

ALTO
Internet-Draft
Intended status: Informational
Expires: January 16, 2014

M. Stiemerling, Ed.
NEC Europe Ltd.
S. Kiesel, Ed.
University of Stuttgart
S. Previdi
Cisco
M. Scharf
Alcatel-Lucent Bell Labs
July 15, 2013

ALTO Deployment Considerations
draft-ietf-alto-deployments-07

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. This memo discusses deployment related issues of ALTO. It addresses different use cases of ALTO such as peer-to-peer file sharing and CDNs, security considerations, recommendations for network administrators, and also 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 16, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. General Considerations	4
2.1. Placement of ALTO Entities	4
2.2. Relationship between ALTO and Applications	6
2.3. Provided Guidance	6
2.3.1. Keeping Traffic Local in Network	7
2.3.2. Off-Loading Traffic from Network	8
2.3.3. Intra-Network Localization/Bottleneck Off-Loading	8
2.4. Provisioning ALTO Maps	10
3. Deployment Considerations by ISPs	10
3.1. Requirement by ISPs	10
3.1.1. Requirement for Traffic Optimization	10
3.1.2. Other Requirements	11
3.2. Considerations for Different Types of ISPs	11
3.2.1. Very small ISPs with simple Network Structure	11
3.2.2. Large ISPs with a Fixed Network	12
3.2.3. ISPs with Mobile Network	13
4. Using ALTO for P2P	15
4.1. Using ALTO for Tracker-based Peer-to-Peer Applications	17
4.2. Expectations of ALTO	22
5. Using ALTO for CDNs	22
5.1. Request Routing using the Endpoint Cost Service	22
5.1.1. ALTO Topology Vs Network Topology	23
5.1.2. Topology Computation and ECS Delivery	23
5.1.3. Ranking Service	24
5.1.4. Ranking and Network Events	24
5.1.5. Caching and Lifetime	24
5.1.6. Redirection	25
5.1.7. Groups and Costs	25
6. Advanced Features	26
6.1. Cascading ALTO Servers	26
6.2. ALTO for IPv4 and IPv6	27
6.3. Monitoring ALTO	27

6.3.1.	Monitoring Metrics Definition	27
6.3.2.	Monitoring Data Sources	28
6.3.3.	Monitoring Structure	28
7.	Known Limitations of ALTO	29
7.1.	Limitations of Map-based Approaches	29
7.2.	Limitations of Non-Map-based Approaches	31
7.3.	General Challenges	31
8.	Extensions to the ALTO Protocol	32
8.1.	Host Group Descriptors	32
8.2.	Rating Criteria	33
8.2.1.	Distance-related Rating Criteria	33
8.2.2.	Charging-related Rating Criteria	34
8.2.3.	Performance-related Rating Criteria	34
8.2.4.	Inappropriate Rating Criteria	35
9.	API between ALTO Client and Application	35
10.	Security Considerations	35
10.1.	Information Leakage from the ALTO Server	36
10.2.	ALTO Server Access	36
10.3.	Faking ALTO Guidance	37
11.	Conclusion	37
12.	References	37
12.1.	Normative References	38
12.2.	Informative References	38
Appendix A.	Contributors List and Acknowledgments	39
Authors'	Addresses	39

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 [RFC6708].

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

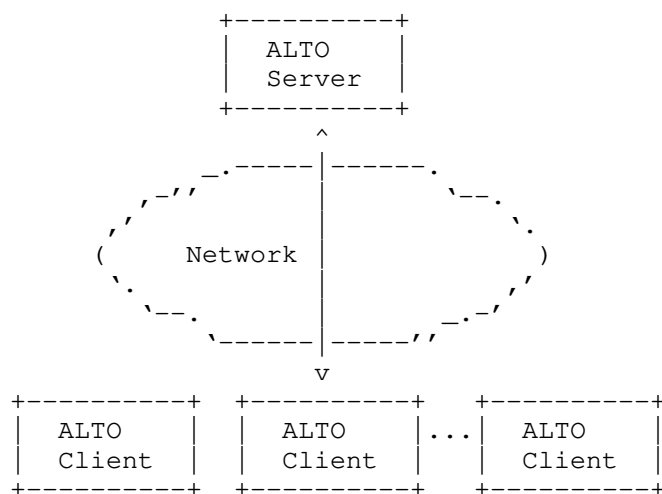
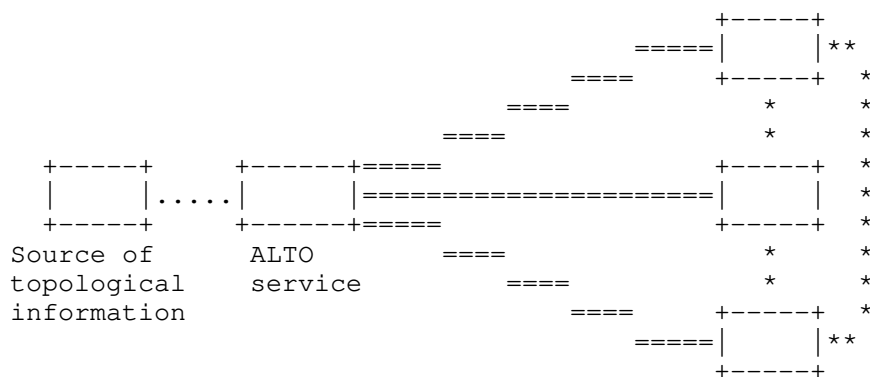


Figure 1: Baseline Deployment Scenario of the ALTO Protocol

2.1. Placement of ALTO Entities

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, or if the ALTO client is located on a resource directory, as shown in Figure 3.

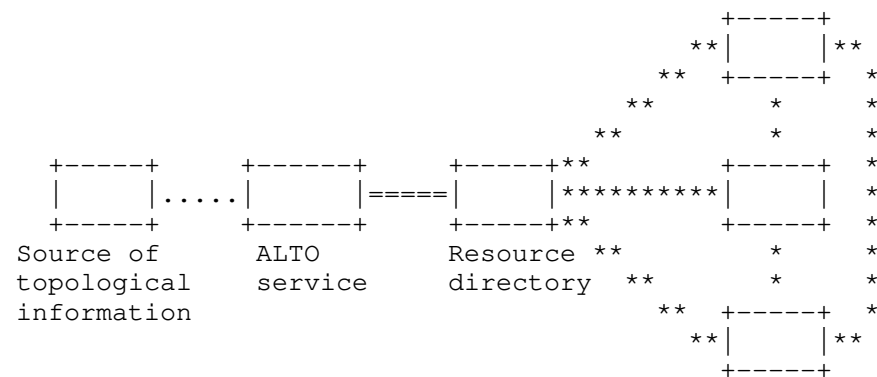


Legend:

=== ALTO client protocol
*** Application protocol
... Provisioning protocol

Figure 2: Overview of protocol interaction between ALTO elements without a resource directory

Figure 2 shows the operational model for applications that do not use a resource directory. An example would be a peer-to-peer file sharing application that does not use a tracker, such as edonky.



Legend:
=== ALTO client protocol
*** Application protocol
... Provisioning protocol

Figure 3: Overview of protocol interaction between ALTO elements with a resource directory

In Figure 3, a use case with a resource directory is illustrated, e.g., a tracker in peer-to-peer filesharing. Both deployment scenarios differ in the number of ALTO clients that access an ALTO service: If ALTO clients are implemented in a resource directory, ALTO servers are accessed by a limited and less dynamic set of clients, whereas in the general case any host in the Internet could be an ALTO client.

Using ALTO in CDNs may be similar to a resource directory [I-D.jenkins-alto-cdn-use-cases]. The ALTO server can also be queried by CDN entities to get a guidance about where the a particular client accessing data in the CDN is exactly located in the ISP's network.

2.2. Relationship between ALTO and Applications

ALTO is a general-purpose solution and it is intended to be used by a wide-range of applications. This implies that there are different possibilities where the ALTO entities are actually located, i.e., if the ALTO clients and the ALTO server are in the same ISP's domain, or if the clients and the ALTO server are managed/owned/located in different domains.

High-level differences between different ALTO deployments are:

1. Trust model: The deployment of ALTO can differ depending on whether ALTO client and ALTO server are operated within the same organization and/or network, or not. This changes a lot of constraints, because the trust model is very different. For instance, as discussed later in this memo, the level-of-detail of maps can depend on who the involved parties actually are.
2. User group: The main use case of ALTO is to provide guidance to any Internet application. However, an operator of an ALTO server could also decide to only offer guidance to a set of well-known ALTO clients, e. g., after authentication and authorization. In the peer-to-peer application use case, this could imply that only selected trackers are allowed to access the ALTO server. The security implications of using ALTO in closed groups differ a lot from the public Internet.
3. Destinations: In general, an ALTO server has to be able to provide guidance for all potential destinations. Yet, in practice a given ALTO client may only be interested in a subset of destinations, e. g., only in the network cost between a limited set of resource providers. For instance, CDN optimization may not need the full ALTO cost maps, because traffic between individual residential users is not in scope. This may imply that an ALTO server only has to provide the costs that matter for a given user, e. g., by customized maps.

The following sections enumerate different classes of use cases for ALTO, and they discuss the deployment implications of each of them.

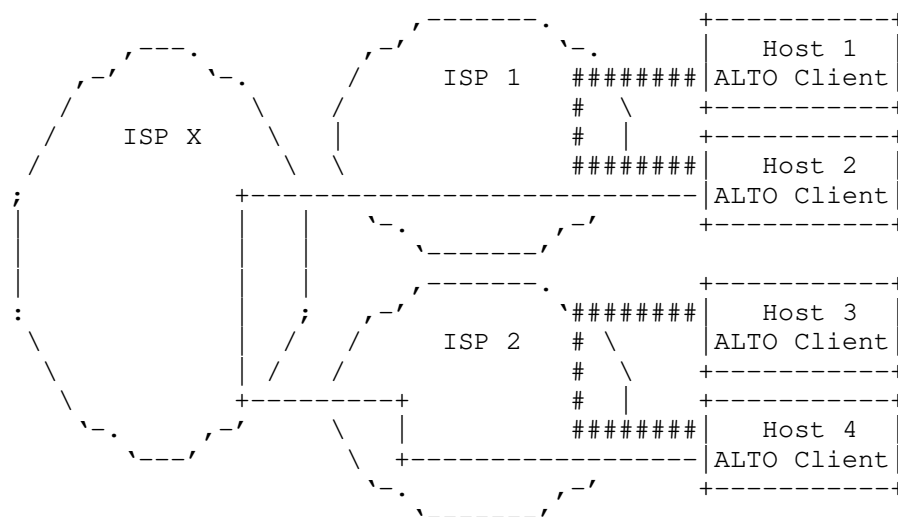
However, it must be emphasized that any application using ALTO must also work if no ALTO servers can be found or if no responses to ALTO queries are received, e.g., due to connectivity problems or overload situation (see also [RFC6708]).

2.3. Provided Guidance

ALTO gives guidance to applications on what IP addresses or IP prefixes are to be preferred according to the operator of the ALTO server. The ALTO protocol gives only the means to let the ALTO server operator to express its preference, whatever this preference is.

2.3.1. Keeping Traffic Local in Network

ALTO guidance can be used to let applications prefer other hosts within the same network operator's network instead of randomly connecting to other hosts that are located in another operator's network. Here, a network operator would always express to prefer hosts in its own network while hosts located outside its own network are to be avoided (i. e., they are undesired to be considered by the applications). Figure 4 shows such a scenario where hosts prefer hosts in the same network (e.g., Host 1 and Host 2 in ISP1 and Host 3 and Host 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. The network operator may have the desire to guide hosts in its own network to use hosts 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.

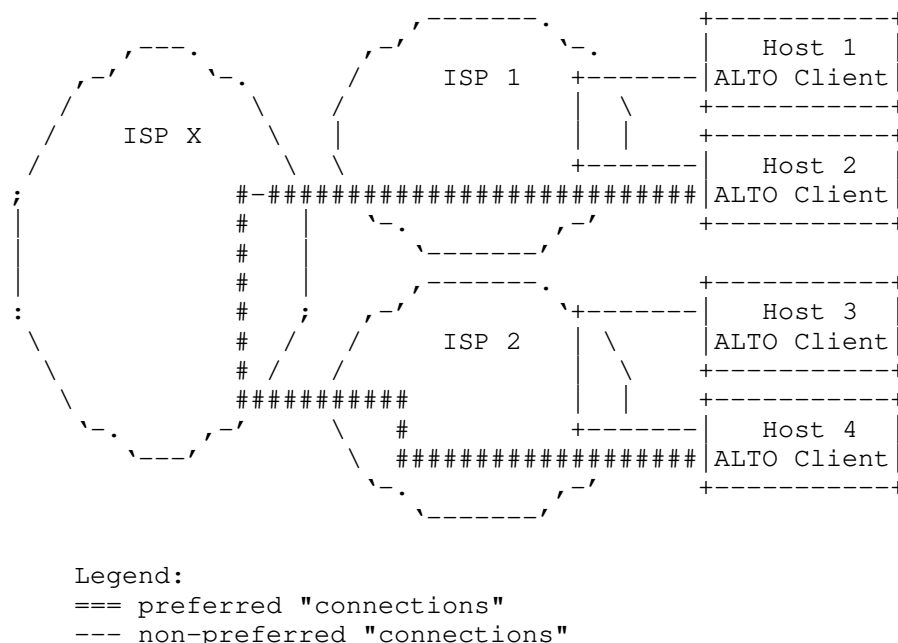


Figure 5: ALTO Traffic Network De-Localization

Figure 5 shows the result of such a guidance process where Host 2 prefers a connection with Host 4 instead of Host 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 hosts on which internal hosts are to be preferred. For instance, to guide hosts 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 Host 1 and Host 2 are located in Net 2 of ISP1 and connect via a low capacity link to the core (Net 1) of the same ISP1. Host 1 and Host 2 would both exchange their data with remote hosts, probably clogging the bottleneck link.

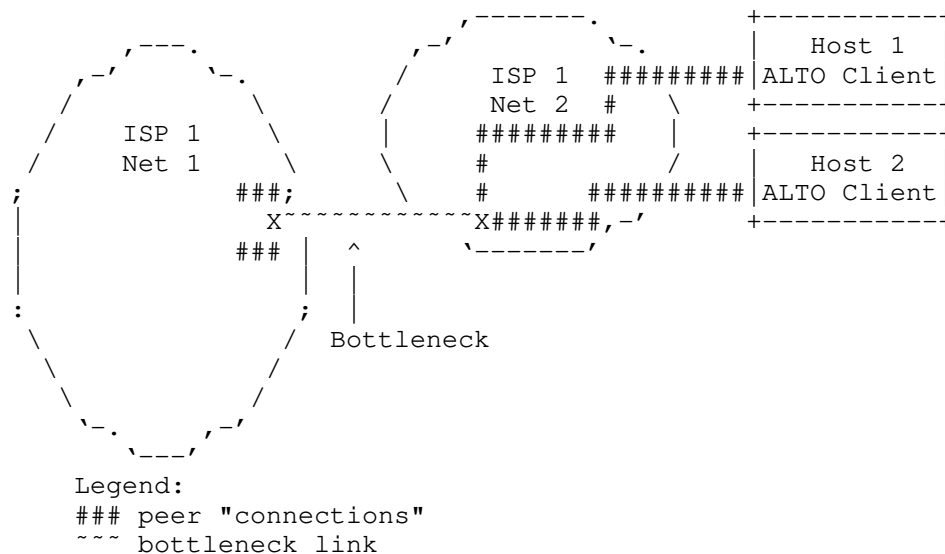
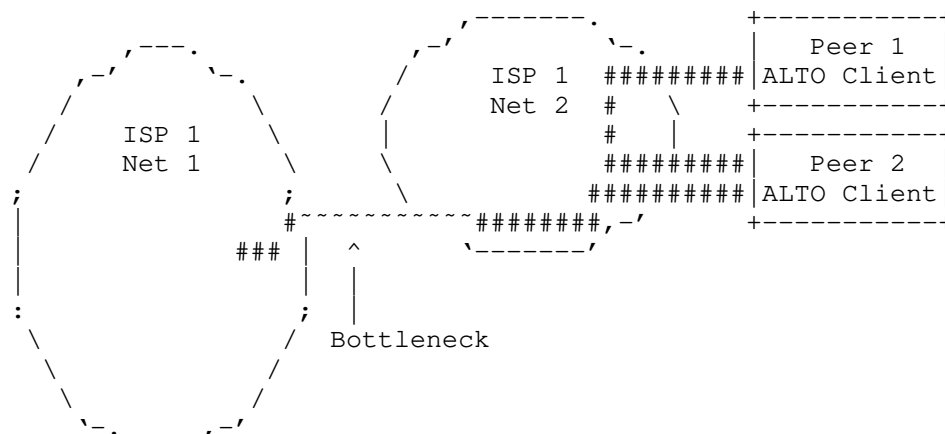


Figure 6: Without Intra-Network ALTO Traffic Localization

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



Legend:
peer "connections"
~~~ bottleneck link

Figure 7: With Intra-Network ALTO Traffic Localization

TBD: describe how maps would look like.

#### 2.4. Provisiong ALTO Maps

TBD: This section will describe how ALTO maps in the protocol can be populated before using them. The maps can significantly differ depending on the use case, the network architecture, and the trust relationship between ALTO server and ALTO client, etc.

### 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 by ISPs

##### 3.1.1. Requirement for Traffic Optimization

ALTO enables ISPs to perform traffic engineering by influencing application resouce selections. This can help to reduce inter-domain traffic. The 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 grouped into several network domains. The core network includes one or several backbone

networks, which are connected to multiple aggregation, metro, and access networks. If traffic can be limited to access networks, this decreases the usage of backbone and thus helps to save resources and costs.

Compared to fixed networks, mobile networks have some special characteristics, 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 hosts in the fixed network should avoid to retrieve data from hosts in the mobile networks, and hosts in the mobile networks should prefer the data retrieval from the hosts in the fixed networks.

### 3.1.2. Other Requirements

Providing ALTO guidance results in a win-win situation both for network providers and users of the ALTO information. Applications possibly get a better performance, while the the network provider has means to optimize the traffic engineering and thus its costs.

Still, ISPs may have other important requirements when deploying ALTO: In particular, an ISP may not be willing to expose sensitive operational details of its network. The topology abstraction of ALTO enables an ISP to expose the network topology at a desired granularity only.

## 3.2. Considerations for Different Types of 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:

$C1 < C2$

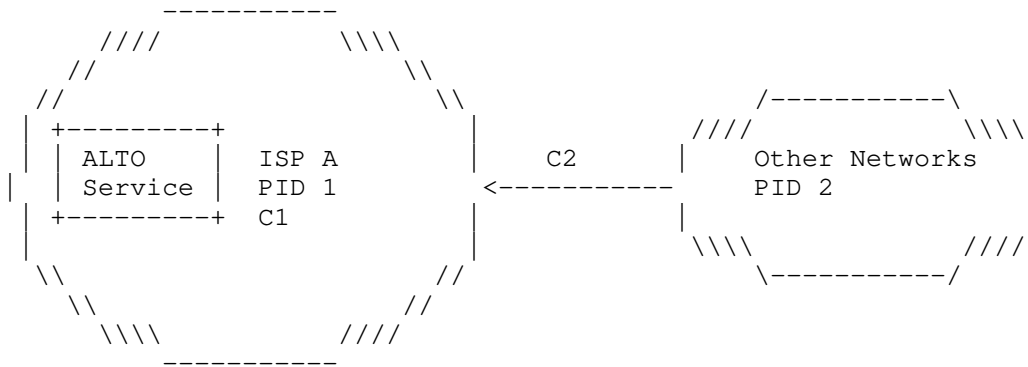


Figure 8: ALTO deployment in small ISPs

### 3.2.2. Large ISPs with a Fixed Network

For large ISPs with fixed network, 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].

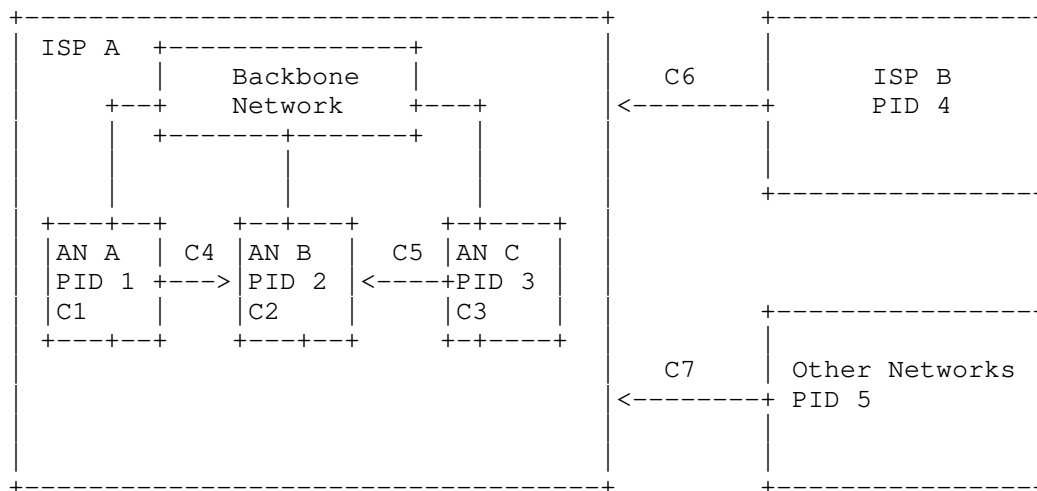


Figure 9: ALTO deployment in large ISPs with layered fixed network structures

### 3.2.3. ISPs with Mobile Network

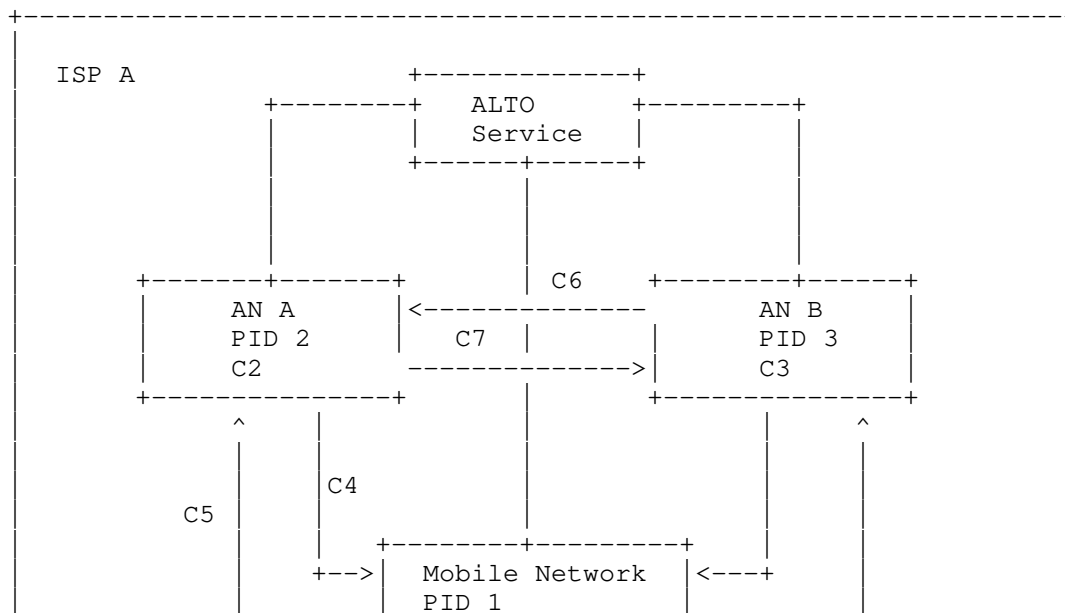
For ISPs with mobile network and fixed network, the traffic optimizing problems they focus will be optimizing the mobile traffic, except problems on last hop section. Wireless radio frequency resource is scarce and costly in mobile network. The requirement of traffic optimization in mobile network is 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: ( $C_4 < C_1$ ). 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: ( $C_2 < C_6$ ,  $C_5 = \text{maximum cost}$ ). This means that, to other optimization area, requiring data from mobile network should be avoided.



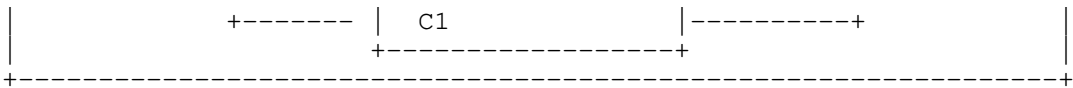


Figure 10: ALTO deployment in ISPs with mobile network

4. Using ALTO for P2P

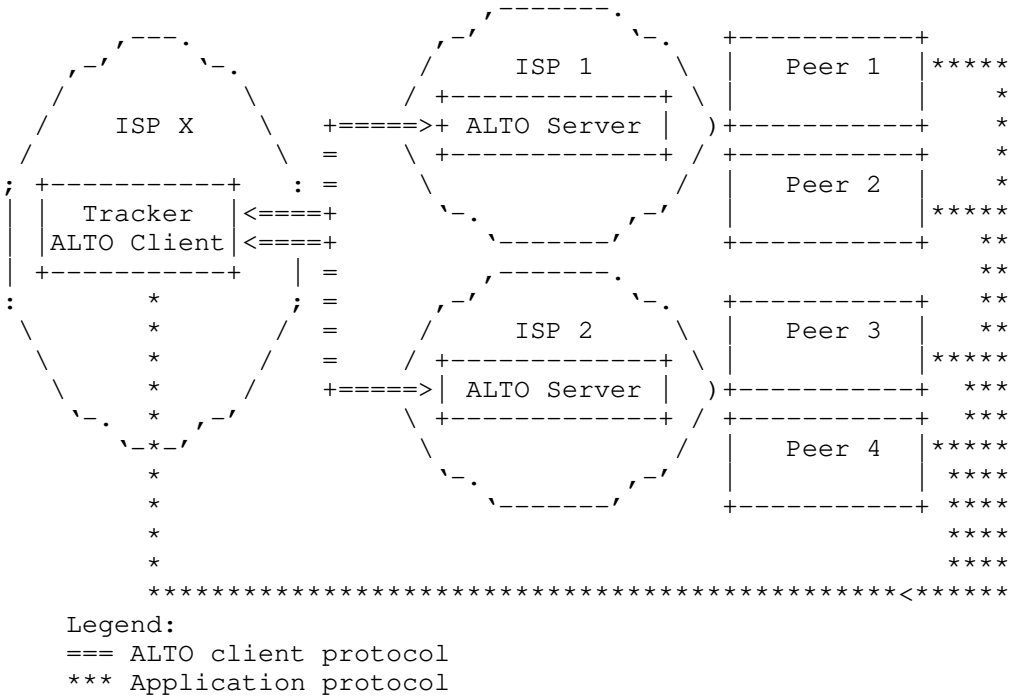


Figure 11: Global tracker accessing ALTO server at various ISPs

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. A tracker outside the network of the ISP is the typical use case. For instance, a tracker like Pirate Bay can serve Bittorrent peers world-wide. 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.ietf-alto-server-discovery]. 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. This is called peer exchange (PEX) in bittorrent, for instance.

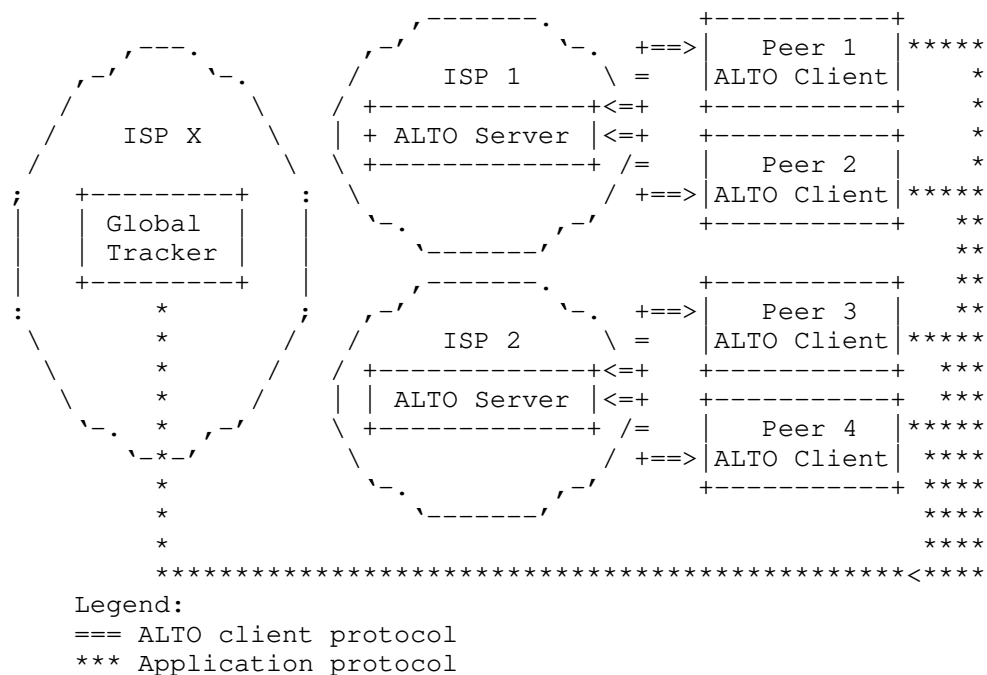
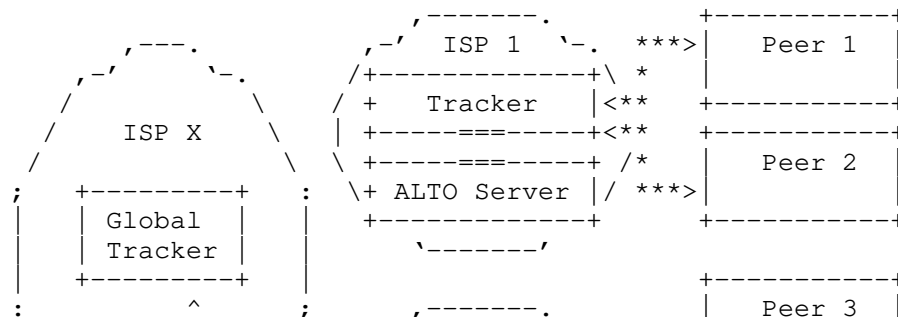


Figure 12: Global Tracker - Local ALTO Servers

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.





\*\*\* Application protocol

guidance are most beneficial in the initial phase after the resource consumer has decided to access a resource, as long as only few resource providers are known. Later, when the resource consumer has already exchanged some data with other peers and measured the transmission speed, the relative importance of ALTO may dwindle.

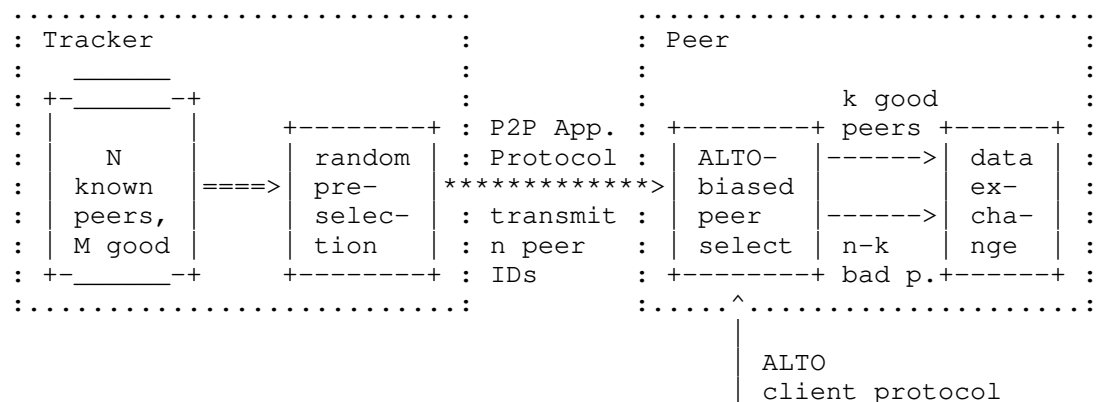
The ALTO protocol specification [I-D.ietf-alto-protocol] details how an ALTO client can query an ALTO server for guiding information and receive the corresponding replies. However, in the considered scenario of a tracker-based P2P application, there are two fundamentally different possibilities where to place the ALTO client:

1. ALTO client in the resource consumer ("peer")
2. ALTO client in the resource directory ("tracker")

In the following, both scenarios are compared in order to explain the need for third-party ALTO queries.

In the first scenario (see Figure 15), the resource consumer queries the resource directory for the desired resource (F1). The resource directory returns a list of potential resource providers without considering ALTO (F2). It is then the duty of the resource consumer to invoke ALTO (F3/F4), in order to solicit guidance regarding this list.

In the second scenario (see Figure 17), the resource directory has an embedded ALTO client, which we will refer to as RDAC in this document. After receiving a query for a given resource (F1) the resource directory invokes the RDAC to evaluate all resource providers it knows (F2/F3). Then it returns a, possibly shortened, list containing the "best" resource providers to the resource consumer (F4).



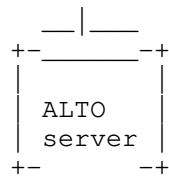


Figure 14: Tracker-based P2P Application with random peer preselection

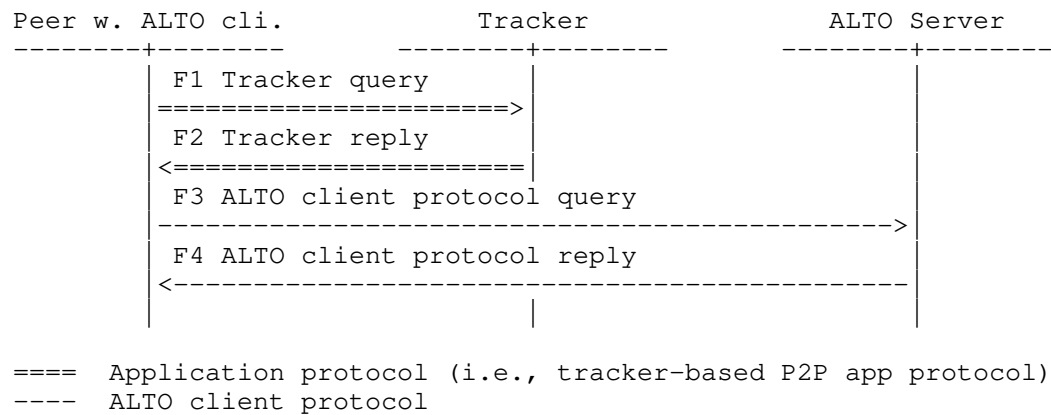


Figure 15: Basic message sequence chart for resource consumer-initiated ALTO query

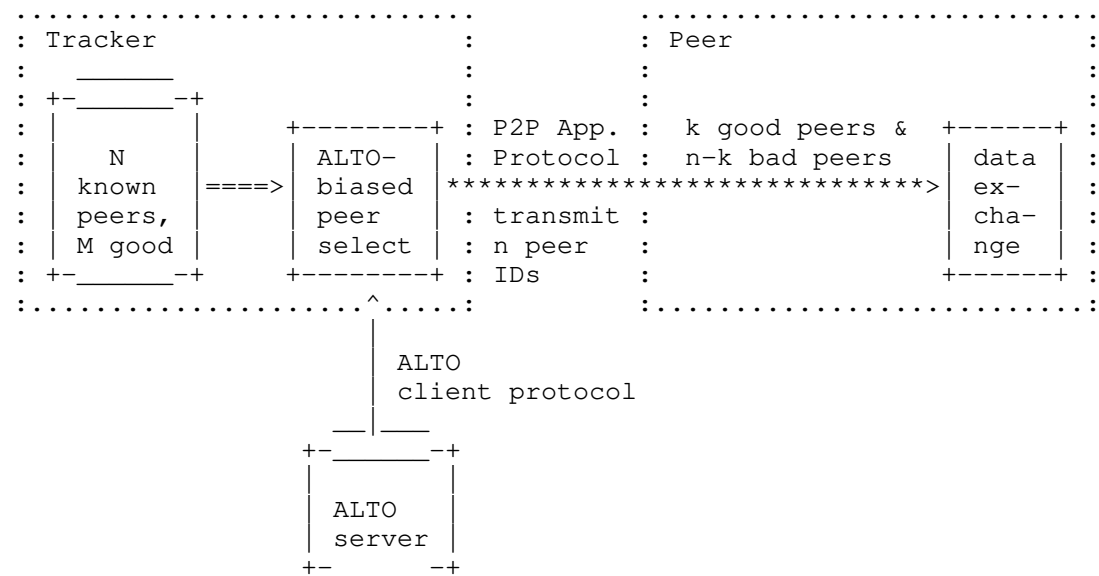
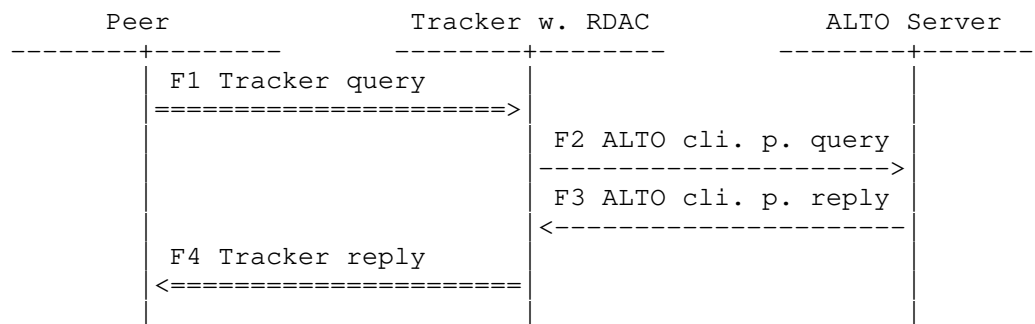


Figure 16: Tracker-based P2P Application with ALTO client in tracker



==== Application protocol (i.e., tracker-based P2P app protocol)  
 ---- ALTO client protocol

Figure 17: Basic message sequence chart for third-party ALTO query

Note: the message sequences depicted in Figure 15 and Figure 17 may occur both in the target-aware and the target-independent query mode (c.f. [RFC6708]). In the target-independent query mode no message exchange with the ALTO server might be needed after the tracker query, because the candidate resource providers could be evaluated using a locally cached "map", which has been retrieved from the ALTO server some time ago.

The problem with the first approach is, that while the resource directory might know thousands of peers taking part in a swarm, the list returned to the resource consumer is usually shortened for efficiency reasons. Therefore, the "best" (in the sense of ALTO) potential resource providers might not be contained in that list anymore, even before ALTO can consider them.

For illustration, consider a simple model of a swarm, in which all peers fall into one of only two categories: assume that there are "good" ("good" in the sense of ALTO's better-than-random peer selection, based on an arbitrary desired rating criterion) and "bad" peers only. Having more different categories makes the maths more complex but does not change anything to the basic outcome of this analysis. Assume that the swarm has a total number of  $N$  peers, out of which are  $M$  "good" and  $N-M$  "bad" peers, which are all known to the tracker. A new peer wants to join the swarm and therefore asks the tracker for a list of peers.

If, according to the first approach, the tracker randomly picks  $n$  peers from the  $N$  known peers, the result can be described with the hypergeometric distribution. The probability that the tracker reply contains exactly  $k$  "good" peers (and  $n-k$  "bad" peers) is:

$$P(X=k) = \frac{\frac{\binom{m}{k} \binom{N-m}{n-k}}{\binom{N}{n}}}{\frac{\binom{N}{n}}{\binom{N}{n}}}$$

$$\text{with } \frac{\binom{n}{k}}{k! (n-k)!} = \frac{n!}{k! (n-k)!} \quad \text{and} \quad n! = n * (n-1) * (n-2) * \dots * 1$$

The probability that the reply contains at most  $k$  "good" peers is:  
 $P(X \leq k) = P(X=0) + P(X=1) + \dots + P(X=k)$ .

For example, consider a swarm with  $N=10,000$  peers known to the tracker, out of which  $M=100$  are "good" peers. If the tracker randomly selects  $n=100$  peers, the formula yields for the reply:  $P(X=0)=36\%$ ,  $P(X \leq 4)=99\%$ . That is, with a probability of approx. 36% this list does not contain a single "good" peer, and with 99% probability there are only four or less of the "good" peers on the list. Processing this list with the guiding ALTO information will ensure that the few favorable peers are ranked to the top of the list; however, the benefit is rather limited as the number of favorable peers in the list is just too small.

Much better traffic optimization could be achieved if the tracker would evaluate all known peers using ALTO, and return a list of 100 peers afterwards. This list would then include a significantly higher fraction of "good" peers. (Note, that if the tracker returned "good" peers only, there might be a risk that the swarm might disconnect and split into several disjunct partitions. However, finding the right mix of ALTO-biased and random peer selection is out of the scope of this document.)

Therefore, from an overall optimization perspective, the second scenario with the ALTO client embedded in the resource directory is advantageous, because it is ensured that the addresses of the "best" resource providers are actually delivered to the resource consumer. An architectural implication of this insight is that the ALTO server discovery procedures must support third-party discovery. That is, as

the tracker issues ALTO queries on behalf of the peer which contacted the tracker, the tracker must be able to discover an ALTO server that can give guidance suitable for the that respective peer.

#### 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) [I-D.jenkins-alto-cdn-use-cases]. 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).

##### 5.1. 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 in order 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 obtained 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 interesting host and cache the results for later usage.

Maps Services and ECS deliver similar ALTO service by allowing the CDN 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.

#### 5.1.1. ALTO Topology Vs Network Topology

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 which level of granularity, it 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 (e.g.: There will be no leaking of OSPF/ISIS/BGP databases to ALTO clients).

#### 5.1.2. Topology Computation and ECS Delivery

ECS allows the CDN not to 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).

The ALTO server delivering ECS may integrate various information sources such as routing topology, policies, state and performance, geo-location, etc, and deliver the ranking service to the CDN upon request. The network topology information is controlled, managed by the ALTO server and the CDN benefits from ranking services in order to optimize application layer mechanisms used for content location

selection. This allows the ALTO server to enhance and modify the way the topology information sources are used and combined without requiring any update in the mechanisms the ECS is delivered and do not require any update process between ALTO and the CDN.

#### 5.1.3. Ranking Service

When a user request a given content, the CDN locates the content in one or more caches and executes a selection algorithms in order 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 and performance information in addition to other information sources (policy, geo-location, state and performance).

Once the ALTO server 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.

#### 5.1.4. Ranking and Network Events

ALTO server ranks addresses based on topology information it acquires from 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, and in the case the rankings are based on routing (IP/MPLS) topology, it is obvious that network events may impact the ranking computation. The scope of the ECS service delivered to a CDN 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.

#### 5.1.5. Caching and Lifetime

Each reply sent back by the ALTO server to the ALTO client running in the CDN 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 CDN 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's a few CDN component interacting with the ALTO server there are no scalability issues in maintaining state about clients in the ALTO server.

#### 5.1.6. Redirection

When ALTO server receives an ECS request, it may not have the most appropriate topology information in order to accurately determine the ranking. 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 rankings 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).

#### 5.1.7. 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 SP networks use in order to exchange information about their prefix reachability. BGP Community us an attribute used

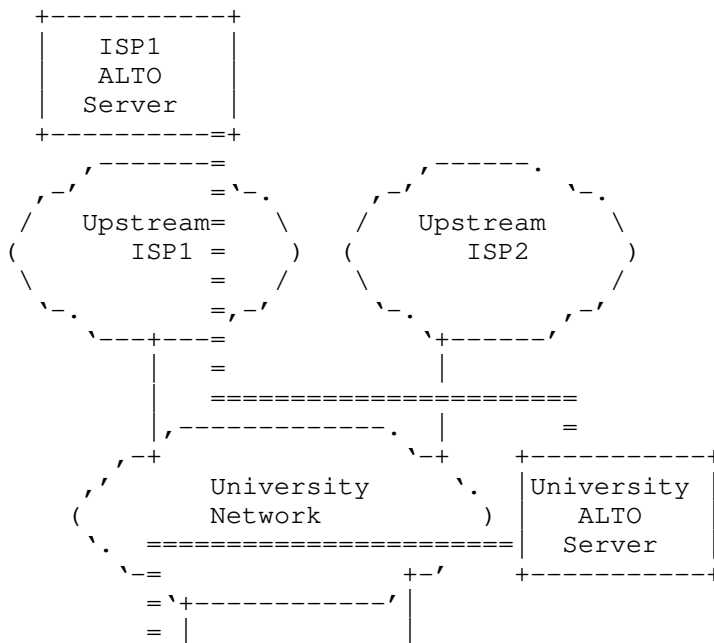
to tag a prefix so to group prefixes based on mostly any criteria (as an example, most SP networks originate BGP prefixes with communities identifying the Point of Presence (PoP) where the prefix has been originated).

The ALTO server may leverage the BGP information that is available in the SP 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 allows a dynamic computation of cost between groups.

## 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 18 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.



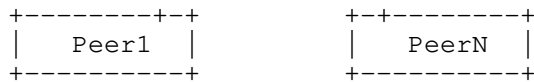


Figure 18: Cascaded ALTO Server

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

- o 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.
- o 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.

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

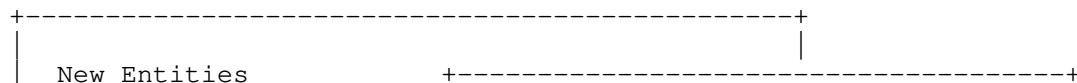
### 6.3.2. Monitoring Data Sources

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.



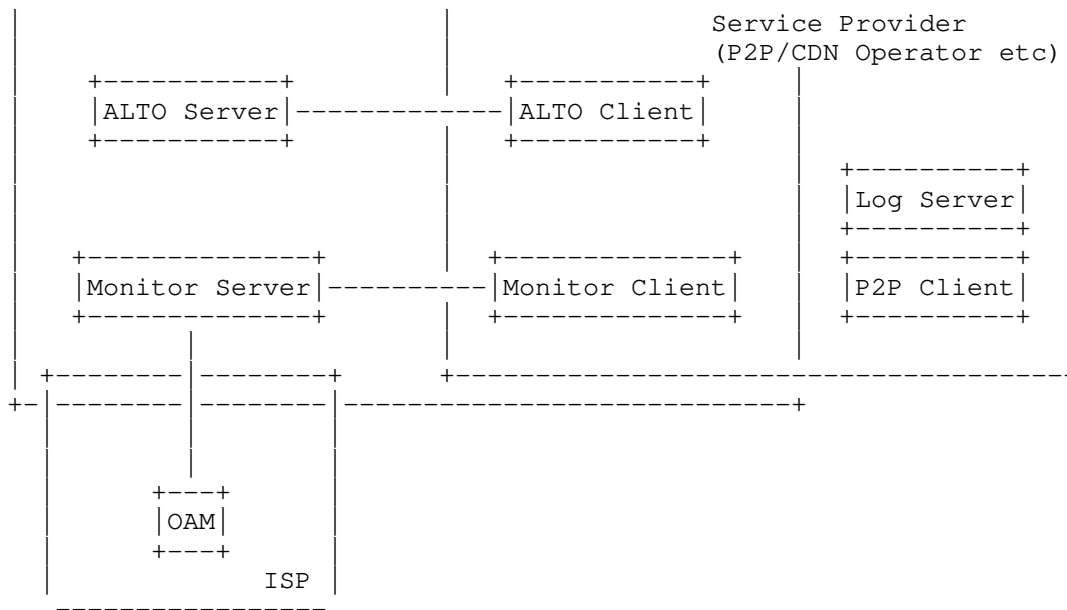


Figure 19: Monitoring Structure

## 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 router that serves a variety of access networks;
4. Network costs between IP prefixes may change depending on the ISP's routing and traffic engineering.

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.

For 4): The routing and traffic engineering inside an ISP network, as well as the peering with other autonomous systems, can change dynamically and affect the information exposed by an ALTO server. As a result, cost map and possibly also network maps can change.

## 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 frequently 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. The results of the ECS are also more difficult to cache than ALTO maps.

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. In case of peer-to-peer networks, 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. For other use cases of ALTO, in particular in more controlled environments, both approaches might be feasible and it is more an engineering tradeoff whether to use a map-based or query-based ALTO service.

## 8. Extensions to the ALTO Protocol

This section lists possible future 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:

- o Autonomous System (AS) number
- o 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

- o 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.
- o Absolute topological distance, expressed in the number of traversed autonomous systems (AS).
- o 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).
- o Absolute physical distance, based on knowledge of the approximate geolocation (continent, country) of an IP address.

### 8.2.2. Charging-related Rating Criteria

- o 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.

- o 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
- o 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.
- o 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
- o 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

- o the ALTO service is not an admission control system

- o the ALTO service may not know the instantaneous congestion status of the network
- o the ALTO service may not know all link bandwidths, i.e., where the bottleneck really is, and there may be shared bottlenecks
- o the ALTO service may not know whether the candidate peer itself is overloaded
- o the ALTO service may not know whether the candidate peer throttles the bandwidth it devotes for the considered application
- o 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:

- o 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.

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

#### 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 peer-to-peer applications, a potential deployment scenario is that an ALTO server 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).

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 not be 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.

There are also use cases where the access to the ALTO server has to be much more strictly controlled, i. e., where an authentication and authorization of the ALTO client to the server may be needed. For instance, in case of CDN optimization the provider of an ALTO service as well as potential users are possibly well-known. Only CDN entities may need ALTO access; access to the ALTO servers by residential users may neither be necessary nor be desired.

### 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 try to contact this IP address, possibly causing a Denial of Service (DoS) attack.

## 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-17 (work in progress), July 2013.
- [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-08 (work in progress), March 2013.
- [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-03 (work in progress), June 2012.
- [I-D.kamei-p2p-experiments-japan]  
Kamei, S., Momose, T., Inoue, T., and T. Nishitani, "ALTO-Like Activities and Experiments in P2P Network Experiment Council", draft-kamei-p2p-experiments-japan-09 (work in progress), October 2012.
- [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-04 (work in progress), March 2012.
- [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.

[RFC6708] Kiesel, S., Previdi, S., Stiernerling, M., Woundy, R., and  
Y. Yang, "Application-Layer Traffic Optimization (ALTO)  
Requirements", RFC 6708, September 2012.

#### Appendix A. Contributors List and Acknowledgments

This memo is the result of contributions made by several people, such  
as:

- o Xianghue Sun, Lee Kai, and Richard Yang contributed Section 3 and  
Section 6.3.
- o Stefano Previdi contributed Section 5 on "Using ALTO for  
CDNs".

Martin Stiernerling is partially supported by the CHANGE project (  
<http://www.change-project.eu>), a research project supported by the  
European Commission under its 7th Framework Program (contract no.  
257422). 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 CHANGE project or the European Commission.

#### Authors' Addresses

Martin Stiemerling (editor)  
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 (editor)  
University of Stuttgart, Computing Center  
Allmandring 30  
Stuttgart 70550  
Germany

Email: [ietf-alto@skiesel.de](mailto:ietf-alto@skiesel.de)

Stefano Previdi  
Cisco Systems, Inc.  
Via Del Serafico 200  
Rome 00191  
Italy

Email: [sprevidi@cisco.com](mailto:sprevidi@cisco.com)

Michael Scharf  
Alcatel-Lucent Bell Labs  
Lorenzstrasse 10  
Stuttgart 70435  
Germany

Email: [michael.scharf@alcatel-lucent.com](mailto:michael.scharf@alcatel-lucent.com)



ALTO WG  
Internet-Draft  
Intended status: Standards Track  
Expires: January 15, 2014

R. Alimi, Ed.  
Google  
R. Penno, Ed.  
Cisco Systems  
Y. Yang, Ed.  
Yale University  
July 14, 2013

ALTO Protocol  
draft-ietf-alto-protocol-17.txt

Abstract

Applications using the Internet already have access to some topology information of Internet Service Provider (ISP) networks. For example, views to Internet routing tables at looking glass servers are available and can be practically downloaded to many application clients. What is missing is knowledge of the underlying network topologies from the point of view of ISPs. In other words, what an ISP prefers in terms of traffic optimization -- and a way to distribute it.

The Application-Layer Traffic Optimization (ALTO) Service provides network information (e.g., basic network location structure and preferences of network paths) with the goal of modifying network resource consumption patterns while maintaining or improving application performance. The basic information of ALTO is based on abstract maps of a network. These maps provide a simplified view, yet enough information about a network for applications to effectively utilize them. Additional services are built on top of the maps.

This document describes a protocol implementing the ALTO Service. Although the ALTO Service would primarily be provided by ISPs, other entities such as content service providers could also operate an ALTO service. Applications that could use this service are those that have a choice to which end points to connect. 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 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 15, 2014.

#### Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                                                           |    |
|-----------------------------------------------------------|----|
| 1. Introduction . . . . .                                 | 7  |
| 1.1. Problem Statement . . . . .                          | 7  |
| 1.2. Design Overview . . . . .                            | 8  |
| 2. Terminology . . . . .                                  | 8  |
| 2.1. Endpoint . . . . .                                   | 8  |
| 2.2. Endpoint Address . . . . .                           | 8  |
| 2.3. Network Location . . . . .                           | 9  |
| 2.4. ALTO Information . . . . .                           | 9  |
| 2.5. ALTO Information Base . . . . .                      | 9  |
| 2.6. ALTO Service . . . . .                               | 9  |
| 3. Architecture . . . . .                                 | 9  |
| 3.1. ALTO Service and Protocol Scope . . . . .            | 9  |
| 3.2. ALTO Information Reuse and Redistribution . . . . .  | 11 |
| 4. ALTO Information Service Framework . . . . .           | 11 |
| 4.1. ALTO Information Services . . . . .                  | 12 |
| 4.1.1. Map Service . . . . .                              | 12 |
| 4.1.2. Map Filtering Service . . . . .                    | 12 |
| 4.1.3. Endpoint Property Service . . . . .                | 13 |
| 4.1.4. Endpoint Cost Service . . . . .                    | 13 |
| 5. Network Map . . . . .                                  | 13 |
| 5.1. Provider-defined Identifier (PID) . . . . .          | 13 |
| 5.2. Endpoint Addresses . . . . .                         | 14 |
| 5.2.1. IP Addresses . . . . .                             | 14 |
| 5.3. Example Network Map . . . . .                        | 15 |
| 6. Cost Map . . . . .                                     | 15 |
| 6.1. Cost Types . . . . .                                 | 16 |
| 6.1.1. Cost Metric . . . . .                              | 16 |
| 6.1.2. Cost Mode . . . . .                                | 16 |
| 6.2. Cost Map Structure . . . . .                         | 17 |
| 6.3. Network Map and Cost Map Dependency . . . . .        | 18 |
| 6.4. Cost Map Update . . . . .                            | 18 |
| 7. Endpoint Properties . . . . .                          | 18 |
| 7.1. Endpoint Property Type . . . . .                     | 19 |
| 7.1.1. Endpoint Property Type: pid . . . . .              | 19 |
| 8. Protocol Specification: General Processing . . . . .   | 19 |
| 8.1. Overall Design . . . . .                             | 19 |
| 8.2. Notation . . . . .                                   | 19 |
| 8.3. Basic Operation . . . . .                            | 20 |
| 8.3.1. Client Discovering Information Resources . . . . . | 20 |
| 8.3.2. Client Requesting Information Resources . . . . .  | 21 |
| 8.3.3. Server Responding to IR Request . . . . .          | 21 |
| 8.3.4. Client Handling Server Response . . . . .          | 22 |
| 8.3.5. Authentication and Encryption . . . . .            | 22 |
| 8.3.6. Information Refresh . . . . .                      | 23 |
| 8.3.7. HTTP Cookies . . . . .                             | 23 |
| 8.3.8. Parsing . . . . .                                  | 23 |

|         |                                                       |    |
|---------|-------------------------------------------------------|----|
| 8.4.    | Information Resource: Attributes                      | 23 |
| 8.4.1.  | Resource ID                                           | 23 |
| 8.4.2.  | Media Type                                            | 24 |
| 8.4.3.  | Capabilities                                          | 24 |
| 8.4.4.  | Accepts Input Parameters                              | 24 |
| 8.5.    | Information Resource Directory                        | 24 |
| 8.5.1.  | Media Type                                            | 24 |
| 8.5.2.  | Encoding                                              | 25 |
| 8.5.3.  | Example                                               | 26 |
| 8.5.4.  | Delegation and Multiple Choices                       | 29 |
| 8.5.5.  | Usage Considerations                                  | 31 |
| 8.6.    | Information Resource: Content Encoding                | 31 |
| 8.6.1.  | Meta Information                                      | 32 |
| 8.6.2.  | Data Information                                      | 32 |
| 8.6.3.  | Example                                               | 32 |
| 8.7.    | Protocol Errors                                       | 33 |
| 8.7.1.  | Media Type                                            | 33 |
| 8.7.2.  | Resource Format and Error Codes                       | 33 |
| 8.7.3.  | Overload Conditions and Server Unavailability         | 34 |
| 9.      | Protocol Specification: Basic ALTO Data Types         | 34 |
| 9.1.    | PID Name                                              | 35 |
| 9.2.    | Resource ID                                           | 35 |
| 9.3.    | Version Tag                                           | 35 |
| 9.4.    | Endpoints                                             | 36 |
| 9.4.1.  | Address Type                                          | 36 |
| 9.4.2.  | Endpoint Address                                      | 36 |
| 9.4.3.  | Endpoint Prefixes                                     | 37 |
| 9.4.4.  | Endpoint Address Group                                | 37 |
| 9.5.    | Cost Mode                                             | 38 |
| 9.6.    | Cost Metric                                           | 38 |
| 9.7.    | Cost Type                                             | 39 |
| 9.8.    | Endpoint Property                                     | 39 |
| 10.     | Protocol Specification: Service Information Resources | 39 |
| 10.1.   | Map Service                                           | 40 |
| 10.1.1. | Network Map                                           | 40 |
| 10.1.2. | Cost Map                                              | 42 |
| 10.2.   | Map Filtering Service                                 | 45 |
| 10.2.1. | Filtered Network Map                                  | 45 |
| 10.2.2. | Filtered Cost Map                                     | 48 |
| 10.3.   | Endpoint Property Service                             | 51 |
| 10.3.1. | Endpoint Property                                     | 51 |
| 10.4.   | Endpoint Cost Service                                 | 54 |
| 10.4.1. | Endpoint Cost                                         | 55 |
| 11.     | Use Cases                                             | 58 |
| 11.1.   | ALTO Client Embedded in P2P Tracker                   | 59 |
| 11.2.   | ALTO Client Embedded in P2P Client: Numerical Costs   | 60 |
| 11.3.   | ALTO Client Embedded in P2P Client: Ranking           | 61 |
| 12.     | Discussions                                           | 62 |

|                                                                                       |    |
|---------------------------------------------------------------------------------------|----|
| 12.1. Discovery . . . . .                                                             | 62 |
| 12.2. Hosts with Multiple Endpoint Addresses . . . . .                                | 63 |
| 12.3. Network Address Translation Considerations . . . . .                            | 63 |
| 12.4. Endpoint and Path Properties . . . . .                                          | 64 |
| 13. IANA Considerations . . . . .                                                     | 64 |
| 13.1. application/alto-* Media Types . . . . .                                        | 64 |
| 13.2. ALTO Cost Metric Registry . . . . .                                             | 65 |
| 13.3. ALTO Endpoint Property Type Registry . . . . .                                  | 67 |
| 13.4. ALTO Address Type Registry . . . . .                                            | 67 |
| 13.5. ALTO Error Code Registry . . . . .                                              | 68 |
| 14. Security Considerations . . . . .                                                 | 69 |
| 14.1. Authenticity and Integrity of ALTO Information . . . . .                        | 69 |
| 14.1.1. Risk Scenarios . . . . .                                                      | 69 |
| 14.1.2. Protection Strategies . . . . .                                               | 69 |
| 14.1.3. Limitations . . . . .                                                         | 70 |
| 14.2. Potential Undesirable Guidance from Authenticated ALTO<br>Information . . . . . | 70 |
| 14.2.1. Risk Scenarios . . . . .                                                      | 70 |
| 14.2.2. Protection Strategies . . . . .                                               | 70 |
| 14.3. Confidentiality of ALTO Information . . . . .                                   | 71 |
| 14.3.1. Risk Scenarios . . . . .                                                      | 71 |
| 14.3.2. Protection Strategies . . . . .                                               | 71 |
| 14.3.3. Limitations . . . . .                                                         | 72 |
| 14.4. Privacy for ALTO Users . . . . .                                                | 72 |
| 14.4.1. Risk Scenarios . . . . .                                                      | 72 |
| 14.4.2. Protection Strategies . . . . .                                               | 72 |
| 14.5. Availability of ALTO Service . . . . .                                          | 73 |
| 14.5.1. Risk Scenarios . . . . .                                                      | 73 |
| 14.5.2. Protection Strategies . . . . .                                               | 73 |
| 15. Manageability Considerations . . . . .                                            | 73 |
| 15.1. Operations . . . . .                                                            | 73 |
| 15.1.1. Installation and Initial Setup . . . . .                                      | 74 |
| 15.1.2. Migration Path . . . . .                                                      | 74 |
| 15.1.3. Requirements on Other Protocols and Functional<br>Components . . . . .        | 74 |
| 15.1.4. Impact and Observation on Network Operation . . . . .                         | 75 |
| 15.2. Management . . . . .                                                            | 75 |
| 15.2.1. Management Interoperability . . . . .                                         | 75 |
| 15.2.2. Management Information . . . . .                                              | 76 |
| 15.2.3. Fault Management . . . . .                                                    | 76 |
| 15.2.4. Configuration Management . . . . .                                            | 76 |
| 15.2.5. Performance Management . . . . .                                              | 76 |
| 15.2.6. Security Management . . . . .                                                 | 77 |
| 16. References . . . . .                                                              | 77 |
| 16.1. Normative References . . . . .                                                  | 77 |
| 16.2. Informative References . . . . .                                                | 78 |
| Appendix A. Acknowledgments . . . . .                                                 | 80 |
| Appendix B. Design History and Merged Proposals . . . . .                             | 81 |

|                               |    |
|-------------------------------|----|
| Appendix C. Authors . . . . . | 82 |
| Authors' Addresses . . . . .  | 82 |

## 1. Introduction

### 1.1. Problem Statement

This document defines the ALTO Protocol, which provides a solution for the problem stated in [RFC5693]. Specifically, in today's networks, network information such as network topologies, link availability, routing policies, and path costs are hidden from the application layer, and many applications benefited from such hiding of network complexity. However, new applications, such as application-layer overlays, can benefit from information about the underlying network infrastructure. In particular, these modern network applications can be adaptive, and hence become more network-efficient (e.g., reduce network resource consumption) and achieve better application performance (e.g., accelerated download rate), by leveraging network-provided information.

At a high level, the ALTO Protocol specified in this document is a unidirectional interface that allows a network to publish its network information such as network locations, costs between them at configurable granularities, and endhost properties to network applications. The information published by the ALTO protocol should benefit both the network and the applications (consumers of the information). Either the operator of the network or a third-party (e.g., an information aggregator) can retrieve or derive related information of the network and publish it using the ALTO Protocol. When a network provides information through the ALTO Protocol, we say that the network provides the ALTO Service.

To better understand the goal of the ALTO Protocol, we provide a short, non-normative overview of the benefits of ALTO to both networks and applications:

- o A network that provides an ALTO Service can achieve better utilization of its networking infrastructure. For example, by using ALTO as a tool to interact with applications, a network is able to provide network information to applications so that the applications can better manage traffic on more expensive or difficult to provision links such as long distance, transit or backup links. During the interaction, the network can choose to protect its sensitive and confidential network state information, by abstracting real metric values into non-real numerical scores or ordinal ranking.
- o An application that uses an ALTO Service can benefit from better knowledge of the network to avoid network bottlenecks. For example, an overlay application can use information provided by the ALTO Service to avoid selecting peers connected via high-delay

links (e.g., some intercontinental links). Using ALTO to initialize each node with promising ("better-than-expected") peers, an adaptive peer-to-peer overlay may achieve faster, better convergence.

## 1.2. Design Overview

The ALTO Protocol specified in this document meets the ALTO requirements specified in [RFC5693], and unifies multiple protocols previously designed with similar intentions. See Appendix A for a list of people and Appendix B for a list of proposals that have made significant contributions to this effort.

The ALTO Protocol uses a REST-ful design [Fielding-Thesis], and encodes its requests and responses using JSON [RFC4627]. These designs are chosen because of their flexibility and extensibility. In addition, these designs make it possible for ALTO to be deployed at scale by leveraging existing HTTP [RFC2616] implementations, infrastructures and deployment experience.

## 2. 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, Network Location, ALTO Information, ALTO Information Base, and ALTO Service.

### 2.1. Endpoint

An Endpoint is an application or host that is capable of communicating (sending and/or receiving messages) on a network.

An Endpoint is typically either a Resource Provider or Resource Consumer.

### 2.2. Endpoint Address

An Endpoint Address represents the communication address of an endpoint. Common forms of Endpoint Addresses include IP address, MAC address, overlay ID, and phone number. An Endpoint Address can be network-attachment based (e.g., IP address) or network-attachment agnostic (e.g., MAC address).



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

### 2.3. Network Location

Network Location is a generic term denoting a single Endpoint or a group of Endpoints. For instance, it can be a single IPv4 or IPv6 address, an IPv4 or IPv6 prefix, or a set of prefixes.

### 2.4. ALTO Information

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

### 2.5. 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.

### 2.6. ALTO Service

A network that provides ALTO Information through the ALTO protocol is said to provide the ALTO Service.

## 3. Architecture

We now define the ALTO architecture and the ALTO Protocol's place in the overall architecture.

### 3.1. ALTO Service and Protocol Scope

Each network region in the global Internet can provide its ALTO Service, which conveys network information from the perspective of that network region. A network region in this context can be an Autonomous System (AS), an ISP, a region smaller than an AS or ISP, or a set of ISPs. The specific network region that an ALTO Service represents will depend on the ALTO deployment scenario and ALTO service discovery mechanism.

Specifically, the ALTO Service of a network region defines network Endpoints (and aggregations thereof) and generic costs amongst them from the region's perspective. The network Endpoints may include all Endpoints in the global Internet. Hence, we say that the network information provided by the ALTO Service of a network region represents the "my-Internet View" of the network region.

To better understand the ALTO Service and the role of the ALTO Protocol, we show in Figure 1 the overall ALTO 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.

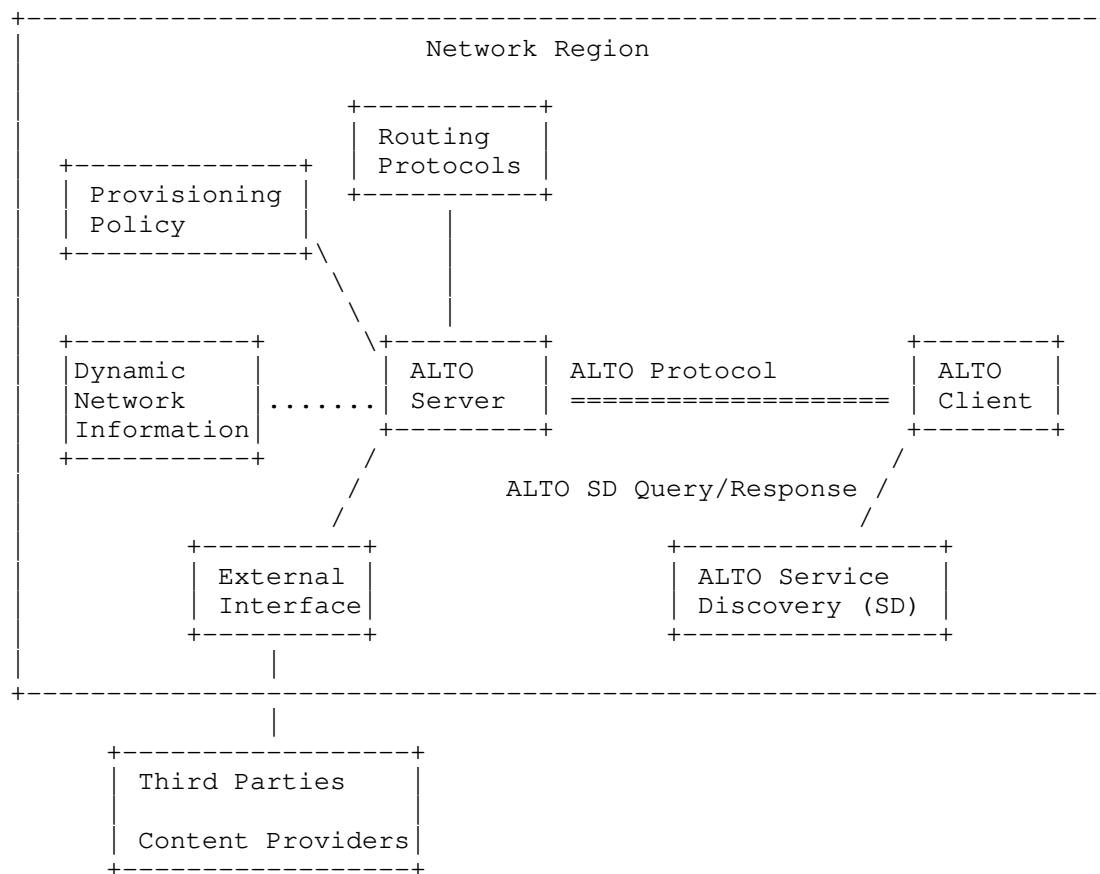


Figure 1: Basic ALTO Architecture.

Figure 1 illustrates that the ALTO Information provided by an ALTO Server may be influenced (at the service provider's discretion) by other systems. In particular, the ALTO Server can aggregate information from multiple systems to provide an abstract and unified

view that can be more useful to applications. Examples of other systems 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 are outside the scope of this specification. Recall that while the ALTO Protocol may convey dynamic network information, it is not intended to replace near-real-time congestion control protocols.

It may also be possible for an ALTO Server 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.

### 3.2. ALTO Information Reuse and Redistribution

ALTO Information may be useful to a large number of applications and users. At the same time, distributing ALTO Information must be efficient and not become a bottleneck.

The design of the ALTO Protocol allows integration with the existing HTTP caching infrastructure to redistribute ALTO Information. If caching or redistribution is used, the response message to an ALTO Client may be returned from a third-party.

Application-dependent mechanisms, such as P2P DHTs or P2P file-sharing, may be used to cache and redistribute ALTO Information. This document does not define particular mechanisms for such redistribution.

Additional protocol mechanisms (e.g., expiration times and digital signatures for returned ALTO information) are left for future investigation.

## 4. ALTO Information Service Framework

The ALTO Protocol conveys network information through services, where each service defines a set of related functionalities. An ALTO Client can query each service individually. All of the services defined in ALTO are said to form the ALTO service framework and are provided through a common transport protocol, messaging structure and encoding, and transaction model. Functionalities offered in different services can overlap.

In this document, we focus on achieving the goals of conveying (1)

Network Locations, which denote the locations of Endpoints at a network, (2) provider-defined costs for paths between pairs of Network Locations, and (3) network related properties of endhosts. We achieve the goals by defining the Map Service, which provides the core ALTO information to clients, and three additional services: the Map Filtering Service, Endpoint Property Service, and Endpoint Cost Service. Additional services can be defined in companion documents. Below we give an overview of the services. Details of the services will be presented in the following sections.

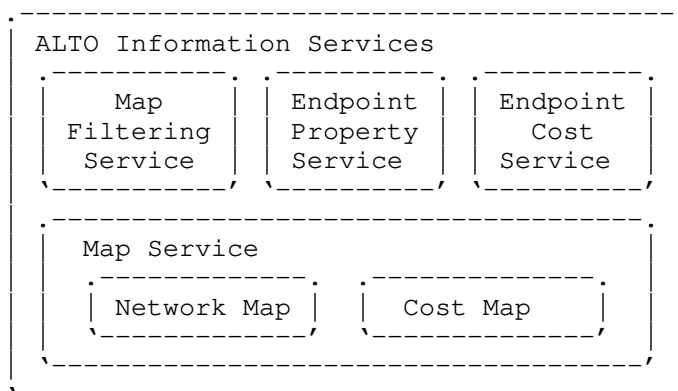


Figure 2: ALTO Service Framework.

#### 4.1. ALTO Information Services

##### 4.1.1. 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 5) provides the full set of Network Location groupings defined by the ALTO Server and the Endpoints contained within each grouping. The Cost Map (see Section 6) 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.

##### 4.1.2. Map Filtering Service

Resource constrained ALTO Clients may benefit from filtering of query results at the ALTO Server. This avoids that an ALTO Client first spends network bandwidth and CPU cycles to collect results and then performs client-side filtering. The Map Filtering Service allows

ALTO Clients to query an ALTO Server on Network Map and Cost Map based on additional parameters.

#### 4.1.3. Endpoint Property Service

This service allows ALTO Clients to look up properties for individual Endpoints. An example property of an Endpoint is its Network Location (i.e., its grouping defined by the ALTO Server). Another example property is its connectivity type such as ADSL (Asymmetric Digital Subscriber Line), Cable, or FTTH (Fiber To The Home).

#### 4.1.4. 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.

### 5. Network Map

An ALTO Network Map defines a grouping of network endpoints. In this document, we use Network Map to refer to the syntax and semantics of how an ALTO Server distributes the grouping. This document does not discuss the internal representation of this data structure within the ALTO Server.

The definition of Network Map is based on the observation that in reality, many endpoints are close by to one another in terms of network connectivity. By treating a group of close-by endpoints together as a single entity, an ALTO Server indicates aggregation of these endpoints due to their proximity. This aggregation can also lead to greater scalability without losing critical information when conveying other network information (e.g., when defining Cost Map).

#### 5.1. Provider-defined Identifier (PID)

One issue is that 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 Point of Presence (PoP) may be considered close.

ALTO introduces provider-defined Network Location identifiers called Provider-defined Identifiers (PIDs) to provide 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. Specifically, a PID is a US-ASCII string of type

PIDName (see Section 9.1) and its associated set of Endpoint Addresses. As we discussed above, there can be many different ways of grouping the endpoints and assigning PIDs. For example, a PID may denote a subnet, a set of subnets, a metropolitan area, a PoP, an autonomous system, or a set of autonomous systems.

A key use case of PIDs is to specify network preferences (costs) between PIDs instead of individual endpoints. This allows cost information to be more compactly represented and updated at a faster time scale than the network aggregations themselves. 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 endpoints within a PoP into a single PID in the Network Map. The cost may be encoded to indicate that Network Locations within the same PID are preferred; for example,  $\text{cost}(\text{PID}_i, \text{PID}_i) == c$  and  $\text{cost}(\text{PID}_i, \text{PID}_j) > c$  for  $i \neq j$ . Section 6 provides further details on using PIDs to represent costs in an ALTO Cost Map.

## 5.2. Endpoint Addresses

The endpoints aggregated into a PID are denoted by endpoint addresses. There are many types of addresses, such as IP addresses, MAC addresses, or overlay IDs. This specification only considers IP addresses.

### 5.2.1. IP Addresses

When either an ALTO Client or an 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 with the shortest enclosing prefix of the addresses provided in the map. For a map with full IPv4 reachability, this would mean including the 0.0.0.0/0 prefix in a PID; for full IPv6 reachability, this would be the ::/0 prefix.

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.

### 5.3. Example Network Map

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

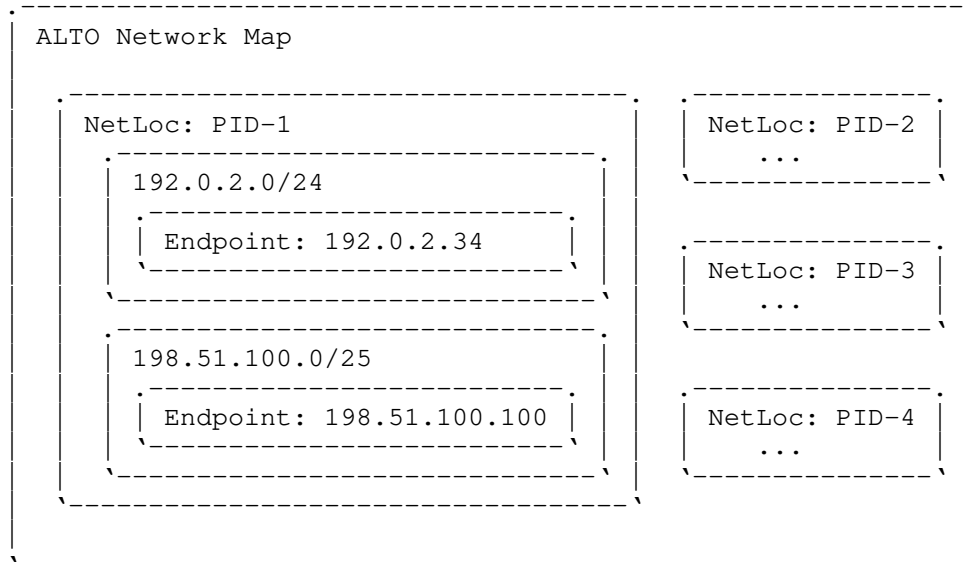


Figure 3: Example Network Map.

## 6. 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 defined by PIDs. Each Path Cost is the end-to-end cost when a unit of traffic goes from the source to the destination.

As cost is directional from the source to the destination, an application, when using ALTO Information, may independently determine how the Resource Consumer and Resource Provider are designated as the source or destination in an ALTO query, and hence how to utilize the Path Cost provided by ALTO Information. For example, if the cost is expected to be correlated with throughput, a typical application concerned with bulk data retrieval may use the Resource Provider as the source, and Resource Consumer as the destination.

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.

## 6.1. Cost Types

Path Costs have attributes:

- o Metric: identifies what the costs represent;
- o Mode: identifies how the costs should be interpreted.

The combination of a metric and a mode defines a Cost Type. Certain queries for Cost Maps allow the ALTO Client to indicate the desired Cost Type.

### 6.1.1. Cost Metric

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

Cost metrics are indicated in protocol messages as strings.

#### 6.1.1.1. Cost Metric: routingcost

An ALTO Server MUST offer the 'routingcost' Cost Metric.

This Cost Metric conveys a generic measure for the cost of routing traffic from a source to a destination. A lower value indicates 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.

### 6.1.2. 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 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' modes. An ALTO Client SHOULD be cognizant of operations when a desired Cost Mode is not supported. For example, an ALTO Client desiring numerical costs may adjust its behaviors if only the ordinal Cost Mode is available. Alternatively, an ALTO Client desiring ordinal costs may construct ordinal costs from retrieved numerical values, if only the numerical Cost Mode is available.

#### 6.1.2.1. 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. normalization or computing ratios for weighted load-balancing) on the returned costs. The values are floating-point numbers.

#### 6.1.2.2. Cost Mode: ordinal

This Cost Mode is indicated by the string 'ordinal'. This mode indicates that the costs values in a Cost Map are a ranking (relative to all other values in a Cost Map), with a lower value indicating a higher preference. The values are non-negative integers. Ordinal cost values in a Cost Map need not be unique nor contiguous. In particular, it is possible that two entries in a map 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 costs known to the ALTO Server.

### 6.2. 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 of the  $m \times n$  communicating pairs (i.e., Src\_1  $\rightarrow$  Dst\_1, ..., Src\_1  $\rightarrow$  Dst\_n, ..., Src\_m  $\rightarrow$  Dst\_1, ..., Src\_m  $\rightarrow$  Dst\_n). If the ALTO Server does not define a Path Cost for a particular pair, it may be omitted. 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  $m \times n$  entries.

### 6.3. 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 a tuple of (1) an ID for the resource (e.g., a Network Map), and (2) a tag (an opaque string) associated with the version of that resource. A Network Map distributed by an ALTO Server includes its Version Tag. A Cost Map referring to PIDs also includes Version Tag for the Network Map on which it is based.

Two Network Maps are the same if they have the same Version Tag. Whenever the content of the Network Map maintained by an ALTO Server changes, tag MUST also be changed. Possibilities of setting the tag component include the last-modified timestamp for the Network Map, or a hash of its contents, where the collision probability is considered zero in practical deployment scenarios.

### 6.4. Cost Map Update

An ALTO Server can update a Cost Map at any time. Hence, the same Cost Map retrieved from the same ALTO Server but from different requests can be inconsistent.

## 7. Endpoint Properties

An endpoint property defines a network-aware property of an endpoint.

### 7.1. Endpoint Property Type

For each endpoint and an endpoint property type, there can be a value for the property. The type of an Endpoint property is indicated in protocol messages as a string. The value depends on the specific property. For example, for a property such as whether an endpoint is metered, the value is a true or false value.

#### 7.1.1. Endpoint Property Type: pid

An ALTO Server MUST define the 'pid' Endpoint Property Type, which provides the PID of an endpoint. Since the PID of an endpoint depends on the Network Map, the Version Tag of the full (unfiltered) Network Map used to return the pid property MUST be included.

## 8. Protocol Specification: General Processing

This section first specifies general client and server processing. The details of specific services will be covered in the following sections.

### 8.1. Overall 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 information resources, which are distinguished by their media types [RFC2046]. An ALTO Client may construct an HTTP request for a particular information resource (including any parameters, if necessary), and an ALTO Server returns the requested information resource in an HTTP response.
- o Information Resource Directory (IRD): An ALTO Server provides to ALTO Clients a list of available information resources and the URI at which each is provided. This document refers to this list as the Information Resource Directory. ALTO Clients consult the directory to determine the services provided by an ALTO Server.

### 8.2. Notation

This document uses 'JSONString', 'JSONNumber', 'JSONBool' to indicate the JSON string, number, and boolean types, respectively. The type 'JSONValue' indicates a JSON value, as specified in Section 2.1 of [RFC4627].

We use an adaptation of the C-style struct notation to define the

members (names/values) of JSON objects. An optional member is enclosed by [ ], and an array is indicated by two numbers in angle brackets, <m..n>, where m indicates the minimal number of values, and n is the maximum. When we write \* for n, it means no upper bound. In the definitions, the JSON names of the members are case sensitive.

For example, the definition below defines a new type Type4, with three members named "name1", "name2", and "name3" respectively. The member named "name3" is optional, and the member named "name2" is an array of at least one value.

```
object {  
  Type1  name1;  
  Type2  name2<1..*>;  
  [Type3 name3;]  
} Type4;
```

We also define dictionary maps (or maps for short) from strings to JSON values. For example, the definition below defines a Type3 object as a map. Type1 must be defined as string, and Type2 can be any type.

```
object-map {  
  Type1  -> Type2;  
} Type3;
```

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

### 8.3. 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].

#### 8.3.1. Client Discovering Information Resources

To discover available Information Resources, an ALTO Client requests 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 8.5 for a detailed specification.

#### 8.3.2. Client 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 ALTO Client MUST use the HTTP POST method to send the parameters.

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.

#### 8.3.3. Server Responding to IR Request

Upon receiving a request for an Information Resource that the ALTO Server can provide, the ALTO Server MUST return the requested Information Resource. In other cases, to be more informative ([I-D.ietf-httpbis-p2-semantics]), the ALTO Server MAY provide the ALTO Client with an Information Resource Directory indicating how to reach the desired information resource, or return an ALTO error object; see Section 8.7 for more details on ALTO error handling.

It is possible for an ALTO Server to leverage caching HTTP intermediaries to respond to both GET and POST requests by including explicit freshness information (see Section 14 of [RFC2616]). Caching of POST requests is not widely implemented by HTTP intermediaries, however an alternative approach is for an ALTO Server, in response to POST requests, to return an HTTP 303 status code ("See Other") indicating to the ALTO Client that the resulting Information Resource is available via a GET request to an alternate URL. HTTP intermediaries that do not support caching of POST requests could then cache the response to the GET request from the ALTO Client following the alternate URL in the 303 response if the response to the subsequent GET request contains explicit freshness information.

The ALTO Server MUST indicate the type of its response using a media type (i.e., the Content-Type HTTP header of the response).

#### 8.3.4. Client Handling Server Response

##### 8.3.4.1. 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.

##### 8.3.4.2. Handling Server Response and IRD

After receiving an Information Resource Directory, the Client can consult it to determine if any of the offered URIs contain the desired Information Resource. However, an ALTO Client MUST NOT assume that the media type returned by the ALTO Server for a request to a URI is the media type advertised in the IRD or specified in its request (i.e., the client must still check the Content-Type header). The expectation is that the media type returned should normally be the media type advertised and requested, but in some cases it may legitimately not be so.

In particular, it is possible for an ALTO Client to receive an Information Resource Directory from an ALTO Server as a response to its request for a specific Information Resource. In this case, the ALTO Client may ignore the response or still parse the response. To indicate that an ALTO Client will always check if a response is an Information Resource Directory, the ALTO Client can indicate in the "Accept" header of a HTTP request that it can accept Information Resource Directory; see Section 8.5 for the media type.

##### 8.3.4.3. Handling Error Conditions

If an ALTO Client does not successfully receive a desired Information Resource from a particular ALTO Server (i.e., server response indicates error or there is no response), the Client 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, when used in a peer-to-peer system).

#### 8.3.5. Authentication and Encryption

When server and/or client authentication, encryption, and/or integrity protection are required, an ALTO Server MUST support SSL/TLS [RFC5246] as a mechanism. For cases such as a public ALTO

service or deployment scenarios where there is an implicit trust relationship between the client and the server and the network infrastructure connecting them is secure, SSL/TLS may not be necessary. See [RFC6125] for considerations regarding verification of server identity.

#### 8.3.6. Information Refresh

An ALTO Client MAY determine the frequency at which ALTO Information is refreshed based information made available via HTTP.

#### 8.3.7. HTTP Cookies

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

#### 8.3.8. 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 unknown fields when processing ALTO messages.

### 8.4. Information Resource: Attributes

An Information Resource 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, including its data format, its capabilities, and its accepted input parameters. These attributes are published by an ALTO Server in its Information Resource Directory.

#### 8.4.1. Resource ID

Each Information Resource MUST be given a unique ID. The ID MUST be unique amongst all resources offered by the ALTO Server, including those defined in Information Resource Directories linked from this ALTO Server. The ID SHOULD remain stable even when the data provided by that resource changes. IDs SHOULD NOT be re-used for different resources over time. For example, even though the number of PIDs in a Network Map may be adjusted, its Resource ID should remain the same. Similarly, if the entries in a Cost Map are updated, its Resource ID should remain the same.

#### 8.4.2. Media Type

ALTO uses Media Type [RFC2046] to uniquely indicate the data format used to encode the content to be transmitted between an ALTO Server and an ALTO Client in the HTTP entity body.

#### 8.4.3. Capabilities

The Capabilities associated with an Information Resource announced by an ALTO Server indicates specific capabilities that the server can provide. For example, if an ALTO Server allows an ALTO Client to specify cost constraints when the Client requests a Cost Map Information Resource, the Server advertises the cost-constraints capability for its Cost Map Information Resource.

#### 8.4.4. Accepts Input Parameters

An ALTO Server may allow an ALTO Client to supply input parameters when requesting certain Information Resources. The associated accepts attribute of an Information Resource is a Media Type, which indicates how the Client specifies the input parameters as contained in the entity body of the HTTP POST request.

### 8.5. Information Resource Directory

An ALTO Server uses Information Resource Directory to publish available Information Resources and their aforementioned attributes. 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]. For recommendations on how the URI may look like, see [I-D.ietf-alto-server-discovery].

#### 8.5.1. Media Type

The media type to indicate an information directory is "application/alto-directory+json".



## 8.5.2. Encoding

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

```
object {
  IRDMeta      meta;
  IRDResourceEntry resources<1..*>;
} InfoResourceDirectory;

object-map {
  JSONString -> JSONValue;
} IRDMeta;

object {
  ResourceID      id;
  JSONString      uri;
  JSONString      media-type;
  [JSONString     accepts;]
  [Capabilities    capabilities;]
  [ResourceID      uses<0..*>;]
} IRDResourceEntry;

object {
  ...
} Capabilities;
```

where the "meta" member provides definitions related with the IRD itself, or can be used when defining multiple individual Information resources;

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 specifies:

uri A URI at which the ALTO Server provides one or more Information Resources, or an Information Resource Directory indicating additional Information Resources. URIs can be relative and MUST be resolved according to Section 5 of [RFC3986].

**media-type** The media type of Information Resource (see Section 8.4.2) available via GET or POST requests to the corresponding URI or "application/alto-directory+json", which indicates that the response for a request to the URI will be an Information Resource Directory for URIs discoverable via the URI.

**accepts** The media type of input parameters (see Section 8.4.4) accepted by POST requests to the corresponding URI. If this member is not present, it MUST be assumed to be empty.

**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 object. 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 documented in this specification or an extension document describing the capability.

**uses** A list of Resource IDs corresponding to resources on which this resource directly depends. An ALTO Server SHOULD include in this list any resource that the ALTO Client would need to retrieve in order to interpret the contents of this resource. For example, a Cost Map resource should include in this list the Network Map on which it depends. Likewise, an Endpoint Property resource providing the 'pid' property should indicate the Network Map on which it is based. ALTO Clients may wish to consult this list in order to pre-fetch necessary resources.

If an entry has an empty list for "accepts", then the corresponding URI MUST support GET requests. If an entry has a non-empty "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.

#### 8.5.3. Example

The following is an example Information Resource Directory returned by an ALTO Server.

```
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: TBA

Content-Type: application/alto-directory+json

```
{
  "meta" : {
    "cost-types": {
      "num-routing": {"cost-mode" : "numerical",
                     "cost-metric": "routingcost",
                     "description": "My default"},
      "num-hop":    {"cost-mode" : "numerical",
                     "cost-metric": "hopcount"},
      "ord-routing": {"cost-mode" : "ordinal",
                     "cost-metric": "routingcost"},
      "ord-hop":    {"cost-mode" : "ordinal",
                     "cost-metric": "hopcount"}
    }
  },
  "resources" : [
    {
      "id" : "default-network-map",
      "uri" : "http://alto.example.com/networkmap",
      "media-type" : "application/alto-networkmap+json"
    }, {
      "id" : "numerical-routing-cost-map",
      "uri" : "http://alto.example.com/costmap/num/routingcost",
      "media-type" : "application/alto-costmap+json",
      "capabilities" : {
        "cost-type-names" : [ "num-routing" ]
      },
      "uses": [ "default-network-map" ]
    }, {
      "id" : "numerical-hopcount-cost-map",
      "uri" : "http://alto.example.com/costmap/num/hopcount",
      "media-type" : "application/alto-costmap+json",
      "capabilities" : {
        "cost-type-names" : [ "num-hop" ]
      },
      "uses": [ "default-network-map" ]
    }, {
      "id" : "custom-maps-resources",
      "uri" : "http://custom.alto.example.com/maps",
      "media-type" : "application/alto-directory+json",
    }, {
      "id" : "endpoint-property",
      "uri" : "http://alto.example.com/endpointprop/lookup",
      "media-type" : "application/alto-endpointprop+json",
      "accepts" : "application/alto-endpointpropparams+json",
    }
  ]
}
```

```

    "capabilities" : {
      "prop-types" : [ "pid" ]
    },
    "uses": [ "default-network-map" ]
  }, {
    "id" : "endpoint-cost",
    "uri" : "http://alto.example.com/endpointcost/lookup",
    "media-type" : "application/alto-endpointcost+json",
    "accepts" : "application/alto-endpointcostparams+json",
    "capabilities" : {
      "cost-constraints" : true,
      "cost-type-names" : [ "num-routing", "num-hop",
                           "ord-routing", "ord-hop" ]
    }
  }
]
}

```

Specifically, the "meta" member of the example IRD defines a field named "cost-types", which defines the names of cost types for this IRD. For example, "num-routing" in the example is the name that refers to a Cost Type with Cost Mode being "numerical" and Cost Metric being "routingcost". The value of "cost-types" is of type IRDMetaCostTypes defined below; see Section 9.7 for the definition of CostType.

The names defined in "cost-types" can be used in one or more "resources" entries. For example, the second entry of "resources" defines a Cost Map. The "cost-type-names" of its "capabilities" specifies that this resource supports a Cost Type named as "num-routing". The ALTO Client looks up the name "num-routing" in "cost-types" of the IRD to obtain the Cost Type named as "num-routing". The last entry of "resources" uses all four names defined in "cost-types".

```

object-map {
  JSONString -> CostType;
} IRDMetaCostTypes;

```

The "resources" array of the example IRD defines six Information Resources. For example, the last entry is to provide the Endpoint Cost Service, which is indicated by the media-type "application/alto-endpointcost+json". An ALTO Client should use uri "http://alto.example.com/endpointcost/lookup" to access the service. The ALTO Client should format its request body to be the

"application/alto-endpointcostparams+json" media type, as specified by the "accepts" attribute of the Information Resource. The "cost-type-names" member of the "capabilities" attribute of the Information Resource includes 4 defined cost types from the "meta" member of the IRD. Hence, one can verify that the Endpoint Cost Information Resource supports both Cost Metrics 'routingcost' and 'hopcount', each available for both 'numerical' and 'ordinal'. When requesting the Information Resource, an ALTO Client can specify cost constraints, as indicated by the "cost-constraints" member of the "capabilities" attribute.

#### 8.5.4. Delegation and Multiple Choices

ALTO Information Resource Directory provides flexibility to an ALTO Server (e.g., delegation) so that it MAY indicate multiple Information Resources using one URI endpoint. In the example above, the ALTO Server provides additional Network and Cost Maps via a separate subdomain, "custom.alto.example.com". In particular, the maps available via this subdomain are Filtered Network and Cost Maps as well as pre-generated maps for the "hopcount" and "routingcost" Cost Metrics in the "ordinal" Cost Mode.

Consider the preceding example. The fourth entry of "resources" provides additional Network and Cost Maps via a separate subdomain: "custom.alto.example.com". This delegation is indicated by the media-type "application/alto-directory+json". The ALTO Client can discover the maps available at "custom.alto.example.com" by successfully performing a request to "http://custom.alto.example.com/maps":

```
GET /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: TBA
Content-Type: application/alto-directory+json
```

```
{
  "meta" : {
    "cost-types": {
      "num-routing": {"cost-mode" : "numerical",
                     "cost-metric": "routingcost",
                     "description": "My default"},
```

```
        "num-hop":      {"cost-mode" : "numerical",
                          "cost-metric": "hopcount"}
        "ord-routing": {"cost-mode" : "ordinal",
                          "cost-metric": "routingcost"},
        "ord-hop":      {"cost-mode" : "ordinal",
                          "cost-metric": "hopcount"}
    }
},
"resources" : [
    {
        "id" : "filtered-network-map",
        "uri" : "http://custom.alto.example.com/networkmap/filtered",
        "media-type" : "application/alto-networkmap+json",
        "accepts" : "application/alto-networkmapfilter+json"
    }, {
        "id" : "filtered-cost-map",
        "uri" : "http://custom.alto.example.com/costmap/filtered",
        "media-type" : "application/alto-costmap+json",
        "accepts" : "application/alto-costmapfilter+json",
        "capabilities" : {
            "cost-constraints" : true,
            "cost-type-names" : [ "num-routing", "num-hop",
                                  "ord-routing", "ord-hop" ]
        },
        "uses": [ "default-network-map" ]
    }, {
        "id" : "ordinal-routing-cost-map",
        "uri" : "http://custom.alto.example.com/ord/routingcost",
        "media-type" : [ "application/alto-costmap+json",
                          "application/alto-routingcost+json" ],
        "capabilities" : {
            "cost-type-names" : [ "ord-routing" ]
        },
        "uses": [ "default-network-map" ]
    }, {
        "id" : "ordinal-hopcount-cost-map",
        "uri" : "http://custom.alto.example.com/ord/hopcount",
        "media-type" : "application/alto-costmap+json",
        "capabilities" : {
            "cost-type-names" : [ "ord-hop" ],
        },
        "uses": [ "default-network-map" ]
    }
]
}
```

#### 8.5.5. Usage Considerations

##### 8.5.5.1. 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 8.5.3, a full Cost Map with "numerical" Cost Mode and "routingcost" Cost Metric 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 cachable. However, an ALTO Client may need to use POST, for example, to get ALTO costs or properties that are for a restricted set of PIDs or Endpoints, or to update cached information previously acquired via GET requests."

##### 8.5.5.2. 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".

#### 8.6. Information Resource: Content Encoding

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

Specifically, each Information Resource has a single top-level JSON object of type InfoResourceEntity:

```
object {  
  InfoResourceMeta      meta;  
  InfoResourceDataType  data;  
} InfoResourceEntity;
```

with members:

meta meta-information pertaining to the Information Resource;  
data the data contained in the Information Resource.

#### 8.6.1. Meta Information

Meta information is encoded as a JSON object. This document does not specify any members, but it is defined here as a standard container for extensibility. Specifically, `InfoResourceMetaData` is defined as:

```
object-map {  
  JSONString -> JSONValue  
} InfoResourceMetaData;
```

#### 8.6.2. Data Information

The "data" member of the `InfoResourceEntity` encodes the resource-specific data. In this document, we define four specific `InfoResourceDataType`: `InfoResourceNetworkMap`, `InfoResourceCostMap`, `InfoResourceEndpointProperty`, and `InfoResourceEndpointCostMap`, whose structures will be detailed below.

#### 8.6.3. Example

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

```
HTTP/1.1 200 OK  
Content-Length: 40  
Content-Type: application/alto-costmap+json
```

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



## 8.7. Protocol 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' entity body. If no ALTO-layer information is available, an ALTO Server may omit an ALTO Error resource from the response.

With or without additional ALTO-layer error information, an ALTO Server MUST set an appropriate HTTP status code. It is important to note that the HTTP Status Code and ALTO Error Resource have distinct roles. An ALTO Error Resource provides detailed information about 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.

### 8.7.1. Media Type

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

### 8.7.2. Resource Format and Error Codes

An ALTO Error Resource has the format:

```
object {  
  JSONString code;  
} ErrorResponseEntity;
```

where:

code An ALTO Error Code defined in Table 1. Note that the ALTO Error Codes defined in Table 1 are limited to support the error conditions needed for purposes of this document. Additional status codes may be defined in companion or extension documents.

| ALTO Error Code         | Description                                      |
|-------------------------|--------------------------------------------------|
| E_SYNTAX                | Parsing error in request (including identifiers) |
| E_JSON_FIELD_MISSING    | Required field missing                           |
| E_JSON_VALUE_TYPE       | JSON Value of unexpected type                    |
| E_INVALID_COST_MODE     | Invalid cost mode                                |
| E_INVALID_COST_METRIC   | Invalid cost metric                              |
| E_INVALID_PROPERTY_TYPE | Invalid property type                            |

Table 1: Defined ALTO Error Codes.

If multiple errors are present in a single request (e.g., a request uses a JSONString when a JSONNumber 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.

#### 8.7.3. Overload Conditions and Server Unavailability

If an ALTO Server detects that it cannot handle a request from an ALTO Client due to excessive load, technical problems, or system maintenance, it SHOULD do one of the following:

- o Return an HTTP 503 ("Service Unavailable") status code to the ALTO Client. As indicated by [RFC2616], a the Retry-After HTTP header may be used to indicate when the ALTO Client should retry the request.
- o Return an HTTP 307 ("Temporary Redirect") status code indicating an alternate ALTO Server that may be able to satisfy the request.

The ALTO Server MAY also terminate the connection with the ALTO Client.

The particular policy applied by an ALTO Server to determine that it cannot service a request is outside of the scope of this document.

## 9. Protocol Specification: Basic ALTO Data Types

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

### 9.1. 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 (code points 0x30-0x39, 0x41-0x5A, and 0x61-0x7A), the hyphen ('-', code point 0x2D), the colon (':', code point 0x3A), the at ('@', code point 0x40), or the '.' separator (code point 0x2E). 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.

### 9.2. Resource ID

A Resource ID uniquely identifies an particular resource (e.g., a Network Map) within an ALTO Server (see Section 8.5).

A Resource ID is encoded as a US-ASCII string. The string MUST be no more than 64 characters, and MUST NOT contain any ASCII character below 0x21 or above 0x7E.

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

### 9.3. Version Tag

A Version Tag is defined as:

```
object {  
  ResourceID resource-id;  
  JSONString tag;  
} VersionTag;
```

The 'resource-id' attribute is an ID corresponding a resource (e.g., a Network Map) in an Information Resource Directory to which the Version Tag refers, and 'tag' encoded as a case-sensitive US-ASCII string. The 'tag' string MUST be no more than 64 characters, and MUST NOT contain any ASCII character below 0x21 or above 0x7E.

Two values of the VersionTag are equal if and only if both the the 'resource-id' attributes are byte-for-byte equal and the 'tag' attributes are byte-for-byte equal.

#### 9.4. 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.

##### 9.4.1. Address Type

Address Types are encoded as US-ASCII strings consisting of only alphanumeric characters (code points 0x30-0x39, 0x41-0x5A, and 0x61-0x7A). This document defines the address type 'ipv4' to refer to IPv4 addresses, and 'ipv6' to refer to IPv6 addresses. All Address Type identifiers appearing in an HTTP request or response with an 'application/alto-\*' media type MUST be registered in the ALTO Address Type registry (see Section 13.4).

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

##### 9.4.2. 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.

###### 9.4.2.1. IPv4

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

###### 9.4.2.2. IPv6

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

###### 9.4.2.3. 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.

#### 9.4.3. 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.

##### 9.4.3.1. IPv4

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

##### 9.4.3.2. IPv6

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

#### 9.4.4. 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-map {  
  AddressType -> EndpointPrefix<0..*>;  
} EndpointAddrGroup;
```

In particular, an Endpoint Address Group is a JSON object representing a map, where each key is the string corresponding to an address type, and the corresponding value is an array listing 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"
  ]
}
```

#### 9.5. 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.

#### 9.6. Cost Metric

A Cost Metric 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 (code points 0x30-0x39, 0x41-0x5A, and 0x61-0x7A), the hyphen ('-', code point 0x2D), the colon (':', code point 0x3A), or the '.' separator (0x2E). The '.' separator is reserved for future use and **MUST NOT** be used unless specifically indicated by a companion or extension document.

Identifiers prefixed with 'priv:' are reserved for Private Use [RFC5226]. Identifiers prefixed with 'exp:' are reserved for Experimental use. For an identifier with the 'priv:' or 'exp:' prefix, an additional string (e.g., company identifier or random string) **MUST** follow to reduce potential collisions. For example, a short string after 'exp:' to indicate the starting time of a specific experiment is recommended. All other identifiers appearing in an HTTP request or response with an 'application/alto-\*' media type **MUST** be registered in the ALTO Cost Metrics registry Section 13.2.

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

### 9.7. Cost Type

The combination of a `CostMetric` and a `CostMode` defines a `CostType`:

```
object {  
  CostMetric cost-metric;  
  CostMode   cost-mode;  
  [JSONString description;]  
} CostType;
```

'description', if present, MUST contain a US-ASCII string with a human-readable description of the cost-metric and cost-mode. An ALTO Client MAY present this string to a developer, as part of a discovery process. But the field SHOULD NOT be interpreted by an ALTO Client.

### 9.8. 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 (code points 0x30-0x39, 0x41-0x5A, and 0x61-0x7A), the hyphen ('-', code point 0x2D), the colon (':', code point 0x3A), or the '.' separator (0x2E). The '.' separator is reserved for future use and MUST NOT be used unless specifically indicated by a companion or extension document.

Identifiers prefixed with 'priv:' are reserved for Private Use [RFC5226]. Identifiers prefixed with 'exp:' are reserved for Experimental use. For an identifier with the 'priv:' or 'exp:' prefix, an additional string (e.g., company identifier or random string) MUST follow to reduce potential collisions. For example, a short string after 'exp:' to indicate the starting time of a specific experiment is recommended. 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 13.3.

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

## 10. Protocol Specification: Service Information Resources

This section documents the individual Information Resources defined to provide the services defined in this document.

## 10.1. 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.

### 10.1.1. Network Map

The Network Map Information Resource lists for each PID, the network locations (endpoints) within the PID. An ALTO Server **MUST** provide at least one Network Map.

#### 10.1.1.1. Media Type

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

#### 10.1.1.2. HTTP Method

The Network Map resource is requested using the HTTP GET method.

#### 10.1.1.3. Accept Input Parameters

None.

#### 10.1.1.4. Capabilities

None.

#### 10.1.1.5. Response

The "data" member of the returned InfoResourceEntity for a Network Map is an object of type InfoResourceNetworkMap:

```
object {  
  VersionTag      map-vtag;  
  NetworkMapData map;  
} InfoResourceNetworkMap;  
  
object-map {  
  PIDName -> EndpointAddrGroup;  
} NetworkMapData;
```

with members:



map-vtag The Version Tag (Section 6.3) of the Network Map.

map The Network Map data itself.

NetworkMapData is a JSON object representing a dictionary map with each key representing a single PID, and the value the associated set of endpoint addresses.

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

#### 10.1.1.6. 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: TBA
Content-Type: application/alto-networkmap+json
```

```
{
  "meta" : {},
  "data" : {
    "map-vtag" : {
      "resource-id": "default-network-map",
      "tag": "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"
        ]
      }
    }
  }
}
```

The encodings were chosen for readability and compactness. If lookup efficiency at runtime is crucial, then the returned Cost Map and Network Map can be transformed into data structures offering more efficient lookup. For example, one may store the Cost Map as a matrix, and the Network Map as a trie-based data structure, which may allow efficient longest-prefix matching of IP addresses.

#### 10.1.2. 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 Metric and Cost Mode. This resource **MUST** be provided for at least the

'routingcost' Cost Metric.

#### 10.1.2.1. Media Type

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

#### 10.1.2.2. HTTP Method

The Cost Map resource is requested using the HTTP GET method.

#### 10.1.2.3. Accept Input Parameters

None.

#### 10.1.2.4. Capabilities

The capabilities of an ALTO Server URI providing an unfiltered cost map is a JSON Object of type CostMapCapabilities:

```
object {  
  JSONString cost-type-names<1..*>;  
} CostMapCapabilities;
```

with member:

**cost-type-names** A sequence of CostType names defined in "cost-types" of the "meta" member of an IRD. These represent the Cost Types that are supported via the corresponding URI in the IRD. If there is more than one Cost Type in this list, then the ALTO Server SHOULD return an IRD to the client to lead it towards the URIs for the corresponding Cost Maps. Since an unfiltered Cost Map is requested via an HTTP GET that accepts no input parameters, an ALTO Client MUST be led towards a resource that has a single element in the 'cost-type-names' list.

#### 10.1.2.5. Response

The "data" member of the returned InfoResourceEntity for a Cost Map is an object of type InfoResourceCostMap:

```
object {  
  CostType      cost-type;  
  VersionTag    map-vtag;  
  CostMapData   map;  
} InfoResourceCostMap;
```

```
object-map {  
  PIDName -> DstCosts;  
} CostMapData;
```

```
object-map {  
  PIDName -> JSONValue;  
} DstCosts;
```

with members:

cost-type Cost Type (Section 9.7) used in the Cost Map.

map-vtag The Version Tag (Section 6.3) of the full (unfiltered) Network Map used to generate the Cost Map.

map The Cost Map data itself.

CostMapData is a dictionary map object, with each key being the PIDName string identifying the corresponding Source PID, and value being a type of DstCosts, which denotes the associated costs from the Source PID to a set of destination PIDs (Section 6.2). An implementation of the protocol in this document SHOULD assume that the cost is a JSONNumber and fail to parse if it is not, unless the implementation is using an extension to this document that indicates when and how costs of other data types are signaled.

The returned Cost Map MUST include the Path Cost for each (Source PID, Destination PID) pair for which a Path Cost is defined. An ALTO Server MAY omit entries for which a Path Cost is not defined (e.g., both the Source and Destination PIDs contain addresses outside of the Network Provider's administrative domain).

#### 10.1.2.6. 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: TBA
Content-Type: application/alto-costmap+json

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

Similar to the Network Map case, we considered array-based encoding for "map", but chose the current encoding for clarity.

## 10.2. 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.

### 10.2.1. Filtered Network Map

A Filtered Network Map is a Network Map Information Resource (Section 10.1.1) 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.

#### 10.2.1.1. Media Type

As a Filtered Network Map is a Network Map, it uses the media type defined for Network Map at Section 10.1.1.1.

#### 10.2.1.2. HTTP Method

A Filtered Network Map is requested using the HTTP POST method.

#### 10.2.1.3. Accept Input Parameters

An ALTO Client supplies filtering parameters by specifying media type "application/alto-networkmapfilter+json" with HTTP POST body containing a JSON Object of type ReqFilteredNetworkMap, where:

```
object {  
  PIDName pids<0..*>;  
  [AddressType address-types<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.

**address-types** Specifies list of address types to be included in the returned Filtered Network Map. If the "address-types" member is not specified, or the list of address types is empty, the ALTO Server MUST interpret the list as if it contained a list of all address types known to the ALTO Server. The ALTO Server MUST interpret entries appearing multiple times as if they appeared only once.

#### 10.2.1.4. Capabilities

None.

#### 10.2.1.5. Response

See Section 10.1.1.5 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. Similarly, the ALTO Server MUST only enumerate addresses within each PID that have types which were specified (implicitly or explicitly) in the request. If the input parameters contain an address type that is not currently known to the ALTO Server, the ALTO Server MUST behave as if the address type did not appear in the input parameters.

The Version Tag included in the response MUST correspond to the full (unfiltered) Network Map Information Resource from which the filtered information is provided. This ensures that a single, canonical Version Tag is used independent of any filtering that is requested by an ALTO Client.

#### 10.2.1.6. Example

```
POST /networkmap/filtered HTTP/1.1
Host: custom.alto.example.com
Content-Length: 27
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: TBA
Content-Type: application/alto-networkmap+json
```

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

### 10.2.2. Filtered Cost Map

A Filtered Cost Map is a Cost Map Information Resource (Section 10.1.2) 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.

#### 10.2.2.1. Media Type

As a Filtered Cost Map is a Cost Map, it uses the media type defined for Cost Map at Section 10.1.2.1.

#### 10.2.2.2. HTTP Method

A Filtered Cost Map is requested using the HTTP POST method.

#### 10.2.2.3. Accept Input Parameters

The input parameters for a Filtered Map 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 {  
  CostType    cost-type;  
  [JSONString constraints<0..*>;]  
  [PIDFilter  pids;]  
} ReqFilteredCostMap;
```

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

with members:

**cost-type** The CostType (Section 9.7) for the returned costs. The cost-metric and cost-mode fields MUST match one of the supported Cost Types indicated in this resource's capabilities (Section 10.2.2.4). The ALTO Client SHOULD omit the description field, and if present, the ALTO Server MUST ignore the description field.



**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 10.2.2.4) indicate that constraint support is not available. A constraint contains two entities separated by whitespace: (1) an operator, 'gt' for greater than, 'lt' for less than, 'ge' for greater than or equal to, 'le' for less than or equal to, or 'eq' for equal to; (2) a target cost value. The cost value is a number that MUST be defined in the same units as the Cost Metric indicated by the cost-metric parameter. ALTO Servers SHOULD use at least IEEE 754 double-precision floating point [IEEE.754.2008] to store the cost value, 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.

#### 10.2.2.4. Capabilities

The URI providing this resource supports all capabilities documented in Section 10.1.2.4 (with identical semantics), plus additional capabilities. In particular, the capabilities are defined by a JSON object of type `FilteredCostMapCapabilities`:

```
object {  
  JSONString cost-type-names<1..*>;  
  JSONBool cost-constraints;  
} FilteredCostMapCapabilities;
```

with members:

**cost-type-names** See Section 10.1.2.4 and note that the array can have 1 to many cost types.

**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. ALTO Clients should be aware that constraints may not have the intended effect for cost maps with the 'ordinal' Cost Mode

since ordinal costs are not restricted to being sequential integers.

#### 10.2.2.5. Response

See Section 10.1.2.5 for the format.

The returned Cost Map MUST contain only source/destination pairs that have been 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 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.

Note that ALTO Clients should verify that the Version Tag included in the response is consistent with the Version Tag of the Network Map used to generate the request (if applicable). If it is not, the ALTO Client may wish to request an updated Network Map, identify changes, and consider requesting a new Filtered Cost Map.

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

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

```
{
  "meta" : {},
  "data" : {
    "cost-type": {"cost-mode" : "numerical",
                  "cost-metric" : "routingcost"},
    "map-vtag" : {
      "resource-id": "default-network-map",
      "tag": "1266506139"
    },
    "map" : {
      "PID1": { "PID1": 0, "PID2": 1, "PID3": 2 }
    }
  }
}
```

### 10.3. Endpoint Property Service

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

#### 10.3.1. 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 one or more Endpoint Property resources, then

at least one MUST provide the 'pid' property.

#### 10.3.1.1. Media Type

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

#### 10.3.1.2. HTTP Method

The Endpoint Property resource is requested using the HTTP POST method.

#### 10.3.1.3. Accept Input Parameters

An ALTO Client supplies the endpoint properties to be queried through a media type "application/alto-endpointpropparams+json", and specifies in the HTTP POST entity body a JSON Object of type ReqEndpointProp:

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

with members:

**properties** List of endpoint properties to be returned for each endpoint. Each specified property MUST be included in the list of supported properties indicated by this resource's capabilities (Section 10.3.1.4). 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.

#### 10.3.1.4. 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 EndpointPropertyCapabilities:

```
object {  
  EndpointPropertyType prop-types<1..*>;
```

```
} EndpointPropertyCapabilities;
```

with members:

prop-types The Endpoint Properties (see Section 9.8) supported by the corresponding URI.

#### 10.3.1.5. Response

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

```
object {  
  VersionTag          map-vtag; [DEPEND ON PROPERTIES]  
  EndpointPropertyMapData map;  
} InfoResourceEndpointProperty;  
  
object-map {  
  TypedEndpointAddr -> EndpointProps;  
} EndpointPropertyMapData;  
  
object {  
  EndpointPropertyType -> JSONValue;  
} EndpointProps;
```

EndpointPropertyMapData 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 EndpointPropertyType. An implementation of the protocol in this document SHOULD assume that the property value is a JSONString and fail to parse if it is not, unless the implementation is using an extension to this document that indicates when and how property values of other data types are signaled.

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's value for a particular endpoint, then it MUST omit that property from the response for only that endpoint.

The ALTO Server MAY include the Version Tag (Section 6.3) of the full

(unfiltered) Network Map used to generate the response (if desired and applicable) as the 'map-vtag' member in the response. If the 'pid' property is returned for any endpoints in the response, the 'map-vtag' member is REQUIRED. Otherwise, it is OPTIONAL.

#### 10.3.1.6. Example

```
POST /endpointprop/lookup HTTP/1.1
Host: alto.example.com
Content-Length: 96
Content-Type: application/alto-endpointpropparams+json
Accept: application/alto-endpointprop+json,application/alto-error+json
```

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

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

```
{
  "meta" : {},
  "data": {
    "map-vtag" : {
      "resource-id": "default-network-map",
      "tag": "1266506139"
    },
    "map" : {
      "ipv4:192.0.2.34" : { "pid": "PID1", "example-prop": "1" },
      "ipv4:203.0.113.129" : { "pid": "PID3" }
    }
  }
}
```

#### 10.4. 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.

#### 10.4.1. Endpoint Cost

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

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 14.3 for additional details.

##### 10.4.1.1. Media Type

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

##### 10.4.1.2. HTTP Method

The Endpoint Cost resource is requested using the HTTP POST method.

##### 10.4.1.3. Accept Input Parameters

An ALTO Client supplies the endpoint cost parameters through a media type "application/alto-endpointcostparams+json", with an HTTP POST entity body of a JSON Object of type ReqEndpointCostMap:

```
object {
  CostType          cost-type;
  [JSONString       constraints<0..*>;]
  EndpointFilter    endpoints;
} ReqEndpointCostMap;

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

with members:

**cost-type** The Cost Type (Section 9.7) to use for returned costs. The cost-metric and cost-mode fields MUST match one of the supported Cost Types indicated in this resource's capabilities (Section 10.4.1.4). The ALTO Client SHOULD omit the description field, and if present, the ALTO Server MUST ignore the description field.

**constraints** Defined equivalently to the "constraints" input parameter of a Filtered Cost Map (see Section 10.2.2).

**endpoints** A list of Source Endpoints and Destination Endpoints for which Path Costs are to be returned. If the list of Source or Destination 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 12.3 for discussion and considerations regarding this mode). The Source and Destination Endpoint lists MUST NOT be both empty. The ALTO Server MUST interpret entries appearing multiple times in a list as if they appeared only once.

#### 10.4.1.4. Capabilities

In this document, we define `EndpointCostCapabilities` the same as `FilteredCostMapCapabilities`. See Section 10.2.2.4.

#### 10.4.1.5. Response

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

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

object-map {
  TypedEndpointAddr -> EndpointDstCosts;
} EndpointCostMapData;

object-map {
  TypedEndpointAddr -> JSONValue;
} EndpointDstCosts;
```

`InfoResourceEndpointCostMap` has members:

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

**map** The Endpoint Cost Map data itself.

`EndpointCostMapData` is a dictionary map object with each key representing a `TypedEndpointAddr` string identifying the Source Endpoint specified in the input parameters; the name for a member is. For each Source Endpoint, a `EndpointDstCosts` dictionary map object



denotes the associated cost to each Destination Endpoint specified in input parameters. An implementation of the protocol in this document SHOULD assume that the cost value is a JSONNumber and fail to parse if it is not, unless the implementation is using an extension to this document that indicates when and how costs of other data types are signaled. If the ALTO Server does not define a cost value from a Source Endpoint to a particular Destination Endpoint, it MAY be omitted from the response.

## 10.4.1.6. Example

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: 195
Content-Type: application/alto-endpointcostparams+json
Accept: application/alto-endpointcost+json,application/alto-error+json
```

```
{
  "cost-type": {"cost-mode" : "ordinal",
               "cost-metric" : "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: 231
Content-Type: application/alto-endpointcost+json
```

```
{
  "meta" : {},
  "data" : {
    "cost-type": {"cost-mode" : "ordinal",
                 "cost-metric" : "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
      }
    }
  }
}
```

## 11. Use Cases

The sections below depict typical use cases. While these use cases focus on peer-to-peer applications, ALTO can be applied to other

environments such as CDNs [I-D.jenkins-alto-cdn-use-cases].

#### 11.1. ALTO Client Embedded in P2P Tracker

Many currently-deployed P2P systems use a Tracker to manage swarms and perform peer selection. Such a P2P Tracker can already use a variety of information to perform peer selection to meet application-specific goals. By acting as an ALTO Client, the 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.

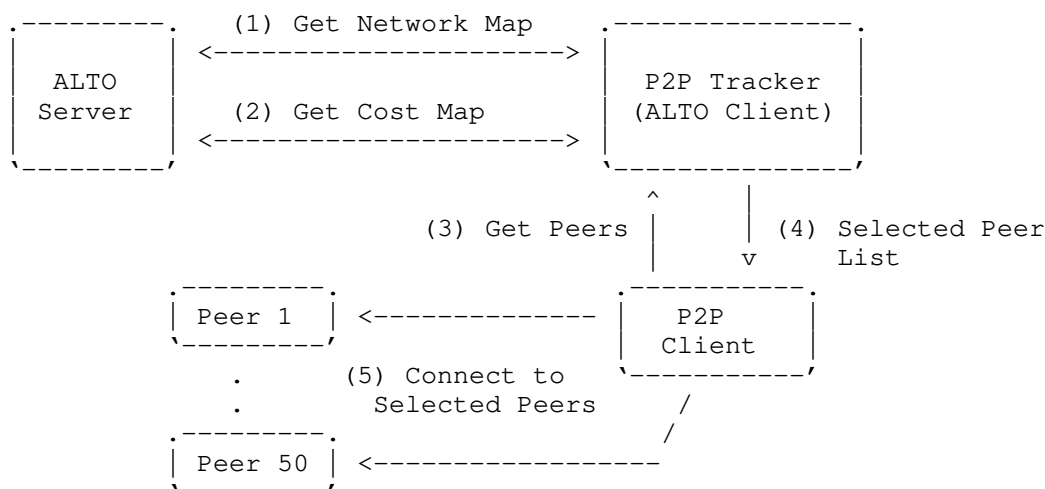


Figure 4: ALTO Client Embedded in P2P Tracker

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 from the ALTO Server using the Network Map query the Network Map covering all PIDs. The Network Map includes the IP prefixes contained in each PID, allowing the P2P tracker to locally map P2P clients into PIDs.

2. The P2P Tracker requests from the ALTO Server the Cost Map amongst all PIDs identified in the preceding step.
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.

#### 11.2. 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.

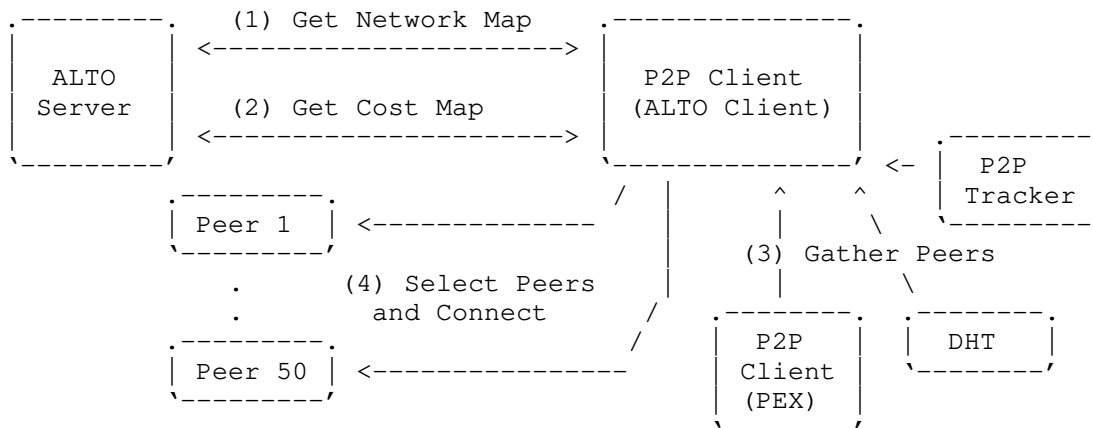


Figure 5: ALTO Client Embedded in P2P Client

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.

#### 11.3. 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.

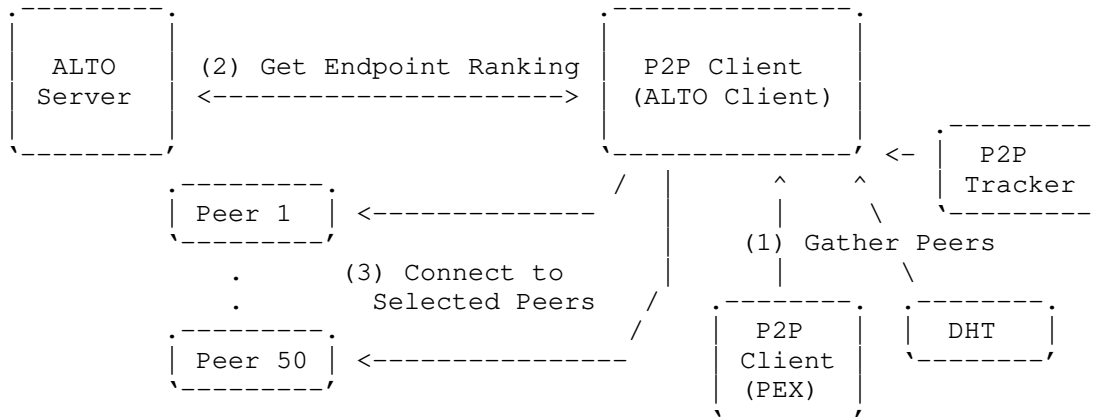


Figure 6: ALTO Client Embedded in P2P Client: Ranking

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.

## 12. Discussions

### 12.1. 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 8.5) to locate an Information Resource with the desired ALTO Information.

## 12.2. Hosts with Multiple Endpoint Addresses

In practical deployments, a particular host can be reachable using multiple addresses (e.g., a wireless IPv4 connection, a wireline IPv4 connection, and a wireline IPv6 connection). In general, the particular network path followed when sending packets to the host will depend on the address that is used. Network providers may prefer one path over another. 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 endpoint addresses and address types. With this support, the ALTO Protocol allows an 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.

## 12.3. 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.

There may be cases where an ALTO Client needs to determine its own IP address, such as when specifying a source Endpoint Address in the Endpoint Cost Service. It is possible that an ALTO Client has multiple network interface addresses, and that some or all of them may require NAT for connectivity to the public Internet.

If a public IP address is required for a network interface, the ALTO Client SHOULD use the Session Traversal Utilities for NAT (STUN) [RFC5389]. 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.

ALTO Clients should be cognizant that the network path between Endpoints can depend on multiple factors, e.g., source address, and destination address used for communication. An ALTO Server provides information based on Endpoint Addresses (more generally, Network Locations), but the mechanisms used for determining existence of connectivity or usage of NAT between Endpoints are out of scope of this document.

#### 12.4. 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.

### 13. IANA Considerations

#### 13.1. 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 8.5    |
| application | alto-networkmap+json         | Section 10.1.1 |
| application | alto-networkmapfilter+json   | Section 10.2.1 |
| application | alto-costmap+json            | Section 10.1.2 |
| application | alto-costmapfilter+json      | Section 10.2.2 |
| application | alto-endpointprop+json       | Section 10.3.1 |
| application | alto-endpointpropparams+json | Section 10.3.1 |
| application | alto-endpointcost+json       | Section 10.4.1 |
| application | alto-endpointcostparams+json | Section 10.4.1 |
| application | alto-error+json              | Section 8.7    |

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 14.

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 2 for 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: Internet Engineering Task Force  
(mailto:iesg@ietf.org).

### 13.2. ALTO Cost Metric Registry

This document requests the creation of an ALTO Cost Metric registry, listed in Table 3, to be maintained by IANA.

| Identifier  | Intended Semantics  |
|-------------|---------------------|
| routingcost | See Section 6.1.1.1 |
| priv:       | Private use         |
| exp:        | Experimental use    |

Table 3: ALTO Cost Metrics.

This registry serves two purposes. First, it ensures uniqueness of identifiers referring to ALTO Cost Metrics. Second, it provides references to particular semantics of allocated Cost Metrics to be applied by both ALTO Servers and applications utilizing ALTO Clients.

New ALTO Cost Metrics 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 Metric 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 Metric should be interpreted. Updates and deletions of ALTO Cost Metrics follow the same procedure.

Registered ALTO Cost Metric identifiers MUST conform to the syntactical requirements specified in Section 9.6. 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:

- o Identifier: The name of the desired ALTO Cost Metric.
- o 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.
- o Security Considerations: ALTO Costs expose information to ALTO Clients. As such, proper usage of a particular Cost Metric 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 Metric.

This specification requests registration of the identifier 'routingcost'. Semantics for this Cost Metric are documented in Section 6.1.1.1, and security considerations are documented in Section 14.3.

### 13.3. ALTO Endpoint Property Type Registry

This document requests the creation of an ALTO Endpoint Property Types registry, listed in Table 4, to be maintained by IANA.

| Identifier | Intended Semantics |
|------------|--------------------|
| pid        | See Section 7.1.1  |
| priv:      | Private use        |
| exp:       | Experimental use   |

Table 4: ALTO Endpoint Property Types.

The maintenance of this registry is similar to that of the preceding ALTO Cost Metrics.

### 13.4. ALTO Address Type Registry

This document requests the creation of an ALTO Address Type registry, listed in Table 5, to be maintained by IANA.

| Identifier | Address Encoding  | Prefix Encoding   | Mapping to/from IPv4/v6 |
|------------|-------------------|-------------------|-------------------------|
| ipv4       | See Section 9.4.2 | See Section 9.4.3 | Direct mapping to IPv4  |
| ipv6       | See Section 9.4.2 | See Section 9.4.3 | Direct mapping to IPv6  |

Table 5: ALTO Address Types.

This registry serves two purposes. First, it ensures uniqueness of identifiers referring to ALTO Address Types. Second, it states the requirements for allocated Address Type identifiers.

New ALTO Address 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 the new ALTO Address Types and their security considerations has been provided. The provided documentation should indicate how an address of a registered type is encoded as an EndpointAddr and, if possible, a compact method (e.g., IPv4 and IPv6 prefixes) for encoding a set of addresses as an EndpointPrefix. Updates and deletions of ALTO Address Types follow the same procedure.

Registered ALTO Address Type identifiers MUST conform to the syntactical requirements specified in Section 9.4.1. Identifiers are to be recorded and displayed as ASCII strings.

Requests to add a new value to the registry MUST include the following information:

- o Identifier: The name of the desired ALTO Address Type.
- o Endpoint Address Encoding: The procedure for encoding an address of the registered type as an EndpointAddr (see Section 9.4.2).
- o Endpoint Prefix Encoding: The procedure for encoding a set of addresses of the registered type as an EndpointPrefix (see Section 9.4.3). If no such compact encoding is available, the same encoding used for a singular address may be used. In such a case, it must be documented that sets of addresses of this type always have exactly one element.
- o Mapping to/from IPv4/IPv6 Addresses: If possible, a mechanism to map addresses of the registered type to and from IPv4 or IPv6 addresses should be specified.
- o Security Considerations: In some usage scenarios, Endpoint Addresses carried in ALTO Protocol messages may reveal information about an ALTO Client or an ALTO Service Provider. Applications and ALTO Service Providers using addresses of the registered type should be made aware of how (or if) the addressing scheme relates to private information and network proximity.

This specification requests registration of the identifiers 'ipv4' and 'ipv6', as shown in Table 5.

### 13.5. ALTO Error Code Registry

This document requests the creation of an ALTO Error Code registry, listed in Table 1, to be maintained by IANA.

## 14. Security Considerations

Some environments and use cases of ALTO require consideration of security attacks on ALTO Servers and Clients. In order to support those environments interoperably, the ALTO requirements document [RFC6708] outlines minimum-to-implement authentication and other security requirements. Below we consider the threats and protection strategies.

### 14.1. Authenticity and Integrity of ALTO Information

#### 14.1.1. Risk Scenarios

An attacker may want to provide false or modified ALTO Information Resources or Information Resource Directory to ALTO Clients to achieve certain malicious goals. As an example, an attacker may provide false endpoint properties. For example, suppose that a network supports an endpoint property named "hasQuota" which reports if the endpoint has usage quota. An attacker may want to generate a false reply to lead to unexpected charges to the endpoint. An attack may also want to provide false Cost Map. For example, by faking a Cost Map that highly prefers a small address range or a single address, the attacker may be able to turn a distributed application into a Distributed Denial of Service (DDoS) tool.

Depending on the network scenario, an attacker can attack authenticity and integrity of ALTO Information Resources using various techniques, including, but not limited to, sending forged DHCP replies in an Ethernet, DNS poisoning, and installing a transparent HTTP proxy that does some modifications.

#### 14.1.2. Protection Strategies

ALTO protects the authenticity and integrity of ALTO Information (both Information Directory and individual Information Resources) by leveraging the authenticity and integrity mechanisms in TLS. In particular, the ALTO Protocol requires that HTTP over TLS [RFC2818] MUST be supported, when protecting the authenticity and integrity of ALTO Information is required. The rules in [RFC2818] for a client to verify server identity using server certificates MUST be supported. ALTO Providers who request server certificates and certification authorities who issue ALTO-specific certificates SHOULD consider the recommendations and guidelines defined in [RFC6125]

Software engineers developing and service providers deploying ALTO should make themselves familiar with up-to-date Best Current Practices on configuring HTTP over TLS.

#### 14.1.3. Limitations

The protection of HTTP over TLS for ALTO depends on that the domain name in the URI for the Information Resources is not comprised. This will depend on the protection implemented by service discovery.

A deployment scenario may require redistribution of ALTO information to improve scalability. When authenticity and integrity of ALTO information are still required, then ALTO Clients obtaining ALTO information through redistribution must be able to validate the received ALTO information. Support for this validation is not provided in this document, but may be provided by extension documents.

#### 14.2. Potential Undesirable Guidance from Authenticated ALTO Information

##### 14.2.1. Risk Scenarios

The ALTO Service makes it possible for an ALTO Provider to influence the behavior of network applications. An ALTO Provider may be hostile to some applications and hence try to use ALTO Information Resources to achieve certain goals [RFC5693]: "redirecting applications to corrupted mediators providing malicious content, or applying policies in computing Cost Map based on criteria other than network efficiency." See [I-D.ietf-alto-deployments] for additional discussions on faked ALTO Guidance.

A related scenario is that an ALTO Server could unintentionally give "bad" guidance. For example, if many ALTO Clients follow the Cost Map or Endpoint Cost guidance without doing additional sanity checks or adaptation, more preferable hosts and/or links could get overloaded while less preferable ones remain idle; see AR-14 of [RFC6708] for related application considerations.

##### 14.2.2. Protection Strategies

To protect applications from undesirable ALTO Information Resources, it is important to note that there is no protocol mechanism to require conforming behaviors on how applications use ALTO Information Resources. An application using ALTO may consider including a mechanism to detect misleading or undesirable results from using ALTO Information Resources. For example, if throughput measurements do not show "better-than-random" results when using the Cost Map to select resource providers, the application may want to disable ALTO usage or switch to an external ALTO Server provided by an "independent organization" (see AR-20 and AR-21 in [RFC 6708]). If the first ALTO Server is provided by the access network service

provider and the access network service provider tries to redirect access to the external ALTO Server back to the provider's ALTO Server or try to tamper with the responses, the preceding authentication and integrity protection can detect such a behavior.

### 14.3. Confidentiality of ALTO Information

#### 14.3.1. Risk Scenarios

Although in many cases ALTO Information Resources may be regarded as non-confidential information, there are deployment cases where ALTO Information Resources can be sensitive information that can pose risks if exposed to unauthorized parties. We discuss the risks and protection strategies for such deployment scenarios.

For example, an attacker may infer details regarding the topology, status, and operational policies of a network through the Network and Cost Maps. As a result, a sophisticated attacker may be able to infer more fine-grained topology information than an ISP hosting an ALTO Server intends to disclose. The attacker can leverage the information to mount effective attacks such as focusing on high-cost links.

Revealing some endpoint properties may also reveal additional information than the Provider intended. For example, when adding the line bitrate as one endpoint property, such information may be potentially linked to the income of the habitants at the network location of an endpoint.

In [RFC6708] Section 5.2.1, three types of risks associated with the confidentiality of ALTO Information Resources are identified: risk type (1) Excess disclosure of the ALTO service provider's data to an authorized ALTO Client; risk type (2) Disclosure of the ALTO service provider's data (e.g., network topology information) to an unauthorized third party; and risk type (3) Excess retrieval of the ALTO service provider's data by collaborating ALTO Clients. Section 10 of [I-D.ietf-alto-deployments] also discusses information leakage from ALTO.

#### 14.3.2. Protection Strategies

To address risk types (1) and (3), the Provider of an ALTO Server must be cognizant that the network topology and provisioning information provided through ALTO may lead to attacks. ALTO does not require any particular level of details of information disclosure, and hence the Provider should evaluate how much information is revealed and the associated risks.

To address risk type (2), the ALTO Protocol need confidentiality. Since ALTO requires that HTTP over TLS MUST be supported, the confidentiality mechanism is provided by HTTP over TLS.

For deployment scenarios where client authentication is desired to address risk type (2), ALTO requires that HTTP Digest Authentication MUST be supported to achieve ALTO Client Authentication to limit the number of parties with whom ALTO information is directly shared. Depending on the use-case and scenario, an ALTO Server may apply other access control techniques to restrict access to its services. Access control can also help to prevent Denial-of-Service attacks by arbitrary hosts from the Internet. See [I-D.ietf-alto-deployments] for a more detailed discussion on this issue.

#### 14.3.3. Limitations

ALTO Information Providers 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.

#### 14.4. Privacy for ALTO Users

##### 14.4.1. Risk Scenarios

The ALTO Protocol provides mechanisms in which the ALTO Client serving a user can send messages containing Network Location Identifiers (IP addresses or fine-grained PIDs) to the ALTO Server. This is particularly true for the Endpoint Property, Endpoint Cost, and fine-grained Filtered Map services. The ALTO Server or a third-party who is able to intercept such messages can store and process obtained information in order to analyze user behaviors and communication patterns. The analysis may correlate information collected from multiple clients to deduce additional application/content information. Such analysis can lead to privacy risks. For a more comprehensive classification of related risk scenarios, see cases 4, 5, and 6 in [RFC 6708], Section 5.2.

##### 14.4.2. Protection Strategies

To protect user privacy, an ALTO Client should be cognizant about potential ALTO Server tracking through client queries. An ALTO Client may consider the possibility of relying only on Network Map for PIDs and Cost Map amongst PIDs to avoid passing IP addresses of other endpoints (e.g., peers) to the ALTO Server. When specific IP addresses are needed (e.g., when using the Endpoint Cost Service), an



ALTO Client may consider obfuscation techniques such as specifying a broader address range (i.e., a shorter prefix length) or by zeroing out or randomizing the last few bits of IP addresses. Note that obfuscation may yield less accurate results.

#### 14.5. Availability of ALTO Service

##### 14.5.1. Risk Scenarios

An attacker may want to disable ALTO Service as a way to disable network guidance to large scale applications. 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).

##### 14.5.2. Protection Strategies

ALTO Provider should be cognizant of the workload at the ALTO Server generated by certain ALTO Queries, such as certain queries to the Map Service, the Map Filtering Service and the Endpoint Cost (Ranking) Service. One way to limit Denial-of-Service attacks is to employ access control to the ALTO Server. The ALTO Server can also indicate overload and reject repeated requests that can cause availability problems. More advanced protection schemes such as computational puzzles [I-D.jennings-sip-hashcash] may be considered in an extension document.

An ALTO Provider should also leverage the fact that the Map Service allows ALTO Servers to pre-generate maps that can be distributed to many ALTO Clients.

#### 15. Manageability Considerations

This section details operations and management considerations based on existing deployments and discussions during protocol development. It also indicates where extension documents are expected to provide appropriate functionality discussed in [RFC5706] as additional deployment experience becomes available.

##### 15.1. Operations

#### 15.1.1. Installation and Initial Setup

The ALTO Protocol is based on HTTP. Thus, configuring an ALTO Server may require configuring the underlying HTTP server implementation to define appropriate security policies, caching policies, performance settings, etc.

Additionally, an ALTO Service Provider will need to configure the ALTO information to be provided by the ALTO Server. The granularity of the topological map and the cost map is left to the specific policies of the ALTO Service Provider. However, a reasonable default may include two PIDs, one to hold the endpoints in the provider's network and the second PID to represent full IPv4 and IPv6 reachability (see Section 5.2.1), with the cost between each source/destination PID set to 1. Another operational issue that the ALTO Service Provider needs to consider is that the filtering service can degenerate into a full map service when the filtering input is empty. Although this choice as the degeneration behavior provides continuity, the operational impact should be considered.

Implementers employing an ALTO Client should attempt to automatically discover an appropriate ALTO Server. Manual configuration of the ALTO Server location may be used where automatic discovery is not appropriate. Methods for automatic discovery and manual configuration are discussed in [I-D.ietf-alto-server-discovery].

Specifications for underlying protocols (e.g., TCP, HTTP, SSL/TLS) should be consulted for their available settings and proposed default configurations.

#### 15.1.2. Migration Path

This document does not detail a migration path for ALTO Servers since there is no previous standard protocol providing the similar functionality.

There are existing applications making use of network information discovered from other entities such as whois, geo-location databases, or round-trip time measurements, etc. Such applications should consider using ALTO as an additional source of information; ALTO need not be the sole source of network information.

#### 15.1.3. Requirements on Other Protocols and Functional Components

The ALTO Protocol assumes that HTTP client and server implementations exist. It also assumes that JSON encoder and decoder implementations exist.

An ALTO Server assumes that it can gather sufficient information to populate Network and Cost maps. "Sufficient information" is dependent on the information being exposed, but likely includes information gathered from protocols such as IGP and EGP Routing Information Bases (see Figure 1). Specific mechanisms have been proposed (e.g., [I-D.medved-alto-svr-apis]) and are expected to be provided in extension documents.

#### 15.1.4. Impact and Observation on Network Operation

ALTO presents a new opportunity for managing network traffic by providing additional information to clients. The potential impact to network operation is large.

Deployment of an ALTO Server may shift network traffic patterns. Thus, an ALTO Service Provider should consider impacts on (or integration with) traffic engineering and the deployment of a monitoring service to observe the effects of ALTO operations. Note that ALTO-specific monitoring and metrics are discussed in 6.3 of [I-D.ietf-alto-deployments] and future versions of that document. In particular, an ALTO Service Provider may observe that ALTO Clients are not bound to ALTO Server guidance as ALTO is only one source of information.

An ALTO Service Provider should ensure that appropriate information is being exposed. Privacy implications for ISPs are discussed in Section 14.3. Both ALTO Service Providers and those using ALTO Clients should be aware of the impact of incorrect or faked guidance (see Section 10.3 of [I-D.ietf-alto-deployments] and future versions of that document).

### 15.2. Management

#### 15.2.1. Management Interoperability

A common management API would be desirable given that ALTO Servers may typically be configured with dynamic data from various sources, and ALTO Servers are intended to scale horizontally for fault-tolerance and reliability. A specific API or protocol is outside the scope of this document, but may be provided by an extension document.

Logging is an important functionality for ALTO Servers and, depending on the deployment, ALTO Clients. Logging should be done via syslog [RFC5424].

#### 15.2.2. Management Information

A Management Information Model (see Section 3.2 of [RFC5706]) is not provided by this document, but should be included or referenced by any extension documenting an ALTO-related management API or protocol.

#### 15.2.3. Fault Management

Monitoring ALTO Servers and Clients is described in Section 6.3 of [I-D.ietf-alto-deployments] and future versions of that document.

#### 15.2.4. Configuration Management

Standardized approaches and protocols to configuration management for ALTO are outside the scope of this document, but this document does outline high-level principles suggested for future standardization efforts.

An ALTO Server requires at least the following logical inputs:

- o Data sources from which ALTO Information is derived. This can either be raw network information (e.g., from routing elements) or pre-processed ALTO-level information in the form of a Network Map, Cost Map, etc.
- o Algorithms for computing the ALTO information returned to clients. These could either return information from a database, or information customized for each client.
- o Security policies mapping potential clients to the information that they have privilege to access.

Multiple ALTO Servers can be deployed for scalability. A centralized configuration database may be used to ensure they are providing the desired ALTO information with appropriate security controls. The ALTO information (e.g., Network Maps and Cost Maps) being served by each ALTO Server, as well as security policies (HTTP authentication, SSL/TLS client and server authentication, SSL/TLS encryption parameters) intended to serve the same information should be monitored for consistency.

#### 15.2.5. Performance Management

An exhaustive list of desirable performance information from a ALTO Servers and ALTO Clients are outside of the scope of this document. The following is a list of suggested ALTO-specific to be monitored based on the existing deployment and protocol development experience:

- o Requests and responses for each service listed in a Information Directory (total counts and size in bytes).
- o CPU and memory utilization
- o ALTO map updates
- o Number of PIDs
- o ALTO map sizes (in-memory size, encoded size, number of entries)

#### 15.2.6. Security Management

Section 14 documents ALTO-specific security considerations. Operators should configure security policies with those in mind. Readers should refer to HTTP [RFC2616] and SSL/TLS [RFC5246] and related documents for mechanisms available for configuring security policies. Other appropriate security mechanisms (e.g., physical security, firewalls, etc) should also be considered.

## 16. References

### 16.1. 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.
- [RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform  
Resource Identifier (URI): Generic Syntax", STD 66,  
RFC 3986, January 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.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.
- [RFC5389] Rosenberg, J., Mahy, R., Matthews, P., and D. Wing, "Session Traversal Utilities for NAT (STUN)", RFC 5389, October 2008.
- [RFC5424] Gerhards, R., "The Syslog Protocol", RFC 5424, March 2009.
- [RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, October 2009.
- [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.
- [RFC6708] Kiesel, S., Previdi, S., Stiemerling, M., Woundy, R., and Y. Yang, "Application-Layer Traffic Optimization (ALTO) Requirements", RFC 6708, September 2012.

## 16.2. Informative References

- [BitTorrent]  
"Bittorrent Protocol Specification v1.0",  
<<http://wiki.theory.org/BitTorrentSpecification>>.
- [Fielding-Thesis]  
Fielding, R., "Architectural Styles and the Design of Network-based Software Architectures", University of California, Irvine, Dissertation 2000, 2000.
- [I-D.akonjang-alto-proxidor]

Akonjang, O., Feldmann, A., Previdi, S., Davie, B., and D. Saucez, "The PROXIDOR Service", draft-akonjang-alto-proxidior-00 (work in progress), March 2009.

[I-D.ietf-alto-deployments]  
Stiemerling, M., Kiesel, S., and S. Previdi, "ALTO Deployment Considerations", draft-ietf-alto-deployments-06 (work in progress), February 2013.

[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-08 (work in progress), March 2013.

[I-D.ietf-httpbis-p2-semantics]  
Fielding, R. and J. Reschke, "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", draft-ietf-httpbis-p2-semantics-22 (work in progress), February 2013.

[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-03 (work in progress), June 2012.

[I-D.medved-alto-svr-apis]  
Medved, J., Ward, D., Peterson, J., Woundy, R., and D. McDysan, "ALTO Network-Server and Server-Server APIs", draft-medved-alto-svr-apis-00 (work in progress), March 2011.

[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.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.

[RFC5706] Harrington, D., "Guidelines for Considering Operations and Management of New Protocols and Protocol Extensions", RFC 5706, November 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 Sebastian Kiesel (University of Stuttgart) and Jan Seedorf (NEC) for substantial contributions to the Security Considerations section. Ben Niven-Jenkins (Velocix), Wendy Roome, Michael Scharf and Sabine Randriamasy (Alcatel-Lucent) gave substantial feedback and suggestions on the protocol design.

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 (ATT), 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 (ATT), Ingmar Poesse (DT Labs/TU Berlin), James Royalty (Pando Networks), Damien Saucez (UCL) Thomas Scholl (ATT), Emilio Sepulveda (Telefonica), Avi Silberschatz (Yale University), Hassan Sipra (Bell Canada), Georgios Smaragdakis (DT Labs/TU Berlin), Haibin Song (Huawei), Oliver Spatscheck (ATT), See-Mong Tang (Microsoft), Jia Wang (ATT), Hao Wang (Yale University), Ye Wang (Yale University), Haiyong Xie (Yale University).

## Appendix B. Design History and Merged Proposals

The ALTO Protocol specified in this document consists of contributions from

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

- o ATTP [ATTP]; and
- o Proxidor [I-D.akonjang-alto-proxidor].

#### Appendix C. 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)  
Cisco Systems  
170 West Tasman Dr  
San Jose CA  
USA

Email: repenno@cisco.com

Y. Richard Yang (editor)  
Yale University  
51 Prospect St  
New Haven CT  
USA

Email: [yry@cs.yale.edu](mailto:yry@cs.yale.edu)



ALTO  
Internet-Draft  
Intended status: Standards Track  
Expires: September 22, 2013

S. Kiesel  
University of Stuttgart  
M. Stiernerling  
NEC Europe Ltd.  
N. Schwan  
Stuttgart, Germany  
M. Scharf  
Alcatel-Lucent Bell Labs  
H. Song  
Huawei  
March 21, 2013

ALTO Server Discovery  
draft-ietf-alto-server-discovery-08

Abstract

The goal of Application-Layer Traffic Optimization (ALTO) is to provide guidance to applications that have to select one or several hosts from a set of candidates capable of providing a desired resource. ALTO is realized by a client-server protocol. Before an ALTO client can ask for guidance it needs to discover one or more ALTO servers that can provide suitable guidance.

This document specifies a procedure for resource consumer initiated ALTO server discovery, which can be used if the ALTO client is embedded in the resource consumer.

## Terminology and Requirements Language

This document makes use of the ALTO terminology defined in [RFC5693].

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 September 22, 2013.

## Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                                                                               |    |
|-------------------------------------------------------------------------------|----|
| 1. Introduction . . . . .                                                     | 4  |
| 2. ALTO Server Discovery Procedure Overview . . . . .                         | 5  |
| 3. ALTO Server Discovery Procedure Specification . . . . .                    | 6  |
| 3.1. Step 1: Retrieving the Domain Name . . . . .                             | 6  |
| 3.1.1. Step 1, Option 1: User input . . . . .                                 | 6  |
| 3.1.2. Step 1, Option 2: DHCP . . . . .                                       | 6  |
| 3.2. Step 2: U-NAPTR Resolution . . . . .                                     | 7  |
| 4. Deployment Considerations . . . . .                                        | 8  |
| 4.1. Issues with Home Gateways . . . . .                                      | 8  |
| 4.2. Issues with Multihoming, Mobility and Changing IP<br>Addresses . . . . . | 8  |
| 5. IANA Considerations . . . . .                                              | 10 |
| 6. Security Considerations . . . . .                                          | 11 |
| 6.1. General Security Considerations . . . . .                                | 11 |
| 6.2. Security Considerations for U-NAPTR . . . . .                            | 11 |
| 7. References . . . . .                                                       | 13 |
| 7.1. Normative References . . . . .                                           | 13 |
| 7.2. Informative References . . . . .                                         | 13 |
| Appendix A. Contributors List and Acknowledgments . . . . .                   | 15 |
| Authors' Addresses . . . . .                                                  | 16 |

## 1. Introduction

The goal of Application-Layer Traffic Optimization (ALTO) is to provide guidance to applications that have to select one or several hosts from a set of candidates capable of providing a desired resource [RFC5693]. ALTO is realized by a client-server protocol; see requirement AR-1 in [RFC6708]. Before an ALTO client can ask for guidance it needs to discover one or more ALTO servers that can provide suitable guidance.

This document specifies a procedure for resource consumer initiated ALTO server discovery, which can be used if the ALTO client is embedded in the resource consumer. In other words, this document tries to meet requirement AR-32 in [RFC6708] while AR-33 is out of scope. A different approach, which tries to meet requirement AR-33, i.e., third-party ALTO server discovery, is addressed in [I-D.kist-alto-3pdisc].

The ALTO protocol specification [I-D.ietf-alto-protocol] is based on HTTP and expects the discovery procedure to yield an HTTP(S) URI. Therefore, this procedure is based on U-NAPTR [RFC4848]. It tries to directly find one or more ALTO server(s) that can give suitable guidance to the ALTO client. Other schemes, such as discovering a random ALTO server (which might not be able to give suitable guidance to the client in question) and asking it to redirect the client to a better server, are not considered in this document.

A more detailed discussion of various options where to place the functional entities comprising the overall ALTO architecture can be found in [I-D.ietf-alto-deployments].



## 2. ALTO Server Discovery Procedure Overview

The ALTO protocol specification [I-D.ietf-alto-protocol] expects that the ALTO discovery procedure yields the HTTP(S) URI of the ALTO server's Information Resource Directory, which gives further information about the capabilities and services provided by that ALTO server.

The ALTO server discovery procedure is performed in two steps:

1. One or - in case of multiple interfaces and/or IPv4/v6 dual stack operation - more DNS domain names are yielded, either by manual input or by means of DHCP.
2. These DNS domain names are used for U-NAPTR lookups yielding one or more URIs. Further DNS lookups may be necessary to determine the ALTO server's IP address(es).

The primary means for retrieving the DNS domain name is DHCP. However, there may be situations where DHCP is not available or does not return a suitable value. Furthermore, there might be situations in which the user wishes to override the value that could be retrieved from DHCP. In these situations, manual input may be used.

Typically, but not necessarily, the DNS domain name is the domain name in which the client is located, i.e., a PTR lookup on the client's IP address would yield a similar name. However, due to the widespread use of network address translation (NAT), trying to determine the DNS domain name through a PTR lookup on an interface's IP address is not recommended for resource consumer initiated ALTO server discovery.

### 3. ALTO Server Discovery Procedure Specification

As already outlined in Section 2 the ALTO server discovery procedure is performed in two steps, which will be specified in Section 3.1 and Section 3.2, respectively.

#### 3.1. Step 1: Retrieving the Domain Name

##### 3.1.1. Step 1, Option 1: User input

A user may want to use an ALTO service instance provided by an entity that is not the operator of the underlying IP network. Therefore, we allow the user to specify a DNS domain name, for example in a configuration file option. An example domain name is:

```
my-alternative-alto-provider.example.org
```

In case no ALTO-specific NAPTR records are found, we consider the discovery process based on user input as failed. A client MAY try to continue with DHCP (see below). If DHCP-based discovery succeeds the software SHOULD inform the user that the user input has been ignored and replaced by information retrieved from the network.

##### 3.1.2. Step 1, Option 2: DHCP

As a second option network operators may configure the domain name to be used for service discovery within an access network using DHCP.

RFC 5986 [RFC5986] defines DHCP IPv4 and IPv6 access network domain name options to identify a domain name that is suitable for service discovery within the access network. RFC 2132 [RFC2132] defines the DHCP IPv4 domain name option. While this option is less suitable, it still may be useful if the RFC 5986 option is not available.

For IPv6, the ALTO server discovery procedure MUST try to retrieve DHCP option 57 (OPTION\_V6\_ACCESS\_DOMAIN). If no such option can be retrieved the procedure fails for this interface. For IPv4, the ALTO server discovery procedure MUST try to retrieve DHCP option 213 (OPTION\_V4\_ACCESS\_DOMAIN). If no such option can be retrieved, the procedure SHOULD try to retrieve option 15 (Domain Name). If neither option can be retrieved the procedure fails for this interface. If a result can be retrieved it will be used as an input for the next step (U-NAPTR resolution). One example result could be:

```
example.net
```

### 3.2. Step 2: U-NAPTR Resolution

The first step of the ALTO server discovery procedure (see Section 3.1) yielded one or - in case of multiple interfaces and/or IPv4/v6 dual stack operation - several domain names, which will be used as U-NAPTR/DDDS (URI-Enabled NAPTR/Dynamic Delegation Discovery Service) [RFC4848] application unique strings. An example is:

example.net

In the second step, the ALTO Server discovery procedure uses a U-NAPTR [RFC4848] lookup with the "ALTO" Application Service Tag and either the "http" or the "https" Application Protocol Tag to obtain one or more URIs (indicating protocol, host and possibly path elements) for the ALTO server's Information Resource Directory. In this document, only the HTTP and HTTPS URI 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 result can be any valid HTTP(S) URI.

The following two U-NAPTR resource records can be used for mapping "example.net" to the HTTPS URI `https://altoserver.example.net/secure/directory` or the HTTP URI `http://altoserver.example.net/directory`, with the former being preferred.

example.net.

```
IN NAPTR 100 10 "u" "ALTO:https"  
"!.*!https://altoserver.example.net/secure/directory!" ""
```

```
IN NAPTR 200 10 "u" "ALTO:http"  
"!.*!http://altoserver.example.net/directory!" ""
```

If no ALTO-specific U-NAPTR records can be retrieved, the discovery procedure fails for this domain name (and the corresponding interface and IP protocol version). If further domain names yielded by Step 1 are known, the discovery procedure may perform the corresponding U-NAPTR lookups immediately. However, before retrying a lookup that has failed, a client MUST wait a time period that is appropriate for the encountered error (NXDOMAIN, timeout, etc.).

## 4. Deployment Considerations

### 4.1. Issues with Home Gateways

Section 3.1.2 describes the usage of a DHCP option. It enables the network operator of the network, in which the ALTO client is located, to provide a DNS domain name. 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 unknown DHCP options are not passed through some home gateways, effectively eliminating the DHCP option.

Another well-known issue is the usage of home gateway specific DNS domain names which "override" the DNS domain name provided by the network operator. For instance, a host behind a home gateway may receive a DNS domain name ".local" instead of "example.net". In general, this domain name is not usable for the server discovery procedure, unless a DNS server in the home gateway resolves the corresponding NAPTR lookup correctly, e.g., by means of a DNS split horizon approach.

### 4.2. Issues with Multihoming, Mobility and Changing IP Addresses

If the user decides to enter the DNS domain name manually, only one set of ALTO servers will be discovered, irrespectively of multihoming and mobility. Particularly in mobile scenarios this can lead to undesirable results.

The DHCP-based discovery method can discover different sets of ALTO servers for each interface and address family (i.e., IPv4/v6). In general, if a client wishes to communicate using one of its interfaces and using a specific IP address family, it SHOULD query the ALTO server(s) that have been discovered for this specific interface and address family. Selecting an interface and IP address family, as well as comparing results returned from different ALTO servers, is out of the 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 re-run the ALTO server discovery procedure without relying on earlier gained information.

There are several challenges with DNS on hosts with multiple interfaces [RFC6418], which can affect the ALTO server discovery. If the DNS resolution is performed on the wrong interface, it can return an ALTO server that could provide sub-optimal or wrong guidance. Finding the best ALTO server for multi-interfaced hosts is outside the scope of this document.

When using Virtual Private Network (VPN) connections there is usually no DHCP. The user has to enter the DNS domain name manually. For good optimization results, a DNS domain name corresponding to the VPN concentrator, not corresponding to the user's current location, has to be entered. Similar considerations apply for Mobile IP.

## 5. IANA Considerations

IANA is requested to register the following U-NAPTR [RFC4848] application service tag for ALTO:

Application Service Tag: ALTO

Intended usage: see [RFC5693] or: "The goal of Application-Layer Traffic Optimization (ALTO) is to provide guidance to applications that have to select one or several hosts from a set of candidates capable of providing a desired resource."

Defining Publication: The specification contained within this document

Contact information: The authors of this document

Author/Change controller: The IESG

Interoperability considerations: No interoperability issues are known or expected. This tag is to be registered specifically for ALTO, which is a new application without any legacy deployments.

Security considerations: see Section 6 and in particular Section 6.2 of this document.

Related publications: This document specifies a procedure for discovering an HTTP or HTTPS URI of an ALTO server. HTTP is specified in [RFC2616] and HTTPS is specified in [RFC2818]. The HTTP(S)-based ALTO protocol is specified in [I-D.ietf-alto-protocol].

Application Protocol Tag: This document specifies how to use the application service tag "ALTO" with the application protocol tags "http" (defining publication: [RFC2616] and "https" (defining publication: [RFC2818]), which have already been registered in the respective IANA registry. Therefore, IANA is not requested by this document to register any new application protocol tag.

## 6. Security Considerations

### 6.1. General Security Considerations

There are two different failures for the ALTO server discovery, which can both be caused by malicious attacks or by configuration problems, e.g., in case of DNS configuration errors or multi-homed hosts.

First, the discovery might not be able to discover an ALTO server, even if a suitable ALTO server exists. In that case, ALTO guidance will not be used. The resulting application performance and traffic distribution will subsequently correspond to a deployment scenario without ALTO guidance.

Second, the discovery procedure may discover a sub-optimal or wrong ALTO server. Such an ALTO server may either not be able to provide information for a given resource consumer (e.g., behind a NAT), thus rendering the ALTO service useless. Alternatively, said ALTO server may provide suboptimal or forged information. In the latter case, attackers could try to use ALTO to affect the traffic distribution or the performance of applications. Users may then observe performance problems, and network operators could detect traffic anomalies. A potential counter-measure is to disable the use of the ALTO service.

Security issues of ALTO in general and potential solutions are also discussed in [I-D.ietf-alto-protocol].

### 6.2. Security Considerations 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.



## 7. References

### 7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, March 1997.
- [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.
- [RFC4848] Daigle, L., "Domain-Based Application Service Location Using URIs and the Dynamic Delegation Discovery Service (DDDS)", RFC 4848, April 2007.
- [RFC5986] Thomson, M. and J. Winterbottom, "Discovering the Local Location Information Server (LIS)", RFC 5986, September 2010.

### 7.2. Informative References

- [I-D.ietf-alto-deployments] Stiemerling, M., Kiesel, S., and S. Previdi, "ALTO Deployment Considerations", draft-ietf-alto-deployments-06 (work in progress), February 2013.
- [I-D.ietf-alto-protocol] Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-14 (work in progress), February 2013.
- [I-D.kist-alto-3pdisc] Kiesel, S., Krause, K., and M. Stiemerling, "Third-Party ALTO Server Discovery (3pdisc)", draft-kist-alto-3pdisc-02 (work in progress), February 2013.
- [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.
- [RFC4033] Arends, R., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005.

- [RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, October 2009.
- [RFC6418] Blanchet, M. and P. Seite, "Multiple Interfaces and Provisioning Domains Problem Statement", RFC 6418, November 2011.
- [RFC6708] Kiesel, S., Previdi, S., Stiemerling, M., Woundy, R., and Y. Yang, "Application-Layer Traffic Optimization (ALTO) Requirements", RFC 6708, September 2012.

## Appendix A. Contributors List and Acknowledgments

The initial version of this document was co-authored by Marco Tomsu.

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.

Olafur Gudmundsson provided an excellent DNS expert review on an earlier version of this document.

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.

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 CHANGE project (<http://www.change-project.eu>), a research project supported by the European Commission under its 7th Framework Program (contract no. 257422). 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 CHANGE project or the European Commission.

## Authors' Addresses

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

Email: [ietf-alto@skiesel.de](mailto:ietf-alto@skiesel.de)  
URI: <http://www.rus.uni-stuttgart.de/nks/>

Martin Stiernerling  
NEC Laboratories Europe  
Kurfuerstenanlage 36  
Heidelberg 69115  
Germany

Phone: +49 6221 4342 113  
Email: [martin.stiernerling@neclab.eu](mailto:martin.stiernerling@neclab.eu)  
URI: <http://ietf.stiernerling.org>

Nico Schwan  
Stuttgart, Germany

Email: [ietf@nico-schwan.de](mailto:ietf@nico-schwan.de)

Michael Scharf  
Alcatel-Lucent Bell Labs  
Lorenzstrasse 10  
Stuttgart 70435  
Germany

Email: [michael.scharf@alcatel-lucent.com](mailto:michael.scharf@alcatel-lucent.com)  
URI: [www.alcatel-lucent.com/bell-labs](http://www.alcatel-lucent.com/bell-labs)

Haibin Song  
Huawei

Email: [melodysong@huawei.com](mailto:melodysong@huawei.com)



ALTO  
Internet-Draft  
Intended status: Standards Track  
Expires: December 29, 2013

S. Kiesel  
University of Stuttgart  
R. Penno  
Cisco Systems  
June 27, 2013

ALTO Server Discovery based on well-known IP Address  
draft-kiesel-alto-ip-based-srv-disc-02

Abstract

The goal of Application-Layer Traffic Optimization (ALTO) is to provide guidance to applications that have to select one or several hosts from a set of candidates capable of providing a desired resource. ALTO is realized by a client-server protocol.

This document establishes a well-known IP address for the ALTO service and specifies how ALTO clients embedded in the resource consumer can use it to access the ALTO service.

## Terminology and Requirements Language

This document makes use of the ALTO terminology defined in RFC 5693 [RFC5693].

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 December 29, 2013.

## Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                                                                       |    |
|-----------------------------------------------------------------------|----|
| 1. Introduction . . . . .                                             | 4  |
| 2. ALTO Server Discovery based on well-known IP Address . . . . .     | 5  |
| 2.1. Well-Known ALTO Server Discovery IP Address (WkAsdIPa) . . . . . | 5  |
| 2.2. Well-Known ALTO Server Discovery URIs (WkAsdURI) . . . . .       | 5  |
| 2.3. ALTO Discovery Client behavior . . . . .                         | 5  |
| 2.4. ALTO Discovery Server behavior . . . . .                         | 6  |
| 3. Deployment Considerations . . . . .                                | 7  |
| 4. IANA Considerations . . . . .                                      | 8  |
| 4.1. Registration of IPv4 Special Purpose Address . . . . .           | 8  |
| 4.2. Registration of IPv6 Special Purpose Address . . . . .           | 9  |
| 5. Security Considerations . . . . .                                  | 11 |
| 6. References . . . . .                                               | 12 |
| 6.1. Normative References . . . . .                                   | 12 |
| 6.2. Informative References . . . . .                                 | 12 |
| Authors' Addresses . . . . .                                          | 14 |



## 1. Introduction

The goal of Application-Layer Traffic Optimization (ALTO) is to provide guidance to applications that have to select one or several hosts from a set of candidates capable of providing a desired resource [RFC5693]. ALTO is realized by a client-server protocol, see requirement AR-1 in [RFC6708]. An HTTP based ALTO client protocol is specified in [I-D.ietf-alto-protocol].

Before an ALTO client can ask for guidance it needs to discover one or more ALTO servers that can provide suitable guidance. Several algorithms have been specified that produce a suitable HTTP URI for a given ALTO client (i.e., the URI may vary for different clients or different points of network attachment, etc.). These approaches are based on user input or DHCP [I-D.ietf-alto-server-discovery], a "reverse DNS" (PTR) lookup [I-D.kist-alto-3pdisc], or redirection within the application protocol [I-D.kiesel-alto-alto4alto]. However, each of these approaches has technical or operational issues that will hinder the fast deployment of ALTO.

This document follows a different approach: it establishes a well-known address for the ALTO service (TBD: this approach could easily be generalized in order to discover other services as well. But this is for further study). All ALTO clients seeking ALTO guidance are expected to send requests to this address. It is then the duty of "the network" to direct the query to a suitable server. This (re-)directing could be done on several layers, e.g., by resolving a well-known DNS domain name to different IP addresses (DNS split horizon), or by routing IP packets with the well-known IP address to different servers. This document follows the second option, as ALTO is closely related to IP routing and routing costs.

This document specifies a procedure that can be used if the ALTO client is embedded in the resource consumer. In other words, this document tries to meet requirement AR-32 in [RFC6708] while AR-33 is out of scope. Note that AR-20 mandates that "an ALTO client protocol must be designed in a way that the ALTO service can be provided by an entity that is not the operator of the underlying IP network." Though not violating said requirement, the procedure specified here is not helpful to fulfill it.

A more detailed discussion of various options where to place the functional entities comprising the overall ALTO architecture can be found in [I-D.ietf-alto-deployments].

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

## 2. ALTO Server Discovery based on well-known IP Address

### 2.1. Well-Known ALTO Server Discovery IP Address (WkAsdIPa)

IANA is requested to register (see Section 4) a single IPv4 address 192.0.0.X (TBD) and a single IPv6 address 2001:YYYY::ZZZZ (TBD) within the respective Special Purpose Address Registries as the well-known IP anycast addresses for the ALTO service. These addresses are called WkAsdIPa (well-known ALTO server discovery IP address(es)) in this document.

### 2.2. Well-Known ALTO Server Discovery URIs (WkAsdURI)

The Well-Known ALTO Server Discovery URIs (WkAsdURI) are formed using the HTTP or HTTPS protocol identifier, the WkAsdIPa in their literal forms (for literal IPv6 addresses in URIs see [RFC2732]), and a constant suffix. That is, there are four WkAsdURIs (TBD: replace X, Y, Z with real values assigned by IANA):

`http://192.0.0.X/alto`

`https://192.0.0.X/alto`

`http://[2001:YYYY::ZZZZ]/alto`

`https://[2001:YYYY::ZZZZ]/alto`

### 2.3. ALTO Discovery Client behavior

ALTO Clients that need to discover an ALTO server use the HTTP GET method [RFC2616] to access one WkAsdURI, e.g. GET `http://192.0.0.X/alto`. They MUST be prepared to receive an HTTP 307 temporary redirect to the ALTO server's Information Resource Directory URI (Sec. 6.7 of [I-D.ietf-alto-protocol]).

For hosts equipped with multiple interfaces and/or using IPv4/v6 dual stack, this discovery method might yield different Information Resource Directory URIs for each interface and address family (i.e., IPv4/v6). In general, if a client wishes to communicate using one of its interfaces and using a specific IP address family, it SHOULD use this interface and the IP address associated with this interface to access the WkAsdURI of the corresponding IP address family. Selecting an interface and IP address family, as well as comparing results returned from different ALTO servers, is out of the scope of this document.

TBD: rules for retrying (timers, etc.) in case of failure.

TBD: rules for caching discovery results.

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 re-run the ALTO server discovery procedure without relying on earlier gained information.

#### 2.4. ALTO Discovery Server behavior

ALTO discovery servers MUST listen on the WkAsdIPa on the HTTP and HTTPS ports for incoming HTTP(S) requests. They MUST answer GET requests to WkAsdURI using the 307 (Temporary Redirect) status code and redirect to an ALTO server's Information Resource Directory URI.

The ALTO discovery server MAY consider the client's address and other information when generating the reply, in order to redirect to different ALTO servers depending on the client's identity or location within the network topology.

The Information Resource Directory itself MUST NOT reside on a WkAsdIPa, and it MUST NOT reside on an URI that resolves via DNS to a WkAsdIPa. After issuing the 307 status code ALTO discovery servers MUST close the HTTP(S) connection.

Rationale for the requirements in the previous paragraph: The goal is to keep the TCP connection to the WkAsdIPa as short as possible. When using anycast routing, IP packets belonging to an established TCP connection could be diverted to another ALTO discovery server due to state changes in the routing protocol or due to scheduled maintenance. Keeping the connection duration as short as possible reduces the risk of stalled or aborted connections. A UDP based lookup using one query packet and one reply packet (e.g., based on httpu) would eliminate that risk. However, there seems not to be a well-standardized candidate protocol and studies [Levine2006] suggest that short-lived TCP connections work well enough with anycast routing.

TBD: do we need some URI such as `http://192.0.0.X/discovery-server-identity` in order to be able to identify the (misbehaving) discovery server that currently serves us?

TBD: how should the ALTO discovery server handle GET requests to other URIs or other HTTP methods?

TBD: should the discovery server always redirect http requests to the http URI of the information resource Directory and redirect https always to https? Or are there other reasonable scenarios?

### 3. Deployment Considerations

Network operators have to install one or more ALTO discovery servers as specified above. Depending on the the network deployment scenario they may use IP routing tables, HTTP proxies with URI rewriting, or other suitable mechanisms to direct GET-requests for a WkAsdURI to one of these servers.

[TBD: explain in more detail] This works fine even with cascaded access routers with NATs. After each router hop the operator may decide whether to handle the discovery requests, e.g., using a static routing table entry, or whether let them flow "automatically" towards the internet backbones using the default routing table entry.

TBD: what happens if an operator does not deploy these scheme? Requests could be dropped at administrative borders. As an alternative, there could be "public" discovery servers to answer all queries that had not been answered in the respective originating access network. These servers could use the third-party ALTO server discovery procedure [I-D.kist-alto-3pdisc] to find the redirection target based on the client's IP address.

[TBD: explain in more detail] The advantage of this scheme is that it does not need support in home gateways, which would harm quick deployment. This scheme also doesn't need new interfaces between the operating system and applications, e.g., for passing DHCP options from the operating system to the application.

#### 4. IANA Considerations

##### 4.1. Registration of IPv4 Special Purpose Address

IANA is requested to register a single IPv4 address in the IANA IPv4 Special Purpose Address Registry [RFC5736].

[RFC5736] itemizes some information to be recorded for all designations:

1. The designated address prefix.

Prefix: TBD by IANA. Prefix length: /32

2. The RFC that called for the IANA address designation.

This document.

3. The date the designation was made.

TBD.

4. The date the use designation is to be terminated (if specified as a limited-use designation).

Unlimited. No termination date.

5. The nature of the purpose of the designated address (e.g., unicast experiment or protocol service anycast).

protocol service anycast.

6. For experimental unicast applications and otherwise as appropriate, the registry will also identify the entity and related contact details to whom the address designation has been made.

N/A.

7. The registry will also note, for each designation, the intended routing scope of the address, indicating whether the address is intended to be routable only in scoped, local, or private contexts, or whether the address prefix is intended to be routed globally.

Typically used within a network operator's network domain, but in principle globally routable.

8. The date in the IANA registry is the date of the IANA action, i.e., the day IANA records the allocation.

TBD.

#### 4.2. Registration of IPv6 Special Purpose Address

IANA is requested to register a single IPv6 address in the IANA IPv6 Special Purpose Address Block [RFC4773].

[RFC4773] itemizes some information to be recorded for all designations:

1. The designated address prefix.

Prefix: TBD by IANA. Prefix length: /128

2. The RFC that called for the IANA address designation.

This document.

3. The date the designation was made.

TBD.

4. The date the use designation is to be terminated (if specified as a limited-use designation).

Unlimited. No termination date.

5. The nature of the purpose of the designated address (e.g., unicast experiment or protocol service anycast).

protocol service anycast.

6. For experimental unicast applications and otherwise as appropriate, the registry will also identify the entity and related contact details to whom the address designation has been made.

N/A.

7. The registry will also note, for each designation, the intended routing scope of the address, indicating whether the address is intended to be routable only in scoped, local, or private contexts, or whether the address prefix is intended to be routed globally.

Typically used within a network operator's network domain, but in principle globally routable.

8. The date in the IANA registry is the date of the IANA action, i.e., the day IANA records the allocation.

TBD.

## 5. Security Considerations

TBD

Issue: how to deal with TLS certificates for HTTPS?

TBD: rules for filtering route at administrative boundaries



## 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.
- [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.
- [RFC2732] Hinden, R., Carpenter, B., and L. Masinter, "Format for Literal IPv6 Addresses in URL's", RFC 2732, December 1999.
- [RFC4773] Huston, G., "Administration of the IANA Special Purpose IPv6 Address Block", RFC 4773, December 2006.
- [RFC5736] Huston, G., Cotton, M., and L. Vegoda, "IANA IPv4 Special Purpose Address Registry", RFC 5736, January 2010.

### 6.2. Informative References

- [I-D.ietf-alto-deployments]  
Stiemerling, M., Kiesel, S., and S. Previdi, "ALTO Deployment Considerations", draft-ietf-alto-deployments-06 (work in progress), February 2013.
- [I-D.ietf-alto-protocol]  
Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-16 (work in progress), May 2013.
- [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-08 (work in progress), March 2013.
- [I-D.kiesel-alto-alto4alto]  
Kiesel, S., "Using ALTO for ALTO server selection", draft-kiesel-alto-alto4alto-00 (work in progress), July 2010.
- [I-D.kist-alto-3pdisc]  
Kiesel, S., Krause, K., and M. Stiemerling, "Third-Party ALTO Server Discovery (3pdisc)", draft-kist-alto-3pdisc-03 (work in progress), May 2013.
- [Levine2006]

Levine, M., Lyon, B., and T. Underwood, "TCP Anycast - Don't believe the FUD. Operational experience with TCP and Anycast.", Presentation at NANOG37 <http://www.nanog.org/meetings/nanog37/presentations/matt.levine.pdf>, June 2006.

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

[RFC6708] Kiesel, S., Previdi, S., Stiernerling, M., Woundy, R., and Y. Yang, "Application-Layer Traffic Optimization (ALTO) Requirements", RFC 6708, September 2012.

Authors' Addresses

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

Email: [ietf-alto@skiesel.de](mailto:ietf-alto@skiesel.de)  
URI: <http://www.rus.uni-stuttgart.de/nks/>

Reinaldo Penno  
Cisco Systems  
170 West Tasman Dr  
San Jose CA  
USA

Email: [repenno@cisco.com](mailto:repenno@cisco.com)



ALTO  
Internet-Draft  
Intended status: Informational  
Expires: January 16, 2014

H. Song  
Y. Lee  
Huawei  
V. Lopez  
D. Lopez  
Telefonica I+D  
L. Deng  
W. Chen  
China Mobile  
July 15, 2013

Extension Use Cases and Requirements for ALTO  
draft-song-alto-usecase-ext-00

Abstract

This document describes new usecases for ALTO, and identifies its related requirements to extend the ALTO protocol. The use cases in this document include overlay routing, NaaS, data center information, and P2P cache.

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 16, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                                                           |    |
|-----------------------------------------------------------|----|
| 1. Introduction . . . . .                                 | 2  |
| 2. Terminology . . . . .                                  | 3  |
| 3. Use Cases and Requirements . . . . .                   | 3  |
| 3.1. Minor Extensions in ISP or Overlay Network . . . . . | 3  |
| 3.1.1. Overlay Routing . . . . .                          | 4  |
| 3.1.2. Inter NSP ASQ . . . . .                            | 5  |
| 3.1.3. Network As A Service (NaaS) . . . . .              | 6  |
| 3.1.4. P2P Cache . . . . .                                | 6  |
| 3.2. Data Center Network . . . . .                        | 9  |
| 3.2.1. Data Center Network Deployment . . . . .           | 9  |
| 3.2.2. VM Migration Between Data Centers . . . . .        | 10 |
| 4. References . . . . .                                   | 11 |
| 4.1. Normative References . . . . .                       | 11 |
| 4.2. Informative References . . . . .                     | 11 |
| Authors' Addresses . . . . .                              | 12 |

## 1. Introduction

ALTO protocol [I-D.ietf-alto-protocol] provides an interface to applications with appropriate information to guide an optimal node selection when there are more than one application nodes providing the same service. It usually aggregates network locations into PIDs, and assigns lower cost value for a PID pair that are topologically close. So when application node follows the advice from ALTO server to choose one resource provider with a PID that has lower cost from its own PID, with higher probability the application node can keep the content request and response traffic flow intra domain, which can reduce the suffering increasing interdomain traffic for ISPs, and avoid the congestion in the backbone network. More factors for node selection can be considered, such as pricing, congestion, and etc.

The existing ALTO protocol has its limitations too. For example, in a cost map it only gives one cost value between source PID and destination PID, assuming there is only one path between them. But it can be routed through different paths in overlay routing. So we propose to add a "via" parameter as an extension to the cost map. In this document, we give use cases first, and then the possible way to extend the ALTO protocol to achieve it.

## 2. Terminology

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]. And the following terms used in this document have their definitions below.

I2AEX: Infrastructure to Application Exposure.

ALTO: application layer traffic optimization. For ALTO protocol, please refer to . [I-D.ietf-alto-protocol]

IaaS: Infrastructure as a Service. One common IaaS service is leasing virtual machines with appropriate bandwidth to tenants.

NaaS: Networking as a service. The common NaaS services include dynamic VPN service, virtual network service, and etc.

AP: a wireless access point (WAP) is a device that allows wireless devices to connect to a wired network using WLAN. The AP usually connects to a AC as a standalone device, but it can also be an integral component of the router itself.

AC: a wireless access controller (WAC) is the network entity that provides wire access via APs to the network infrastructure in the data plane, control plane, management plane, or a combination therein.

Forwarding Cache: is a traditional content cache, which caches content flows from outside its coverage and serves subsequent requestors under its coverage for the content.

Reverse Cache: is a special content cache proposed for WLAN accessing networks, which caches content flows from inside its coverage and serves subsequent requests from outside its coverage for the content.

Bidirectional Cache: is a combination of a forwarding cache and a reverse cache.

Cooperative Cache: is a content cache deployed by network operators in cooperation with specific content delivering SPs (e.g. P2P streaming services), which participates in the overlay's service provision explicitly.

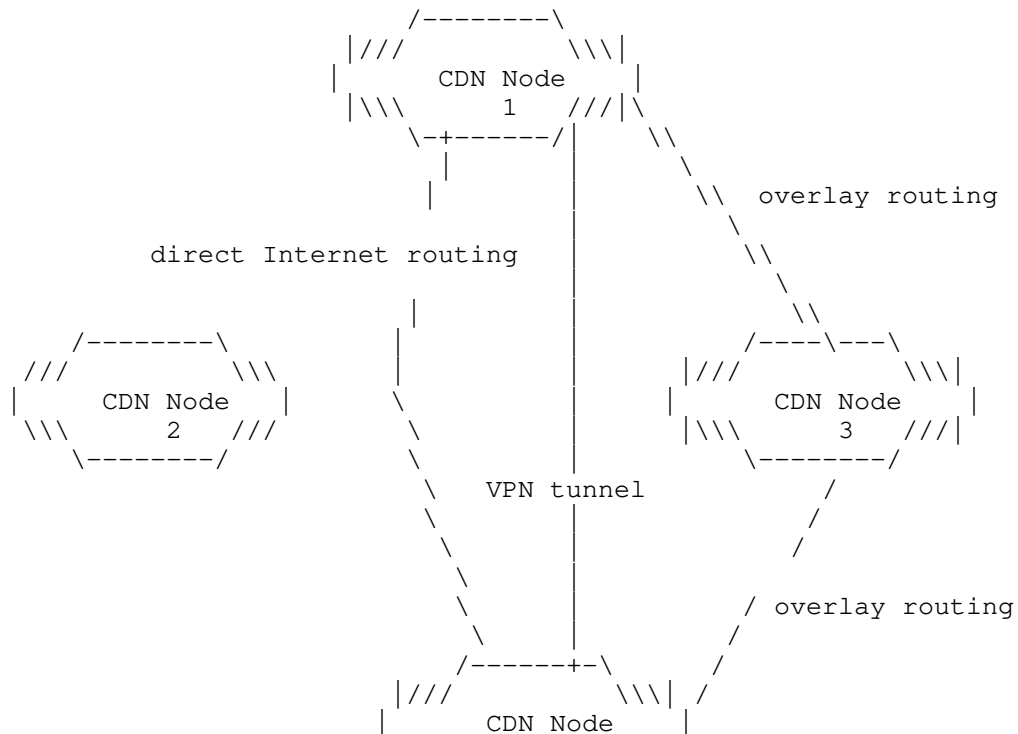
## 3. Use Cases and Requirements

### 3.1. Minor Extensions in ISP or Overlay Network

### 3.1.1. Overlay Routing

An overlay network is a computer network which is built on the top of another network. Nodes in the overlay can be thought of as being connected by virtual or logical links, each of which corresponds to a path, perhaps through many physical links, in the underlying network[overlay\_network]. One example of overlay network over IP network is CDN network. A CDN network consists of many CDN nodes with different levels. One edge CDN node often needs to pull content from another node that is in a higher distribution level position in the CDN topology. There usually can be several paths to send the content from the source CDN node to the edge CDN node. One way obviously is the direct IP routing. And if the direct routing path is not good, then the source CDN node will select another CDN node as the intermediate node to transport the content to the that destination edge node, which will be more efficiency than the direct routing path. Of course, there are usually more than one intermediate node available, and the source CDN node needs to select a "best" one.

In some cases, there can also be a VPN tunnel between two CDN nodes to transfer contents.





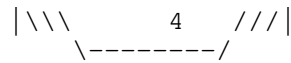


Figure 1. different ways for sending content from Node 1 to Node 4

So the transport between two overlay nodes can be from:

direct routing

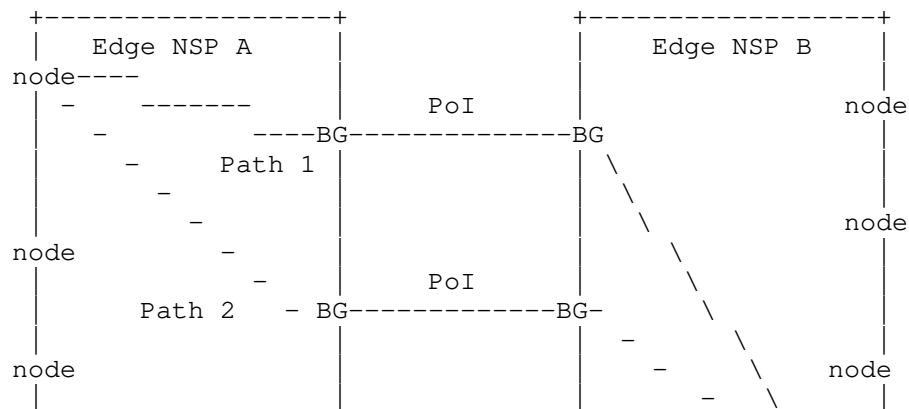
one/more intermediate overlay nodes

a VPN tunnel

The proposed extension is to add a "via" parameter to each cost value, and the value of the "via" parameter can be "direct routing", or location identifiers that can represent intermediate overlay nodes (such like IP address), or "VPN".

### 3.1.2. Inter NSP ASQ

This use case is similar to the overlay routing use case. When two ASes are connected through more than one pairs of border gateway routers, then there are more than one paths from a node in one of these two ASes to another node in the other AS. And these paths through different pair of border routers may have different service quality. An ALTO server can contemplate the service quality through the locations in one AS to its different boarder gateways, and between boarder gateways, and through the other boarder gateway to network locations in the other AS, and then provide guidance to applications to choose an appropriate path for the service routing.



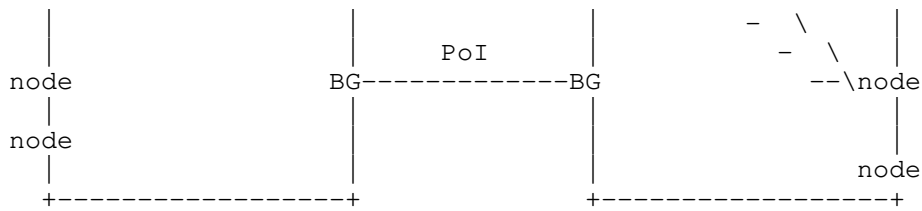


Figure 2 Inter-NSP ASQ use case

The "via" extension is also proposed to be used for this use case, and the value of it can be the identifier of the boarder gateway.

### 3.1.3. Network As A Service (NaaS)

Network As A Service (NaaS) enables network operators to give connectivity service to multiple users on top of the same physical infrastructure. This connectivity service can be offered to different customers, which have an interface to request for more bandwidth to the network in case they need more capacity. Although end users may have an interface to request for bandwidth to the network, an interface is required to disseminate to the end points with the changes in the network configuration. There are two options to disseminating information using ALTO protocol:

- o Dissemination of bandwidth information. ALTO can inform with a cost map related to unreserved bandwidth so end points can decide which connections they may use depending on the capacity in the connections. This can be used to update routing tables in the end points or priorities to interconnect two end points. Due to the dynamicity of traffic, this unreserved bandwidth is based on administrative reservations done through control plane protocols like RSVP-TE.
- o Bandwidth pricing. ALTO protocol can disseminate a cost map related to price of the connectivity between locations (such like PIDs). This information can be used to advertize customers, which is the cost to request for more bandwidth between two locations periodically. This can change depending on links utilization in the physical infrastructure. The cost advertize by ALTO is not directly the price charged to the customer, but a cost related to.

### 3.1.4. P2P Cache

Efforts have been put on using forwarding caches to reduce P2P traffic in cross domain scenarios, which demonstrates great

improvement in user experience and considerable cost reduction at interworking points. What's more, bidirectional caches are proposed to be deployed at the AC level for mitigation of undesirable downlink congestion caused by consistent uplink P2P traffic, as shown in Figure 5, the reverse cache can provide uploading service instead of the WLAN peers under the AC's coverage.

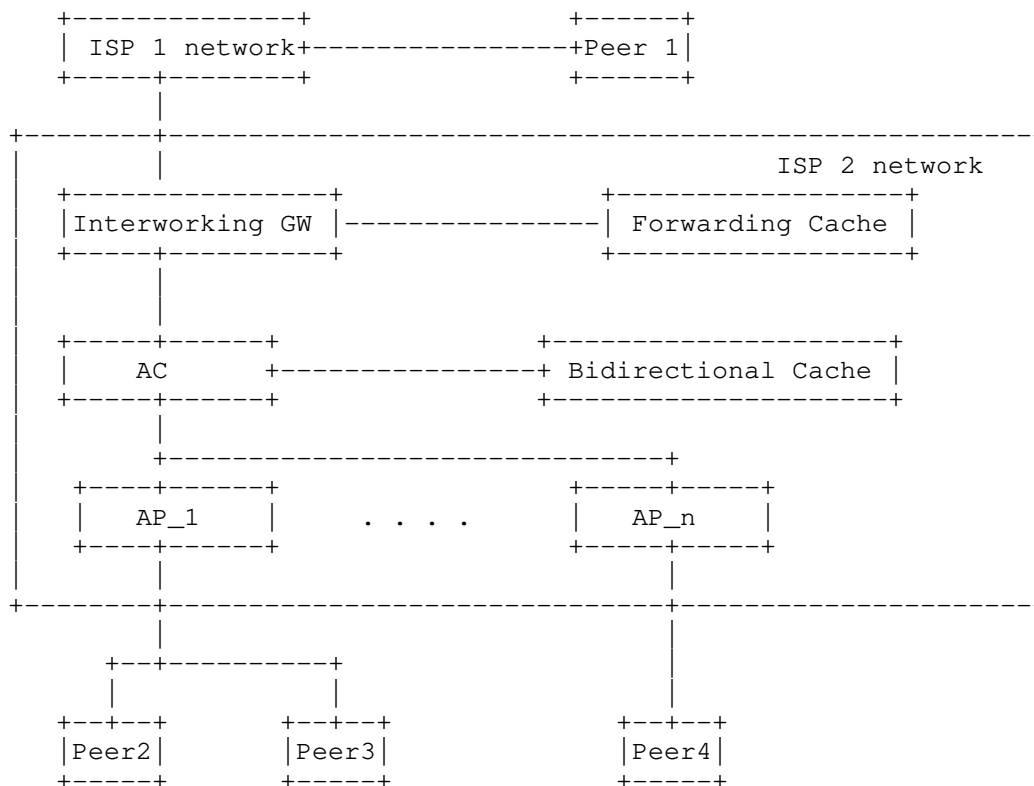


Figure 1: Architecture of T/B-cache in WLAN

With various P2P caches deployed, especially at a position as low as the AC-level, it could be sub-optimal to simply use the accessing network type as the divider for different PIDs and assign sufficient high cost within the wireless PID to prefer accessing remote peers over local peers blindly. Therefore, it is expected that the cooperation between the network operator and the P2P SP in building up cooperative caching system and sharing information through ALTO protocol about these facilities bring benefits to both parties.

A straightforward proposal would be to use locations of caches as dividers of different DIPs to further partition intra-ISP network domain and mark costs among them according to the location and type of relevant caches. However, as there is both CAPEX and OPEX expenditures for dedicated P2P Cache devices, it may be cost-efficient for caches to make buffering/serving decisions based on the popularity of the specific content. In addition, in cooperative mode, a P2P cache may be under the content scheduling of the specific P2P SP instead of the direct control of the network operator. How to expose this application-relevant information to ALTO under such context is an open issue.

Luckily, in the cooperative-mode, a cache is playing as a normal peer under the tracker, and the latter can make the "right" decision in choosing in favor of the former under the guidance of the ALTO response while the tracker itself would take care of the content availability problem. If the cache doesn't have the content in question, it would no appear in the peer list handed in to ALTO server by the tracker.

In this case, the ALTO server can collect the information about caching sub-system in the network, identify those "caching" peers in the peer list of an cost request from an ALTO client, and arrange the returned rank list accordingly. For example, a simple candidate-ranking policy for a cost query to a WLAN peer, could be caching peers at the begining, then inside wired peers, and lastly outside wired peers.

Moreover, the P2P SP and WLAN network operator may benefit even more by group popular files accroding to peers' geographic location or access types, and adapt its internal caching scheduling decisions about which files to be cached on which spot. In other words, it would be helpful that the ALTO server provides the client with the requesting peer's subscription types (i.e. wired/WLAN/ cellular/...) as well as geographic locations.

The proposed extension to ALTO is to distinguish peers not only according to IP prefixes, but also peer's access types and whether it's a caching server or not. This kind of information can be acquired through network management system or application system. And it also requires that for endpoint property lookup, "exact matching" has higher priority than "IP prefix matching".

### 3.2. Data Center Network

#### 3.2.1. Data Center Network Deployment

Infrastructure as a service (IaaS) is a way how the data center provides its services. There are different kinds of resources in a data center, physical machines, virtual machines, switches, firewalls, computing power, storage space, and electric power. The draft [I-D.lee-alto-ext-dc-resource] proposes collecting data center resource information to make use of such information for a key decision to allocate the application request to an "optimal" Data Center location in which to host the application request. Key constraints in this decision include resource availability (e.g., memory, storage, CPU, etc.), DC network cost, DC network resource constraints (e.g., bandwidth), structure constraints (e.g., Data Center power consumption) and others.

Combined computing and network resource optimization is of value to both application owners and data center operators. For example a data center operator with multiple buildings in a metropolitan area may also want to balance compute and network costs.

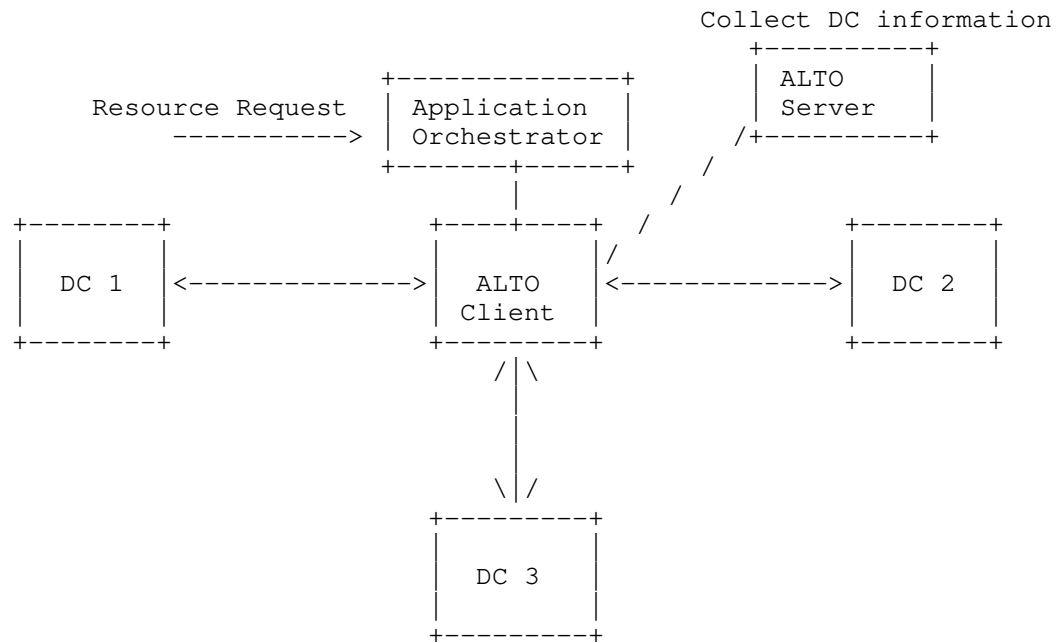


Figure 3 Data Center Resource Deployment use case

The intended ALTO protocol extension is going to provide the following information:

- Data Center Identifier (DCI)
- Data Center Location Identifier (e.g., IP address of the gateway node)
- Time Stamp
- Abstracted Memory Usage
- Abstracted CPU Level
- Abstracted Power Consumption Level
- DC Network cost
- DC Network resource constraints

### 3.2.2. VM Migration Between Data Centers

Giant or large applications usually have to rent virtual machine resources in more than one data centers for its application deployment. These virtual machines do not only communicate with the end users, but also with other virtual machines. Some applications rent dedicated VPN links for the traffic among data centers, and some applications pay money for the traffic among data centers that go through Internet. There is a requirement to collect each VM traffic pattern and direction among data centers, and consider them together with the traffic pricing information, and use some specific algorithm, to give advice on VM migration. For example, the algorithm may let the VMs that have much communication traffic migrate into one data center, so as to reduce the traffic among data centers, and save money for the application.

A new "cost type" extension is proposed in ALTO, which represents the cost between VMs, with regarding to the combined traffic volume and pricing information. An application uses ALTO client to retrieve this kind of information, and consider it together with the location and any specific constraints of each VM, then decide whether to migrate VMs that have high cost volume between each other into one single data center, so as to reduce inter-DC traffic.

The actual use case is far more complex than the figure below. Because each VM may communicate with multiple VMs, and a more complex algorithm should be used for the VM migration. The application have to compute the new total cost after the migration and compare it with that before the migration.



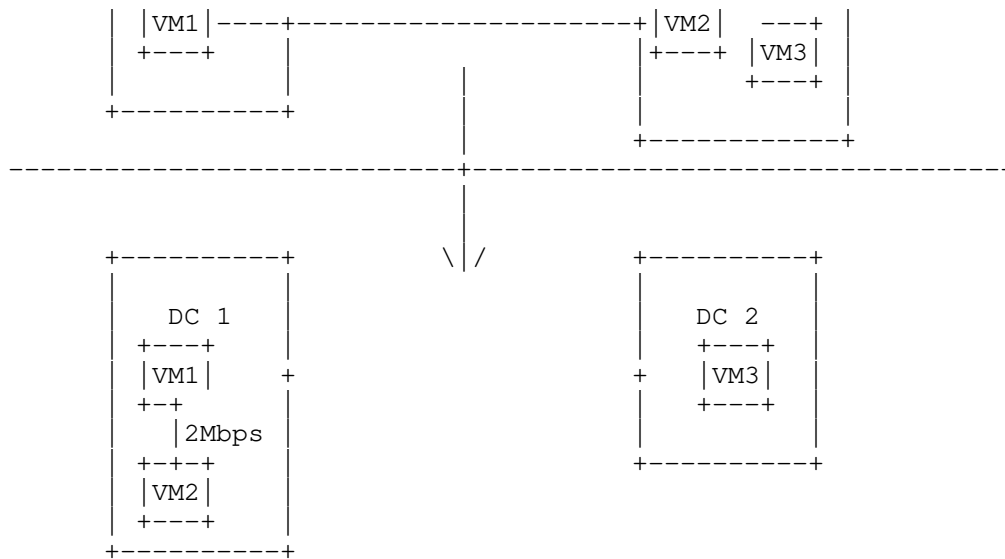


Figure 4: Inter-DC VM migration use case

## 4. References

### 4.1. Normative References

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

[overlay\_network]  
 , "overlay network", .

[http://en.wikipedia.org/wiki/Overlay\\_network](http://en.wikipedia.org/wiki/Overlay_network)

### 4.2. Informative References

[I-D.lee-alto-ext-dc-resource]  
 Lee, Y., Bernstein, G., and D. Dhody, "ALTO Extensions for Collecting Data Center Resource Information", draft-lee-alto-ext-dc-resource-02 (work in progress), July 2013.

[I-D.ietf-alto-protocol]  
 Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-16 (work in progress), May 2013.

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

[I-D.ietf-alto-deployments]  
Stimerling, M., Kiesel, S., and S. Previdi, "ALTO Deployment Considerations", draft-ietf-alto-deployments-06 (work in progress), February 2013.

#### Authors' Addresses

Haibin Song  
Huawei

Email: haibin.song@huawei.com

Young Lee  
Huawei

Email: leeyoung@huawei.com

Victor Lopez  
Telefonica I+D

Email: vlopez@tid.es

Diego R. Lopez  
Telefonica I+D

Email: diego@tid.es

Lingli Deng  
China Mobile

Email: denglingli@chinamobile.com

Wei Chen  
China Mobile

Email: chenwei@chinamobile.com



Application-Layer Traffic Optimization  
Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 9, 2014

Q. Wu  
L. Xia  
Huawei  
July 8, 2013

JSON Format Extensions for Traffic Engineering (TE) performance metrics  
in the ALTO Information Resource Directory  
draft-wu-alto-json-te-00

## Abstract

The base ALTO specification defines two properties for cost metric attribute in the Cost MAP, including 'hopcount' and 'routingcost'. This specification adds five new properties and one new parameter for Traffic Engineering(TE) performance related constraint attribute associated with cost metric attribute 'routingcost' in the ALTO Information Resource Directory: Link Delay, Delay Variation, Packet Loss, Residual Bandwidth, Available Bandwidth, linkstate.

## 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 9, 2014.

## Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                                                 |    |
|-------------------------------------------------|----|
| 1. Introduction . . . . .                       | 3  |
| 2. Conventions used in this document . . . . .  | 4  |
| 3. Cost Metric Extensions: properties . . . . . | 5  |
| 3.1. property: linkdelay . . . . .              | 5  |
| 3.2. property: linkjitter . . . . .             | 6  |
| 3.3. property: linkloss . . . . .               | 7  |
| 3.4. property: residualbandwidth . . . . .      | 8  |
| 3.5. property: availablebandwidth . . . . .     | 9  |
| 4. Cost Metric Extensions: Parameters . . . . . | 11 |
| 4.1. parameter: linkstate . . . . .             | 11 |
| 5. Security Considerations . . . . .            | 13 |
| 6. IANA Considerations . . . . .                | 14 |
| 7. Normative References . . . . .               | 15 |
| Authors' Addresses . . . . .                    | 16 |

## 1. Introduction

The ALTO protocol [I.D-ietf-alto-protocol] uses a REST-ful design [Fielding-Thesis], and encodes its requests and responses using JSON. In ALTO architecture [I.D-ietf-alto-protocol], the ALTO server allows alto information to be gathered from multiple systems (e.g., routing protocol). [I.D-ietf-ospf-te-metric-extensions] describes extensions to OSPF TE called "OSPF TE Metric Extensions", that can be used to distribute network performance information (such as link delay, delay variation, packet loss, residual bandwidth, and available bandwidth). The mechanism defined in [I.D-ietf-ospf-te-metric-extensions] can be used by an ALTO Server to retrieve the necessary performance information supplementing the prefix and network topology data gathered from other sources in the underlying network.

In the ALTO Information Resource Directory, Network and Cost Map are two core ALTO Information provided to clients. The TE performance metric can be represented using Cost MAP. The base ALTO specification [I.D-ietf-alto-protocol] defines one typical cost metric attribute for Cost Type in the Cost MAP (i.e., 'routingcost') and uses constraint attribute to list additional constraints to which elements of the Cost Map are related. This specification adds five new properties and one new parameter for constraint attribute associated with 'routingcost' cost metric attribute in alto information service: Link Delay, Link Jitter, Packet Loss, Residual Bandwidth, Available Bandwidth, linkstate.

## 2. Conventions used in this document

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 [RFC2119].

Syntax specifications shown here use the augmented Backus-Naur Form (ABNF) as described in [RFC5234], and are specified as in the base JSON specification [RFC4627].

### 3. Cost Metric Extensions: properties

#### 3.1. property: linkdelay

Namespace:

property name: linkdelay

Purpose: To specify the average link delay between two directly connected neighboring peers in the network.

Value type: A single number value containing an integer component that may be prefixed with an optional minus sign, which may be followed by a fraction part and/or an exponent part.

Cardinality: \*1

Member parameters: N/A

Description: This is intended to be a cost constraint attribute used together with cost metric attribute 'routingcost'. 'routingcost' may also be used with other cost constraint attributes that is used to specify cost constraints. If 'linkdelay' is present, 'routingcost' MUST have at most one 'linkdelay'.

Format definition:

```
LINKDELAY-param =  
"VALUE"("gt"/"lt"/"eq"/"ge"/"le") ("number" / "object")  
LINKDELAY-value = number / object  
; Value type and VALUE parameter MUST match.
```

Examples:

```
"data": {  
  "cost type": {  
    "cost-mode": "numerical",  
    "cost-metric": "routingcost",  
    "constraints" : {"linkdelay"},  
    "endpoints": {  
      "srcs": [ "ipv4:192.0.2.2" ],  
      "dsts": [  
        "ipv4:192.0.2.89",  
        "ipv4:198.51.100.34",  
        "ipv4:203.0.113.45"  
      ]  
    }  
  }  
  "map": {  
    "ipv4:192.0.2.2": {
```

```
"ipv4:192.0.2.89": 0.0[linkdelay eq 0.0],
"ipv4:198.51.100.34": 15.0[linkdelay eq 3.0],
"ipv4:203.0.113.45": 1.0[linkdelay eq 12.0],
  }
}
```

### 3.2. property: linkjitter

Namespace:

Property name: linkjitter

Purpose: To specify the average link delay variation between two directly connected neighboring peers.

Value type: A single number value containing an integer component that may be prefixed with an optional minus sign, which may be followed by a fraction part and/or an exponent part.

Cardinality: \*1

Member parameters: N/A

Description: This is intended to be a constraint attribute value used together with 'routingcost' cost metric attribute. 'routingcost' may also be used with other cost constraint attributes that is used to specify cost constraints. If 'linkjitter' is present, 'routingcost' MUST have at most one 'linkjitter'.

Format definition:

```
LINKJITTER-param =
"VALUE"("gt"/"lt"/"eq"/"ge"/"le") ("number" / "object")
LINKJITTER-value = number / object
; Value type and VALUE parameter MUST match.
```

Examples:

```
"data": {
  "cost type": {
    "cost-mode": "numerical",
    "cost-metric": "routingcost",
    "constraints" : {"linkdelay", "linkjitter"}
    "endpoints": {
      "srcs": [ "ipv4:192.0.2.2" ],
      "dsts": [
        "ipv4:192.0.2.89",
```

```

        "ipv4:198.51.100.34",
        "ipv4:203.0.113.45"
      ]
    }
    "map": {
      "ipv4:192.0.2.2": {
        "ipv4:192.0.2.89": 0[linkdelay eq0.0,linkjitter eq0.00],
        "ipv4:198.51.100.34": 5[linkdelay eq3.0,linkjitter eq1.0],
        "ipv4:203.0.113.45": 2[linkdelay eq12.0,linkjitter eq5.0],
      }
    }
  }

```

### 3.3. property: linkloss

Namespace:

Property name: linkloss

Purpose: To specify a percentage of the total traffic sent over a configurable interval between two directly connected neighboring peers.

Value type: A single number value containing an integer component that may be prefixed with an optional minus sign, which may be followed by a fraction part and/or an exponent part.

Cardinality: \*1

Format definition: This is intended to be a constraint attribute value used together with 'routingcost' cost metric attribute. 'routingcost' may also be used with other cost constraint attributes that is used to specify cost constraints. If 'linkloss' is present, 'routingcost' MUST have at most one 'linkloss'.

Format definition:

```

LINKLOSS-param =
"VALUE"("gt"/"lt"/"eq"/"ge"/"le") ("number" / "object")
LINKLOSS-value = number / object
; Value type and VALUE parameter MUST match.

```

Examples:

```

"data": {
  "cost type": {
    "cost-mode": "numerical",
    "cost-metric": "routingcost",
    "constraints": { "linkloss" },
    "endpoints": {

```

```

        "srcs": [ "ipv4:192.0.2.2" ],
        "dsts": [
            "ipv4:192.0.2.89",
            "ipv4:198.51.100.34",
            "ipv4:203.0.113.45"
        ]
    }
    "map": {
        "ipv4:192.0.2.2": {
            "ipv4:192.0.2.89": 0 [linkloss eq0],
            "ipv4:198.51.100.34": 1 [linkloss eq0.0001],
            "ipv4:203.0.113.45": 0 [linkloss eq0],
        }
    }

```

### 3.4. property: residualbandwidth

Namespace:

Property name: residualbandwidth

**Purpose:** To specify Maximum Link Bandwidth minus the bandwidth currently allocated between two directly connected neighboring peers. For a link, residual bandwidth is defined to be Maximum Bandwidth minus the bandwidth currently allocated to RSVP-TE packets. For a bundled link, residual bandwidth is defined to be the sum of the component link residual bandwidths.

**Value type:** A single number value containing an integer component that may be prefixed with an optional minus sign, which may be followed by a fraction part and/or an exponent part.

**Cardinality:** \*1

**Member parameters:** N/A

**Description:** This is intended to be a constraint attribute value used together with 'routing cost' cost metric attribute. 'routingcost' may also be used with other cost constraint attributes that is used to specify cost constraints. If 'residualbw' is present, 'routingcost' MUST have at most one 'residualbw'.

**Format definition:**

```

RESIDUALBANDWIDTH-param =
"VALUE"("gt"/"lt"/"eq"/"ge"/"le") ("number" / "object")

```



RESIDUALBANDWIDTH-value = number / object  
; Value type and VALUE parameter MUST match.

Examples:

```
"data": {
  "cost type": {
    "cost-mode": "numerical",
    "cost-metric": "routingcost",
    "constraints": { "residbw" },
    "endpoints": {
      "srcs": [ "ipv4:192.0.2.2" ],
      "dsts": [
        "ipv4:192.0.2.89",
        "ipv4:198.51.100.34",
        "ipv4:203.0.113.45"
      ]
    }
  }
  "map": {
    "ipv4:192.0.2.2": {
      "ipv4:192.0.2.89": 0[residbw eq0.000000],
      "ipv4:198.51.100.34": 5[residbw eq12.5],
      "ipv4:203.0.113.45": 2[residbw eq5.9],
    }
  }
}
```

### 3.5. property: availablebandwidth

Namespace:

Property name: availablebandwidth

Purpose: To specify the available bandwidth on a link between two directly connected neighboring peers. For a link, available bandwidth is defined to be residual bandwidth minus the measured bandwidth used for the actual forwarding of non-RSVP-TE packets. For a bundled link, available bandwidth is defined to be the sum of the component link available bandwidths.

Value type: A single number value containing an integer component that may be prefixed with an optional minus sign, which may be followed by a fraction part and/or an exponent part.

Cardinality: \*1

Member parameters: N/A

Description: This is intended to be a constraint attribute value used together with 'routing cost' cost metric attribute. 'routingcost' may also be used with other cost constraint attributes that is used to specify cost constraints. If 'availablebw' is present, 'routingcost' MUST have at most one 'availablebw'.

Format definition:

```
AVAILABLEBANDWIDTH-param =  
"VALUE"("gt"/"lt"/"eq"/"ge"/"le") ("number" / "object")  
AVAILABLEBANDWIDTH-value = number / object  
; Value type and VALUE parameter MUST match.
```

Examples:

```
"data": {  
  "cost type": {  
    "cost-mode": "numerical",  
    "cost-metric": "routingcost",  
    "constraints" : {"residbw", "availbw"}  
    "endpoints": {  
      "srcs": [ "ipv4:192.0.2.2" ],  
      "dsts": [  
        "ipv4:192.0.2.89",  
        "ipv4:198.51.100.34",  
        "ipv4:203.0.113.45"  
      ]  
    }  
  }  
  "map": {  
    "ipv4:192.0.2.2": {  
      "ipv4:192.0.2.89": 0[residbw eq0,availbw eq0],  
      "ipv4:198.51.100.34": 0[residbw eq12.5,availbw eq10.5],  
      "ipv4:203.0.113.45": 0[residbw eq5.9,availbw eq3.9],  
    }  
  }  
}
```

#### 4. Cost Metric Extensions: Parameters

The following sections define Parameters used within Properties definitions.

##### 4.1. parameter: linkstate

Namespace:

Parameter name: linkstate

Purpose: Used in a multi-valued property to indicate whether it is steady state link performance.

Description: When a property is multi-valued, LINKSTATE can be used to construct a steady state performancetopology for initial tunnel path computation, or to verify alternative failover paths. The LINKSTATE is set when the measured value of this parameter exceeds its configured maximum threshold. The LINKSTATE is cleared when the measured value falls below its configured threshold. LINKSTATE should be used together with properties we defined in the section 3.

Format definition:

LINKSTATE-param = "LINKSTATE=" INDEX-value

LINKSTATE-value = integer

Examples:

```
object {
    JSONBOOL linkstate;
} linkdelay;

"data": {
    "cost type": {
        "cost-mode": "numerical",
        "cost-metric": "routingcost"
    }
    "constraints": { "linkdelay"
        endpoints: {
            "srcs": [ "ipv4:192.0.2.2" ],
            "dsts": [
                "ipv4:192.0.2.89"
            ]
        }
    }
}

"map": {
    "ipv4:192.0.2.2": {
        "ipv4:192.0.2.89": 0.0[linkdelay[linkstate eq 0] eq 10],
    }
}
```

## 5. Security Considerations

The properties defined in this document present no security considerations beyond those in Section 14 of the base ALTO specification [draft-ietf-alto-protocol].

## 6. IANA Considerations

IANA has added the following entries to the ALTO cost map Properties registry, defined in Section 3 of [RFCXXX].

| Namespace | Property   | Reference              |
|-----------|------------|------------------------|
|           | linkdelay  | [RFCxxxx], Section 3.1 |
|           | linkjitter | [RFCxxxx], Section 3.2 |
|           | linkloss   | [RFCxxxx], Section 3.3 |
|           | residbw    | [RFCxxxx], Section 3.4 |
|           | availbw    | [RFCxxxx], Section 3.5 |

IANA has added the following entries to the "ALTO cost map Parameters" registry, defined in [RFCxxxx] Section 4.1.

| Name-space | Parameter | Reference              |
|------------|-----------|------------------------|
|            | LINKSTATE | [RFCxxxx], Section 4.1 |

## 7. Normative References

- [ALTO] Alimi, R., "ALTO Protocol",  
ID draft-ietf-alto-protocol-16, May 2013.
- [OSPF] Giacalone, S., "OSPF Traffic Engineering (TE) Metric  
Extensions", ID draft-ietf-ospf-te-metric-extensions-04,  
June 2013.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate  
Requirement Levels", March 1997.
- [RFC4627] Crockford, D., "The application/json Media Type for  
JavaScript Object Notation (JSON)", RFC 4627, July 2006.
- [RFC5234] Crocker, D., "Augmented BNF for Syntax Specifications:  
ABNF", RFC 5234, January 2008.

Authors' Addresses

Qin Wu  
Huawei  
101 Software Avenue, Yuhua District  
Nanjing, Jiangsu 210012  
China

Email: [sunseawq@huawei.com](mailto:sunseawq@huawei.com)

Liang Xia  
Huawei  
101 Software Avenue, Yuhua District  
Nanjing, Jiangsu 210012  
China

Email: [frank.xialiang@huawei.com](mailto:frank.xialiang@huawei.com)





ALTO WG  
Internet-Draft  
Intended status: Standards Track  
Expires: January 16, 2014

Y. Yang, Ed.  
Yale University  
July 15, 2013

ALTO Topology Considerations  
draft-yang-alto-topology-00.txt

Abstract

The Application-Layer Traffic Optimization (ALTO) Service has defined Network and Cost maps to provide basic network information. In this document, we discuss some initial thinking on adding topology in ALTO.

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 16, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

|                                                                 |   |
|-----------------------------------------------------------------|---|
| 1. Introduction . . . . .                                       | 3 |
| 2. Motivation using Examples . . . . .                          | 4 |
| 2.1. Single-Switch . . . . .                                    | 4 |
| 2.2. Multiple Switches . . . . .                                | 4 |
| 2.3. Network Constraints/Policies of a Fixed E2E Path . . . . . | 4 |
| 2.4. Multi-Layer Topology . . . . .                             | 5 |
| 2.5. Multicast and Broadcast Topology . . . . .                 | 5 |
| 3. Sketch of Schema . . . . .                                   | 5 |
| 4. Graph Transformations to Build Topology/Overlays . . . . .   | 7 |
| 5. Operations on Exported Topology . . . . .                    | 8 |
| 6. Security Considerations . . . . .                            | 8 |
| 7. IANA Considerations . . . . .                                | 8 |
| 8. Acknowledgments . . . . .                                    | 8 |
| 9. References . . . . .                                         | 8 |
| 9.1. Normative References . . . . .                             | 8 |
| 9.2. Informative References . . . . .                           | 8 |
| Author's Address . . . . .                                      | 9 |

## 1. Introduction

Topology is a basic information component that a network can provide to network management tools and applications. Example tools and applications that can utilize network topology include traffic engineering, network services (e.g., VPN) provisioning, PCE, application overlays, among others [RFC5693, I-D.amante-i2rs-topology-use-cases, I-D.lee-alto-app-net-info-exchange].

A basic challenge in exposing network topology is that there can be multiple representations of the topology of the same network infrastructure, and each representation may be better suited for its own set of deployment scenarios. For example, the current base ALTO protocol [I-D.ietf-alto-protocol] is designed for a setting of exposing network topology using the extreme "my-Internet-view" representation, which does not report any internal network switches, and hence is a "single-switch" abstraction. We interpret the word "switch" in the generic sense of network equipment in this document, not limited to L2 devices. An issue of this abstraction is that there are applications who may need details about network elements (e.g., specific network switches and links), but these are not exposed in the single-switch topology abstraction. An opposite of the single-switch representation is the complete raw topology, spanning across multiple layers, to include all details of network states such as endhosts attachment, physical links, physical switch equipment, and logical structures (e.g., LSPs) already built on top of physical infrastructure devices. A problem of the raw topology representation, however, is that its exposure may violate privacy constraints. Also, a large raw topology may be overwhelming and unnecessary for specific applications.

In this document, we discuss an extension of ALTO for topology exposure. We focus on a particular network. We assume a raw network topology, i.e., the ground truth. How the raw topology information is collected is outside the scope of this document.

The organization of this document is not a typical normative document. In particular, we first introduce concepts through examples, to better motivate the design. Then we introduce a sketch of schema for exposing topology in ALTO. There are details of the schema that are not specified and the intention is to integrate with other designs such as [I-D.lee-alto-app-net-info-exchange]. Next we give a framework of topology transformations to help with the understanding of deriving multiple representations of the topology of the same network infrastructure. We finish by pointing out operations based on new ALTO topology exposure.

## 2. Motivation using Examples

We distinguish between endhosts and the network infrastructure of the network. Endhosts are sources and destinations of data that the network infrastructure carries. The network itself is neither the source or the destination of data.

For the given network, it provides "access ports" or access points where digital signal from endhosts enter and leave the network. One should understand "access ports" in a general sense. For example, an access port can be a physical Ethernet port connecting to a specific endhost, or it can be a port connecting to a CE which connects to a large number of endhosts. Let AP be the set of access ports that the network provides.

### 2.1. Single-Switch

A high-level abstraction of a network topology is only the set AP, and one can visualize the network as a single switch. At each ap in AP, a set of endhosts can be reached as destinations. Let  $\text{dest}(\text{ap})$  denote the set of endhosts reachable at ap. The base ALTO protocol introduces PID to represent a partition of the set AP. Each subset in the partition is named as a PID, and the complete partition is conveyed as the Network Map. The ALTO base protocol then conveys the pair-wise connection properties from one PID to another PID through the "single-switch". This is the Cost Map.

### 2.2. Multiple Switches

Now, assume that the network actually consists of multiple switches, and the application needs to know more detailed topology. To help with the understanding, we consider the example case that the network has three switches, s1, s2 and s3. Each switch is connected to the other. The set AP is naturally divided as AP1, AP2, and AP3, denoting the access ports connected to the three switches respectively. The topology then exposed is simple to represent: there are three components: PIDs: {AP1, AP2, AP3}, Switches: {s1, s2, s3}, and Links: {s1->s2, s2->s1, ..., s2->s3, s3->s2}. It is straightforward to extend ALTO to represent the two additional components: Switches and Links.

### 2.3. Network Constraints/Policies of a Fixed E2E Path

Although the preceding 3-component representation is suited for some settings, e.g., traffic engineering who works on the raw topology, some other applications may need to or should only know a topology that encodes existing network constraints or policies. Note that such constraints may also come from another network tool or

application, to allow modular management composition.

For example, there can be a constraint, policy, or modular composition of the result of another application that endhosts from ap1 in AP1 connected to s1 must use the path s1 -> s2 -> s3 to reach endhosts at ap3 in AP3. To encode such a constraint to an application, there can be two choices: (1) create virtual switches and links still use the uniform graph-based representation; or (2) enumerate such a constraint in an end-to-end overlay representation.

#### 2.4. Multi-Layer Topology

Now assume that the link s1 -> s2 is actually a given optical path, and s1 -> s3 is another given optical path, and the deployment scenario requires that this detail be exposed to the tool or application on top of topology exposure, for example, to evaluate reliability considering shared risk link groups. To handle such a case, one can encode the optical topology in a graph representation, and also include (layer 3) end-to-end entries s1 -> s2 and s1 -> s3 to specify the paths or some transformation of the paths such as encoded, opaque shared-risk-link group numbers for each of the s1 -> s2 and s1 -> s3 paths.

#### 2.5. Multicast and Broadcast Topology

Next consider more complexity. Assume that the link from s1 -> s2 is actually a wireless link and the application may benefit in knowing that s1 -> s2 and s1 -> s3 can be active simultaneously. In other words, s1 -> [s2, s3] is a broadcast link. Knowing such links can be beneficial in settings such as wireless opportunistic routing.

### 3. Sketch of Schema

Given the preceding, we consider the following schema, which consists of EndhostMap, Topology, and Overlays.

EndhostMap: which encodes PIDs representing endhosts.

```

object {
    VersionTag      map-vtag;
    EndhostMapData  map;           // CHANGE: rename NetworkMap
                                   // to EndhostMap??

} InfoResourceEndhostMap;

object-map {
    PIDName -> EndpointAddrGroup; // already defined in base ALTO
} EndhostMapData;

```

Topology: A network can define 0 to multiple topology maps, where each topology consists of switches and links:

```

object {
    VersionTag      map-vtag;
    SwitchMapData   switches;
    LinkMapData     links;

} InfoResourceTopology;

object-map {
    JSONString -> SwitchProperties; // switch name to properties
} SwitchMapData;

object {
    AccessLinks     alinks;       // between a PID to a switch
    TransportLinks  tlinks;       // between two switches
} LinkMapData;

```

(Overlay) paths: A network can define 0 to multiple overlays on top of a given topology, and path can be recursive:

```

object {
    PathType        type;         // E2ECostMap; LSPs; ...
    [PathMapData    map;]         // depends on type,
                                   // if it is E2ECostMap,
                                   // it is InfoResourceCostMap
                                   // defined in [alto-protocol]
} PathMap;

```

#### 4. Graph Transformations to Build Topology/Overlays

The preceding sections give a top-down derivation. In this section, we give a graph transformation framework to build the schema from a raw topology  $G(0)$ . The network conducts transformations on  $G(0)$  to obtain other topologies, with the following objectives:

1. Simplification:  $G(0)$  may have too many details that are unnecessary for the receiving app (assume intradomain, and hence no security problem); and
2. Preservation of privacy: there are details that the receiving app should not be allowed to see; and
3. Convey of logical structure (e.g., MPLS paths already computed); and
4. Convey of capability constraints (the network can have limitations, e.g., it uses only shortest path routing); and
5. Allow modular composition: path from one point to another point is delegated to another app.

The transformation of  $G(0)$  is to achieve/encode the preceding. For conceptual clarity, we assume that the network uses a given set of operators. Hence, given a sequence of operations and starting from  $G(0)$ , the network builds  $G(1)$ , to  $G(2)$ , ...

Below is a list of basic operators that the network may use to transform from  $G(n-1)$  to  $G(n)$ :

- o O1: Deletion of a switch/port/link from  $G(n-1)$ ;
- o O2: Switch aggregation: a set  $V_s$  of switches are merged as one new (logical) switch, links/ports connected to switches in  $V_s$  are now connected to the new logical switch, and then all switches in  $V_s$  are deleted;
- o O3: Path representation: For a given extra path from A to R1 to R2 ... to B in  $G(n-1)$ , a new (logical) link A  $\rightarrow$  B is added; if the constraint is that A  $\rightarrow$  must use the path, it will be put into the Overlay;
- o O4: Switch split: A switch  $s$  in  $G(n-1)$  becomes two (logical) switches  $s_1$  and  $s_2$ . The links connected to  $s_1$  is a subset of the original links connected to  $s$ ; so is  $s_2$ .



## 5. Operations on Exported Topology

Going beyond the basic topology exposure from the network and applications/tools, we anticipate that applications and tools can derive results and feed to topology. In particular, we consider the following operations:

- o Instantiation of app guidance in real network: The details of instantiation will be outside the scope of this document. Example protocols include PCEP Extensions for Stateful PCE [I-D.ietf-pce-stateful-pce], RSVP LSP's and their associated characteristics, (i.e.: head and tail-end LSR's, bandwidth, priority, preemption, etc.). The reason that we choose the preceding operator set is that they are "implementable".
- o We also anticipate topology guided mapping of other data: to allow applications to subscribe to statistics and link status from the derived topology.

## 6. Security Considerations

This document has not conducted its security analysis.

## 7. IANA Considerations

This document does not specified its IANA considerations, yet.

## 8. Acknowledgments

The author thanks discussions with Erran Li, Tianyuan Liu, Andreas Voellmy, Haibin Song, and Yan Luo.

## 9. References

### 9.1. Normative References

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

### 9.2. Informative References

- [I-D.amante-i2rs-topology-use-cases]  
Amante, S., Medved, J., Previdi, S., and T. Nadeau,  
"Topology API Use Cases",

draft-amante-i2rs-topology-use-cases-00 (work in progress), February 2013.

[I-D.ietf-alto-protocol]

Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-17 (work in progress), July 2013.

[I-D.lee-alto-app-net-info-exchange]

Lee, Y., Bernstein, G., Choi, T., and D. Dhody, "ALTO Extensions to Support Application and Network Resource Information Exchange for High Bandwidth Applications", draft-lee-alto-app-net-info-exchange-02 (work in progress), July 2013.

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

#### Author's Address

Y. Richard Yang (editor)  
Yale University  
51 Prospect St  
New Haven CT  
USA

Email: yry@cs.yale.edu

