

ALTO WG
Internet-Draft
Intended status: Informational
Expires: April 24, 2014

G. Bernstein, Ed.
Grotto Networking
Y. Yang, Ed.
Yale University
Y. Lee, Ed.
Huawei Technologies
October 21, 2013

ALTO Topology Service: Uses Cases, Requirements, and Framework
draft-bernstein-alto-topo-00

Abstract

Exposing additional topology information of networks to applications and users beyond that of the current ALTO protocol can enable many important existing and emerging use cases, and many network providers already provide additional information about their networks. At the same time, there is no standard for exposing network topology in a manner that provides simplification via abstraction to the application layer and information hiding via abstraction to the network provider. In this document, we provide a survey of use-cases for extended network topology information, present some initial requirements for such services, and then give a framework of how to integrate such an extended ALTO topology service with network control infrastructure.

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 April 24, 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
1.1. Requirements Language	3
2. Uses Cases	3
2.1. Technology Specific Examples	4
3. Requirements	5
4. ALTO Topology Framework	5
4.1. Abstract Topology Representation	5
4.2. Sources of Raw Topology Information	5
4.3. Service/Client Specific Topology Abstraction	5
4.4. Reservation System Compatibility	5
5. Acknowledgements	6
6. IANA Considerations	6
7. Security Considerations	6
8. References	6
8.1. Normative References	6
8.2. Informative References	6
Authors' Addresses	7

1. Introduction

Topology is a basic property of a network. Hence there is a spectrum of use cases where an application (or user) can benefit from obtaining some knowledge of the topology of the network that it uses or considers using, beyond the "single-switch" abstraction topology abstraction presented in the ALTO Base Protocol [I-D.ietf-alto-protocol] as discussed in [I-D.yang-alto-topology].

As a simple case, many networks already provide public views to their topologies so that current or potential users of their networks can learn more about their networks; for example, see Verizon [1]; Comcast [2]; CenturyLink [3]; BT [4]; China Telecom [5]; Internet 2 [6]. A user (application) with such information may conduct a wide variety of analysis, for example, in determining its service provider(s).

For more advanced use cases such as in a programmatic setting, a topology manager of a network may expose a topology of the network to an application so that the application can provide its input regarding the operations of the network. A concrete example setting is the recent development of Software Defined Networking (SDN); for example see OpenDayLight [7]; Maple [8].

The objective of this document is three folds: (1) it surveys general uses cases and existing designs of how network topologies are exposed to applications; (2) it presents the requirements in exposing network topologies; and (3) it gives a framework of how network topologies to applications can be integrated into network control.

1.1. 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].

2. Uses Cases

Uses cases generally relate to some type of cost metric optimization, application policy, resource requirements (bandwidth), and/or performance criteria such as delay. In the following we give a non-exhaustive list of uses cases for a extended ALTO topology service.

Large Bandwidth

Applications that make extensive use of network bandwidth resources are discussed in [I-D.bernstein-alto-large-bandwidth-cases]. In addition to a general discussion of large bandwidth requirements specific examples of video on demand and inter-data center networking are given. An optimization example for a scheduled backup service can be found at <http://www.grotto-networking.com/BackupExample.html> [9].

Enhanced Reliability

GMPLS [RFC3945] and GMPLS routing [RFC4202] in particular have included enhanced reliability support in the form of shared risk link group (SRLG) information that lets a path computation entity understand which links are at risk of simultaneous failures (fate sharing). In addition in optical networks link and node diverse paths are a common method to enhance reliability [OptControl].

However in many cases only the application may have a full view of its reliability needs. For example consider a high reliability application making use of multiple data centers

for redundancy and increased reliability, such reliability would be significantly diminished if the paths to those data centers shared similar fates.

Latency Sensitivity

From high performance gaming to high frequency trading latency can critically impact application performance. However, reductions in latency may need to be factored against other costs or resource requirements. As mentioned in <http://cacm.acm.org/magazines/2013/10/168186-barbarians-at-the-gateways/abstract> [10] some high frequency trading applications need to make use of both a low latency path and a high bandwidth path.

Policy Enforcement

Many application specific requirements such as the HIPPA privacy rule, can place restrictions on where a certain customers data may be kept, or what geographic regions a customers data can traverse, etc... Enhancing topology information made available to an application can help it ensure such requirements are satisfied.

2.1. Technology Specific Examples

Here we furnish a partial list of examples that illustrate one or more properties desirable in an extended ALTO topology service.

SDN: Project Floodlight

Project floodlight provides limited inter switch topology information <https://github.com/wallnerrryan/floodlight/blob/master/example/graphTopo.py> [11].

SDN: Open Daylight

The Open Daylight project is aiming to supply a "north bound" topology service https://jenkins.opendaylight.org/controller/job/controller-merge/ws/opendaylight/northbound/topology/target/site/wsdocs/el_ns0_topology.html [12].

Grid Computing - OGF NML

The Open Grid Forum has developed a general Network Markup Language <http://www.ogf.org/documents/GFD.206.pdf> [13]. This borrows concepts from GMPLS and ITU-T G.805 models. However, it is not aimed at application layer users, but rather grid computing operators.

Fiber Maps (multiple carriers)

TBD.

HPC - cluster placement problem
TBD.

3. Requirements

Formal requirements to come...

4. ALTO Topology Framework

The framework portion of this document, like most IETF frameworks, is an informational section that shows how various systems could come together to form an extended ALTO topology service.

4.1. Abstract Topology Representation

References [I-D.lee-alto-app-net-info-exchange] and [I-D.yang-alto-topology] provide tentative models and encodings for abstract topology representation.

4.2. Sources of Raw Topology Information

From management systems, to proprietary interfaces to routing systems, to i2rs...

4.3. Service/Client Specific Topology Abstraction

Although only the topology/resource abstraction format would be subject to standardization, this section will illustrate some techniques that can be efficiently used to derived service and client specific topology abstractions. References [I-D.lee-alto-app-net-info-exchange] and [I-D.yang-alto-topology] give examples of how raw network topology information can be processed into abstracted application specific form. A lengthier paper with more examples and technology considerations can be found at [14].

4.4. Reservation System Compatibility

As mentioned in the requirements ALTO topology extensions must be able to work with technologies that require resource reservations as well as those that don't. In implementing an overall system the information supplied by an extended ALTO topology service will need to be compatible with a "reservation system" if there is one.

At the IETF we have seen similar requirements for compatibility between GMPLS routing and signaling systems, particularly via the concept of loose routes.

5. Acknowledgements

Hopefully we'll have lots of interested folks commenting and we'll give them credit here.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

All drafts are required to have a security considerations section and this will as we flesh it out.

8. References

8.1. Normative References

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

[min_ref] authSurName, authInitials., "Minimal Reference", 2006.

8.2. Informative References

[I-D.bernstein-alto-large-bandwidth-cases]
Bernstein, G. and Y. Lee, "Use Cases for High Bandwidth Query and Control of Core Networks", draft-bernstein-alto-large-bandwidth-cases-00 (work in progress), June 2011.

[I-D.ietf-alto-protocol]
Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-20 (work in progress), October 2013.

[I-D.lee-alto-app-net-info-exchange]
Lee, Y., Dhody, D., Wu, Q., Bernstein, G., and T. Choi, "ALTO Extensions to Support Application and Network Resource Information Exchange for High Bandwidth Applications ", draft-lee-alto-app-net-info-exchange-03 (work in progress), October 2013.

[I-D.yang-alto-topology]
Yang, Y., "ALTO Topology Considerations", draft-yang-alto-topology-00 (work in progress), July 2013.

[OptControl]
Bernstein, G., Rajagopalan, B., and D. Saha, "Optical Network Control", 2004.

[RFC3945] Mannie, E., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", RFC 3945, October 2004.

[RFC4202] Kompella, K. and Y. Rekhter, "Routing Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4202, October 2005.

Authors' Addresses

Greg M. Bernstein (editor)
Grotto Networking
Fremont, CA
US

Phone: +01 510 623 8575
Email: gregb@grotto-networking.com

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

Email: yry@cs.yale.edu

Young Lee (editor)
Huawei Technologies
1700 Alma Drive, Suite 500
Plano, TX 75075
USA

Phone: (927) 509-5599
Email: ylee@huawei.com

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: April 22, 2014

Q. Fu, Ed.
Z. Cao
China Mobile
H. Song
Huawei
October 19, 2013

What's the Impact of Virtualization to ALTO?
draft-fu-alto-nfv-usecase-00

Abstract

This draft presents a use case of Application Layer Traffic Optimization (ALTO) with the emergence of Network Function Virtualization (NFV). 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 emerging Network Functions Virtualisation (NFV), as currently being in progress in ETSI NFV, leverages standard IT virtualisation technology to consolidate many network equipment types onto industry standard high volume servers, switches and storage.

Status of This Memo

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

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

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

This Internet-Draft will expire on April 22, 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. Impact of Virtualized Endpoints	4
4. ALTO usecase with NFV	6
5. Interaction Architecture of ALTO and NFV	6
6. Informative References	7
Authors' Addresses	7

1. Introduction

This draft present a use case of Application Layer Traffic Optimization (ALTO) with the emergence of Network Function Virtualization (NFV). 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. Typical deployment scenarios of ALTO include P2P and CDN, in which P2P tracker or CDN request router queries ALTO server for network map and cost map, in order to make decisions on which peer to select for content sharing.

The emerging Network Functions Virtualisation (NFV), as currently being in progress in ETSI NFV, leverages standard IT virtualisation technology to consolidate many network equipment types onto industry standard high volume servers, switches and storage. The NFV architecture in ETSI ongoing work includes an NFV Management and Orchestrator (M&O), the VNF(Virtualized Network Function) and the VNFI(Virtualized Network Function Infrastructure), as is shown in Figure 1. The NFV M&O is responsible for creating and managing the VNFs on the VNFI. Interactions between NFV M&O, VNF and VNFI are beyond scope of this draft.

With the trend of various network functions being virtualized, there will be impacts on cost and network characteristics of the service endpoints. Under the ALTO architecture, we analyze the problems and the necessity of extending the ALTO protocols to faithfully reveal

the network to the clients. The central problem this draft would like to investigate is: what's the impact of virtualization to ALTO.

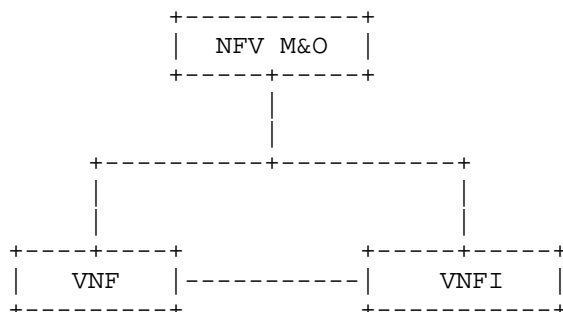


Figure 1: NFV Architecture in Brief

This draft analyzes the impacts of virtualized endpoints to application layer traffic optimization and presents a usecase of ALTO in CDN and P2P network with the peers as a VNF(Virtualized Network Function). .

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].

We use the following terms defined in [RFC5693]. Application, Peer, ALTO service, ALTO server, ALTO client, ALTO query, ALTO Reply.

And the following terms used in this document have their definitions from the NFV end to end architecture [NFVE2E].

NFV: network function virtualization. NFV technology uses the commodity servers to replace the dedicated hardware boxes for the network functions, for example, home gateway, enterprise access router, carrier grade NAT and etc. So as to improve the re-usability, allow more vendors into the market, and reduce time to market. NFV architecture includes a NFV controller (orchestrator) to manage the virtual network functions and the infrastructure resources.

NF: A functional building block within an operator's network infrastructure, which has well-defined external interfaces and a well-defined functional behavior. Note that the totality of all network functions constitutes the entire network and services infrastructure of an operator/service provider. In practical terms,

a Network Function is today often a network node or physical appliance.

VNF: virtual network function, an implementation of an executable software program that constitutes the whole or a part of an NF that can be deployed on a virtualization infrastructure.

VM: virtual machines, a program and configuration of part of a host computer server. Note that the Virtual Machine inherits the properties of its host computer server e.g. location, network interfaces.

SLA: Service Layer Agreement.

3. Impact of Virtualized Endpoints

This section analyzes the impact of virtualization when application or service endpoints are deployed on virtualized infrastructure.

It is generally believed that generic computing equipment is difficult to accomplish the same capability of specialized and dedicated equipment. Operator network normally consists of many dedicated equipment, and the services running on them are not virtualized. NFV initiatives investigate the use cases, architecture and requirements of moving network functions to the virtualized infrastructure.

We analyze the impacts of virtualized endpoints to application layer traffic optimization for the following aspects.

1. Performance. The NFV framework is claimed to be able to instantiate and configure any given VNF over the underlying infrastructure so that the resulting VNF instance performance is conforming to the expressed requirement. Using appropriate VNF configuration schemes [I-D.song-opsawg-virtual-network-function-config], the operator or service provider can express their performance requirement. From this point, it is the same as physical and non-virtualized service endpoints. The difference is that the service assurance of virtualized endpoints is more difficult to ensure.

2. Portability. Different from physical equipment, NFV framework is able to provide the capability to load, execute and move VNFs across different but standard multi-vendor environments, and have to support an interface to decouple VNF associated software instances from the underlying infrastructure. Portability has impacts on the mobility and network location of the service points, which in the turn will impact the service point selection process and service continuity.
3. Elasticity. The NFV framework is able to allow VNFs to be scaled with SLA requirements, on-demand scaling or automatically scaling. With the elasticity capability, VNF endpoints capability with respect to computing and networking are dynamic. The ALTO discovery and selection process will be impacted to reflect such dynamic information.
4. Resilience. NFV framework provides the necessary mechanisms to allow VNF to be recreated after a failure. In addition to OAM in traditional non-virtualized environment, the NFV M&O will manage the metrics such as packet loss rate, latency, delay variation of flows, maximum time to detect and recover from faults. All of these information will be valuable to ALTO client.
5. Energy efficient. Studies have indicated that NFV could potentially deliver up to 50% energy saving compared with traditional appliance based network infrastructure. In service point selection, this could be a criteria when the service provider is interested in saving power.
6. Service assurance. Dedicated carrier-grade devices normally have requirements like 99.999%, but the such high availability is still challenging for VNFs. The ALTO server should be aware of the assurance level of these virtualized endpoints.
7. Network infrastructure maintenance. The VNFs may be bridged and linked using the virtualized switches on the computing node. The network layer performance and availability metrics are only possible to collect when the OAM have established the tunnels to the these virtual network infrastructure. For example, normal PING can only reflect the physical computing node availability, but cannot reflect the VMs bridged using virtual switches and hidden with tunnel encapsulations.

4. ALTO usecase with NFV

The emergence of NFV means that some legacy devices which used to work on a physical server, now can be moved to a VM and work as a VNF. Under such circumstance, the NFV M&O can act as a Dynamic Network Info provider for ALTO.

The following paragraph will present a usecase of ALTO in CDN with NFV. In the CDN network, the user agent first makes initial request to the Request Router. The Request Router will first query the ALTO server for network and cost map to select an appropriate surrogate. The Request Router then responds to the UA with a redirection to the selected surrogate. The UA then connects directly to the suggested surrogate to obtain the content.

When a certain surrogate changes to a VNF and is managed by a NFV M&O, The NFV M&O can dynamically update the network and cost info of the surrogate to the ALTO server. In the meantime, the NFV M&O should also keep ALTO server informed about the virtualized nature of the VNF surrogate, since its performance might be lower than physical devices. In the migration stage of NFV, in which VNF and physical devices coexist in the network, ALTO server may consider the virtualized nature of VNF as a rating criteria that should inform the clients. Clients may choose physical devices instead of VNF surrogates due to consideration of performance.

In the P2P scenario, similar situations can also happen when peers become VNFs. In this case, NFV M&O should also inform ALTO server about the virtualize nature of the VNF peers. And P2P trackers can take such nature into consideration when selecting peers to obtain content.

5. Interaction Architecture of ALTO and NFV

A vertical architecture is proposed in this draft for ALTO and NFV interaction, in which NFV M&O is in responsible of info update to the ALTO server, as is shown in Figure 2.

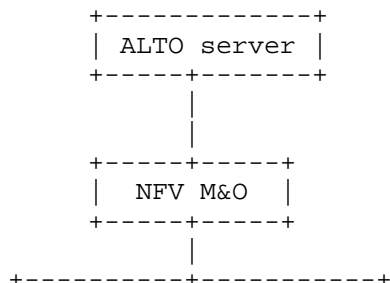




Figure 2 ALTO and NFV interaction architecture

In this architecture, NFV M&O can automatically update fine or coarse grained VNF info to the ALTO server timely. The virtualized nature of the VNFs should be informed to the ALTO server by NFV M&O as a rating criteria. In the meantime, details of VNF can be updated to the ALTO server by NFV M&O according to privacy privilege configured by the user.

6. Informative References

- [I-D.song-opsawg-virtual-network-function-config]
Song, H. and Z. Cao, "The Problems of Virtual Network Function Configuration", draft-song-opsawg-virtual-network-function-config-01 (work in progress), October 2013.
- [NFVE2E] , "Network Functions Virtualisation: End to End Architecture, <http://docbox.etsi.org/ISG/NFV/70-DRAFT/0010/NFV-0010v016.zip>", .
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, October 2009.

Authors' Addresses

Qiao Fu (editor)
China Mobile
China
China

Email: fuqiao1@outlook.com

Zhen Cao
China Mobile
Xuanwumenxi Ave. No.32
Beijing 100053
China

Email: zehn.cao@gmail.com, caozhen@chinamobile.com

Haibin Song
Huawei

Email: haibin.song@huawei.com

ALTO
Internet-Draft
Intended status: Informational
Expires: April 24, 2014

M. Stiemerling, Ed.
NEC Europe Ltd.
S. Kiesel, Ed.
University of Stuttgart
S. Previdi
Cisco
M. Scharf
Alcatel-Lucent Bell Labs
October 21, 2013

ALTO Deployment Considerations
draft-ietf-alto-deployments-08

Abstract

Many Internet applications are used to access resources such as pieces of information or server processes that are available in several equivalent replicas on different hosts. This includes, but is not limited to, peer-to-peer file sharing applications. The goal of Application-Layer Traffic Optimization (ALTO) is to provide guidance to applications that have to select one or several hosts from a set of candidates, which 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 April 24, 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. ALTO Entities	4
2.1.1. Baseline Scenario	4
2.1.2. Placement of ALTO Entities	4
2.2. Classification of Deployment Scenarios	6
2.2.1. Deployment Degrees of Freedom	6
2.2.2. Information Exposure Models	7
2.2.3. More Advanced Deployments	7
3. Deployment Considerations by ISPs	9
3.1. Objectives for the Guidance to Applications	9
3.1.1. General Objectives for Traffic Optimization	9
3.1.2. Inter-Network Traffic Localization	10
3.1.3. Intra-Network Traffic Localization	11
3.1.4. Network Off-Loading	13
3.1.5. Application Tuning	14
3.2. Provisioning of ALTO Maps	14
3.2.1. Data Sources	14
3.2.2. Privacy Requirements	14
3.2.3. Map Partitioning and Grouping	15
3.2.4. Rating Criteria and/or Cost Calculation	15
3.3. Known Limitations of ALTO	18
3.3.1. Limitations of Map-based Approaches	18
3.3.2. Limitations of Non-Map-based Approaches	20
3.4. Map Examples for Different Types of ISPs	20
3.4.1. Small ISP with Single Internet Uplink	20
3.4.2. ISP with Several Fixed Access Networks	22
3.4.3. ISP with Fixed and Mobile Network	24
3.5. Deployment Experiences	25
4. Using ALTO for P2P Traffic Optimization	25
4.1. Overview	26
4.1.1. Usage Scenario	26
4.1.2. Applicability of ALTO	29

4.2.	Deployment Recommendations	29
4.2.1.	ALTO Services	29
4.2.2.	Guidance Considerations	29
5.	Using ALTO for CDNs	33
5.1.	Overview	33
5.1.1.	Usage Scenario	33
5.1.2.	Applicability of ALTO	33
5.2.	Deployment Recommendations	34
5.2.1.	ALTO Services	34
5.2.2.	Guidance Considerations	35
6.	Other Use Cases	36
6.1.	Monitoring Data Reporting	36
6.2.	Virtual Private Networks (VPNs)	36
6.3.	In-Network Caching	36
7.	Security Considerations	37
7.1.	Information Leakage from the ALTO Server	37
7.2.	ALTO Server Access	38
7.3.	Faking ALTO Guidance	38
8.	Conclusion	39
9.	References	39
9.1.	Normative References	39
9.2.	Informative References	39
Appendix A.	Appendix: Monitoring ALTO	41
A.1.	Monitoring Metrics Definition	41
A.2.	Monitoring Data Sources	42
A.3.	Monitoring Structure	42
Appendix B.	Appendix: API between ALTO Client and Application	43
Appendix C.	Contributors List and Acknowledgments	43
Authors' Addresses		44

1. Introduction

Many Internet applications are used to access resources such as pieces of information or server processes that are available in several equivalent replicas on different hosts. This includes, but is not limited to, peer-to-peer (P2P) file sharing applications and Content Delivery Networks (CDNs). 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, which are able to provide a desired resource. The basic ideas and 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

2.1. ALTO Entities

2.1.1. Baseline Scenario

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.

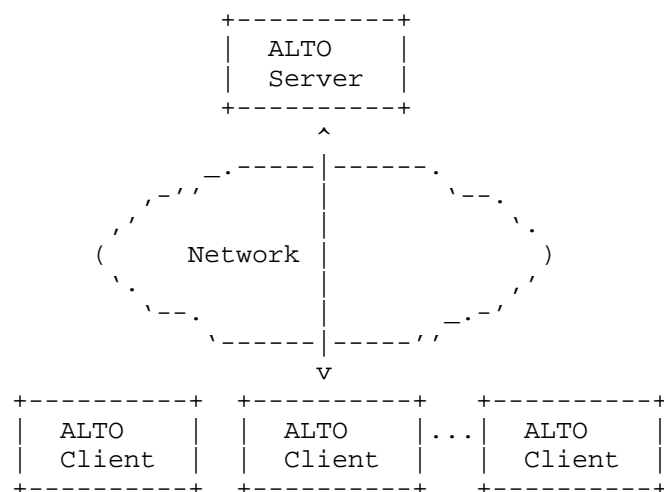
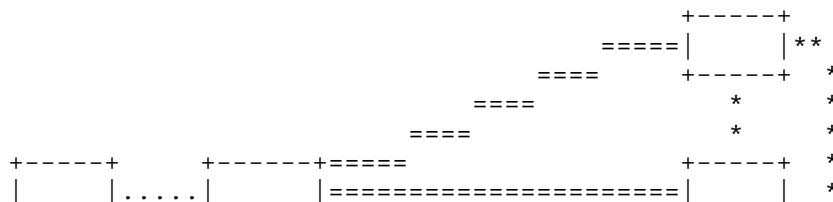


Figure 1: Baseline Deployment Scenario of the ALTO Protocol

2.1.2. 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.



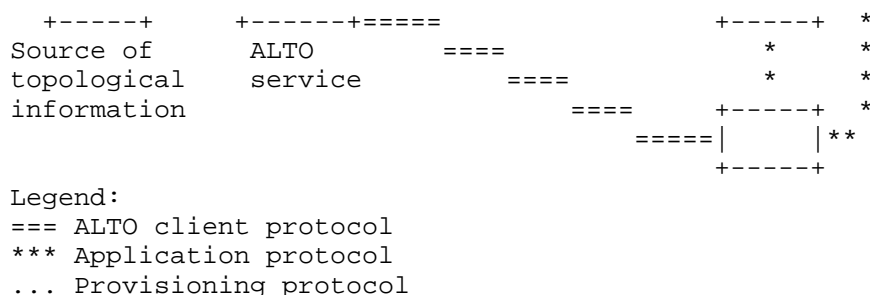


Figure 2: Overview of protocol interaction between ALTO elements without a resource directory

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 edonkey.

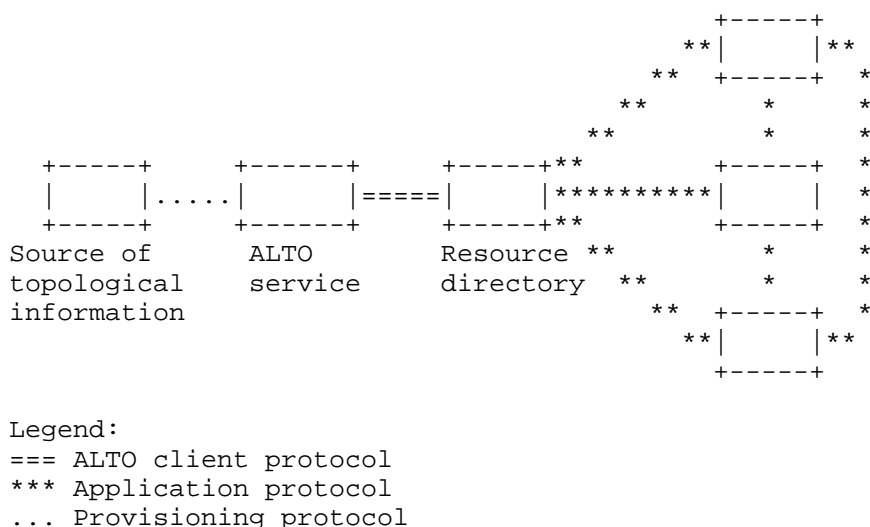


Figure 3: Overview of protocol interaction between ALTO elements with a resource directory

In Figure 3, a use case with a resource directory is illustrated, e.g., a tracker in peer-to-peer filesharing. Both deployment scenarios may differ in the number of ALTO clients that access an ALTO service: If ALTO clients are implemented in a resource directory, ALTO servers may be accessed by a limited and less dynamic set of clients, whereas in the general case any host could be an ALTO client. This use case is further detailed in Section 4.

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, as discussed in Section 5.

2.2. Classification of Deployment Scenarios

2.2.1. Deployment Degrees of Freedom

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.

ALTO deployments can be differentiated e.g. according to the following aspects:

1. Applicable 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 affects 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. Size of 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 from the public Internet.
3. Covered 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 situations (see also [RFC6708]).

2.2.2. Information Exposure Models

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. 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 partially be 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 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, approach 1 requires 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. The ALTO protocol supports this scheme by the Network and Cost Map Service.

Approach 2 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, approach 2 requires that the clients send their operational information to the server. This approach is realized by the ALTO Endpoint Cost Service (ECS).

Both approaches have their pros and cons, as detailed in Section 3.3.

2.2.3. More Advanced Deployments

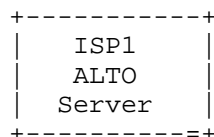
From an ALTO client's perspective, there are two fundamental ways to use ALTO:

1. Single server: An ALTO client only obtains guidance from a single ALTO server instance, e.g., an ALTO server that is offered by the network service provider of the corresponding access network. This ALTO server can be discovered e.g. by ALTO server discovery [I-D.ietf-alto-server-discovery].
2. Multiple servers: An ALTO client is aware of more than one ALTO server. This scenario is mostly identical to the former one if all those servers provide the same guidance (e.g., load balancing). Yet, an ALTO client can also decide to access multiple servers providing different guidance, possibly from different operators. In that case, it may be difficult for an ALTO client to compare the guidance from different servers. How to discover multiple servers is an open issue.

There are also different options regarding the guidance offered by an ALTO server:

1. Authorative servers: An ALTO server instance can provide guidance for all destinations for all kinds of ALTO clients.
2. Cascaded servers: An ALTO server may itself include an ALTO client and query other ALTO servers, e.g., for certain destinations. This results in a cascaded deployment of ALTO servers, as further explained below.
3. Inter-server synchronization: Different ALTO servers may communicate by other means. This approach is not further discussed in this document.

An assumption of the ALTO solution is that ISPs operate ALTO servers independently, irrespectively of other ISPs. This may be true for most envisioned deployments of ALTO but there are certain deployments that may have different settings. Figure 4 shows such a setting with a university network that is connected to two upstream providers. ISP2 is 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.



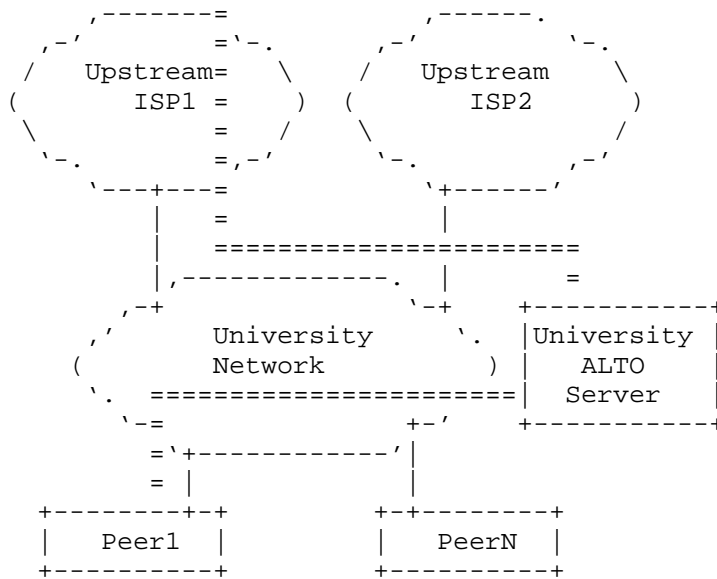


Figure 4: Cascaded ALTO Server

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 is an example for cascaded ALTO servers.

3. Deployment Considerations by ISPs

3.1. Objectives for the Guidance to Applications

3.1.1. General Objectives for Traffic Optimization

The Internet is a large network consisting of many networks worldwide. These networks are built by network operators or Internet Service Providers (named ISP in this memo), and these networks provide network connectivity to access networks, such as cable networks, xDSL networks, 3G/4G mobile networks, etc. Some of these networks are also built by universities or big organizations. These network providers 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.

The objective of ALTO is to give 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 means to let the ALTO server operator express its preference, whatever this preference is.

ALTO enables ISPs to perform traffic engineering by influencing application resource selections. This traffic engineering can have different objectives:

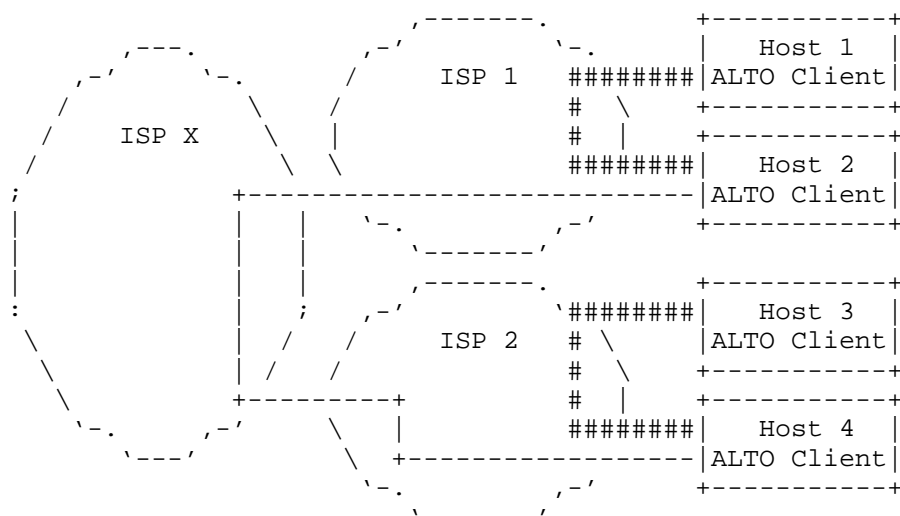
1. Inter-network traffic localization: ALTO can help to reduce inter-domain traffic. The networks of ISPs are connected through peering points. From a business view, the inter-network settlement is needed for exchanging traffic between these networks. These peering agreements can be costly. To reduce these costs, a simple objective is to decrease the traffic exchange across the peering points and thus keep the traffic in the own network or Autonomous System (AS) as far as possible.
2. Intra-network traffic localization: In case of large ISPs, the network may be grouped into several networks, domains, or Autonomous Systems (ASs). 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.
3. Network off-loading: Compared to fixed networks, mobile networks have some special characteristics, including smaller link bandwidth, high cost, limited radio frequency resource, and limited terminal battery. In mobile networks, the usage of wireless link should be decreased as far as possible and be used efficiently. For example, in the case of a P2P service, the hosts in fixed networks should avoid retrieving data from hosts in the mobile networks, and hosts in mobile networks should prefer the data retrieval from hosts in fixed networks.
4. Application tuning: ALTO is also a powerful tool to optimize the performance of applications that depend on the network and perform resource selection decisions.

In the following, these objectives are explained in more detail with deployment examples.

3.1.2. Inter-Network Traffic Localization

ALTO guidance can be used to keep traffic local in a network. An ALTO server can 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 its preference for 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 5 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 5: 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.

3.1.3. Intra-Network Traffic Localization

The above sections described the results of the ALTO guidance on an inter-network level. However, ALTO can also be used for intra-network localization. In this case, ALTO provides guidance which internal hosts are to be preferred inside a single network or, e.g., one AS. 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. If Host 1 and Host 2 exchange their data with remote hosts, they would probably congest the bottleneck link.

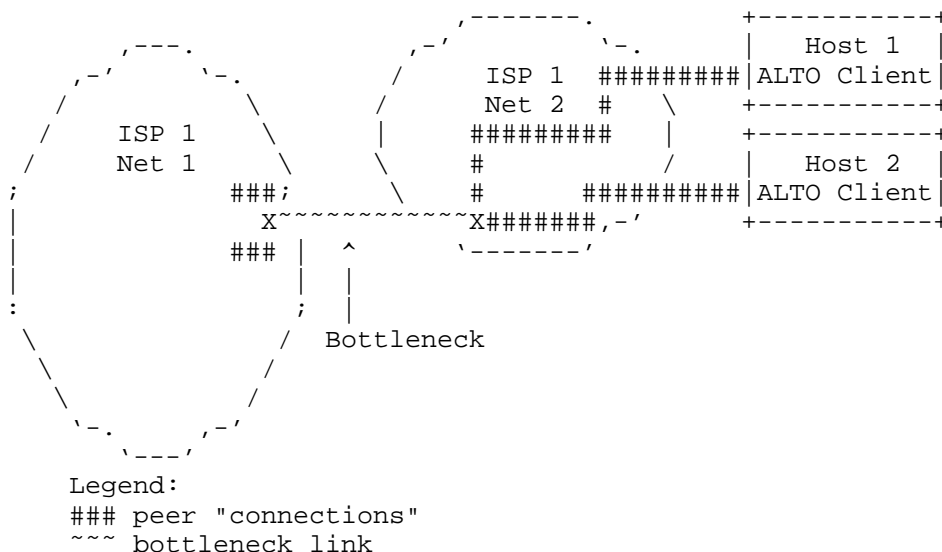
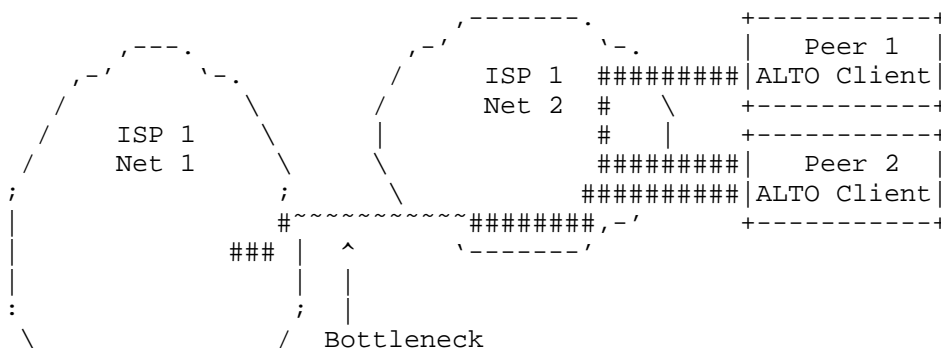


Figure 6: Without Intra-Network ALTO Traffic Localization

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.



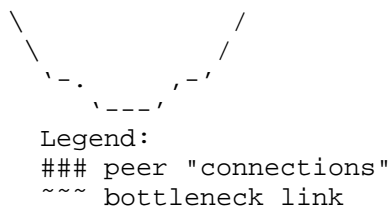


Figure 7: With Intra-Network ALTO Traffic Localization

3.1.4. Network Off-Loading

Another scenario is off-loading traffic from networks. This use of ALTO can be beneficial in particular 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.

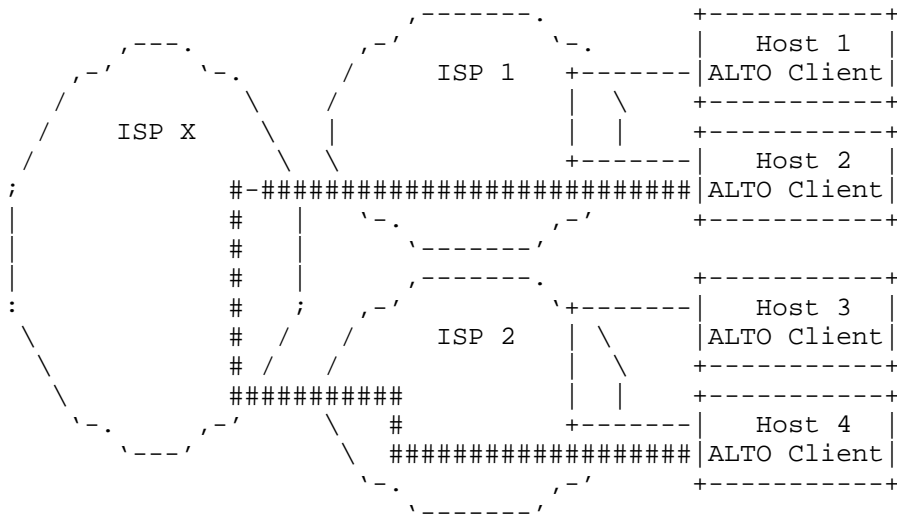


Figure 8: ALTO Traffic Network De-Localization

Figure 8 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 5.

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

3.1.5. Application Tuning

ALTO can also provide guidance to optimize the application-level topology of networked applications, e.g., by exposing network performance information. Applications can often run own measurements to determine network performance, e.g., by active delay measurements or bandwidth probing, but such measurements result in overhead and complexity. Accessing an ALTO server can be a simpler alternative. In addition, an ALTO server may also expose network information that applications cannot easily measure or reverse-engineer.

3.2. Provisioning of ALTO Maps

3.2.1. Data Sources

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.

The ALTO server builds an ALTO-specific network topology that represents the network as it should be understood and utilized by the application. 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 aspects of the network are visible and required by the given use case and/or application.

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. The network topology information is controlled and managed by the ALTO server. 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.

3.2.2. Privacy 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.

With the ALTO Endpoint Cost Service, the ALTO client does not have to implement any specific algorithm or mechanism in order to retrieve, maintain and process network topology information (of any kind). The complexity of the network topology (computation, maintenance and distribution) is kept in the ALTO server and ECS is delivered on demand. This allows the ALTO server to enhance and modify the way the topology information sources are used and combined. This simplifies the enforcement of privacy policies of the ISP.

The ALTO Network Map and Cost Map service expose an abstracted view on the ISP network topology. Therefore, in this case care is needed when constructing those maps, as further discussed in Section 3.2.3.

3.2.3. Map Partitioning and Grouping

Host group descriptors are used in the ALTO client protocol to describe the location of a host in the network topology. These identifiers are called Partition ID (PID) and e.g. expand to a set of IP address ranges (CIDR).

An automated ALTO implementation may use dynamic algorithms to aggregate network topology. However, it is often desirable to have a mechanism through which the network operator can control the level and details of network aggregation based on a set of requirements and constraints.

IP/MPLS networks make use of a common mechanism to aggregate and group prefixes that is called BGP Communities. BGP is the protocol all ISP networks use in order to exchange information about their prefix reachability. BGP Community is an attribute used to tag a prefix to group prefixes based on mostly any criteria (as an example, most ISP 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 defined between groups. Alternatively, there are algorithms that allow a dynamic computation of cost between groups.

3.2.4. Rating Criteria and/or Cost Calculation

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 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.

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.

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.

Performance-related rating criteria:

- 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.

These rating criteria are subject to the remarks below:

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.

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.

3.3. Known Limitations of ALTO

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

3.3.1. Limitations of Map-based Approaches

The specification of the ALTO protocol [I-D.ietf-alto-protocol] uses 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 aggregate. A set of IP prefixes is called partition and the associated Host Group Descriptor is called Partition ID (PID). The "costs" between the various partition IDs is stored in a second map, the cost map. Map-based approaches lower the signaling load on the server as maps have to be retrieved only if they change.

One 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 IP subnets from time to time;
2. ISPs reallocate IP 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 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 reallocated 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.

3.3.2. 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 (ECS). 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.

3.4. Map Examples for Different Types of ISPs

3.4.1. Small ISP with Single Internet Uplink

For a small ISP, the inter-domain traffic optimizing problem is how to decrease the traffic exchanged with other ISPs, because of high settlement costs. By using the ALTO service to optimize traffic, a small ISP can define two "optimization areas": one is its own network; the other one consists of all other network destinations. The cost map can be defined as follows: the cost of link between clients of inner ISP's networks is lower than between clients of outer ISP's networks and clients of inner ISP's network. As a result, a host with ALTO client inside the network of this ISP will prefer retrieving data from hosts connected to the same ISP.

An example is given in Figure 9. It is assumed that ISP A is a small ISP only having one access network. As operator of the ALTO service, ISP A can define its network 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 cost inside the network of ISP A. C2 is denoted as the cost from PID2 to PID1, and C3 from PID1 to PID2. For the sake of simplifity, in the following C2=C3 is assumed. In order to keep traffic local inside ISP A, it makes sense to define: $C1 < C2$

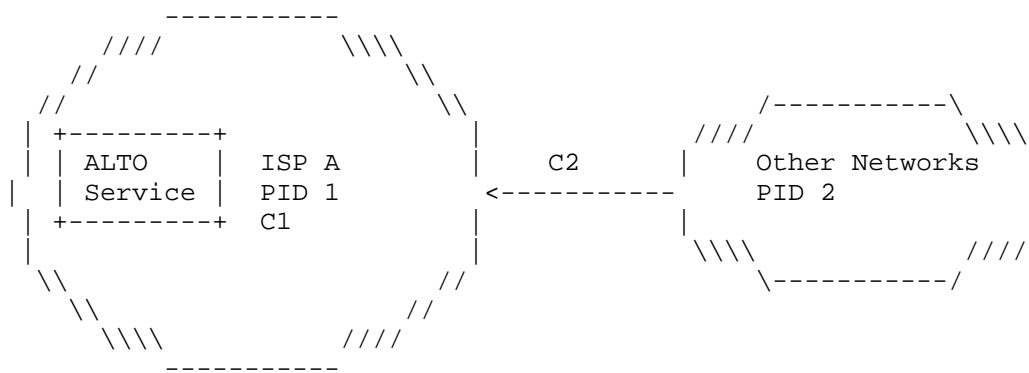


Figure 9: Example ALTO deployment in small ISPs

A simplified extract of the corresponding ALTO network and cost maps is listed in Figure 10 and Figure 11, assuming that the network of ISP A has the IPv4 address ranges 192.0.2.0/24 and 198.51.100.0/25. In this example, the cost values C1 and C2 can be set to any number $C1 < C2$.

```
HTTP/1.1 200 OK
...
Content-Type: application/alto-networkmap+json

{
```

```

...
"network-map" : {
  "PID1" : {
    "ipv4" : [
      "192.0.2.0/24",
      "198.51.100.0/25"
    ]
  },
  "PID2" : {
    "ipv4" : [
      "0.0.0.0/0"
    ],
    "ipv6" : [
      "::/0"
    ]
  }
}
}

```

Figure 10: Example ALTO network map

HTTP/1.1 200 OK

```

...
Content-Type: application/alto-costmap+json

{
  ...
  "cost-type" : { "cost-mode" : "numerical",
                  "cost-metric": "routingcost"
                },
  "cost-map" : {
    "PID1": { "PID1": C1, "PID2": C2 },
    "PID2": { "PID1": C2, "PID2": 0 },
  }
}

```

Figure 11: Example ALTO cost map

3.4.2. ISP with Several Fixed Access Networks

For a large ISP with a fixed network comprising several access networks and a core network, the traffic optimizing problems will include (1) using the backbone network efficiently, (2) adjusting the traffic balance in different access networks according to traffic conditions and management policies, and (3) achieving a reduction of settlement costs with other ISPs.

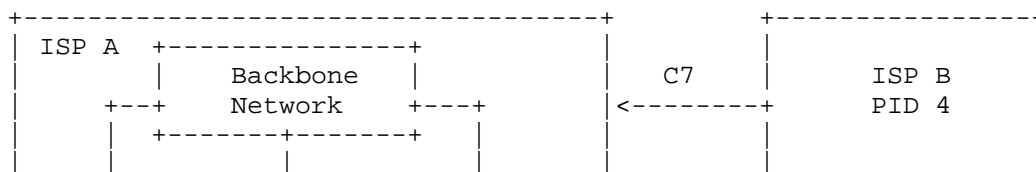
Such a large ISP deploying an ALTO service may want to optimize its traffic according to the network topology of its access networks. For example, each access network could be defined to be one optimization area, i.e., traffic should be kept locally within that area if possible. Then the costs between those access networks can be defined according to a corresponding traffic optimizing requirement by this ISP. One example setup is further described below and also shown in Figure 12.

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 on the backbone network inside the Autonomous System of ISP A; and the second requirement is to decrease the P2P traffic to other ISPs, i.e., other Autonomous Systems. The second requirement can be assumed to have priority over the first one. Also, we assume that the settlement rate with ISP B is lower than with other ISPs. Then ISP A can deploy an ALTO service to meet these traffic optimization requirements. In the following, we will give an example of an ALTO setting and configuration according to these requirements.

In inner network of ISP A, we can define each access network to be one optimization area, and assign one PID to each access network, such as PID 1, PID 2, and PID 3. Because of different peerings with different outer ISPs, we define ISP B to be one optimization area, and we assign PID 4 to it. We define all other networks to be one optimization area and assign PID 5 to it.

We assign costs (C_1 , C_2 , C_3 , C_4 , C_5 , C_6 , C_7 , C_8) as shown in Figure 12. Cost C_1 is denoted as the link cost in inner AN A (PID 1), and C_2 and C_3 are defined accordingly. C_4 is denoted as the link cost from PID 1 to PID 2, and C_5 is the corresponding cost from PID 3, which is assumed to have a similar value. C_6 is the cost between PID 1 and PID 3. For simplicity, we assume symmetrical costs between the AN in this example. C_7 is denoted as the link cost from the ISP B to ISP A. C_8 is the link cost from other networks to ISP A.

According to previous discussion of the first requirement and the second requirement, the relationship of these costs will be defined as: $(C_1, C_2, C_3) < (C_4, C_5, C_6) < (C_7) < (C_8)$



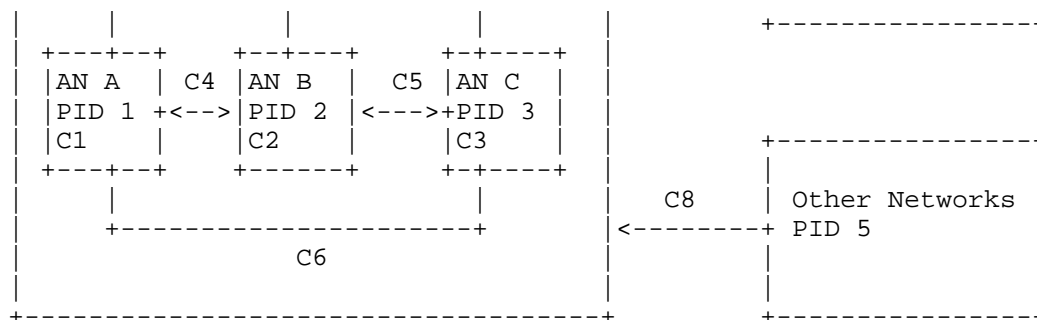


Figure 12: ALTO deployment in large ISPs with layered fixed network structures

3.4.3. ISP with Fixed and Mobile Network

An ISP with both mobile network and fixed network my focus on optimizing the mobile traffic by keeping traffic in the fixed network as far as possible, because wireless bandwidth is a scarce resource and traffic is costly in mobile network. In such a case, the main requirement of traffic optimization could be decreasing the usage of radio resources in the mobile network. An ALTO service can be deployed to meet these needs.

Figure 13 shows an example: ISP A operates one mobile network, which is connected to a backbone network. The ISP also runs two fixed access networks AN A and AN B, which are also connected to the backbone network. In this network structure, the mobile network can be defined as one optimization area, and PID 1 can be assigned to it. Access networks AN A and B can also be defined as optimization areas, and PID 2 and PID 3 can be assigned, respectively. The cost values are then defined as shown in Figure 13.

To decrease the usage of wireless link, the relationship of these costs can be defined as follows:

From view of mobile network: $C4 < C1$. This means that clients in mobile network requiring data resource from other clients will prefer clients in AN A to clients in the mobile network. This policy can decrease the usage of wireless link and power consumption in terminals.

From view of AN A: $C2 < C6$, $C5 = \text{maximum cost}$. This means that clients in other optimization area will avoid retrieving data from the mobile network.

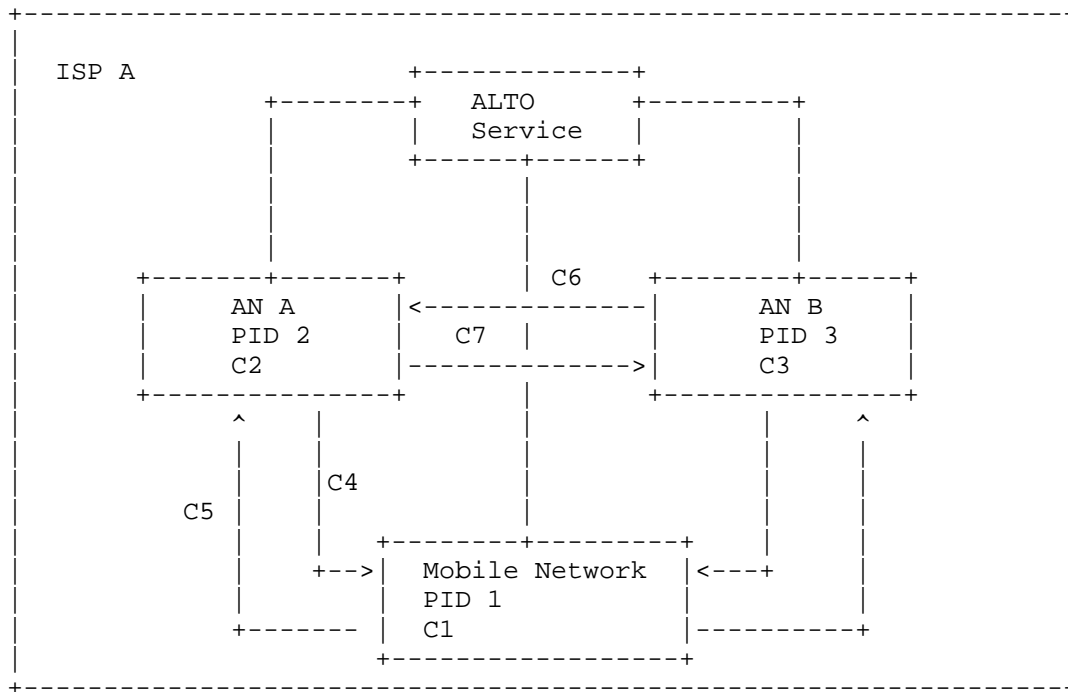


Figure 13: ALTO deployment in ISPs with mobile network

3.5. Deployment Experiences

The examples in the previous section are simple and do not consider specific requirements inside access networks, such as different link types. Deploying an ALTO service in real network will have to require further network conditions and requirements. One real example is described in greater detail in reference [I-D.lee-alto-chinatelecom-trial].

Also, experiments have been conducted with ALTO-like deployments in Internet Service Provider (ISP) networks. For instance, NTT performed tests with their HINT server implementation and dummy nodes to gain insight on how an ALTO-like service influence peer-to-peer systems [I-D.kamei-p2p-experiments-japan]. The results of an early experiment conducted in the Comcast network are documented in [RFC5632].

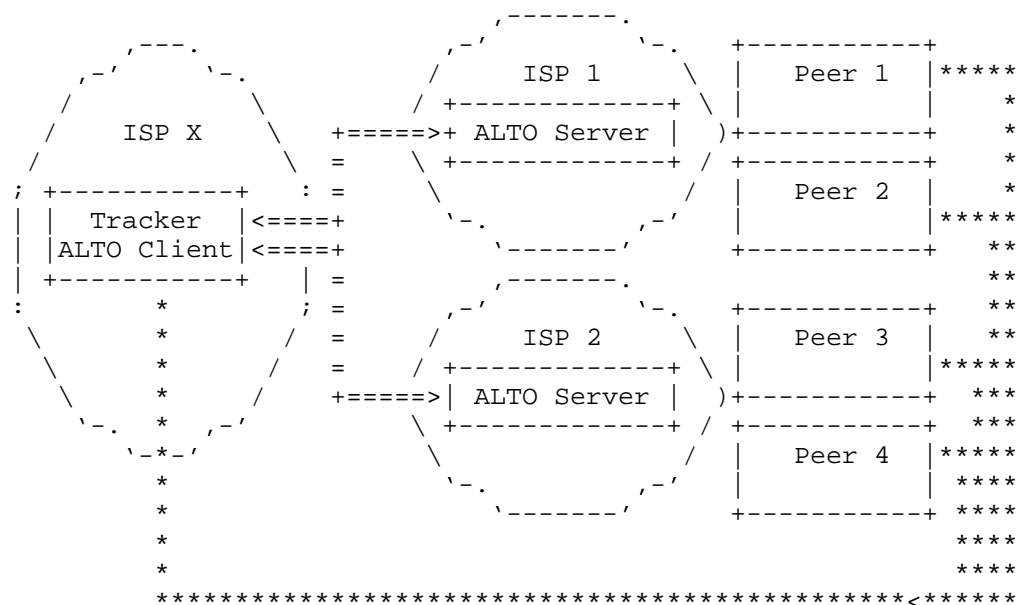
4. Using ALTO for P2P Traffic Optimization

4.1. Overview

4.1.1. Usage Scenario

The scope of this section is the interaction of peer-to-peer applications that use a centralized resource directory ("tracker"), with the ALTO service. In this scenario, the resource consumer ("peer") asks the resource directory for a list of candidate resource providers, which can provide the desired resource.

For efficiency reasons (i.e., message size), usually only a subset of all resource providers known to the resource directory will be returned to the resource consumer. Some or all of these resource providers, plus further resource providers learned by other means such as direct communication between peers, will be contacted by the resource consumer for accessing the resource. The purpose of ALTO is giving guidance on this peer selection, which is supposed to yield better-than-random results. The tracker response as well as the ALTO 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.



Legend:

=== ALTO client protocol

*** Application protocol

Figure 14: Global tracker accessing ALTO server at various ISPs

Figure 14 depicts a tracker-based system, in which 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 giving 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.

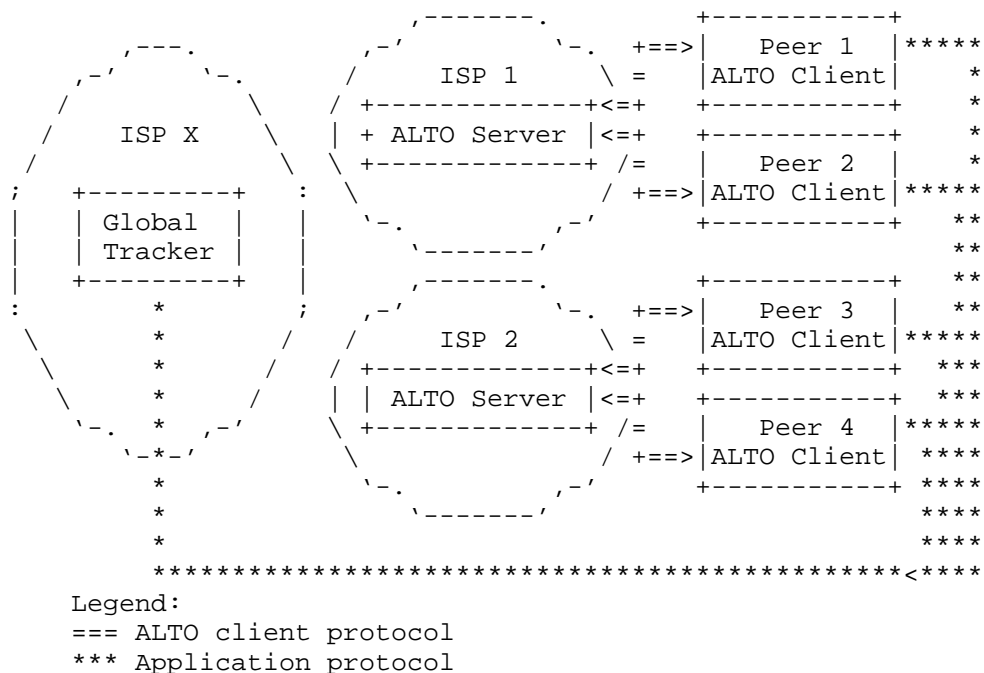


Figure 15: Global Tracker - Local ALTO Servers

The scenario in Figure 15 lets the peers directly communicate with their ISP's ALTO server (i.e., ALTO client embedded in the peers), giving thus the peers the most control on which information they query for, as they can integrate information received from trackers and through direct peer-to-peer knowledge exchange.

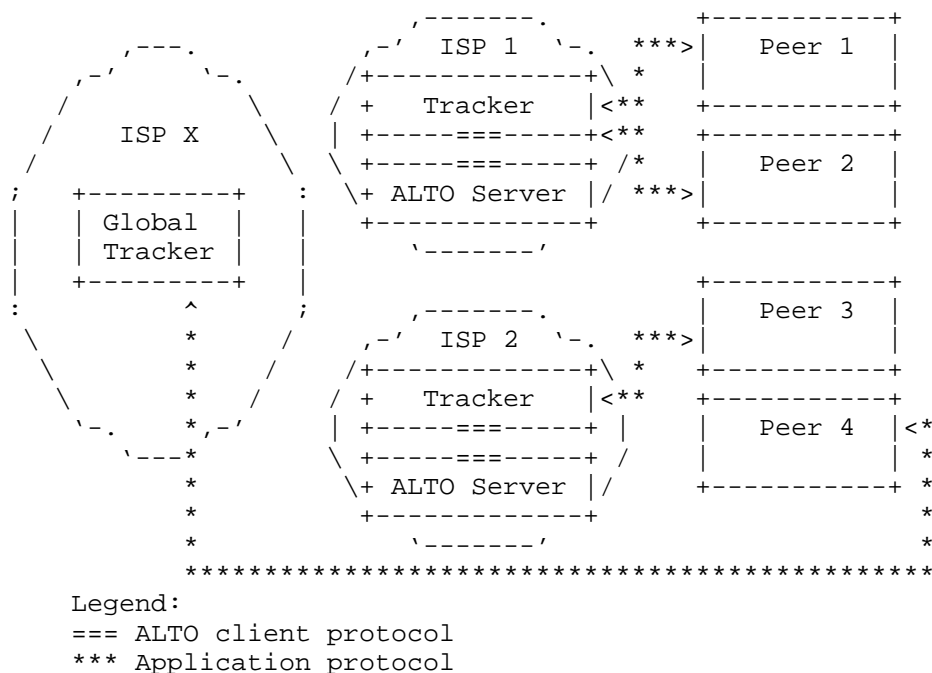


Figure 16: P4P approach with local tracker and local ALTO server

There are some attempts to let ISP's to deploy their own trackers, as shown in Figure 16. In this case, the client has no chance to get guidance from the ALTO server, other than talking to the ISP's tracker. However, the peers would have still chance the contact other trackers, deployed by entities other than the peer's ISP.

Figure 16 and Figure 14 ostensibly take peers the possibility to directly query the ALTO server, if the communication with the ALTO server is not permitted for any reason. However, considering the plethora of different applications of ALTO, e.g., multiple tracker and non-tracker based P2P systems and or applications searching for relays, it seems to be beneficial for all participants to let the peers directly query the ALTO server. The peers are also the single point having all operational knowledge to decide whether to use the ALTO guidance and how to use the ALTO guidance. This is a preference for the scenario depicted in Figure Figure 15.

4.1.2. Applicability of ALTO

TODO

4.2. Deployment Recommendations

4.2.1. ALTO Services

In case of peer-to-peer networks, there is basically a dilemma which ALTO service to use: The Cost Map Service is seen as the only working solution by peer-to-peer software vendors and the Endpoint Cost Service 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.

4.2.2. Guidance Considerations

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 18), 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 20), 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).

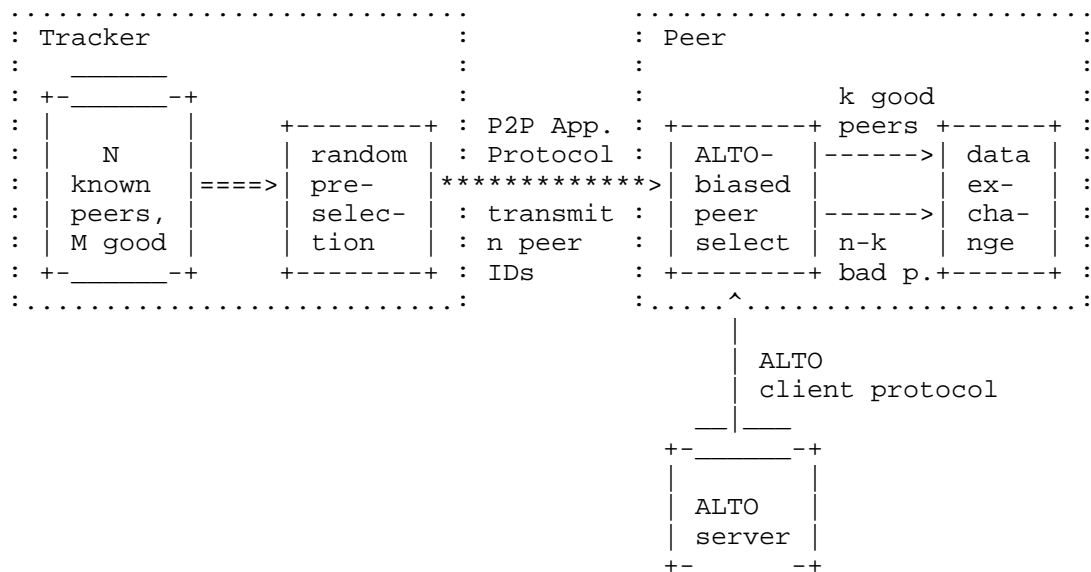


Figure 17: Tracker-based P2P Application with random peer preselection

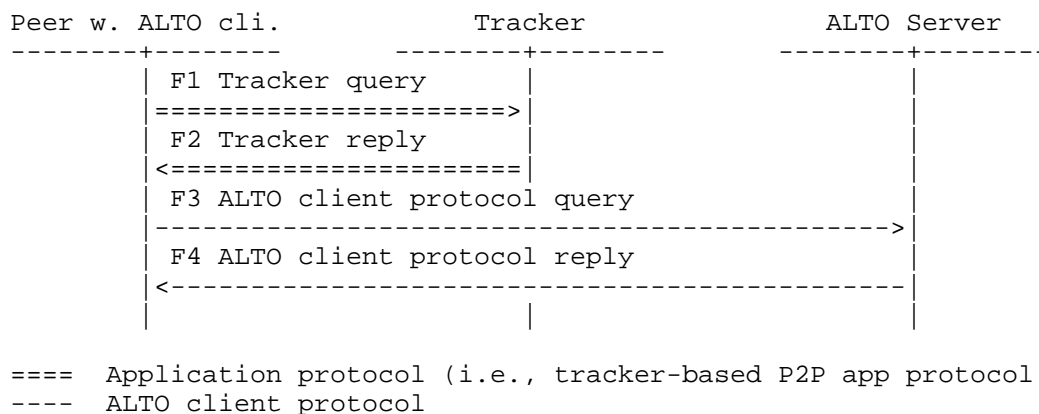
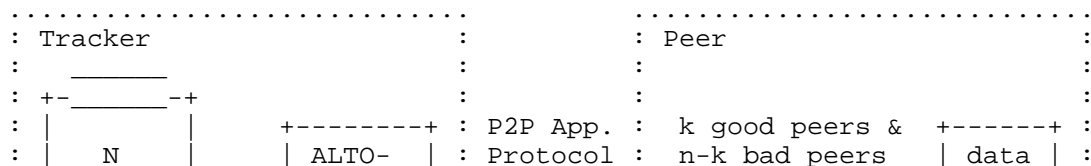


Figure 18: Basic message sequence chart for resource consumer-initiated ALTO query



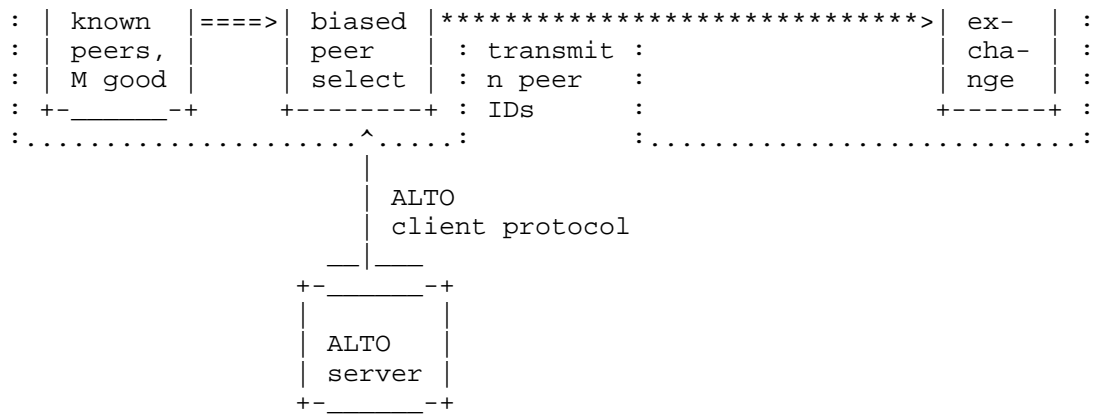


Figure 19: Tracker-based P2P Application with ALTO client in tracker

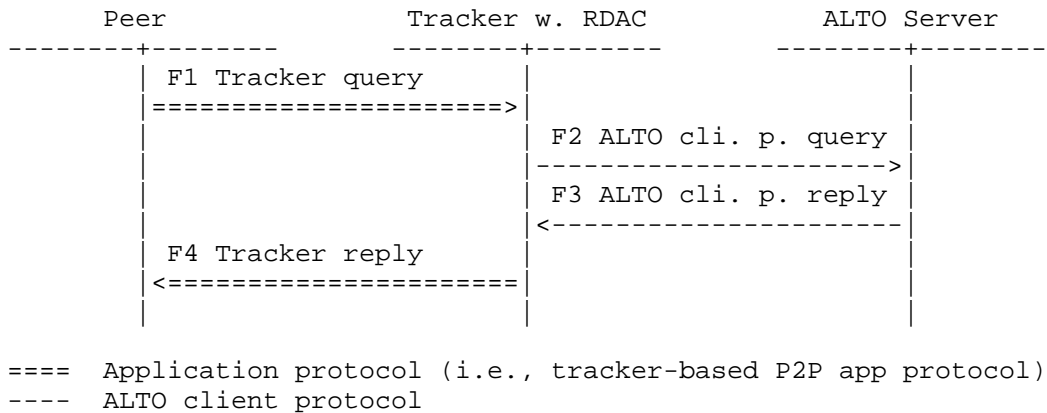


Figure 20: Basic message sequence chart for third-party ALTO query

Note: the message sequences depicted in Figure 18 and Figure 20 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:

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

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

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 that respective peer.

5. Using ALTO for CDNs

5.1. Overview

5.1.1. Usage Scenario

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).

5.1.2. Applicability of ALTO

TODO: Rewording required

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.

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.

5.2. Deployment Recommendations

5.2.1. ALTO Services

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).

5.2.2. Guidance Considerations

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.

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.

6. Other Use Cases

This section briefly surveys and references other use cases that have been suggested for ALTO.

6.1. Monitoring Data Reporting

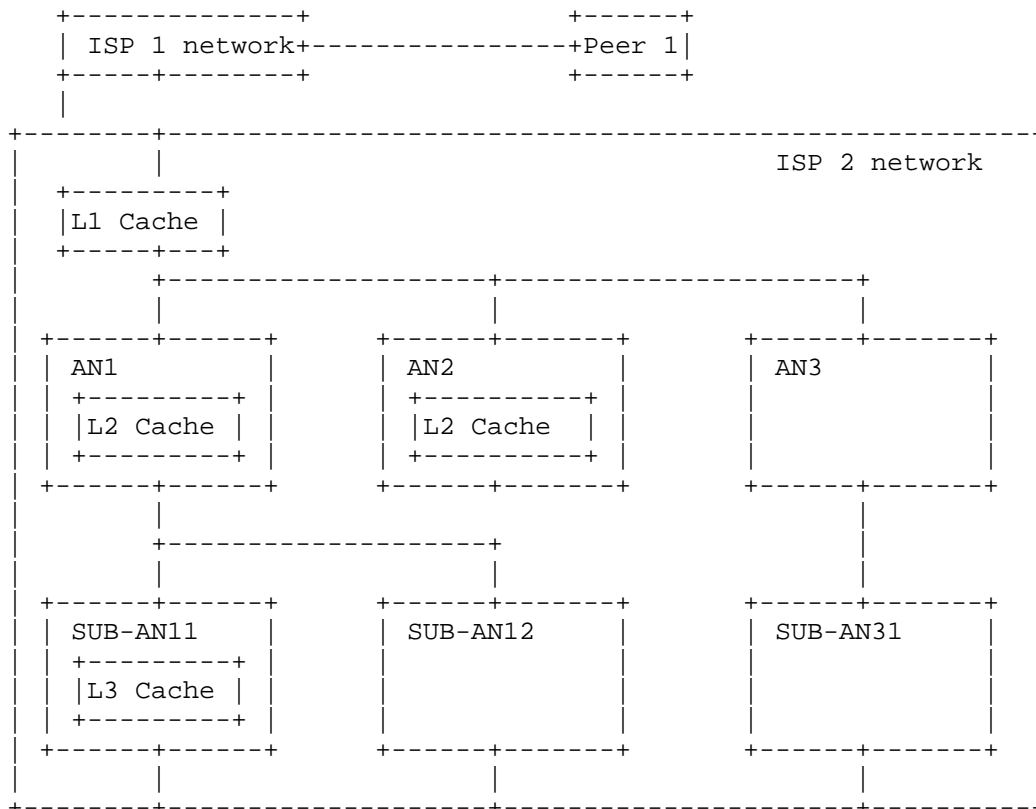
TODO

6.2. Virtual Private Networks (VPNs)

TODO

6.3. In-Network Caching

Deployment of intra-domain P2P caches has been proposed for a cooperations between the network operator and the P2P service providers, e.g., to reduce the bandwidth consumption in access networks [I-D.deng-alto-p2pcache].



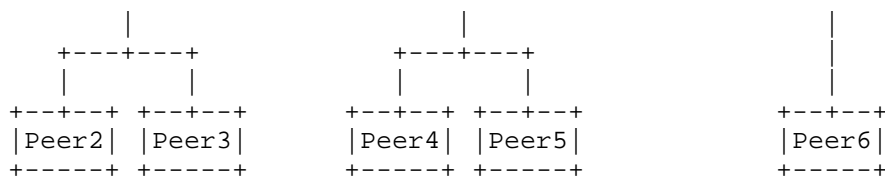


Figure 21: General architecture of intra-ISP caches

Figure 21 depicts the overall architecture of a potential P2P cache deployments inside an ISP 2 with various access network types. As shown in the figure, P2P caches may be deployed at various levels, including the interworking gateway linking with other ISPs, internal access network gateways linking with different types of accessing networks (e.g. WLAN, cellular and wired), and even within an accessing network at the entries of individual WLAN sub-networks. Moreover, depending on the network context and the operator's policy, each cache can be a Forwarding Cache or a Bidirectional Cache [I-D.deng-alto-p2pcache].

In such a cache architecture, the locations of caches could be used as dividers of different PIDs to guide intra-ISP network abstraction and mark costs among them according to the location and type of relevant caches.

Further details and deployment considerations can be found in [I-D.deng-alto-p2pcache].

7. Security Considerations

The ALTO protocol itself as well as the ALTO client and server raise new security issues beyond the ones 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.

7.1. 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 considered 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.

7.2. 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 15). 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 14), e.g., by a third-party tracker or by a CDN system outside the ISP's network, the access restrictions have to be looser. 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.

7.3. Faking ALTO Guidance

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

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

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

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

8. Conclusion

This document discusses how the ALTO protocol can be deployed in different use cases and provides corresponding guidance and recommendations to network administrators and application developers.

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

9.2. Informative References

- [I-D.deng-alto-p2pcache] Lingli, D., Chen, W., Yi, Q., and Y. Zhang, "Considerations for ALTO with network-deployed P2P caches", draft-deng-alto-p2pcache-02 (work in progress), July 2013.

- [I-D.ietf-alto-protocol]
Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-20 (work in progress), October 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-10 (work in progress), September 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-hl2]
Kiesel, S. and M. Stiemerling, "ALTO H12", draft-kiesel-alto-hl2-02 (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. Appendix: 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?]

A.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.

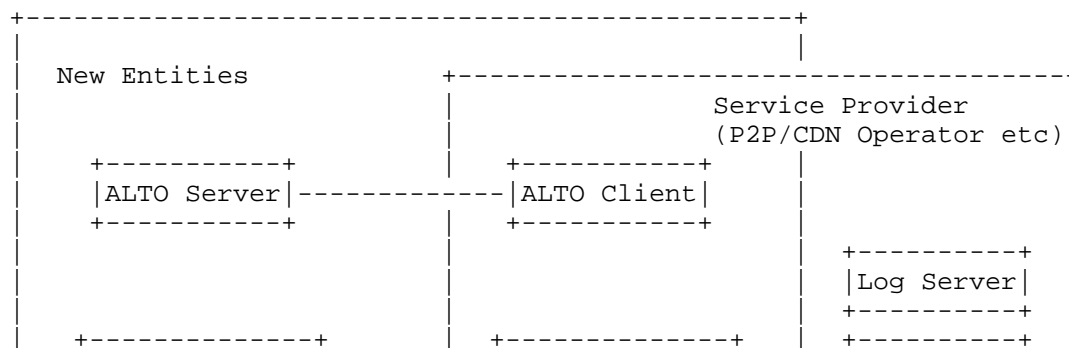
A.2. Monitoring Data Sources

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.

A.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.



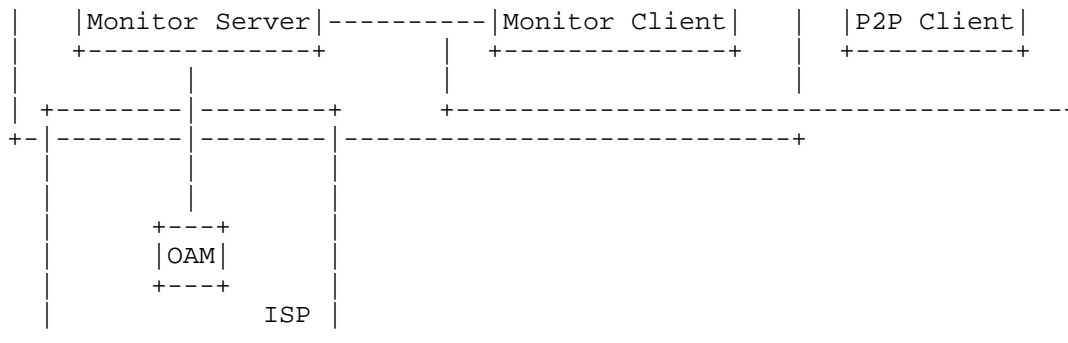


Figure 22: Monitoring Structure

Appendix B. Appendix: API between ALTO Client and Application

This section 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.

Appendix C. 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 text on ISP deployment requirements and monitoring.
- o Stefano Previdi contributed parts of the Section 5 on "Using ALTO for CDNs".
- o Rich Woundy contributed text to Section 3.3.
- o Lingli Deng, Wei Chen, Qiuchao Yi, Yan Zhang contributed Section 6.3.
- o Thomas-Rolf Banniza carefully reviewed the document.

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

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

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

Email: sprevidi@cisco.com

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

Email: michael.scharf@alcatel-lucent.com

ALTO WG
Internet-Draft
Intended status: Standards Track
Expires: April 5, 2014

R. Alimi, Ed.
Google
R. Penno, Ed.
Cisco Systems
Y. Yang, Ed.
Yale University
October 2, 2013

ALTO Protocol
draft-ietf-alto-protocol-20.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 network 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 April 5, 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	13
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	17
6.2.	Cost Map Structure	18
6.3.	Network Map and Cost Map Dependency	18
6.4.	Cost Map Update	19
7.	Endpoint Properties	19
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	20
8.3.	Basic Operations	21
8.3.1.	Client Discovering Information Resources	21
8.3.2.	Client Requesting Information Resources	21
8.3.3.	Server Responding to IR Request	22
8.3.4.	Client Handling Server Response	22
8.3.5.	Authentication and Encryption	23
8.3.6.	Information Refreshing	23
8.3.7.	HTTP Cookies	23
8.3.8.	Parsing of Unknown Fields	24

8.4.	Server Response Encoding	24
8.4.1.	Meta Information	24
8.4.2.	Data Information	24
8.5.	Protocol Errors	25
8.5.1.	Media Type	25
8.5.2.	Response Format and Error Codes	25
8.5.3.	Overload Conditions and Server Unavailability	26
9.	Protocol Specification: Information Resource Directory	27
9.1.	Information Resource Attributes	27
9.1.1.	Resource ID	27
9.1.2.	Media Type	27
9.1.3.	Capabilities	28
9.1.4.	Accepts Input Parameters	28
9.1.5.	Dependent Resources	28
9.2.	Information Resource Directory (IRD)	28
9.2.1.	Media Type	28
9.2.2.	Encoding	29
9.2.3.	Example	31
9.2.4.	Delegation using IRD	33
9.2.5.	Considerations of Using IRD	35
10.	Protocol Specification: Basic Data Types	36
10.1.	PID Name	36
10.2.	Resource ID	36
10.3.	Version Tag	36
10.4.	Endpoints	37
10.4.1.	Address Type	37
10.4.2.	Endpoint Address	37
10.4.3.	Endpoint Prefixes	38
10.4.4.	Endpoint Address Group	38
10.5.	Cost Mode	39
10.6.	Cost Metric	39
10.7.	Cost Type	40
10.8.	Endpoint Property	40
10.8.1.	Resource Specific Endpoint Properties	40
10.8.2.	Global Endpoint Properties	41
11.	Protocol Specification: Service Information Resources	41
11.1.	Meta Information	41
11.2.	Map Service	41
11.2.1.	Network Map	41
11.2.2.	Cost Map	44
11.3.	Map Filtering Service	46
11.3.1.	Filtered Network Map	47
11.3.2.	Filtered Cost Map	49
11.4.	Endpoint Property Service	53
11.4.1.	Endpoint Property	54
11.5.	Endpoint Cost Service	57
11.5.1.	Endpoint Cost	57
12.	Use Cases	60

12.1. ALTO Client Embedded in P2P Tracker	61
12.2. ALTO Client Embedded in P2P Client: Numerical Costs	62
12.3. ALTO Client Embedded in P2P Client: Ranking	63
13. Discussions	64
13.1. Discovery	64
13.2. Hosts with Multiple Endpoint Addresses	65
13.3. Network Address Translation Considerations	65
13.4. Endpoint and Path Properties	66
14. IANA Considerations	66
14.1. application/alto-* Media Types	66
14.2. ALTO Cost Metric Registry	67
14.3. ALTO Endpoint Property Type Registry	69
14.4. ALTO Address Type Registry	69
14.5. ALTO Error Code Registry	70
15. Security Considerations	71
15.1. Authenticity and Integrity of ALTO Information	71
15.1.1. Risk Scenarios	71
15.1.2. Protection Strategies	71
15.1.3. Limitations	72
15.2. Potential Undesirable Guidance from Authenticated ALTO Information	72
15.2.1. Risk Scenarios	72
15.2.2. Protection Strategies	72
15.3. Confidentiality of ALTO Information	73
15.3.1. Risk Scenarios	73
15.3.2. Protection Strategies	73
15.3.3. Limitations	74
15.4. Privacy for ALTO Users	74
15.4.1. Risk Scenarios	74
15.4.2. Protection Strategies	74
15.5. Availability of ALTO Service	75
15.5.1. Risk Scenarios	75
15.5.2. Protection Strategies	75
16. Manageability Considerations	75
16.1. Operations	75
16.1.1. Installation and Initial Setup	76
16.1.2. Migration Path	76
16.1.3. Requirements on Other Protocols and Functional Components	76
16.1.4. Impact and Observation on Network Operation	77
16.2. Management	77
16.2.1. Management Interoperability	77
16.2.2. Management Information	78
16.2.3. Fault Management	78
16.2.4. Configuration Management	78
16.2.5. Performance Management	78
16.2.6. Security Management	79
17. References	79

17.1. Normative References	79
17.2. Informative References	80
Appendix A. Acknowledgments	82
Appendix B. Design History and Merged Proposals	83
Appendix C. Authors	84
Authors' Addresses	84

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 (i.e., the 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-random") 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

We use the term ALTO Information Base to refer to the internal representation of ALTO Information maintained by an 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.

The ALTO Service specified in this document 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. One may

note that the "my-Internet view" defined in this document does not specify the internal topology of a network, and hence, we say that it provides a "single-switch" abstraction. Extensions to this document may provide topology details in "my-Internet view".

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.

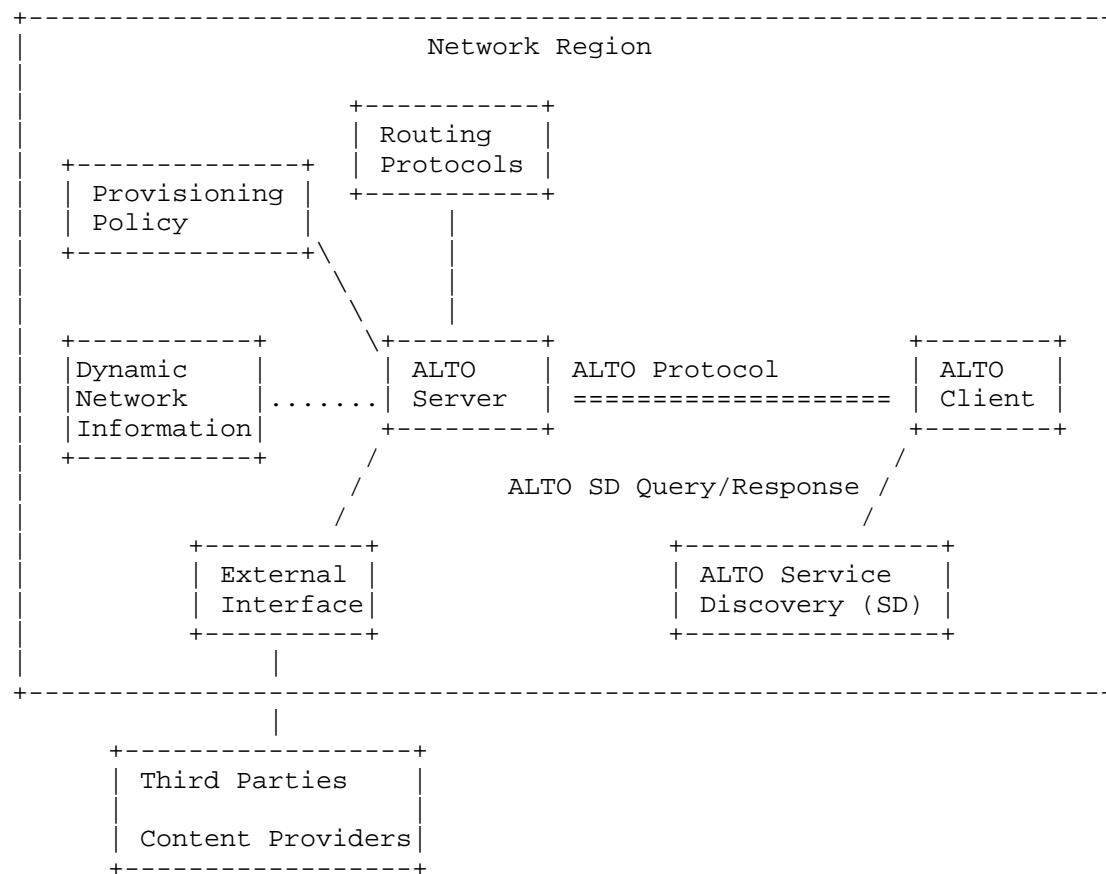


Figure 1: Basic ALTO Architecture.

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 request 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.

The goals of the services defined in this document are to convey (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. The aforementioned goals are achieved 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.

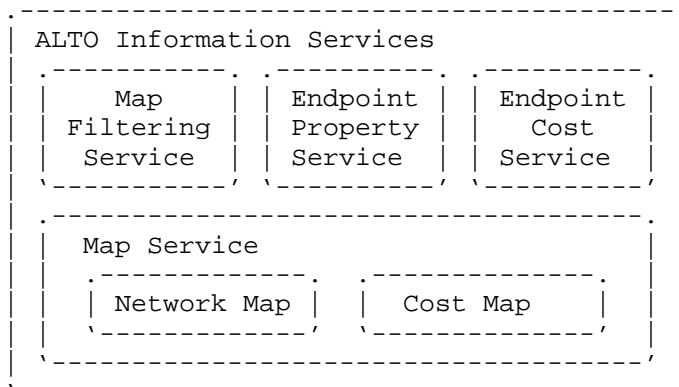


Figure 2: ALTO Service Framework.

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. A Network Map (See Section 5) provides a full set of Network Location groupings defined by the ALTO Server and the Endpoints contained within each grouping. A Cost Map (see Section 6) provides costs between a 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 10.1) 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. Interpreting the PIDs defined in a Network Map using the "single-switch" abstraction, one can consider that each PID represents an abstract port (PoP) that connects a set of endpoints.

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 endhosts 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.

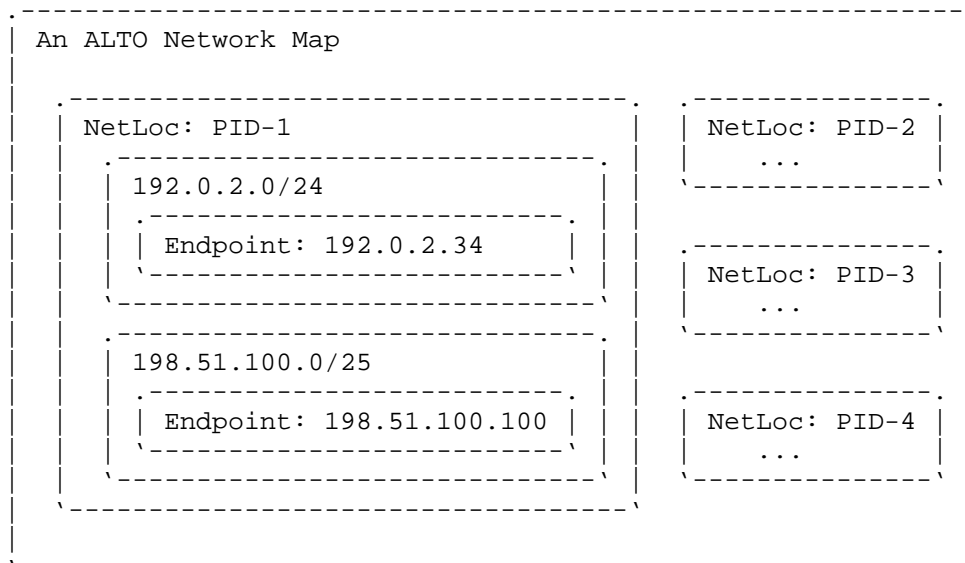


Figure 3: Example Network Map.

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 policy.

For a given Network Map, an ALTO Cost Map defines Path Costs pairwise amongst sets of source and destination Network Locations defined by PIDs defined in the Network Map. Each Path Cost is the end-to-end cost when a unit of traffic goes from the source to the destination.

Since 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, a 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. For a given ALTO Server, the combination of Cost Type and Network Map defines a key. In other words, an ALTO Server MUST NOT define two Cost Maps with the same Cost Type, Network Map pair.

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 that 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 request 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 request specifies 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 -> Dst_1, ..., Src_1 -> Dst_n, ..., Src_m -> Dst_1, ..., Src_m -> 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 for each Network Map that it provides.

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: An ALTO Server provides a set of network information resources. Each information resource has a media type [RFC2046]. An ALTO Client may construct an HTTP request for a particular information resource (including any parameters, if necessary), and the 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 fields (names/values) of JSON objects. An optional field 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 fields are case sensitive.

For example, the definition below defines a new type Type4, with three field members (or fields for short) named "name1", "name2", and "name3" respectively. The field named "name3" is optional, and the field 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 defined as any type.

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

We use subtyping to denote that one type is derived from another type. The example below denotes that TypeDerived is derived from TypeBase. TypeDerived includes all fields defined in TypeBase. If TypeBase does not have a field named "name1", TypeDerived will have a new field named "name1". If TypeBase already has a field named "name1" but with a different type, TypeDerived will have a field named "name1" with the type defined in TypeDerived (i.e., Type1 in the example).

```
object {  
    Type1    name1;  
} TypeDerived : TypeBase;
```

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 Operations

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 Information Resource Directories. Informally, an Information Resource Directory enumerates URIs at which an ALTO Server offers Information Resources.

Specifically, using the ALTO Discovery protocol, an ALTO Client obtains a URI through which it can request an Information Resource Directory (IRD). We refer to this IRD as the Root IRD of the ALTO Client. Each entry in an IRD indicates a URI at which an ALTO Server accepts requests, and returns either an Information Resource or an Information Resource Directory that references additional Information Resources. Beginning with its Root IRD and following links to IRDs recursively, an ALTO Client can discover all Information Resources available to it. We refer to this set of Information Resources as the Information Resource Closure of the ALTO Client. By inspecting its Information Resource Closure, 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.

See Section 9.2 for a detailed specification on IRDs.

8.3.2. Client Requesting Information Resources

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 8.5 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 9.2 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 Refreshing

An ALTO Client MAY determine the frequency at which ALTO Information is refreshed based on 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 of Unknown Fields

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

8.4. Server Response Encoding

Though each type of ALTO Server response (i.e., an Information Resource Directory, an individual Information Resource, or an error message) has its distinct syntax and hence its unique Media Type, they are designed to have a similar structure: a meta field providing meta definitions, and another field containing the data, if needed.

Specifically, we define the base type of each ALTO Server response as `ResponseEntityBase`:

```
object {  
  ResponseMeta          meta;  
} ResponseEntityBase;
```

with field:

meta meta-information pertaining to the response.

8.4.1. Meta Information

Meta information is encoded as a map object for flexibility. Specifically, `ResponseMeta` is defined as:

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

8.4.2. Data Information

The data component of the response encodes the response-specific data. In this document, we derive five types from `ResponseEntityBase` to add different types of data component: `InforResourceDirectory` (Section 9.2.2), `InfoResourceNetworkMap` (Section 11.2.1.6), `InfoResourceCostMap` (Section 11.2.2.6), `InfoResourceEndpointProperties` (Section 11.4.1.6), and `InfoResourceEndpointCostMap` (Section 11.5.1.6).

8.5. 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.

8.5.1. Media Type

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

8.5.2. Response Format and Error Codes

An ALTO Error Response MUST include the "code" key in the "meta" field of the response. The value of "code" MUST be 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_MISSING_FIELD	A required JSON field is missing
E_INVALID_FIELD_TYPE	The type of the value of a JSON field is invalid
E_INVALID_FIELD_VALUE	The value of a JSON field is invalid

Table 1: Defined ALTO Error Codes.

After an ALTO Server receives a request, it needs to verify the syntactic and semantic validity of the request. The following paragraphs in this section are intended to illustrate the usage of the error codes defined above during the verification. An individual implementation may define its message processing in a different

order.

In the first step after an ALTO Server receives a request, it checks the syntax of the request body (i.e., whether the JSON structure can be parsed), and indicates a syntax error using the error code `E_SYNTAX`.

A request without syntax errors may still be invalid. An error case is that the request misses a required field. The server indicates such an error using the error code `E_MISSING_FIELD`. This document defines required fields for Network Map Filtering (Section 11.3.1.3), Cost Map Filtering (Section 11.3.2.3), Endpoint Properties (Section 11.4.1.3), and Endpoint Cost (Section 11.5.1.3). For an `E_MISSING_FIELD` error, the server may include the optional "field" key in the "meta" field of the response, to indicate the missing field.

A request with the correct fields might use a wrong type for the value of a field. For example, the value of a field could be a `JSONString` when a `JSONNumber` is expected. The server indicates such an error using the error code `E_INVALID_FIELD_TYPE`. The server may include the optional "field" key in the "meta" field of the response, to indicate the field that contains the wrong type.

A request with the correct fields and types of values for the fields may specify a wrong value for a field. For example, a cost map filtering request may specify a wrong value of `CostMode` in the "cost-type" field (Section 11.3.2.3). The server indicates such an error with the error code `E_INVALID_FIELD_VALUE`. For an `E_INVALID_FIELD_VALUE` error, the server may include the optional "field" key in the "meta" field of the response, to indicate the field that contains the wrong value. The server may also include the optional "value" key in the "meta" field of the response to indicate the wrong value that triggered the error.

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 techniques.

8.5.3. 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: Information Resource Directory

As we discussed, an ALTO Client starts by retrieving an Information Resource Directory, which specifies the attributes of individual Information Resources that an ALTO Server provides.

9.1. Information Resource Attributes

In this document, each Information Resource has five attributes associated with it, including its assigned ID, its response format, its capabilities, its accepted input parameters, and other resources that it may depend on. The function of an Information Resource Directory is to publishes these attributes.

9.1.1. Resource ID

Each Information Resource that an ALTO Client can request MUST be assigned an ID that is unique amongst all Information Resources in the Information Resource Closure of the client. The ID SHOULD remain stable even when the data provided by that resource changes. 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. IDs SHOULD NOT be re-used for different resources over time.

9.1.2. 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.

9.1.3. Capabilities

The Capabilities attribute of an Information Resource 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, then the Server advertises the cost-constraints capability of the Cost Map Information Resource.

9.1.4. 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.

9.1.5. Dependent Resources

The information provided in an Information Resource may use information provided in some other resources (e.g., a Cost Map uses the PIDs defined in a Network Map). The uses attribute conveys such information.

9.2. Information Resource Directory (IRD)

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].

9.2.1. Media Type

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

9.2.2. Encoding

An Information Resource Directory response may include in "meta" the "cost-types" key, whose value is of type IRDMetaCostTypes defined below, where CostType is defined in Section 10.7:

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

The function of "cost-types" is to assign names to a set of CostTypes that can be used in one or more "resources" entries in the IRD to simplify specification. The names defined in "cost-types" in an IRD are local to the IRD.

For a Root IRD, "meta" MUST include the "default-alto-network-map" key, which specifies the Resource ID of a Network Map. When there are multiple Network Maps defined in an IRD (e.g., with different levels of granularity), the "default-alto-network-map" key provides a guideline to simple clients that use only one Network Map.

The data component of an Information Resource Directory response is named "resources", which is a JSON object of type IRDResourceEntries:

```
object {  
  IRDResourceEntries resources;  
} InfoResourceDirectory : ResponseEntityBase;
```

```
object-map {  
  ResourceID -> IRDResourceEntry;  
} IRDResourceEntries;
```

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

```
object {  
  ...  
}
```


} Capabilities;

An IRDResourceEntries object is a dictionary map keyed by ResourceIDs, where ResourceID is defined in Section 10.2. The value of 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 to the URI of the IRD and MUST be resolved according to Section 5 of [RFC3986].

media-type The media type of Information Resource (see Section 9.1.2) 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 9.1.4) accepted by POST requests to the corresponding URI. If this field 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 field 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, defined in the same IRD, that define the resources on which this resource directly depends. An ALTO Server SHOULD include in this list any resources 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. 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.

9.2.3. Example

The following is an example Information Resource Directory returned by an ALTO Server to an ALTO Client. Assume it is the Root IRD of the Client.

```
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"
      }
    },
    "default-alto-network-map" : "my-default-network-map"
  },
  "resources" : {
    "my-default-network-map" : {
      "uri" : "http://alto.example.com/networkmap",
      "media-type" : "application/alto-networkmap+json"
    },
    "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": [ "my-default-network-map" ]
    },
    "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": [ "my-default-network-map" ]
    },
    "custom-maps-resources" : {
        "uri" : "http://custom.alto.example.com/maps",
        "media-type" : "application/alto-directory+json"
    },
    "endpoint-property" : {
        "uri" : "http://alto.example.com/endpointprop/lookup",
        "media-type" : "application/alto-endpointprop+json",
        "accepts" : "application/alto-endpointpropparams+json",
        "capabilities" : {
            "prop-types" : [ "my-default-network-map.pid",
                           "priv:ietf-example-prop" ]
        },
    },
    "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 "cost-types" key of "meta" of the example IRD defines names for four cost types in 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". This name is used in the second entry of "resources", which defines a Cost Map. In particular, 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".

Another key defined in "meta" of the example IRD is "default-alto-network-map", which has value "my-default-network-map", which is the Resource ID of a Network Map that will be defined in "resources".

The "resources" field of the example IRD defines six Information Resources. For example, the second entry, which is assigned a Resource ID "numerical-routing-cost-map", provides a Cost Map, as indicated by the media-type "application/alto-costmap+json". The Cost Map is based on the Network Map defined with Resource ID "my-default-network-map". As another example, the last entry, which is assigned Resource ID "endpoint-cost", provides 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" field of the "capabilities" attribute of the Information Resource includes four defined cost types specified in the "cost-types" key of "meta" 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" field of the "capabilities" attribute.

9.2.4. Delegation using IRD

ALTO Information Resource Directory provides flexibility to provide ALTO Service (e.g., delegation to another domain). Consider the preceding example. Assume that the ALTO Server running at alto.example.com wants to delegate some Information Resources to a separate subdomain: "custom.alto.example.com". In particular, assume that the maps available via this subdomain are filtered Network Maps, filtered Cost Maps, and some pre-generated maps for the "hopcount" and "routingcost" Cost Metrics in the "ordinal" Cost Mode. The fourth entry of "resources" in the preceding example IRD implements the delegation. The entry has a media-type of "application/alto-directory+json", and an ALTO Client can discover the Information Resources available at "custom.alto.example.com" if its request to "http://custom.alto.example.com/maps" is successful:

```
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" : {
    "filtered-network-map" : {
      "uri" : "http://custom.alto.example.com/networkmap/filtered",
      "media-type" : "application/alto-networkmap+json",
      "accepts" : "application/alto-networkmapfilter+json",
      "uses": [ "my-default-network-map" ]
    },
    "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": [ "my-default-network-map" ]
  },
  "ordinal-routing-cost-map" : {
    "uri" : "http://custom.alto.example.com/ord/routingcost",
    "media-type" : "application/alto-costmap+json",
    "capabilities" : {
      "cost-type-names" : [ "ord-routing" ]
    },
    "uses": [ "my-default-network-map" ]
  },
  "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": [ "my-default-network-map" ]
  }
}

```

Note that the subdomain does not define Network Map, and uses the Network Map with Resource ID "my-default-network-map" defined in the Root IRD.

9.2.5. Considerations of Using IRD

9.2.5.1. 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 9.2.3, 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."

9.2.5.2. 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".

10. Protocol Specification: Basic Data Types

This section details the format of basic data types.

10.1. 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), the underline ('_', code point 0x5F), or the '.' separator (code point 0x2E). The '.' separator is reserved for future use and MUST NOT be used unless specifically indicated in this document, or an extension document.

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

10.2. Resource ID

A Resource ID uniquely identifies an particular resource (e.g., a Network Map) within an ALTO Server (see Section 9.2).

A Resource ID is encoded as a US-ASCII string with the same format as that of PIDName.

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

10.3. Version Tag

A Version Tag is defined as:

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

```
} VersionTag;
```

The 'resource-id' attribute is the Resource ID of a resource (e.g., a Network Map) defined in the Information Resource Directory, and 'tag' is 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 'resource-id' attributes are byte-for-byte equal and the 'tag' attributes are byte-for-byte equal.

10.4. 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.

10.4.1. 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 14.4).

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

10.4.2. 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.

10.4.2.1. IPv4

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

10.4.2.2. IPv6

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

10.4.2.3. 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.

10.4.3. 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.

10.4.3.1. IPv4

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

10.4.3.2. IPv6

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

10.4.4. 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"  
  ]  
}
```

10.5. 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.

10.6. 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), the underline ('_', code point 0x5F), 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 that appear in an HTTP request or response with an 'application/alto-*' media type and indicate Cost Metrics MUST be registered in the ALTO Cost Metrics registry Section 14.2.

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

10.7. 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.

10.8. Endpoint Property

We distinguish two types of Endpoint Properties: Resource Specific Endpoint Properties and Global Endpoint Properties. The type 'EndpointPropertyType' is used in this document to indicate a US-ASCII string denoting either a Resource Specific Endpoint Property or a Global Endpoint Property.

10.8.1. Resource Specific Endpoint Properties

We define only one Resource Specific Endpoint Property in this document: pid. It has the following format: a Resource ID, followed by the '.' separator (0x2E), followed by "pid". An example is "my-default-networkmap.pid".

10.8.2. Global Endpoint Properties

An Global 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 underline ('_', code point 0x5F). Note that the '.' separator is not allowed so that there is no ambiguity on whether an endpoint property is global or resource specific.

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 for Endpoint Properties appearing in an HTTP request or response with an 'application/alto-*' media type **MUST** be registered in the ALTO Endpoint Property registry Section 14.3.

11. Protocol Specification: Service Information Resources

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

11.1. Meta Information

For the "meta" field of the response to an individual Information Resource, we define two generic keys: "vtag", which is the Version Tag of the current Information Resource; and "dependent-vtags", which is an array of Version Tags, to indicate the Version Tags of the resources that this resource depends on.

11.2. Map Service

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

11.2.1. Network Map

A Network Map Information Resource defines a set of PIDs, and for each PID, lists the network locations (endpoints) within the PID. An ALTO Server **MUST** provide at least one Network Map.

11.2.1.1. Media Type

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

11.2.1.2. HTTP Method

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

11.2.1.3. Accept Input Parameters

None.

11.2.1.4. Capabilities

None.

11.2.1.5. Uses

None.

11.2.1.6. Response

The "meta" field of a Network Map response MUST include "vtag", which is the Version Tag of the retrieved Network Map.

The data component of a Network Map response is named "network-map", which is a JSON object of type NetworkMapData:

```
object {  
  NetworkMapData network-map;  
} InfoResourceNetworkMap : ResponseEntityBase;  
  
object-map {  
  PIDName -> EndpointAddrGroup;  
} NetworkMapData;
```

Specifically, a NetworkMapData object is a dictionary map keyed by PIDs, and each value representing the associated set of endpoint addresses of a PID.

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

11.2.1.7. 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" : {
    "vtag" : [
      { "resource-id": "my-default-network-map",
        "tag": "1266506139"
      }
    ]
  },
  "network-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"
      ]
    }
  }
}
```

Note that the encoding of a Network Map response was chosen for readability and compactness. If lookup efficiency at runtime is

crucial, then the returned Network Map can be transformed into data structures offering more efficient lookup. For example, one may store the Network Map as a trie-based data structure, which may allow efficient longest-prefix matching of IP addresses.

11.2.2. Cost Map

A 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.

11.2.2.1. Media Type

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

11.2.2.2. HTTP Method

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

11.2.2.3. Accept Input Parameters

None.

11.2.2.4. 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..1>;  
} CostMapCapabilities;
```

with field:

cost-type-names Note that the array **MUST** include a single CostType name defined by key "cost-types" in "meta" of the IRD. This is because an unfiltered Cost Map (accept == "") is requested via an HTTP GET that accepts no input parameters. As a contrast, for filtered cost maps (see Section 11.3.2), the array can have multiple elements.

11.2.2.5. Uses

The Resource ID of the Network Map based on which the Cost Map will be defined. Recall (Section 6) that the combination of a Network Map and a CostType defines a key. In other words, an ALTO Server MUST NOT define two Cost Maps with the same Cost Type, Network Map pair.

11.2.2.6. Response

The "meta" field of a Cost Map response MUST include the "dependent-vtags" key, whose value is a single-element array to indicate the Version Tag of the Network Map used, where the Network Map is specified in "uses" of the IRD. The "meta" MUST also include "cost-type", to indicate the Cost Type (Section 10.7) of the Cost Map.

The data component of a Cost Map response is named "cost-map", which is a JSON object of type CostMapData:

```
object {  
  CostMapData cost-map;  
} InfoResourceCostMap : ResponseEntityBase;  
  
object-map {  
  PIDName -> DstCosts;  
} CostMapData;  
  
object-map {  
  PIDName -> JSONValue;  
} DstCosts;
```

Specifically, a CostMapData object 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).

Similar to Network Map, the encoding of Cost Map was chosen for readability and compactness. If lookup efficiency at runtime is crucial, then the returned Cost Map can be transformed into data structures offering more efficient lookup. For example, one may store a Cost Map as a matrix.

11.2.2.7. 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" : {
    "dependent-vtags" : [
      { "resource-id": "my-default-network-map",
        "tag": "1266506139"
      }
    ],
    "cost-type" : { "cost-mode" : "numerical",
                   "cost-metric": "routingcost"
                 }
  },
  "cost-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.

11.3. Map Filtering Service

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

11.3.1. Filtered Network Map

A Filtered Network Map is a Network Map Information Resource (Section 11.2.1) for which an ALTO Client may supply a list of PIDs to be included. A Filtered Network Map MAY be provided by an ALTO Server.

11.3.1.1. Media Type

Since a Filtered Network Map is still a Network Map, it uses the media type defined for Network Map at Section 11.2.1.1.

11.3.1.2. HTTP Method

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

11.3.1.3. 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 fields:

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" field 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.

11.3.1.4. Capabilities

None.

11.3.1.5. Uses

The Resource ID of the Network Map based on which the filtering is performed.

11.3.1.6. Response

The format is the same as unfiltered Network Map. See Section 11.2.1.6 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 "vtag" of 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.

11.3.1.7. Example

```
POST /networkmap/filtered HTTP/1.1
Host: custom.alto.example.com
Content-Length: TBA
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" : {
    "vtag" : [
      { "resource-id": "my-default-network-map",
        "tag": "1266506139"
      }
    ]
  },
  "network-map" : {
    "PID1" : {
      "ipv4" : [
        "192.0.2.0/24",
        "198.51.100.0/24"
      ]
    },
    "PID2" : {
      "ipv4": [
        "198.51.100.128/24"
      ]
    }
  }
}
```

11.3.2. Filtered Cost Map

A Filtered Cost Map is a Cost Map Information Resource (Section 11.2.2) for which an ALTO Client may supply additional parameters limiting the scope of the resulting Cost Map. A Filtered Cost Map MAY be provided by an ALTO Server.

11.3.2.1. Media Type

Since a Filtered Cost Map is still a Cost Map, it uses the media type defined for Cost Map at Section 11.2.2.1.

11.3.2.2. HTTP Method

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

11.3.2.3. 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 fields:

cost-type The CostType (Section 10.7) 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 11.3.2.4). 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 11.3.2.4) 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" field is not present, both lists MUST be interpreted by the ALTO Server as containing the full set of currently-defined PIDs.

11.3.2.4. Capabilities

The URI providing this resource supports all capabilities documented in Section 11.2.2.4 (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 fields:

cost-type-names See Section 11.2.2.4 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 field 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.

11.3.2.5. Uses

The Resource ID of the Network Map based on which the Cost Map will be filtered.

11.3.2.6. Response

The format is the same as an unfiltered Cost Map. See Section 11.2.2.6 for the format.

The "dependent-vtags" key in the "meta" field specifies a single element, which is the Version Tag of the Network Map used in filtering. 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.

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.

11.3.2.7. 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" : {
    "dependent-vtags" : [
      { "resource-id": "my-default-network-map",
        "tag": "1266506139"
      }
    ],
    "cost-type": { "cost-mode" : "numerical",
                  "cost-metric" : "routingcost"
                }
  },
  "cost-map" : {
    "PID1": { "PID1": 0, "PID2": 1, "PID3": 2 }
  }
}
```

11.4. Endpoint Property Service

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

11.4.1. Endpoint Property

An Endpoint Property resource provides information about properties for individual endpoints. It MAY be provided by an ALTO Server.

11.4.1.1. Media Type

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

11.4.1.2. HTTP Method

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

11.4.1.3. 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 fields:

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 11.4.1.4). The ALTO Server MUST interpret entries appearing multiple times as if they appeared only once.

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

11.4.1.4. Capabilities

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

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

with field:

prop-types The Endpoint Properties (see Section 10.8) supported by the corresponding URI.

In particular, the Information Resource Closure MUST provide the look up of pid for every Network Map defined.

11.4.1.5. Uses

None.

11.4.1.6. Response

The "dependent-vtags" key in the "meta" field of the response MUST include the Version Tags of all Network Maps whose 'pid' is queried.

The data component of an Endpoint Properties response is named "endpoint-properties", which is a JSON object of type EndpointPropertyMapData, where:

```
object {  
  EndpointPropertyMapData endpoint-properties;  
} InfoResourceEndpointProperties : ResponseEntityBase;
```

```
object-map {  
  TypedEndpointAddr -> EndpointProps;  
} EndpointPropertyMapData;
```

```
object {  
  EndpointPropertyType -> JSONValue;  
} EndpointProps;
```

Specifically, an EndpointPropertyMapData object 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.

11.4.1.7. Example

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

```
{
  "properties" : [ "my-default-networkmap.pid",
                  "priv:ietf-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" : {
    "dependent-vtags" : [
      { "resource-id": "my-default-network-map",
        "tag": "1266506139"
      }
    ]
  },
  "endpoint-properties": {
    "ipv4:192.0.2.34" : { "my-default-network-map.pid": "PID1",
                          "priv:ietf-example-prop": "1" },
    "ipv4:203.0.113.129" : { "my-default-network-map.pid": "PID3" }
  }
}
```

11.5. Endpoint Cost Service

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

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

11.5.1. Endpoint Cost

An 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 15.3 for additional details.

11.5.1.1. Media Type

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

11.5.1.2. HTTP Method

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

11.5.1.3. 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 fields:

cost-type The Cost Type (Section 10.7) 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 11.5.1.4). 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 11.3.2).

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 13.3 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.

11.5.1.4. Capabilities

In this document, we define `EndpointCostCapabilities` the same as `FilteredCostMapCapabilities`. See Section 11.3.2.4.

11.5.1.5. Uses

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 implementation of an ECS resource may compute the cost between two endpoints as the cost between the PIDs corresponding to the endpoints, using one of the exposed network and cost maps defined by the server. However, to preserve flexibility, the ECS resource MAY omit declaring in the "uses" attribute the network map and/or cost map on which it depends.

11.5.1.6. Response

The "meta" field of an Endpoint Cost response MUST include the "cost-type" key, to indicate the Cost Type used.

The data component of an Endpoint Cost response is named "endpoint-cost-map", which is a JSON object of type `EndpointCostMapData`:

```
object {  
  EndpointCostMapData endpoint-cost-map;  
} InfoResourceEndpointCostMap : ResponseEntityBase;  
  
object-map {  
  TypedEndpointAddr -> EndpointDstCosts;  
} EndpointCostMapData;  
  
object-map {  
  TypedEndpointAddr -> JSONValue;  
} EndpointDstCosts;
```

Specifically, an `EndpointCostMapData` object is a dictionary map with each key representing a `TypedEndpointAddr` string identifying the Source Endpoint specified in the input parameters. 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.

11.5.1.7. Example

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: TBA
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: TBA
Content-Type: application/alto-endpointcost+json
```

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

12. 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].

12.1. 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.

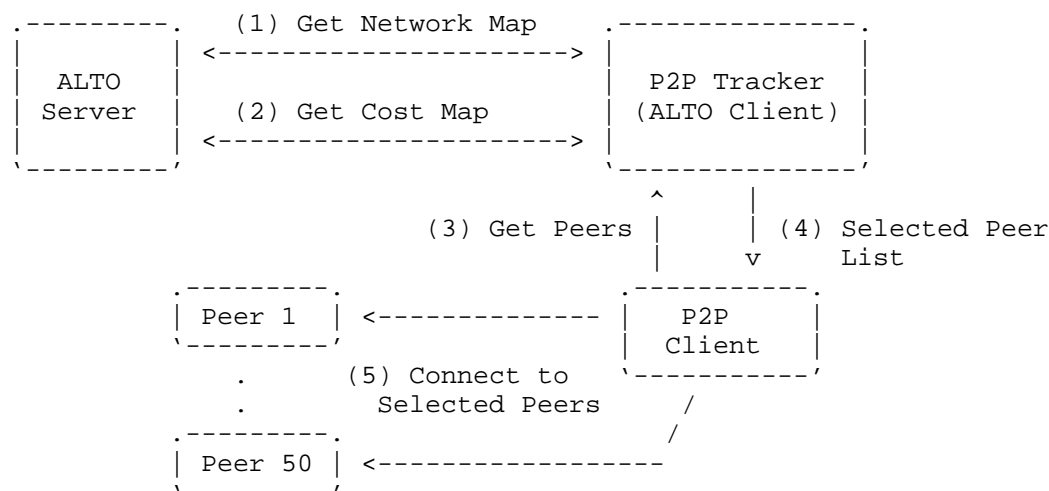


Figure 4: ALTO Client Embedded in P2P Tracker

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.

12.2. 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.

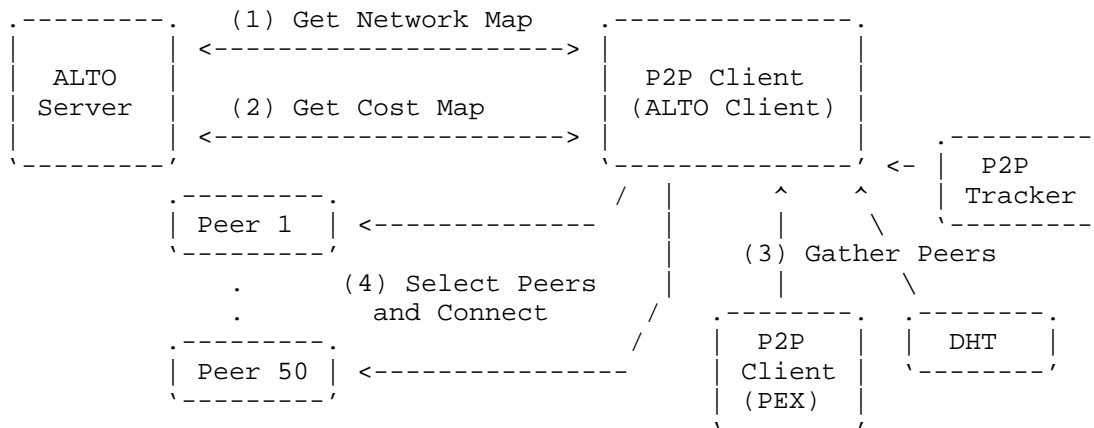


Figure 5: ALTO Client Embedded in P2P Client

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.

12.3. 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.

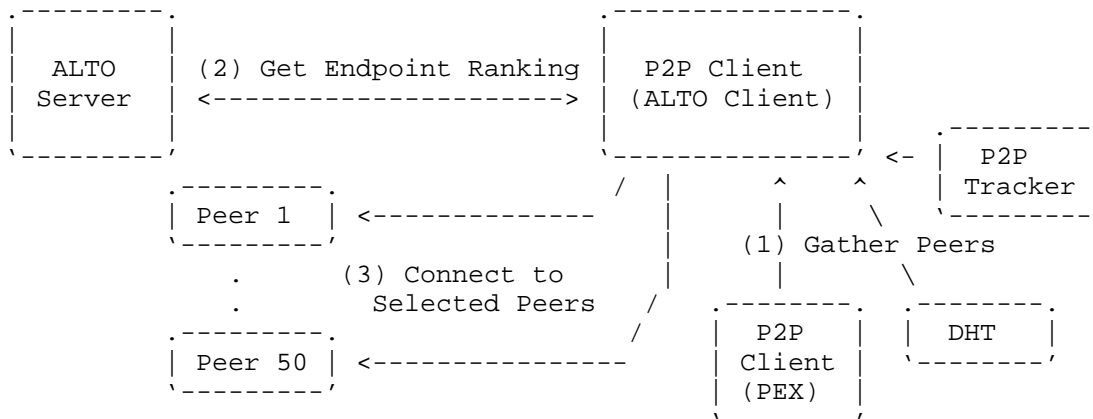


Figure 6: ALTO Client Embedded in P2P Client: Ranking

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.

13. Discussions

13.1. 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 9.2) to locate an Information Resource with the desired ALTO Information.

13.2. 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.

13.3. 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.

13.4. 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.

14. IANA Considerations

14.1. 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 9.2
application	alto-networkmap+json	Section 11.2.1
application	alto-networkmapfilter+json	Section 11.3.1
application	alto-costmap+json	Section 11.2.2
application	alto-costmapfilter+json	Section 11.3.2
application	alto-endpointprop+json	Section 11.4.1
application	alto-endpointpropparams+json	Section 11.4.1
application	alto-endpointcost+json	Section 11.5.1
application	alto-endpointcostparams+json	Section 11.5.1
application	alto-error+json	Section 8.5

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 15.

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).

14.2. 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 10.6. 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 the this Cost Metric are documented in Section 6.1.1.1, and security considerations are documented in Section 15.3.

14.3. 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.

14.4. 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 10.4.2	See Section 10.4.3	Direct mapping to IPv4
ipv6	See Section 10.4.2	See Section 10.4.3	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 10.4.1. 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 10.4.2).
- o Endpoint Prefix Encoding: The procedure for encoding a set of addresses of the registered type as an EndpointPrefix (see Section 10.4.3). 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.

14.5. ALTO Error Code Registry

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

15. 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.

15.1. Authenticity and Integrity of ALTO Information

15.1.1. 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.

15.1.2. 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.

15.1.3. 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.

15.2. Potential Undesirable Guidance from Authenticated ALTO Information

15.2.1. 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.

15.2.2. 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.

15.3. Confidentiality of ALTO Information

15.3.1. 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 inhabitants 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.

15.3.2. 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.

15.3.3. 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.

15.4. Privacy for ALTO Users

15.4.1. 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.

15.4.2. 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.

15.5. Availability of ALTO Service

15.5.1. 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).

15.5.2. 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.

16. 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.

16.1. Operations

16.1.1.1. 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.

16.1.1.2. 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.

16.1.1.3. 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.

16.1.4. 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 15.3. 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).

16.2. Management

16.2.1. 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].

16.2.2. 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.

16.2.3. 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.

16.2.4. 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.

16.2.5. 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)

16.2.6. Security Management

Section 15 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.

17. References

17.1. 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., Stiernerling, M., Woundy, R., and Y. Yang, "Application-Layer Traffic Optimization (ALTO) Requirements", RFC 6708, September 2012.

17.2. 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., Previdi, S., and M. Scharf, "ALTO Deployment Considerations", draft-ietf-alto-deployments-07 (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-10 (work in progress), September 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-24 (work in progress), September 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), Michael Scharf and Sabine Randriamasy (Alcatel-Lucent) gave substantial feedback and suggestions on the protocol design. We are particularly grateful to the substantial contributions of Wendy Roome (Alcatel-Lucent).

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

ALTO Working Group

Internet Draft
Intended status: standard

Young Lee
Dhruv Dhody
Qin Wu
Huawei
Greg Bernstein
Grotto Networking
Tae Sang Choi
ETRI

October 21, 2013

ALTO Extensions to Support Application and Network Resource
Information Exchange for High Bandwidth Applications in TE networks

draft-lee-alto-app-net-info-exchange-04.txt

Status of this Memo

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

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

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

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

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

This Internet-Draft will expire on April 21, 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.

Abstract

This draft proposes ALTO information model and protocol extensions to support application and network resource information exchange for high bandwidth applications in partially controlled and controlled environments as part of the infrastructure to application information exposure (i2aex) initiative.

Table of Contents

1. Introduction.....	3
2. Problem Statement.....	5
3. ALTO Constraints Filtering Extension Model.....	8
3.1. ALTO Query from Application Stratum to Network Stratum....	8
3.2. ALTO Response from Network Stratum to Application Stratum	10
3.3. Information Model of ALTO Query from Application Stratum to Network Stratum.....	10
3.4. Information Model of ALTO Response from Network Stratum to Application Stratum.....	11
3.5. ALTO Protocol Extension for Constraints Filtering Mechanism.....	11
3.6. Multiple Service Class.....	13
3.6.1. Gold Service.....	13
3.6.2. Silver Service.....	15
3.6.3. Bronze Service.....	17
4. ALTO Protocol Extension for Graph Representation Mechanism....	19
5. Summary and Conclusion.....	19
6. Security Considerations.....	19
7. IANA Considerations.....	19
8. References.....	19
8.1. Informative References.....	19
Author's Addresses.....	21
Intellectual Property Statement.....	21
Disclaimer of Validity.....	22

1. Introduction

This draft proposes ALTO information model and protocol extensions to support application and network resource information exchange for high bandwidth applications in partially controlled and controlled environments as part of the infrastructure to application information exposure (i2aex) initiative. The Controlled and partially controlled ALTO environments referred to here are those where general access to a specific ALTO server may be restricted to a qualified list of clients.

This draft is build upon the previously introduced High Bandwidth Use Cases [HighBW] and assumes that the network type carrying high bandwidth is a Traffic-Engineered (TE) network. In [HighBW], we have discussed two generic use cases that motivate the usefulness of general interfaces for cross stratum optimization in the network core. In our first use case, network resource usage became significant due to the aggregation of many individually unique client demands. In the second use case where data centers are sending large amount of data with each other, bandwidth usage was already significant enough to warrant the use of traffic engineered "express lanes" (e.g., private line service). We introduce third use case where inter-CDN providers may want to expose controlled network resource usage information so that CDN applications (e.g., request routing server) can utilize such information when appropriate decisions (e.g., request routing redirection) are needed.

These use cases result in optimization problems that trade off computational versus network costs and constraints. Both featured use cases show the usefulness of an ALTO interface between the application and network strata in optimizing the networked applications.

In particular, this draft introduces: (i) enhanced constraints filtering extensions to the ALTO protocol to reduce extraneous information transfer and enhance information hiding from the network's perspective; (ii) constrained cost graph mechanism encoding that enables enhanced application traffic optimization that was introduced by [HighBW].

In controlled and partially controlled environments in which operators are willing to share additional network stratum resource information such as bandwidth constraints or additional cost types of topology (e.g., graph or summary), it can be useful to reduce the amount of information transferred from the ALTO server to the ALTO client.

In considering information exchange between the application stratum and the network stratum, especially from the network stratum to the application stratum, the degree of information details is one of the key concerns from the network providers' standpoint. On the one hand, the network information has to be useful to the application; on the other hand, the provided network information should hide details about the network. In order to achieve these desired goals, a simple enhancement to ALTO protocol would help significantly both in reducing/filtering the amount of information and in increasing the usefulness of the information from network to application.

Figure 1 shows ALTO Client-Server Architecture for Application-Network information Exchange. Figure 1 shows that ALTO Client in the application stratum can interface with ALTO Server in the network stratum. With this architecture, a simple ALTO query mechanism from application (via ALTO client) to network (via ALTO server) can be implemented. According to this architecture, ALTO Client is assumed to interact with the Application Orchestrator that has the knowledge of the end-user (i.e., source) application requirement, Data Center locations (i.e., destinations) and their resource information.

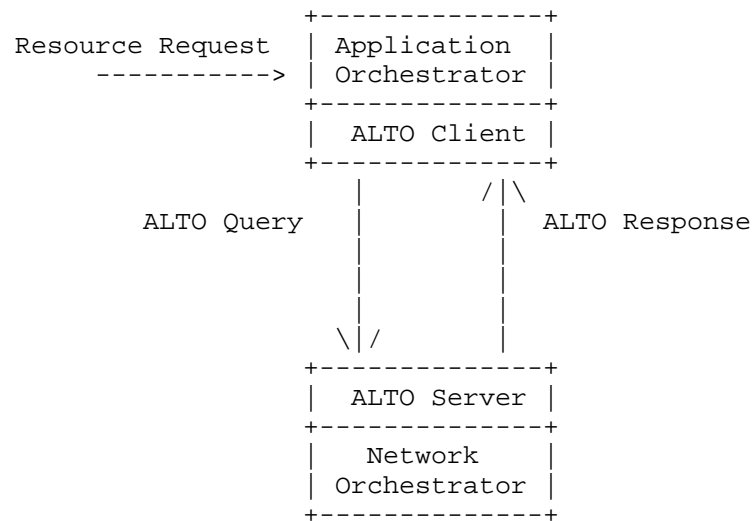


Figure 1 ALTO Client-Server Architecture for Application-Network information Exchange

The Application Orchestration functions depicted in Figure 1 interfacing data centers and end-users are out of the scope of this document. For simplicity purpose, Figure 1 doesn't depict the detailed relationship between ALTO client and server. In fact, both client and server don't need to be in the same administration boundary. It can be inter-operator and one to many relationships. For example, in the cases of inter-CDN environment or generic multi-domain environment, ALTO client represents a request routing server of upstream CDN operator and it interacts with multiple downstream CDN operators for their network resource information to make efficient decisions for desired quality CDN services. Interaction methods can either iterative or recursive. That is, ALTO client can interact with multiple ALTO servers directly or ALTO client can interact with one representative ALTO server and subsequent interaction is done by the representative one to rest of ALTO servers.

The organization of this document is as follows. Section 2 discusses the ALTO architecture in the context of the application and network strata interaction. Section 3 provides ALTO Information model and protocol extension to support ALTO Constraints Filtering Mechanism. Section 4 provides ALTO information model and the protocol extension to support ALTO Constrained Cost Graph Mechanism.

2. Problem Statement

One critical issue in Application-Network information exchange in ALTO is the amount of information exchanged between the application and network strata. The information provided by network providers can be not so useful to the application stratum unless such information is abstracted into an appropriate level the that application stratum can understand.

In partially controlled and controlled environments, network providers can furnish appropriately abstracted and pruned information to the application stratum with the cooperation of the application stratum that can indicate some level of filtering and pruning in its query.

To reduce extraneous information this draft allows for "filtering" (or "pruning") of the following information in query/response of the ALTO pull model:

- . Topology Filtering - reduction of the results to only those specified set of source(s) and destination(s) instead of the entire network cost map. Note that this mechanism is not new in the current ALTO protocol. In the context of application-network information exchange, this topology filtering can be

of a tremendous help in reducing the amount of data exchanged between application and network.

- . Multiple Service Class: ALTO server may provide multiple class of service (Gold, Silver, or Bronze) and allow application to request them accordingly.
- . Multiple Cost: Alto server should be able to provide multiple cost for a end to end path or abstract links in the graph.
- . Optimization Criteria: The optimization criteria that the ALTO server may use. For example, the criteria can be least number of hops, least amount of delay (latency), etc.
- . Constraint Filtering on paths or graphs (e.g., bandwidth, latency, hop count, packet loss, etc.) - reduction of results to only those that meet ALTO client specified cost bounds.

As discussed in [HighBW], in a controlled environment optimization is significantly enhanced by sharing data related to bandwidth constraints and additional cost measures [MultiCost], [TE-cost]. Such information may be considered sensitive to the network provider just as application data, e.g., usage, demand, etc., may be considered sensitive to an application provider. Section 3 provides ALTO information model and protocol extensions to support topology, multiple service class, constraints filtering mechanism.

Multiple Service Class (such as gold, silver and bronze services) MAY be supported by the ALTO server. These service classes could specify how the network is used (for ex exclusively reserved for the application, protection provided etc). The Application should further provision/reserve the network using some mechanisms which are out of scope of this document. Some example of services: _

. Gold Service

This service could be used to specify that an exact path meeting the application needs should be found. This path would be provisioned and resources reserved exclusively for its use. An example could be a private enterprise DC, which wish to offload to a public DC during peak load.

. Silver Service

This service could be used to get the path properties between User regions and DC. It could also specify some basic constraints that all of them should satisfy. These paths would be provisioned and resources maybe reserved. The Application may further assign end user request to a particular DC by using the network information of these paths. An example could be a gaming server geographically dispersed at multiple DC. The end-user (gamer) could be dynamically

assigned to the DC by looking at the past assignment, DC load and network properties.

. Bronze Service

This service could be used to specify that a simple best effort path should be found. This path would not be provisioned and resources will not be reserved. The service could still return the network information to the application which can use this information for DC selection by taking network information into consideration. An example could be a HD video service, which may use the network info to select video source for the end user.

While it is important to reduce and filter the information amount from network to application, for some applications that require stringent QoS objectives (e.g., bandwidth and latency), simple summary source-destination network resource information (i.e., summary form of topology) may not provide sufficient details to the application stratum. For example, suppose that a multiple number of large concurrent flows need to be scheduled from application to network. In such a case, a summary form of network topology that only shows source-destination bandwidth availability may not show the bottleneck links over which more than one flow may compete for the link bandwidth resource. This problem was indicated by [HighBW]. The following are the excerpts from [HighBW].

Consider the network shown in Figure 2, where DC indicates a datacenter, ER an end user region (as in the end user aggregation use case), N a switching node of some sort, and L a link. The link capacities and costs are also shown on the figure as well as a cost map between [ER1, ER2] and [DC1, DC2, DC3]. Since the network has a tree structure (very unusual but easier to draw in ASCII art), the cost map is unique.

As an illustration, assume that the maximum available capacity between any individual end region and a data center is 5 units(i.e., $L1=L2=L5=L6=5$). However, link L3 (capacity 8 units) represents a bottle neck to all the data centers (L3 is on all the paths to DC1, DC2, or DC3 from all end regions, ER1 and ER2).

ALTO Cost Map is shown in the lower right corner of Figure 2. This summary cost map does not provide enough details on the bottle necks. The lower left corner shows Link Capacity Cost, from which the bottle necks can be shown such that multi-flow commodity scheduling can be made possible to avoid such bottle necks.

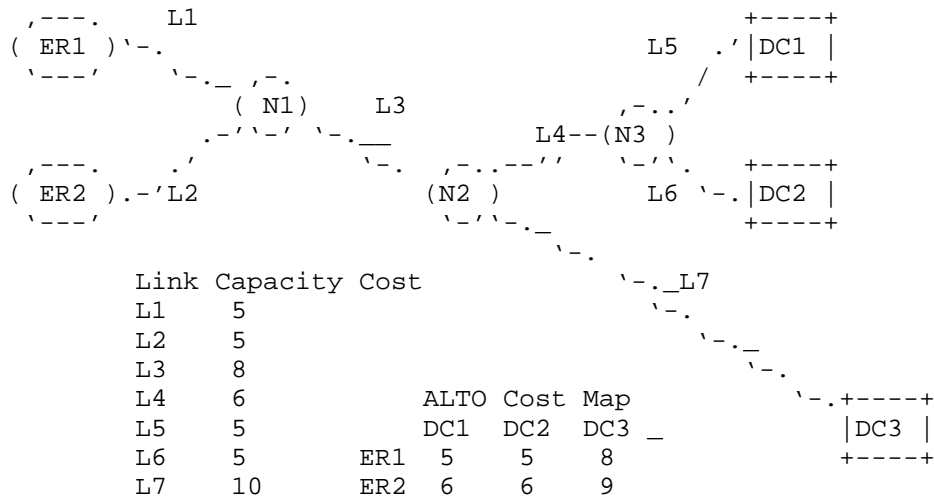


Figure 2. Example network illustrating bottlenecks

With the current ALTO cost map structure, the least cost path from ER1 would be either to DC1 or DC2. However, with the proposed capacitated cost map, the connection from ER1 to DC3 could be a better choice than the rest depending on the relative cost of network resources to data center resources.

A more general and relatively efficient alternative is to provide the requestor with a capacitated and multiply weighted graph that approximates and abstracts the capabilities of the network as seen by the source and destination location sets. This document provides ALTO information model and protocol extensions to support the graph model in Section 4.

3. ALTO Constraints Filtering Extension Model

3.1. ALTO Query from Application Stratum to Network Stratum

In order for the network stratum to provide its resource information, the application stratum needs to provide the End Point Cost Map to the network stratum. The End Point Cost Map should include the following information at a minimum:

- . End Point Source Address(es) /* these are the respective addresses of the nearest PE's to the user/client location */
- . End Point Destination Address(es) /* these are the respective addresses of the nearest PE's to a set of the candidate Data Center locations that can provide service to the user request. */

Note that how ALTO client derives the End Point Source/Destination addresses in terms of the nearest PE's is beyond the scope of this document.

- . Service-Class:= {gold, silver, bronze} /*the service class as described in this document*/
- . Cost Type:= 'routingcost' as defined by base specification. Additional cost (ex. latency, hopcount) are defined in [MultiCost] and [TE-cost].
- . Cost Mode :={summary, graph} /* the cost map can be either a summary form or a graph form */
 - o Cost Mode: summary

This cost mode is indicated by string 'summary'. This mode indicates that the returned costs contain end-to-end values which can be used by application stratum for better selection of resources.
 - o Cost Mode: graph

This cost mode is indicated by string 'graph' in which case an abstract topology is returned to the application.
- . Constraints /* a set of constraints that apply to the requested path summary or graph for filtering. For instance, constraints can be like bandwidth greater than 'x', latency less than 'y', hopcount less than 'z', packetloss less than 'a' etc. */
- . Objective-function (or Optimization Criteria): The summary or the graph should be computed based on optimizing which parameter - IGP cost, latency, residual bandwidth, etc. This is for future use.

3.2. ALTO Response from Network Stratum to Application Stratum

In response to the ALTO Query from the Application Stratum, the Network Stratum needs to provide the filtered Cost Map Data of the feasible path found. The Filtered End Cost Map Data should include the following information at a minimum:

- . The list of feasible Source-Destination pair and its Cost Type
- . For each feasible S-D pair, indicate the following as specified in Section 3.4:
 - o Service Class;
 - o Cost Mode;
 - o Cost Type;
 - o Endpoint Cost Map Data
- . Parameter Values /* indicate the actual values of each constraint requested */

Note that in case of Graph, each S-D pair is the source of the abstract link and the destination of the abstract link.

3.3. Information Model of ALTO Query from Application Stratum to Network Stratum

Alto query:

```
Object{
  TypedEndpointAddr  Src<1...*>; /*atleast one source*/
  TypedEndpointAddr  Dsts<2...*>; /*atleast two destinations*/
}EndpointList;

Object{
  ServiceClass        service-class;
  CostMode             cost-mode;
  CostType            cost-type;
  [JSONString         constraints<0...*>; ]
  [JSONString         ObjectiveFunction]
  EndpointList         endpoints;
}EndpointCostMapReq;
```

3.4. Information Model of ALTO Response from Network Stratum to Application Stratum

Alto response:

```
Object-map{
  JSONString      costparam;
} EndpointCostParam ;

Object-map{
  TypedEndpointAddr -> EndpointCostParam<1...*>;
} EndpointCosts ;

Object-map{
  TypedEndpointAddr -> EndpointCosts;
} EndpointCostMapData ;

Object{
  ServiceClass      service-class;
  CostMode          cost-mode;
  CostType          cost-type;
  [EndpointCostMapData  map;]
}EndpointCostMapRsp;
```

The Alto response consist of map (EndpointCostMapData) which is map containing the S-D pairs information. For each destination, its parameters (rank, cost etc) is included using EndpointCostParam.

3.5. ALTO Protocol Extension for Constraints Filtering Mechanism

This section provides the ALTO protocol extensions based on the information model discussed in Sections 3.3. and 3.4. The scenario provided in this section is that the ALTO client in the Application Stratum requests the summary cost map from the network with one source and three destinations.

In this particular example, the ALTO client requests the filtered summary of the network path subject to: bandwidth ≥ 20 , latency < 10 , hop count < 5 and packet loss < 0.03 .

The ALTO server provides the resulted network paths in summary.

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: [TODO]
Content-Type: application/alto-csoendpointcostparams+json
Accept: application/alto-csoendpointsummary+json,application/alto-
error+json
{
  "service-class" : "silver",
  "cost-mode" : "summary",
  "cost-type" : "routingcost",
  "constraints": ["availbw gt 20", "delay lt 10", "hopcount lt 5",
"pktloss lt 0.03"],
  "endpoints" : {
    "srcs": [ "ipv4:192.0.2.2" ],
    "dsts": [
      "ipv4:192.0.2.89",
      "ipv4:198.51.100.34",
      "ipv4:203.0.113.45"
    ]
  }
}
```

```
HTTP/1.1 200 OK
Content-Length: [TODO]
Content-Type: application/alto-csoendpointsummary+json
{
  "meta" : {},
  "data" : {
    "service-class" : "silver",
    "cost-mode" : "summary",
    "cost-type" : "routingcost",
    "map" : {
      "ipv4:192.0.2.2": {
        "ipv4:192.0.2.89" : [ "delay eq 5",
          "hopcount eq 8", "pktloss eq 0.01", cost eq
100" ],
        "ipv4:18.51.100.34" : [ "delay eq 9",
```

```
                                "hopcount eq 10", "pktloss eq 0.02", cost
eq 120" ],
    "ipv4:203.0.113.45" : [ "delay eq 40",
                                "hopcount eq 12", "pktloss eq 0.02", cost
eq 50" ]
    }
  }
}
```

3.6. Multiple Service Class

The examples of various class of service is as follows, note that these examples are for illustrative purpose only.

3.6.1. Gold Service

As an example of a Gold service, consider a customer (say an Enterprise Private DC) who pays Top-Dollar to setup network based on the actual demand. The Path (maybe a TE LSP) would not be used by any other customer / application giving guarantee of service and best QoE to the application. The ALTO request/response may be used first to get the network states and later the path may also be provisioned by some mechanism which is out of scope of this document.

In this example, the application may like to find out the ranking of the destinations (DC) from the network point of view. It may further set the filtering constraints for bandwidth (bw), delay etc. The ALTO server first filter the destination that do not meet the constraints, further it provides ranking information based on the requested costtype.

Alto Request:

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: [TODO]
Content-Type: application/alto-csoendpointcostparams+json
Accept: application/alto-
csoendpointsummary+json,application/alto-
error+json
{
  "service-class" : "gold",
```

```
    "cost-mode" : "summary",
    "cost-type" : "routingcost",
    "constraints": ["availbwgt 20", "delay lt 10",
                    "pktloss lt 0.03", "jitter lt 10", "hopcount
lt 5" ],
    "endpoints" : {
      "srcs": [ "ipv4:192.0.2.2" ],
      "dsts": [
        "ipv4:192.0.2.89",
        "ipv4:198.51.100.34",
        "ipv4:203.0.113.45"
      ]
    }
  }
```

ALTO server would factor in the filtering constraints and provide only the ranking information to the application.

Alto Response:

```
HTTP/1.1 200 OK
Content-Length: [TODO]
Content-Type: application/alto-csoendpointsummary+json
{
  "meta" : {},
  "data" : {
    "service-class" : "gold",
    "cost-mode" : "summary",
    "cost-type" : "routingcost",
    "map" : {
      "ipv4:192.0.2.2": {
        "ipv4:192.0.2.89" : [ "rank eq 3" ],
        "ipv4:198.51.100.34" : [ "rank eq 1" ],
        "ipv4:203.0.113.45" : [ "rank eq 2" ]
      }
    }
  }
}
```

Note that above is just an example, a gold service may also choose to get detailed end to end information or an abstract graph.

3.6.2. Silver Service

As an example of a Silver service, consider a customer (say a Online Gaming Company) which will pay flat subscription fees to connect end user-regions to the DC hosting the online gaming servers.. In this case during the setup phase a flat full mesh of paths are established between the User regions and the Data Centers.

The Application gaming load balancer would handle the gaming end user by allocating him to a particular DC (gaming server). The reserved resources during admin setup are allocated to multiple end user requests.

In this example, application may want to know the end to end properties of the path between the user-regions and the DC. It may further set the filtering constraints for bandwidth (bw), delay etc.

Alto Request:

POST /endpointcost/lookup HTTP/1.1

```
Host: alto.example.com
Content-Length: [TODO]
Content-Type: application/alto-csoendpointcostparams+json
Accept: application/alto-csoendpointsummary+json,application/alto-
error+json
{
  "service-class" : "silver",
  "cost-mode" : "summary",
  "cost-type" : "routingcost",
  "constraints": ["availbwgt 20", "delay lt 10",
                  "pktloss lt 0.03", "jitter lt 10", "hopcount
lt 5" ],
  "endpoints" : {
    "srcs": [
      "ipv4:192.0.2.2",
      "ipv4:192.0.2.10"
    ],
    "dsts": [
      "ipv4:192.0.2.89",
      "ipv4:198.51.100.34",
      "ipv4:203.0.113.45"
```



```
    ]
  }
}
```

ALTO server would factor in the filtering constraints and provide the end to end cost parameters to the application.

Alto Response:

```
HTTP/1.1 200 OK
Content-Length: [TODO]
Content-Type: application/alto-csoendpointsummary+json
{
  "meta" : {},
  "data" : {
    "service-class" : "silver",
    "cost-mode" : "summary",
    "cost-type" : "routingcost",
    "map" : {
      "ipv4:192.0.2.2": {
        "ipv4:192.0.2.89" : [ "delay eq 5", "jitter eq 5",
                             "pktloss eq 0.01", "hopcount eq 8",
"cost eq 100" ],
        "ipv4:198.51.100.34" : [ "delay eq 9", "jitter eq 3",
                             "pktloss eq 0.02", "hopcount eq 10",
"cost eq 500" ],
        "ipv4:203.0.113.45" : [ "delay eq 4", "jitter eq 4",
                             "pktloss eq 0.02", "hopcount eq 12",
"cost eq 200" ]
      }

      "ipv4:192.0.2.10": {
        "ipv4:192.0.2.89" : [ "delay eq 4", "jitter eq 4",
                             "pktloss eq 0.03", "hopcount eq 6",
"cost eq 300" ],
        "ipv4:203.0.113.45" : [ "delay eq 6", "jitter eq 6",
                             "pktloss eq 0.04", "hopcount eq 8",
"cost eq 400" ]
      }
    }
  }
}
```

Note that above is just an example, a silver service may also choose to get an abstract graph in response.

3.6.3. Bronze Service

As an example of a Bronze service, consider a customer (say a Video service) doesn't reserve resources but pays a small fee to get an abstract view of the network. Best effort service, use IP best effort path (instead of reserved paths used by gold, silver). The application (global load balancer) could get the network abstract topology and would further handle the end user request by allocating them to a particular DC or CDN.

In this example, application may rely on the basic IP best effort but would like to know the abstract topology that could be used by the application to find out bottleneck etc. Note that no constraints are passed in this example and graph is requested.

Alto Request:

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: [TODO]
Content-Type: application/alto-csoendpointcostparams+json
Accept: application/alto-
csoendpointsummary+json,application/alto-
error+json
{
  "service-class" : "bronze",
  "cost-mode" : "graph",
  "cost-type" : "routingcost",
  "endpoints" : {
    "srcs": [
      "ipv4:192.0.2.2",
      "ipv4:192.0.2.10"    ],
    "dsts": [
      "ipv4:192.0.2.89",
      "ipv4:198.51.100.34",
      "ipv4:203.0.113.45"
    ]
  }
}
```

ALTO server would prepare an abstract network graph based on the source(s) and destination(s). The graph may also include some internal (maybe abstract) nodes (ex 192.0.2.20 and 192.0.2.30).

Alto Response:

HTTP/1.1 200 OK

Content-Length: [TODO]

Content-Type: application/alto-csoendpointsummary+json

```
{
  "meta" : {},
  "data" : {
    "service-class" : "bronze",
    "cost-mode" : "graph",
    "cost-type" : "routingcost",
    "map": {
      "ipv4:192.0.2.2": {
        "ipv4:192.0.2.20" : [ "delay eq 9", "jitter eq 2",
                              "pktloss eq 0.04", "availbw eq 20",
"cost eq 100" ]
      }

      "ipv4:192.0.2.20": {
        "ipv4:192.0.2.89" : [ "delay eq 5", "jitter eq 2",
                              "pktloss eq 0.02", "availbw eq 30",
"cost eq 100" ],
        "ipv4:198.51.100.34" : [ "delay eq 3", "jitter eq 2",
                              "pktloss eq 0.01", "availbw eq 50",
"cost eq 400" ]
      }

      "ipv4:192.0.2.10": {
        "ipv4:192.0.2.30" : [ "delay eq 4", "jitter eq 2",
                              "pktloss eq 0.01", "availbw eq 60",
"cost eq 300" ]
      }

      "ipv4:192.0.2.30": {
        "ipv4:203.0.113.45" : [ "delay eq 2", "jitter eq 2",
                              "pktloss eq 0.03", "availbw eq 10",
"cost eq 200" ]
      }
    }
  }
}
```

}

Note that above is just an example, a bronze service may also choose to get end to end information instead of an abstract graph in response.

Note that the EndpointCostMapData can be used for both the Graph representation as well as the end to end path.

4. ALTO Protocol Extension for Graph Representation Mechanism

The encoding details for graph representation mechanism are shown in Section 3.6.3 where the use of graph in a Bronze service is described.

5. Summary and Conclusion

TBD

6. Security Considerations

TBD

7. IANA Considerations

TBD

8. Acknowledgements

The authors would like to thank Richard Yang and Sabine Randriamasy for many helpful comments that greatly improved the contents of this draft.

9. References

9.1. Informative References

[HighBW] G. Bernstein and Y. Lee, "Use Cases for High Bandwidth Query and Control of Core Networks," draft-bernstein-alto-large-bandwidth-cases, work in progress.

[MultiCost] S. Randriamasy, Ed., "Multi-Cost ALTO," draft-randriamasy-alto-multi-cost, work in progress.

[TE-cost] Q. Wu, et. al. "JSON Format Extensions for Traffic Engineering (TE) performance metrics in the ALTO Information Resource Directory, draft-wu-alto-json-te, work in progress.

Author's Addresses

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

Dhruv Dhody
Huawei Technologies, India
Email: dhruv.dhody@huawei.com

Qin Wu
Huawei Technologies, China
Email: bill.wu@huawei.com

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

Tae-Sang Choi
ETRI
161 Gajong-Dong, Yusong-Gu
Daejeon, Republic of Korea
Phone: (8242) 860-5628
Email: choits@etri.re.kr

Intellectual Property Statement

The IETF Trust takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in any IETF Document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights.

Copies of Intellectual Property disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or

the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement any standard or specification contained in an IETF Document. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

All IETF Documents and the information contained therein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgment

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

ALTO WG
Internet Draft
Intended status: Standards Track

W. Roome
Alcatel-Lucent Bell Labs
Y. Yang
Yale
October 21, 2013

Expires April 2014

PID Property Extension for ALTO Protocol
draft-roome-alto-pid-properties-00.txt

Abstract

This document extends the Application-Layer Traffic Optimization (ALTO) protocol [I-D.ietf-alto-protocol] by defining PID-based properties in much the same way that the original ALTO protocol defines endpoint-based properties.

Requirement 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 April 24, 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
 - 1.1. ALTO PIDs
 - 1.2. ALTO Endpoint Properties
 - 1.3. PID Properties
 - 1.4. Inheritance Via Nested PIDs
- 2. Services
 - 2.1. Full PID Property Map Service
 - 2.2. Filtered PID Property Map Service
 - 2.3. Extension to ALTO Endpoint Property Service
- 3. Security Considerations
- 4. IANA Considerations
- 5. References
 - 6.1. Normative References
 - 6.2. Informative References

1. Introduction

1.1. ALTO PIDs

The ALTO protocol defines a PID (Provider-defined Identifier) as a collection of endpoint addresses. Each PID has a name, and the PID's address set is defined by one or more endpoint address prefixes called CIDRs [RFC4632]. An ALTO server uses PIDs by providing one or more Network Maps, each of which is defined by a collection of PIDs.

PID specifications can overlap. For example, if PID1 is 10.0.0.0/8, and PID2 is 10.0.1.0/24, then all endpoints in PID2 are also in PID1. However, ALTO requires that an endpoint address be in one, and only one, PID. ALTO resolves this ambiguity by saying that if an endpoint address matches several CIRDs, the endpoint is in the PID with the CIDR with the longest prefix. We refer to this PID as the home PID of the endpoint. Thus, for the example, 10.0.1.5 is in PID2, and 10.0.2.6 is in PID1.

Although not required by the ALTO protocol, the hierarchical structure of the PIDs in a Network Map may reflect the logical structure of the network. In particular, although it is not required, the endpoints in a PID may be in the same geographical area.

1.2. ALTO Endpoint Properties

The ALTO protocol defines endpoint properties as a set of (name, value) pairs associated with each selected endpoint address. An ALTO server defines those properties, and the ALTO protocol allows a client to obtain those properties from a server.

1.3. PID Properties

This document proposes extending the property concept by allowing PIDs to have properties. This is useful when the endpoints in a given PID share common properties. Examples are "country code", "continent code", "ISP", "lat/long bounding box", "endpoint type" (server farm, end users, cell data connections, etc).

1.4. Implicit Inheritance Via Nested PIDs

In this document, we define PID properties to take advantage of the fact that PID definitions can overlap, or nest. That is, an ALTO server may define PID1, PID2 and PID3 such that all CIDRs defined in PID2 are also covered by the CIDRs in PID1; so are the CIDRs defined in PID3. Hence, we say that PID2 and PID3 can be considered "sub-PIDs" of PID1.

To avoid potential issues of "multi-inheritance", for example, when PID2 is also a "sub-PID" of PID4, we consider only the case that the derived inheritance forms a tree. In other words, for the example that PID2 is sub-PID of both PID1 and PID4, then either PID1 is a sub-PID of PID4 or vice versa. Hence, we can say uniquely the direct parent of a PID. Future ALTO extensions may consider explicit definitions of nesting, for example, by specifying that PID1 consists of PID2 and PID2, without implicit derivation.

With nesting, we define that PID2 and PID3 would inherit all properties of its ancestors, for example PID1, unless overridden in the sub-PIDs. For example, an ALTO server might define continent-level PIDs, as well as country-level or region-level sub-PIDs. If the ALTO server defines a "continent code" property for the continent-level PIDs, the country-level PIDs will automatically inherit that property. Such inheritance reduces information redundancy.

2. Services

In the interests of simplicity, we will give an overview of the proposed services, rather than detailed descriptions.

2.1. Full PID Property Map Service

Analogous to ALTO's Full Cost Map Service, a Full PID Map Service returns properties defined for all PIDs in a Network Map.

This is a GET request. The response message is similar to that of ALTO's Endpoint Property Service, but with PID names instead of endpoint addresses. The IRD entry for the service defines a "prop-

types" capability with the names of the properties that this service returns, and specifies a "uses" attribute for the Network Map defining the PIDs.

In the interests of limiting the response message size, the Full PID Property Map Service would NOT enumerate inherited property values. Thus if PID1 defines PROP1, and if PID2 is contained within PID1 and does not override the value for PROP1, then the response message gives a value for PROP1 in PID1, but not in PID2. In this case the client is expected to deduce the inheritance. That is feasible because the client has all information needed to do that.

2.2. Filtered PID Property Map Service

Analogous to ALTO's Filtered Cost Map Service, a Filtered PID Map Service returns a subset of the Full PID Property Map. The client specifies the desired property and PID names.

This is a POST request. The response message is the same as for the Full PID Property Map Service. The request message is similar to the request message for ALTO's Endpoint Property Service, except with PID names instead of endpoint addresses. The IRD entry for the service defines a "prop-types" capability with the names of the properties this service returns, and specifies a "uses" attribute for the Network Map defining the PIDs.

Unlike the Full Filtered PID Property Service, the Filtered PID Property Service would explicitly enumerate inherited property values. Thus if PID1 defines PROP1, and if PID2 is contained within PID1 and does not override the value for PROP1, then the response message includes PID1's value for PROP1 in PID2's properties. This is necessary because the Filtered PID Property Map response does not give the client enough information to deduce the inherited properties. For consistency, the Filtered PID Property Service would enumerate inherited properties for a PID even if the client also requested properties for all PIDs that containing that PID.

2.3. Potential Integration with ALTO Endpoint Property Service

When one considers inheritance and considers that each endpoint defines a leaf of the PID inheritance tree, with its direct parent being its home PID, then each endpoint will inherit the properties of its ancestor PIDs. We propose extending the current Endpoint Property Service (EPS) to allow EPS to use PID properties as a default. Specifically, if the IRD for an EPS "uses" a Network Map resource, then if that EPS does not define a property value for a given endpoint, but the PID containing that endpoint does define a value for that property, then the EPS will return the PID property. As with

the Filtered PID Property Map Service, sub-PIDs would inherit property values from higher-level PIDs.

3. Security Considerations

There are no security considerations relevant to this document.

4. IANA Considerations

No actions are required from IANA as result of the publication of this document.

5. References

5.1. Normative References

[I-D.ietf-alto-protocol] Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-20 (work in progress), October 2013.

[RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, August 2006.

5.2. Informative References

Authors' Addresses

Wendy Roome
Alcatel-Lucent Bell Labs
600 Mountain Ave, Rm 2B-234
Murray Hill, NJ 07974
USA
Email: w.roome@alcatel-lucent.com

Y. Richard Yang
Yale University
51 Prospect St
New Haven CT
USA
Email: yry@cs.yale.edu

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: January 16, 2014

M. Scharf, Ed.
V. Gurbani, Ed.
G. Soprovich
V. Hilt
Alcatel-Lucent
July 15, 2013

The Virtual Private Network (VPN) Service in ALTO: Use Cases,
Requirements and Extensions
draft-scharf-alto-vpn-service-01

Abstract

The Application-Layer Traffic Optimization (ALTO) protocol is designed to allow entities with knowledge about the network infrastructure to export such information to applications that need to choose one or more resources from a candidate set. This document provides motivation for using ALTO in a Virtual Private Network (VPN) environment. We discuss use cases, requirements, and possible extensions to the base ALTO protocol that will be needed to support VPN services.

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. Overview	3
2. Terminology	4
3. Encompassing example	4
3.1. A VPN scenario	4
3.2. Exemplary use of ALTO	6
4. Use cases	9
4.1. Use case 1: Application guidance in an L3VPN	10
4.2. Use case 2: Application guidance in an L2VPN	11
4.3. Use case 3: VPN guidance without addresses	12
4.4. Use case 4: Extending the VPN	12
4.5. Use case 5: Shrinking the VPN	13
4.6. Use case 6: VPN selection	14
5. Requirements and gap analysis	14
5.1. Requirements	14
5.2. Gap analysis	15
5.3. Differences from other proposed ALTO extensions	16
6. Security considerations	18
7. IANA considerations	18
8. References	18
8.1. Normative References	18
8.2. Informative References	18
Appendix A. Acknowledgements	19
Authors' Addresses	19

1. Overview

Virtual Private Network (VPN) technology is widely used in public and private networks to create groups of users that are separated from other users of the network and allows these users to communicate among them as if they were on a private network. According to [RFC4364], the generic term "Virtual Private Network" is used to refer to a specific set of sites as either an intranet or an extranet that have been configured to allow communication. A site is a member of at least one VPN and may be a member of many.

Service providers offer different types of VPNs. [RFC4026] distinguishes between Layer 2 VPN (L2VPN) and Layer 3 VPN (L3VPN) using different sub-types. Virtual Private LAN Service (VPLS) is an L2VPN provider service that emulates the full functionality of a traditional Local Area Network (LAN) [RFC4762]. A VPLS makes it possible to interconnect several LAN segments over a packet switched network.

Another solution is an L3VPN, which interconnects sets of hosts and routers based on Layer 3 addresses. In this context, a virtual private network is defined in [RFC4364] as follows:

Consider a set of "sites" that are attached to a common network that we call "the backbone". Now apply some policy to create a number of subsets of that set, and impose the following rule: two sites may have IP interconnectivity over that backbone only if at least one of these subsets contains them both.

These subsets are Virtual Private Networks (VPNs). Two sites have IP connectivity over the common backbone only if there is some VPN that contains them both. Two sites that have no VPN in common have no connectivity over that backbone.

VPNs can also include "pseudo L1/L2" connectivity, such as pseudowire emulation (PWE) carrying legacy TDM or ATM circuits for point to point connectivity. Further examples are integrated optical solutions delivering light paths or integrated optical and Ethernet transport. It is instructive to note that point-to-point VPN services of this type rarely carry VPN edge addresses within the network; e.g., packets are encapsulated and transported without any kind of address facing the customer drop side of the network.

A VPN may also include mechanisms to enhance the level of separation (e.g., by end-to-end encryption), but the discussion of such mechanisms is outside the scope of this document. In the following, the term "VPN" is used to refer to provider supplied virtual private networking.

The ALTO protocol [I-D.ietf-alto-protocol] is designed to provide network information (e.g., basic network location structure, preferences of network paths) with the goal of modifying network resource consumption patterns while maintaining or improving application performance. The most important use case is providing application guidance in the global Internet, so that applications do not have to perform excessive measurements on their own. For the very same reason, topology exposure is also very useful in VPNs. But the constraints for using ALTO in L3VPNs or L2VPNs differ from the public Internet. This document presents these use cases and discusses requirements and extensions to the base ALTO protocol that will be needed to realize the VPN Service in ALTO.

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].

3. Encompassing example

3.1. A VPN scenario

Below, we present an example for a VPN scenario that describes an environment for an ALTO VPN Service. This scenario is subsequently used to analyze specific use cases.

We consider the following: there are two distinct entities, one, the network service provider (NSP) who owns the network and offers a VPN to the second entity, the customer, who has premises in four different locations that shall be interconnected by that VPN. The sites could be office branches, data centers, etc. Throughout this document, we assume the following four sites:

- o Site 1

- Location name: SITE-CHICAGO

- Geography Degree: 41.85 N, 87.65 W

- o Site 2

- Location name: SITE-OTTAWA

- Geography Degree: 45.24 N, 75.43 W

- o Site 3

Location name: SITE-SANFRANCISCO

Geography Degree: 37.75 N, 122.28 W

- o Site 4

Location name: SITE-PARIS

Geography Degree: 48.86 N, 2.35 E

It is assumed that these sites are interconnected by a VPN that may be identified by the hypothetical name "vpn42". This document specifically considers two different VPN types for the interconnection:

- o L3VPN: The local area networks at each site will have a certain IP subnet ranges, for instance 10.0.1.0/24 at site 1, 10.0.2.0/24 at site 2, etc.
- o L2VPN: All sites form part for a flat sub-IP network, e.g. a logical Ethernet segment. Different to a local network, the network potentially interconnects geographically remote sites.

The VPN will not necessarily be static. The customer could possibly modify the VPN and add new VPN sites, e. g., to handle peak-load demand or to consolidate VPN sites to account for reduced traffic. The service provider could offer a Web portal or other Operation Support Systems (OSS) solutions that allow the customer to grow or consolidate the VPN. Details on how the customer can configure VPNs are outside the scope of this document.

Furthermore, we assume that the customer is running at least one application that can benefit from application-level traffic optimization, e.g., using application-internal routing mechanisms or placement functions. For instance, typical uses cases for VPN customers could be:

- o Enterprise application optimization: Enterprise customers often run distributed applications that exchange large amounts of data, e.g., for synchronization of replicated data bases. Both for placement of replicas as well as for the scheduling of transfers insight into network topology information could be useful.
- o Private cloud computing solution: An enterprise customer could run own data centers at the four sites. The cloud management system could want to understand the network costs between different sites

for intelligent routing and placement decisions of Virtual Machines (VMs) among the VPN sites.

- o Cloud-bursting: One or more VPN endpoints could be located in a public cloud. If an enterprise customer needs additional resources, they could be provided by a public cloud, which is accessed through the VPN. Network topology awareness would help to decide in which data center of the public cloud those resources should be allocated.

These examples focus on enterprise customers of NSPs, which are typical users of provider-supplied VPNs. Such VPN customers typically have no insight into the network topology that transports the VPN. For instance, the actual delay between two VPN sites may significantly depend on the routing in the NSP MPLS/IP network. If better-than-random decisions are required, applications have to rely on own measurements. An alternative would be guidance by an ALTO server offered by the NSP.

It is important to emphasize that other scenarios and use cases exist and the examples enunciated so far are merely used to illustrate how ALTO can be used in a VPN context. A common characteristic of these use cases is that applications will not necessarily run in the public Internet, and they will typically not be accessed by residential customers. The internal use of ALTO by a specific application is not considered in this document.

3.2. Exemplary use of ALTO

In the example VPN described in the previous section, it would be beneficial if an ALTO server would expose cost maps or provide a ranking service that represents the costs between different sites, e.g., endpoints of the VPN. Similar to existing use cases of ALTO, this enables an application integrating an ALTO client to use this information for application-level traffic optimization. This results in the following scenario:

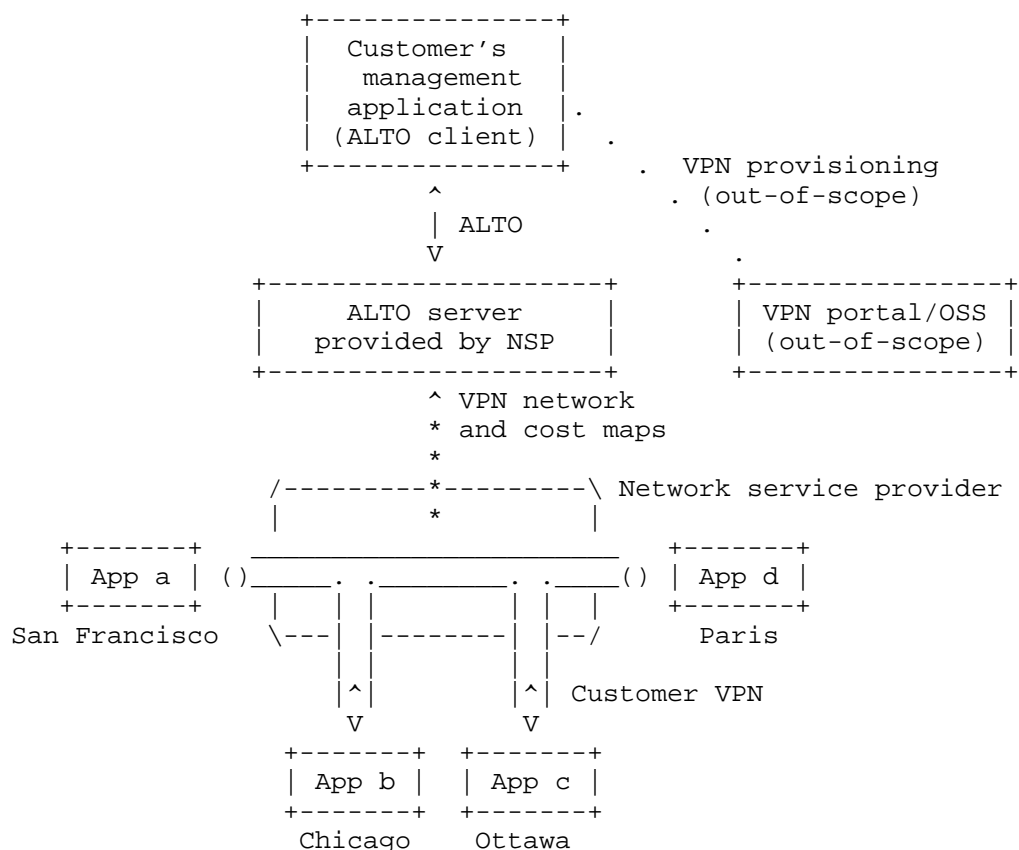


Figure 1: Overview of an ALTO usage scenario

The network service provider could operate an ALTO server. An ALTO client in an application could then retrieve an ALTO cost map by querying a corresponding URI, such as:

```
uri: http://alto.nsp.org/vpn42/costmap
```

The NSP can assign PIDs to each of the VPN endpoints; this renders computations at the ALTO server to fit in the current model of using the protocol. A corresponding example would be:

Site 1: PID "pid14"

Site 2: PID "pid21"

Site 3: PID "pid11"

Site 4: PID "pid27"

The example below further expands on the VPN by demonstrating the resulting network topology provided to an ALTO server. The picture corresponds to the VPN of the customer and also includes the Provider Edge (PE) routers and Customer Edge (CE) devices:

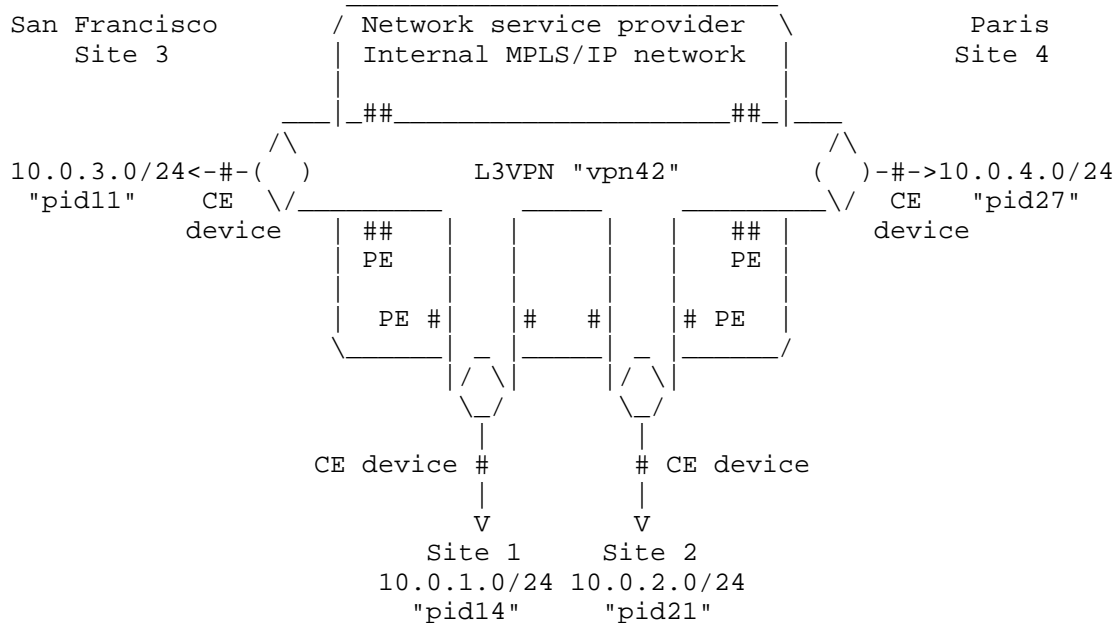


Figure 2: Example for mapping of VPN sites to ALTO PIDs

The costs exposed by the ALTO server can be based on routing costs inside the service provider network or other network topology information, such as delay measurements, traffic engineering (TE) data, etc. As with other use cases of ALTO, the costs can reflect the service provider's preference and policies regarding communication between the involved VPN endpoints.

Generally, two different types of applications can consume the information provided by the ALTO server. The first class can be composed of discrete application instances executing at the various sites that are interconnected by the VPN. ALTO is used to optimize the routing or resource consumption among those application instances. A typical examples is a distributed database, i.e., an enterprise backend system. In Figure 1, these application instances are referred to as "App a", "App b", "App c", and "App d". Generally speaking, this usage mirrors the canonical use of ALTO in

unstructured P2P networks or Content Delivery Networks (CDN) networks whereby a rendezvous is desired between a consumer and a plurality of producers. In this document, we label this class of applications by the term "user applications".

The second class represents management applications that typically work on VPN level. In addition to consulting an ALTO server provided by the NSP, this type of application possibly has its own understanding of what resources are available at different sites, and it could possibly even trigger more complex actions such as building out VPNs, e. g., by contacting a VPN portal of the NSP. In Figure 1, as well as the rest of this document, uses the term "management application" for this use case. An example would be an orchestration solution for cloud computing resources. It could use the topology and cost maps illustrated in Figure 2 to control VPN placement. In principle, management applications have some similarity to centralized resource directories in P2P networks (e. g., trackers), which are an important existing use case for ALTO. Yet, unlike resource directories in the Internet, a VPN typically interconnects mainly sites within one administrative domain.

There may also be an overlap between both types of applications. Furthermore, in particular for the first class of applications, the customer could run an own ALTO server, which could expose topology map and cost maps with further details only visible to the VPN customer (e.g., network segments behind NATs). Since such information is independent of the use of a VPN and typically not known to an NSP, these usage scenarios are not further detailed in this memo.

4. Use cases

Current VPNs provide no clear mechanism to convey information about the network infrastructure to management or user applications using that VPN, e.g. regarding preferences or topological properties of the network service provider network. Applications thus have to rely on other mechanisms such as local measurements to optimize their traffic. The ALTO protocol has been designed to overcome such limitations in the Internet. ALTO, being a well-established, generic, and flexible protocol, can be used in VPNs, too.

We now present various use cases that exhibit the utility of considering a VPN extension of ALTO. Through a series of use cases we demonstrate how a VPN customer and the NSP can use ALTO; we also highlight similarities and differences when using ALTO in the general Internet.

4.1. Use case 1: Application guidance in an L3VPN

The NSP providing the L3VPN service can offer an ALTO server that exposes network and cost information to applications running traffic over that VPN. Since an L3VPN is IP-based, this use case is in principle similar to the use cases already addressed by the ALTO base protocol.

Example 1: Consider the customer in Section 3.1 that has four VPN sites. A user application in one site (say Site 1) would now like to find out which of the other sites (Site 2 to Site 4) are topologically close to Site 1, perhaps to determine where to replicate a certain data set. A corresponding ALTO query would return the costs between those sites. The user application could then select a host in the corresponding subnetwork and connect to that endpoint.

Example 2: In addition to network proximity information, the user application could also be interested in guidance regarding network parameters that cannot be measured directly. For instance, a relevant parameter for a VPN site could be the level of redundancy for that VPN site, e.g., whether there is resilience by network protection schemes in the NSP network.

Example 3: It is quite common for VPN Customer Equipment (CE) to be multi-homed at the Provider Edge (PE). A CE may well home into to several PE routers and thus may have different network cost functions. For instance, assume that in Site 1 the CE will peer to a local PE1 and remote PE2. The cost to reach Site 2 in the VPN could be 1575 for PE1 and 2250 for PE2. The CE will thus choose to steer traffic from Site 1 to Site 2 toward PE1. While the realization of such traffic steering is outside the scope of this document, CE multi-homing places an explicit need to expose more than one set of network costs for a VPN endpoint.

In principle, the existing ALTO services such as network and cost map can provide such guidance. However, it is important to note that a VPN might not run in a public environment. The IP address ranges inside a VPN might not be globally unique or routable. Furthermore, a provider based VPN service normally maintains a strict separation between service provider addressing (such as addresses or Provider Edge routers) and customer addressing. As a result, an ALTO server will not expose the internal IP addressing of the network service provider, making it difficult to identify services using IP addresses in general. In a BGP L3VPN, the VPRN BGP Route Distinguisher could possibly be used as a service identifier, but it is unclear whether an application of a customer or the ALTO client will indeed know such network-internal information of the NSP and whether the NSP would

want to expose it. Also, it would make sense to define an ALTO VPN extension independent of a specific VPN technology.

The network costs in a VPN depends on VPN topology, which needs to be taken into consideration when calculating ALTO information. Given that VPNs are often offered by a single network service provider, ALTO cost information could include information that may be available for a single autonomous system, but difficult to gather in the Internet as a whole. Examples would be the provisioned bandwidth, network-internal latencies, or the path resilience. In a static VPN environment e.g. with a reserved resources in an MPLS/IP wide area network, these costs can be assumed to be rather stable and e. g. reflect the reserved bandwidth between VPN sites. For an application it is simpler and less intrusive to obtain such information about the VPN from the network instead of performing measurements, which would possibly require special probe instances at the different VPN sites (e. g., data centers). But as the encoding of such costs in ALTO is independent of the usage of a VPN, this document does not mandate any specific way how to build ALTO cost maps.

4.2. Use case 2: Application guidance in an L2VPN

The use case outlined in Example 1 also exists for L2VPNs, which are an important technology to transparently interconnect different LAN segments of enterprise users. Again, applications could benefit from getting insight into topological properties of the wide area network providing the L2VPN service, in order to avoid the overhead of own measurements.

Example 4: The user application described in Section 3.1 again wants to find out how well connected (topologically close) Site 1 is to Site 2, 3, or 4. Different from the previous example, all sites are now part of the same Layer 2 subnet. Another example for an application that would benefit from ALTO is a cloud management system. Such a management application could be interested in finding out whether migrating of a Virtual Machine from Site 1 to another site would improve performance, perhaps due to better connectivity or lower latency.

While this use case is in principle similar to the previous one, there is a major difference regarding addressing: Unlike the L3VPN, an L2VPN is not necessarily IP-based; it may use MAC addresses instead of IP addresses. While IP addresses can be aggregated easily and represented succinctly using CIDR notation, MAC addresses do not lend themselves to such aggregation and representation. Furthermore, MAC addresses are not useful to applications themselves. And finally, MAC addresses may not readily be known and available to an ALTO server of the network service provider. And even if they are,

an ALTO map using MAC addresses will be very large. In summary, use of MAC addresses is not scalable and nor does it denote any hierarchy that can be used for aggregation. Some other means of identifying services and hosts will be required when using ALTO in L2VPNs.

4.3. Use case 3: VPN guidance without addresses

The VPN interconnects different sites through the network service provider's network. An application might be interested in getting topology information among those sites without knowing actual addresses or identifiers used internally by the VPN. In fact, a VPN site may not even have an address known or visible to applications, e.g., a pseudo-wire VPN.

Example 5: A management application might ask for all VPN sites (i. e., corresponding PIDs) that have a delay less than 40ms or a routing cost less than 55, from VPN Site 1. A specific example for such an application might be cloud management system that uses application-level traffic optimization mechanisms. In the scenario introduced in Section 3.1, such an application may have a-priori information, learned from e.g. a VPN portal, about the VPN type and/or VPN identifiers ("vpn42") as well as some unique site identifier such as "SITE-CHICAGO" but no network addresses. The query could also be more complex or include constraints, e. g., limited to a particular TE class. Note that the ATLO protocol does not necessarily have to support the query constraint itself; if corresponding maps are available, the application can analyze the data itself.

Example 6: In absence of well-known existing network identifiers, a management application might want to query for VPN sites based on yet other attributes, such as geographical distance. For example, an application might want to find all the VPN sites (i. e., corresponding PIDs) within 50 KM of 45.35N, 75.92W. Such geographic queries would be typical of policies bounding delay by geographic distance or administrative and legal requirements.

Such application guidance is obviously similar to existing use of the ALTO cost map or ranking services except that the queries are not based on network addresses.

4.4. Use case 4: Extending the VPN

The customer can possibly grow the VPN to include new sites that are connected at a later time to the VPN. The actual mechanisms for VPN reconfiguration are outside the scope of this document.

Example 7: A management application could be interested in guidance for VPN sites that are currently not part of the VPN, but that would

be available e. g. to increase capacity or geographic coverage. Assume that two sites Site 1 and Site 2 are already connected to the VPN. Some time later, scale-out to a third site is required, and the application has to decide whether Site 3 or 4 is better suited for a new application instance. This is an realistic example for a cloud management system that is geographically distributed. Such a system would then have to decide whether Site 3 or Site 4 is topologically closer to the existing VPN endpoints, in order to determine the best location from the network point of view. An ALTO server could provide guidance on the offnet distance of Sites 3 and 4 to the existing VPN sites.

Apparently, the question whether to actually extend the VPN in a specific way may also include decisions outside the scope of ALTO, such as price information or other commercial or legal policies. The actual VPN re-configuration and attachment of a new site to the VPN topology requires back-office interaction and provisioning actions by separate, orthogonal mechanisms such as a Path Computation Element (PCE). Actual path setup by a PCE is independent from the selection of a suitable target site. But it makes sense to use the well-established ALTO methods in order to get at an early stage network proximity information as input information for the selection and configuration process. Applications typically cannot measure the network performance to destinations not already part of the VPN.

For a network service provider, customer guidance for VPN extension by ALTO offers a new possibility to optimize its internal traffic engineering. For instance, an operator could recommend to customers not to connect to a destination operating in protection mode, e.g., after a fiber cut, because in such a case the network may have less sparse resources. Note that a customer is not able to measure such constraints. ALTO is a simple interface to expose such information to applications.

From an ALTO perspective, growing VPN sites possibly results in different types of endpoints, some of which may exist a-priori but not be reachable within the VPN. They could possibly be understood as "shadow" PIDs that become active once the VPN is extended. Once the VPN is modified, new endpoints or PIDs may be created, i. e., the ALTO network and cost maps may have to be updated accordingly after the VPN is re-configured.

4.5. Use case 5: Shrinking the VPN

Much like a VPN may grow dynamically, it can also shrink when the resources in the VPN are underutilized. Instead of keeping the underutilized resources alive, the VPN operator may decide to consolidate the resources and remove sites from the VPN.

Example 8: Once again, consider the customer in Section 3.1 that has four VPN sites. Based on low resource demand, the management application may wonder whether Site 1 (Chicago) and Site 2 (Ottawa) can be consolidated, e. g., by moving resources between them. One important constraint for such a decision could network proximity information. After such a consolidation, the VPN network and cost maps will be updated to reflect the new topology.

From an ALTO server perspective, this use case is similar to a general application guidance. Yet, there could be a benefit for the service provider to provide special guidance regarding removal of VPN endpoints if there is a benefit for its internal traffic engineering (e. g., consolidation of network resources used by several VPN customers).

4.6. Use case 6: VPN selection

In a more advanced use case, ALTO could also be a selection function to choose VPNs based on network cost criteria.

Example 9: In a multi-homing environment, ALTO could be used to select one VPN out of several candidates to reach a certain destination, taking into account smaller costs, e. g., according to distance or to preferences of the network service provider network.

This use case differs from the previous examples since more than one VPN is involved, i. e., the ALTO guidance is not used to perform application-layer traffic optimization within one VPN, but instead across different VPNs.

5. Requirements and gap analysis

5.1. Requirements

Based on the scenarios listed in Section 4, several requirements can be derived for a VPN Service in ALTO:

REQ 1: The existing ALTO protocol and RESTful interface should be used as far as possible to enable an NSP to expose properties of a VPN.

REQ 2: A VPN Service must not require that network service provider expose internal addressing, such as internal addresses or loopback addresses of the Provider Edge (PE) routers.

REQ 3: A VPN Service must use the PID concept of the base ALTO protocol as far as possible, i. e., the VPNs and network entities in

the VPNs can be identified by PIDs. This permits use of the existing ALTO services such as the map service for VPNs, as well as the inherent topology abstraction provided by ALTO.

REQ 4: A VPN Service must be possible for different VPN types, i. e., it must not be limited to L3VPNs only.

REQ 5: The VPN Service must support use cases where IP addresses are not the only form of network identification.

REQ 6: If IP addresses are used, a VPN Service must not assume that IP address are globally routable or unique.

REQ 7: A VPN Service should include certain attributes that are unique to a VPN and that are not represented by the current set of attributes in the base ALTO protocol. Examples include location name of a site, geography coordinates (degree/digital), role, default policy, or geography restriction.

REQ 8: The PID must be selectable using standard ALTO filtering. A standard interface query should allow finding resources using, say, the location name attribute or the geography attributes.

REQ 9: The PID should be selectable using a filter that computes matching sites within a certain distance of a particular geographic coordinate based on latitude and longitude, in case that no other address information is known in advance.

REQ 10: Incremental build out (as well as the shrinking) of resources that are part of the VPN must be supported, i. e., the ALTO VPN service should also be able to expose information about new sites to be attached to the VPN, or provide guidance for removal of sites.

REQ 11: Information about a VPN must only be exposed to authorized users of that VPN.

5.2. Gap analysis

In the following we analyze to which extent the requirements of a VPN Service can be met by the existing ALTO protocol.

REQ 1: This is an inherent, general requirement for any new use or extension of ALTO.

REQ 2: This requirement can be supported in ALTO today, because it is left to the service provider which information to expose e.g. in ALTO cost maps.

REQ 3: The PID concept itself is generic and thus can fulfill this requirement.

REQ 4: L3VPNs are rather similar to existing use cases of ALTO in the Internet. Insofar as L2VPNs or pseudo-wire VPNs have the notion of some address, ALTO seems to be able to handle these through an extension that extends the definition of an address to include other identifiers besides IP addresses.

REQ 5: Use of ALTO with network identifiers that are not IP addresses requires work. There is a need to analyze how to name VPNs and endpoints and how to achieve a mapping to the information stored in the ALTO server.

REQ 6: ALTO can be used as of now with IP address ranges that are not globally routable. However, it must be emphasized that private VPN environments without uplink to the global Internet may only have connectivity to a limited number of IP subnets, i. e., the ALTO server will not be able to provide any reasonable guidance for most parts of the IP address space. Also, the ALTO server operator must take into account that IP address ranges in different VPNs may overlap, possibly also with the transport network infrastructure (e. g., PE routers).

REQ 7: Extensions to ALTO will be needed.

REQ 8: Assuming extensions in REQ 7, filtering should be fairly easy.

REQ 9: Extensions to ALTO will be needed, aligned with REQ 6.

REQ 10: This requirement will possibly require extensions to ALTO, e. g., to distinguish between endpoints that are already attached to the VPN and sites outside the VPN. Changes of the VPN topology are likely to change the ALTO maps, i.e., standard ALTO mechanism for incremental updates and push notifications would be of added value.

REQ 11: Existing authentication and access control mechanisms for ALTO could be sufficient to meet this requirement, subject to further analysis.

5.3. Differences from other proposed ALTO extensions

There have been various other proposals for ALTO extensions. In the following, we discuss why none of these extensions addresses the requirements of using ALTO in VPNs.

A use case of ALTO for traffic optimization in high bandwidth core networks is discussed in [I-D.bernstein-alto-large-bandwidth-cases].

It is proposed to enhance ALTO by bandwidth constraint representations, focusing on high-speed circuit switched optical networks that have a fixed capacity. However, L2VPNs or L3VPNs can be deployed in an MPLS/IP network without any bandwidth guarantees. An encoding of network parameters such as bandwidth in ALTO is therefore entirely orthogonal to the use of VPNs. The ALTO extensions suggested by [I-D.bernstein-alto-large-bandwidth-cases] are therefore not required by the use cases summarized in this document.

A related extension proposal [I-D.lee-alto-app-net-info-exchange] defines enhanced filtering constraints for ALTO, as well as a constrained cost graph encoding. The objective of the filtering is to retrieve paths or graphs for given constraints (e.g., bandwidth, latency, hop count, packet loss, etc.). This proposal basically enhances the way how ALTO can represent the costs in a network. However, the core challenge in VPNs is the addressing and lookup of VPN endpoints. With the VPN service, ALTO can be used in L2VPNs or L3VPNs with the existing encodings for cost maps, i. e., the extensions of [I-D.lee-alto-app-net-info-exchange] are orthogonal as well.

A general data center resource information model has been suggested in [I-D.lee-alto-ext-dc-resource]. According to that model, the ALTO server also includes data-center information not related at all to the network, such as compute resources, memory, power consumption, etc. This implies a significant extension of the scope of ALTO. While VPNs are an important technology to interconnect data centers, the ALTO VPN service solely focuses on networking cost, and ALTO extensions are limited to the minimum set of additional protocol features that are required in a VPN context. This memo does not argue that ALTO shall be used as a generic data center information exchange protocol.

[I-D.xie-alto-sdn-use-cases] presents an architecture how ALTO can be used if data-forwarding plane and control plane are separated. In such an architecture, ALTO could be used to exchange connectivity information between controllers in different domains. This proposal is entirely disjoint to the problem addressed by this document. Since the separation of data-forwarding and control plane is an internal network design issue, it does not matter for the ALTO VPN service how Network Service Provider control their infrastructure, and existing management solutions can be applied as well. Even though the realization of network control and management of VPNs is outside the scope of this document, we note that existing L2VPN and L3VPN solutions often integrate data-forwarding and control plane.

In summary, this document proposes a small and well-focused extension

to enable the use of ALTO in VPN environments, given that the current address types and information models of ALTO is not sufficient in some cases. This document does explicitly not suggest any non-networking or technology-specific ALTO extension.

6. Security considerations

TBD.

7. IANA considerations

TBD.

8. References

8.1. Normative 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.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC4026] Andersson, L. and T. Madsen, "Provider Provisioned Virtual
Private Network (VPN) Terminology", RFC 4026, March 2005.
- [RFC4364] Rosen, E. and Y. Rekhter, "BGP/MPLS IP Virtual Private
Networks (VPNs)", RFC 4364, February 2006.
- [RFC4762] Lasserre, M. and V. Kompella, "Virtual Private LAN Service
(VPLS) Using Label Distribution Protocol (LDP) Signaling",
RFC 4762, January 2007.

8.2. Informative References

- [I-D.bernstein-alto-large-bandwidth-cases]
Bernstein, G. and Y. Lee, "Use Cases for High Bandwidth
Query and Control of Core Networks",
draft-bernstein-alto-large-bandwidth-cases-02 (work in
progress), July 2012.
- [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.

[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.xie-alto-sdn-use-cases]

Xie, H., Tsou, T., Lopez, D., and H. Yin, "Use Cases for
ALTO with Software Defined Networks",
draft-xie-alto-sdn-use-cases-01 (work in progress),
June 2012.

Appendix A. Acknowledgements

TBD.

Authors' Addresses

Michael Scharf (editor)
Alcatel-Lucent

Email: Michael.Scharf@alcatel-lucent.com

Vijay K. Gurbani (editor)
Alcatel-Lucent

Email: vkg@bell-labs.com

Greg Soprovich
Alcatel-Lucent

Email: Greg.Soprovich@alcatel-lucent.com

Volker Hilt
Alcatel-Lucent

Email: volker.hilt@bell-labs.com

CDNI
Internet-Draft
Intended status: Informational
Expires: April 24, 2014

J. Seedorf
NEC
Y. Yang
Yale
October 21, 2013

CDNI Footprint and Capabilities Advertisement using ALTO
draft-seedorf-cdni-fci-alto-00

Abstract

Network Service Providers (NSPs) are currently considering to deploy Content Delivery Networks (CDNs) within their networks. As a consequence of this development, there is a need for interconnecting these local CDNs. The necessary interfaces for inter-connecting CDNs are currently being defined in the Content Delivery Networks Interconnection (CDNI) WG. This document focuses on the CDNI Footprint & Capabilities Advertisement interface (FCI). Specifically, this document outlines how the solutions currently being defined in the Application Layer Traffic Optimization (ALTO) WG can facilitate Footprint & Capabilities Advertisement in a CDNI context, i.e. how the CDNI FCI can be realised with the ALTO protocol. Concrete examples of how ALTO can be integrated within CDNI request routing and in particular in the process of selecting a downstream CDN are given. The examples in this document are based on the use cases and examples currently being discussed in the CDNI WG.

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 April 24, 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. ALTO within CDNI Request Routing	3
3. Assumptions and High-Level Design Considerations	4
3.1. General Assumptions and Consideration	4
3.2. Semantics for Footprint/Capabilities Advertisement	5
4. Selection of a Downstream CDN with ALTO	7
4.1. Footprint and Capabilities Advertisement using ALTO Network Map and PID Properties	7
4.2. Conveying additional information with ALTO Cost Maps	8
4.3. Example of Selecting a Downstream CDN based on ALTO Maps	9
4.4. Advantages of using ALTO	10
5. Useful ALTO extensions for CDNI Request Routing	10
6. Security Considerations	12
7. Summary and Outlook	12
8. Acknowledgements	12
9. References	13
9.1. Normative References	13
9.2. Informative References	13
Authors' Addresses	15

1. Introduction

Many Network Service Providers (NSPs) are currently considering or have already started to deploy Content Delivery Networks (CDNs) within their networks. As a consequence of this development, there is a need for interconnecting these local CDNs. Content Delivery Networks Interconnection (CDNI) has the goal of standardizing protocols to enable such interconnection of CDNs [RFC6707].

The CDNI problem statement [RFC6707] envisions four interfaces to be standardized within the IETF for CDN interconnection:

- o CDNI Request Routing Interface

- o CDNI Metadata Interface
- o CDNI Logging Interface
- o CDNI Control Interface

This document focuses solely on the CDNI Request Routing Interface, which can be further divided into two interfaces (see [RFC6707] for a detailed description): the CDNI Request Routing Redirection interface (RI), and the CDNI Footprint & Capabilities Advertisement interface (FCI). This document presents how one may use ALTO as a protocol for CDNI Footprint & Capabilities Advertisement. Concrete examples of how the CDNI FCI can be implemented with the ALTO protocol [I-D.ietf-alto-protocol] are given. The examples used in this document are based on the use cases and request routing proposals currently being discussed in the CDNI WG [RFC6770] [I-D.peterson-CDNI-strawman] and in the ALTO WG [I-D.jenkins-alto-cdn-use-cases].

A previous version of this document [I-D.seedorf-alto-for-cdni] contained detailed examples of actual request routing and surrogate selection with ALTO, i.e. how ALTO could be used for implementing the CDNI Request Routing Redirection interface (RI). This version solely focuses on implementing the CDNI Footprint & Capabilities Advertisement interface (FCI) with ALTO, i.e. the selection of a downstream CDN and how ALTO can support such downstream CDN selection.

Throughout this document, we use the terminology for CDNI defined in [I-D.ietf-cdni-problem-statement].

2. ALTO within CDNI Request Routing

The main purpose of the CDNI Request Routing Interface is described in [RFC6707] as follows: "The CDNI Request Routing interface enables a Request Routing function in an Upstream CDN to query a Request Routing function in a Downstream CDN to determine if the Downstream CDN is able (and willing) to accept the delegated Content Request. It also allows the Downstream CDN to control what should be returned to the User Agent in the redirection message by the upstream Request Routing function." On a high level, the scope of the CDNI Request Routing Interface therefore contains two main tasks:

- o A) Determining if the downstream CDN is willing to accept a delegated content request
- o B) Redirecting the content request coming from an upstream CDN to the proper entry point or entity in the downstream CDN

More precisely, in [I-D.ietf-cdni-framework] the request routing interface is broadly divided into two functionalities:

- o 1) the asynchronous advertisement of footprint and capabilities by a dCDN that allows a uCDN to decide whether to redirect particular user requests to that dCDN (the CDNI FCI)
- o 2) the synchronous operation of actually redirecting a user request (the CDNI RI)

Application Layer Traffic Optimization (ALTO) is an approach for guiding the resource provider selection process in distributed applications that can choose among several candidate resources providers to retrieve a given resource. By conveying network layer (topology) information, an ALTO server can provide important information to "guide" the resource provider selection process in distributed applications. Usually, it is assumed that an ALTO server conveys information these applications cannot measure themselves [RFC5693].

Originally, ALTO was motivated by the huge amount of cross-ISP traffic generated by P2P applications [RFC5693]. Recently, however, ALTO is also being considered for improving the request routing in CDNs [I-D.jenkins-alto-cdn-use-cases]. In this context, it has also been proposed to use ALTO for selecting an entry-point in a downstream NSP's network (see section 3.4 "CDN delivering Over-The-Top of a NSP's network" in [I-D.jenkins-alto-cdn-use-cases]). Also, the CDNI problem statement explicitly mentions ALTO as a candidate protocol for "algorithms for selection of CDN or Surrogate by Request-Routing systems" [I-D.ietf-cdni-problem-statement]. Yet, there have not been concrete proposals so far on how to use ALTO in the context of CDN interconnection. This document tries to close this gap by giving some examples on how ALTO could be used within CDNI request routing.

3. Assumptions and High-Level Design Considerations

In this section we list some assumptions and design issues to be considered when using ALTO for the CDNI Footprint and Capabilities Advertisement interface

3.1. General Assumptions and Consideration

Below we list some general assumptions and considerations:

- o As explicitly being out-of-scope for CDNI [I-D.ietf-cdni-problem-statement], the examples used in this document assume that ingestion of content or acquiring content

across CDNs is not part of request routing as considered within CDNI standardization work. The focus of using ALTO (as considered in this document) is hence on request routing only, assuming that the content (desired by the end user) is available in the downstream CDN (or can be acquired by the downstream CDN by some means).

- o Federation Model: "Footprint and Capabilities Advertisement" and in general CDN request routing depends on the federation model among the CDN providers. Designing a suitable solution thus depends on whether a solution is needed for different settings, where CDNs consist of both NSP CDNs (serving individual ASes) and general, traditional CDNs (such as Akamai). We assume that CDNI is not designed for a setting where only NSP CDNs each serve a single AS only.
- o In this document, we assume that the upstream CDN (uCDN) makes the decision on selecting a downstream CDN, based on information that each downstream CDN has made available to the upstream CDN. Further, we assume that in principle more than one dCDN may be suitable for a given end-user request (i.e. different dCDNs may claim "overlapping" footprints). The uCDN hence potentially has to select among several candidate downstream CDNs for a given end user request.
- o It is not clear what kind(s) of business, contract, and operational relationships two peering CDNs may form. For the Internet, we see provider-customer and peering as two main relations; providers may use different charging models (e.g., 95-percentile, total volume) and may provide different SLAs. Given such unknown characteristics of CDN peering business agreements, we should design the protocol to support as much diverse potential business and operational models as possible.

3.2. Semantics for Footprint/Capabilities Advertisement

The CDNI document on "Footprint and Capabilities Semantics" [I-D.spp-cdni-rr-foot-cap-semantics] defines the semantics for the CDNI FCI. It thus provides guidance on what Footprint and Capabilities mean in a CDNI context and how a protocol solution should in principle look like. Here we briefly summarize the key points of the semantics of Footprint and Capabilities (for a detailed discussion, the reader is referred to [I-D.spp-cdni-rr-foot-cap-semantics]):

- o Often, footprint and capabilities are tied together and cannot be interpreted independently from each other. In such cases, i.e. where capabilities must be expressed on a per footprint basis, it

may be beneficial to combine footprint and capabilities advertisement.

- o Given that a large part of Footprint and Capabilities Advertisement will actually happen in contractual agreements, the semantics of CDNI Footprint and Capabilities advertisement refer to answering the following question: what exactly still needs to be advertised by the CDNI FCI? For instance, updates about temporal failures of part of a footprint can be useful information to convey via the CDNI request routing interface. Such information would provide updates on information previously agreed in contracts between the participating CDNs. In other words, the CDNI FCI is a means for a dCDN to provide changes/updates regarding a footprint and/or capabilities it has prior agreed to serve in a contract with a uCDN.
- o It seems clear that "coverage/reachability" types of footprint must be supported within CDNI. The following such types of footprint are mandatory and must be supported by the CDNI FCI:

- * List of ISO Country Codes

- * List of AS numbers

- * Set of IP-prefixes

A 'set of IP-prefixes' must be able to contain full IP addresses, i.e., a /32 for IPv4 and a /128 for IPv6, and also IP prefixes with an arbitrary prefix length. There must also be support for multiple IP address versions, i.e., IPv4 and IPv6, in such a footprint.

- o For all of these mandatory-to-implement footprint types, footprints can be viewed as constraints for delegating requests to a dCDN: A dCDN footprint advertisement tells the uCDN the limitations for delegating a request to the dCDN. For IP prefixes or ASN(s), the footprint signals to the uCDN that it should consider the dCDN a candidate only if the IP address of the request routing source falls within the prefix set (or ASN, respectively). The CDNI specifications do not define how a given uCDN determines what address ranges are in a particular ASN. Similarly, for country codes a uCDN should only consider the dCDN a candidate if it covers the country of the request routing source. The CDNI specifications do not define how a given uCDN determines the country of the request routing source. Multiple footprint constraints are additive, i.e. the advertisement of different types of footprint narrows the dCDN candidacy cumulatively.

- o The following capabilities seem useful as 'base' capabilities, i.e. ones that are needed in any case and therefore constitute mandatory capabilities to be supported by the CDNI FCI:
 - * Delivery Protocol (e.g., HTTP vs. RTMP)
 - * Acquisition Protocol (for acquiring content from a uCDN)
 - * Redirection Mode (e.g., DNS Redirection vs. HTTP Redirection as discussed in [I-D.ietf-cdni-framework])
 - * Capabilities related to CDNI Logging (e.g., supported logging mechanisms)
 - * Capabilities related to CDNI Metadata (e.g., authorization algorithms or support for proprietary vendor metadata)

4. Selection of a Downstream CDN with ALTO

Under the considerations stated in Section 3, ALTO can help the upstream CDN provider to select a proper downstream CDN provider for a given end user request as follows: Each downstream CDN provider hosts an ALTO server which provides ALTO information (i.e. ALTO network maps and ALTO cost maps [I-D.ietf-alto-protocol]) to an ALTO client at the upstream CDN provider. Network maps provided by each of several candidate downstream CDNs can provide information to the upstream CDN provider about each dCDN's "coverage/reachability" as well as capabilities.

4.1. Footprint and Capabilities Advertisement using ALTO Network Map and PID Properties

Conceptually, the footprint and capabilities interface of a dCDN is easy to specify: It is a function that given an endhost, returns if the dCDN is willing to serve the endhost, and the capabilities available to that endhost (e.g., "delivery-protocol": ["HTTP", "RTMP"], "acquisition-protocol": ["HTTP"], "redirection-mode": ["HTTP-redirect"], "logging-mechanism": ["TBD"], and "meta-capabilities": []).

Specifying the preceding for each endhost can be redundant, and one may use PIDs defined in ALTO. Specifically, an ALTO network map contains a "set of Network Location groupings" [I-D.ietf-alto-protocol]. The groupings are defined in the form of so-called "PIDs". A PID is an identifier to group network location endpoints, e.g. IP-addresses in the form of prefixes (see section 4 in [I-D.ietf-alto-protocol] for details).

Applying the basic idea of ALTO PIDs to the preceding, abstract mapping specification, by aggregating endhosts with the same capabilities in the same PID, we obtain CDNI FCI using ALTO Network Maps as simply (1) a Network Map which defines a set of PIDs, and (2) a PID Property Map [draft-roome-alto-pid-properties] that defines the properties of each PID, where the properties define the capabilities.

With the preceding Network Map and PID Property Map, the upstream CDN provider can easily match a given end user request with the footprint and capabilities of the downstream CDN providers. Whenever the footprint and/or capabilities of a dCDN change, the ALTO server of the dCDN changes its data, and the uCDN can obtain the update through ALTO incremental updates. Future extensions to ALTO to add notifications can be integrated when they become available.

In particular, this document does not define how a dCDN aggregates the endhosts into PIDs, to allow flexibility in (anticipated) updates.

In this document, we define the following PID properties, which each must be a JSON array, to convey all mandatory capabilities (see Section 3.2):

- o delivery-protocol
- o acquisition-protocol
- o redirection-mode
- o login-mechanism
- o meta-capabilities

To complement the preceding capabilities mapping, we require that an uCDN has access to ALTO Network Map(s) that can map from an endhost to Country Code and AS Number. Such mapping may or may not be specific to CDNI but can be a general mapping. Specifically, the uCDN should have access to ALTO Network Map(s) with Properties include:

- o country-code
- o asn

4.2. Conveying additional information with ALTO Cost Maps

An ALTO cost map contains costs between defined groupings of a corresponding network map (i.e. costs between PIDs): "An ALTO Cost Map defines Path Costs pairwise amongst sets of source and destination Network Locations" [I-D.ietf-alto-protocol]. This concept enables the provider of a cost map to express (and quantify) preferences of a destination network location with respect to a given source network location.

In the context of CDNI, the ALTO cost map concept is an extensive tool to convey additional information about the footprint or capabilities of a downstream CDN. The cost map concept provides a means for a downstream CDN provider to convey numeric values associated with a PID, e.g. in order to convey metrics associated with a footprint or a capability. This may be useful for future, non-mandatory types of footprint or capabilities.

One way to use ALTO cost maps would have these maps of the type N-to-m, i.e. 'costs' are expressed for each of N end user source PIDs to m dCDN request router PIDs. Semantically, a source PID in a CDNI ALTO cost map is thus the end user location, whereas a destination PID is a (group of) request router(s) to which the uCDN redirects the end user request. Note that this perspective is driven by the CDNI request routing. An alternative way - seen from the perspective of content retrieval - would be to have a m-to-N cost map where the source is always the dCDN and the destination is the end user (with the semantic "if the source dCDN would deliver content to an end user in the destination PID, the costs would be the following). With explicit destination PIDs reflecting different entries to the same dCDN, the dCDN can convey shortcut or differentiated quality of services.

4.3. Example of Selecting a Downstream CDN based on ALTO Maps

In the following, we will outline an example of dCDN selection by a uCDN based on ALTO maps provided by dCDNs. Consider the following example: An upstream CDN (uCDN) has agreed on CDN interconnection with several downstream CDNs (dCDN-a, dCDN-b, and dCDN-c). Each of these downstream CDNs runs an ALTO server to provide aforementioned ALTO information. Whenever the upstream CDN receives a request from an end user and has determined that this request is best served by an interconnected dCDN, the uCDN uses ALTO maps to make a redirection decision. For a given request, assume that only the ALTO network maps provided by dCDN-a and dCDN-c include the endhost. The uCDN first looks up the PIDs of the endhost in the two network maps from the two dCDNs, then search the PID properties to find out the capabilities of each dCDN for the endhost. If only one dCDN supports the required capabilities, then the uCDN chooses the dCDN. Otherwise, if Cost Maps are available to provide additional server

selection information (e.g., a Cost Map defining latency), the uCDN picks the dCDN with better cost performance.

4.4. Advantages of using ALTO

The following reasons make ALTO a suitable candidate protocol for downstream CDN selection as part of CDNI request routing and in particular for a FCI protocol:

- o CDN request routing is done at the application layer. ALTO is a protocol specifically designed to improve application layer traffic (and application layer connections among hosts on the Internet) by providing additional information to applications that these applications could not easily retrieve themselves. For CDNI, this is exactly the case: a uCDN wants to improve application layer CDN request routing by using dedicated information (provided by a dCDN) that the uCDN could not easily obtain otherwise.
- o The semantics of an ALTO network are an exact match for the needed information to convey a footprint by a downstream CDN, in particular if such a footprint is being expressed by IP-prefix ranges.
- o ALTO cost maps are suitable to express various types of numeric values and can hence be used by an upstream CDN to obtain metrics for capabilities associated with a given dCDN for a given footprint. Further, an ALTO cost map could also convey relevant network topology information other than simply routing hops or reachability. This facilitates advanced and more sophisticated selection of a downstream CDN based on various metrics by the upstream CDN and increases flexibility to cover different use cases and business models for CDN interconnection.
- o Flexible granularity: The concept of the PID and ALTO network/cost maps allows for different degrees of granularity. This enables a dCDN to differentiate the delivery quality for serving an end user request on a fine granularity depending on the end user location (and not only express delivery quality e.g. on an AS-level). It remains at the discretion of each dCDN how fine-granular the ALTO network and cost maps are that it publishes.
- o ALTO maps can be signed and hence provide inherent integrity protection (see Section 6)

5. Useful ALTO extensions for CDNI Request Routing

It is envisioned that yet-to-be-defined ALTO extensions will be standardized that make the ALTO protocol more suitable and useful for applications other than the originally considered P2P use case [I-D.marocco-alto-next]. Some of these extensions to the ALTO protocol would be useful for ALTO to be used as a protocol within CDNI request routing, and in particular within the "Footprint and Capabilities Advertisement" part of the CDNI request routing interface.

The following proposed extensions to ALTO would be beneficial to facilitate CDNI request routing with ALTO as outlined in Section 4:

- o Server-initiated Notifications and Incremental Updates: In case the footprint or the capabilities of a downstream CDN change abruptly (i.e. unexpectedly from the perspective of an upstream CDN), server initiated notifications would enable a dCDN to directly inform an upstream CDN about such changes. Consider the case where - due to failure - part of the footprint of the dCDN is not functioning, i.e. the CDN cannot serve content to such clients with reasonable QoS. Without server-initiated notifications, the uCDN might still use a very recent network and cost map from dCDN, and therefore redirect request to dCDN which it cannot serve. Similarly, the possibility for incremental updates would enable efficient conveyance of the aforementioned (or similar) status changes by the dCDN to the uCDN. A proposal for server-initiated ALTO updates can be found in [I-D.marocco-alto-ws]. A discussion of incremental ALTO updates can be found in [I-D.schwan-alto-incr-updates].
- o Content Availability on Hosts: A dCDN might want to express CDN capabilities in terms of certain content types (e.g. codecs/formats, or content from certain content providers). A new endpoint property for ALTO that would be able to express such "content availability" would enable a dCDN to make available such information to an upstream CDN. This would enable a uCDN to determine if a given dCDN actually has the capabilities for a given request with respect to the type of content requested.

- o Resource Availability on Hosts or Links: The capabilities on links (e.g. maximum bandwidth) or caches (e.g. average load) might be useful information for an upstream CDN for optimized downstream CDN selection. For instance, if a uCDN receives a streaming request for content with a certain bitrate, it needs to know if it is likely that a dCDN can fulfill such stringent application-level requirements (i.e. can be expected to have enough consistent bandwidth) before it redirects the request. In general, if ALTO could convey such information via new endpoint properties, it would enable more sophisticated means for downstream CDN selection with ALTO.

6. Security Considerations

One important security consideration is the proper authentication of advertisement information provided by a downstream CDN. The ALTO protocol provides a specification for a signature of ALTO maps (see 8.2.2. of [I-D.ietf-alto-protocol]). ALTO thus provides a proper means for protecting the integrity of footprint advertisement information.

More Security Considerations will be discussed in a future version of this document.

7. Summary and Outlook

This document presented concrete examples of how ALTO can be used within the downstream CDN selection of CDNI Request Routing. Further, the document provides arguments why ALTO is a meaningful protocol in this context. Essentially, ALTO network and cost maps are a means to provide detailed and various types of information to an upstream CDN, in order to facilitate well-considered downstream CDN selection.

The intention of this document is to find consensus in the CDNI WG that ALTO is a useful protocol for CDNI request routing, and that ALTO has many benefits for proper selection of a downstream CDN. The overall objective is to form agreement on how ALTO should be used within the CDNI request routing protocol. It is the intention to capture the outcome of such continuing discussions in future versions of this document.

8. Acknowledgements

Jan Seedorf is partially supported by the CHANGE project (CHANGE: Enabling Innovation in the Internet Architecture through Flexible Flow-Processing Extensions, <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.

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

9. References

9.1. Normative References

- [RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, October 2009.
- [RFC6707] Niven-Jenkins, B., Le Faucheur, F., and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", RFC 6707, September 2012.
- [RFC6770] Bertrand, G., Stephan, E., Burbridge, T., Eardley, P., Ma, K., and G. Watson, "Use Cases for Content Delivery Network Interconnection", RFC 6770, November 2012.

9.2. Informative References

- [I-D.peterson-cdni-strawman]
Peterson, L. and J. Hartman, "Content Distribution Network Interconnection (CDNI) Problem Statement", draft-peterson-cdni-strawman-01 (work in progress), May 2011.
- [I-D.ietf-cdni-problem-statement]
Niven-Jenkins, B., Faucheur, F., and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", draft-ietf-cdni-problem-statement-08 (work in progress), June 2012.
- [I-D.marocco-alto-next]
Marocco, E. and V. Gurbani, "Extending the Application-Layer Traffic Optimization (ALTO) Protocol", draft-marocco-alto-next-00 (work in progress), January 2012.

- [I-D.ietf-alto-protocol]
Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-20 (work in progress), October 2013.
- [I-D.ietf-cdni-requirements]
Leung, K. and Y. Lee, "Content Distribution Network Interconnection (CDNI) Requirements", draft-ietf-cdni-requirements-11 (work in progress), October 2013.
- [I-D.ietf-cdni-use-cases]
Bertrand, G., Emile, S., Burbridge, T., Eardley, P., Ma, K., and G. Watson, "Use Cases for Content Delivery Network Interconnection", draft-ietf-cdni-use-cases-10 (work in progress), August 2012.
- [I-D.marocco-alto-ws]
Marocco, E. and J. Seedorf, "WebSocket-based server-to-client notifications for the Application-Layer Traffic Optimization (ALTO) Protocol", draft-marocco-alto-ws-01 (work in progress), July 2012.
- [I-D.schwan-alto-incr-updates]
Schwan, N. and B. Roome, "ALTO Incremental Updates", draft-schwan-alto-incr-updates-02 (work in progress), July 2012.
- [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.seedorf-alto-for-cdni]
Seedorf, J., "ALTO for CDNI Request Routing", draft-seedorf-alto-for-cdni-00 (work in progress), October 2011.
- [I-D.ietf-cdni-framework]
Peterson, L. and B. Davie, "Framework for CDN Interconnection", draft-ietf-cdni-framework-06 (work in progress), October 2013.
- [I-D.liu-cdni-cost]
Liu, H., "A Cost Perspective on Using Multiple CDNs", draft-liu-cdni-cost-00 (work in progress), October 2011.
- [I-D.spp-cdni-rr-foot-cap-semantics]
Seedorf, J., Peterson, J., Previdi, S., Brandenburg, R., and K. Ma, "CDNI Request Routing: Footprint and

Capabilities Semantics", draft-spp-cdni-rr-foot-cap-
semantics-04 (work in progress), February 2013.

Authors' Addresses

Jan Seedorf
NEC Laboratories Europe, NEC Europe Ltd.
Kurfuersten-Anlage 36
Heidelberg 69115
Germany

Phone: +49 (0) 6221 4342 221
Email: jan.seedorf@neclab.eu
URI: <http://www.neclab.eu>

Y.R. Yang
Yale University
51 Prospect Street
New Haven 06511
USA

Email: yry@cs.yale.edu
URI: <http://www.cs.yale.edu/~yry/>

LMAP
Internet-Draft
Intended status: Informational
Expires: April 24, 2014

J. Seedorf
NEC
D. Goergen
R. State
University of Luxembourg
V. Gurbani
Bell Labs, Alcatel-Lucent
E. Marocco
Telecom Italia
October 21, 2013

ALTO for Querying LMAP Results
draft-seedorf-lmap-alto-02

Abstract

In the context of Large-Scale Measurement of Broadband Performance (LMAP), measurement results are currently made available to the public either at the finest granularity level (e.g. as a list of results of all individual tests), or in a very high level human-readable format (e.g. as PDF reports). This document argues that there is a need for an intermediate way to provide access to large-scale network measurement results, flexible enough to enable querying of specific and possibly aggregated data. The Application-Layer Traffic Optimization (ALTO) Protocol, defined with the goal to provide applications with network information, seems a good candidate to fulfill such a role. Finally, we describe our methodology for analyzing the United States Federal Communication Commission's (FCC) Measuring Broadband America (MBA) dataset to derive required topology and cost maps suitable for consumption by an ALTO server.

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 April 24, 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. Example Use Cases	5
3. Advantages of using ALTO	6
4. Examples	7
4.1. Download speeds	7
4.1.1. Network map	8
4.1.2. Cost map	9
5. Discussion of Useful ALTO Extensions	10
6. Case study: Analyzing a large-scale dataset	11
6.1. Challenges in data analysis	11
6.2. Geo-locating the units	12
7. Security considerations	15
8. IANA considerations	16
9. Conclusion	17
10. References	18
10.1. Normative References	18
10.2. Informative References	18
Appendix A. Acknowledgment	19
Authors' Addresses	20

1. Introduction

Recently, there is a discussion on standardizing protocols that would allow measurements of broadband performance on a large scale (LMAP [I-D.schulzrinne-lmap-requirements]). In principle, the vision is that "user networks gather data, either on their own initiative or instructed by a measurement controller, and then upload the measurement results to a designated measurement server."

Apart from protocols that can be used to gather measurement data and to upload such data to dedicated servers, there is also a need for protocols to retrieve - potentially aggregated - measurement results for a certain network (or part of a network), possibly in an automated way. Currently, two extremes are being used to provide access to large-scale measurement results: One the one hand, highly aggregated results for certain networks may be made available in the form of PDFs or figures. Such presentations may be suitable for certain use cases, but certainly do not allow a user (or entity such as a service provider) to select specific criteria and then create corresponding results. On the other hand, complete and detailed results may be made available in the form of comma-separated-values (csv) files. Such data sets typically include the complete results being measured on a very fine-grained level and usually imply large file sizes (of result data sets). Such detailed result data sets are very useful e.g. for the scientific community because they enable to execute complex data analytics algorithms or queries to analyse results.

Considering the two extremes discussed above, this document argues that there is a need for an intermediate way to provide access to large-scale network measurement results: It must be possible to query for specific, possibly aggregated, results in a flexible way. Otherwise, entities interested in measurement results either cannot select what kind of result aggregation they desire, or must always fetch large amounts of detailed results and process these huge datasets themselves. The need for a flexible mechanism to query for dedicated, partial results becomes evident when considering use cases where a service provider or a process wants to use certain measurement results in an automated fashion. For instance, consider a video streaming service provider which wants to know for a given end-user request the average download speed by the end user's access provider in the end user's region (e.g. to optimize/parametrize its http adaptive streaming service). Or consider a website which is interested in retrieving average connectivity speeds for users depending on access provider, region, or type of contract (e.g. to be able to adapt web content on a per-request basis according to such statistics).

This document argues that use cases as described above may enhance the value of measurements of broadband performance on a large scale (LMAP), given that it is possible to query for selected results in an automated fashion. Therefore, in order to facilitate such use cases, a protocol is needed that enables to query LMAP measurements results while allowing to specify certain parameters that narrow down the particular data (i.e. measurement results) the issuer of the query is interested in. This document argues that ALTO [RFC5693] [I-D.ietf-alto-protocol] could be a suitable candidate for such a flexible LMAP result query protocol.

2. Example Use Cases

To motivate the usefulness of ALTO for querying LMAP results, consider some key use cases:

- o Video Streaming Service Provider: For HTTP adaptive streaming, it may be very useful to be able to query for average measurement values regarding a particular end user's access network provider. For instance, consider a video streaming service provider that queries LMAP measurement results to retrieve for a given end-user request the average download speed by the end user's access provider in the end user's region. Such data could help the service provider to optimize/parametrize its HTTP adaptive streaming service.
- o Website Front End Optimization: A website might be interested in statistics about average connectivity types or download speeds for a given end user request in order to dynamically adapt HTML/CSS/JavaScript content depending on such information (sometimes referred to as "Front End Optimization"). For instance, image compression may or may not be employed depending on the average connectivity type/speed of a user in a given region or with a given access network provider.
- o Display estimation of service quality or total download time to users: A webservice could use statistics about average download speeds for a given ISP and/or region to estimate Quality-of-Service for provided services (e.g. to indicate to the user what Quality-of-Experience to expect when clicking on a given link) or to estimate (and display to the user) the total download time for given content.
- o Troubleshooting: In general, any service on the Internet may be interested in LMAP data for troubleshooting. In case a service does not work as expected (e.g. low throughput, high packet loss, ...), it may be of value for the service provider to retrieve (fairly) recent measurement data regarding the host that is requesting the service.
- o TBD: add more use cases

3. Advantages of using ALTO

The ALTO protocol [I-D.ietf-alto-protocol] specifies a very lightweight JSON-based encoding for network information and can play an important role in querying the measurement results as we argue in Section 2.

ALTO is designed on two abstractions that are useful here. First is the abstraction of the physical network topology into an aggregated but logical topology. In this abstract topological view, referred to as "network map", individual hosts are aggregated into a well defined network location identifier called a PID. Hosts could be aggregated into the PID depending on certain identifying characteristics such as geographical location, serving ISP, network mask, nominal access speed, or any mix of them. The "network map" abstraction is essential for exporting network information in a scalable and privacy-preserving way.

The second abstraction that is useful for LMAP is the notion of a "cost map". Each PID identified in the network map can, in a sense, become a vertex in a cost map, and each edge joining adjacent vertices can have an associated cost. The cost can be defined by the measurement server and can indicate routing hops, the financial cost of sending data over the link, available bandwidth on the link with bottlenecked links increasingly showing a smaller value, or a user-defined cost attribute that allows arbitrary reasoning.

The ALTO protocol defines several basic services based on such abstractions, but additional ones can be easily defined as extensions.

There are other advantages to using ALTO as well. The protocol is defined as a set of REST APIs on top of HTTP. The data carried by the protocol is encoded as JSON. Queries can be performed by clients locally after downloading the entire topological and cost maps or clients can send filtered requests to the ALTO server such that the ALTO server performs the required computation and returns the results. The protocol supports a set of atomic constraints related to equality that can be used to filter results and only obtain a set of interest to the query.

Additionally, protocol extensions that could also be useful for the LMAP usage scenario (e.g. extensions for incremental updates, for asynchronous change notifications and for encoding of multiple costs within the same cost map) have been proposed and are currently being discussed in the ALTO WG.

4. Examples

[NOTE: syntax most certainly wrong!]

4.1. Download speeds

This section shows, as an example, how average download speeds measured in a given time interval can be reported. The aggregation approach in this case is based on ISP and geographical location. Two types of data are reported in this example:

- o data collected from measurements against specific endpoints (e.g. active measurements);
- o data collected from all measurements (e.g. passive measurements).

4.1.1. Network map

```
{
  "meta" : {},
  "data" : {
    "map-vtag" : "1266506139",
    "map" : {
      "ISP1-GEO1" : {
        "ipv4" : [ "10.1.0.0/16", "172.20.0.0/16" ]
      },
      "ISP2-GEO1" : {
        "ipv4" : [ "10.2.0.0/17" ]
      },
      "ISP3-GEO1" : {
        "ipv4" : [ "10.3.0.0/16" ]
      },
      "ISP2-GEO2" : {
        "ipv4" : [ "10.2.128.0/17" ]
      },
      "ISP4-GEO2" : {
        "ipv4" : [ "10.4.0.0/16" ]
      },
      .
      .
      .
      "MSMNT-CL1" : {
        "ipv4" : [ "192.168.0.0/30" ]
      },
      "TOTAL" : {
        "ipv4" : [ "0.0.0.0/0" ]
      }
    }
  }
}
```

4.1.2. Cost map

```
{
  "meta" : {},
  "data" : {
    "cost-mode" : "numerical",
    "cost-type" : "avg-dl-speed",
    "map-vtag" : "1266506139",
    "time-interval" : "2629740",
    "map" : {
      "ISP1-GEO1": { "MSMNT-CL1" : 13.2,
                     "TOTAL" : 10.2},
      "ISP2-GEO1": { "MSMNT-CL1" : 11.4,
                     "TOTAL" : 12.3},
      "ISP3-GEO1": { "MSMNT-CL1" : 13.2,
                     "TOTAL" : 10.2},
      .
      .
      .
    }
  }
}
```

5. Discussion of Useful ALTO Extensions

The base ALTO Protocol as specified in [I-D.ietf-alto-protocol] can in principle be used to enable a more flexible way to provide access to large-scale network measurement results as discussed in the previous sections of this document. However, certain extensions to the base ALTO Protocol that have recently been proposed in the ALTO WG would allow to better enable the use cases discussed in Section 2:

- o Server-initiated Notifications: In [I-D.marocco-alto-ws], it has been proposed to enhance the ALTO protocol such that servers can notify clients about newly available ALTO maps. In the context of this document, this extension would allow applications to be notified when certain new LMAP measurements are available, such as new measurement results on average download speeds. These new results could then be downloaded and used immediately by applications.
- o Incremental Updates: In [I-D.schwan-alto-incr-updates], it has been proposed to enhance the ALTO protocol with incremental updates, such that clients can retrieve partial updates for ALTO maps instead of always downloading a full ALTO map (even when only a small fraction of the ALTO map has changed compared to a previous version). When ALTO is used for querying LMAP results, the corresponding ALTO maps may potentially be quite large (e.g. when a webservice queries for particular, detailed results regarding a whole ISP). In this case, incremental ALTO updates would be a very useful mechanism for applications to retrieve updates of ALTO maps, as a reduced amount of data would be needed for transmitting these maps.

6. Case study: Analyzing a large-scale dataset

Measuring broadband performance is increasingly important as communications continue to move towards the Internet. Internet service providers (ISP), national agencies and other entities gather broadband data and may provide some, or all, of the dataset to the public for analysis. As we argue above, there are two extremes prevalent for presenting large-scale data. One is in the form of charts, figures, or summarized reports amenable for easy and quick consumption. The other extreme includes releasing raw data in the form of large files containing tables formatted as values separated by a delimiter. While the former is indispensable to acquire a summary view of the dataset, it does not suffice for additional analysis beyond what is presented. Conversely, the problem with the latter option (raw files) is that the unsuspecting user perusing them is lost in the deluge of data.

We offer the argument that a reasonable medium between the two extremes may be the ALTO protocol [I-D.ietf-alto-protocol]. A necessary prerequisite for using ALTO is abstracting the network information into a form that is suitable for consumption by the protocol. The implication of using ALTO is that data from any large-scale measurement effort must first be distilled in two maps: a topology map and a cost map. Further analysis and ad-hoc queries can be subsequently performed on the normalized dataset.

In the United States, the Federal Communication Commission (FCC) has embarked on a nationwide performance study of residential wireline broadband service [fcc]. Our aim is to use the raw datasets from this study for analysis and to create a topology map and a cost map from this dataset. ALTO queries aimed at these maps will enable users and interested parties to fulfill the use cases listed in Section 2.

6.1. Challenges in data analysis

The FCC Measuring Broadband America (MBA) study consisted of 7,782 volunteers spread across the United States with adequate geographic diversity. Volunteers opted in for the study, however, each of the volunteers remained anonymous. An opaque integral number (`unit_id`) represented a subscriber in the raw dataset. This `unit_id` remains constant during the duration of the study in the dataset and uniquely identifies a volunteer subscriber, even if the subscriber switches the ISP. More detail about the methodology used is described in [fcc].

The dataset consisted of 12 tables, each table corresponding to the data drawn from a certain performance test. For the analysis we

present in this document we focus on the "curr_dns" table, which contains the time taken for the ISP's recursive DNS resolver to return a DNS A RR for a popular website domain name. This test was ran approximately every hour in a 24-hour period, and produced about 75-78 million records per month. This resulted in a typical file size in the range of 6-7 GBytes per month. We note that the "curr_dns" table is one of the smaller tables in the dataset.

The first challenge, therefore, was to arrive at computing resources comparable in scale with respect to the dataset consisting of millions of records spread across gigabyte-sized files. To analyze the volume of data we used a canonical Map-Reduce computational paradigm on a Hadoop cluster (more details on the methodology are outlined in Section 6.2).

A second, more pressing challenge, was to identify the geographic location of the unit_ids generating the data. In order to derive a topological map and impose costs on the links, it is important to know the physical locations of the unit_ids that contributed the measurements. However, in the MBA dataset, the population is anonymized and the individual subscriber reporting the measurement data is simply referred to by an opaque integral number. Therefore, an important task was to use the information in the public tables to reveal a coarse location of the subscriber.

We outline the methodology we used to do so in the next section. We stress that this methodology does not identify the specific location of a subscriber, who still remains anonymous. Instead, it simply locates the subscriber in a larger metropolitan region. This level of granularity suffices for our work.

6.2. Geo-locating the units

To geo-locate the units, we simply note that broadband subscriber devices are likely to be configured using DHCP by their ISP. Besides imparting an IP address to the subscriber device, DHCP also populates the DNS name servers the subscriber devices uses for DNS queries. In most installations, these DNS name servers are located in close physical proximity of the subscriber device. The FCC technical appendix states that the DNS resolution tests were targeted directly at the ISP's recursive resolvers to circumvent caching and users configuring the subscriber device to circumvent the ISP's DNS resolvers. Therefore, a reasonable approximation of a subscribers geo-location could be the geographic location of the DNS name server serving the subscriber. We use this very heuristic to geo-locate a subscriber.

Thus our first, and very simple filter consisted of obtaining a

mapping from a unit_id (representing a subscriber) to one or more DNS name servers that the unit_id is sending DNS requests to. It turned out that while this was a necessary condition for advancing, it was not a sufficient one. The raw data would need to be further processed to reduce inconsistencies and remove outliers. A number of interesting artifacts were uncovered during further processing of the data. These artifacts informed the selection of the unit_ids for further analysis.

The artifacts are documented below.

- o A handful of unit_ids were geo-located in areas outside the contiguous United States, such as Ukraine, Poland or the United Kingdom. We theorize that the subscribers corresponding to the unit_ids geo-located outside the contiguous United States had simply configured their devices to use alternate DNS servers, probably located outside the United States. We removed these records before conducting our analysis.
- o We also observed a reasonable number of non-ISP DNS resolvers, especially Google's 8.8.8.8 and 8.8.4.4 and OpenDNS 208.67.222.222 and 208.67.220.220. These 4 public DNS servers are geo-located in California. We removed these records to ensure that the specific location that these resolvers represented was not oversampled.
- o We noticed that a large number of unit_ids were being geo-located in Potwin, Kansas. Intrigued as to why there appeared to be a large population of Internet users being located in a small rural community in Kansas, we investigated further. It appears that Potwin, Kansas is the geographical center of the United States and a number of ISPs have chosen to establish data centers in or around the Potwin area. These ISPs generally locate their primary or secondary DNS name servers in Potwin-area data centers, thus accounting for the popularity of Potwin as an Internet destination. We continue to further investigate on minimizing the impact of such natural aggregation points that, if not accounted for, will skew our results in an unwarranted direction.
- o We observed some unit_ids changing ISPs during the observation period. This is a normal occurrence and to the extent that the unit_id is geo-located in the same geographical area after the change in ISP, we do not exclude such unit_ids from further analysis.

Subsequent filters extracted the stable unit_ids from our dataset. In order to determine which unit_id are stable, i.e., remain constant with respect to their geographic location over the observation period from January to December 2012, we extracted for each unit_id the IP

address of each DNS name server it consulted. This is obtained by applying the map reduce paradigm on the DNS dataset. We extracted for each `unit_id` the triggered DNS servers and obtained the individual DNS servers accessed by a `unit_id`. This was repeated for each month of the observation period. The resulting sets were cleaned up of private IP addresses and other artifacts discussed above. The cleaned set consisted of about 8000 distinct `unit_id`.

In order to determine the stability of each `unit_id` we proceeded to sum up the occurrences of IP addresses over the whole observation period separated in monthly files. If the IP address of a DNS server occurred 12 times this meant that the `unit_id` always accessed the same DNS server and therefore remained stable over the observation period. The obtained stable `unit_ids`, around 1500, will be used for further analysis. Assuming a 99% confidence level and ± 3 point margin of error, we will require a sample of 1494 `unit_ids`. With our stable `unit_id` set of 1500 `unit_ids`, we are now positioned to perform further analysis on the dataset to create the full topology and cost maps.

Table 1 presents a sample of the geographic location data that we have uncovered for `unit_ids`. A complete list of identified units superimposed on the geographical map of the United States is available at <http://cdb.io/13UOHgD>.

Unit ID	City, State	Latitude/Longitude
872	Morganville, NJ	40.35950089,-74.26280212
885	Madison, WI	43.07310104,-89.40119934
898	Foley, AL	30.40660095,-87.68360138
7969	Manteca, CA	37.79740143,-121.2160034
8024	Quincy, MA	42.25289917,-71.00229645

Sample unit identification tuples

Table 1

7. Security considerations

There are no security artifacts invalidated due to our analysis in Section 6. All of our analysis was performed on publicly available data. However, we do note that some privacy may have been lost based on our analysis. In the raw dataset, the unit identifiers are opaque strings with no immediate correlation with a geographic location. After our analysis, while the unit identifiers still remain opaque, they are nonetheless correlated to a specific, though coarse, geographic location.

8. IANA considerations

This document does not contain any IANA considerations.

9. Conclusion

This document argues that, compared to existing solutions, there may be a need for a more flexible way to provide access to large-scale network measurement results. Further, the document argues that the ALTO protocol is a good candidate to enable querying for specific, possibly aggregated, measurement results in a flexible way. Examples of how such a flexible query mechanism for large-scale measurement results could look like based on ALTO are given.

With respect to the case study in Section 6, identification of the geographic location of the unit_ids generating the performance data is essential in order to continue the work. We have presented a methodology and some early results in identifying a geographic location. This location, although coarse, suffices for our future work that will consist of further data mining and analysis to create appropriate ALTO network and cost maps.

10. References

10.1. Normative References

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

10.2. Informative References

- [I-D.ietf-alto-protocol]
Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-20 (work in progress), October 2013.
- [I-D.marocco-alto-ws]
Marocco, E. and J. Seedorf, "WebSocket-based server-to-client notifications for the Application-Layer Traffic Optimization (ALTO) Protocol", draft-marocco-alto-ws-01 (work in progress), July 2012.
- [I-D.schulzrinne-lmap-requirements]
Schulzrinne, H., Johnston, W., and J. Miller, "Large-Scale Measurement of Broadband Performance: Use Cases, Architecture and Protocol Requirements", draft-schulzrinne-lmap-requirements-00 (work in progress), September 2012.
- [I-D.schwan-alto-incr-updates]
Schwan, N. and B. Roome, "ALTO Incremental Updates", draft-schwan-alto-incr-updates-02 (work in progress), July 2012.
- [fcc] United States Federal Communications Commission, "Measuring Broadband America", Accessed July 12, 2013, <http://www.fcc.gov/measuring-broadband-america>.

Appendix A. Acknowledgment

Jan Seedorf is partially supported by the mPlane project (mPlane: an Intelligent Measurement Plane for Future Network and Application Management), a research project supported by the European Commission under its 7th Framework Program (contract no. 318627). 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 mPlane project or the European Commission.

Authors' Addresses

Jan Seedorf
NEC
Kurfuerstenanlage 36
Heidelberg 69115
Germany

Phone: +49 6221 4342 221
Fax: +49 6221 4342 155
Email: seedorf@neclab.eu

David Goergen
University of Luxembourg

Email: david.goergen@uni.lu

Radu State
University of Luxembourg

Email: radu.state@uni.lu

Vijay K. Gurbani
Bell Labs, Alcatel-Lucent

Email: vkg@bell-labs.com

Enrico Marocco
Telecom Italia
Via G. Reiss Romoli, 274
Turin 10148
Italy

Email: enrico.marocco@telecomitalia.it

ALTO
Internet-Draft
Intended status: Standards Track
Expires: April 24, 2014

H. Song
Huawei
Y. Sun
ICT Chinese Academy of Sciences
October 21, 2013

ALTO Protocol Extension For Overlay Routing
draft-song-alto-overlay-routing-00

Abstract

This document describes an ALTO protocol extension for overlay routing. It considers three different methods to route traffic from a data source to a data receiver, which are direct Internet routing, VPN tunnel, and overlay routing via intermediate/relay node(s), analyze their use cases in real world and then proposes an extension to ALTO protocol so as to support a ALTO client to get cost value between hosts via these different routing methods.

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 April 24, 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	5
3. Overlay Routing Cost Service Extension	5
3.1. Overlay Routing Cost	5
3.1.1. Media Type	5
3.1.2. HTTP Method	5
3.1.3. Accept Input Parameters	5
3.1.4. Capabilities	6
3.1.5. Uses	7
3.1.6. Response	7
3.1.7. Example	8
4. References	9
4.1. Normative References	9
4.2. Informative References	9
Authors' Addresses	9

1. Introduction

ALTO protocol [I-D.ietf-alto-protocol] provides an interface to applications with appropriate information to guide an optimal node selection based on the Internet service provider's policy 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. 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.

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, or between two different data center locations to transfer data. Note that in this document, the VPN refers to the VPN service provided by the Internet Service Provider, which is used to guarantee the quality of delivery service. The service/content providers often classify the data into delay-sensitive and delay-insensitive, Because VPN tunnel is more expensive than the direct Internet routing method, and at the same time has higher QoS guarantee than the direct Internet routing method. The delay-sensitive data is usually transported via the VPN tunnel and the delay-insensitive data can be transported over the VPN tunnel if there is available capability for it, but can also be transported over the direct Internet routing method in order to leave VPN capability for delay-sensitive data. The overlay routing method via intermediate nodes (can be an intermediate CDN server, a data center gateway and etc.) is an optimization to the direct Internet routing method, thus the QoS is a little higher than the direct Internet routing method, but it is not as good as a VPN tunnel.

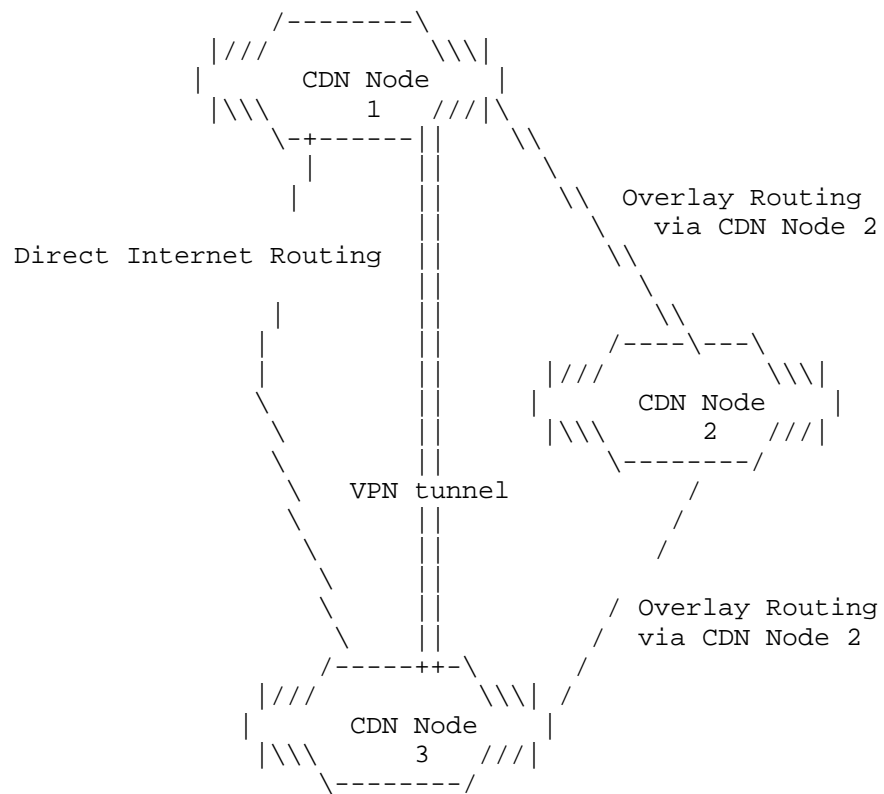


Figure 1. Different ways for sending content from Node 1 to Node 3

So the transport between two Internet hosts can be from:

- direct Internet routing
- one/more intermediate overlay nodes
- a VPN tunnel

In this document, we propose a new overlay routing cost service for ALTO protocol, which can be used by the ALTO client to compare the routing cost through different paths between two Internet hosts. As it is not usually to use two or more relay nodes, we do not consider that in this document. Actually, if there are two or more relay nodes, it can be considered as multiple one relay node and use the service provided by this document multiple times to solve the issue.

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]. The concepts and data formats are consistent with ALTO protocol base document

. [I-D.ietf-alto-protocol]

3. Overlay Routing Cost Service Extension

The reader is assumed to read ALTO protocol before this document, especially section 8 of ALTO protocol.

The Overlay Routing Cost Service provides information about costs between individual endpoints through direct Internet routing, VPN or overlay routing.

3.1. Overlay Routing Cost

An Overlay Routing Cost resource provides information about costs between individual endpoints from different paths.

3.1.1. Media Type

The media type of Overlay Routing Cost is "application/alto-overlayroutingcost+json".

3.1.2. HTTP Method

The Overlay Routing Cost resource is requested using the HTTP POST method.

3.1.3. Accept Input Parameters

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

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

```
TypedEndpointAddr  srcs<0..*>;
TypedEndpointAddr  dstc<0..*>;
JSONBool drr;
JSONBool vpn;
[TypedEndpointAddr relays<0..*>;]
} EndpointFilter;
```

with fields:

cost-type The Cost Type (Section 10.7 of ALTO protocol) 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. 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 11.3.2 of ALTO protocol).

endpoints A list of endpoints including source endpoints, relay endpoints and destination endpoints for which path costs are to be returned. If "drr" is true, the ALTO server MUST provide the cost of direct Internet routing from the source endpoint to the destination endpoint. If the value provided by a ALTO Client for "vpn" is true and a provider's VPN is existed between a source and destination endpoints pairs, the ALTO server MUST provide the cost of using the VPN tunnel. If the value provided by a ALTO Client for "vpn" is true, but the ALTO server cannot find the VPN information between any source and destination endpoints, it MUST ignore it. The ALTO server MUST provide the cost from source endpoints to destination endpoints through each relay node that is listed in the relay node array. 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 13.3 for discussion and considerations regarding this mode). The Source and Destination Endpoint lists MUST NOT be both empty. The relay node list can be empty. Note that ALTO client SHOULD NOT set "drr" to true, "vpn" to false and the relay node list to empty in a single request, in that case, the ALTO client should use Endpoint Cost Service.

3.1.4. Capabilities

In this document, we define `OverlayRoutingCostCapabilities` the same as `FilteredCostMapCapabilities`. See Section 11.3.2.4 of ALTO protocol.

3.1.5. Uses

None.

3.1.6. Response

The "meta" field of an Overlay Routing Cost response MUST include the "cost-type" key, to indicate the Cost Type used.

The data component of an Overlay Routing Cost response is named "overlay-routing-cost-map", which is a JSON object of type OverlayRoutingCostMapData:

```
object {  
  [OverlayRoutingCostMapData overlay-routing-cost-map;]  
  [EndpointTwoLevelCosts vpncost;]  
  [EndpointTwoLevelCosts drrrcost;]  
} InfoResourceOverlayRoutingCostMap : ResponseEntityBase;  
  
object-map {  
  TypedEndpointAddr -> EndpointTwoLevelCosts;  
} OverlayRoutingCostMapData;  
  
object-map {  
  TypedEndpointAddr -> EndpointOneLevelCosts;  
} EndpointTwoLevelCosts;  
  
object-map {  
  TypedEndpointAddr -> JSONValue;  
} EndpointOneLevelCosts;
```

Specifically, an OverlayRoutingCostMapData object is a dictionary map with each key representing a TypedEndpointAddr string identifying the Source Endpoint specified in the input parameters, and for each Source Endpoint, a EndpointTwoLevelCosts dictionary map object has each key representing a TypedEndpointAddr identifying the Destination Endpoint, and for each Destination Endpoint, a EndpointOneLevelCosts dictionary map object denotes the associated cost with each relay nodes specified in the input parameters.

The key "vpncost" is a dictionary map with each key representing a TypedEndpointAddr string identifying the Source Endpoint specified in the input parameters, and for each Source Endpoint, a EndpointOneLevelCost dictionary map object denotes the associated VPN cost to each Destination Endpoint if existed. So the vpncost may only contain a few values where there are VPN tunnels between source endpoints and destination endpoints.

The key "drrcost" is a dictionary map with each key representing a TypedEndpointAddr string identifying the Source Endpoint specified in the input parameters, and for each Source Endpoint, a EndpointOneLeveCost dictionary map object denotes the associated cost to each Destination Endpoint.

Note that ALTO server SHOULD NOT provide only a single direct routing cost map.

3.1.7. Example

```
POST /overlayroutingcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: TBA
Content-Type: application/alto-overlayroutingcostparams+json
Accept: application/alto-overlayroutingcost+json,application/alto-error+json

{
  "cost-type": { "cost-mode" : "ordinal",
                 "cost-metric" : "routingcost" },
  "endpoints" : {
    "srcs": [ "ipv4:1.1.1.1" ],
    "dsts": [
      "ipv4:2.2.2.2",
      "ipv4:3.3.3.3",
    ],
    "drr": false,
    "vpn": true,
    "relays": [ "ipv4:6.6.6.6",
                "ipv4:7.7.7.7"
    ]
  }
}
```

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

```
{
  "meta" : {
    "cost-type": { "cost-mode" : "ordinal",
                   "cost-metric" : "routingcost"
    }
  },
  "overlay-routing-cost-map" : {
    "ipv4:1.1.1.1": {
      "ipv4:2.2.2.2": {
```

```
        "ipv4:6.6.6.6": 1,  
        "ipv4:7.7.7.7": 2,  
    },  
    "ipv4:3.3.3.3": {  
        "ipv4:6.6.6.6": 2,  
        "ipv4:7.7.7.7": 3  
    }  
}  
}  
}  
"vpncost": {  
    "ipv4:1.1.1.1": {  
        "ipv4:2.2.2.2": 0  
    }  
}  
}
```

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

4.2. Informative References

[I-D.ietf-alto-protocol]
 Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-20 (work in progress), October 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

Sun Yi
ICT Chinese Academy of Sciences

Email: sunyi@ict.ac.cn

ALTO Working Group
Internet-Draft
Intended status: Standards Track
Expires: April 24, 2014

Q. Wu
Y. Lee
D. Dhody
Huawei
S. Randriamasy
Alcatel-Lucent
October 21, 2013

ALTO Traffic Engineering Cost Metrics
draft-wu-alto-te-metrics-00

Abstract

Cost Metric is a basic concept in Application-Layer Traffic Optimization (ALTO). It is used in both the Cost Map Service and the Endpoint Cost Service. Future extensions to ALTO may also use Cost Metric.

Different applications may benefit from different Cost Metrics. For example, a Resource Consumer may prefer Resource Providers that have low latency to the Resource Consumer. However the base ALTO protocol [ALTO] has defined only a single cost metric, i.e., the generic "routingcost" metric (Sec. 14.2 of ALTO base specification [ALTO]).

In this document, we define XXX Cost Metrics, derived from OSPF-TE and ISIS-TE, to measure network delay, jitter, packet loss, hop count, and bandwidth. The metrics defined in this document provide a relatively comprehensive set of Cost Metrics for ALTO focusing on traffic engineering. Additional Cost Metrics such as financial cost metrics may be defined in other documents.

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 April 24, 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. Metric: delay	5
4. Metric: delayjitter	7
5. Metric: pktloss	9
6. Cost Metric: hopcount	11
7. Metrics: bandwidth	12
8. Metric: maxbw	14
9. Maximum Reserved Bandwdith: maxreservbw	15
10. Metric: unreservbw	16
11. Metric: residuebw	17
12. Metric: availbw	18
13. Metric: utilbw	19
14. Security Considerations	20
15. IANA Considerations	21
16. References	22
16.1. Normative References	22
16.2. Informative References	22
Appendix A. Filtering constraint Extensions	23
Appendix B. Contributing Authors Addresses	25
Authors' Addresses	26

1. Introduction

Cost Metric is a basic concept in Application-Layer Traffic Optimization (ALTO). It is used in both the Cost Map Service and the Endpoint Cost Service. In particular, applications may benefit from knowing network performance measured in several Cost Metrics. For example, a more delay sensitive application may focus on latency, and a more bandwidth-sensitive application may focus on available bandwidth.

In this document, we define X Cost Metrics, extending the base ALTO protocol [ALTO], which has defined only a single Cost Metric, i.e., the generic "routingcost" metric (Sec. 14.2 of ALTO base specification [ALTO]).

The Cost Metrics that we define in this document focus on traffic engineering. Additional metrics may be defined in other documents. In particular, the Cost Metrics that we define in this document can be gathered from routing systems; [OSPF-TE], [ISIS-TE], [BGP-LS] and [BGP-PM] define mechanisms that allow an ALTO Server to retrieve and derive the necessary information to provide the metrics that we define in this document.

Note that the metrics that the ALTO Server retrieves may be defined for only links, and hence, the server will need to compose the link metrics to obtain path metrics used in services such as the Cost Map Service. In this definition, we define the metrics to be independent of link or path, considering that future ALTO extensions may define link-based services, and hence the defined metrics should still be usable.

One challenge in defining the metrics is that performance metrics often depend on configuration parameters. For example, the value of packet loss rate depends on the measurement interval. We handle this issue [YRY: IMPORTANT TO SOLVE]

The definitions of a set Cost Metrics can allow us to extend the base protocol (e.g., allowing output and constraints use different Cost Metrics), but such extensions are not in the scope of this document.

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. Metric: delay

Cost Metric name: delay

Metric Description: To specify spatial and temporal aggregated delay over the specified source and destination. The spatial aggregation unit is specified in the query context (e.g., PID to PID, or endhost to endhost); and the temporal unit is specified as the measurement interval.

Metric Unit: The unit is microseconds.

Metric Value Type:

A single 'JSONNumber' type value containing a non-negative integer component that may be followed by a fraction part and/or an exponent part.

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

Measurement details: YRY: SPECIFY MORE DETAILS.

Example 1:

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

```
{
  "cost-type": { "cost-mode" : "numerical",
                 "cost-metric" : "delay" },
  "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: TBA
Content-Type: application/alto-endpointcost+json
{
  "meta" : {
    "cost-type": { "cost-mode" : "numerical",
```

```
        "cost-metric" : "delay"
    },
    "endpoint-cost-map" : {
        "ipv4:192.0.2.2": {
            "ipv4:192.0.2.89"      : 10,
            "ipv4:198.51.100.34"   : 20,
            "ipv4:203.0.113.45"    : 30,
        }
    }
}
```

4. Metric: delayjitter

Cost Metric name: delayjitter

Metric Description: To specify the average delay variation over a configurable interval for each source/destination pair between two endpoints (network locations) in the network. It could be either end to end jitter or the jitter associated with a link (linkjitter). The unit is microseconds.

Cost Metric Value type:

A single 'JSONNumber' type 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.

Purpose: This is intended to be a constraint attribute value. It could be used as a cost metric constraint attribute used together with cost metric attribute 'routingcost' or on its own or as a returned cost metric in the response.

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

Measurement timing: Gather and update at the configurable interval if it is link attribute. See [OSPF-TE] for configurable interval. The configurable interval for end to end jitter could be same as link.

Measurement points with Potential Measurement Domain: The measurement point could be at any endpoint between source and destination in the network.

Examples:

POST /endpointcost/lookup HTTP/1.1

Host: alto.example.com

Content-Length: TBA

Content-Type: application/alto-endpointcostparams+json

Accept: application/alto-endpointcost+json,application/alto-error+json

```
{
  "cost-type": {"cost-mode" : "numerical",
               "cost-metric" : "delayjitter"},
  "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: TBA
Content-Type: application/alto-endpointcost+json
{
  "meta": {
    "cost type": {
      "cost-mode": "numerical",
      "cost-metric": "delayjitter"
    }
  },
  "endpoint-cost-map": {
    "ipv4:192.0.2.2": {
      "ipv4:192.0.2.89" : 0
      "ipv4:198.51.100.34" : 1
      "ipv4:203.0.113.45" : 5
    }
  }
}
```

5. Metric: pktloss

Cost Metric name: pktloss

Metric Description: To specify a percentage of the total traffic sent over a configurable interval for each source/destination pair between two endpoints(network locations) in the network. It could be either end to end packet loss or the packet loss associated with a link (linkloss).

Cost Metric 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.

Purpose: This is intended to be a constraint attribute value. It could be used as a cost metric constraint attribute used together with cost metric attribute 'routingcost' or on its own or as a returned cost metric in the response.

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

Measurement timing: Gather and update at the configurable interval if it is link attribute. See [OSPF-TE] for configurable interval. The configurable interval for end to end packet loss could be same as link.

Measurement points with Potential Measurement Domain: The measurement point could be at any endpoint between source and destination in the network.

Examples:

POST /endpointcost/lookup HTTP/1.1

Host: alto.example.com

Content-Length: TBA

Content-Type: application/alto-endpointcostparams+json

Accept: application/alto-endpointcost+json,application/alto-error+json

```
{
  "cost-type": {"cost-mode" : "numerical",
               "cost-metric" : "pktloss"},
  "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: TBA
Content-Type: application/alto-endpointcost+json
{
  "meta": {
    "cost type": {
      "cost-mode": "numerical",
      "cost-metric": "pktloss"
    }
  },
  "endpoint-cost-map": {
    "ipv4:192.0.2.2": {
      "ipv4:192.0.2.89" : 0,
      "ipv4:198.51.100.34": 1,
      "ipv4:203.0.113.45" : 0,
    }
  }
}
```

6. Cost Metric: hopcount

Cost Metric name: hopcount

Metric Description: To specify the number of hops in the path between the source endpoint and the destination endpoint.

Editor Note: Need to specify which layer (IP perhaps), details TBD for multiple-layer aspect.

Cost Metric Value type:

A single 'JSONNumber' type value containing an integer component that may be prefixed with an optional minus sign.

Purpose: This is intended to be a constraint attribute value. It could be used as a cost metric constraint attribute used together with cost metric attribute 'routingcost' or on its own or as a returned cost metric in the response.

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

7. Metrics: bandwidth

Cost Metric name: bandwidth

Metric Description: To specify Bandwidth over a configurable interval for each source/destination pair between two endpoints (network locations) in the network. It could be either aggregated bandwidth for end to end path or the bandwidth associated with a link. The units are bytes per second.

Cost Metric Value type:

A single 'JSONNumber' type 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.

Purpose: This is intended to be a constraint attribute value. It could be used as a cost metric constraint attribute used together with cost metric attribute 'routingcost' or on its own or as a returned cost metric in the response.

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

This is just a definition of the costtype 'bandwidth'. The use of this cost is always in conjunction with what it represents, which could be Max Bandwidth (maxbw), Residual Bandwidth (residuebw) etc.

Examples: (based on Residual Bandwidth (residuebw))

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

```
{
  "cost-type": {"cost-mode" : "numerical",
               "cost-metric" : "residuebw"},
  "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: TBA  
  Content-Type: application/alto-endpointcost+json  
  {  
    "meta": {  
      "cost type": {  
        "cost-mode": "numerical",  
        "cost-metric": "residualbw"  
      }  
    },  
    "endpoint-cost-map": {  
      "ipv4:192.0.2.2": {  
        "ipv4:192.0.2.89" : 0,  
        "ipv4:198.51.100.34": 2000,  
        "ipv4:203.0.113.45" : 5000,  
      }  
    }  
  }
```

8. Metric: maxbw

A maxbw is gathered using [RFC3630], [RFC3784] or [BGP-LS]. It could be either maximum bandwidth for end to end path or the bandwidth associated with a link. It is extended from Bandwidth Cost metric and defined as:

```
Object {  
  BWType      max;  
  [PIDName    srcPID;]  
  [PIDName    dstPID;]  
  [JSONBool   state;] //TRUE = not steady for src/dst pair; FALSE = steady;  
  Bandwidth   bw;  
}maxbw;
```

9. Maximum Reserved Bandwidth: maxreservbw

A maxreservbw is gathered using [RFC3630], [RFC3784] or [BGP-LS]. It could be either maximum reserved bandwidth for end to end path or the bandwidth associated with a link. It is extended from Bandwidth Cost metric and defined as:

```
Object {  
  BWType      maxreserved;  
  [PIDName    srcPID;]  
  [PIDName    dstPID;]  
  [JSONBool   state;] //TRUE = not steady for src/dst pair; FALSE = steady;  
  Bandwidth bw;  
}maxreservbw;
```


10. Metric: unreservbw

A unreservbw is gathered using [RFC3630], [RFC3784] or [BGP-LS]. It could be either unreserved bandwidth for end to end path or the bandwidth associated with a link. It is extended from Bandwidth Cost metric and defined as:

```
Object {  
  BWType unreserved;  
  [PIDName srcPID;]  
  [PIDName dstPID;]  
  [JSONBool state;] //TRUE = not steady for src/dst pair; FALSE = steady;  
  Bandwidth bw<1,8>  
}unreservbw;
```

//This bandwidth is per priority [TBD].

11. Metric: residuebw

A residuebw is gathered using [OSPF-TE], [ISIS-TE] or [BGP-PM]. It could be either residual bandwidth for end to end path or the bandwidth associated with a link. It is extended from Bandwidth Cost metric and defined as:

```
Object {  
  BWType Residue;  
  [PIDName srcPID;]  
  [PIDName dstPID;]  
  [JSONBool state;] //TRUE = not steady for src/dst pair; FALSE = steady;  
  Bandwidth bw;  
}residuebw;
```

12. Metric: availbw

A availbw is gathered using [OSPF-TE], [ISIS-TE] or [BGP-PM]. It could be either available bandwidth for end to end path or the bandwidth associated with a link. It is extended from Bandwidth Cost metric and defined as:

```
Object {  
  BWType Available;  
  [PIDName srcPID;]  
  [PIDName dstPID;]  
  [JSONBool state;] //TRUE = not steady for src/dst pair; FALSE = steady;  
  Bandwidth bw;  
}availbw;
```

13. Metric: utilbw

A utilbw is gathered using [OSPF-TE], [ISIS-TE] or [BGP-PM]. It could be either utilized bandwidth for end to end path or the bandwidth associated with a link. It is extended from Bandwidth Cost metric and defined as:

```
Object {  
  BWType Utilized;  
  [PIDName srcPID;]  
  [PIDName dstPID;]  
  [JSONBool state;] //TRUE = not steady for src/dst pair; FALSE = steady;  
  Bandwidth bw;  
}utilbw;
```

14. Security Considerations

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

15. IANA Considerations

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

Namespace	Property	Reference
	delay	[RFCxxxx], Section 3.1
	jitter	[RFCxxxx], Section 3.2
	pktloss	[RFCxxxx], Section 3.3
	bandwidth	[RFCxxxx], Section 3.4
	hopcount	[RFCxxxx], Section 3.5
	maxbw	[RFCxxxx], Section 3.6
	maxresbw	[RFCxxxx], Section 3.7
	unresdbw	[RFCxxxx], Section 3.8
	residbw	[RFCxxxx], Section 3.9
	availbw	[RFCxxxx], Section 3.10
	utilbw	[RFCxxxx], Section 3.11

16. References

16.1. Normative References

- [ALTO] Alimi, R., "ALTO Protocol", ID draft-ietf-alto-protocol-16, May 2013.
- [BGP-LS] Gredler, H., "North-Bound Distribution of Link-State and TE Information using BGP", ID draft-ietf-idr-ls-distribution-03, May 2013.
- [BGP-PM] Wu, Q., "BGP attribute for North-Bound Distribution of Traffic Engineering (TE) performance Metrics", ID draft-wu-idr-te-pm-bgp-02, October 2013.
- [ISIS-TE] Giacalone, S., "ISIS Traffic Engineering (TE) Metric Extensions", ID draft-ietf-isis-te-metric-extensions-01, October 2013.
- [OSPF-TE] 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.

16.2. Informative References

- [RFC6390] Clark, A. and B. Claise, "Framework for Performance Metric Development", RFC 6390, July 2011.

Appendix A. Filtering constraint Extensions

Section 10.2.2.3 of "ALTO: Application Layer Traffic Optimization Protocol" [I.D-ietf-alto-protocol] states:

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

object {
    PIDName srcls<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.

"

In the JSON Object of type ReqFilteredCostMap, the constraint attribute is expressed as:

```
"
[gt | lt | ge | le | eq ] <value>
"
```


In this specification, the constraint attribute is changed to

```
"  
<cost-type2> [gt | lt | ge | le | eq ] <value>  
"
```

Accordingly, the constraints definition is changed to:

```
"  
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 three entities separated by whitespace: (1)an cost type  
is by default cost-type in the JSON Object of type ReqFilteredCostMap.  
In addition, it could be another cost-type used for the returned cost  
(2) 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; (3) 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.  
"
```

Editor-Notes: Filtering constraint extension should move to another document defining multi-metrics filtering in the future.

Appendix B. Contributing Authors Addresses

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

Email: yry@cs.yale.edu

Authors' Addresses

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

Email: sunseawq@huawei.com

Young Lee
Huawei
1700 Alma Drive, Suite 500
Plano, TX 75075
USA

Email: leeyoung@huawei.com

Dhruv Dhody
Huawei
Leela Palace
Bangalore, Karnataka 560008
INDIA

Email: dhruv.ietf@gmail.com

Sabine Randriamasy
Alcatel-Lucent

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 R_1 to R_2 ... 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

