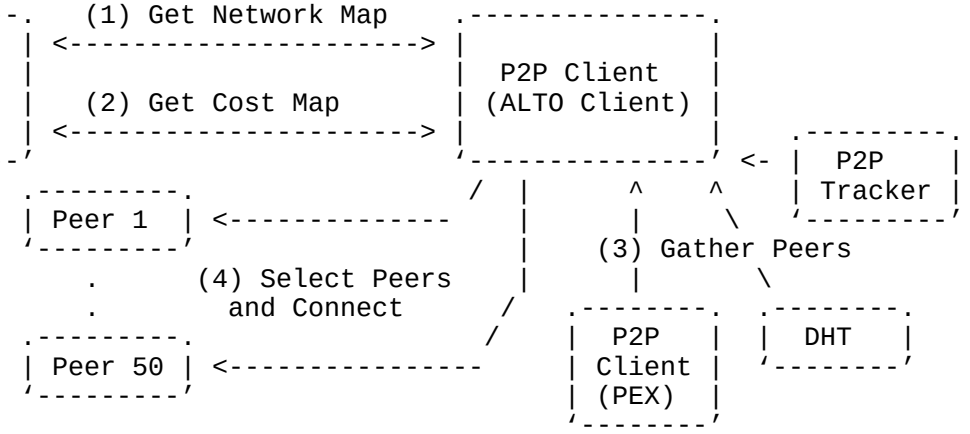


Figure 5: ALTO Client Embedded in P2P Client

Figure 5 shows an example use case where a P2P Client locally applies ALTO information to select peers. The use case proceeds as follows:

1. The P2P Client requests the Network Map covering all PIDs from the ALTO Server servicing its own ISP.

2. The P2P Client requests the Cost Map amongst all PIDs from the ALTO Server. The Cost Map by default specifies numerical costs.

3. The P2P Client discovers peers from sources such as Peer Exchange (PEX) from other P2P Clients, Distributed Hash Tables (DHT), and P2P Trackers.

4. The P2P Client uses ALTO information as part of the algorithm for selecting new peers, and connects to the selected peers.

9.3. ALTO Client Embedded in P2P Client: Ranking

It is also possible for a P2P Client to offload the selection and ranking process to an ALTO Server. In this use case, the ALTO Client gathers a list of known peers in the swarm, and asks the ALTO Server to rank them.

As in the use case using numerical costs, the P2P Client typically only queries the ALTO Server servicing its own ISP.

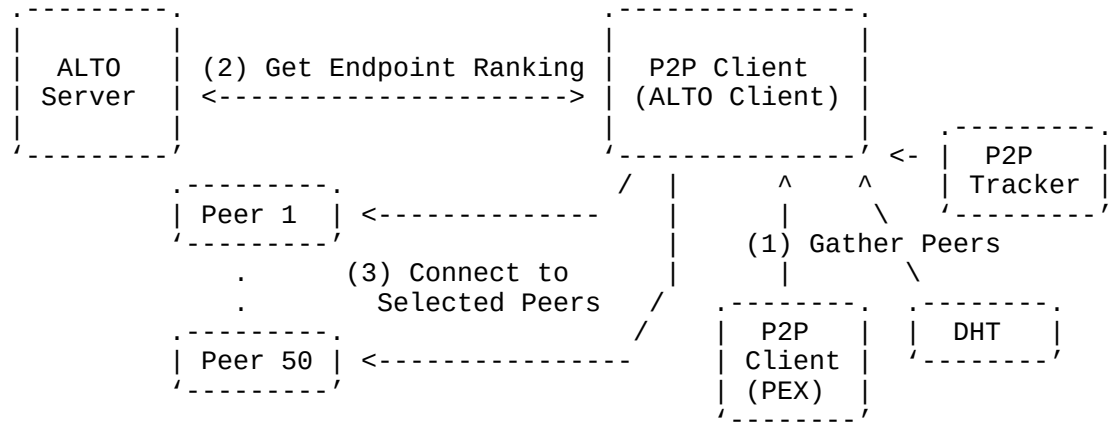


Figure 6: ALTO Client Embedded in P2P Client: Ranking

Figure 6 shows an example of this scenario. The use case proceeds as follows:

1. The P2P Client discovers peers from sources such as Peer Exchange (PEX) from other P2P Clients, Distributed Hash Tables (DHT), and P2P Trackers.

2. The P2P Client queries the ALTO Server's Ranking Service, including discovered peers as the set of Destination Endpoints, and indicates the 'ordinal' Cost Mode. The response indicates the ranking of the candidate peers.

3. The P2P Client connects to the peers in the order specified in the ranking.

10. Discussions

10.1. Discovery

The discovery mechanism by which an ALTO Client locates an appropriate ALTO Server is out of scope for this document. This document assumes that an ALTO Client can discover an appropriate ALTO Server. Once it has done so, the ALTO Client may use the Information Resource Directory (see Section 7.6) to locate an Information Resource with the desired ALTO Information.

10.2. Hosts with Multiple Endpoint Addresses

In practical deployments, especially during the transition from IPv4 to IPv6, a particular host may be reachable using multiple addresses. Furthermore, the particular network path followed when sending packets to the host may differ based on the address that is used. Network providers may prefer one path over another (e.g., one path my have a NAT64 middlebox). An additional consideration may be how to handle private address spaces (e.g., behind carrier-grade NATs).

To support such behavior, this document allows multiple types of endpoint addresses. In supporting multiple address types, the ALTO Protocol also allows ALTO Service Provider the flexibility to indicate preferences for paths from an endpoint address of one type to an endpoint address of a different type. Note that in general, the path through the network may differ dependent on the types of addresses that are used.

Note that there are limitations as to what information ALTO can provide in this regard. In particular, a particular ALTO Service provider may not be able to determine if connectivity with a particular endhost will succeed over IPv4 or IPv6, as this may depend upon information unknown to the ISP such as particular application implementations.

10.3. Network Address Translation Considerations

At this day and age of NAT v4<->v4, v4<->v6 [RFC6144], and possibly v6<->v6[I-D.mrw-nat66], a protocol should strive to be NAT friendly and minimize carrying IP addresses in the payload, or provide a mode of operation where the source IP address provide the information necessary to the server.

The protocol specified in this document provides a mode of operation where the source network location is computed by the ALTO Server (i.e., the the Endpoint Cost Service) from the source IP address found in the ALTO Client query packets. This is similar to how some P2P Trackers (e.g., BitTorrent Trackers - see "Tracker HTTP/HTTPS Protocol" in [BitTorrent]) operate.

The ALTO client SHOULD use the Session Traversal Utilities for NAT (STUN) [RFC5389] to determine a public IP address to use as a source Endpoint address. If using this method, the host MUST use the "Binding Request" message and the resulting "XOR-MAPPED-ADDRESS" parameter that is returned in the response. Using STUN requires cooperation from a publicly accessible STUN server. Thus, the ALTO client also requires configuration information that identifies the STUN server, or a domain name that can be used for STUN server discovery. To be selected for this purpose, the STUN server needs to provide the public reflexive transport address of the host.

10.4. Mapping IPs to ASNs

It may be desired for the ALTO Protocol to provide ALTO information including ASNs. Thus, ALTO Clients may need to identify the ASN for a Resource Provider to determine the cost to that Resource Provider.

Applications can already map IPs to ASNs using information from a BGP Looking Glass. To do so, they must download a file of about 1.5MB when compressed (as of October 2008, with all information not needed for IP to ASN mapping removed) and periodically (perhaps monthly) refresh it.

Alternatively, the Network Map query in the Map Filtering Service defined in this document could be extended to map ASNs into a set of IP prefixes. The mappings provided by the ISP would be both smaller and more authoritative.

For simplicity of implementation, it’s highly desirable that clients only have to implement exactly one mechanism of mapping IPs to ASNs.

10.5. Endpoint and Path Properties

An ALTO Server could make available many properties about Endpoints beyond their network location or grouping. For example, connection type, geographical location, and others may be useful to applications. This specification focuses on network location and grouping, but the protocol may be extended to handle other Endpoint properties.

11. IANA Considerations

11.1. application/alto-* Media Types

This document requests the registration of multiple media types, listed in Table 2.

Type	Subtype	Specification
application	alto-directory+json	Section 7.6
application	alto-serverinfo+json	Section 7.7.1.1
application	alto-networkmap+json	Section 7.7.2.1
application	alto-networkmapfilter+json	Section 7.7.3.1
application	alto-costmap+json	Section 7.7.2.2
application	alto-costmapfilter+json	Section 7.7.3.2
application	alto-endpointprop+json	Section 7.7.4.1
application	alto-endpointpropparams+json	Section 7.7.4.1
application	alto-endpointcost+json	Section 7.7.5.1
application	alto-endpointcostparams+json	Section 7.7.5.1
application	alto-error+json	Section 7.4

Table 2: ALTO Protocol Media Types

Type name: application

Subtype name: This documents requests the registration of multiple subtypes, as listed in Table 2.

Required parameters: n/a

Optional parameters: n/a

Encoding considerations: Encoding considerations are identical to those specified for the 'application/json' media type. See [RFC4627].

Security considerations: Security considerations relating to the generation and consumption of ALTO protocol messages are discussed in Section 12.

Interoperability considerations: This document specifies format of conforming messages and the interpretation thereof.

Published specification: This document is the specification for these media types; see Table 2for the section documenting each media type.

Applications that use this media type: ALTO Servers and ALTO Clients either standalone or embedded within other applications.

Additional information:

Magic number(s): n/a

File extension(s): This document uses the mime type to refer to protocol messages and thus does not require a file extension.

Macintosh file type code(s): n/a

Person & email address to contact for further information: See "Authors' Addresses" section.

Intended usage: COMMON

Restrictions on usage: n/a

Author: See "Authors' Addresses" section.

Change controller: See "Authors' Addresses" section.

11.2. ALTO Cost Type Registry

This document requests the creation of an ALTO Cost Type registry to be maintained by IANA.

This registry serves two purposes. First, it ensures uniqueness of identifiers referring to ALTO Cost Types. Second, it provides references to particular semantics of allocated Cost Types to be applied by both ALTO Servers and applications utilizing ALTO Clients.

New ALTO Cost Types are assigned after Expert Review [RFC5226]. The Expert Reviewer will generally consult the ALTO Working Group or its successor. Expert Review is used to ensure that proper documentation regarding ALTO Cost Type semantics and security considerations has been provided. The provided documentation should be detailed enough to provide guidance to both ALTO Service Providers and applications utilizing ALTO Clients as to how values of the registered ALTO Cost Type should be interpreted. Updates and deletions of ALTO Cost Types follow the same procedure.

Registered ALTO Cost Type identifiers MUST conform to the syntatical requirements specified in Section 7.5.4. Identifiers are to be recorded and displayed as ASCII strings.

Identifiers prefixed with 'priv:' are reserved for Private Use. Identifiers prefixed with 'exp:' are reserved for Experimental use.

Requests to add a new value to the registry MUST include the

following information:

- Identifier: The name of the desired ALTO Cost Type.

- Intended Semantics: ALTO Costs carry with them semantics to guide their usage by ALTO Clients. For example, if a value refers to a measurement, the measurement units must be documented. For proper implementation of the ordinal Cost Mode (e.g., by a third-party service), it should be documented whether higher or lower values of the cost are more preferred.

- Security Considerations: ALTO Costs expose information to ALTO Clients. As such, proper usage of a particular Cost Type may require certain information to be exposed by an ALTO Service Provider. Since network information is frequently regarded as proprietary or confidential, ALTO Service Providers should be made aware of the security ramifications related to usage of a Cost Type.

This specification requests registration of the identifier 'routingcost'. Semantics for the this Cost Type are documented in Section 5.1.1.1, and security considerations are documented in Section 12.1.

11.3. ALTO Endpoint Property Registry

This document requests the creation of an ALTO Endpoint Property registry to be maintained by IANA.

This registry serves two purposes. First, it ensures uniqueness of identifiers referring to ALTO Endpoint Properties. Second, it provides references to particular semantics of allocated Endpoint Properties to be applied by both ALTO Servers and applications utilizing ALTO Clients.

New ALTO Endpoint Properties are assigned after Expert Review [RFC5226]. The Expert Reviewer will generally consult the ALTO Working Group or its successor. Expert Review is used to ensure that proper documentation regarding ALTO Endpoint Property semantics and security considerations has been provided. The provided documentation should be detailed enough to provide guidance to both ALTO Service Providers and applications utilizing ALTO Clients as to how values of the registered ALTO Endpoint Properties should be interpreted. Updates and deletions of ALTO Endpoint Properties follow the same procedure.

Registered ALTO Endpoint Property identifiers MUST conform to the syntatical requirements specified in Section 7.5.5. Identifiers are

to be recorded and displayed as ASCII strings.

Identifiers prefixed with 'priv:' are reserved for Private Use. Identifiers prefixed with 'exp:' are reserved for Experimental use.

Requests to add a new value to the registry MUST include the following information:

- Identifier: The name of the desired ALTO Endpoint Property.
- Intended Semantics: ALTO Endpoint Properties carry with them semantics to guide their usage by ALTO Clients. For example, if a value refers to a measurement, the measurement units must be documented. For proper implementation of the ordinal Cost Mode (e.g., by a third-party service), it should be documented whether higher or lower values of the cost are more preferred.
- Security Considerations: ALTO Endpoint Properties expose information to ALTO Clients. As such, proper usage of a particular Endpoint Properties may require certain information to be exposed by an ALTO Service Provider. Since network information is frequently regarded as proprietary or confidential, ALTO Service Providers should be made aware of the security ramifications related to usage of an Endpoint Property.

This specification requests registration of the identifier 'pid'. Semantics for the this Endpoint Property are documented in Section 4.1, and security considerations are documented in Section 12.1.

12. Security Considerations

12.1. Privacy Considerations for ISPs

ISPs must be cognizant of the network topology and provisioning information provided through ALTO Interfaces. ISPs should evaluate how much information is revealed and the associated risks. On the one hand, providing overly fine-grained information may make it easier for attackers to infer network topology. In particular, attackers may try to infer details regarding ISPs' operational policies or inter-ISP business relationships by intentionally posting a multitude of selective queries to an ALTO server and analyzing the responses. Such sophisticated attacks may reveal more information than an ISP hosting an ALTO server intends to disclose. On the other hand, revealing overly coarse-grained information may not provide benefits to network efficiency or performance improvements to ALTO Clients.

12.2. ALTO Clients

Applications using the information must be cognizant of the possibility that the information is malformed or incorrect. Even if an ALTO Server has been properly authenticated by the ALTO Client, the information provided may be malicious because the ALTO Server and its credentials have been compromised (e.g., through malware). Other considerations (e.g., relating to application performance) can be found in Section 6 of [RFC5693].

ALTO Clients should also be cognizant of revealing Network Location Identifiers (IP addresses or fine-grained PIDs) to the ALTO Server, as doing so may allow the ALTO Server to infer communication patterns. One possibility is for the ALTO Client to only rely on Network Map for PIDs and Cost Map amongst PIDs to avoid passing IP addresses of their peers to the ALTO Server.

In addition, ALTO clients should be cautious not to unintentionally or indirectly disclose the resource identifier (of which they try to improve the retrieval through ALTO-guidance), e.g., the name/ identifier of a certain video stream in P2P live streaming, to the ALTO server. Note that the ALTO Protocol specified in this document does not explicitly reveal any resource identifier to the ALTO Server. However, for instance, depending on the popularity or other specifics (such as language) of the resource, an ALTO server could potentially deduce information about the desired resource from information such as the Network Locations the client sends as part of its request to the server.

12.3. Authentication, Integrity Protection, and Encryption

SSL/TLS can provide encryption of transmitted messages as well as authentication of the ALTO Client and Server. HTTP Basic or Digest authentication can provide authentication of the client (combined with SSL/TLS, it can additionally provide encryption and authentication of the server).

An ALTO Server may optionally use authentication (and potentially encryption) to protect ALTO information it provides. This can be achieved by digitally signing a hash of the ALTO information itself and attaching the signature to the ALTO information. There may be special use cases where encryption of ALTO information is desirable. In many cases, however, information sent out by an ALTO Server may be regarded as non-confidential information.

ISPs should be cognizant that encryption only protects ALTO information until it is decrypted by the intended ALTO Client. Digital Rights Management (DRM) techniques and legal agreements

protecting ALTO information are outside of the scope of this document.

12.4. ALTO Information Redistribution

It is possible for applications to redistribute ALTO information to improve scalability. Even with such a distribution scheme, ALTO Clients obtaining ALTO information must be able to validate the received ALTO information to ensure that it was generated by an appropriate ALTO Server. Further, to prevent the ALTO Server from being a target of attack, the verification scheme must not require ALTO Clients to contact the ALTO Server to validate every set of information. Contacting an ALTO server for information validation would also undermine the intended effect of redistribution and is therefore not desirable.

Note that the redistribution scheme must additionally handle details such as ensuring ALTO Clients retrieve ALTO information from the correct ALTO Server. See [I-D.gu-alto-redistribution] for further discussion. Details of a particular redistribution scheme are outside the scope of this document.

To fulfill these requirements, ALTO Information meant to be redistributable contains a digital signature which includes a hash of the ALTO information signed by the ALTO Server with its private key. The corresponding public key is included in the Server Information resource Section 7.7.1.1, along with the certificate chain to a Root Certificate generated by the ALTO Service Provider. To prevent man-in-the-middle attacks, an ALTO Client SHOULD perform the Server Information resource request over SSL/TLS and verify the server identity according to [RFC6125].

The signature verification algorithm is detailed in Section 8.1.3.3.

12.5. Denial of Service

ISPs should be cognizant of the workload at the ALTO Server generated by certain ALTO Queries, such as certain queries to the Map Filtering Service and Ranking Service. In particular, queries which can be generated with low effort but result in expensive workloads at the ALTO Server could be exploited for Denial-of-Service attacks. For instance, a simple ALTO query with n Source Network Locations and m Destination Network Locations can be generated fairly easily but results in the computation of n*m Path Costs between pairs by the ALTO Server (see Section 5.2). One way to limit Denial-of-Service attacks is to employ access control to the ALTO server. Another possible mechanism for an ALTO Server to protect itself against a multitude of computationally expensive bogus requests is to demand

that each ALTO Client to solve a computational puzzle first before allocating resources for answering a request (see, e.g., [I-D.jennings-sip-hashcash]). The current specification does not use such computational puzzles, and discussion regarding tradeoffs of such an approach would be needed before including such a technique in the ALTO Protocol.

ISPs should also leverage the fact that the the Map Service allows ALTO Servers to pre-generate maps that can be useful to many ALTO Clients.

12.6. ALTO Server Access Control

In order to limit access to an ALTO server (e.g., for an ISP to only allow its users to access its ALTO server, or to prevent Denial-of-Service attacks by arbitrary hosts from the Internet), an ALTO server may employ access control policies. Depending on the use-case and scenario, an ALTO server may restrict access to its services more strictly or rather openly (see [I-D.stiemerling-alto-deployments] for a more detailed discussion on this issue).

13. References

13.1. Normative References

[IEEE.754.2008] Institute of Electrical and Electronics Engineers, "Standard for Binary Floating-Point Arithmetic", IEEE Standard 754, August 2008.

[RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996.

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

[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.

[RFC3339] Klyne, G., Ed. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, July 2002.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, July 2005.

[RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", RFC 4627, July 2006.

[RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, August 2006.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

[RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, August 2010.

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, March 2011.

13.2. Informative References

[BitTorrent]
"Bittorrent Protocol Specification v1.0", <http://wiki.theory.org/BitTorrentSpecification>.

[I-D.akonjang-alto-proxidor]
Akonjang, O., Feldmann, A., Previdi, S., Davie, B., and D. Saucez, "The PROXIDOR Service", draft-akonjang-alto-proxidor-00 (work in progress), March 2009.

[I-D.gu-alto-redistribution]
Yingjie, G., Alimi, R., and R. Even, "ALTO Information Redistribution", draft-gu-alto-redistribution-03 (work in progress), July 2010.

[I-D.ietf-alto-reqs]
Previdi, S., Stiemerling, M., Woundy, R., and Y. Yang, "Application-Layer Traffic Optimization (ALTO)

Requirements", draft-ietf-alto-reqs-08 (work in progress), March 2011.

[I-D.ietf-alto-server-discovery]
Kiesel, S., Stiemerling, M., Schwan, N., Scharf, M., and S. Yongchao, "ALTO Server Discovery", draft-ietf-alto-server-discovery-00 (work in progress), May 2011.

[I-D.jennings-sip-hashcash]
Jennings, C., "Computational Puzzles for SPAM Reduction in SIP", draft-jennings-sip-hashcash-06 (work in progress), July 2007.

[I-D.mrw-nat66]
Wasserman, M. and F. Baker, "IPv6-to-IPv6 Network Prefix Translation", draft-mrw-nat66-16 (work in progress), April 2011.

[I-D.p4p-framework]
Alimi, R., Pasko, D., Popkin, L., Wang, Y., and Y. Yang, "P4P: Provider Portal for P2P Applications", draft-p4p-framework-00 (work in progress), November 2008.

[I-D.saumitra-alto-multi-ps]
Das, S., Narayanan, V., and L. Dondeti, "ALTO: A Multi Dimensional Peer Selection Problem", draft-saumitra-alto-multi-ps-00 (work in progress), October 2008.

[I-D.saumitra-alto-queryresponse]
Das, S. and V. Narayanan, "A Client to Service Query Response Protocol for ALTO", draft-saumitra-alto-queryresponse-00 (work in progress), March 2009.

[I-D.shalunov-alto-infoexport]
Shalunov, S., Penno, R., and R. Woundy, "ALTO Information Export Service", draft-shalunov-alto-infoexport-00 (work in progress), October 2008.

[I-D.stiemerling-alto-deployments]
Stiemerling, M. and S. Kiesel, "ALTO Deployment Considerations", draft-stiemerling-alto-deployments-06 (work in progress), January 2011.

[I-D.wang-alto-p4p-specification]
Wang, Y., Alimi, R., Pasko, D., Popkin, L., and Y. Yang,

"P4P Protocol Specification",
draft-wang-alto-p4p-specification-00 (work in progress),
March 2009.

[P4P-SIGCOMM08]
Xie, H., Yang, Y., Krishnamurthy, A., Liu, Y., and A. Silberschatz, "P4P: Provider Portal for (P2P) Applications", SIGCOMM 2008, August 2008.

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

[RFC6144] Baker, F., Li, X., Bao, C., and K. Yin, "Framework for IPv4/IPv6 Translation", RFC 6144, April 2011.

Appendix A. Acknowledgments

Thank you to Jan Seedorf for contributions to the Security Considerations section. We would like to thank Yingjie Gu and Roni Even for helpful input and design concerning ALTO Information redistribution.

We would like to thank the following people whose input and involvement was indispensable in achieving this merged proposal:

Obi Akonjang (DT Labs/TU Berlin),

Saumitra M. Das (Qualcomm Inc.),

Syon Ding (China Telecom),

Doug Pasko (Verizon),

Laird Popkin (Pando Networks),

Satish Raghunath (Juniper Networks),

Albert Tian (Ericsson/Redback),

Yu-Shun Wang (Microsoft),

David Zhang (PPLive),

Yunfei Zhang (China Mobile).

We would also like to thank the following additional people who were

involved in the projects that contributed to this merged document: Alex Gerber (AT&T), Chris Griffiths (Comcast), Ramit Hora (Pando Networks), Arvind Krishnamurthy (University of Washington), Marty Lafferty (DCIA), Erran Li (Bell Labs), Jin Li (Microsoft), Y. Grace Liu (IBM Watson), Jason Livingood (Comcast), Michael Merritt (AT&T), Ingmar Poese (DT Labs/TU Berlin), James Royalty (Pando Networks), Damien Saucez (UCL) Thomas Scholl (AT&T), Emilio Sepulveda (Telefonica), Avi Silberschatz (Yale University), Hassan Sipra (Bell Canada), Georgios Smaragdakis (DT Labs/TU Berlin), Haibin Song (Huawei), Oliver Spatscheck (AT&T), See-Mong Tang (Microsoft), Jia Wang (AT&T), Hao Wang (Yale University), Ye Wang (Yale University), Haiyong Xie (Yale University).

Appendix B. Authors

[[CmtAuthors: RFC Editor: Please move information in this section to the Authors' Addresses section at publication time.]]

Stefano Previdi
Cisco

Email: sprevidi@cisco.com

Stanislav Shalunov
BitTorrent

Email: shalunov@bittorrent.com

Richard Woundy
Comcast

Richard_Woundy@cable.comcast.com

Authors' Addresses

Richard Alimi (editor)
Google
1600 Amphitheatre Parkway
Mountain View CA
USA

Email: ralimi@google.com

Reinaldo Penno (editor)
Juniper Networks
1194 N Mathilda Avenue
Sunnyvale CA
USA

Email: rpenno@juniper.net

Y. Richard Yang (editor)
Yale University
51 Prospect St
New Haven CT
USA

Email: yry@cs.yale.edu

Network Working Group
Internet-Draft
Intended status: Informational
Expires: January 12, 2012

S. Kiesel, Ed.
University of Stuttgart
S. Previdi
Cisco Systems, Inc.
M. Stiemerling
NEC Europe Ltd.
R. Woundy
Comcast Corporation
Y R. Yang
Yale University
July 11, 2011

Application-Layer Traffic Optimization (ALTO) Requirements
draft-ietf-alto-reqs-11.txt

Abstract

Many Internet applications are used to access resources, such as pieces of information or server processes, which are available in several equivalent replicas on different hosts. This includes, but is not limited to, peer-to-peer file sharing applications. The goal of Application-Layer Traffic Optimization (ALTO) is to provide guidance to applications, which have to select one or several hosts from a set of candidates, that are able to provide a desired resource. This guidance shall be based on parameters that affect performance and efficiency of the data transmission between the hosts, e.g., the topological distance. The ultimate goal is to improve performance (or Quality of Experience) in the application while reducing resource consumption in the underlying network infrastructure.

This document enumerates requirements for specifying, assessing, or comparing protocols and implementations.

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 January 12, 2012.

Table of Contents

1. Introduction

The motivation for Application-Layer Traffic Optimization (ALTO) is described in the ALTO problem statement [RFC5693].

The goal of ALTO is to provide information which can help peer-to-peer (P2P) applications to make better decisions with respect to peer selection. However, ALTO may be useful for non-P2P applications as well. For example, clients of client-server applications may use information provided by ALTO to select one of several servers or information replicas. As another example, ALTO information could be used to select a media relay needed for NAT traversal. The goal of these informed decisions is to improve performance (or Quality of Experience) in the application while reducing resource consumption in the underlying network infrastructure.

Usually, it would be difficult or even impossible for application entities to acquire this information by other mechanisms (e.g., using measurements between the peers of a P2P overlay), because of complexity or because it is based on network topology information, network operational costs, or network policies, which the respective network provider does not want to disclose in detail.

The logical entities that provide the ALTO service do not take part in the actual user data transport, i.e., they do not implement functions for relaying user data. They may be placed on various kinds of physical nodes, e.g., on dedicated servers, as auxiliary processes in routers, on "trackers" or "super peers" of a P2P application, etc.

2. Terminology and Architectural Framework

2.1. Requirements Notation

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].

2.2. ALTO Terminology

This document uses the following ALTO-related terms, which are defined in [RFC5693]:

Application, Peer, P2P, Resource, Resource Identifier, Resource Provider, Resource Consumer, Transport Address, Overlay Network, Resource Directory, ALTO Service, ALTO Server, ALTO Client, ALTO Query, ALTO Response, ALTO Transaction, Local Traffic, Peering Traffic, Transit Traffic, Application protocol, ALTO Client Protocol, Provisioning protocol.

Furthermore, the following additional terms will be used:

- Host Group Descriptor: Information used to describe one or more Internet hosts (such as the resource consumer which seeks ALTO guidance, or one or more candidate resource providers) and their location within the network topology. This can be, for example, a single IP address, an address prefix or address range that contains the host(s), or an autonomous system (AS) number. Different options may provide different levels of detail. Depending on the system architecture, this may have implications on the quality of the guidance ALTO is able to provide, on whether recommendations can be aggregated, and on how much privacy-sensitive information about users might be disclosed to additional parties.

- Host Characteristics Attribute: Properties of a host (other than the host group descriptor), in particular related to its attachment to the network. This information may be stored in an ALTO server and transmitted via an ALTO protocol. It may be evaluated according to the rating criteria.

- Rating Criterion: The condition or relation that defines the "better" in "better-than-random peer selection", which is the ultimate goal of ALTO. Examples may include "host’s Internet access is not subject to volume based charging (flat rate)" or "low topological distance". Some rating criteria, such as "low topological distance", need to include a reference point, i. e., "low topological distance from a given resource consumer", which

can be described by means of a host group descriptor.

2.3. Architectural Framework for ALTO

There are various architectural options for how ALTO could be implemented, and specifying or mandating one specific architecture is out of the scope of this document.

The ALTO Working Group Charter [ALTO-charter] itemizes several key components, which shall be elaborated and specified by the ALTO Working Group. The ALTO problem statement [RFC5693] defines a terminology (see Section 2.2) and presents a figure that gives a high-level overview of protocol interaction between ALTO elements.

This document itemizes requirements for the following components and information elements that are part of the above-mentioned architecture:

- An ALTO client protocol, which is used for sending ALTO queries and ALTO responses between ALTO client and ALTO server.
- The discovery mechanism, which will be used by ALTO clients in order to find out where to send ALTO requests.
- The overall architecture, especially with respect to security and privacy issues.
- Host group descriptors, which are used to describe the location of a host in the network topology.
- Rating criteria, i. e., conditions or relations that shall be evaluated in order to generate the ALTO guidance.

3. ALTO Requirements

[*** Note to the RFC editor: before publication as an RFC, please remove the draft version number from the requirements numbering, i.e., change ARv11-1 to AR-1, and so on. Furthermore, remove this note. ***]

3.1. ALTO Client Protocol

3.1.1. General Requirements

REQ. ARv11-1: The ALTO service is provided by one or more ALTO servers. ALTO servers MUST implement an ALTO client protocol, for receiving ALTO queries from ALTO clients and for sending the corresponding ALTO responses.

REQ. ARv11-2: ALTO clients MUST implement an ALTO client protocol, for sending ALTO queries to ALTO servers and for receiving the corresponding ALTO responses.

REQ. ARv11-3: The format of the ALTO query message MUST allow the ALTO client to solicit guidance for selecting appropriate resource providers.

REQ. ARv11-4: The format of the ALTO response message MUST allow the ALTO server to express its guidance for selecting appropriate resource providers.

The detailed specification of an ALTO client protocol is out of the scope of this document. However, this document enumerates requirements for ALTO, to be considered when specifying, assessing, or comparing protocols and implementations.

3.1.2. Host Group Descriptor Support

The ALTO guidance is based on the evaluation of several resource providers or groups of resource providers, which are characterized by means of host group descriptors, considering one or more rating criteria.

REQ. ARv11-5: An ALTO client protocol MUST support the host group descriptor types "IPv4 address prefix" and "IPv6 address prefix". They can be used to specify the IP address of one host, or an IP address range (in CIDR notation), which contains all hosts in question.

REQ. ARv11-6: An ALTO client protocol MUST be extensible to enable support of other host group descriptor types in future. An ALTO

client protocol specification MUST define an appropriate procedure for adding new host group descriptor types, e.g., by establishing an IANA registry.

REQ. ARv11-7: ALTO clients and ALTO servers MUST clearly identify the type of each host group descriptor sent in ALTO queries or responses.

REQ. ARv11-8: For host group descriptor types other than "IPv4 address prefix" and "IPv6 address prefix", the host group descriptor type identification MUST be supplemented by a reference to a facility, which can be used to translate host group descriptors of that type to IPv4/IPv6 address prefixes, e.g., by means of a mapping table or an algorithm.

REQ. ARv11-9: Protocol functions for mapping other host group descriptor types to IPv4/IPv6 address prefixes SHOULD be designed and specified as part of an ALTO client protocol, and the corresponding address mapping information SHOULD be made available by the same entity that wants to use these host group descriptors within an ALTO client protocol. However, an ALTO server or an ALTO client MAY also send a reference to an external mapping facility, e.g., a translation table to be obtained via an alternative mechanism.

REQ. ARv11-10: An ALTO client protocol specification MUST define mechanisms, which can be used by the ALTO server to indicate that a host group descriptor used by the ALTO client is of an unsupported type, or that the indicated mapping mechanism could not be used.

REQ. ARv11-11: An ALTO client protocol specification MUST define mechanisms, which can be used by the ALTO client to indicate that a host group descriptor used by the ALTO server is of an unsupported type, or that the indicated mapping mechanism could not be used.

3.1.3. Rating Criteria Support

REQ. ARv11-12: An ALTO client protocol specification MUST define a rating criterion that can be used to express and evaluate the "relative operator’s preference." This is a relative measure, i.e., it is not associated with any unit of measurement. A more-preferred rating according to this criterion indicates that the application should prefer the respective candidate resource provider over others with less-preferred ratings (unless information from non-ALTO sources suggests a different choice, such as transmission attempts suggesting that the path is currently congested). The operator of the ALTO server does not have to disclose how and based on which data the ratings are actually computed. Examples could be: cost for peering or transit traffic, traffic engineering inside the network, and other

policies.

REQ. ARv11-13: An ALTO client protocol MUST be extensible to enable support of other rating criteria types in future. An ALTO client protocol specification MUST define an appropriate procedure for adding new rating criteria types, e.g., by establishing an IANA registry.

REQ. ARv11-14: ALTO client protocol specifications MUST NOT define rating criteria closely related to the instantaneous network congestion state, whose primary aim is to serve an alternative to established congestion control strategies, such as using TCP-based transport.

> One design assumption for ALTO is that it is acceptable that the host characteristics attributes, which are stored and processed in the ALTO servers for giving the guidance, are updated rather infrequently. Typical update intervals may be several orders of magnitude longer than the typical network-layer packet round-trip time (RTT). Therefore, ALTO cannot be a replacement for TCP-like congestion control mechanisms. The definition of alternate approaches for congestion control is explicitly a non-goal for the ALTO working group [ALTO-charter].

REQ. ARv11-15: Applications using ALTO guidance MUST NOT rely on the ALTO guidance to avoid causing network congestion. Instead, applications MUST use other appropriate means, such as TCP based transport, to avoid causing excessive congestion.

REQ. ARv11-16: The ALTO query message SHOULD allow the ALTO client to express which rating criteria should be considered, as well as their relative relevance for the specific application that will eventually make use of the guidance.

REQ. ARv11-17: The ALTO response message SHOULD allow the ALTO server to express which rating criteria have been considered when generating the response.

REQ. ARv11-18: An ALTO client protocol specification MUST define mechanisms, which can be used by the ALTO client and the ALTO server to indicate that a rating criteria used by the other party is of an unsupported type.

3.1.4. Placement of Entities and Timing of Transactions

With respect to the placement of ALTO clients, several modes of operation exist:

- One mode of ALTO operation is that an ALTO client may be embedded directly in the resource consumer, i.e., the application protocol entity that will eventually initiate data transmission to/from the selected resource provider(s) in order to access the desired resource. For example, an ALTO client could be integrated into the peer of a P2P application that uses a distributed algorithm such as "query flooding" for resource discovery.

- Another mode of operation is to integrate the ALTO client into a third party such as a resource directory, which may issue ALTO queries to solicit preference on potential resource providers, considering the respective resource consumer. For example, an ALTO client could be integrated into the tracker of a tracker-based P2P application, in order to request ALTO guidance on behalf of the peers contacting the tracker.

REQ. ARv11-19: An ALTO client protocol MUST support the mode of operation in which the ALTO client is directly embedded in the resource consumer.

REQ. ARv11-20: An ALTO client protocol MUST support the mode of operation in which the ALTO client is embedded in a third party, which performs queries on behalf of resource consumers.

REQ. ARv11-21: An ALTO client protocol MUST be designed in a way that the ALTO service can be provided by an entity which is not the operator of the underlying IP network.

REQ. ARv11-22: An ALTO client protocol MUST be designed in a way that different instances of the ALTO service operated by different providers can coexist.

With respect to the timing of ALTO queries, several modes of operation exist:

- In target-aware query mode, an ALTO client performs the ALTO query when the desired resource and a set of candidate resource providers are already known, i. e., after DHT lookups, queries to the resource directory, etc.

- In target-independent query mode, ALTO queries are performed in advance or periodically, in order to receive comprehensive, "target-independent" guidance, which will be cached locally and evaluated later, when a resource is to be accessed.

REQ. ARv11-23: An ALTO client protocol MUST support at least one of these two modes, either the target-aware or the target-independent query mode.

REQ. ARv11-24: An ALTO client protocol SHOULD support both the target-aware and the target-independent query mode.

REQ. ARv11-25: An ALTO client protocol SHOULD support version numbering, TTL (time-to-live) attributes, and/or similar mechanisms in ALTO transactions, in order to enable time validity checking for caching, and to enable comparisons of multiple recommendations obtained through redistribution.

REQ. ARv11-26: An ALTO client protocol SHOULD allow the ALTO server to add information about appropriate modes of re-use to its ALTO responses. Re-use may include redistributing an ALTO response to other parties, as well as using the same ALTO information in a resource directory to improve the responses to different resource consumers, within the specified lifetime of the ALTO response. The ALTO server SHOULD be able to express that

- no re-use should occur
- re-use is appropriate for a specific "target audience", i.e., a set of resource consumers explicitly defined by a list of host group descriptors. The ALTO server MAY specify a "target audience" in the ALTO response, which is only a subset of the known actual "target audience", e.g., if required by operator policies
- re-use is appropriate for any resource consumer that would send (or cause a third party sending on behalf of it) the same ALTO query (i.e., with the same query parameters, except for the resource consumer ID, if applicable) to this ALTO server
- re-use is appropriate for any resource consumer that would send (or cause a third party sending on behalf of it) the same ALTO query (i.e., with the same query parameters, except for the resource consumer ID, if applicable) to any other ALTO server, which was discovered (using an ALTO discovery mechanism) together with this ALTO server
- re-use is appropriate for any resource consumer that would send (or cause a third party sending on behalf of it) the same ALTO query (i.e., with the same query parameters, except for the resource consumer ID, if applicable) to any ALTO server in the whole network

REQ. ARv11-27: An ALTO client protocol MUST support the exchange of ALTO transactions even if the ALTO client is located in the private address realm behind a network address translator (NAT). There are different types of NAT, see [RFC4787] and [RFC5382].

3.1.5. Protocol Extensibility

REQ. ARv11-28: An ALTO client protocol MUST include support for adding protocol extensions in a non-disruptive, backward-compatible way.

REQ. ARv11-29: An ALTO client protocol MUST include protocol versioning support, in order to clearly distinguish between incompatible versions of the protocol.

3.1.6. Error Handling and Overload Protection

REQ. ARv11-30: An ALTO client protocol MUST use TCP based transport.

REQ. ARv11-31: An ALTO client protocol specification MUST specify mechanisms, or detail how to leverage appropriate mechanisms provided by underlying protocol layers, which can be used by an ALTO server to inform clients about an impending or occurring overload situation, and require them to throttle their query rate.

In particular, as a simple way of achieving some basic form of throttling, an ALTO server MAY answer ALTO queries with a "Retry After: {point in time | time delta}" message. This "Retry After" MAY be sent as part of the ALTO reply together with the requested guiding information, or as a standalone (error) message not giving the requested guidance.

REQ. ARv11-32: An ALTO client protocol specification MUST specify mechanisms, or detail how to leverage appropriate mechanisms provided by underlying protocol layers, which can be used by an ALTO server to inform clients about an impending or occurring overload situation, and redirect them to another ALTO server.

REQ. ARv11-33: An ALTO client protocol specification MUST specify mechanisms, or detail how to leverage appropriate mechanisms provided by underlying protocol layers, which can be used by an ALTO server to inform clients about an impending or occurring overload situation, and terminate the conversation with the ALTO client.

REQ. ARv11-34: An ALTO client protocol specification MUST specify mechanisms, or detail how to leverage appropriate mechanisms provided by underlying protocol layers, which can be used by an ALTO server to inform clients about its inability to answer queries due to technical problems or system maintenance, and advise them to retry after an indicated point in time or after an indicated period of time has elapsed.

REQ. ARv11-35: An ALTO client protocol specification MUST specify

mechanisms, or detail how to leverage appropriate mechanisms provided by underlying protocol layers, which can be used by an ALTO server to inform clients about its inability to answer queries due to technical problems or system maintenance, and redirect them to another ALTO server.

REQ. ARv11-36: An ALTO client protocol specification MUST specify mechanisms, or detail how to leverage appropriate mechanisms provided by underlying protocol layers, which can be used by an ALTO server to inform clients about its inability to answer queries due to technical problems or system maintenance, and terminate the conversation with the ALTO client.

Note: The existence of the above-mentioned protocol mechanisms does not imply that an ALTO server must use them when facing an overload, technical problem, or maintenance situation, respectively. Some servers may be unable to use them in that situation, or they may prefer to simply refuse the connection or not to send any answer at all.

3.2. ALTO Server Discovery

An ALTO client protocol is supported by one or more ALTO server discovery mechanisms, which may be used by ALTO clients in order to determine one or more ALTO servers, to which ALTO requests can be sent. This section enumerates requirements for an ALTO client, as well as general requirements to be fulfilled by the ALTO server discovery mechanisms.

REQ. ARv11-37: ALTO clients which are embedded in the resource consumer MUST be able to use an ALTO server discovery mechanism, in order to find one or several ALTO servers that can provide ALTO guidance suitable for the resource consumer. This mode of operation is called "resource consumer initiated ALTO server discovery".

REQ. ARv11-38: ALTO clients which are embedded in a resource directory and perform third-party ALTO queries on behalf of a remote resource consumer MUST be able to use an ALTO server discovery mechanism, in order to find one or several ALTO servers that can provide ALTO guidance suitable for the respective resource consumer. This mode of operation is called "third-party ALTO server discovery".

REQ. ARv11-39: ALTO clients MUST be able to perform resource consumer initiated ALTO server discovery, even if they are located behind a network address translator (NAT).

REQ. ARv11-40: ALTO clients MUST be able to perform third-party ALTO server discovery, even if they are located behind a network address

translator (NAT).

REQ. ARv11-41: ALTO clients MUST be able to perform third-party ALTO server discovery, even if the resource consumer, on behalf of which the ALTO query will be sent, is located behind a network address translator (NAT).

REQ. ARv11-42: ALTO server discovery mechanisms SHOULD leverage an existing protocol or mechanism, such as DNS, DHCP, or PPP based automatic configuration, etc. A single mechanism with a broad spectrum of applicability SHOULD be preferred over several different mechanisms with narrower scopes.

REQ. ARv11-43: Every ALTO server discovery mechanism SHOULD be able to return the respective contact information for multiple ALTO servers.

REQ. ARv11-44: Every ALTO server discovery mechanism SHOULD be able to indicate preferences for each returned ALTO server contact information.

3.3. Security and Privacy

REQ. ARv11-45: An ALTO client protocol specification MUST specify mechanisms for the authentication of ALTO servers, or how to leverage appropriate mechanisms provided by underlying protocol layers.

REQ. ARv11-46: An ALTO client protocol specification MUST specify mechanisms for the authentication of ALTO clients, or how to leverage appropriate mechanisms provided by underlying protocol layers.

REQ. ARv11-47: An ALTO client protocol specification MUST specify mechanisms for the encryption of messages, or how to leverage appropriate mechanisms provided by underlying protocol layers.

REQ. ARv11-48: The operator of an ALTO server MUST NOT assume that an ALTO client will implement mechanisms or comply with rules that limit the ALTO client’s ability to redistribute information retrieved from the ALTO server to third parties.

REQ. ARv11-49: An ALTO client protocol MUST support different levels of detail in queries and responses, in order to protect the privacy of users, to ensure that the operators of ALTO servers and other users of the same application cannot derive sensitive information.

REQ. ARv11-50: An ALTO client protocol MAY include mechanisms that can be used by the ALTO client when requesting guidance to specify the resource (e.g., content identifiers) it wants to access. An ALTO

server MUST provide adequate guidance even if the ALTO client prefers not to specify the desired resource (e.g., keeps the data field empty). The mechanism MUST be designed in a way that the operator of the ALTO server cannot easily deduce the resource identifier (e.g., file name in P2P file sharing) if the ALTO client prefers not to specify it.

REQ. ARv11-51: An ALTO client protocol specification MUST specify appropriate mechanisms for protecting the ALTO service against DoS attacks, or how to leverage appropriate mechanisms provided by underlying protocol layers.

4. IANA Considerations

This requirements document does not mandate any immediate IANA actions. However, such IANA considerations may arise from future ALTO specification documents which try to meet the requirements given here.

5. Security Considerations

5.1. High-level security considerations

High-level security considerations for the ALTO service can be found in the "Security Considerations" section of the ALTO problem statement document [RFC5693].

5.2. Information Disclosure Scenarios

The unwanted disclosure of information is one key concern related to ALTO. This section presents a classification and discussion of information disclosure scenarios and potential countermeasures.

5.2.1. Classification of Information Disclosure Scenarios

- (1) Excess disclosure of ALTO server operator's data to an authorized ALTO client. The operator of an ALTO server has to feed information, such as tables mapping host group descriptors to host characteristics attributes, into the server, thereby enabling it to give guidance to ALTO clients. Some operators might consider the full set of this information confidential (e.g., a detailed map of the operator's network topology), and might want to disclose only a subset of it or somehow obfuscated information to an ALTO client.

- (2) Disclosure of the application behavior to the ALTO server. The operator of an ALTO server could infer the application behavior (e.g., content identifiers in P2P file sharing applications, or lists of resource providers that are considered for establishing a connection) from the ALTO queries sent by an ALTO client.

- (3) Disclosure of ALTO server operator's data (e.g., network topology information) to an unauthorized third party. There are a three sub-cases here:

  * (3a) An ALTO server sends the information directly to an unauthorized ALTO client.

  * (3b) An unauthorized party snoops on the data transmission from the ALTO server to an authorized ALTO client.

  * (3c) An authorized ALTO client knowingly forwards the information it had received from the ALTO server to an unauthorized party.

- (4) Disclosure of the application behavior to an unauthorized third party.

- (5) Excess retrieval of ALTO server operator's data by collaborating ALTO clients. Several authorized ALTO clients could ask an ALTO server for guidance, and redistribute the responses among each other (see also case 3c). By correlating the ALTO responses they could find out more information than intended to be disclosed by the ALTO server operator.

5.2.2. Discussion of Information Disclosure Scenarios

Scenario (1) may be addressed by the ALTO server operator choosing the level of detail of the information to be populated into the ALTO server and returned in the responses. For example, by specifying a broader address range (i.e., a shorter prefix length) than a group of hosts in question actually uses, an ALTO server operator may control to some extent how much information about the network topology is disclosed. Furthermore, access control mechanisms for filtering ALTO responses according to the authenticated ALTO client identity might be installed in the ALTO server, although this might not be effective given the lack of efficient mechanisms for addressing (3c) and (5), see below.

(2) can and needs to be addressed in several ways: If the ALTO client is embedded in the resource consumer, the resource consumer's IP address (or the "public" IP address of the outermost NAT in front of the resource consumer) is disclosed to the ALTO server as a matter of principle, because it is in the source address fields of the IP headers. By using a proxy, the disclosure of source addresses to the ALTO server can be avoided at the cost of disclosing them to said proxy. If, in contrast, the ALTO client is embedded in a third party (e.g., a resource directory) which issues ALTO requests on behalf of resource consumers, it is possible to hide the exact addresses of the resource consumers from the ALTO server, e.g., by zeroing-out or randomizing the last few bits of IP addresses. However, there is the potential side effect of yielding inaccurate results.

The disclosure of candidate resource providers' addresses to the ALTO server can be avoided by allowing ALTO clients to use the target-independent query mode. In this mode of operation, guiding information (e.g., "maps") is retrieved from the ALTO server and used entirely locally by the ALTO client, i.e., without sending host location attributes of candidate resource providers to the ALTO server. In the target-aware query mode, this issue can be addressed by ALTO clients through obfuscating the identity of candidate resource consumers, e.g., by specifying a broader address range (i.e., a shorter prefix length) than a group of hosts in question

actually uses, or by zeroing-out or randomizing the last few bits of IP addresses. However, there is the potential side effect of yielding inaccurate results.

(3a), (3b), and (4) may be addressed by authentication, access control, and encryption schemes for the ALTO client protocol. However, deployment of encryption schemes might not be effective given the lack of efficient mechanisms for addressing (3c) and (5), see below.

Straightforward authentication and encryption schemes will not help solving (3c) and (5), and there is no other simple and efficient mechanism known. The cost of complex approaches, e.g., based on digital rights management (DRM), might easily outweigh the benefits of the whole ALTO solution, and therefore they are not considered as a viable solution. That is, ALTO server operators must be aware that (3c) and (5) cannot be prevented from happening, and therefore they should feed only such data into an ALTO server, which they do not consider sensitive with respect to (3c) and (5).

These insights are reflected in the requirements in this document.

5.3. Security Requirements

For a set of specific security requirements please refer to Section 3.3 of this document.

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.

6.2. Informative References

[ALTO-charter] Marocco, E. and V. Gurbani, "Application-Layer Traffic Optimization (ALTO) Working Group Charter (http://tools.ietf.org/wg/alto/charters?item=charter-alto-2011-04-28.txt)", April 2011.

[RFC4787] Audet, F. and C. Jennings, "Network Address Translation (NAT) Behavioral Requirements for Unicast UDP", BCP 127, RFC 4787, January 2007.

[RFC5382] Guha, S., Biswas, K., Ford, B., Sivakumar, S., and P. Srisuresh, "NAT Behavioral Requirements for TCP", BCP 142, RFC 5382, October 2008.

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

Appendix A. Contributors List and Acknowledgments

The initial version of this document was co-authored by Laird Popkin.

The authors would like to thank

- Vijay K. Gurbani <vkg@alcatel-lucent.com>
- Enrico Marocco <enrico.marocco@telecomitalia.it>

for fostering discussions that lead to the creation of this document, and for giving valuable comments on it.

The authors were supported by the following people, who have contributed to this document:

- Richard Alimi <ralimi@google.com>
- Zoran Despotovic <despotovic@docomolab-euro.com>
- Jason Livingood <Jason_Livingood@cable.comcast.com>
- Saverio Niccolini <saverio.niccolini@nw.neclab.eu>
- Michael Scharf <michael.scharf@alcatel-lucent.com>
- Nico Schwan <nico.schwan@alcatel-lucent.com>
- Jan Seedorf <jan.seedorf@nw.neclab.eu>

The authors would like to thank the members of the P2PI and ALTO mailing lists for their feedback.

Laird Popkin and Y. Richard Yang are grateful to the many contributions made by the members of the P4P working group and Yale Laboratory of Networked Systems. The P4P working group is hosted by DCIA.

Martin Stiemerling is partially supported by the COAST project (COntent Aware Searching, retrieval and sTreaming, http://www.coast-fp7.eu), a research project supported by the European Commission under its 7th Framework Program (contract no. 248036). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the COAST project or the European Commission.

Authors' Addresses

Sebastian Kiesel (editor)
University of Stuttgart Computing Center
Networks and Communication Systems Department
Allmandring 30
70550 Stuttgart
Germany

Email: ietf-alto@skiesel.de
URI: http://www.rus.uni-stuttgart.de/nks/

Stefano Previdi
Cisco Systems, Inc.

Email: sprevidi@cisco.com

Martin Stiemerling
NEC Laboratories Europe

Email: martin.stiemerling@neclab.eu
URI: http://ietf.stiemerling.org

Richard Woundy
Comcast Corporation

Email: Richard_Woundy@cable.comcast.com

Yang Richard Yang
Yale University

Email: yry@cs.yale.edu

ALTO
Internet-Draft
Intended status: Standards Track
Expires: January 12, 2012

S. Kiesel
University of Stuttgart
M. Stiemerling
NEC Europe Ltd.
N. Schwan
M. Scharf
Alcatel-Lucent Bell Labs
H. Song
Huawei

July 11, 2011

ALTO Server Discovery
draft-ietf-alto-server-discovery-01

Abstract

The goal of Application-Layer Traffic Optimization (ALTO) is to provide guidance to applications, which have to select one or several hosts from a set of candidates that are able to provide a desired resource.

Entities seeking guidance need to discover and possibly select an ALTO server to ask. This is called ALTO server discovery. This memo describes an ALTO server discovery mechanism based on several alternative mechanisms that are applicable in a diverse set of ALTO deployments.

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 January 12, 2012.

Copyright Notice

Table of Contents

1. Introduction

The goal of Application-Layer Traffic Optimization (ALTO) is to provide guidance to applications, which have to select one or several hosts from a set of candidates, that are able to provide a desired resource [RFC5693]. The requirements for ALTO are itemized in [I-D.ietf-alto-reqs]. ALTO is realized by a client-server protocol. ALTO clients send queries to ALTO servers, in order to solicit guidance.

ALTO clients have to discover suitable ALTO servers. Therefore the output of the herein defined ALTO discovery procedure tells the ALTO client which ALTO servers to send the queries to. The ALTO discovery procedure, as part of the ALTO client, can be embedded in the resource consumer, which will eventually access the desired resource. As an alternative, they can be embedded in a resource directory, which assists resource consumers in finding appropriate resource providers. In some specific peer-to-peer application protocols these resource directories are called "trackers". Finally the ALTO server discovery procedure can be embedded in the resource consumer, whereas the ALTO client is embedded in the resource directory. ALTO queries, which are issued by a resource directory on behalf of a resource consumer, are referred to as third-party ALTO queries. The various possibilities to place ALTO servers and the placement of ALTO clients is discussed in [I-D.ietf-alto-deployments].

No matter where ALTO server and client are located, clients have to first find out if there is an ALTO server deployed that is in charge for them, and second they have to get the contact information of that server, i.e., the IP address, port number, and probably transport protocol (which defaults to TCP for the ALTO protocol specification [I-D.ietf-alto-protocol]).

The goal of this memo is to propose a uniform mechanism for all types of ALTO client deployments that is implementable and deployable at a fast pace, i.e., without creating other deployment dependencies for ALTO. We propose a schema which employs the UNAPTR mechanism [RFC4848] to determine the URI of the ALTO server and where mutliple input methods to the UNAPTR process can be used.

Comments and discussions about this memo should be directed to the ALTO working group: alto@ietf.org.

1.1. History

[RFC editor’s note: Please remove this section before publication.]

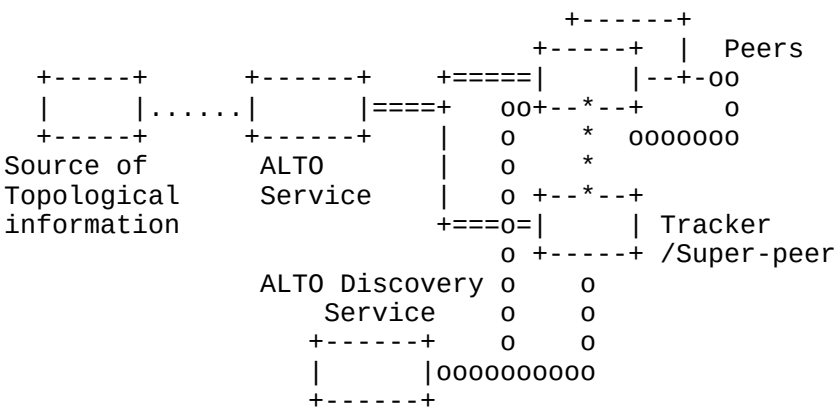
This document represents a merge of features from two previous

drafts:

- draft-kiesel-alto-3pdisc-04
- draft-song-alto-server-discovery-03

1.2. Discovery Scenarios

Figure 1 below shows an overview on the different entities of a generic ALTO framework. The ALTO Server discovery mechanism is used by the peer-to-peer (P2P) application in order retrieve the point of contact of the ALTO Service.



Legend:

=== ALTO query protocol
ooo ALTO service discovery protocol
*** Application protocol (out of scope)
... Provisioning or initialization (out of scope)

Figure 1: ALTO Discovery Overview

Hereby the ALTO service discovery scenarios are classified into two types: one is the ALTO server discovery by the resource consumer, and the other is the ALTO server discovery by a third party, such as application trackers. Before the specification of the discovery mechanism the following section illustrates and discusses both scenarios.

1.2.1. ALTO Server Discovery by Resource Consumers

The ALTO service discovery in some scenarios needs to be performed by the resource consumer itself. In particular in P2P applications without a tracker like DHTs and other conventional client/server applications.

In addition also P2P application which are tracker based may embed the ALTO client into the resource consumer to allow peers a selection of peers after retrieving the peer list from the application tracker. Another option is that the resource consumer peer sends its ALTO server address information to the application tracker or any other third party entity, which in turn will contact the specific ALTO server in order to retrieve ALTO guidance on behalf of the resource consumer.

The following figure illustrates this scenario, showing the relationship between the different entities as discussed before.

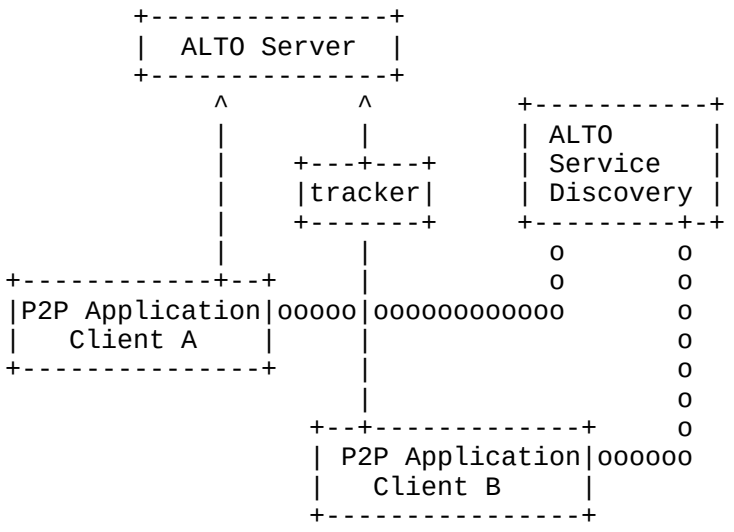


Figure 2: Resource Consumer ALTO Server Discovery (Example)

1.2.2. ALTO Server Discovery by a Third Party

Some P2P applications have trackers, and these applications might not need to have their clients looking for the ALTO server guidance. In these scenarios trackers query the ALTO servers for guidance themselves, and then return the final ranked result to the application clients. However, application clients are distributed among different network operators and autonomous systems. Trackers

thus need to find different ALTO servers for the clients located in different operator networks or autonomous systems. In such scenarios the discovery is thus not performed by the resource consumer, but a third party entity on behalf of the resource consumer.

Figure 3 shows an example for a third party ALTO server discovery. For Client1 (1), the tracker has not cached yet the mapping between Client1's network operator and its ALTO server address, so it uses the ALTO Discovery Service to determine the address of the ALTO server in that operator's domain (2). Then the tracker interacts with ALTO Server1 (3)(4) on behalf of Client1 (to get the network map and cost map), finally, the ranked list is sent back to Client1 (5). For Client2, the tracker has cached the mapping between Client2's network operator and ALTO Server2's address, so it does not need to perform the discovery process (which are the labels (a),(b), (c), and (d)). If the application tracker already has the network map and cost map from ALTO Server2, then it does not need to query the ALTO Server for network map and cost map frequently.

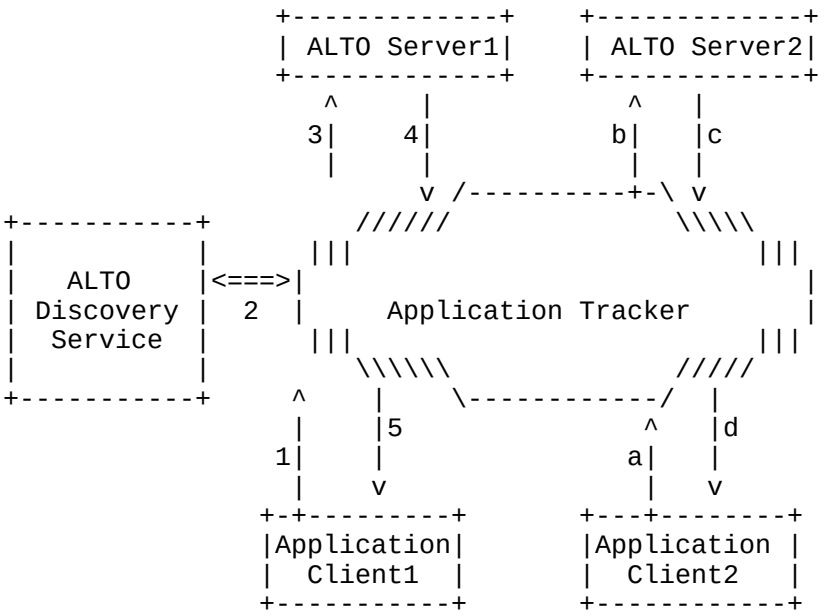


Figure 3: Third Party ALTO Server Discovery (Example)

1.3. Pre-Conditions

The whole document assumes certain pre-conditions, such as:

- The ALTO server discovery procedure is executed on a per IP address base. Multiple IP addresses per interface or multiple IP addresses assigned to different IP interfaces require to repeat the procedure for every IP address. It may be fine to group IP addresses according their domain suffixes and to perfom the procedure for such a group. However, this is out of scope of this document.[Editor's note: this may relate to the work of the MIF WG]

- The ALTO server discovery procedure is executed on a per IP family base, i.e., seperate for IPv4 and IPv6. It is up to the ALTO client to decide which of the possible multiple results of different IP address families to use. The choice of whether to use IPv4 or IPv6 is out of scope of this document.

- A change of the IP address at an interface invalidates the result of the ALTO server discovery procedure. For instance, if the IP address assigned to a mobile host changes due to host mobility, it is required to run the ALTO server discovery procedure for the new IP address without relying on earlier gained information.

2. Protocol Overview

We define multiple alternatives to discover the IP address of the ALTO server, as there are a number of ways possible how such information can be provided to the ALTO client. The choice of method is up to the local network deployment. For instance, there can be deployments where the ALTO server in charge for ALTO client is provisioned by the network operator and communicated to the ALTO client's host via a DHCP option, while in other deployments no such means may exist. It should be noted that there is no silver bullet solution to the ALTO server discovery, as there too many deployment scenarios in the server discovery space.

The following figure illustrates the different protocols that are used to find the URI of a suitable ALTO server.

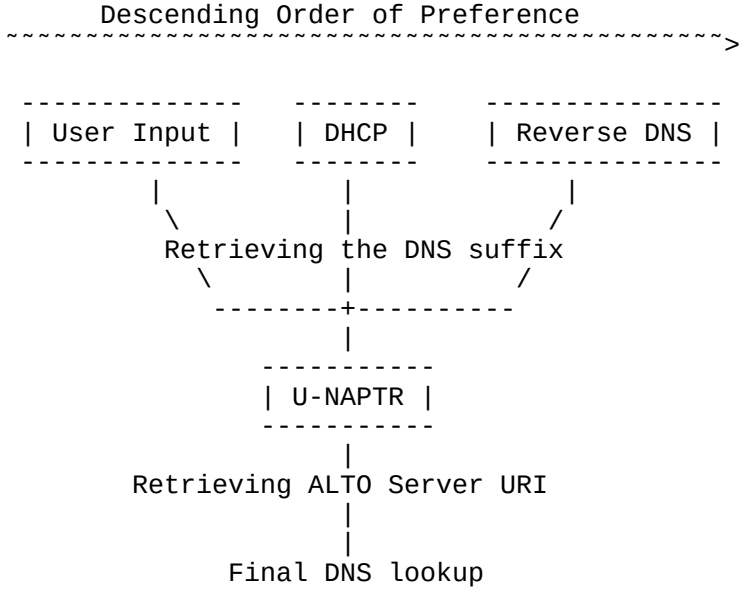


Figure 4: Protocol Overview

Figure 4 illustrates the U-NAPTR based resolution process to retrieve the ALTO Server URL. As a precondition for resolution the U-NAPTR process needs the right domain name as input. This domain name is determined by the IP address of the client and the DNS suffix of the access network where the client is registered in. In order to retrieve the DNS suffix we specify three options, as are listed in descending order of preference:

User input: a user may manually specify the DNS suffix on its own, either to access a 3rd party ALTO service provider or as it does know such information. This input may also origin from a web page where the user downloads the configuration which is loaded as user input.

DHCP: a network provider provides the DNS suffix through a DHCP option.

Reverse DNS: the DNS system can be used to retrieve the DNS suffix through reverse lookup of an FQDN associated with an IP address. This is the last resort if all other options failed.

3. Retrieving the URI by U-NAPTR

This section specifies the U-NAPTR based resolution process. To start the U-NAPTR resolution process a domain name is required as input. Thus the section is devided into two parts: Section 3.1 describes the U-NAPTR resolution process itself. How the client identifies this DNS suffix of the access network where the resource consumer is registered in is described in Section 3.2.

3.1. U-NAPTR Resolution

ALTO servers are identified by U-NAPTR/DDDS (URI-Enabled NAPTR/ Dynamic Delegation Discovery Service) [RFC4848] application unique strings, in the form of a DNS name. An example is 'altoserver.example.com'.

Clients need to use the U-NAPTR [RFC4848] specification described below to obtain a URI (indicating host and protocol) for the applicable ALTO service. In this document, only the HTTP and HTTPS URL schemes are defined, as the ALTO protocol specification defines the access over both protocols, but no other [I-D.ietf-alto-protocol]. Note that the HTTP URL can be any valid HTTP(s) URL, including those containing path elements.

The following two DNS entries show the U-NAPTR resolution for "example.com" to the HTTPS URL https://altoserver.example.com/secure or the HTTP URL http://altoserver.example.com, with the former being preferred.

```
example.com.

IN NAPTR 100  10   "u"    "ALTO:https"
     "!.*!https://altoserver.example.com/secure!"  ""

IN NAPTR 200  10   "u"    "ALTO:http"
     "!.*!http://altoserver.example.com!"  ""
```

3.2. Retrieving the Domain Name

The U-NAPTR resolution process requires a domain name as input. The algorithm that is applied to determine this domain name is described in this section. We specify three different options. In option 1 the user manually configures a specific ALTO service instance that he wants to use. Option 2 defines a DHCP option to allow the network service provider a remote configuration of the client. In option 3 the client tries to get the domain name by performing a reverse DNS lookup on its IP address.

The resource consumer may have private IP addresses and public IP addresses and depending on the deployment it might be necessary to determine for all IP addresses the ALTO server in charge of. To determine its public IP address the resource consumer may need to use STUN[RFC5389] or BEP24[bep24]. For the following examples we assume that the IP address of the resource consumer is a.b.c.d.

3.2.1. Option 1: User input

A user may want to use a third party ALTO service instance. Therefore we allow the user to specify a DNS suffix on its own, for example in a config file option. The DNS suffix given by the user is combined with the IP address of the resource consumer to allow the third party ALTO service to direct the client to a suitable ALTO server based on the location of the client. A possible DNS suffix entered by the user may be:

myaltoprovider.org

This DNS suffix is prepended with the IP address of the resource consumer in reverse order to compose the domain name used for the final U-NAPTR lookup Section 3.1. In case there are multiple ALTO servers deployed, the third party ALTO service instance can direct the ALTO client to the ALTO server closest to the client based on the IP address.

Multiple lookups with different domain names might be necessary to complete the U-NAPTR resolution process. If there is no response for a lookup the domain name is shortened by one part for the succeeding lookup, until a lookup is successful, as for example

d.c.b.a.myaltoprovider.org.

c.b.a.myaltoprovider.org.

b.a.myaltoprovider.org.

a.myaltoprovider.org.

myaltoprovider.org.

3.2.2. Option 2: DHCP

As a second option network operators can configure the domain name to be used for service discovery within an access network. RFC 5986[RFC5986] defines DHCP IPv4 and IPv6 access network domain name options that identify a domain name that is suitable for service discovery within the access network. The ALTO server discovery

procedure uses these DHCP options to retrieve the domain name as an input for the U-NAPTR resolution. One example could be:

example.com

3.2.3. Option 3: Reverse DNS Lookup

The last option to get the domain name is to use a DNS PTR query for the IP address of the resource consumer. The local DNS server resolves the IP address to the FQDN that also contains the DNS suffix for the respective IP address. A possible answer for a PTR lookup for d.c.b.a.in-addr.apra might be, for example:

d-c-b-a.dsl.westcoast.myisp.net

This domain name can be used for the final U-NAPTR lookup Section 3.1. If there is no response to the lookup the domain name is shortened by one part for one succeeding lookup. If there is still no response we consider the reverse lookup being failed. The domain names used for the example as described above are:

d-c-b-a.dsl.westcoast.myisp.net.

dsl.westcoast.myisp.net.

4. Applicability

This section discusses the applicability of the proposed solution with respect to the resource consumer server discovery and the third party deployment scenarios. Each section discusses the proposed steps that are needed to determine the ALTO Server URI.

4.1. Applicability for Resource Consumer Server Discovery

In this scenario the ALTO server discovery procedure is performed by the resource consumer, for example a peer in a P2P system. After the discovery the peer does the ALTO query on its own, or it might share the ALTO server contact information with a third party, for example a tracker, which then executes the ALTO query on behalf of the peer.

To complete the ALTO server discovery process the resource consumer first SHOULD check whether the user has provider the domain name through manual configuration. If this is not the case the next step SHOULD be to check for the access network domain name DHCP option (Section 3.2.2). Finally the client SHOULD try to retrieve the domain name by the last option, the DNS reverse lookup on its IP address as described in Section 3.2.3.

A client can have several candidate IP addresses that it may use for the discovery process. For example if it is located behind a NAT, a private and a public IP address may be used for the discovery process. It depends on the deployment scenario which of the IP addresses is the correct one. Thus it is out-of-scope of this document to specify how exactly the client finds the right IP address. However in the following we list methods that may be used in order to determine these candidate IP addresses. Generally in P2P environments peers already have implemented mechanisms for NAT-traversal. This includes proprietary solutions to determine a peer’s public IP address, for example by asking a neighbour peer about its record of the own IP address. Non-proprietary solutions that are favorable include the Session Traversal Utilities for NAT (STUN) [RFC5986] protocol to determine the public address. If the client is behind a residential gateway another option may be to use Universal Plug and Play (UPnP) [UPnP-IGD-WANIPConnection1] or the NAT Port Mapping Protocol (NAT-PMP) [I-D.cheshire-nat-pmp].

In case the ALTO discovery client has determined the domain name through one of the described options it proceedes with the U-NAPTR lookup as described in Section 3.1.

4.2. Applicability for Third Party Server Discovery

In case of the third party server discovery deployment scenario the entity performing the ALTO server discovery process is different from the resource consumer. Typically the resource consumer is a peer whereas the ALTO client is a resource directory which seeks for ALTO guidance on behalf of the peer. Another use case for the third party discovery is an application that looks for ALTO guidance transparently for the resource consumer, for example a CDN.

Here the ALTO server discovery process can also retrieve guidance through the DHCP option or manual user configuration, but only if the provided discovery information is forwarded by the resource consumer to the third party entity. In this case, additional mechanisms for the forwarding of this discovery information need to be specified. However these mechanisms are out of scope of this doument.

If the third party entity cannot obtain this discovery information, the ALTO server discovery process relies on retrieving the domain name used as input to the U-NAPTR lookup through reverse DNS lookup of the IP address of the resource consumer as described in Section 3.2.3. Usually the third party entity already knows the IP address of the resource consumer which was used to establish the initial connection. In general this IP address is a public address, either of the resource consumer or of the last NAT on the path to the ALTO client. This makes the IP address a good candidate for the DNS PTR query. Thus, we expect that the DNS query will be successfully resolved to the FQDN of the domain where the resource consumer is registered in.

In case the resource consumer needs guidance for a different IP address, for example one from a private network, we recommend that the resource consumer discovers the server itself and forwards the ALTO server contact information directly to the third party entity, which in turn can then do the third party ALTO query. Again, forwarding the contact information from the resource consumer to the third party entity is out of scope of this document.

5. Deployment Considerations

The mechanism specified in this document needs some configuration effort in order to work properly.

5.1. Reverse DNS Lookup

Especially the domain name retrieved through the reverse DNS lookup (PTR records) and the U-NAPTR entry need to be coordinated. In this section we discuss this configuration for different scenarios.

5.1.1. Private customers or very small businesses

For private customers and very small businesses that are DSL or cable customers often a dynamically assigned IP address is provisioned. Here, the reverse DNS lookup (PTR records) are controlled by the ISP and they point to the ISP's domain, e.g.:

p5B203EA1.dip.t-dialin.net.

dslb-084-056-144-100.pools.arcor-ip.net.

187-4-222-157.bnut3700.dsl.brasiltelecom.net.br.

65-154-39-69.ispnetbilling.com.

197-151-94-178.pool.ukrtel.net.

In this case, it would be the responsibility of the respective ISP to provide U-NAPTR entries for the DNS suffix without the endhost part, e.g.:

dip.t-dialin.net.

pools.arcor-ip.net.

bnut3700.dsl.brasiltelecom.net.br.

ispnetbilling.com.

pool.ukrtel.net.

5.1.2. Medium-size customer networks

The second class of customers have their own DNS domain but only one single upstream ISP, e.g.:

(1) ISP my-isp.net assigns an IP address a.b.c.d to its customer

(2) The customer decides that reverse mapping for a.b.c.d should be whatever.customerdomain.com

(3) If the customer wants to support ALTO, he has to ask the ISP for the URI of the ISP's ALTO server which can give guidance to a.b.c.d. Assume that ISP replies it is http://altoserver.my-isp.net

(4) The customer establishes a U-NAPTR entry for his domain

```
customerdomain.com.   IN NAPTR 200  10   "u"    "ALTO:http"
     "!.*!http://altoserver.my-isp.net!"  ""
```

5.1.3. Large Customers

For very large customers with multiple upstream connections we assume that they have their very own traffic optimization policies and thus run their own ALTO server anyway. In this case they need to manage their DNS entries accordingly.

5.2. DHCP option for DNS Suffix

Section 3.2.2 describes the usage of a DHCP option which allows the network operator of the network where the ALTO client is attached to, to provide a DNS suffix. However, this assumes that this particular DHCP option is correctly passed from the DHCP server to the actual host with the ALTO client, and that the particular host understands this DHCP option. This memo assumes the client to be able to understand the proposed DHCP option, otherwise there is no further use of the DHCP option, but the client has to use the other proposed mechanisms.

There are well-known issues with the handling of DHCP options in home gateways. One issue is that unkown DHCP options are not passed through some home gateways, effectively eliminating the DHCP option.

Another well-known issues is the usage of home gateway specific DNS suffixes which "override" the DNS suffix provided by the network operator. For instance, a host behind a home gateway may receive a DNS suffix ".local" instead of "example.com". This suffix is not usuable for the server discovery procedure.

[Editor's note: This section needs references about the well-known issues with home gateways and it relates to the FUN activity on home gateways which needs to be explored further.

6. IANA Considerations

This document registers the following U-NAPTR application service tag:

Application Service Tag: ALTO

Defining Publication: The specification contained within this document.

This document registers the following U-NAPTR application protocol tags:

- Application Protocol Tag: http

  Defining Publication: RFC 2616 [RFC2616]

- Application Protocol Tag: https

  Defining Publication: RFC 2818 [RFC2818]

7. Security Considerations

7.1. General

This is still to be done in later revision of this draft, as the draft evolves heavily right now.

7.2. For U-NAPTR

The address of an ALTO server is usually well-known within an access network; therefore, interception of messages does not introduce any specific concerns.

The primary attack against the methods described in this document is one that would lead to impersonation of an ALTO server since a device does not necessarily have a prior relationship with an ALTO server.

An attacker could attempt to compromise ALTO discovery at any of three stages:

1. providing a falsified domain name to be used as input to U-NAPTR

2. altering the DNS records used in U-NAPTR resolution

3. impersonation of the ALTO server

This document focuses on the U-NAPTR resolution process and hence this section discusses the security considerations related to the DNS handling. The security aspects of obtaining the domain name that is used for input to the U-NAPTR process is described in respective documents, such as [RFC5986].

The domain name that is used to authenticated the ALTO server is the domain name in the URI that is the result of the U-NAPTR resolution. Therefore, if an attacker was able to modify or spoof any of the DNS records used in the DDDS resolution, this URI could be replaced by an invalid URI. The application of DNS security (DNSSEC) [RFC4033] provides a means to limit attacks that rely on modification of the DNS records used in U-NAPTR resolution. Security considerations specific to U-NAPTR are described in more detail in [RFC4848].

An "https:" URI is authenticated using the method described in Section 3.1 of [RFC2818]. The domain name used for this authentication is the domain name in the URI resulting from U-NAPTR resolution, not the input domain name as in [RFC3958]. Using the domain name in the URI is more compatible with existing HTTP client software, which authenticate servers based on the domain name in the URI.

An ALTO server that is identified by an "http:" URI cannot be authenticated. If an "http:" URI is the product of the ALTO discovery, this leaves devices vulnerable to several attacks. Lower layer protections, such as layer 2 traffic separation might be used to provide some guarantees.

8. Open Issues

Here are a few open issues to be clarified:

Handling of reverse DNS lookups for IPv6: Refer to [RFC4472] for a discussion about the issues.

Missing reverse DNS entries for an IP address: There may be cases where the reverse DNS lookup does not yield any result. However, this will leave the ALTO client with no choice, other than giving up. This needs better documentation.

How to handled multiple results: For instance, a host behind a NAT that yields an ALTO server in the private IP address domain and one in the public IP address domain. Whom to ask?

Normative Language The current version of this memo lacks the proper normative language in many places.

9. Conclusion

This document describes a general ALTO server discovery process and discusses how the process can be applied in different deployment scenarios, including the resouce consumer discovery as well as the third party discovery.

10. References

10.1. Normative References

[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.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC3958] Daigle, L. and A. Newton, "Domain-Based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS)", RFC 3958, January 2005.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

10.2. Informative References

[I-D.cheshire-nat-pmp] Cheshire, S., "NAT Port Mapping Protocol (NAT-PMP)", draft-cheshire-nat-pmp-03 (work in progress), April 2008.

[I-D.ietf-alto-deployments] Stiemerling, M. and S. Kiesel, "ALTO Deployment Considerations", draft-ietf-alto-deployments-01 (work in progress), March 2011.

[I-D.ietf-alto-protocol] Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-04 (work in progress), May 2010.

[I-D.ietf-alto-reqs] Kiesel, S., Previdi, S., Stiemerling, M., Woundy, R., and Y. Yang, "Application-Layer Traffic Optimization (ALTO) Requirements", draft-ietf-alto-reqs-08 (work in progress), March 2011.

[RFC4472] Durand, A., Ihren, J., and P. Savola, "Operational Considerations and Issues with IPv6 DNS", RFC 4472, April 2006.

[RFC4848] Daigle, L., "Domain-Based Application Service Location

Using URIs and the Dynamic Delegation Discovery Service (DDDS)", RFC 4848, April 2007.

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

[RFC5986] Thomson, M. and J. Winterbottom, "Discovering the Local Location Information Server (LIS)", RFC 5986, September 2010.

[UPnP-IGD-WANIPConnection1] UPnP Forum, "Internet Gateway Device (IGD) Standardized Device Control Protocol V 1.0: WANIPConnection:1 Service Template Version 1.01 For UPnP Version 1.0", DCP 05-001, November 2001.

[bep24] Harrison, D., "Tracker Returns External IP", BEP http://bittorrent.org/beps/bep_0024.html.

Appendix A. Contributors List and Acknowledgments

The initial version of this document was co-authored by Marco Tomsu <marco.tomsu@alcatel-lucent.com>.

Hannes Tschofenig provided the initial input to the U-NAPTR solution part. Hannes and Martin Thomson provided excellent feedback and input to the server discovery.

The authors would also like to thank the following persons for their contribution to this document or its predecessors: Richard Alimi, David Bryan, Roni Even, Gustavo Garcia, Jay Gu, Xingfeng Jiang, Enrico Marocco, Victor Pascual, Y. Richard Yang, Yu-Shun Wang, Yunfei Zhang, Ning Zong.

Marco Tomsu and Nico Schwan are partially supported by the ENVISION project (http://www.envision-project.org), a research project supported by the European Commission under its 7th Framework Program (contract no. 248565). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ENVISION project or the European Commission.

Michael Scharf is supported by the German-Lab project (http://www.german-lab.de) funded by the German Federal Ministry of Education and Research (BMBF).

Martin Stiemerling is partially supported by the COAST project (COntent Aware Searching, retrieval and sTreaming, http://www.coast-fp7.eu), a research project supported by the European Commission under its 7th Framework Program (contract no. 248036). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the COAST project or the European Commission.

Authors' Addresses

Sebastian Kiesel
University of Stuttgart Computing Center
Allmandring 30
Stuttgart 70550
Germany

Email: ietf-alto@skiesel.de
URI: http://www.rus.uni-stuttgart.de/nks/

Martin Stiemerling
NEC Laboratories Europe
Kurfuerstenanlage 36
Heidelberg 69115
Germany

Phone: +49 6221 4342 113
Email: martin.stiemerling@neclab.eu
URI: http://ietf.stiemerling.org

Nico Schwan
Alcatel-Lucent Bell Labs
Lorenzstrasse 10
Stuttgart 70435
Germany

Email: nico.schwan@alcatel-lucent.com
URI: www.alcatel-lucent.com/bell-labs

Michael Scharf
Alcatel-Lucent Bell Labs
Lorenzstrasse 10
Stuttgart 70435
Germany

Email: michael.scharf@alcatel-lucent.com
URI: www.alcatel-lucent.com/bell-labs

Haibin Song
Huawei

Email: melodysong@huawei.com

Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 19, 2011

B. Niven-Jenkins, Ed.
Velocix (Alcatel-Lucent)
G. Watson
BT
N. Bitar
Verizon
J. Medved
Juniper Networks
S. Previdi
Cisco Systems
June 17, 2011

Use Cases for ALTO within CDNs
draft-jenkins-alto-cdn-use-cases-01

Abstract

For some time, Content Distribution Networks (CDNs) have been used in the delivery of some Internet services (e.g. delivery of websites, software updates and video delivery) as they provide numerous benefits including reduced delivery cost for cacheable content, improved quality of experience for end users and increased robustness of delivery.

In order to derive the optimal benefit from a CDN it is preferable to deliver content from the servers (caches) that are "closest" to the End User requesting the content, where "closest" may be as simple as "geographical or network distance" combined with CDN server load within a location, but may also consider other more complex combinations of metrics and CDN or Network Service Provider (NSP) policies.

There are a number of different ways in which a CDN may obtain the necessary network topology and/or cost information to allow it to serve End Users from the most optimal servers/locations, such as static configuration, passively listening to routing protocols directly, active probing of underlying network(s), or obtaining topology and cost by querying an information service such as the ALTO map & cost services.

This document describes the use cases for a CDN to be able to obtain network topology and cost information from an ALTO server(s).

Status of this Memo

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

This Internet-Draft will expire on December 19, 2011.

Copyright Notice

Table of Contents

1. Introduction

For some time, Content Distribution Networks (CDNs) have been used in the delivery of some Internet services (e.g. delivery of websites, software updates and video delivery) as they provide numerous benefits including reduced delivery cost for cacheable content, improved quality of experience for end users and increased robustness of delivery.

A CDN typically consists of a network of servers often attached to Network Service Provider (NSP) networks. The point of attachment is often as close to content consumers and peering points as economically or operationally feasible in order to decrease traffic load on the NSP backbone and to provide better user experience measured by reduced latency and higher throughput.

As the volume of video and multimedia content delivered over the Internet is rapidly increasing and expected to continue doing so in the future, existing CDN providers are scaling up their infrastructure and many NSPs are deploying their own CDNs. The result of such deployments is typically that more CDN servers are being deployed within NSP networks and those CDN servers are being deployed in locations that are "deeper" (i.e. geographically closer to the NSP’s End Users) than was previously the case.

In order to derive the optimal benefit from a CDN it is preferable to deliver content from the servers (caches) that are "closest" to the End User requesting the content, where "closest" may be as simple as "geographical or network distance" combined with CDN server load within a location, but may also consider other more complex combinations of metrics and CDN or NSP policies.

When CDN servers are deployed outside of an NSP’s network or in a small number of central locations within an NSP’s network a simplified view of the NSP’s topology or an approximation of proximity is typically sufficient to enable the CDN to serve End Users from the optimal server/location. As CDN servers are deployed deeper within NSP networks it becomes necessary for the CDN to have more detailed knowledge of the underlying network topology and costs between network locations in order to enable the CDN to serve End Users from the most optimal servers for the NSP.

There are a number of different ways in which a CDN may obtain the necessary network topology and/or cost information to allow it to serve End Users from the most optimal servers/locations, such as static configuration, passively listening to routing protocols directly, active probing of underlying network(s), or obtaining topology and cost by querying an information service such as the ALTO

map & cost services.

The rest of this document describes the use cases for a CDN to be able to obtain network topology and cost information from an ALTO server(s).

1.1. Terminology

The following terms are taken from [I-D.jenkins-cdni-problem-statement] and repeated here for completeness.

Content Distribution Network (CDN) / Content Delivery Network (CDN): Network infrastructure in which the network elements cooperate at layers 4 through layer 7 for more effective delivery of Content to User Agents. Typically a CDN consists of a Request Routing system, a Distribution System (that includes a set of Surrogates), a Logging System and a CDN control system.

Content Service Provider (CSP): Provides a Content Service to End Users (which the End Users access via a User Agent). A CSP may own the Content made available as part of the Content Service, or may license content rights from another party.

End User (EU): The 'real' user of the system, typically a human but maybe some combination of hardware and/or software emulating a human (e.g. for automated quality monitoring etc.)

Network Service Provider (NSP): Provides network-based connectivity/ services to Users.

Surrogate: A device/function that interacts with other elements of the CDN for the control and distribution of Content within the CDN and interacts with User Agents for the delivery of the Content.

User Agent (UA): Software (or a combination of hardware and software) through which the End User interacts with the Content Service. The User Agent will communicate with the CSP's Service for the selection of content and one or more CDNs for the delivery of the Content. Such communication is not restricted to HTTP and may be via a variety of protocols. Examples of User Agents (non-exhaustive) are: Browsers, Set Top Boxes (STB), Dedicated content applications (e.g. media players), etc.

2. CDN overview

This section provides a high level and simplified overview of the

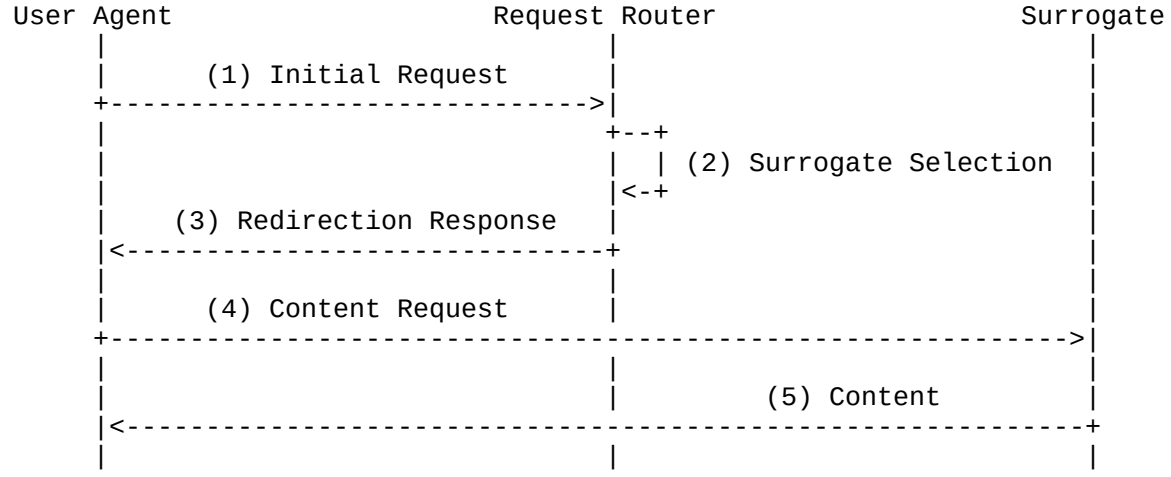
operation of a CDN to help put the ALTO & CDN use cases into context.

A typical CDN consists of a number of functional components, however in the context of ALTO three functional components are of interest: The Request Routing function, the Surrogate (i.e. caching) function and the Origin function.

The Request Routing function within a CDN is responsible for receiving content requests from User Agents, obtaining and maintaining necessary information about a set of candidate Surrogates, and for selecting and redirecting the User Agent to the appropriate Surrogate.

The Surrogate function interacts with other elements of the CDN for the control and distribution of Content within the CDN and interacts with User Agents for the delivery of the Content.

The figure below shows a high level call flow showing the interaction between a User Agent, Request Router and Surrogate for the delivery of content in a single CDN.



1. The User Agent makes an initial request to the CDN. Depending on the type of content being delivered and the configuration of the CDN this request may be an application (e.g. HTTP, RTMP, etc.) level request directly from the User Agent or may be a DNS request via the User Agent's assigned DNS proxy.
2. The Request Router selects an appropriate Surrogate (or set of Surrogates) based on the User Agent's (or its proxy's) IP address, the Request Router's knowledge of the network topology and reachability cost between CDN caches and end users, and any

additional CDN policies.

3. The Request Router responds to the UA's initial request with an appropriate response containing a redirection to the selected cache, for example by returning an appropriate DNS A/AAAA record, a HTTP 302 redirect, etc.
4. The User Agent uses the information provided in the Redirection Response to connect directly to the Surrogate and request the desired content.
5. If CDN policy allows the User Agent to receive the requested content, the Surrogate delivers the content to the User Agent.
   A. [Not Shown] If the Surrogate does not have a copy of the requested content then it obtains it from the appropriate Origin Server.

   Note: A Surrogate may not communicate with the Origin directly and instead obtain the requested content from other surrogates or caching layers in the CDN hierarchy. The details of how content requests filter through the CDN hierarchy to the Origin are internal to a specific CDN and are out of scope of this document.

3. CDN & ALTO Use Cases

The primary use case for ALTO in a CDN context is to improve the selection of a CDN Surrogate or Origin. The CDN makes use of an ALTO server to choose a better CDN Surrogate or Origin than would otherwise be the case. In its simplest form an ALTO server would provide an NSP with the capability to offer a service to a CDN which provides network map and cost information that the CDN can use to enhance its surrogate and/or Origin selection.

Although it is possible to obtain raw network map and cost information in other ways, for example passively listening to the NSP's routing protocols, the use of an ALTO service to expose that information may provide additional control to the NSP over how their network map/cost is exposed. Additionally it may enable the NSP to maintain a functional separation between their routing plane and network map computation functions. This may be attractive for a number of reasons, for example:

- The ALTO service could provide a filtered view of the network and/or cost map that relates to CDN locations and their proximity to end users, for example to allow the NSP to control the level of topology detail they are willing to share with the CDN.
- The ALTO service could apply additional policies to the network map and cost information to provide a CDN-specific view of the network map/cost, for example to allow the NSP to encourage the CDN to use network links that would not ordinarily be preferred by

a Shortest Path First routing calculation.
- The routing plane may be operated and controlled by a different operational entity (even within a single NSP) to the CDN and the ALTO service could provide a layer of separation because:
  - The CDN is not able to passively listen to routing protocols.
  - The NSP is not willing to allow the CDN to passively listen to routing protocols, e.g. because the NSP is concerned the CDN may inadvertently interfere with the routing plane or because the routing plane and the CDN are operated by different operational entities/groups (including different entities within the same NSP).

The use cases in this document are not necessarily specific as to the relationship between the commercial/operational entity that "owns" the ALTO service and the commercial/operational entity that "owns" the CDN service as it is assumed that such relationships will be deployment specific. Although the ownership of each service may affect the level of topology detail that the ALTO service will be permitted to expose, it is assumed that the general requirements a CDN places on the ALTO service should not change provided that the ALTO server is able to expose sufficient topology for the CDN to make appropriate surrogate and/or Origin selection decisions.

In general, the ALTO service is expected to be operated by an entity or entities that wish to optimize or otherwise influence request routing decisions. Some, non-exhaustive, examples of such entities are:

- The entity that operates the CDN’s underlying network (e.g. the "CDN deployed within a Broadband network" described in Section 3.3).
- An NSP that wishes to optimize over-the-top content delivery from a CDN that is deployed outside of its network (e.g. the "CDN delivering Over-The-Top of a NSP" described in Section 3.4).
- An NSP (that may or may not operate a CDN) or a CDN that wishes to advertise which End Users are reachable via its network/CDN (e.g. the exposing "End User reachability" use cases described in Section 3.1 and Section 3.2).

The following sections outline some specific, non-exhaustive, example use cases, which are subsets of the primary use case outlined above but applied to specific usage examples to demonstrate how a CDN could make use of ALTO services.

3.1. Exposing NSP End User Reachability to a CDN

In order for a Request Router to be able to make surrogate selection decisions, the Request Router needs to have information on which End User IP subnets are reachable via which networks or network

locations. The granularity of location information required depends on the specific deployment of the CDN relative to the End Users. For example, an Over-The-Top CDN whose surrogates are deployed only within the Internet "backbone" may only require knowledge of which End User IP subnets are reachable via which NSPs' networks, whereas a CDN deployed within a particular NSP's network requires a finer granularity of knowledge, i.e. which End User IP subnets are reachable via which regions within that NSP's network.

Such reachability information is often available via dynamic routing protocols, however it is likely that in a number of deployment scenarios that peering of the routing plane of the network with a CDN would be deemed unacceptable (e.g. where the CDN is operated by an entity other than the NSP(s) operating the underlying network).

Provided that some common mapping between ALTO PIDs and network locations (or entire networks) is known to both the NSP and the CDN, the network map services offered by ALTO could be used to expose which End User IP subnets are reachable via a particular network or network locations in order to export End User reachability to a Request Router to enable the NSP to expose End User reachability while also giving the NSP the ability to control the granularity of any End User reachability to network location mapping while also avoiding routing plane peering between the NSP and the CDN.

3.2. Exposing CDN End User Reachability to CSPs

This use case is similar to the use case described in Section 3.1 however in this case it is the CDN that wishes to expose which End User IP subnets the CDN is capable of delivering services to.

In some deployments a particular CDN may not have reachability to (or may not wish to offer services to) every End User IP subnet reachable via the global Internet, for example because the CDN is only deployed within certain networks or geographic regions and the CDN is either unable (due to lack of reachability) or unwilling (due to cost or policy) to serve all End Users reachable via the global Internet.

The reachability offered by a particular CDN may not include all the End User IP subnets that a particular CSP requires in order to serve all of that CSP's customers and therefore if the CSP wishes to make use of the services offered by a CDN that can only serve a subset of their customers the CSP must have knowledge of which End User IP subnets a particulart CDN is able to serve, so that they can select an appropriate CDN to use to deliver their service to particular subsets of their customers.

In such cases, the network map services offered by ALTO could be used

to expose to a CSP which End User IP subnets are reachable via a particular CDN. In the case where the CDN is operated by an NSP using ALTO in this way could also enable the NSP to separate the exposure of End User subnets reachable via their CDN from those reachable via their underlying network.

3.3. CDN deployed within a Broadband network

In this use case an NSP is providing Broadband services to its customers and has deployed a CDN within its Broadband network to alleviate the cost and/or improve the User Experience of content services for its Broadband customers.

The topology of Broadband access/backhaul networks is often much more constrained than metro/core networks. If CDN surrogates are deployed within the access/backhaul network, for a given set of End Users, the NSP is likely to want to utilise the surrogates deployed in the same access/backhaul region as those End Users in preference to surrogates deployed within the metro/core or within other access/backhaul regions.

It is common for Broadband subscribers to obtain their IP addresses dynamically and in many deployments the IP subnets allocated to a particular access/backhaul region can change relatively frequently. For example new IP subnets are added as the subscriber base grows, IP subnets are moved from one Broadband product in the NSP's portfolio to another as customers migrate in order to optimise the NSP's IP address utilisation, or they are simply moved as part of IP address management, etc.

Additionally, in certain cases, CDN surrogates deployed in a particular network region may become overloaded, leading to the CDN selecting alternative surrogates in a different region of the network for content delivery. If this occurs, an NSP may wish to influence such a decision, for example because the NSP would prefer a surrogate to be selected that is deployed in the the next best (cost-wise to the NSP) location.

In order to meet the NSP's objective of utilising their CDN to constrain access/backhaul costs and/or improve User Experience it is important that the CDN is able to select the most appropriate surrogate for a given set of End User IP subnets. Although the network topology is often reasonably static, in networks where the IP subnets allocated to a Broadband region are changing relatively frequently, static configuration of End User IP Subnets to CDN surrogates is possible but some NSPs may consider the operational burden of having to update such static configuration too high and would prefer the CDN to be able to dynamically obtain network map and

cost information.

The NSP could make use of an ALTO service to expose a cost mapping/ ranking between End User IP subnets (within that NSP's network) and CDN surrogate IP subnets/locations to meet its requirements while avoiding static configuration or direct integration of the CDN into its IP routing plane and to avoid the CDN being required to implement network layer routing computations.

3.4. CDN delivering Over-The-Top of a NSP's network

In this use case a CDN is deployed within one or more NSPs' networks but is delivering content "Over-The-Top" into another NSP's network (which we will call NSP Z) where the CDN is not deployed.

The CDN is unlikely to have direct visibility of NSP Z's network topology and may have a choice of entry points into NSP Z's network from which it could serve content to NSP Z's End Users. For example because NSP Z has direct peering links with the CDN in a number of locations or NSP Z has transit and/or peering relationships with several other NSPs where the CDN is deployed. NSP Z may wish to influence the locations from which the CDN serves content based on some factor(s) that it does not wish to expose directly or that might change over time. For example the available transit/peering capacity in different locations, the cost of connectivity to different locations, etc.

For example, a CSP is using NSP A's CDN and another NSP (NSP Z) has peering with NSP A in Los Angeles and New York. NSP Z would like to influence which peering location NSP A's CDN delivers content out of for NSP Z's end users by using their knowledge of the peering capacity they have deployed in LA & NY and the capacity they have between those peering locations and groups of end users without directly exposing their internal topology to NSP A.

An NSP could make use of an ALTO service to expose a cost mapping/ ranking between End User IP subnets (within that NSP's network) and entry points into that NSP's network in order to try to influence the locations from which the CDN serves content into that NSP's network.

3.5. CDN acquiring content from multiple upstream sources (Origins)

Before a surrogate within a CDN is able to deliver content to an End User it must first have a copy of the content that the End User is requesting. Content may be obtained by surrogates in advance of it being requested (pre-positioned) by End Users or it may be obtained by surrogates dynamically in response to End User requests for the content (on-demand).

The ultimate source of the content (i.e. where the 'master' copy is permanently stored) is typically referred to as the content's Origin (or Origin Server), however CDNs often employ an internal hierarchy of caching layers so that surrogates do not necessarily obtain content directly from the Origin. Such a hierarchy provides a number of benefits, for example reducing the number of requests for content received by the Origin (and therefore reducing the scaling requirements on the Origin), more efficient use of the underlying network as fewer copies of the content is required to traverse the same network links, etc.

For a particular CSP's content service multiple, possibly independently addressable, Origins may be used for resiliency and the Origin(s) may be deployed in a distributed manner across multiple geographic locations.

For the rest of this use case "upstream source" is used to mean either the Origin itself as well as other sources of the content, for example another caching layer within the CDN that has (or will obtain on demand) a copy of the content but is not the actual Origin.

Therefore, for a particular item of content, a surrogate may have a choice of upstream sources (both internal to the CDN and external Origins) from which it could obtain the content.

When presented with a choice of upstream sources, a surrogate may utilise some combination of policy and heuristics to decide which upstream sources (and in which order) it should attempt to use to obtain the content. A CDN may wish to utilise network topology & cost information as one of the inputs into such a content source selection process, for example to weight upstream sources that are topologically close to the surrogate that requires the content.

Additionally, where the CDN is deployed within one or more NSP networks, an NSP may want to try to influence the choice of upstream sources, for example the NSP may prefer the CDN to use content sources that are deployed within that NSP's network or within networks with which it has direct peering agreements with over other content sources.

An NSP (or a CSP) could provide an ALTO service which a CDN could use to obtain network topology and/or cost/ranking information to use as an input into surrogates' selection decisions for content sources.

3.6. Additional Use Cases

The following additional use case may be relevant to ALTO and will be described in more detail in a future version of this document:

- Inter-provider CDN / CDN Interconnect.

4. IANA Considerations

This document makes no specific request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

5. Security Considerations

TBD.

6. Contributing Authors

Reinaldo Penno
Juniper Networks
Email: rpenno@juniper.net

Richard Alimi
Google
Email: ralimi@google.com

Richard Yang
Yale University
Email: ryr@yale.edu

7. Acknowledgements

The authors would like to thank Vijay Gurbani and Volker Hilt for their review comments and contributions to the text.

8. Normative References

[I-D.jenkins-cdni-problem-statement] Niven-Jenkins, B., Faucheur, F., and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", draft-jenkins-cdni-problem-statement-02 (work in progress), March 2011.

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

Authors' Addresses

Ben Niven-Jenkins (editor)
Velocix (Alcatel-Lucent)
326 Cambridge Science Park
Milton Road, Cambridge CB4 0WG
UK

Email: ben@velocix.com

Grant Watson
BT

Email: grant.watson@bt.com

Nabil Bitar
Verizon
40 Sylvan Road
Waltham, MA 02145
USA

Email: nabil.bitar@verizon.com

Jan Medved
Juniper Networks

Email: jmedved@juniper.net

Stefano Previdi
Cisco Systems

Email: sprevidi@cisco.com

ALTO
Internet-Draft
Intended status: Standards Track
Expires: September 15, 2011

S. Kiesel
University of Stuttgart
M. Stiemerling
NEC Europe Ltd.
N. Schwan
M. Scharf
Alcatel-Lucent Bell Labs
M. Tomsu
Alcatel-Lucent
H. Song
Huawei
March 14, 2011

ALTO Server Discovery Protocol
draft-kiesel-alto-3pdisc-05

Abstract

The goal of Application-Layer Traffic Optimization (ALTO) is to provide guidance to applications, which have to select one or several hosts from a set of candidates that are able to provide a desired resource.

Entities seeking guidance need to discover and possibly select an ALTO server to ask. This is called ALTO server discovery. This memo describes an ALTO server discovery mechanism based on several alternative mechanisms that are applicable in a diverse set of ALTO deployments.

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 September 15, 2011.

Copyright Notice

Table of Contents

1. Introduction

The goal of Application-Layer Traffic Optimization (ALTO) is to provide guidance to applications, which have to select one or several hosts from a set of candidates, that are able to provide a desired resource [RFC5693]. The requirements for ALTO are itemized in [I-D.ietf-alto-reqs]. ALTO is realized by a client-server protocol. ALTO clients send queries to ALTO servers, in order to solicit guidance.

ALTO clients have to discover suitable ALTO servers. Therefore the output of the herein defined ALTO discovery procedure tells the ALTO client which ALTO servers to send the queries to. The ALTO discovery procedure, as part of the the ALTO client, can be embedded in the resource consumer, which will eventually access the desired resource. As an alternative, they can be embedded in a resource directory, which assists resource consumers in finding appropriate resource providers. In some specific peer-to-peer application protocols these resource directories are called "trackers". Finally the ALTO server discovery procedure can be embedded in the resource consumer, whereas the ALTO client is embedded in the resource directory. ALTO queries, which are issued by a resource directory on behalf of a resource consumer, are referred to as third-party ALTO queries. The various possibilities to place ALTO servers and the placement of ALTO clients is discussed in [I-D.stiemerling-alto-deployments].

No matter where ALTO server and client are located, clients have to first find out if there is an ALTO server deployed that is in charge for them, and second they have to get the contact information of that server, i.e., the IP address, port number, and probably transport protocol (which defaults to TCP for [I-D.ietf-alto-protocol]).

The goal of this memo is to propose a uniform mechanism for all types of ALTO client deployments that is implementable and deployable at a fast pace, i.e., without creating other deployment dependecies for ALTO. We propose to use a combination of DHCP and DNS to retrieve the URL of the resposnsible ALTO server.

Comments and discussions about this memo should be directed to the ALTO working group: alto@ietf.org.

1.1. History

This document represents a merge of features from two previous drafts:

- draft-kiesel-alto-3pdisc-04

o draft-song-alto-server-discovery-03

1.2. Discovery Scenarios

Figure 1 below shows an overview on the different entities of a generic ALTO framework. The ALTO Server discovery mechanism is used by the p2p application in order retrieve the point of contact of the ALTO Service.

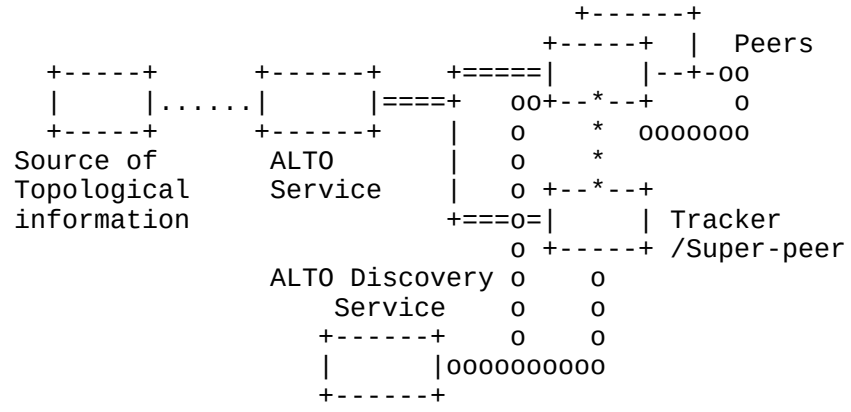


Figure 1: ALTO Discovery Overview

Hereby the ALTO service discovery scenarios are classified into two types: one is the ALTO server discovery by the resource consumer, and the other is the ALTO server discovery by a third party, such as application trackers. Before the specification of the discovery mechanism the following section illustrates and discusses both scenarios.

1.2.1. ALTO Server Discovery by Resource Consumers

The ALTO service discovery in some scenarios needs to be performed by the resource consumer itself. In particular in p2p applications without a tracker like DHTs and other conventional client/server applications.

In addition also p2p application which are tracker based may embed the ALTO client into the resurce consumer to allow peers a peer selection after retrieving peer list from the application tracker. Another option is that the reource consumer peer sends its ALTO server address information to the application tracker or any other third party entity, which in turn will contact the specific ALTO server and in order to retrieve ALTO guidance on behalf of the resource consumer.

The following figure illustrates this scenario, showing the relationship between the different entities as discussed before.

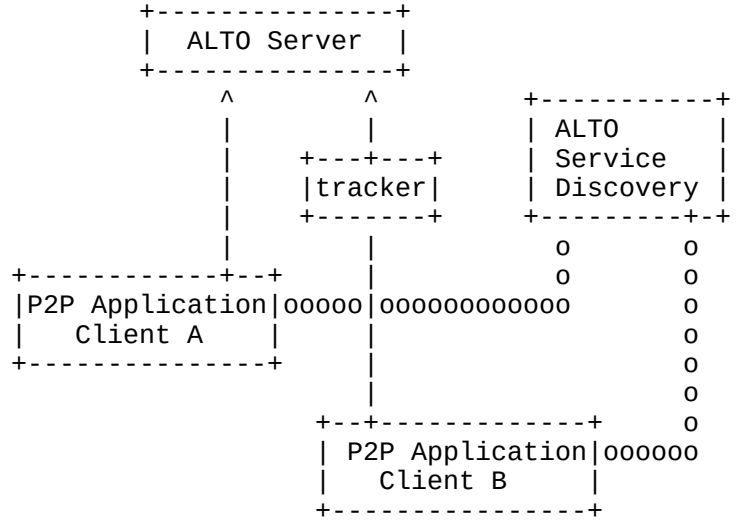


Figure 2: Resource Consumer ALTO Server Discovery (Example)

1.2.2. ALTO Server Discovery by a Third Party

Some p2p applications have trackers, and these applications might not need to have their clients looking for the ALTO server guidance. In these scenarios trackers query the ALTO servers for guidance themselves, and then return the final ranked result to the application clients. However, application clients are distributed among different network operators and autonomous systems. Trackers thus need to find different ALTO servers for the clients located in different operator networks or autonomous systems. In such scenarios the discuvery is thus not performed by the resource consumer, but a third party entity on behalf of the resource consumer.

Figure 3 shows an example for a third party ALTO server discovery. For client 1, the tracker has not cached yet the mapping between

client 1's network operator and its ALTO server address, so it queries the DNS server for the ALTO server address in that operator's domain. And then the tracker interacts with the ALTO server on behalf of client 1(to get the network map and cost map), finally, the ranked list is sent back to client 1. For client 2, the tracker has cached the mapping between client 2's network operator and its ALTO server address, so it does not need to query the DNS for the address of ALTO server 2. If the Application tracker already has the network map and cost map from ALTO Server2, then it does not to query the ALTO Server for network map and cost map frequently.

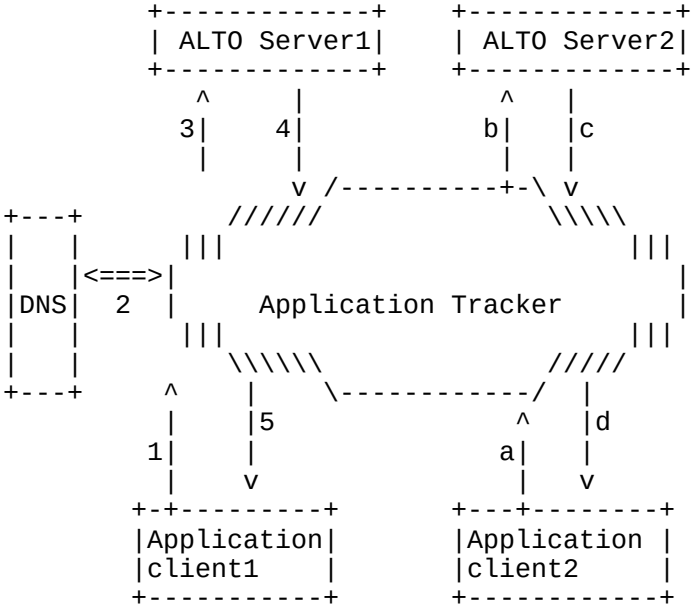


Figure 3: Third Party ALTO Server Discovery (Example)

1.3. Pre-Conditions

The whole document assumes certain pre-conditions, such as:

- The ALTO server discovery procedure is executed on a per IP address base. Multiple IP addresses per interface or multiple IP addresses assigned to different IP interfaces require to repeat the procedure for every IP address. It may be fine to group IP addresses according their domain suffixes and to perfom the procedure for such a group. However, this is out of scope of this document.

- The ALTO server discovery procedure is executed on a per IP family base, i.e., seperate for IPv4 and IPv6. It is up to the ALTO client to decide which of the possible multiple results of different IP address families to use. The choice of whether to use IPv4 or IPv6 is out of scope of this document.

- A change of the IP address at an interface invalidates the result of the ALTO server discovery procedure. For instance, if the IP address assigned to a mobile host changes due to host mobility, it is required to run the ALTO server discovery procedure for the new IP address without relying on earlier gained information.

2. Protocol Overview

We define multiple alternatives to discover the IP address of the ALTO server, as there are a number of ways possible how such information can be provided to the ALTO client. The choice of method is up to the local network deployment. For instance, there can be deployments where the ALTO server in charge for ALTO client is provisioned by the network operator and communicated to the ALTO client's host via a DHCP option, while in other deployments no such means may exist.

The following figure illustrates the different protocols that are used to find the URI of a suitable ALTO server.

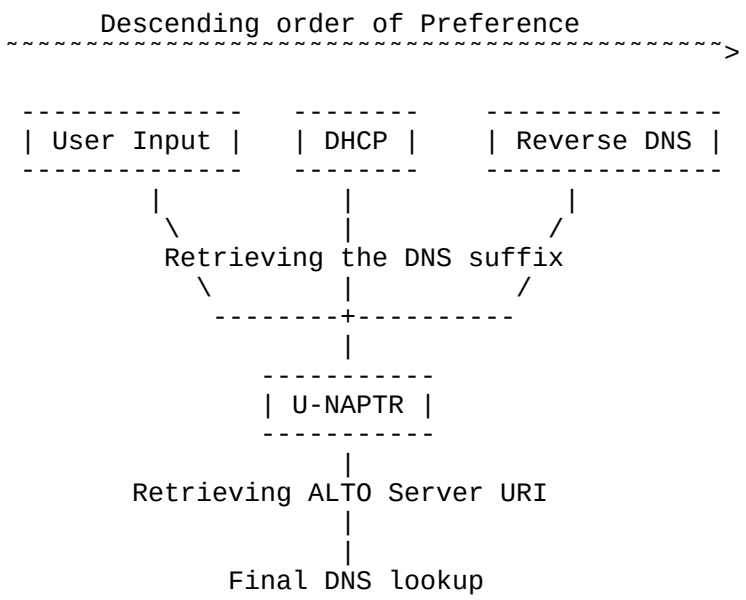


Figure 1: Protocol Overview

The figure above illustrates the U-NAPTR based resolution process to retrieve the ALTO Server URL. As a precondition for resolution the U-NAPTR process needs the right domain name as input. This domain name is determined by the IP address of the client and the DNS suffix of the access network where the client is registered in. In order to retrieve the DNS suffix we specify three options:

User input: a user may manually specify the DNS suffix on its own, either to access a 3rd party ALTO service provider or as it does know such information.

DHCP: a network provider provides the DNS suffix through a DHCP option.

Reverse DNS: the DNS system can be used to retrieve the DNS suffix through reverse lookup of an FQDN associated with an IP address. This is the last resort if all other options failed.

3. Retrieving the URI by U-NAPTR

This section specifies the U-NAPTR based resolution process. To start the U-NAPTR resolution process a domain name as input is needed. Thus the section is devided into two parts: Section 3.1 describes the U-NAPTR resolution process itself. How the client identifies this DNS suffix of the access network where the resource consumer is registered in is described in Section 3.2.

3.1. U-NAPTR Resolution

ALTO servers are identified by U-NAPTR/DDDS (URI-Enabled NAPTR/ Dynamic Delegation Discovery Service) [RFC4848] application unique strings, in the form of a DNS name. An example is 'altoserver.example.com'.

Clients need to use the U-NAPTR [RFC4848] specification described below to obtain a URI (indicating host and protocol) for the applicable ALTO service. In this document, only the HTTP and HTTPS URL schemes are defined. Note that the HTTP URL can be any valid HTTP URL, including those containing path elements.

The following two DNS entries show the U-NAPTR resolution for "example.com" to the HTTPS URL https://altoserver.example.com/secure or the HTTP URL http://altoserver.example.com, with the former being preferred.

```
example.com.

IN NAPTR 100  10   "u"    "ALTO:https"
     "!.*!https://altoserver.example.com/secure!"  ""

IN NAPTR 200  10   "u"    "ALTO:http"
     "!.*!http://altoserver.example.com!"  ""
```

3.2. Retrieving the Domain Name

The U-NAPTR resolution process requires a domain name as input. The algorithm that is applied to determine this domain name is described in this section. We specify three different options. In option 1 the user manually configures a specific ALTO service instance that he wants to use. Option 2 defines a DHCP option to allow the network service provider a remote configuration of the client. In option 3 the client tries to get the domain name by performing a reverse DNS lookup on its IP address.

The resource consumer may have private IP addresses and public IP

addresses and depending on the deployment it might be necessary to determine for all IP addresses the ALTO server in charge of. To determine its public IP address the resource consumer may need to use STUN[RFC5389] or BEP24[bep24]. For the following examples we assume that the IP address of the resource consumer is a.b.c.d.

3.2.1. Option 1: User input

A user may want to use a third party ALTO service instance. Therefore we allow the user to specify a DNS suffix on its own, for example in a config file option. The DNS suffix given by the user is combined with the IP address of the resource consumer to allow the third party ALTO service to direct the client to a suitable ALTO server based on the location of the client. A possible DNS suffix entered by the user may be:

myaltoprovider.org

This DNS suffix is prepended with the IP address of the resource consumer in reverse order to compose the domain name used for the final U-NAPTR lookup Section 3.1. In case there are multiple ALTO servers deployed, the third party ALTO service instance can direct the ALTO client to the ALTO server closest to the client based on the IP address.

Multiple lookups with different domain names might be necessary to complete the U-NAPTR resolution process. If there is no response for a lookup the domain name is shortened by one part for the succeeding lookup, until a lookup is successful, as for example

d.c.b.a.myaltoprovider.org.

c.b.a.myaltoprovider.org.

b.a.myaltoprovider.org.

a.myaltoprovider.org.

myaltoprovider.org.

3.2.2. Option 2: DHCP

As a second option network operators can configure the domain name to be used for service discovery within an access network. RFC 5986[RFC5986] defines DHCP IPv4 and IPv6 access network domain name options that identify a domain name that is suitable for service discovery within the access network. The ALTO server discovery procedure uses these DHCP options to retrieve the domain name as an

input for the U-NAPTR resolution. One example could be:

example.com

3.2.3. Option 3: Reverse DNS Lookup

The last option to get the domain name is to use a DNS PTR query for the IP address of the resource consumer. The local DNS server resolves the IP address to the FQDN that also contains the DNS suffix for the respective IP address. A possible answer for a PTR lookup for d.c.b.a.in-addr.apra might be, for example:

d-c-b-a.dsl.westcoast.myisp.net

This domain name can be used for the final U-NAPTR lookup Section 3.1. If there is no response to the lookup the domain name is shortened by one part for one succeeding lookup. If there is still no response we consider the reverse lookup being failed. The domain names used for the example as described above are:

d-c-b-a.dsl.westcoast.myisp.net.

dsl.westcoast.myisp.net.

4. Applicability

This section discusses the applicability of the proposed solution with respect to the resource consumer server discovery and the third party deployment scenarios. Each section discusses the proposed steps that are needed to determine the ALTO Server URI.

4.1. Applicability for Resource Consumer Server Discovery

In this scenario the ALTO server discovery procedure is performed by the resource consumer, for example a peer in a P2P system. After the discovery the peer does the ALTO query on its own, or it might share the ALTO server contact information with a third party, for example a tracker, which then does the ALTO query on behalf of the peer.

To complete the ALTO server discovery process the resource consumer first SHOULD check whether the user has provider the domain name through manual configuration. If this is not the case the next step SHOULD be to check for the access network domain name DHCP option (Section 3.2.2). Finally the client SHOULD try to retrieve the domain name by the last option, the DNS reverse lookup on its IP address as described in Section 3.2.3.

In case the ALTO discovery client has determined the domain name through one of the described options it proceedes with the U-NAPTR lookup as described in Section 3.1.

4.2. Applicability for Third Party Server Discovery

In case of the third party server discovery deployment scenario the entity performing the ALTO server discovery process is different from the resource consumer. Typically the resource consumer is a peer whereas the ALTO client is a resource directory which seeks for ALTO guidance on behalf of the peer. Another use case for the third party discovery is an application that looks for ALTO guidance transparently for the resource consumer, for example a CDN.

Here the ALTO server discovery process can also retrieve guidance through the DHCP option or manual user configuration, but only if the provided discovery information is forwarded by the resource consumer to the third party entity. In this case, additional mechanisms for the forwarding of this discovery information need to be specified. However these mechanisms are out of scope of this doument.

If the third party entity cannot obtain this discovery information, the ALTO server discovery process relies on retrieving the domain name used as input to the U-NAPTR lookup through reverse DNS lookup of the IP address of the resource consumer as described in

Section 3.2.3. Usually the third party entity already knows the IP address of the resource consumer which was used to establish the initial connection. In general this IP address is a public address, either of the resource consumer or of the last NAT on the path to the ALTO client. This makes the IP address a good candidate for the DNS PTR query. Thus, we expect that the DNS query will be successfully resolved to the FQDN of the domain where the resource consumer is registered in.

In case the resource consumer needs guidance for a different IP address, for example one from a private network, we recommend that the resource consumer discovers the server itself and forwards the ALTO server contact information directly to the third party entity, which in turn can then do the third party ALTO query. Again, forwarding the contact information from the resource consumer to the third party entity is out of scope of this document.

5. Deployment Considerations

The mechanism specified in this document needs some configuration effort in order to work properly. Especially the domain name retrieved through the reverse DNS lookup (PTR records) and the U-NAPTR entry need to be coordinated. In this section we discuss this configuration for different scenarios.

5.1. Private customers or very small businesses

For private customers and very small businesses that are DSL or cable customers often a dynamically assigned IP address is provisioned. Here, the reverse DNS lookup (PTR records) are controlled by the ISP and they point to the ISP's domain, e.g.:

p5B203EA1.dip.t-dialin.net.

dslb-084-056-144-100.pools.arcor-ip.net.

187-4-222-157.bnut3700.dsl.brasiltelecom.net.br.

65-154-39-69.ispnetbilling.com.

197-151-94-178.pool.ukrtel.net.

In this case, it would be the responsibility of the respective ISP to provide U-NAPTR entries for the DNS suffix without the endhost part, e.g.:

dip.t-dialin.net.

pools.arcor-ip.net.

bnut3700.dsl.brasiltelecom.net.br.

ispnetbilling.com.

pool.ukrtel.net.

5.2. Medium-size customer networks

The second class of customers have their own DNS domain but only one single upstream ISP, e.g.:

(1) ISP my-isp.net assigns an IP address a.b.c.d to its customer

(2) The customer decides that reverse mapping for a.b.c.d should be whatever.customerdomain.com

(3) If the customer wants to support ALTO, he has to ask the ISP for the URI of the ISP's ALTO server which can give guidance to a.b.c.d. Assume that ISP replies it is http://altoserver.my-isp.net

(4) The customer establishes a U-NAPTR entry for his domain

```
customerdomain.com.   IN NAPTR 200  10   "u"    "ALTO:http"
     "!.*!http://altoserver.my-isp.net!"  ""
```

5.3. Large Customers

For very large customers with multiple upstream connections we assume that they have their very own traffic optimization policies and thus run their own ALTO server anyway. In this case they need to manage their DNS entries accordingly.

6. IANA Considerations

This document registers the following U-NAPTR application service tag:

Application Service Tag: ALTO

Defining Publication: The specification contained within this document.

This document registers the following U-NAPTR application protocol tags:

- Application Protocol Tag: http

  Defining Publication: RFC 2616 [RFC2616]

- Application Protocol Tag: https

  Defining Publication: RFC 2818 [RFC2818]

7. Security Considerations

7.1. General

This is still to be done in later revision of this draft, as the draft evolves heavily right now.

7.2. For U-NAPTR

The address of an ALTO server is usually well-known within an access network; therefore, interception of messages does not introduce any specific concerns.

The primary attack against the methods described in this document is one that would lead to impersonation of a ALTO server since a device does not necessarily have a prior relationship with a ALTO server.

An attacker could attempt to compromise ALTO discovery at any of three stages:

1. providing a falsified domain name to be used as input to U-NAPTR

2. altering the DNS records used in U-NAPTR resolution

3. impersonation of the ALTO

This document focuses on the U-NAPTR resolution process and hence this section discusses the security considerations related to the DNS handling. The security aspects of obtaining the domain name that is used for input to the U-NAPTR process is described in respective documents, such as [I-D.ietf-geopriv-lis-discovery].

The domain name that is used to authenticated the ALTO server is the domain name in the URI that is the result of the U-NAPTR resolution. Therefore, if an attacker were able to modify or spoof any of the DNS records used in the DDDS resolution, this URI could be replaced by an invalid URI. The application of DNS security (DNSSEC) [RFC4033] provides a means to limit attacks that rely on modification of the DNS records used in U-NAPTR resolution. Security considerations specific to U-NAPTR are described in more detail in [RFC4848].

An "https:" URI is authenticated using the method described in Section 3.1 of [RFC2818]. The domain name used for this authentication is the domain name in the URI resulting from U-NAPTR resolution, not the input domain name as in [RFC3958]. Using the domain name in the URI is more compatible with existing HTTP client software, which authenticate servers based on the domain name in the URI.

An ALTO server that is identified by an "http:" URI cannot be authenticated. If an "http:" URI is the product of the ALTO discovery, this leaves devices vulnerable to several attacks. Lower layer protections, such as layer 2 traffic separation might be used to provide some guarantees.

8. Open Issues

Here are a few open issues to be clarified:

Handling of reverse DNS lookups for IPv6: Refer to [RFC4472] for a discussion about the issues.

Missing reverse DNS entries for an IP address: There may be cases where the reverse DNS lookup does not yield any result. However, this will leave the ALTO client with no choice, other than giving up. This needs better documentation.

How to handled multiple results: For instance, a host behind a NAT that yields an ALTO server in the private IP address domain and one in the public IP address domain. Whom to ask?

Suffix Issues Document issues with suffix information provided by DHCP or by other means. For instance, a host behind a NAT may have a configured DNS suffix ".local". This suffix is not usuable for the server discovery procedure.

9. Contributors

Hannes Tschofenig provided the initial input to the U-NAPTR solution part. Hannes and Martin Thomson provided excellent feedback and input to the server discovery.

The authors would also like to thank the following persons for their contribution to this document or predecessors:

Roni Even

Gustavo Garcia

Yu-Shun Wang

Victor Pascual

Richard Alimi

Yunfei Zhang

Y. Richard Yang

Xingfeng Jiang

Jay Gu

Ning Zong

David Bryan

Enrico Marocco

10. Conclusion

This document describes a general ALTO server discovery process and discusses how the process can be applied in different deployment scenarios, including the resouce consumer discovery as well as the third party discovery.

11. References

11.1. Normative References

[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.

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.

[RFC3958] Daigle, L. and A. Newton, "Domain-Based Application Service Location Using SRV RRs and the Dynamic Delegation Discovery Service (DDDS)", RFC 3958, January 2005.

[RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

[RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.

11.2. Informative References

[I-D.ietf-alto-protocol] Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-04 (work in progress), May 2010.

[I-D.ietf-alto-reqs] Kiesel, S., Previdi, S., Stiemerling, M., Woundy, R., and Y. Yang, "Application-Layer Traffic Optimization (ALTO) Requirements", draft-ietf-alto-reqs-08 (work in progress), March 2011.

[I-D.ietf-geopriv-lis-discovery] Thomson, M. and J. Winterbottom, "Discovering the Local Location Information Server (LIS)", draft-ietf-geopriv-lis-discovery-15 (work in progress), March 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.

[I-D.stiemerling-alto-deployments] Stiemerling, M. and S. Kiesel, "ALTO Deployment

Considerations", draft-stiemerling-alto-deployments-03 (work in progress), June 2010.

[RFC4472] Durand, A., Ihren, J., and P. Savola, "Operational Considerations and Issues with IPv6 DNS", RFC 4472, April 2006.

[RFC4848] Daigle, L., "Domain-Based Application Service Location Using URIs and the Dynamic Delegation Discovery Service (DDDS)", RFC 4848, April 2007.

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

[RFC5986] Thomson, M. and J. Winterbottom, "Discovering the Local Location Information Server (LIS)", RFC 5986, September 2010.

[bep24] Harrison, D., "Tracker Returns External IP", BEP http://bittorrent.org/beps/bep_0024.html.

Appendix A. Acknowledgments

The authors would like to thank Haibin Song, Richard Alimi, and Roni Even for fruitful discussions during the 75th IETF meeting.

Hannes Tschofenig provided the initial input to the U-NAPTR solution part. Hannes and Martin Thomson provided excellent feedback and input to the server discovery.

Marco Tomsu and Nico Schwan are partially supported by the ENVISION project (http://www.envision-project.org), a research project supported by the European Commission under its 7th Framework Program (contract no. 248565). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ENVISION project or the European Commission.

Michael Scharf is supported by the German-Lab project (http://www.german-lab.de) funded by the German Federal Ministry of Education and Research (BMBF).

Martin Stiemerling is partially supported by the COAST project (COntent Aware Searching, retrieval and sTreaming, http://www.coast-fp7.eu), a research project supported by the European Commission under its 7th Framework Program (contract no. 248036). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the COAST project or the European Commission.

Authors' Addresses

Sebastian Kiesel
University of Stuttgart Computing Center
Allmandring 30
Stuttgart 70550
Germany

Email: ietf-alto@skiesel.de
URI: http://www.rus.uni-stuttgart.de/nks/

Martin Stiemerling
NEC Laboratories Europe
Kurfuerstenanlage 36
Heidelberg 69115
Germany

Phone: +49 6221 4342 113
Email: martin.stiemerling@neclab.eu
URI: http://ietf.stiemerling.org

Nico Schwan
Alcatel-Lucent Bell Labs
Lorenzstrasse 10
Stuttgart 70435
Germany

Email: nico.schwan@alcatel-lucent.com
URI: www.alcatel-lucent.com/bell-labs

Michael Scharf
Alcatel-Lucent Bell Labs
Lorenzstrasse 10
Stuttgart 70435
Germany

Email: michael.scharf@alcatel-lucent.com
URI: www.alcatel-lucent.com/bell-labs

Marco Tomsu
Alcatel-Lucent
Lorenzstrasse 10
Stuttgart 70435
Germany

Email: marco.tomsu@alcatel-lucent.com
URI: www.alcatel-lucent.com/bell-labs

Haibin Song
Huawei

Email: melodysong@huawei.com

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: September 15, 2011

R. Penno
J. Medved
Juniper Networks
R. Alimi
Google
R. Yang
Yale University
S. Previdi
Cisco Systems

March 14, 2011

ALTO and Content Delivery Networks
draft-penno-alto-cdn-03

Abstract

Networking applications can request through the ALTO protocol information about the underlying network topology from the ISP or Content Provider (henceforth referred as Provider) point of view. In other words, information about what a Provider prefers in terms of traffic optimization -- and a way to distribute it. The ALTO Service provides information such as preferences of network resources with the goal of modifying network resource consumption patterns while maintaining or improving application performance.

One of the main use cases of the ALTO Service is its integration with Content Delivery Networks (CDN). The purpose of this draft is twofold: first, to describe how ALTO can be used in existing and new CDNs, both within an ISP and in separate organizational entities from the ISP; second, to collect requirements for ALTO usage in CDNs and to provide recommendations into the development of the ALTO protocol for better support of CDNs.

Requirements Language

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 RFC 2119 [RFC2119].

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-

Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on September 15, 2011.

Copyright Notice

Table of Contents

1. Introduction

Content Delivery Networks are becoming increasingly important in the Internet [ARBOR] and many CDNs today already use some form of proximity such as latency-based proximity [GoogleCDN]. But in many cases the content provider/distributor and the Internet Service Provider (ISP) are disjoint entities. Consequently, even if content servers are co-located into the ISP's networks, there is not a standardized way to share server location and/or network topology information. Therefore a natural step forward would be to use ALTO to share this information.

Another key aspect of ALTO in the context of CDNs deployments is that it is desirable that no changes to the hosts are needed (or that changes to hosts would be transparent to the user). In other words, a traditional web browser using standard HTTP flow is all there is needed to take advantage of ALTO information. This is a significant difference from the P2P applications where a special client is typically needed and ALTO is normally used as a way to reduce operational expense.

2. Scope

This document discusses how Content Delivery Networks can benefit from ALTO through integration of the ALTO Service with the main request routing techniques. There are two objectives:

- Present basic integration schemes of ALTO into CDNs.
- Provide protocol recommendations to ALTO: Whenever a new requirement on protocol functionality is identified to achieve integration with CDNs, it will be enumerated with 'REQ-<N>'. Each requirement is documented in a section of its own in order to foster parallel discussions and possible adoption.

3. Terminology

We use the following terms defined in ALTO Problem Statement [RFC5693]: Application, ALTO Service, ALTO Server, ALTO Client, ALTO Query, ALTO Reply, ALTO Transaction.

In addition to the above, the following terms are defined:

Content-aware Proximity Request Routing Function: The Request Routing function knows about locations and presence of content & media objects in the network. Therefore the redirection to a CDN

node is made based on both the availability of content and/or content-type in that CDN node and the proximity of the CDN node to the requesting user.

Service-aware Proximity Request Routing Function: The Request Routing function knows about locations of CDN nodes in the network and redirects user to the closest CDN node. A redirection is made irrespective of content presence in the CDN node; if content is not present, the node will be populated with the content while the content is being served to the user.

HTTP Request Routing Function: a Content-aware or Service-aware Proximity Request Routing function for HTTP. It embeds an HTTP Server that performs HTTP Redirects, an ALTO client that retrieves network mapping from the ALTO Server, and a Location Database which stores network mappings received from the ALTO Client. The HTTP Server consults the Location Database when making redirection decisions.

4. Request Routing as an Integration Point of ALTO into CDN

Content Distribution is a rich and evolving field. New architectures and approaches (e.g., a hybrid architecture using both servers and P2P) continue to be developed in the research community and industry. Several CDN architectures are being deployed in production. While we would like to provide a survey of each possible CDN architecture and show how it may be integrated with ALTO, it would be a daunting task to track such a rapidly-changing field.

One scheme that is out of the scope of this document is P2P-only CDNs, where the application tracker takes the role of the ALTO Client, fetching the Network and Cost Maps from the ALTO Server and integrating them with its peer database. The result is a peer database that takes into account both the current peer metrics, such as peer availability or content availability, and network metrics, such as topological localization. This architecture, in the context of file sharing, has been studied extensively and trialed by ISPs such as Comcast [RFC5632] and China Telecom [I-D.lee-alto-chinatelecom-trial] under the ALTO/P4P [P4P] protocol. Thus, P2P-only CDNs are not discussed in this document.

The Request Routing Component of a CDN directs a request to a serving CDN node, and thus is the major integration point to utilize information available through ALTO. Today, multiple request routing schemes have been used even in CDNs with purely server-based infrastructure. The specific schemes include HTTP Redirect, DNS name resolution, and anycast. We focus on HTTP Redirect and DNS name

resolution.

Though anycast is a request routing technique that has been used in deployed CDNs, we do not discuss it in this document. Even though one may be able to integrate ALTO with anycast, we do not believe that this is a proper use of ALTO's capabilities. In particular, ALTO has been developed to improve selection amongst multiple content providers at the application level. In contrast, anycast operates by adjusting the routing layer to match content consumers with the desired content providers. Applying ALTO to routing layer decisions introduces additional complexity because it directly adjusts the routing layer from which the ALTO information is typically generated, creating a tight feedback loop. We leave a more detailed study of integrating ALTO with anycast-based CDNs as future work.

We next briefly review the two mechanisms presented in this document, HTTP Redirect and DNS Request Routing.

4.1. HTTP Redirect

In this mechanism, an HTTP GET request from a host is received by an HTTP Request Routing Function which sends back an HTTP response with Status-Code 302 (Redirect) informing the host of the most preferred location to fetch the content. The HTTP Redirection method is already commonly used in production CDNs as described in RFC3568 [RFC3568]. ALTO integration provides localization services where the device that performs the redirection becomes an ALTO client.

4.2. DNS Request Routing

In this mechanism, the DNS server handling host requests provides the Request Routing Component. When the host performs a DNS query/ lookup, the IP address(es) in the DNS response will indicate the selected location to serve the request.

DNS queries can be either iterative or recursive. Iterative queries can be used with ALTO if the host itself queries the DNS Servers, or if the DNS Proxy used by the host is topologically close to the host. If the Host directly queries the DNS Servers, the authoritative DNS Server can see directly the host's IP address. If the DNS Proxy is topologically close to the Host, its IP address is a good approximation for the host's location. In recursive queries, the authoritative DNS Server sees the IP address of the previous DNS Server in the resolution chain, and the IP address of the host is unknown. DNS-based request routing does not work well with recursive DNS queries.

In an iterative DNS lookup with a DNS Proxy (say for cdn.com), the

host queries the Proxy, which in turn first queries one of the root servers to find the server authoritative for the top-level domain (com in our example). The Proxy then queries the obtained top-level-domain DNS server for the address of the DNS server authoritative for the CDN domain. Finally, the Proxy queries the DNS server that is authoritative for the cdn.com domain. The authoritative DNS Server for cdn.com will perform the request routing to the most appropriate CDN node, based on the source IP address of the requestor. The host will then request the content directly from the CDN Node.

Recently, an EDNS0 option in DNS query has been proposed in [I-D.vandergaast-edns-client-subnet] that will provide a mechanism to carry sufficient network information about the client for the authoritative DNS server to tailor DNS response based on the client's subnet. Using this mechanism, the authoritative DNS server can achieve the same request routing accuracy as that of the HTTP Request Routing Function, and both recursive and iterative queuries can be supported.

5. Basic Scheme of CDN/ALTO Integration

Although HTTP Redirect and DNS are quite different mechanisms to direct a request to a serving CDN node, as we will see, the basic structure of integrating ALTO with them can be quite similar. Thus, we first present common structures. We refer to the HTTP Redirect component or the DNS component of a CDN as a CDN Request Routing Function.

5.1. Basic Integration Scheme

Figure 1 shows a general structure to embed an ALTO Client into a CDN Request Routing Function.

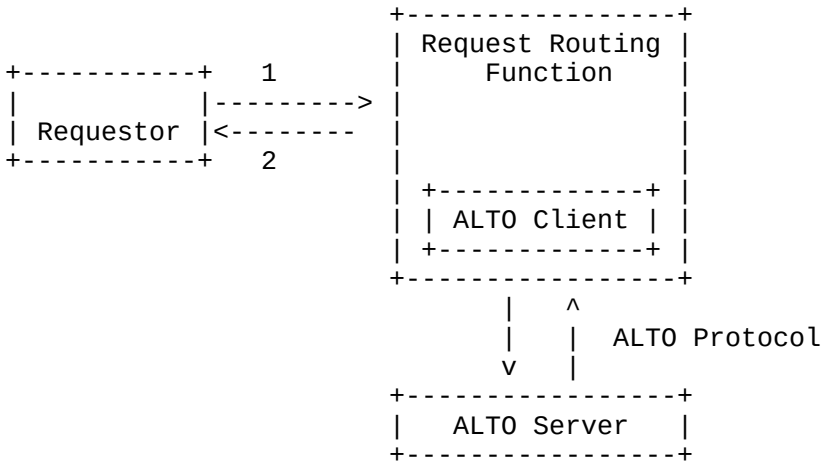


Figure 1: Request Routing Function with ALTO

An ALTO Server may aggregate information from multiple sources, such as routing protocols, traffic engineering policies, and monitoring systems. Thus, ALTO is complementary to existing infrastructure. For further detail, see Figure 1 of [I-D.ietf-alto-protocol].

5.1.1. ALTO for HTTP Redirect

To make the basic scheme more concrete, Figure 2 shows the case that the Request Routing Function is HTTP Redirect.

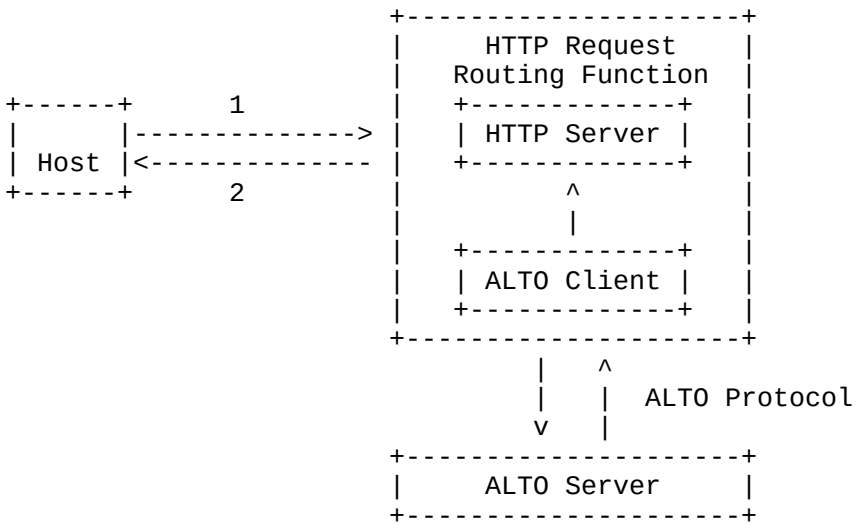


Figure 2: ALTO for HTTP Request Routing Function

5.1.2. ALTO for DNS Resolution

Figure 3 shows the case that the Request Routing Function uses DNS Resolution.

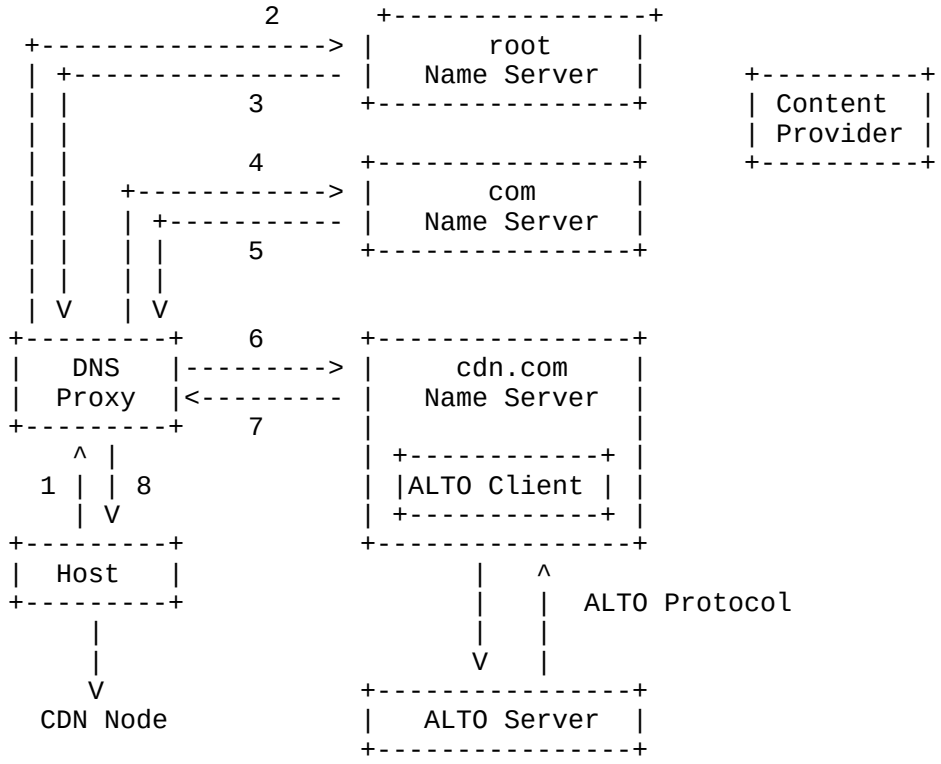


Figure 3: ALTO for DNS Resolution.

5.2. Multi-hop Redirection

The preceding examples show the logical flow for redirection. It is important to state that there maybe multiple redirection hops.

For HTTP Redirect, the requestor may be redirected again by the first CDN node. For DNS, the first DNS server may direct, using aggregated ALTO information (e.g., from multiple ALTO Servers of multiple ISPs), the DNS resolution to a second level DNS server, which then may use more specific ALTO information as well as CDN node status.

6. Request Routing using ALTO Services

Either the Map Service or the Endpoint Cost Service of ALTO can be used by the Request Routing Function. We first discuss two common issues: how to configure ALTO topology at ALTO servers; and how to achieve CDN node discovery and status notification. Then we give specific details on using the Map Service or the Endpoint Cost Service.

6.1. ALTO Topology vs. Network Topology

To answer queries from CDN Request Routing Functions, the ALTO server builds a ALTO-specific network topology that represents the network as it should be understood and utilized by the application layer (the CDN). Besides the security requirements that consist of not delivering any confidential or critical information about the infrastructure, there are efficiency requirements in terms of what visibility of the network, and at which level of granularity, is required by the CDN and more in general by the application layer.

The ALTO server builds topology (for either Map and ECS services) based on multiple sources that may include routing protocols, network policies, state and performance information, geo-location, etc. In all cases, the ALTO topology will not contain any details that would endanger the network integrity and security (for example, there will be no leaking of OSPF/ISIS/BGP databases to ALTO clients).

6.2. CDN Node Discovery and Status Notification

A design issue of integrating ALTO into Request Routing is how CDN Request Routing discovers the available CDN nodes and their locations. The exact mechanism is outside the scope of this document.

It is desirable that not only CDN node locations, but also real-time CDN node status (like health, load, cache utilization, CPU, etc.) is communicated to the Request Routing Function.

Specifically, CDN node status can be retrieved from the existing Load Balancer infrastructure. Most Load Balancers today have mechanisms to poll caches/servers via ping, HTTP Get, traceroute, etc. Most LBs have SNMP trap capabilities to let other devices know about these thresholds. Specification of a particular mechanism or API used to fetch load status information into an ALTO Server is out of scope of this document.

Note that in addition to the CDN node status, network status can also be retrieved from TE/RP databases. The Request Routing Function may

also need to be configured with a proper set of policies and business rules that control routing of requests. For example, it may be desirable to set up a rule that within a CDN certain requests have higher priority.

We see two approaches that CDN node status can be communicated to the Request Routing Function.

6.2.1. CDN Node Status Updates received by Request Routing Function

In the first approach, the Request Routing Function receives CDN Status updates directly.

For example, the Request Routing Function can implement an SNMP agent and get to know whatever is needed.

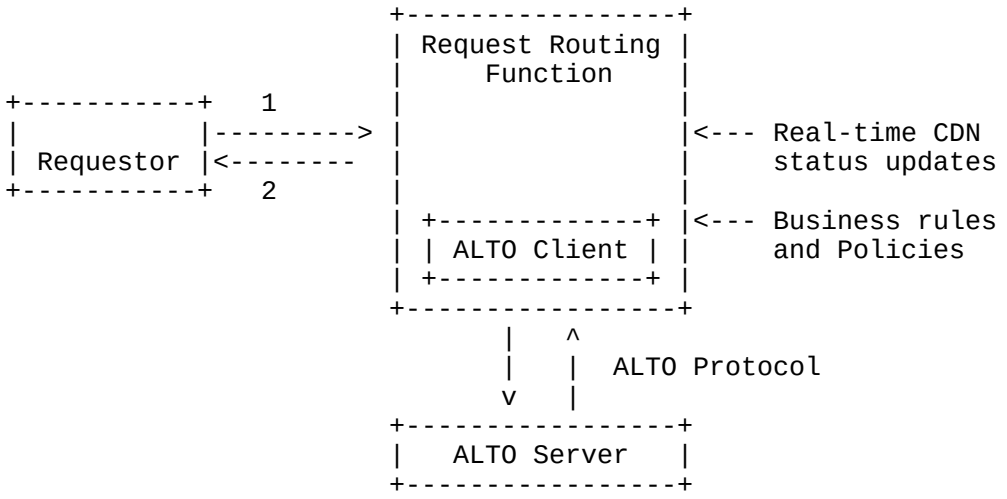


Figure 4: CDN Node Status to Request Routing Function

6.2.2. CDN Node Status Updates received by ALTO

In the second approach, the Request Routing Function receives CDN Status from ALTO instead of CDN nodes.

This model generally simplifies the Request Routing Function. It allows an easier distribution of the Request Routing Function, and to keep real time CDN status data updates in a logically centralized ALTO Server or in an ALTO Server Cluster. It allows for the Request Routing Function and the ALTO Server to be in different administrative domains. For example, the Request Routing Function can be in a Content Provider’s domain; the ALTO Server and CDN Nodes

in a Network Service Provider's domain.

Specifically, ALTO Server could provide an API (for example, a Web Service or XMPP-based API) that could be used by CDN nodes to communicate their status to the ALTO server directly.

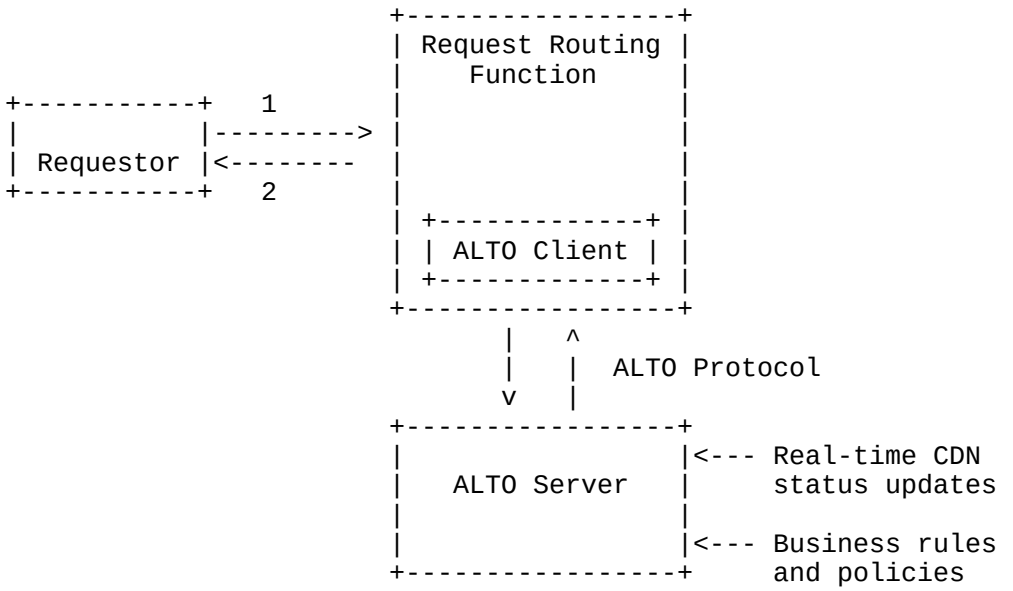


Figure 5: CDN Node Status to ALTO

6.3. Request Routing using the Map Service

The ALTO client embedded in the Request Routing Function fetches the Network and Cost Maps from the ALTO Server and provides that information to the Request Router.

As an illustrative example, we consider the case of HTTP Redirect. A simple Request Router may be given (from an external source) the list of available CDN nodes. The Request Router precomputes a redirection table indexed by source PID with values being the closest CDN nodes. This redirection table can be built based on information from Network and Cost Maps. Then when the Request Router receives an HTTP GET request, it looks up the PID of the source IP address on the request, indexes the redirection table using the request PID to select a CDN node, and finally returns a response that is an HTTP redirect with the URL of the selected CDN node. The URL in 302 Redirect may contain the IP address of the selected CDN node or a domain name instead of IP address due to virtual hosting. Therefore the IP addresses contained in the cost maps may need to be correlated to

domain names a priori. In practice, the redirection table may be indexed by both source and content to provide better redirection.

The illustrative example can also be extended to DNS.

The Network Maps generated by the ALTO Server will contain both Host PIDs and CDN Node PIDs, i.e., Host PIDs contain host subnets; CDN PIDs contain IP addresses of available CDN nodes. Cost Maps may contain only cost from each host PID to each CDN PID and not the full matrix across all PIDs. The reason is that the Request Router may redirect a host only to a CDN node, not to another host as in the P2P case. Moreover, there is no generic way to disambiguate PIDs containing only hosts from PIDs containing CDN nodes.

It is possible that a Request Router may be designated as being responsible only for a fixed set of Host PIDs. This information can be made available to the Request Router before it receives requests from hosts. If the set of Host PIDs is not known ahead of time, the latency for serving requests will be impacted by the capabilities of the ALTO server.

With such information ahead of time, a Request Router that uses the Network Maps Service may pre-download the Network Map for the interesting Host PIDs and the CDN PIDs. It can also start periodically pulling Cost Map for relevant PID 2-tuples.

The Request Router can rely on the ALTO Server generated Cache-Control headers to decide how often to fetch CDN PID network map and Host PID network maps.

For Alto protocol requirements related to request routing with the Map Service see Section 8.1.1 and Section 8.1.2.

6.4. Request Routing using the Endpoint Cost Service

Alternatively, the Request Router may request the Endpoint service from the ALTO client.

Specifically, the Request Router requests the Endpoint Cost Service to rank/rate the content locations (i.e., IP addresses of CDN nodes) based on their distance/cost (by default the Endpoint Cost Service operates based on Routing Distance) from/to the user address.

Once the Request Router obtains from the ALTO Server the ranked list of locations (for the specific user) it can incorporate this information into its selection mechanisms in order to point the user to the most appropriate location.

A Request Router that uses the Endpoint Cost Service may query the ALTO Server for rankings of CDN Node IP addresses for each requesting host and cache the results for later usage.

Maps Services and ECS deliver similar ALTO service by allowing the Request Routing Functino to optimize internal selection mechanisms. Both services deliver similar level of security, confidentiality of layer-specific information (i.e.: application and network) however, Maps and ECS differ in the way the ALTO service is delivered and address a different set of requirements in terms of topology information and network operations.

6.4.1. Topology Computation and ECS Delivery

ECS allows the Request Routing Function to not have to implement any specific algorithm or mechanism in order to retrieve, maintain and process network topology information (of any kind). The complexity of the network topology (computation, maintenance and distribution) is kept in the ALTO server and ECS is delivered on demand. Thus ECS is used in order to implement a lightweight integration of ALTO services in the CDN layer. ECS implies an ALTO and CDN implementation with the necessary scalability in order to cope with the amount of transactions that CDN and ALTO server will have to handle (knowing that the CDN is able to cache ALTO ECS results for further use).

6.4.2. Ranking Service

When a user requests a given content, the Request Routing Function locates the content in one or more caches and executes a selection algorithm to redirect the user to the ’best’ cache. In order to achieve that, the CDN issues an ECS request with the endpoint address (IPv4/IPv6) of the user (content requester) and the set of endpoint addresses of the content caches (content targets). The ALTO server, receives the request and ranks the list of content targets addresses based on their distance from the content requester. By default, according to [I-D.ietf-alto-protocol], the distance represents the routing cost as computed by the routing layer (OSPF, ISIS, BGP) and may take into consideration other routing criteria such as MPLS-VPN (MP-BGP) and MPLS-TE (RSVP), policy and state & performance information.

Once the ALTO server has computed the distance it replies with the ranked list of content target addresses. The list being ranked by distance, the CDN is capable of integrating the rankings into its selection process (that will also incorporate other criteria) and redirect the user accordingly.

6.5. Update, Redirection of ALTO Info to CDN Request Routing

The information provided by an ALTO server to Request Routing is based on topology information of the network. The different methods and algorithms through which the ALTO server computes topology information and rankings is out of the scope of this document. However, update and rediction of such information may have an impact on the integration of ALTO into CDN Request Routing.

6.5.1. ALTO Update and Network Events

In the case that ALTO information is based on routing (IP/MPLS) topology, it is obvious that network events may impact the ALTO computation. The scope of the ALTO information delivered to Request Routing is not to maintain the CDN aware of any possible network topology changes since, due to redundancy of current networks, most of the network events happening in the infrastructure will have limited impact on the CDN. However, catastrophic events such as main trunks failures or backbone partition will have to take into account by the ALTO server so to redirect traffic away from the failure impacted area.

6.5.2. Caching and Lifetime

Each reply sent back by the ALTO server to the ALTO client running in the Request Routing Function has a validity in time so that the CDN can cache the results in order to re-use it and hence reducing the number of transactions between CDN and ALTO server. The ALTO server may indicate in the reply message how long the content of the message is to be considered reliable and insert a lifetime value that will be used by the Request Routing Function in order to cache (and then flush or refresh) the entry.

An ALTO server implementation may want to keep state about ALTO clients so to inform and signal to these clients when a major network event happened so to clear the ALTO cache in the client. In a CDN/ ALTO interworking architecture, where there are only a few CDN components interacting with the ALTO server, there are no scalability issues in maintaining state about clients in the ALTO server.

6.5.3. ALTO Redirection

When ALTO server receives a request from a CDN Request Routing Function, it may not have the most appropriate topology information to reply. In such case, the ALTO server, may want to adopt the following strategies:

- o Reply with available information (best effort).
- o Redirect the request to another ALTO server presumed to have better topology information (redirection).
- o Doing both (best effort and redirection). In this case, the reply message contains both the rankings and the indication of another ALTO server where more accurate information may be delivered.

The decision process that is used to determine if redirection is necessary (and which mode to use) is out of the scope of this document. As an example, an ALTO server may decide to redirect any request having addresses that are located into a remote Autonomous System. In such case the redirection message includes the ALTO server to be used and that resides in the remote AS. Redirection implies communication between ALTO servers so to be able to signal their identity, location and type of visibility (AS number).

6.5.4. Groups and Costs

An automated ALTO implementation may use dynamic algorithms to aggregate network topology. However, it is often desirable to have a mechanism through which the network operator can control the level and details of network aggregation based on a set of requirements and constraints. IP/MPLS networks make use of a common mechanism to aggregate and group prefixes that is called BGP Communities. BGP is the protocol all ISP networks use in order to exchange information about their prefix reachability. BGP Community us an attribute used to tag a prefix so to group prefixes based on mostly any criteria (as an example, most SP networks originate BGP prefixes with communities identifying the Point of Presence (PoP) where the prefix has been originated).

The ALTO server may leverage the BGP information that is available in the ISP network layer and compute group of prefixes. By policy, the ALTO server operator may decide an arbitrary cost to set between groups. Alternatively, there are algorithms that allow dynamic computation of cost between groups.

7. Multiple Administrative Domains

The preceding discussion works well in a single administrative domain setting: the CDN nodes are in the administrative domain of the ISP. However, the CDN nodes, the ISP, and the Request Router can be in different administrative domains. In this section, we consider a few such deployment cases. We use DNS as an example.

7.1. CDN nodes/Request Router in a separate administrative domain from that of ISP

In many situations, the CDN nodes and the Request Router are in a separate network managed by an entity that is distinct from the ISP. Consequently, the CDN nodes belong to a network with its own ALTO server that is distinct from the ALTO server of the ISP where the subscribers belong to.

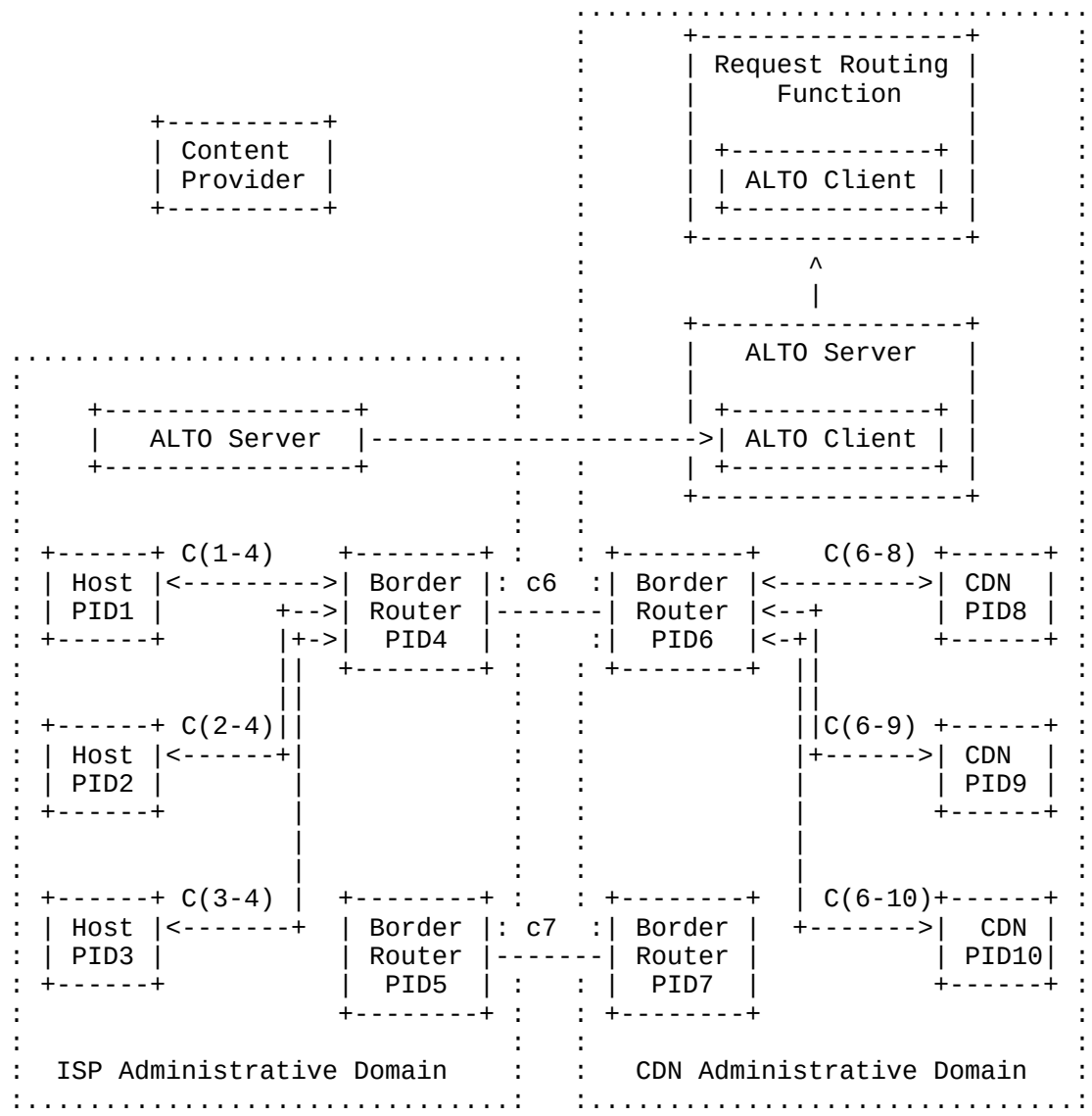


Figure 6: Map advertising between ISP and CDN domains

The ALTO server in the CDN provider network is assumed to be initialized with information about the ISP networks it serves. For every such ISP network, it consults the routing plane to find the set of Border routers. The CDN network ALTO server computes the cost of reaching each Border router from every CDN node (say, C_cdn).

Next, the CDN ALTO server contacts the ISP network's ALTO server and downloads the network map. In order to help the CDN ALTO server compute the cost from a CDN node to a subscriber's PID, we break it down into two parts - the cost from the CDN node to the Border Router (C_cdn) and the cost from the Border Router to the subscriber's PID (say, C_isp). Note that for any chosen exit point, C_cdn may be computed locally by the CDN ALTO Server. However, the fundamental issue is that C_isp depends on the exit point (Border outer) chosen by the CDN. There are multiple ways for the CDN ALTO Server to compute C_isp given the Network Map and Cost Map from the ISP's ALTO Server.

One possibility is for the ISP ALTO Server to define a special Border Router PID (denoted by a PID attribute) which also indicates the corresponding Border Router PID in the CDN. The attributes and values may be agreed-upon by the ISP and CDN when the ALTO Services are configured. For example, in the example shown in Figure 5, the ISP ALTO Server indicates that its PID4 and PID5 are Border PIDs, with corresponding PIDs in the CDN as PID6, and PID7, respectively. Then, CDN ALTO Server can locally compute C_isp = cost(ISP Border Router PID, Subscriber PID).

A second possibility for computing C_isp is to make use of Border Router IP addresses. The CDN's Border Router can locally determine the IP address of the connected border router in the ISP. In this approach, neither the CDN ALTO Server nor the ISP ALTO Server define PID attributes. The ISP ALTO Server is not required to define special PIDs for Border Routers - it only needs to ensure that Border Router IP addresses are aggregated appropriately in its Network Map.

Specifically, we identify two scenarios for the CDN ALTO Server to compute C_isp and C_cdn.

In the first scenario, the CDN does not conduct CDN-level multi-path routing from the CDN nodes to the subscriber hosts. Thus, the routing path from a CDN IP address to a subscriber host IP address is typically uniquely (if no ECMP) determined by the network routing system. In this scenario, for a given CDN node IP address to a subscriber host IP address, the CDN ALTO Server uses the routing system to compute the Border Egress router inside the CDN, and the corresponding Border Ingress router inside the ISP. Then the CDN ALTO Server has C_cdn(CDN node IP, Border Egress router IP inside the CDN), and C_isp(Border Ingress router IP inside the ISP, Subscriber IP). The computation of C_cdn and C_isp can be done using ALTO in the traditional way through either the Network Map and Cost Map or the Endpoint Cost Service.

In the second scenario, the CDN may support CDN-level multi-path

routing from the CDN nodes to the subscriber hosts. In particular, from each CDN node, the CDN has a capability (e.g., through tunneling) to send to a subscriber host IP through multiple Border Egress routers (e.g., through any Egress router that receives an announcement from the ISP of the subscriber host IP). In this case, the cost of reaching a host PID from a given CDN node is then determined as the minimum cost among all possible intermediate Border Routers.

If the network is homogeneous, then a good approximation of the cost between each host PID and a given CDN node can be given as: C_cdn(CDN Node, Border router) + C_isp(Border router, Subscriber PID). In this computation, the Border Router is the one that is on the best path from the CDN node to the Subscriber PID.

The CDN ALTO server now has a cost map that provides the cost from each CDN node to all known Subscriber PIDs. The ALTO client in the CDN DNS server downloads this cost map in preparation for subscriber DNS requests.

When a subscriber DNS request arrives at the CDN provider's DNS server, it looks up the network map and maps the source IP address to a Subscriber PID. It then uses the cost map to pick the best CDN node for this Subscriber PID.

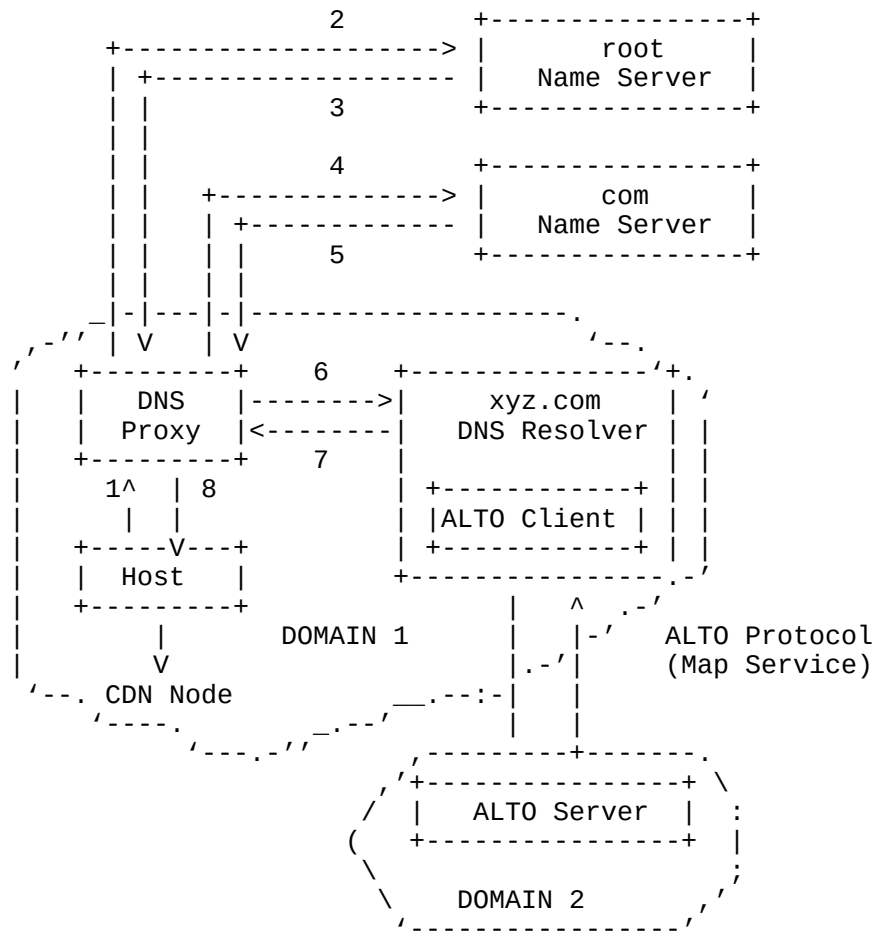
7.2. Managed DNS Domain with Three Administrative Domains

Many organizations / content providers outsource DNS management to the external vendors for various reasons like reliability, performance improvement, DNS security etc. Managed DNS service could be used either with caches owned by the organization itself (section 6.3.1) OR with external CDNs (section 6.3.2)

7.2.1. Managed DNS Redirect to Local CDN

One of the common functions offered by managed DNS service vendor is DNS traffic management where DNS resolver can load balance traffic dynamically across CDN servers.

Typically managed DNS service provider has DNS resolvers spread across geographical locations to improve performance. This also makes easier for DNS resolver to redirect host to the nearest cache. Such a DNS resolver would be an ideal candidate to implement ALTO client where it can fetch network map and cost map from ALTO servers located in the same geographical area only. Load balancing implemented with the knowledge of network and cost map would be more efficient than other mechanisms like round robin.



In the figure above, there exists 2 possibilities:

Case 1: Domain 1 and Domain 2 are connected to the same service provider network. This case is similar to section 6.1

Case 2: Domain 1 and Domain 2 are connected to different service provider network. This case is similar to section 6.2

7.2.2. Managed DNS with CDN-Provided Request Routing

It is also possible to utilize a Managed DNS service and still rely on a CDN's request routing. For example, this could be done if a network provider wishes to utilize a Managed DNS provider, but also wishes to integrate its own CDN using ALTO with DNS-based request routing.

To support this, the network provider may submit any necessary configuration files (e.g., indicating necessary CNAME records) to redirect CDN requests to the CDN's DNS Request Routing mechanism. Requests for the CDN (e.g., 'cdn.isp.com') will then be directed by DNS request routing, while requests for other hosts are handled by the Managed DNS solution.

8. Protocol Recommendations

In the previous sections, this document has taken the approach of providing information on existing CDN approaches and possible benefits of utilizing ALTO. However, in developing the taxonomy, use cases, and deployment scenarios, we have identified cases where the ALTO Protocol [I-D.ietf-alto-protocol] and Server Discovery [I-D.kiesel-alto-3pdisc] [I-D.song-alto-server-discovery] [I-D.stiemerling-alto-dns-discovery] may be lacking capabilities that may be helpful and/or necessary for usage with CDNs. We now focus on detailing these gaps with the goal of providing feedback and recommendations. Note that some protocol changes may be necessary in the core protocol, while others may be implemented as extensions.

This section will be updated to track changes in the ALTO Protocol, ALTO Server Discovery, and accompanying protocols.

8.1. Necessary Additions

This section details changes to the ALTO protocols that would be necessary to make use of ALTO within CDN infrastructures. We classify a change as "necessary" if there is a core feature of a CDN/ ALTO integration that is not possible to implement with the existing protocols.

8.1.1. NA1: PID Attributes

In order to disambiguate between PIDs that contain endpoints of a specific class, a PID property is needed. A PID can be classified as containing "CDN nodes", "Mobile Hosts", "Wireline Hosts", etc. This mechanism can be used to provide an ALTO Client a list of nodes of a particular type, along with the ALTO Costs to each node. In the context of CDNs, the attributes could describe a type of CDN node. For example an Origin would have one type of attribute while an edge cache would have another. This would allow for more intelligent routing.

8.1.2. NA2: PID Attributes and Query

PID attributes can be used by the ALTO Client to select a appropriate host and also passed as a constraint in the map filtering service.

8.2. Helpful Additions

This section details changes to the ALTO Protocol that would be helpful to make use of ALTO within CDN infrastructures. We classify a change as "helpful" if there is a compelling extension to existing CDNs that would be possible with additional functionality within ALTO, or if there is a component of CDN/ALTO integration that could be made more efficient or otherwised improved with additional ALTO functionality.

8.2.1. HA1: Push Mechanism

It is important for the ALTO Service through the ALTO protocol or a companion protocol to provide a push mechanism from server to client. The push mechanism can be a notification that new data is available or the data itself.

8.2.2. HA2: Incremental Map Updates

A natural evolution to the protocol if maps are large and change often is to allow for incremental map updates. In this sense the map contained in the reply would be considered the delta from the previous version.

8.2.3. HA3: ALTO Border Router PID attribute

In order for administrative domains to collate costs across domain boundaries, the border routers may be placed in their own PIDs. Such PIDs may be identified by a Border Router attribute.

8.2.4. HA4: CDN ALTO Server Discovery

In certain deployment scenarios, it may be beneficial for an ALTO client to directly query a CDN’s ALTO Server (instead of the CDN’s ALTO Server only being consulted as a backend process). For example, this can provide more accurate guidance than DNS Request Routing since the client’s IP address may be directly used by the CDN in order to select a cache node. This would require an ALTO Client (e.g., an ISP subscriber) to be able to discover an ALTO Server owned and/or managed by a CDN. This could be done by an extension to the discovery protocol, or it could be done by allowing an ISP’s ALTO Server to redirect certain queries to a CDN ALTO Server.

8.2.5. HA5: Extensible ALTO Cost Maps

Certain deployment scenarios may benefit from additional information being carried within ALTO information. For example, a trusted neighboring ISP B may be able to help ISP A optimize multihoming costs. To provide an extensible way to communicate additional data, the ALTO Protocol could be extended to include opaque data strings (in addition to numeric and ordinal values) in an ALTO Cost Map.

8.2.6. NA4: Federated Deployment of ALTO Servers

There is a need to define how ALTO servers may communicate with each other in a federated model.

9. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

10. Security Considerations

When the ALTO Server and Client are operated by different entities the issue of trust and security comes forward. The exchange of information could be done using the encryption methods already present in HTTP but preventing unauthorized redistribution comes into play. A further issue is if the ALTO information information is transitive, which modifications are allowed.

11. Acknowledgements

We would like to thank Satish Raghunath and Mayuresh Bakshi for valuable input and contributions to this draft. We would also like to thank Nabil Bitar, Manish Bhardwaj, Michael Korolyov, Steven Luong and Ferry Sutanto for their comments.

12. References

12.1. Normative References

[I-D.ietf-alto-protocol]
Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-06 (work in progress),

October 2010.

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

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

12.2. Informative References

[ARBOR] Labovitz, "Internet Traffic and Content Consolidation", 2009, <http://www.ietf.org/proceedings/10mar/slides/plenaryt-4.pdf>.

[GoogleCDN] Madhyastha, H., Jain, S., Srinivasan, S., Krishnamurthy, A., Anderson, T., and J. Gao, "Moving Beyond End-to-End Path Information to Optimize CDN Performance", 2009, <http://research.google.com/pubs/pub35590.html>.

[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.lee-alto-chinatelecom-trial] Li, K., Wang, A., and K. Zhou, "ALTO and DECADE service trial within China Telecom", draft-lee-alto-chinatelecom-trial-00 (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.

[I-D.stiemerling-alto-dns-discovery] Stiemerling, M. and H. Tschofenig, "A DNS-based ALTO Server Discovery Procedure", draft-stiemerling-alto-dns-discovery-00 (work in progress), July 2010.

[I-D.vandergaast-edns-client-subnet] Contavalli, C., Gaast, W., Leach, S., and D. Rodden, "Client subnet in DNS requests", draft-vandergaast-edns-client-subnet-00 (work in

progress), January 2011.

[P4P] Xie, H., Yang, YR., Krishnamurthy, A., Liu, Y., and A. Silberschatz, "P4P: Provider Portal for (P2P) Applications", March 2009.

[RFC3568] Barbir, A., Cain, B., Nair, R., and O. Spatscheck, "Known Content Network (CN) Request-Routing Mechanisms", RFC 3568, July 2003.

[RFC5632] Griffiths, C., Livingood, J., Popkin, L., Woundy, R., and Y. Yang, "Comcast's ISP Experiences in a Proactive Network Provider Participation for P2P (P4P) Technical Trial", RFC 5632, September 2009.

Authors' Addresses

Reinaldo Penno
Juniper Networks

Email: rpenno@juniper.net

Jan Medved
Juniper Networks

Email: jmedved@juniper.net

Richard Alimi
Google

Email: ralimi@google.com

Richard Yang
Yale University

Email: yry@yale.edu

Stefano Previdi
Cisco Systems

Email: sprevidi@cisco.com

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: September 15, 2011

S. Randriamasy, Ed.
Alcatel-Lucent Bell Labs

March 14, 2011

Multi-Cost ALTO
draft-randriamasy-alto-multi-cost-02

Abstract

IETF is designing a new service called ALTO (Application Layer traffic Optimization) that includes a "Network Map Service", an "Endpoint Cost Service" and an "Endpoint (EP) Ranking Service" and thus incentives for application clients to connect to ISP preferred Endpoints. These services provide a view of the Network Provider (NP) topology to overlay clients.

The present draft proposes a light way to extend the information provided by the current ALTO protocol. The purpose is to broaden the possibilities of the Application Clients in two ways: firstly by providing a better mapping of the Selected Endpoints to needs of the growing diversity of Content Networking Applications and to the network conditions, secondly by producing a more robust choice of multiple Endpoints, helping thus out for efficient Multi-Path transfer.

There are 2 parts in this draft: the first part proposes protocol extensions to support requests on multiple CostTypes in 1 transaction; the second part proposes additional CostTypes and Cost attributes related to timeframe and validity period.

Requirements Language

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 RFC 2119 [RFC2119].

Status of this Memo

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 September 15, 2011.

Table of Contents

1. Introduction

IETF is designing a new service called ALTO that provides guidance to P2P applications, which have to select one or several hosts from a set of candidates that are able to provide a desired resource. This guidance shall be based on parameters that affect performance and efficiency of the data transmission between the hosts, e.g., the topological distance. The ultimate goal is to improve Quality of Experience (QoE) in the application while reducing resource consumption in the underlying network infrastructure. The ALTO protocol conveys the Internet View from the perspective of a Provider Network region that spans from a region to one or more Autonomous System (AS). Together with this Network Map, it provides the Provider determined Cost Map between locations of the Network Map. Last, it provides the Ranking of Endpoints w.r.t. their routing cost.

The term Network Provider in this document includes both ISPs, who provide means to transport the data and Content Delivery Network (CDN) operators who care for the dissemination, persistent storage and possibly identification of the best/closest content copy.

The last ALTO protocol draft see [ID-alto-protocol6], gives the possibility to query multiple Endpoint properties at once (see S.7.7.4.1). However section 7.7.3.2 on Cost Map states about both parameters Cost Type and Cost Mode that: "This parameter MUST NOT be specified multiple times". The ALTO requirements draft, see [ID-ALTO-Requirements7] also states in REQ. ARv05-14: "The ALTO client protocol MUST support the usage of several different rating criteria types". In the current protocol draft, there is no specified way to get values for several Cost Types altogether. Currently, the costs are provided in a scalar form, one by one. So that an ALTO Client wanting information for several Cost Types must place a request and receive a response as many times as desired Cost Types. However, vector costs provide a robust and natural input to multi-path connections and getting all costs in one single query/ response transaction saves time and ALTO traffic, thus ressources, thus energy.

The ALTO Problem Statement, see [RFC5693] and the ALTO requirements draft, see [RFC5693] stress that: "information that can change very rapidly, such as transport-layer congestion, is out of scope for an ALTO service. Such information is better suited to be transferred through an in-band technique at the transport layer instead", as "ALTO is not an admission control system "and does not necessarily know about the instant load of endpoints and links. However, longer term statistics or empirical ratings on performance oriented information may still be useful for a reliable choice of candidate endpoints. In addition, given the QoE requirements of nowadays and

future Internet applications, more and more NPs compute and store such information to optimize their traffic. Last, specific ALTO servers can be specified for mobile core networks, which have a smaller scale and can afford and take advantage of using smaller time-scale network information.

Adding QoE-enabling metrics to the Network Provider established routing cost could meet the interests of both the end users and the Providers. Besides, keeping the shortest or cheapest possible path, in addition, saves resources, time and energy.

2. Scope

This draft generalizes the case of a P2P client to include the case of a CDN client, a GRID application client and any Client having the choice in several connection points for data or resource exchange. To do so, it uses the term "Application Client" (AC).

This draft focuses on the use case where the ALTO client is embedded in the Application Client. For P2P applications, the use case where the ALTO Client is embedded in the P2P tracker is also applicable.

It is assumed that Applications likely to use the ALTO service have a choice in connection endpoints as it is the case for most of them. The ALTO service is managed by the Network Provider and reflects its preferences for the choice of endpoints. The NP defines in particular the network map, the routing cost among Network Locations, and which ALTO services are available at a given ALTO server.

The solution proposed in this draft is applicable to fixed networks. It is also meant for smaller networks such as mobile networks.

3. Terminology

Endpoint (EP): can be a Peers, a CDN storage location, a Party in a resource sharing swarm such as Grid or online gaming.

Endpoint Discovery (EP Discovery) : this term embraces the different types of processes used to discover different types of endpoints.

Network provider: includes both ISPs, who provide means to transport the data and Content Delivery Network (CDN) who care for the dissemination, persistent storage and possibly identification of the best/closest content copy.

Application Client (AC): this term generalizes the case of a P2P

client to include the case of a CDN client and of any Client having the choice in several connection points for data or resource exchange.

Traffic Engineered End Point Optimization Tool (TEEPOT): this is a functional entity introduced in this draft, that is linked to an ALTO Client and to an Application Client. Its role is to assist the selection of Endpoints upon Allication needs and the ALTO responses. It can be a specific group of functions or an already existing function.

4. Proposed ALTO services updates

The currently available ALTO services supporting Endpoint evaluation are: Endpoint Cost Service, Cost Map and Filtered Cost Map. The ALTO client may want to simultaneously use a number N>1 of cost metrics referred to as Cost Types in ALTO. The only possibility in the current ALTO protocol is to sequentially place as many requests as desired cost types. This draft proposes to add the following features:

4.1. Endpoint Cost Service with multiple Cost Types

Some application clients may want to consider several metrics to select the endpoints appropriately w.r.t. the application needs. Clients may also want to use multiple paths for the transfer of particular data bulks, possibly selected with several metrics. Therefore the Endpoint Cost Lookup and the Cost Map Services should have the possibility to handle several metrics.

4.2. All Costs Types in one response with vector cost values

Providing all the numerical costs simultaneously with only one request and response exchange saves time, resources and energy. To avoid overloading the network with ALTO traffic with multiple requests for Cost Types, we propose that the Cost values provided by the ALTO server be arranged in a vector. This requires:

- to put the requested cost values in an array or vector having a number N >= 1 of components.
- to define a canonical order that allows to match values in these vectors with Cost Types and Properties.

As specified in the ALTO Requirements [ID-ALTO-Requirements7] "REQ. ARv05-19: The ALTO reply message SHOULD allow the ALTO server to express which rating criteria have been considered when generating

the reply." That is, the ALTO response indicates the mapping between vector components and Cost Types.

Note that in this case, the ALTO client MUST require the Cost Mode "numerical" that is the Mode MUST NOT be "ordinal".

4.3. Proposed additional Cost Types

The current ALTO protocol draft provides examples of metrics in section 5.1.1, that are: air miles, hop-counts or generic routing costs. Statistics or longer term ratings on path bandwidth and latency may also be considered. Additional Endpoint properties may be useful, such as the memory capacity or statistical scores on the load and possibilities of an Endpoint.

4.4. Statistical Costs with a timeframe

The ALTO Requirements Draft [ID-ALTO-Requirements7] advises against instant performance-related cost metrics as they may be easily captured by online mechanisms and in addition, the ALTO service does not know how a Peer manages its sending rate. Application clients however may have good reasons and wise ways to use performance related information in the mid to long term ,on Endpoints that they don’t know in advance and on which they therefore cannot plan measurements. Other applications may wisely use static performance indicators such as nominal memory capacity.

Dynamic performance indicators can be represented by scores, reflecting some overall performance, in a static way or with values periodically updated at intervals typically longer than a network layer packet RTT, as assumed in [ID-ALTO-Requirements7].

If statistical Cost Types are available, the following types of information should report on them:

- their "statistical" nature: for example a mean value, or a median value,
- their timeframe: that is the period over which statistics were computed and the age of the information. By default this timeframe is supposed permanent , that is, the corresponding EP Cost or Property values are permanent. Timeframe information can be easily recovered by attributes listed in [ID-ALTO-Requirements7] such as ’lifetime’ (see REQ. ARv07-29) and an aging mechanism (see REQ. ARv07-29), such as a RFC3339 based TimeStamp.

- the validity period: indicating the date at which the information can be considered obsolete and updated. This can be easily reflected by the 'age' reflecting the date at which the information was generated and 'lifetime' of this information.

'Lifetime' and 'age' should be also available to other applicable 'non statistical' Cost Types, such as 'OccupationLevel' that can be used to describe an empirical and restricted set of load value ranges.

5. Proposed ALTO protocol updates

This section proposes updates or additions to the ALTO protocol to support Multi Cost ALTO Services or provide additional ALTO information. The applicable ALTO services are:

- Cost Map Service,
- Cost Map Filtering Service,
- Endpoint Property Lookup Service,
- Endpoint Cost Lookup Service.

5.1. Proposed updates for Multi-Cost ALTO

If an ALTO client desires several Cost Types, instead of placing as many requests as costs, it may request and receive all the desired cost types in one transaction. The correspondence between the components and the cost type MUST be indicated in the ALTO request.

The ALTO server then, provided it supports the desired cost, and provided it supports the vector cost values, sends one single response where for each {source, destination} pair, the cost values are arranged in a vector, whose component each corresponds to a specified Cost Type. The correspondence between the components and the cost types MUST be indicated in the ALTO response.

The following ALTO protocol services and features need to be updated to enable Multi Cost ALTO transactions.

- Endpoint (EP) Property (see [ID-alto-protocol6])
- Endpoint (EP) Cost (see [ID-alto-protocol6]).
- Cost attributes (see [ID-alto-protocol6]).

- Cost Map (see [ID-alto-protocol6]):
  - between Network Locations (that are groups of 1 or several endpoints).
- Cost Map filtering: need the same updates as for the Cost Map.

5.1.1. Multi-Cost related Attributes

To enable Multi-Cost ALTO Cost Services, we propose the following updates to the Cost Attributes, described in [ID-alto-protocol6] .

- extension of the attribute Cost Type from a single value to a vector of N >= 1 values. If N > 1, then the values WILL be interpreted as numerical values.
- addition of definitions that list and identify the Cost Types supported by the acting ALTO server. These definitions will be formulated according to the syntax defined in Section 7.7 of [ID-alto-protocol6],
- definition of the correspondence between an index "i_typecost" in [1,N] in a cost vector and the ID of the defined cost types and properties.
- optional association of a validity timeframe, indicating how long the information can be considered as up to date.
  - by default the validity timeframe WILL be considered infinite

To the attribute Cost Mode in S.5.1: addition of a rule stipulating that when multiple cost types are requested, then the requested Cost Mode MUST be numerical. If the attribute Cost Length is > 1 and the Cost Mode is set to "ordinal", then one option is that the ALTO Server returns the ’Sucess’ code "E_INVALID_COST_TYPE".

5.2. Proposed additional Properties and Costs

5.2.1. Proposed additional Endpoints properties

The Endpoint Properties given as example in [ID-alto-protocol6] S.3.2.3 mostly apply to fixed end nodes. We propose to add other properties, that are static, contribute to reflect the potential physical abilities of end nodes and therefore may guide their selection. In addition, these properties apply to end nodes connected by any access technology. Example additional properties include:

- o EP capacity in memory,
- o EP nominal bandwidth,
- o EP access technology.

Note that if this service is not supported, it is possible although less convenient to get the information at the overlay level, thus without the ALTO server.

5.2.2. Scoping ALTO information

One way to moderate the ALTO traffic load while maintaining some reliability is to associate the following attributes to the applicable ALTO information:

- o an age attribute indicating when the information was generated.
- o for statistical costs a time period attribute indicating over which period the statistics were collected.
- o a lifetime attribute as proposed in [ID-ALTO-Requirements7] . By defaut, this parameter can be set to infinity.

The Time related values can be used by the aging mechanism as proposed in REQ ARv05-28 of [ID-ALTO-Requirements7] for a better synchronization of Cost Information collected at various times and places.

5.2.3. Proposed additional Cost Types

Additional Cost Types may be used in either the Cost Map or the Endpoint Cost Lookup Services and include:

- o Endpoint availability: indicating how often an Endpoint is reachable, preferebly as a percentage. To be further specified. Possibly with associated Time frame and Time To Expire.
- o Endpoint reliability: indicating how easily an Endpoint is reachable, and / or the degree of continuity of its reachability, preferebly as a percentage. To be further specified.
- o Endpoint Load: indicating the average load, preferably as a percentage, or a quantitative coarse grain index indicating whether this Endpoint is in a rush period or calm period. To be further specified.

- Path robustness: one or more timeframed indicators related to statistical evaluations of the path performance on bandwidth, delay, packet loss, or other such metrics. This Cost can also be represented by a quantitative coarse grain index indicating whether this Endpoint is in a rush period or calm period. To be further specified.

5.3. ALTO Status Codes for Multi-Cost ALTO

If the vector cost structure is not supported, then the ALTO server sends an ALTO status code 7 corresponding to HTTP status code 501 indicating "Invalid cost structure". The ALTO client may then needs to place as many requests as needed Cost Types, and the ALTO server sends as many cost maps or EP cost as needed.

To the attribute Cost Mode in S.5.1 should be associated a rule stipulating that when multiple cost types are requested, then the requested Cost Mode MUST be numerical. If the attribute Cost Length is > 1 and the Cost Mode is set to "ordinal", an option is that the ALTO Server returns the ’Sucess’ code "E_INVALID_COST_TYPE".

5.4. Examples of Multi-Cost ALTO messages

Request and Response syntax. To be further specified.

6. Use case

6.1. Scenario

A Multi-Cost ALTO transaction is illustrated in a simple scenario, where an application client in a terminal wants to use several paths for a data transfer. This scenario applies to a terminal having access to the network via one or several interfaces.

The application client wants for example 3 paths per transfer:

- 1 path optimising the Cost Type ’routingcost’,

- 2 other paths optimizing 2 metrics: the Cost Type ’routingcost’ and an Endpoint property named ’EP memory’.

  * The application client in addition wants these 2 paths to optimize the first criterion with a weight W_PATH_LENGTH equal for example to 0.4 and the second criterion with a weight W_EP_MEMORY equal to 0.6.

* If the EP Property Service provides the information on Endpoint Load, then the application client wants this information in the available lifetime closest to 1 hour.

A TEEPOT connected with the ALTO Client and the Application Client takes in the list of candidate Endpoints from the Application Client and prepares for the ALTO Client the request to the ALTO Server, in particular the following information: vector TimeFrame[EP Cost Length], with components equal to either a value or an indication of "not applicable".

o The list of requested EP Cost Types, that are identified by their index I_CostType.

o

6.2. Illustrative ALTO use case

Figure 1 shows the example scenario in the last IETF ALTO protocol draft, where the ALTO client is embedded in the P2P Client and requires an ALTO server servicing its own ISP to provide the Endpoint Cost for a list of gethered peers.

As written in [ID-alto-protocol6], the use case proceeds as follows:

1. The P2P Client discovers peers from sources such as Peer Exchange (PEX) from other P2P Clients, Distributed Hash Tables (DHT), and P2P Trackers.

2. The P2P Client queries the ALTO Server’s Ranking Service, including discovered peers as the set of Destination Endpoints, and indicates the ’ordinal’ Cost Mode. The response indicates the ranking of the candidate peers.

3. The P2P Client connects to the peers in the order specified in the ranking.

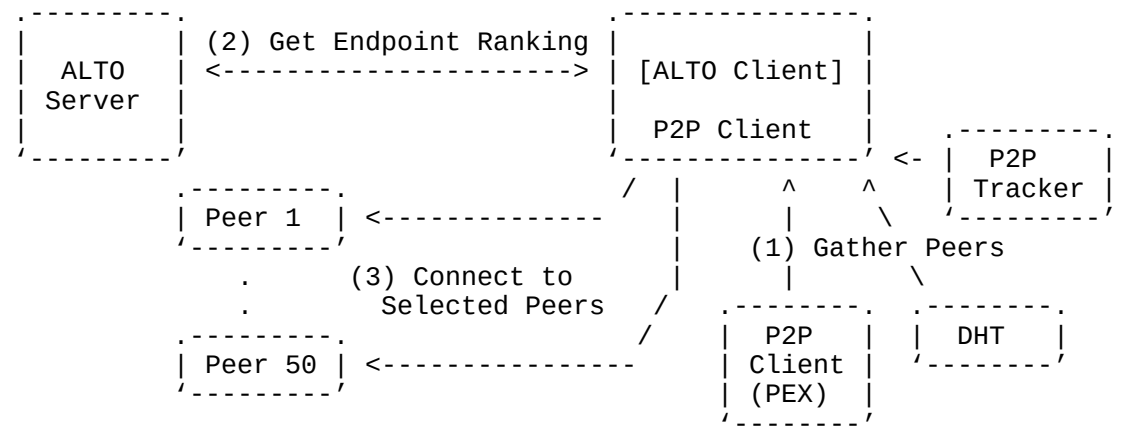


Figure 1:example scenario in the last IETF ALTO protocol draft, where the ALTO client is embedded in the P2P Client

Figure 2 depicts the features and mechanisms added to the current ALTO scenario for Multi-Cost ALTO services, for the use case of Figure 1. The EPs have already been discovered. In this figure, the term Peer is replaced by the term Endpoint (EP), the term P2P Client by Application Client and an Endpoint Tracker for resource Sharing Applications is added to the tools involved in Step (1) Gather Endpoints .

We focus on the ALTO use case where the ALTO client is co-located with an Application client in a terminal node, as not all P2P systems use a P2P tracker for peer discovery and selection as written in section 9.2 of [ID-alto-protocol6]. In Figure 2, the entity called P2P Client mentionned in the current protocol draft is zoomed to an entity called in this draft "Client Block" and that links: the Application Client (AC), its ALTO Client and the Traffic Engineered EP Optimization Tool (TEEPOT).

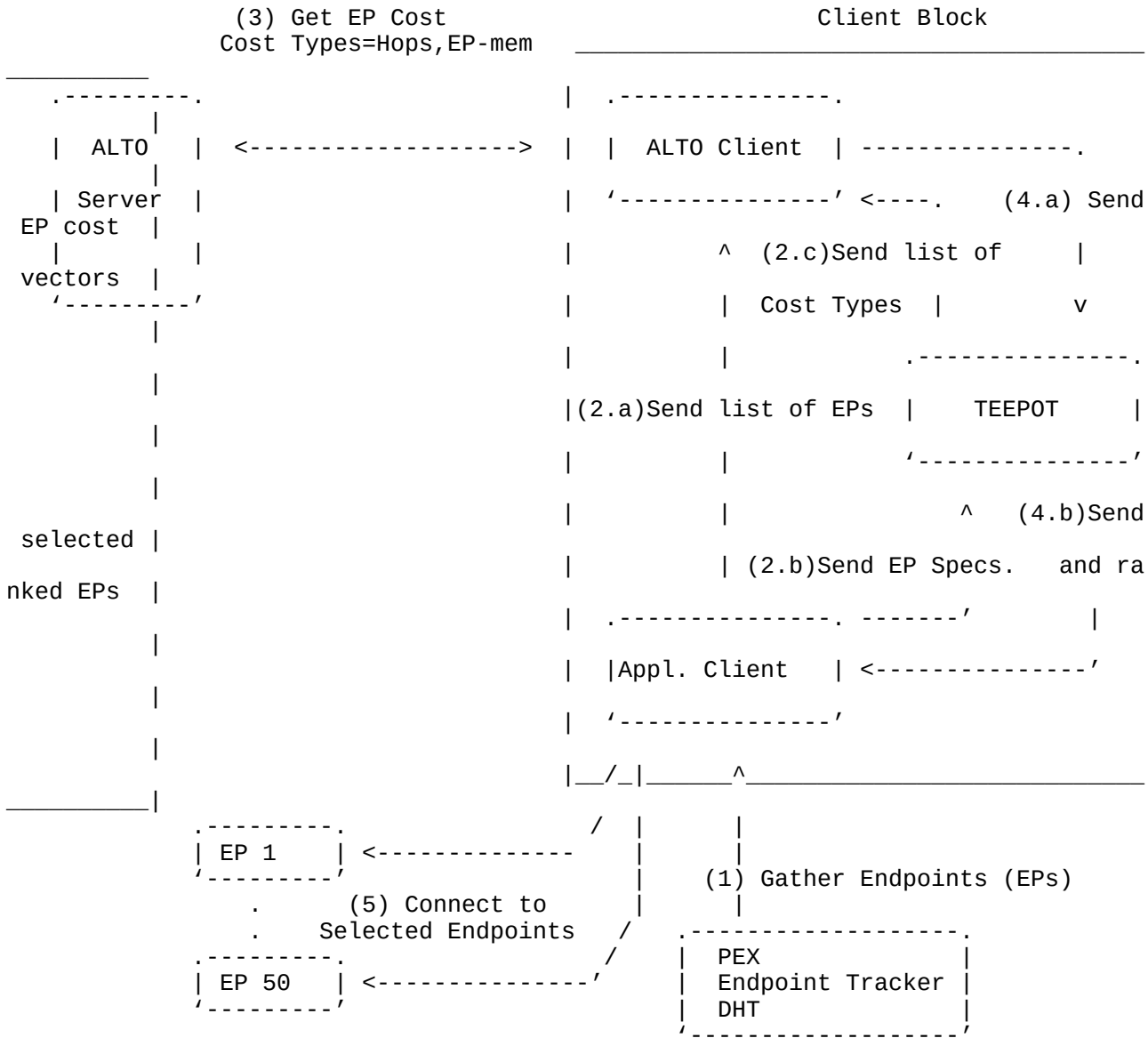


Figure 2: features and mechanisms added to the current ALTO scenario for Multi-Cost ALTO services

The use case in Figure 2 proceeds as follows:

1. The Application Client discovers Endpoints (EPs) from sources such as Peer Exchange (PEX) from other P2P Clients, Distributed Hash Tables (DHT), P2P Trackers or other types of EP trackers.

2. In the "Client Block" gathering the Application Client (AC), its ALTO Client and the Traffic Engineered EP Optimization Tool (TEEPOT):

   A. the Application Client (AC) sends to the ALTO Client the list of the discovered peers as the set of Destination Endpoints.

   B. the Application Client (AC) sends to the TEEPOT the specifications on the EPs to select, according to the needs of the application. For example, AC needs 3 EPs, with 1 EP optimizing the Path Length Metric and 2 EPs optimizing the Path Length and the EP Memory Capacity Score, with respective

weights of 0.4 and 0.6.

C. the TEEPOT indicates to the ALTO Client that the Service to request is EP Cost, with the Cost Mode set to "Numerical", and the Cost Dimension equal to the number of requested metrics and with the index of the requested Cost Types.

3. The ALTO Client queries the ALTO Server's EP Cost Service, sends the list of the discovered peers as the set of Destination Endpoints and the index of requested metrics, corresponding in this example to: "Path Length" and "EP Memory Capacity Score". As the number of requested metrics is > 1, the Cost Mode is implicitely set to 'numerical'. The ALTO Server response contains the set of metric values associated to each EP.

4. In the Client block:

A. The ALTO Client hands to the TEEPOT the list of EPs and their associated value set.

B. The TEEPOT ranks the EPs with some smart algorithm, given the metric weights and then sends the ranked list to the Application Client.

5. The Application Client connects to the selected EPs.

7. IANA Considerations

The current ALTO protocol version includes a Section 11 entitled IANA considerations, where the ALTO Cost Type registry is defined in Section 11.2

7.1. Information for IANA on proposed Cost Types

When a new ALTO Cost Type is defined, accepted by the ALTO working group and requests for IANA registration MUST include the following information, detailed in Section 11.2: Identifier, Intended Semantics, Security Considerations.

7.2. Information for IANA on proposed Endpoint Propeeries

Likewise, an ALTO Endpoint Property Registry could serve the same purposes as the ALTO Cost Type registry. Application to IANA registration for Endpoint Properties would follow a similar process.

8. Acknowledgements

Thank you to Richard Alimi whose reviewing of the previous version of this draft and advises provided a valuable input to its updates.

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.

[RFC5693] "Application Layer Traffic Optimization (ALTO) Problem Statement", October 2009.

9.2. Informative References

[ID-ALTO-Requirements] "draft-ietf-alto-reqs-05.txt", June 2010.

[ID-ALTO-Requirements7] "draft-ietf-alto-reqs-07.txt", January 2011.

[ID-alto-protocol5] ""ALTO Protocol" draft-ietf-alto-protocol-05.txt", July 2010.

[ID-alto-protocol6] , eds., ""ALTO Protocol" draft-ietf-alto-protocol-06.txt", October 2010.

Author's Address

Sabine Randriamasy (editor)
Alcatel-Lucent Bell Labs
Route de Villejust
NOZAY 91460
FRANCE

Email: Sabine.Randriamasy@alcatel-lucent.com

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: January 12, 2012

S. Randriamasy, Ed.
Alcatel-Lucent Bell Labs
N. Schwan

July 11, 2011

Multi-Cost ALTO
draft-randriamasy-alto-multi-cost-03

Abstract

IETF is designing a new service called ALTO (Application Layer traffic Optimization) that includes a "Network Map Service", an "Endpoint Cost Service" and an "Endpoint (EP) Ranking Service" and thus incentives for application clients to connect to ISP preferred Endpoints. These services provide a view of the Network Provider (NP) topology to overlay clients.

The present draft proposes a light way to extend the information provided by the current ALTO protocol. The purpose is to broaden the possibilities of the Application Clients in two ways: firstly by providing a better mapping of the Selected Endpoints to needs of the growing diversity of Content Networking Applications and to the network conditions, secondly by producing a more robust choice of multiple Endpoints, helping thus out for efficient Multi-Path transfer.

There are 2 parts in this draft: the first part initiates protocol extensions to support requests on multiple Cost Types in one single transaction. These first extensions also integrate the discussions within the ALTO Working Group and focus on the Endpoint Cost Service. The second part proposes two use cases motivating further definitions of additional CostTypes and Cost Attributes related to timeframe and validity period.

Requirements Language

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 RFC 2119 [RFC2119].

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 January 12, 2012.

Copyright Notice

Table of Contents

1. Introduction

IETF is designing a new service called ALTO that provides guidance to P2P applications, which have to select one or several hosts from a set of candidates that are able to provide a desired resource. This guidance shall be based on parameters that affect performance and efficiency of the data transmission between the hosts, e.g., the topological distance. The ultimate goal is to improve Quality of Experience (QoE) in the application while reducing resource consumption in the underlying network infrastructure. The ALTO protocol conveys the Internet View from the perspective of a Provider Network region that spans from a region to one or more Autonomous System (AS). Together with this Network Map, it provides the Provider determined Cost Map between locations of the Network Map. Last, it provides the Ranking of Endpoints w.r.t. their routing cost.

The term Network Provider in this document includes both ISPs, who provide means to transport the data and Content Delivery Network (CDN) operators who care for the dissemination, persistent storage and possibly identification of the best/closest content copy.

The last ALTO protocol draft see [ID-alto-protocol], gives the possibility to query multiple Endpoint properties at once (see S.7.7.4.1). However section 7.7.3.2 on Cost Map states about both parameters Cost Type and Cost Mode that: "This parameter MUST NOT be specified multiple times". The ALTO requirements draft, see [ID-ALTO-Requirements7] also states in REQ. ARv05-14: "The ALTO client protocol MUST support the usage of several different rating criteria types". In the current protocol draft, there is no specified way to get values for several Cost Types altogether. Currently, the costs are provided in a scalar form, one by one. So that an ALTO Client wanting information for several Cost Types must place a request and receive a response as many times as desired Cost Types. However, vector costs provide a robust and natural input to multi-path connections and getting all costs in one single query/ response transaction saves time and ALTO traffic, thus ressources, thus energy.

The ALTO Problem Statement, see [RFC5693] and the ALTO requirements draft, see [RFC5693] stress that: "information that can change very rapidly, such as transport-layer congestion, is out of scope for an ALTO service. Such information is better suited to be transferred through an in-band technique at the transport layer instead", as "ALTO is not an admission control system "and does not necessarily know about the instant load of endpoints and links. However, longer term statistics or empirical ratings on performance oriented information may still be useful for a reliable choice of candidate endpoints. In addition, given the QoE requirements of nowadays and

future Internet applications, more and more NPs compute and store such information to optimize their traffic. Last, specific ALTO servers can be specified for mobile core networks, which have a smaller scale and can afford and take advantage of using smaller time-scale network information.

Adding QoE-enabling metrics to the Network Provider established routing cost could meet the interests of both the end users and the Providers. Besides, keeping the shortest or cheapest possible path, in addition, saves resources, time and energy.

2. Scope

This draft generalizes the case of a P2P client to include the case of a CDN client, a GRID application client and any Client having the choice in several connection points for data or resource exchange. To do so, it uses the term "Application Client" (AC).

This draft focuses on the use case where the ALTO client is embedded in the Application Client. For P2P applications, the use case where the ALTO Client is embedded in the P2P tracker is also applicable.

It is assumed that Applications likely to use the ALTO service have a choice in connection endpoints as it is the case for most of them. The ALTO service is managed by the Network Provider and reflects its preferences for the choice of endpoints. The NP defines in particular the network map, the routing cost among Network Locations, and which ALTO services are available at a given ALTO server.

The solution proposed in this draft is applicable to fixed networks. It is also meant for smaller networks such as mobile networks.

3. Terminology

Endpoint (EP): can be a Peer, a CDN storage location, a Party in a resource sharing swarm such as a computation Grid or an online multi-party game.

Endpoint Discovery (EP Discovery) : this term covers the different types of processes used to discover different types of endpoints.

Network Service Provider: includes both ISPs, who provide means to transport the data and Content Delivery Network (CDN) who care for the dissemination, persistent storage and possibly identification of the best/closest content copy.

Application Client (AC): this term generalizes the case of a P2P client to include the case of a CDN client and of any Client having the choice in several connection points for data or resource exchange.

4. Proposed ALTO protocol updates for multi-cost transactions

This section is to be completed and proposes first updates of the ALTO protocol to support Multi Cost ALTO Services or provide additional ALTO information. It integrates discussions on he ALTO mailing list and its goal is to initiate further discussions and protocol update proposals.

If an ALTO client desires several Cost Types, instead of placing as many requests as costs, it may request and receive all the desired cost types in one single transaction. The correspondence between the components and the cost types MUST be indicated in the ALTO request.

The ALTO server then, provided it supports the desired cost, and provided it supports multi-cost ALTO transactions, sends one single response where for each {source, destination} pair. The cost values are arranged in a vector, whose component each corresponds to a specified Cost Type. The correspondence between the components and the cost types MUST be indicated either in the ALTO response or available via the resource directory.

The ALTO services impacted by the Multi-Cost extensions are:

- Information Resources Directory,
- Cost Map Service,
- Cost Map Filtering Service,
- Endpoint Cost Lookup Service.

This draft focuses on the case of the Endpoint Cost Lookup Service

4.1. Multi-Cost Map Service

To be completed

4.1.1. Media Type

4.1.2. HTTP Method

This resource is requested using the HTTP POST method

4.1.3. Input Parameters

4.1.4. Capabilities

4.1.5. Response

4.1.6. Example

4.2. Endpoint Multi-Cost Service

4.2.1. Media Type

4.2.2. HTTP Method

This resource is requested using the HTTP POST method

4.2.3. Input Parameters

Input parameters are supplied in the entity body of the POST request. This document specifies input parameters with a data format indicated by media type "application/alto-endpointcostparams+json", which is a JSON Object of type ReqEndpointCostMap:

object {

TypedEndpointAddr srcs<0..*>; [OPTIONAL]

TypedEndpointAddr dsts<1..*>;

} EndpointFilter;

object {

CostMode cost-mode;

CostType cost-type<1..*>;

JSONString constraints; [OPTIONAL]

EndpointFilter endpoints;

} ReqEndpointCostMap;

with members:

cost-mode The Cost Mode (Section 5.1.2) to use for returned costs. For Multi-Cost requests this Cost Mode SHOULD be numerical.

cost-type The Cost Type (Section 5.1.1) to use for returned costs. All the listed the Cost Types MUST be indicated in this resource's capabilities (Section 7.7.5.1.4).

constraints Defined equivalently to the "constraints" input parameter of a Filtered Cost Map (see Section 7.7.3.2).

endpoints A list of Source Endpoints and Destination Endpoints for which Path Costs are to be returned. If the list of Source Endpoints is empty (or not included), the ALTO Server MUST interpret it as if it contained the Endpoint Address corresponding Alimi, et al. Expires December 29, 2011 [Page 50] Internet-Draft ALTO Protocol June 2011 to the client IP address from the incoming connection (see Section 10.3 for discussion and considerations regarding this mode). The list of destination Endpoints MUST NOT be empty. The ALTO Server MUST interpret entries appearing multiple times in a list as if they appeared only once.

4.2.4. Capabilities

4.2.5. Response

4.2.6. Example

4.3. ALTO Status Codes for Multi-Cost ALTO

If the Multi-cost Service is not supported for either the Cost Map or the Endpoint Service, then the ALTO server sends an ALTO status code 7 corresponding to HTTP status code 501 indicating "Invalid cost structure". The ALTO client may then needs to place as many requests as needed Cost Types, and the ALTO server sends as many cost maps or EP cost as needed.

To the attribute Cost Mode in S.5.1 should be associated a rule stipulating that when multiple cost types are requested, then the requested Cost Mode SHOULD be numerical.

5. Use cases for further Cost Types and Endpoint Properties

The current ALTO protocol [ID-alto-protocol] specification requests the creation of two registries maintained by IANA. The ALTO Cost Type registry ensures that the Cost Types that are represented by an ALTO Cost Map are unique identifiers, and it further contains references to the semantics of the Cost Type. The current

specification registers 'routingcost' as a generic measure for routing traffic from a source to a destination. In a similar way the ALTO Endpoint Property Registry ensures uniquenes of ALTO Endpoint Property identifiers and provides references to particular semantics of the allocated Enpoint Properties. Currently the 'pid' identifier is registered, which serves as an identifier that allows aggregation of network endpoints into network regions. Both registries accept new entries after Expert Review [[ID-alto-protocol]]. New entries are requested to be conform to the respective syntactical requirements, and must include information about the new identifier, the intended semantics as well as security considerations.

The current protocol specification concentrates on the basic use case of optimizing routing costs in NSPs networks. Upcoming use cases however will require both, new Cost Types and new Endpoint Properties. The goal of this section is to describe further forward looking use case scenarios that are likely to benefit from ALTO, and, in future iterations, to convey new Cost Types and Endpoint Properties that are likely to be beneficial for ALTO clients in these scenarios.

5.1. Delay Sensitive Overlay Applications

The ALTO working group has been created to allow P2P applications and NSPs a mutual cooperation, in particular because P2P bulk file-transfer applications have created a huge amount of intra-domain traffic. By aligning overlay topologies according to the 'routingcost' of the underlying network both layers are expected to benefit in terms of reduced costs and improved Quality-of-Experience.

However other types of overlay applications might benefit from a different set of path metrics. In particular for real-time sensitive applications, such as gaming, interactive video conferencing or medical services, creating an overlay topology with respect to a minimized delay is preferable. However it is very hard for a NSP to give accurate guidance for this kind of realtime information, instead probing through end-to-end measurements on the application layer has proven to be the superior mechanism. Still, a NSP might give some guidance to the overlay application, for example by providing statisically preferable paths with respect to the time of a day. Also static information like hopcount can be seen as an indicator for the delay that can be expected. In the following iterations this draft will thus analyse which metrics can realisticaly be provided through ALTO to give delay sensitive applications guidance for peer selection.

5.2. CDN Surrogate Selection

A second use case is motivated through draft [draft-jenkins-alto-cdn-use-cases-01]. The request router in today's CDNs makes a decision about to which surrogate or cache node a content request should be forwarded to. Typically this decision is based on locality aspects, i.e. the surrogate node which is closest to the client is preferred by the request router. An ALTO server hereby is one promising option to allow NSPs to give guidance to the CDN about which cache node would be preferable according to the view of the network by the 'routingcost' Cost Type. Providing this kind of information is in particular important as one trend is to place cache nodes deeper into the network, which results in the need for finer grained information.

While distance today is the predominant metric used for routing decisions, other metrics might allow sophisticated request routing strategies. For example the load a cache node sees in terms of CPU utilization, memory usage or bandwidth utilization might influence routing decisions for load-balancing reasons. There exist numerous ways of gathering and feeding this kind of information into the request routing mechanism. As ALTO is likely to become a standardized interface to provide network topology information, for simplicity other information that is used by a request router could be provided by the ALTO server as well. In the next iterations of this draft we will analyse which of these metrics is suitable to be provided as Cost Type or Endpoint Property for the use case of CDN Surrogate Selection and propose to register them in the respective registries.

6. Proposed additional Properties and Costs

To be further specified

6.1. Scoping ALTO information

One way for the NSP to provide guidance on highly dynamic network state information such as delay and load is to provide them in the form of statistics or as a numerical coarse grain indicator. It is important to have the possibility to reflect that the provided values are applicable for a given time period, for example business hours, and are subject to changes over time.

The following attributes can be associated to the applicable ALTO information:

- o an age attribute indicating when the information was generated.
- o for statistical costs a time period attribute indicating over which duration the statistics were collected.
- o a validity attribute indicating when the provided values should be refreshed. By defaut, this parameter can be set to infinity.

7. IANA Considerations

Information for the ALTO Endpoint property registry maintained by the IANA and related to the new Endpoints supported by the acting ALTO server. These definitions will be formulated according to the syntax defined in Section on "ALTO Endpoint Property Registry" of [ID-alto-protocol],

Information for the ALTO Cost Type Registry maintained by the IANA and related to the new Cost Types supported by the acting ALTO server. These definitions will be formulated according to the syntax defined in Section on "ALTO Cost Type Registry" of [ID-alto-protocol],

7.1. Information for IANA on proposed Cost Types

When a new ALTO Cost Type is defined, accepted by the ALTO working group and requests for IANA registration MUST include the following information, detailed in Section 11.2: Identifier, Intended Semantics, Security Considerations.

7.2. Information for IANA on proposed Endpoint Propeeries

Likewise, an ALTO Endpoint Property Registry could serve the same purposes as the ALTO Cost Type registry. Application to IANA registration for Endpoint Properties would follow a similar process.

8. Acknowledgements

Sabine Randriamasy is partially supported by the MEDIEVAL project (http://www.ict-medieval.eu/), a research project supported by the European Commission under its 7th Framework Program (contract no. 248565). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the MEDIEVAL project or the European Commission.

Nico Schwan is partially supported by the ENVISION project

(http://www.envision-project.org), a research project supported by the European Commission under its 7th Framework Program (contract no. 248565). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ENVISION project or the European Commission.

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.

[RFC5693] "Application Layer Traffic Optimization (ALTO) Problem Statement", October 2009.

9.2. Informative References

[ID-ALTO-Requirements7] "draft-ietf-alto-reqs-07.txt", January 2011.

[ID-alto-protocol] , Eds., ""ALTO Protocol" draft-ietf-alto-protocol-09.txt", June 2011.

[draft-jenkins-alto-cdn-use-cases-01] ""Use Cases for ALTO within CDNs" draft-jenkins-alto-cdn-use-cases-01", June 2011.

Authors' Addresses

Sabine Randriamasy (editor)
Alcatel-Lucent Bell Labs
Route de Villejust
NOZAY 91460
FRANCE

Email: Sabine.Randriamasy@alcatel-lucent.com

Nico Schwan

Phone:
Fax:
Email:
URI: