

ALTO WG
Internet-Draft
Intended status: Informational
Expires: September 2, 2022

LM. Contreras
Telefonica
D. Lachos
BENOCS
C. Rothenberg
Unicamp
S. Randriamasy
Nokia Bell Labs
March 1, 2022

Use of ALTO for Determining Service Edge
draft-contreras-alto-service-edge-04

Abstract

Service providers are starting to deploy and interconnect computing capabilities across the network for hosting network functions and applications. In distributed computing environments, both computing and topological information are necessary in order to determine the more convenient infrastructure where to deploy such a service or application. This document proposes an initial approach towards the use of ALTO to provide such information and assist the selection of appropriate deployment locations for services and applications.

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 <https://datatracker.ietf.org/drafts/current/>.

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

This Internet-Draft will expire on September 2, 2022.

Copyright Notice

Copyright (c) 2022 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 (<https://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. Computing needs	3
3. Usage of ALTO for determining where to deploy a function or application	4
3.1. Integrating compute information in ALTO	5
3.2. Association of compute capabilities to network topology	5
3.3. ALTO architecture for determining serve edge	6
4. Definition of flavors in ALTO property map	6
5. IANA Considerations	7
6. Security Considerations	7
7. Conclusions	7
8. References	8
8.1. Normative References	8
8.2. Informative References	8
Authors' Addresses	9

1. Introduction

The advent of virtualization is enabling the operators with a dynamic instantiation of network functions and applications by using different techniques on top of commoditized computation infrastructures, permitting a flexible and on-demand deployment of services, aligned with the actual needs observed as demanded by the customers.

Operators are starting to deploy distributed computing environments in different parts of the network with the objective of addressing the different service needs in terms of latency, bandwidth, processing capabilities, etc. This is translated in the emergence of a number of data centers of different sizes (e.g., large, medium, small) characterized by distinct dimension of CPUs, memory and storage capabilities, as well as bandwidth capacity for forwarding the traffic generated in and out the corresponding data center.

The probable future situation, with the generalization and proliferation of the edge computing approach, will increase the

potential footprint where a function or application can be deployed. These different dimensioning rules result in a different unitary cost per CPU, memory, and storage in each computing environment because of the scale.

All the available distributed computing capabilities can complicate the decision of what infrastructure use for instantiating a given function or application. Such a decision influences not only the resources that are consumed in a given computing environment, but also the network capacity of the path that connects such environment with the rest of the network from traffic source to destination.

It is then essential for a network operator to have mechanisms assisting on the decision by considering a number of constraints related to the function or application to be deployed understanding how a given decision on the computing environment for the service edge affects to the transport network substrate. This would allow to integrate network capabilities in the function placement decision and further optimize performance of the deployed application.

This document proposes the usage of ALTO [RFC7285] for assisting with such a decision.

2. Computing needs

A given network function or application typically shows certain requirements in terms of processing capabilities (i.e., CPU), as well as volatile memory (i.e., RAM) and storage capacity.

Cloud computing providers, such as Amazon Web Services or Microsoft Azure, typically structure their offerings of computing capabilities by bundling CPU, RAM and storage units as quotas, instances or flavors that can be consumed in an ephemeral or temporal fashion, during the actual lifetime of the required function or application.

This same approach is being taken nowadays for characterizing bundles of resources on the so-called Network Function Virtualization Infrastructure (NFVI) Points of Presence (PoPs) being deployed by the telco operators. Specifically, the Common Network Function Virtualisation Infrastructure Telecom Taskforce (CNTT) [CNTT], [GSMA], jointly hosted by GSMA and the Linux Foundation, is intending to harmonize the definition of above-mentioned computing capability instances or flavors for abstracting capabilities of the underlying NFVI facilitating a more efficient utilization of the infrastructure and simplifying the integration and certification of functions, where certification means the assessment of the expected behavior for a given function according to the level of resources determined by a given flavor.

Focusing on the CNTT ongoing work, the flavors or instances are characterized according to:

- o Type of instance (T): the types of instances are characterized as B (Basic), or N (Network Intensive). The latter can come with extensions for network acceleration for offloading network intensive operations to hardware.
- o Interface Option (I): it refers to the associated bandwidth of the network interface.
- o Compute flavor (F): it refers to a given predefined combination of resources in terms of virtual CPU, RAM, disk, and bandwidth for the management interface.
- o Optional storage extension (S): allows to request additional storage capacity.
- o Optional hardware acceleration characteristics (A): to request specific acceleration capabilities for improving the performance of the infrastructure.

The naming convention of an instance is thus encoded as T.I.F.S.A.

3. Usage of ALTO for determining where to deploy a function or application

ALTO can assist the deployment of a service or application on a specific flavor or instance of the computing substrate by taking into consideration network cost metrics.

A generic and primary approach is to take into account metrics related to the computing environment, such as availability of resources, unitary cost of those resources, etc.

Nevertheless, the function or application to be deployed on top of a given flavor is interconnected outside the computing environment where it is deployed, also requiring to guarantee transport network requirements to ensure the application performance, such as bandwidth, latency, etc.

The objective then is to leverage on ALTO to provide information about the more convenient execution environments to deploy virtualized network functions or applications, allowing the operator to get a coordinated service edge and transport network recommendation.

3.1. Integrating compute information in ALTO

CNTT proposes the existence of a catalogue of compute infrastructure profiles collecting the computing capability instances available to be consumed. Such kind of catalogue could be communicated to ALTO or even incorporated to it.

ALTO server queries are required to support T.I.F.S.A encoding in order to retrieve proper maps from ALTO. Additionally, filtered queries for particular characteristics of a flavor could also be supported.

3.2. Association of compute capabilities to network topology

It is required to associate the location of the available instances with topological information to allow ALTO construct the overall map. The expectation is to manage the network and cloud capabilities by the same entity, handling both network and compute abstractions jointly, producing an integrated map. While this can be straightforward when an ISP own both the network and the cloud infrastructure, it could require multiple administrative domains to interwork for composing the integrated map. Details on potential scenarios will be provided in future versions of the document.

At this stage four potential solutions could be considered:

- o To leverage on (and possibly extend) [I-D.ietf-teas-sf-aware-topo-model] for disseminating topology information together with notion of function location (that would require to be adapted to the existence of available compute capabilities). A recent effort in this direction can be found in [I-D.llc-teas-dc-aware-topo-model].
- o To extend BGP-LS [RFC7752], which is already considered as mechanism for feeding topology information in ALTO, in order to also advertise computing capabilities as well.
- o To combine information from the infrastructure profiles catalogue with topological information by leveraging on the IP prefixes allocated to the gateway providing connectivity to the NFVI PoP.
- o To integrate with Cloud Infrastructure Managers that could expose cloud infrastructure capabilities as in [CNTT], [GSMA].

The viability of these options will be explored in future versions of this document.

3.3. ALTO architecture for determining serve edge

The following logical architecture defines the usage of ALTO for determining service edges.

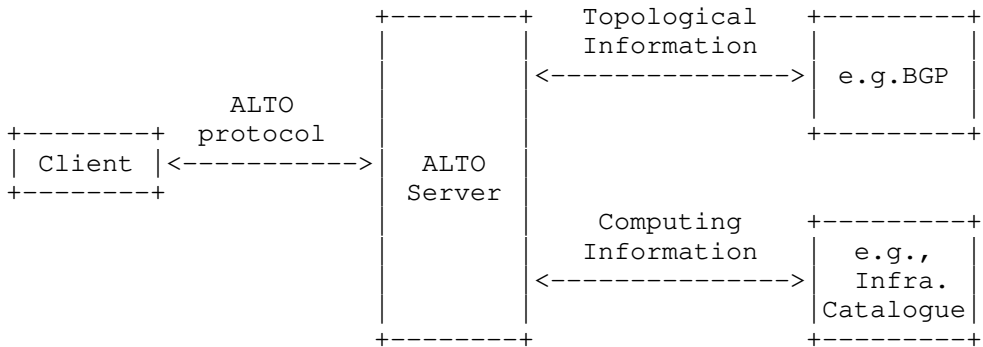


Figure 1: Service Edge Information Exchange

4. Definition of flavors in ALTO property map

The ALTO unified property extension [DRAFT-PM] generalizes the concept of endpoint properties to domains of other entities through property maps. In the context of the CNTT domain, an ALTO property map could be used to expose T.I.F.S.A information (i.e., potential candidate flavors) of available NFVI PoPs where an application or service can be deployed.

Table 1 below shows an illustrative example of an ALTO property map with property values grouped by flavor name.

Flavor Name	Type of instance (T)	Interface Option (I)	Compute flavor (F) {CPU, RAM, disk and bandwidth}	S.	A.
Small-1	Basic	{1, 2, 3, 4, 5, 6, 7, 8, 9 Gbps}	{1, 512 MB, 1 GB, 1 Gbps}
Small-2	Network Intensive	{1, 2, 3, 4, 5, 6, 7, 8, 9 Gbps}	{1, 512 MB, 1 GB, 1 Gbps}
Medium-1	Network Intensive	{25, 50, 75, 100, 125, 150 Gbps}	{2, 4 GB, 40 GB, 1 Gbps}
Large-1	Compute Intensive	{50, 100, 150, 200, 250, 300 Gbps}	{4, 8 GB, 80 GB, 1 Gbps}
Large-2	Compute Intensive	{100, 200, 300, 400, 500, 600 Gbps}	{8, 16 GB, 160 GB, 1 Gbps}
...

Table 1: ALTO Property Map

5. IANA Considerations

This document includes no request to IANA.

6. Security Considerations

TBD.

7. Conclusions

Telco networks will increasingly contain a number of interconnected data centers, of different size and characteristics, allowing flexibility in the dynamic deployment of functions and applications for advance services. The overall objective of this document is to begin a discussion in the ALTO WG regarding the suitability of the ALTO protocol for determining where to deploy a function or application in distributed computing environments. The result of such discussions will be reflected in future versions of this draft.

8. References

8.1. Normative References

- [RFC7285] Alimi, R., Ed., Penno, R., Ed., Yang, Y., Ed., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, DOI 10.17487/RFC7285, September 2014, <<https://www.rfc-editor.org/info/rfc7285>>.
- [RFC7752] Gredler, H., Ed., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and Traffic Engineering (TE) Information Using BGP", RFC 7752, DOI 10.17487/RFC7752, March 2016, <<https://www.rfc-editor.org/info/rfc7752>>.
- [RFC8189] Randriamasy, S., Roome, W., and N. Schwan, "Multi-Cost Application-Layer Traffic Optimization (ALTO)", RFC 8189, DOI 10.17487/RFC8189, October 2017, <<https://www.rfc-editor.org/info/rfc8189>>.

8.2. Informative References

- [CNTT] "Cloud iNfrastructure Telco Taskforce Reference Model", <<https://github.com/cntt-n/CNTT/wiki>>.
- [DRAFT-PM] Roome, W., Randriamasy, S., Yang, Y., Zhang, J., and K. Gao, "Unified Properties for the ALTO Protocol", draft-ietf-alto-unified-props-new-24 (work in progress), February 2022.
- [GSMA] "Cloud Infrastructure Reference Model, Version 2.0", October 2021, <<https://www.gsma.com/newsroom/wp-content/uploads//NG.126-v2.0-1.pdf>>.
- [I-D.ietf-teas-sf-aware-topo-model] Bryskin, I., Liu, X., Lee, Y., Guichard, J., Contreras, L., Ceccarelli, D., Tantsura, J., and D. Shytyi, "SF Aware TE Topology YANG Model", draft-ietf-teas-sf-aware-topo-model-09 (work in progress), February 2022.
- [I-D.llc-teas-dc-aware-topo-model] Lee, Y., Liu, X., and L. Contreras, "DC aware TE topology model", draft-llc-teas-dc-aware-topo-model-01 (work in progress), July 2021.

Authors' Addresses

Luis M. Contreras
Telefonica
Ronda de la Comunicacion, s/n
Madrid 28050
Spain

Email: luismiguel.contrerasmurillo@telefonica.com
URI: <http://lmcontreras.com/>

Danny Alex Lachos Perez
BENOCs GmbH
Reuchlinstrasse 10
Berlin 10553
Germany

Email: dlachos@benocs.com

Christian Esteve Rothenberg
University of Campinas
Av. Albert Einstein 400
Campinas, Sao Paulo 13083-970
Brazil

Email: chesteve@dca.fee.unicamp.br
URI: <https://intrig.dca.fee.unicamp.br/christian/>

Sabine Randriamasy
Nokia Bell Labs
Route de Villejust
Nozay 91460
France

Email: sabine.randriamasy@nokia-bell-labs.com

ALTO & CDNI WGs
Internet-Draft
Intended status: Standards Track
Expires: 21 August 2022

J. Seedorf
HFT Stuttgart - Univ. of Applied Sciences
Y. Yang
Yale University
K. Ma
Ericsson
J. Peterson
NeuStar
J. Zhang
Tongji University
17 February 2022

Content Delivery Network Interconnection (CDNI) Request Routing: CDNI
Footprint and Capabilities Advertisement using ALTO
draft-ietf-alto-cdni-request-routing-alto-22

Abstract

The Content Delivery Networks Interconnection (CDNI) framework in RFC 6707 defines a set of protocols to interconnect CDNs to achieve multiple goals, including extending the reach of a given CDN. A CDNI Request Routing Footprint & Capabilities Advertisement interface (FCI) is needed to achieve the goals of a CDNI. RFC 8008 defines the FCI semantics and provides guidelines on the FCI protocol, but the exact protocol is not specified. This document defines a new Application-Layer Traffic Optimization (ALTO) service, called "CDNI Advertisement Service", that provides an implementation of the FCI, following the guidelines defined in RFC 8008.

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 <https://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 21 August 2022.

Copyright Notice

Copyright (c) 2022 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Terminology and Background	4
2.1. Terminology	4
2.2. Semantics of FCI Advertisement	5
2.3. ALTO Background and Benefits	6
3. CDNI Advertisement Service	8
3.1. Media Type	9
3.2. HTTP Method	9
3.3. Accept Input Parameters	9
3.4. Capabilities	9
3.5. Uses	9
3.6. Response	9
3.7. Examples	12
3.7.1. IRD	12
3.7.2. A Basic Example	15
3.7.3. Incremental Updates	17
4. CDNI Advertisement Service using ALTO Network Map	18
4.1. Network Map Footprint Type: altopid	19
4.2. Examples	19
4.2.1. ALTO Network Map for CDNI Advertisements	19
4.2.2. ALTO PID Footprints in CDNI Advertisements	20
4.2.3. Incremental Updates	21
5. Filtered CDNI Advertisement using CDNI Capabilities	23
5.1. Media Type	23
5.2. HTTP Method	23
5.3. Accept Input Parameters	23
5.4. Capabilities	24
5.5. Uses	24
5.6. Response	24
5.7. Examples	25
5.7.1. A Basic Example	25
5.7.2. Incremental Updates	26

6.	Query Footprint Properties using ALTO Property Map Service	28
6.1.	Representing Footprint Objects as Property Map Entities	28
6.1.1.	ASN Domain	29
6.1.2.	COUNTRYCODE Domain	30
6.2.	Representing CDNI Capabilities as Property Map Entity Properties	30
6.2.1.	Defining Information Resource Media Type for Property Type cdni-capabilities	30
6.2.2.	Intended Semantics of Property Type cdni-capabilities	31
6.3.	Examples	31
6.3.1.	Property Map	31
6.3.2.	Filtered Property Map	32
6.3.3.	Incremental Updates	34
7.	IANA Considerations	35
7.1.	application/alto-cdni+json Media Type	35
7.2.	application/alto-cdnifilter+json Media Type	36
7.3.	CDNI Metadata Footprint Type Registry	38
7.4.	ALTO Entity Domain Type Registry	38
7.5.	ALTO Entity Property Type Registry	39
8.	Security Considerations	39
9.	References	40
9.1.	Normative References	41
9.2.	Informative References	42
	Acknowledgments	43
	Contributors	43
	Authors' Addresses	43

1. Introduction

The ability to interconnect multiple content delivery networks (CDNs) has many benefits, including increased coverage, capability, and reliability. The Content Delivery Networks Interconnection (CDNI) framework [RFC6707] defines four interfaces to interconnect CDNs: (1) the CDNI Request Routing Interface, (2) the CDNI Metadata Interface, (3) the CDNI Logging Interface, and (4) the CDNI Control Interface.

Among these four interfaces, the CDNI Request Routing Interface provides key functions, as specified in [RFC6707]: "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." At a high level, the scope of the CDNI Request Routing Interface, therefore, contains two main tasks: (1) determining if the dCDN (downstream CDN) is

willing to accept a delegated content request, and (2) redirecting the content request coming from a uCDN (upstream CDN) to the proper entry point or entity in the dCDN.

Correspondingly, the Request Routing Interface is broadly divided into two functionalities: (1) the CDNI Footprint & Capabilities Advertisement interface (FCI) defined in [RFC8008], and (2) the CDNI Request Routing Redirection interface (RI) defined in [RFC7975]. This document focuses on the first functionality (CDNI FCI).

Specifically, CDNI FCI allows both an advertisement from a dCDN to a uCDN (push) and a query from a uCDN to a dCDN (pull) so that the uCDN knows whether it can redirect a particular user request to that dCDN.

A key component in defining CDNI FCI is defining objects describing the footprints and capabilities of a dCDN. Such objects are already defined in Section 5 of [RFC8008]. However, no protocol is defined to transport and update such objects between a uCDN and a dCDN.

To define such a protocol, this document specifies an extension of the Application-Layer Traffic Optimization (ALTO) [RFC7285] protocol by introducing a new ALTO service called "CDNI Advertisement Service".

Section 2.3 discusses the benefits in using ALTO as a transport protocol.

2. Terminology and Background

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

The design of CDNI FCI transport using ALTO assumes an understanding of both FCI semantics and ALTO. Hence, this document starts with a non-normative review for both.

2.1. Terminology

The document uses the CDNI terms defined in [RFC6707], [RFC8006] and [RFC8008]. Also, the document uses the ALTO terms defined in [RFC7285] and [I-D.ietf-alto-unified-props-new]. This document uses the following abbreviations:

* ALTO: Application-Layer Traffic Optimization

- * ASN: Autonomous System Number
- * CDN: Content Delivery Network
- * CDNI: CDN Interconnection
- * dCDN: Downstream CDN
- * FCI: CDNI FCI, CDNI Request Routing Footprint & Capabilities Advertisement interface
- * IRD: Information Resource Directory in ALTO
- * PID: Provider-defined Identifier in ALTO
- * uCDN: Upstream CDN

2.2. Semantics of FCI Advertisement

[RFC8008] defines the semantics of CDNI FCI, provides guidance on what Footprint and Capabilities mean in a CDNI context, and specifies the requirements on the CDNI FCI transport protocol. The definitions in [RFC8008] depend on [RFC8006]. Below is a non-normative review of key related points of [RFC8008] and [RFC8006]. For detailed information and normative specification, the reader should refer to these two RFCs.

- * Multiple types of mandatory-to-implement footprints (i.e., ipv4cidr, ipv6cidr, asn, and countrycode) are defined in [RFC8006]. A "Set of IP-prefixes" can contain both full IP addresses (i.e., a /32 for IPv4 or a /128 for IPv6) and 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.
- * Multiple initial types of capabilities are defined in [RFC8008] including (1) Delivery Protocol, (2) Acquisition Protocol, (3) Redirection Mode, (4) Capabilities related to CDNI Logging, and (5) Capabilities related to CDNI Metadata. They are required in all cases and, therefore, considered as mandatory-to-implement capabilities for all CDNI FCI implementations.
- * Footprint and capabilities are defined together and cannot be interpreted independently from each other. Specifically, [RFC8008] integrates footprint and capabilities with an approach of "capabilities with footprint restrictions", by expressing capabilities on a per footprint basis.

- * Specifically, for all 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 Autonomous System Numbers (ASNs), 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. Different types of footprint constraints can be combined together to narrow the dCDN candidacy, i.e., the uCDN should consider the dCDN a candidate only if the request routing source satisfies all the types of footprint constraints in the advertisement.
- * Given that a large part of Footprint and Capabilities Advertisement may happen in contractual agreements, the semantics of CDNI Footprint and Capabilities advertisement refers 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 FCI. 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 that it has previously agreed to serve in a contract with a uCDN. Hence, server push and incremental encoding will be necessary techniques.

2.3. ALTO Background and Benefits

Application-Layer Traffic Optimization (ALTO) [RFC7285] defines an approach for conveying network layer (topology) information to "guide" the resource provider selection process in distributed applications that can choose among several candidate resources providers to retrieve a given resource. Usually, it is assumed that an ALTO server conveys information that these applications cannot measure or have difficulty measuring themselves [RFC5693].

Originally, ALTO was motivated by optimizing cross-ISP traffic generated by P2P applications [RFC5693]. However, ALTO can also be used for improving the request routing in CDNs. In particular, Section 5 of [RFC7971] explicitly mentions ALTO as a candidate protocol to improve the selection of a CDN surrogate or origin.

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

- * **Application Layer-oriented:** 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. This matches the need of CDNI, where a uCDN wants to improve application layer CDN request routing by using information (provided by a dCDN) that the uCDN could not easily obtain otherwise. Hence, ALTO can help a uCDN to select a proper dCDN by first providing dCDNs' capabilities as well as footprints (see Section 3) and then providing costs of surrogates in a dCDN by ALTO cost maps.
- * **Security:** The identification between uCDNs and dCDNs is an important requirement (see Section 8). ALTO maps can be signed and hence provide inherent origin protection. Please see Section 15.1.2 of [RFC7285] for detailed protection strategies.
- * **RESTful design:** The ALTO protocol has undergone extensive revisions in order to provide a RESTful design regarding the client-server interaction specified by the protocol. It is flexible and extensible enough to handle existing and potential future data formats defined by CDNI. It can provide the consistent client-server interaction model for other existing CDNI interfaces or potential future extensions and therefore reduce the learning cost for both users and developers, although they are not in the scope of this document. A CDNI FCI interface based on ALTO would inherit this RESTful design. Please see Section 3.
- * **Error-handling:** The ALTO protocol provides extensive error-handling in the whole request and response process (see Section 8.5 of [RFC7285]). A CDNI FCI interface based on ALTO would inherit this extensive error-handling framework. Please see Section 5.
- * **Map Service:** The semantics of an ALTO network map is an exact match for the needed information to convey a footprint by a dCDN, in particular, if such a footprint is being expressed by IP-prefix ranges. Please see Section 4.
- * **Filtered Map Service:** The ALTO map filtering service would allow a uCDN to query only for parts of an ALTO map. For example, the ALTO filtered property map service can enable a uCDN to query properties of a part of footprints efficiently. Please see Section 6.

- * Server-initiated notifications and incremental updates: When the footprint or the capabilities of a dCDN change (i.e., unexpectedly from the perspective of a uCDN), server-initiated notifications would enable a dCDN to inform a uCDN about such changes directly. 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 recent network and cost map from the dCDN, and therefore redirect requests to the 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. The newest design of ALTO supports server pushed incremental updates [RFC8895].
- * 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). ALTO Entity Property Map [I-D.ietf-alto-unified-props-new] would enable a dCDN to make such information available to a uCDN. This would enable a uCDN to access whether a dCDN has the capabilities for a given type of content requested.
- * 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 a uCDN for optimized dCDN 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 ALTO Entity Property Map [I-D.ietf-alto-unified-props-new], it would enable more sophisticated means for dCDN selection with ALTO. ALTO Path Vector Extension [I-D.ietf-alto-path-vector] is designed to allow ALTO clients to query information such as capacity regions for a given set of flows.

3. CDNI Advertisement Service

The ALTO protocol relies upon the ALTO Information Service framework which consists of multiple services. All ALTO services are "provided through a common transport protocol, messaging structure and encoding, and transaction model" [RFC7285]. The ALTO protocol specification defines multiple initial services, e.g., the ALTO network map service and cost map service.

This document defines a new ALTO service, called "CDNI Advertisement Service", which conveys JSON [RFC8259] objects of media type "application/alto-cdni+json". These JSON objects are used to transport BaseAdvertisementObject objects defined in [RFC8008]. This document specifies how to transport such BaseAdvertisementObject objects via the ALTO protocol with the ALTO "CDNI Advertisement Service". Similar to other ALTO services, this document defines the ALTO information resource for the "CDNI Advertisement Service" as follows.

Note that the encoding of BaseAdvertisementObject reuses the one defined in [RFC8008] and therefore also follows the recommendations of I-JSON (Internet JSON) [RFC7493], which is required by [RFC8008].

3.1. Media Type

The media type of the CDNI Advertisement resource is "application/alto-cdni+json" (see Section 7).

3.2. HTTP Method

A CDNI Advertisement resource is requested using the HTTP GET method.

3.3. Accept Input Parameters

There are no applicable Accept Input parameters.

3.4. Capabilities

There are no applicable capabilities.

3.5. Uses

The "uses" field MUST NOT appear unless the CDNI Advertisement resource depends on other ALTO information resources. If the CDNI Advertisement resource has dependent resources, the resource IDs of its dependent resources MUST be included into the "uses" field. This document only defines one potential dependent resource for the CDNI Advertisement resource. See Section 4 for details of when and how to use it. Future documents may extend the CDNI Advertisement resource and allow other dependent resources.

3.6. Response

The "meta" field of a CDNI Advertisement response MUST include the "vtag" field defined in Section 10.3 of [RFC7285]. This field provides the version of the retrieved CDNI FCI resource.

If a CDNI Advertisement response depends on other ALTO information resources, it MUST include the "dependent-vtags" field, whose value is an array to indicate the version tags of the resources used, where each resource is specified in "uses" of its Information Resource Directory (IRD) entry.

The data component of an ALTO CDNI Advertisement response is named "cdni-advertisement", which is a JSON object of type CDNIAdvertisementData:

```
object {  
  CDNIAdvertisementData cdni-advertisement;  
} InfoResourceCDNIAdvertisement : ResponseEntityBase;  
  
object {  
  BaseAdvertisementObject capabilities-with-footprints<0..*>;  
} CDNIAdvertisementData;
```

Specifically, a CDNIAdvertisementData object is a JSON object that includes only one property named "capabilities-with-footprints", whose value is an array of BaseAdvertisementObject objects. It provides capabilities with footprint restrictions for uCDN to decide the dCDN selection. If the value of this property is an empty array, it means the corresponding dCDN cannot provide any mandatory-to-implement CDNI capabilities for any footprints.

The syntax and semantics of BaseAdvertisementObject are well defined in Section 5.1 of [RFC8008]. A BaseAdvertisementObject object includes multiple properties, including capability-type, capability-value, and footprints, where footprints are defined in Section 4.2.2.2 of [RFC8006].

To be self-contained, below is an equivalent specification of BaseAdvertisementObject described in the ALTO-style notation (see Section 8.2 of [RFC7285]). As mentioned above, the normative specification of BaseAdvertisementObject is in [RFC8008].

```
object {  
  JSONString capability-type;  
  JSONValue capability-value;  
  Footprint footprints<0..*>;  
} BaseAdvertisementObject;  
  
object {  
  JSONString footprint-type;  
  JSONString footprint-value<1..*>;  
} Footprint;
```

For each BaseAdvertisementObject, the ALTO client MUST interpret footprints appearing multiple times as if they appeared only once. If footprints in a BaseAdvertisementObject is null or empty or not appearing, the ALTO client MUST understand that the capabilities in this BaseAdvertisementObject have the "global" coverage, i.e., the corresponding dCDN can provide them for any request routing source.

Note: Further optimization of BaseAdvertisement objects to effectively provide the advertisement of capabilities with footprint restrictions is certainly possible. For example, these two examples below both describe that the dCDN can provide capabilities ["http/1.1", "https/1.1"] for the same footprints. However, the latter one is smaller in its size.

EXAMPLE 1

```
{
  "meta": {...},
  "cdni-advertisement": {
    "capabilities-with-footprints": [
      {
        "capability-type": "FCI.DeliveryProtocol",
        "capability-value": {
          "delivery-protocols": [
            "http/1.1"
          ]
        },
        "footprints": [
          <Footprint objects>
        ]
      },
      {
        "capability-type": "FCI.DeliveryProtocol",
        "capability-value": {
          "delivery-protocols": [
            "https/1.1"
          ]
        },
        "footprints": [
          <Footprint objects>
        ]
      }
    ]
  }
}
```

EXAMPLE 2

```
{
```

```
"meta": {...},
"cdni-advertisement": {
  "capabilities-with-footprints": [
    {
      "capability-type": "FCI.DeliveryProtocol",
      "capability-value": {
        "delivery-protocols": [
          "https/1.1",
          "http/1.1"
        ]
      },
      "footprints": [
        <Footprint objects>
      ]
    }
  ]
}
```

Since such optimizations are not required for the basic interconnection of CDNs, the specifics of such mechanisms are outside the scope of this document.

This document only requires the ALTO server to provide the initial FCI-specific CDNI Payload Types defined in [RFC8008] as the mandatory-to-implement CDNI capabilities.

3.7. Examples

3.7.1. IRD

Below is the IRD of a simple, example ALTO server. The server provides both base ALTO information resources (e.g., network maps) and CDNI FCI related information resources (e.g., CDNI Advertisement resources), demonstrating a single, integrated environment.

Specifically, the IRD announces nine information resources as follows:

- * two network maps,
- * one CDNI Advertisement resource without dependency,
- * one CDNI Advertisement resource depending on a network map,
- * one filtered CDNI Advertisement resource to be defined in Section 5,

- * one property map including "cdni-capabilities" as its entity property,
- * one filtered property map including "cdni-capabilities" and "pid" as its entity properties, and
- * two update stream services:
 - one for updating CDNI Advertisement resources,
 - one for updating property maps

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

```
HTTP/1.1 200 OK
Content-Length: 3531
Content-Type: application/alto-directory+json
```

```
{
  "meta": {
    "default-alto-network-map": "my-default-network-map"
  },
  "resources": {
    "my-default-network-map": {
      "uri": "https://alto.example.com/networkmap",
      "media-type": "application/alto-networkmap+json"
    },
    "my-eu-netmap": {
      "uri": "https://alto.example.com/myeunetmap",
      "media-type": "application/alto-networkmap+json"
    },
    "my-default-cdnifci": {
      "uri": "https://alto.example.com/cdnifci",
      "media-type": "application/alto-cdni+json"
    },
    "my-cdnifci-with-pid-footprints": {
      "uri": "https://alto.example.com/networkcdnifci",
      "media-type": "application/alto-cdni+json",
      "uses": [ "my-eu-netmap" ]
    },
    "my-filtered-cdnifci": {
      "uri": "https://alto.example.com/cdnifci/filtered",
      "media-type": "application/alto-cdni+json",
      "accepts": "application/alto-cdnifilter+json"
    },
    "cdnifci-property-map": {
```

```
"uri": "https://alto.example.com/propmap/full/cdnifci",
"media-type": "application/alto-propmap+json",
"uses": [ "my-default-cdni" ],
"capabilities": {
  "mappings": {
    "ipv4": [ "my-default-cdni.cdni-capabilities" ],
    "ipv6": [ "my-default-cdni.cdni-capabilities" ],
    "countrycode": [
      "my-default-cdni.cdni-capabilities" ],
    "asn": [ "my-default-cdni.cdni-capabilities" ]
  }
}
},
"filtered-cdnifci-property-map": {
  "uri": "https://alto.example.com/propmap/lookup/cdnifci-pid",
  "media-type": "application/alto-propmap+json",
  "accepts": "application/alto-propmapparams+json",
  "uses": [ "my-default-cdni", "my-default-network-map" ],
  "capabilities": {
    "mappings": {
      "ipv4": [ "my-default-cdni.cdni-capabilities",
        "my-default-network-map.pid" ],
      "ipv6": [ "my-default-cdni.cdni-capabilities",
        "my-default-network-map.pid" ],
      "countrycode": [
        "my-default-cdni.cdni-capabilities" ],
      "asn": [ "my-default-cdni.cdni-capabilities" ]
    }
  }
},
"update-my-cdni-fci": {
  "uri": "https://alto.example.com/updates/cdnifci",
  "media-type": "text/event-stream",
  "accepts": "application/alto-updatestreamparams+json",
  "uses": [
    "my-default-network-map",
    "my-eu-netmap",
    "my-default-cdnifci",
    "my-filtered-cdnifci",
    "my-cdnifci-with-pid-footprints"
  ],
  "capabilities": {
    "incremental-change-media-types": {
      "my-default-network-map": "application/json-patch+json",
      "my-eu-netmap": "application/json-patch+json",
      "my-default-cdnifci":
        "application/merge-patch+json,application/json-patch+json",
      "my-filtered-cdnifci":
```

```

        "application/merge-patch+json,application/json-patch+json",
        "my-cdnifci-with-pid-footprints":
        "application/merge-patch+json,application/json-patch+json"
    }
}
},
"update-my-props": {
    "uri": "https://alto.example.com/updates/properties",
    "media-type": "text/event-stream",
    "uses": [
        "cdnifci-property-map",
        "filtered-cdnifci-property-map"
    ],
    "capabilities": {
        "incremental-change-media-types": {
            "cdnifci-property-map":
            "application/merge-patch+json,application/json-patch+json",
            "filtered-cdnifci-property-map":
            "application/merge-patch+json,application/json-patch+json"
        }
    }
}
}
}
}

```

3.7.2. A Basic Example

This basic example demonstrates a simple CDNI Advertisement resource, which does not depend on other resources. There are three BaseAdvertisementObjects in this resource and these objects' capabilities are http/1.1 delivery protocol, [http/1.1, https/1.1] delivery protocol, and https/1.1 acquisition protocol, respectively.

```

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

```

```

HTTP/1.1 200 OK
Content-Length: 1411
Content-Type: application/alto-cdni+json

```

```

{
  "meta": {
    "vtag": {
      "resource-id": "my-default-cdnifci",
      "tag": "da65eca2eb7a10ce8b059740b0b2e3f8eb1d4785"
    }
  },
}

```



```
"cdni-advertisement": {
  "capabilities-with-footprints": [
    {
      "capability-type": "FCI.DeliveryProtocol",
      "capability-value": {
        "delivery-protocols": [
          "http/1.1"
        ]
      },
      "footprints": [
        {
          "footprint-type": "ipv4cidr",
          "footprint-value": [ "192.0.2.0/24" ]
        },
        {
          "footprint-type": "ipv6cidr",
          "footprint-value": [ "2001:db8::/32" ]
        }
      ]
    },
    {
      "capability-type": "FCI.DeliveryProtocol",
      "capability-value": {
        "delivery-protocols": [
          "https/1.1",
          "http/1.1"
        ]
      },
      "footprints": [
        {
          "footprint-type": "ipv4cidr",
          "footprint-value": [ "198.51.100.0/24" ]
        }
      ]
    },
    {
      "capability-type": "FCI.AcquisitionProtocol",
      "capability-value": {
        "acquisition-protocols": [
          "https/1.1"
        ]
      },
      "footprints": [
        {
          "footprint-type": "ipv4cidr",
          "footprint-value": [ "203.0.113.0/24" ]
        }
      ]
    }
  ]
}
```

```

    }
  ]
}

```

3.7.3. Incremental Updates

A benefit of using ALTO to provide CDNI Advertisement resources is that such resources can be updated using ALTO incremental updates [RFC8895]. Below is an example that also shows the benefit of having both JSON merge patch and JSON patch to encode updates.

At first, an ALTO client requests updates for "my-default-cdnifci", and the ALTO server returns the "control-uri" followed by the full CDNI Advertisement response. Then when there is a change in the delivery-protocols in that http/1.1 is removed (from [http/1.1, https/1.1] to only https/1.1) due to maintenance of the http/1.1 clusters, the ALTO server regenerates the new CDNI Advertisement resource and pushes the full replacement to the ALTO client. Later on, the ALTO server notifies the ALTO client that "192.0.2.0/24" is added into the "ipv4" footprint object for delivery-protocol https/1.1 by sending the change encoded by JSON patch to the ALTO client.

```

POST /updates/cdnifci HTTP/1.1
Host: alto.example.com
Accept: text/event-stream,application/alto-error+json
Content-Type: application/alto-updatestreamparams+json
Content-Length: 94

```

```

{
  "add": {
    "my-cdnifci-stream": {
      "resource-id": "my-default-cdnifci"
    }
  }
}

```

```

HTTP/1.1 200 OK
Connection: keep-alive
Content-Type: text/event-stream

```

```

event: application/alto-updatestreamcontrol+json
data: {"control-uri":
data: "https://alto.example.com/updates/streams/3141592653589"}

```

```

event: application/alto-cdni+json,my-cdnifci-stream
data: { ... full CDNI Advertisement resource ... }

```

```

event: application/alto-cdni+json,my-cdnifci-stream
data: {
  data: "meta": {
    data: "vtag": {
      data: "tag": "dasdfa10ce8b059740bddsfasd8eb1d47853716"
    data: }
    data: },
    data: "cdni-advertisement": {
      data: "capabilities-with-footprints": [
        data: {
          data: "capability-type": "FCI.DeliveryProtocol",
          data: "capability-value": {
            data: "delivery-protocols": [
              data: "https/1.1"
            data: ]
          data: },
          data: "footprints": [
            data: { "footprint-type": "ipv4cidr",
              data: "footprint-value": [ "203.0.113.0/24" ]
            data: }
          data: ]
        data: },
        data: { ... other CDNI advertisement object ... }
      data: ]
    data: }
  data: }

```

```

event: application/json-patch+json,my-cdnifci-stream
data: [
  data: { "op": "replace",
    data: "path": "/meta/vtag/tag",
    data: "value": "a10ce8b059740b0b2e3f8eb1d4785acd42231bfe"
  data: },
  data: { "op": "add",
    data: "path": "/cdni-advertisement/capabilities-with-footprints
/0/footprints/0/footprint-value/-",
    data: "value": "192.0.2.0/24"
  data: }
data: ]

```

4. CDNI Advertisement Service using ALTO Network Map

4.1. Network Map Footprint Type: altopid

The ALTO protocol defines a concept called Provider-defined Identifier (PID) to represent a group of IPv4 or IPv6 addresses which can be applied the same management policy. The PID is an alternative to the pre-defined CDNI footprint types (i.e., ipv4cidr, ipv6cidr, asn, and countrycode).

To leverage this concept, this document defines a new CDNI Footprint Type called "altopid". A CDNI Advertisement resource can depend on an ALTO network map resource and use "altopid" footprints to compress its CDNI Footprint Payload.

Specifically, the "altopid" footprint type indicates that the corresponding footprint value is a list of PIDNames as defined in [RFC7285]. These PIDNames are references of PIDs in a network map resource. Hence a CDNI Advertisement resource using "altopid" footprints depends on a network map. For such a CDNI Advertisement resource, the resource id of its dependent network map MUST be included in the "uses" field of its IRD entry, and the "dependent-vtags" field with a reference to this network map MUST be included in its response (see the example in Section 4.2.2).

4.2. Examples

The following examples use the same IRD given in Section 3.7.1.

4.2.1. ALTO Network Map for CDNI Advertisements

Below provides a sample network map whose resource id is "my-eu-netmap". This map is referenced by the CDNI Advertisement example in Section 4.2.2.

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

HTTP/1.1 200 OK
Content-Length: 344
Content-Type: application/alto-networkmap+json
```

```
{
  "meta": {
    "vtag": [
      { "resource-id": "my-eu-netmap",
        "tag": "3ee2cb7e8d63d9fab71b9b34cbf764436315542e"
      }
    ]
  },
  "network-map": {
    "south-france" : {
      "ipv4": [ "192.0.2.0/24", "198.51.100.0/25" ],
      "ipv6": [ "2001:db8::/32" ]
    },
    "germany": {
      "ipv4": [ "203.0.113.0/24" ]
    }
  }
}
```

4.2.2. ALTO PID Footprints in CDNI Advertisements

This example shows a CDNI Advertisement resource that depends on a network map described in Section 4.2.1.

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

HTTP/1.1 200 OK
Content-Length: 736
Content-Type: application/alto-cdni+json

{
  "meta": {
    "dependent-vtags": [
      {
        "resource-id": "my-eu-netmap",
        "tag": "3ee2cb7e8d63d9fab71b9b34cbf764436315542e"
      }
    ]
  },
  "cdn-advertisement": {
    "capabilities-with-footprints": [
      { "capability-type": "FCI.DeliveryProtocol",
        "capability-value": [ "https/1.1" ],
        "footprints": [
          { "footprint-type": "altopid",
            "footprint-value": [ "south-france" ]
          }
        ]
      },
      { "capability-type": "FCI.AcquisitionProtocol",
        "capability-value": [ "https/1.1" ],
        "footprints": [
          { "footprint-type": "altopid",
            "footprint-value": [ "germany", "south-france" ]
          }
        ]
      }
    ]
  }
}
```

4.2.3. Incremental Updates

In this example, the ALTO client is interested in changes of "my-cdnifci-with-pid-footprints" and its dependent network map "my-eu-netmap". Considering two changes, the first one is to change footprints of the https/1.1 delivery protocol capability, and the second one is to remove the "south-france" PID from the footprints of the https/1.1 acquisition protocol capability.

```
POST /updates/cdnifci HTTP/1.1
Host: alto.example.com
Accept: text/event-stream,application/alto-error+json
Content-Type: application/alto-updatestreamparams+json
Content-Length: 185

{
  "add": {
    "my-eu-netmap-stream": {
      "resource-id": "my-eu-netmap"
    },
    "my-netmap-cdnifci-stream": {
      "resource-id": "my-cdnifci-with-pid-footprints"
    }
  }
}

HTTP/1.1 200 OK
Connection: keep-alive
Content-Type: text/event-stream

event: application/alto-updatestreamcontrol+json
data: {"control-uri":
data: "https://alto.example.com/updates/streams/3141592653590"}

event: application/alto-networkmap+json,my-eu-netmap-stream
data: { ... full Network Map of my-eu-netmap ... }

event: application/alto-cdnifci+json,my-netmap-cdnifci-stream
data: { ... full CDNI Advertisement resource ... }

event: application/json-patch+json,my-netmap-cdnifci-stream
data: [
data:   { "op": "replace",
data:     "path": "/meta/vtag/tag",
data:     "value": "dasdfal0ce8b059740bddsfasd8ebld47853716"
data:   },
data:   { "op": "add",
data:     "path":
data:       "/cdni-advertisement/capabilities-with-footprints
data:       /0/footprints/0/footprint-value/-",
data:     "value": "germany"
data:   }
data: ]

event: application/json-patch+json,my-netmap-cdnifci-stream
data: [
data:   { "op": "replace",
```

```
data:      "path": "/meta/vtag/tag",
data:      "value": "a10ce8b059740b0b2e3f8eb1d4785acd42231bfe"
data:    },
data:    { "op": "remove",
data:      "path":
data:        "/cdni-advertisement/capabilities-with-footprints
/1/footprints/0/footprint-value/1"
data:    }
data:  ]
```

5. Filtered CDNI Advertisement using CDNI Capabilities

Sections 3 (Section 3) and 4 (Section 4) describe CDNI Advertisement Service which can be used to enable a uCDN to get capabilities with footprint restrictions from dCDNs. However, since always getting full CDNI Advertisement resources from dCDNs is inefficient, this document introduces a new service named "Filtered CDNI Advertisement Service", to allow a client to filter a CDNI Advertisement resource using a client-given set of CDNI capabilities. For each entry of the CDNI Advertisement response, an entry will only be returned to the client if it contains at least one of the client given CDNI capabilities. The relationship between a filtered CDNI Advertisement resource and a CDNI Advertisement resource is similar to the relationship between a filtered network/cost map and a network/cost map.

5.1. Media Type

A filtered CDNI Advertisement resource uses the same media type defined for the CDNI Advertisement resource in Section 3.1: "application/alto-cdni+json".

5.2. HTTP Method

A filtered CDNI Advertisement resource is requested using the HTTP POST method.

5.3. Accept Input Parameters

The input parameters for a filtered CDNI Advertisement resource 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-cdnifilter+json" which is a JSON object of type ReqFilteredCDNIAdvertisement, where:


```
object {  
  JSONString capability-type;  
  JSONValue capability-value;  
} CDNICapability;  
  
object {  
  CDNICapability cdni-capabilities<0..*>;  
} ReqFilteredCDNIAdvertisement;
```

with fields:

capability-type: The same as Base Advertisement Object's capability-type defined in Section 5.1 of [RFC8008].

capability-value: The same as Base Advertisement Object's capability-value defined in Section 5.1 of [RFC8008].

cdni-capabilities: A list of CDNI capabilities defined in Section 5.1 of [RFC8008] for which footprints are to be returned. If this list is empty, the ALTO server MUST interpret it as a request for the full CDNI Advertisement resource. The ALTO server MUST interpret entries appearing in this list multiple times as if they appeared only once. If the ALTO server does not define any footprints for a CDNI capability, it MUST omit this capability from the response.

5.4. Capabilities

There are no applicable capabilities.

5.5. Uses

Same to the "uses" field of the CDNI Advertisement resource (see Section 3.5).

5.6. Response

If the request is invalid, the response MUST indicate an error, using ALTO protocol error handling specified in Section 8.5 of [RFC7285].

Specifically, a filtered CDNI Advertisement request is invalid if:

- * the value of "capability-type" is null;
- * the value of "capability-value" is null;
- * the value of "capability-value" is inconsistent with "capability-type".

When a request is invalid, the ALTO server MUST return an "E_INVALID_FIELD_VALUE" error defined in Section 8.5.2 of [RFC7285], and the "value" field of the error message SHOULD indicate this CDNI capability.

The ALTO server returns a filtered CDNI Advertisement resource for a valid request. The format of a filtered CDNI Advertisement resource is the same as a full CDNI Advertisement resource (See Section 3.6.)

The returned filtered CDNI Advertisement resource MUST contain all the BaseAdvertisementObject objects satisfying the following condition: The CDNI capability object of each included BaseAdvertisementObject object MUST follow two constraints:

- * The "cdni-capabilities" field of the input includes a CDNI capability object X having the same capability type as it.
- * All the mandatory properties in its capability value is a superset of mandatory properties in capability value of X semantically.

See Section 5.7.1 for a concrete example.

The version tag included in the "vtag" field of the response MUST correspond to the full CDNI Advertisement resource from which the filtered CDNI Advertisement resource is provided. This ensures that a single, canonical version tag is used independently of any filtering that is requested by an ALTO client.

5.7. Examples

The following examples use the same IRD example as in Section 3.7.1.

5.7.1. A Basic Example

This example filters the full CDNI Advertisement resource in Section 3.7.2 by selecting only the http/1.1 delivery protocol capability. Only the second BaseAdvertisementObjects in the full resource will be returned because the second object's capability is http/1.1 and https/1.1 delivery protocols which is the superset of https/1.1 delivery protocol.

```
POST /cdnifci/filtered HTTP/1.1
Host: alto.example.com
Accept: application/alto-cdni+json
Content-Type: application/cdnifilter+json
Content-Length: 176

{
```

```
"cdni-capabilities": [
  {
    "capability-type": "FCI.DeliveryProtocol",
    "capability-value": {
      "delivery-protocols": [ "https/1.1" ]
    }
  }
]
}

HTTP/1.1 200 OK
Content-Length: 570
Content-Type: application/alto-cdni+json

{
  "meta": {
    "vtag": {
      "resource-id": "my-filtered-cdnifci",
      "tag": "da65eca2eb7a10ce8b059740b0b2e3f8eb1d4785"
    }
  },
  "cdni-advertisement": {
    "capabilities-with-footprints": [
      {
        "capability-type": "FCI.DeliveryProtocol",
        "capability-value": {
          "delivery-protocols": [
            "https/1.1",
            "http/1.1"
          ]
        },
        "footprints": [
          {
            "footprint-type": "ipv4cidr",
            "footprint-value": [ "198.51.100.0/24" ]
          }
        ]
      }
    ]
  }
}
```

5.7.2. Incremental Updates

In this example, the ALTO client only cares about the updates of one advertisement object for delivery protocol capability whose value includes "https/1.1". Thus, it adds its limitation of capabilities in "input" field of the POST request.

```
POST /updates/cdnifci HTTP/1.1
Host: fcialtoupdate.example.com
Accept: text/event-stream,application/alto-error+json
Content-Type: application/alto-updatestreamparams+json
Content-Length: 346
```

```
{
  "add": {
    "my-filtered-fci-stream": {
      "resource-id": "my-filtered-cdnifci",
      "input": {
        "cdni-capabilities": [
          {
            "capability-type": "FCI.DeliveryProtocol",
            "capability-value": {
              "delivery-protocols": [ "https/1.1" ]
            }
          }
        ]
      }
    }
  }
}
```

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Type: text/event-stream
```

```
event: application/alto-updatestreamcontrol+json
data: {"control-uri":
data: "https://alto.example.com/updates/streams/3141592653590"}
```

```
event: application/alto-cdni+json,my-filtered-fci-stream
data: { ... filtered CDNI Advertisement resource ... }
```

```
event: application/json-patch+json,my-filtered-fci-stream
data: [
data: {
data:   "op": "replace",
data:   "path": "/meta/vtag/tag",
data:   "value": "a10ce8b059740b0b2e3f8eb1d4785acd42231bfe"
data: },
data: { "op": "add",
data:   "path":
data:   "/cdni-advertisement/capabilities-with-footprints
data:   /0/footprints/0/footprint-value/-",
data:   "value": "192.0.2.0/24"
data: }
```

data:]

6. Query Footprint Properties using ALTO Property Map Service

Besides the requirement of retrieving footprints of given capabilities, another common requirement for uCDN is to query CDNI capabilities of given footprints.

Considering each footprint as an entity with properties including CDNI capabilities, a natural way to satisfy this requirement is to use the ALTO property map as defined in [I-D.ietf-alto-unified-props-new]. This section describes how ALTO clients look up properties for individual footprints. First, it describes how to represent footprint objects as entities in the ALTO property map. Then it describes how to represent footprint capabilities as entity properties in the ALTO property map. Finally, it provides examples of the full property map and the filtered property map supporting CDNI capabilities, and their incremental updates.

6.1. Representing Footprint Objects as Property Map Entities

A footprint object has two properties: footprint-type and footprint-value. A footprint-value is an array of footprint values conforming to the specification associated with the registered footprint type ("ipv4cidr", "ipv6cidr", "asn", "countrycode", and "altopid").

Considering each ALTO entity defined in [I-D.ietf-alto-unified-props-new] also has two properties: entity domain type and domain-specific identifier, a straightforward approach to represent a footprint as an ALTO entity is to represent its footprint-type as an entity domain type, and its footprint value as a domain-specific identifier.

Each existing footprint type can be represented as an entity domain type as follows:

- * According to [I-D.ietf-alto-unified-props-new], "ipv4" and "ipv6" are two predefined entity domain types, which can be used to represent "ipv4cidr" and "ipv6cidr" footprints respectively. Note that both "ipv4" and "ipv6" domains can include not only hierarchical addresses but also individual addresses. Therefore, a "ipv4cidr" or "ipv6cidr" footprint with the longest prefix can also be represented by an individual address entity. When the uCDN receives a property map with individual addresses in an "ipv4" or "ipv6" domain, it can translate them as corresponding "ipv4cidr" or "ipv6cidr" footprints with the longest prefix.

- * "pid" is also a predefined entity domain type, which can be used to represent "altopid" footprints. Note that "pid" is a resource-specific entity domain. To represent an "altopid" footprint, the specifying information resource of the corresponding "pid" entity domain MUST be the dependent network map used by the CDNI Advertisement resource providing this "altopid" footprint.
- * However, no existing entity domain type can represent "asn" and "countrycode" footprints. To represent footprint-type "asn" and "countrycode", this document registers two new entity domains in Section 7 in addition to the ones in [I-D.ietf-alto-unified-props-new].

Here is an example of representing a footprint object of "ipv4cidr" type as a set of "ipv4" entities in the ALTO property map. The representation of the footprint object of "ipv6cidr" type is similar.

```
{ "footprint-type": "ipv4cidr",  
  "footprint-value": ["192.0.2.0/24", "198.51.100.0/24"]  
} --> "ipv4:192.0.2.0/24", "ipv4:198.51.100.0/24"
```

And here is an example of corresponding footprint object of "ipv4cidr" type represented by an individual address in an "ipv4" domain in the ALTO property map. The translation of the entities in an "ipv6" domain is similar.

```
"ipv4:203.0.113.100" --> {  
  "footprint-type": "ipv4cidr",  
  "footprint-value": ["203.0.113.100/32"]  
}
```

6.1.1. ASN Domain

The ASN domain associates property values with Autonomous Systems in the Internet.

6.1.1.1. Entity Domain Type

The entity domain type of the ASN domain is "asn" (in lowercase).

6.1.1.2. Domain-Specific Entity Identifiers

The entity identifier of an entity in an ASN domain MUST be encoded as a string consisting of the characters "as" (in lowercase) followed by the ASN [RFC6793] as a decimal number without leading zeros.

6.1.1.3. Hierarchy and Inheritance

There is no hierarchy or inheritance for properties associated with ASN.

6.1.2. COUNTRYCODE Domain

The COUNTRYCODE domain associates property values with countries.

6.1.2.1. Entity Domain Type

The entity domain type of the COUNTRYCODE domain is "countrycode" (in lowercase).

6.1.2.2. Domain-Specific Entity Identifiers

The entity identifier of an entity in a COUNTRYCODE domain is encoded as an ISO 3166-1 alpha-2 code [ISO3166-1] in lowercase.

6.1.2.3. Hierarchy and Inheritance

There is no hierarchy or inheritance for properties associated with country codes.

6.2. Representing CDNI Capabilities as Property Map Entity Properties

This document defines a new entity property type called "cdni-capabilities". An ALTO server can provide a property map resource mapping the "cdni-capabilities" entity property type for a CDNI Advertisement resource that it provides to an "ipv4", "ipv6", "asn" or "countrycode" entity domain.

6.2.1. Defining Information Resource Media Type for Property Type cdni-capabilities

The entity property type "cdni-capabilities" allows defining resource-specific entity properties. When resource-specific entity properties are defined with entity property type "cdni-capabilities", the defining information resource for a "cdni-capabilities" property MUST be a CDNI Advertisement resource provided by the ALTO server. The media type of the defining information resource for a "cdni-capabilities" property is therefore:

application/alto-cdni+json

6.2.2. Intended Semantics of Property Type cdni-capabilities

A "cdni-capabilities" property for an entity is to indicate all the CDNI capabilities that a corresponding CDNI Advertisement resource provides for the footprint represented by this entity. Thus, the value of a "cdni-capabilities" property MUST be a JSON array. Each element in a "cdni-capabilities" property MUST be an JSON object as format of CDNICapability (see Section 5.3). The value of a "cdni-capabilities" property for an "ipv4", "ipv6", "asn", "countrycode" or "altopid" entity MUST include all the CDNICapability objects satisfying the following conditions: (1) they are provided by the defining CDNI Advertisement resource; and (2) the represented footprint object of this entity is in their footprint restrictions.

6.3. Examples

The following examples use the same IRD example given by Section 3.7.1.

6.3.1. Property Map

This example shows a full property map in which entities are footprints and entities' property is "cdni-capabilities".

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

HTTP/1.1 200 OK
Content-Length: 1522
Content-Type: application/alto-propmap+json
```

```
{
  "property-map": {
    "meta": {
      "dependent-vtags": [
        { "resource-id": "my-default-cdnifci",
          "tag": "7915dc0290c2705481c491a2b4ffbec482b3cf62" }
      ]
    },
    "countrycode:us": {
      "my-default-cdnifci.cdni-capabilities": [
        { "capability-type": "FCI.DeliveryProtocol",
          "capability-value": {
            "delivery-protocols": ["http/1.1"]} }
      ],
      "ipv4:192.0.2.0/24": {
        "my-default-cdnifci.cdni-capabilities": [
```



```

    { "capability-type": "FCI.DeliveryProtocol",
      "capability-value": {
        "delivery-protocols": ["http/1.1"]}
    },
    "ipv4:198.51.100.0/24": {
      "my-default-cdnifci.cdni-capabilities": [
        { "capability-type": "FCI.DeliveryProtocol",
          "capability-value": {
            "delivery-protocols": ["https/1.1", "http/1.1"]}
        }
      ],
    },
    "ipv4:203.0.113.0/24": {
      "my-default-cdnifci.cdni-capabilities": [
        { "capability-type": "FCI.AcquisitionProtocol",
          "capability-value": {
            "acquisition-protocols": ["http/1.1"]}
        }
      ],
    },
    "ipv6:2001:db8::/32": {
      "my-default-cdnifci.cdni-capabilities": [
        { "capability-type": "FCI.DeliveryProtocol",
          "capability-value": {
            "delivery-protocols": ["http/1.1"]}
        }
      ],
    },
    "asn:as64496": {
      "my-default-cdnifci.cdni-capabilities": [
        { "capability-type": "FCI.DeliveryProtocol",
          "capability-value": {
            "delivery-protocols": ["https/1.1", "http/1.1"]}
        }
      ],
    }
  }
}

```

6.3.2. Filtered Property Map

This example uses the filtered property map service to get "pid" and "cdni-capabilities" properties for two footprints "ipv4:192.0.2.0/24" and "ipv6:2001:db8::/32".

```
POST /propmap/lookup/cdnifci-pid HTTP/1.1
Host: alto.example.com
Content-Type: application/alto-propmapparams+json
Accept: application/alto-propmap+json,application/alto-error+json
Content-Length: 181
```

```
{
  "entities": [
    "ipv4:192.0.2.0/24",
    "ipv6:2001:db8::/32"
  ],
  "properties": [ "my-default-cdnifci.cdni-capabilities",
                  "my-default-networkmap.pid" ]
}
```

```
HTTP/1.1 200 OK
Content-Length: 796
Content-Type: application/alto-propmap+json
```

```
{
  "property-map": {
    "meta": {
      "dependent-vtags": [
        { "resource-id": "my-default-cdnifci",
          "tag": "7915dc0290c2705481c491a2b4ffbec482b3cf62" },
        { "resource-id": "my-default-networkmap",
          "tag": "7915dc0290c2705481c491a2b4ffbec482b3cf63" }
      ]
    },
    "ipv4:192.0.2.0/24": {
      "my-default-cdnifci.cdni-capabilities": [
        { "capability-type": "FCI.DeliveryProtocol",
          "capability-value": { "delivery-protocols": ["http/1.1"]} },
      ],
      "my-default-networkmap.pid": "pid1"
    },
    "ipv6:2001:db8::/32": {
      "my-default-cdnifci.cdni-capabilities": [
        { "capability-type": "FCI.DeliveryProtocol",
          "capability-value": { "delivery-protocols": ["http/1.1"]} },
      ],
      "my-default-networkmap.pid": "pid3"
    }
  }
}
```

6.3.3. Incremental Updates

In this example, the ALTO client is interested in updates for the properties "cdni-capabilities" and "pid" of two footprints "ipv4:192.0.2.0/24" and "countrycode:fr".

```
POST /updates/properties HTTP/1.1
Host: alto.example.com
Accept: text/event-stream,application/alto-error+json
Content-Type: application/alto-updatestreamparams+json
Content-Length: 339

{
  "add": {
    "fci-propmap-stream": {
      "resource-id": "filtered-cdnifci-property-map",
      "input": {
        "properties": [ "my-default-cdnifci.cdni-capabilities",
                        "my-default-networkmap.pid" ],
        "entities": [ "ipv4:192.0.2.0/24",
                      "ipv6:2001:db8::/32" ]
      }
    }
  }
}

HTTP/1.1 200 OK
Connection: keep-alive
Content-Type: text/event-stream

event: application/alto-updatestreamcontrol+json
data: {"control-uri":
data: "https://alto.example.com/updates/streams/1414213562373"}

event: application/alto-cdni+json,fci-propmap-stream
data: { ... filtered property map ... }

event: application/merge-patch+json,fci-propmap-stream
data: {
data:   "property-map": {
data:     "meta": {
data:       "dependent-vtags": [
data:         { "resource-id": "my-default-cdnifci",
data:           "tag": "2beeac8ee23c3ddle98a73fd30df80ece9fa5627"},
data:         { "resource-id": "my-default-networkmap",
data:           "tag": "7915dc0290c2705481c491a2b4ffbec482b3cf63"}
data:       ]
data:     },
data:   },
```

```

data:      "ipv4:192.0.2.0/24": {
data:      "my-default-cdnifci.cdni-capabilities": [
data:      { "capability-type": "FCI.DeliveryProtocol",
data:      "capability-value": {
data:      "delivery-protocols": ["http/1.1", "https/1.1"]}}}
data:      }
data:      }
data:      }

event: application/json-patch+json,fci-propmap-stream
data: [
data:  { "op": "replace",
data:      "path": "/meta/dependent-vtags/0/tag",
data:      "value": "61b23185a50dc7b334577507e8ff8c3b409e4"
data:  },
data:  { "op": "replace",
data:      "path":
data:      "/property-map/countrycode:fr/my-default-networkmap.pid",
data:      "value": "pid5"
data:  }
data: ]

```

7. IANA Considerations

This document defines two new media types: "application/alto-cdni+json", as described in Section 7.1, and "application/cdnifilter+json", as described in Section 7.2. It also defines a new CDNI metadata footprint type (Section 7.3), two new ALTO entity domain types (Section 7.4), and a new ALTO entity property type (Section 7.5).

7.1. application/alto-cdni+json Media Type

Type name:
application

Subtype name:
alto-cdni+json

Required parameters:
N/A

Optional parameters:
N/A

Encoding considerations:
Encoding considerations are identical to those specified for the "application/json" media type. See [RFC8259].

Security considerations:

Security considerations related to the generation and consumption of ALTO Protocol messages are discussed in Section 15 of [RFC7285].

Interoperability considerations:

N/A

Published specification:

Section 3 of RFCthis

Applications that use this media type:

ALTO servers and ALTO clients [RFC7285] either stand alone or are embedded within other applications that provides CDNI interfaces for uCDNs or dCDNs.

Fragment identifier considerations:

N/A

Additional information:

Magic number(s): N/A

File extension(s): N/A

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).

7.2. application/alto-cdnifilter+json Media Type

Type name:

application

Subtype name:

alto-cdnifilter+json

Required parameters:

N/A

Optional parameters:

N/A

Encoding considerations:

Encoding considerations are identical to those specified for the "application/json" media type. See [RFC8259].

Security considerations:

Security considerations related to the generation and consumption of ALTO Protocol messages are discussed in Section 15 of [RFC7285].

Interoperability considerations:

N/A

Published specification:

Section 5 of RFCthis

Applications that use this media type:

ALTO servers and ALTO clients [RFC7285] either stand alone or are embedded within other applications that provides CDNI interfaces for uCDNs or dCDNs and supports CDNI capability-based filtering.

Fragment identifier considerations:

N/A

Additional information:

Magic number(s): N/A

File extension(s): N/A

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).

7.3. CDNI Metadata Footprint Type Registry

This document updates the CDNI Metadata Footprint Types Registry created by Section 7.2 of [RFC8006]. A new footprint type is to be registered, listed in Table 1.

Footprint Type	Description	Specification
altopid	A list of PID names	Section 4 of RFCthis

Table 1: CDNI Metadata Footprint Type

[RFC Editor: Please replace RFCthis with the published RFC number for this document.]

7.4. ALTO Entity Domain Type Registry

This document updates the ALTO Entity Domain Type Registry created by Section 11.2 of [I-D.ietf-alto-unified-props-new]. Two new entity domain types are to be registered, listed in Table 2.

Identifier	Entity Address Encoding	Hierarchy & Inheritance	Media Type of Defining Resource	Mapping to ALTO Address Type
asn	See Section 6.1.1.2 of RFCthis	None	None	false
countrycode	See Section 6.1.2.2 of RFCthis	None	None	false

Table 2: Additional ALTO Entity Domain Types

[RFC Editor: Please replace RFCthis with the published RFC number for this document.]

7.5. ALTO Entity Property Type Registry

This document updates the ALTO Entity Property Type Registry created by Section 11.3 of [I-D.ietf-alto-unified-props-new]. A new entity property type is to be registered, listed in Table 3.

Identifier	Intended Semantics	Media Type of Defining Resource
cdni-capabilities	Section 6.2 of RFCthis	application/alto-cdni+json

Table 3: Additional ALTO Entity Property Type

[RFC Editor: Please replace RFCthis with the published RFC number for this document.]

8. Security Considerations

As an extension of the base ALTO protocol [RFC7285], this document fits into the architecture of the base protocol. And hence Security Considerations of the base protocol (Section 15 of [RFC7285]) fully apply when this extension is provided by an ALTO server.

In the context of CDNI Advertisement, the following security risk scenarios should be considered:

- * For authenticity and integrity of ALTO information, an attacker may disguise itself as an ALTO server for a dCDN (e.g., by starting a man-in-the-middle attack), and provide false capabilities and footprints to a uCDN using the CDNI Advertisement service. Such false information may lead a uCDN to (1) select an incorrect dCDN to serve user requests, or (2) skip uCDNs in good conditions. To address this risk, protection strategies in Section 15.1.2 of [RFC7285] can be applied.
- * For potential undesirable guidance from authenticated ALTO information, a dCDN can provide a uCDN with limited capabilities and smaller footprint coverage so that the dCDN can avoid transferring traffic for a uCDN which they should have to transfer. To reduce this risk, the protection strategies in Section 15.2.2 of [RFC7285] can be considered.
- * For confidentiality and privacy of ALTO information, footprint properties integrated with ALTO property maps may expose network location identifiers (e.g., IP addresses or fine-grained PIDs).

To address this risk, the protection strategy for risk types (1) and (3) as described in Section 15.3 of [RFC7285] can be considered.

- * For availability of ALTO services, an attacker may conduct service degradation attacks using services defined in this document to disable ALTO services of a network. It may request potentially large, full CDNI Advertisement resources from an ALTO server in a dCDN continuously, to consume the bandwidth resources of that ALTO server. It may also query filtered property map services with many smaller individual footprints, to consume the computation resources of the ALTO server. To mitigate these risks, the protection strategies in Section 15.5.2 of [RFC7285] can be applied.

Although protection strategies as described in Section 15 of [RFC7285] should be applied to address aforementioned security and privacy considerations, two special cases need to be included as follows:

- * As required by section 7 of [RFC8008],

"All protocols that implement these capabilities and footprint advertisement objects are REQUIRED to provide integrity and authentication services."

Therefore, the uCDN (ALTO Client) MUST be authenticated to the dCDN (ALTO Server). And the dCDN (ALTO Server) MUST support HTTP Digest Authentication and MAY also support TLS mutual authentication. The authentication method will need to be negotiated out of band and is out of scope for this document, as is the approach for provisioning and managing these credentials.

- * One specific information leakage risk introduced by this document could not be addressed by these strategies. In particular, if a dCDN signs agreements with multiple uCDNs without any isolation, this dCDN may disclose extra information of one uCDN to another one. In that case, one uCDN may redirect requests which should not have to be served by this dCDN to it.

To reduce the risk, a dCDN SHOULD isolate full/filtered CDNI Advertisement resources for different uCDNs. It could consider generating URIs of different full/filtered CDNI Advertisement resources by hashing its company ID, a uCDN's company ID as well as their agreements. A dCDN SHOULD avoid exposing all full/filtered CDNI Advertisement resources in one of its IRDs.

9. References

9.1. Normative References

- [I-D.ietf-alto-unified-props-new]
Roome, W., Randriamasy, S., Yang, Y. R., Zhang, J. J., and K. Gao, "An ALTO Extension: Entity Property Maps", Work in Progress, Internet-Draft, draft-ietf-alto-unified-props-new-22, 25 January 2022,
<<https://datatracker.ietf.org/doc/html/draft-ietf-alto-unified-props-new-22>>.
- [ISO3166-1]
ISO (International Organization for Standardization), ., "ISO 3166-1: Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes", 2020.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997,
<<https://www.rfc-editor.org/rfc/rfc2119>>.
- [RFC6793] Vohra, Q. and E. Chen, "BGP Support for Four-Octet Autonomous System (AS) Number Space", RFC 6793, DOI 10.17487/RFC6793, December 2012,
<<https://www.rfc-editor.org/rfc/rfc6793>>.
- [RFC7285] Alimi, R., Ed., Penno, R., Ed., Yang, Y., Ed., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, DOI 10.17487/RFC7285, September 2014,
<<https://www.rfc-editor.org/rfc/rfc7285>>.
- [RFC7493] Bray, T., Ed., "The I-JSON Message Format", RFC 7493, DOI 10.17487/RFC7493, March 2015,
<<https://www.rfc-editor.org/rfc/rfc7493>>.
- [RFC8006] Niven-Jenkins, B., Murray, R., Caulfield, M., and K. Ma, "Content Delivery Network Interconnection (CDNI) Metadata", RFC 8006, DOI 10.17487/RFC8006, December 2016,
<<https://www.rfc-editor.org/rfc/rfc8006>>.
- [RFC8008] Seedorf, J., Peterson, J., Previdi, S., van Brandenburg, R., and K. Ma, "Content Delivery Network Interconnection (CDNI) Request Routing: Footprint and Capabilities Semantics", RFC 8008, DOI 10.17487/RFC8008, December 2016,
<<https://www.rfc-editor.org/rfc/rfc8008>>.

- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/rfc/rfc8174>>.
- [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <<https://www.rfc-editor.org/rfc/rfc8259>>.
- [RFC8895] Roome, W. and Y. Yang, "Application-Layer Traffic Optimization (ALTO) Incremental Updates Using Server-Sent Events (SSE)", RFC 8895, DOI 10.17487/RFC8895, November 2020, <<https://www.rfc-editor.org/rfc/rfc8895>>.

9.2. Informative References

- [I-D.ietf-alto-path-vector] Gao, K., Lee, Y., Randriamasy, S., Yang, Y. R., and J. J. Zhang, "An ALTO Extension: Path Vector", Work in Progress, Internet-Draft, draft-ietf-alto-path-vector-21, 2 February 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-alto-path-vector-21>>.
- [RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, DOI 10.17487/RFC5693, October 2009, <<https://www.rfc-editor.org/rfc/rfc5693>>.
- [RFC6707] Niven-Jenkins, B., Le Faucheur, F., and N. Bitar, "Content Distribution Network Interconnection (CDNI) Problem Statement", RFC 6707, DOI 10.17487/RFC6707, September 2012, <<https://www.rfc-editor.org/rfc/rfc6707>>.
- [RFC7971] Stiemerling, M., Kiesel, S., Scharf, M., Seidel, H., and S. Previdi, "Application-Layer Traffic Optimization (ALTO) Deployment Considerations", RFC 7971, DOI 10.17487/RFC7971, October 2016, <<https://www.rfc-editor.org/rfc/rfc7971>>.
- [RFC7975] Niven-Jenkins, B., Ed. and R. van Brandenburg, Ed., "Request Routing Redirection Interface for Content Delivery Network (CDN) Interconnection", RFC 7975, DOI 10.17487/RFC7975, October 2016, <<https://www.rfc-editor.org/rfc/rfc7975>>.

Acknowledgments

The authors thank Matt Caulfield, Danny Alex Lachos Perez, Daryl Malas and Sanjay Mishra for their timely reviews and invaluable comments. Big thanks also to ALTO WG Chairs (Qin Wu and Vijay Gurbani), and all the directorate reviewers and IESG reviewers (Martin Duke, Erik Kline, Martin Vigoureux, Murray Kucherawy, Roman Danyliw, Zaheduzzaman Sarker, Eric Vyncke, and Francesca Palombini), for their thorough reviews, discussions, guidance and shepherding, that further improve this document.

Jan Seedorf has been partially supported by the GreenICN project (GreenICN: Architecture and Applications of Green Information Centric Networking), a research project supported jointly by the European Commission under its 7th Framework Program (contract no. 608518) and the National Institute of Information and Communications Technology (NICT) in Japan (contract no. 167). 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 GreenICN project, the European Commission, or NICT.

This document has also been supported by the Coordination Support Action entitled 'Supporting European Experts Presence in International Standardisation Activities in ICT' ("StandlCT.eu") funded by the European Commission under the Horizon 2020 Programme with Grant Agreement no. 780439. 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 European Commission.

Contributors

Xiao Shawn Lin
Huawei
2222 Newjinqiao Rd
Shanghai
200125
China
Phone: +86-15316812351
Email: x.shawn.lin@gmail.com

Authors' Addresses

Jan Seedorf
HFT Stuttgart - Univ. of Applied Sciences
Schellingstrasse 24
70174 Stuttgart
Germany
Phone: +49-0711-8926-2801
Email: jan.seedorf@hft-stuttgart.de

Y. Richard Yang
Yale University
51 Prospect Street
New Haven, CT 06511
United States of America
Phone: +1-203-432-6400
Email: yry@cs.yale.edu
URI: <http://www.cs.yale.edu/~yry/>

Kevin J. Ma
Ericsson
43 Nagog Park
Acton, MA 01720
United States of America
Phone: +1-978-844-5100
Email: kevin.j.ma.ietf@gmail.com

Jon Peterson
NeuStar
1800 Sutter St Suite 570
Concord, CA 94520
United States of America
Email: jon.peterson@neustar.biz

Jingxuan Jensen Zhang
Tongji University
4800 Cao'an Hwy
Shanghai
201804
China
Email: jingxuan.zhang@tongji.edu.cn

ALTO
Internet-Draft
Intended status: Experimental
Expires: 21 September 2022

K. Gao
Sichuan University
Y. Lee
Samsung
S. Randriamasy
Nokia Bell Labs
Y.R. Yang
Yale University
J. Zhang
Tongji University
20 March 2022

An ALTO Extension: Path Vector
draft-ietf-alto-path-vector-25

Abstract

This document is an extension to the base Application-Layer Traffic Optimization (ALTO) protocol. It extends the ALTO Cost Map and ALTO Property Map services so that an application can decide which endpoint(s) to connect based on not only numerical/ordinal cost values but also fine-grained abstract information of the paths. This is useful for applications whose performance is impacted by specified components of a network on the end-to-end paths, e.g., they may infer that several paths share common links and prevent traffic bottlenecks by avoiding such paths. This extension introduces a new abstraction called Abstract Network Element (ANE) to represent these components and encodes a network path as a vector of ANEs. Thus, it provides a more complete but still abstract graph representation of the underlying network(s) for informed traffic optimization among endpoints.

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 <https://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 21 September 2022.

Copyright Notice

Copyright (c) 2022 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	4
2. Requirements Languages	6
3. Terminology	6
4. Requirements and Use Cases	7
4.1. Design Requirements	7
4.2. Sample Use Cases	10
4.2.1. Exposing Network Bottlenecks	11
4.2.2. Resource Exposure for CDN and Service Edge	15
5. Path Vector Extension: Overview	17
5.1. Abstract Network Element (ANE)	18
5.1.1. ANE Entity Domain	19
5.1.2. Ephemeral and Persistent ANEs	19
5.1.3. Property Filtering	20
5.2. Path Vector Cost Type	20
5.3. Multipart Path Vector Response	21
5.3.1. Identifying the Media Type of the Root Object	22
5.3.2. References to Part Messages	22
6. Specification: Basic Data Types	23
6.1. ANE Name	23
6.2. ANE Entity Domain	23
6.2.1. Entity Domain Type	23
6.2.2. Domain-Specific Entity Identifier	23
6.2.3. Hierarchy and Inheritance	23
6.2.4. Media Type of Defining Resource	23
6.3. ANE Property Name	24
6.4. Initial ANE Property Types	24
6.4.1. Maximum Reservable Bandwidth	24
6.4.2. Persistent Entity ID	25
6.4.3. Examples	25
6.5. Path Vector Cost Type	26

6.5.1.	Cost Metric: ane-path	26
6.5.2.	Cost Mode: array	27
6.6.	Part Resource ID and Part Content ID	27
7.	Specification: Service Extensions	27
7.1.	Notations	27
7.2.	Multipart Filtered Cost Map for Path Vector	28
7.2.1.	Media Type	28
7.2.2.	HTTP Method	28
7.2.3.	Accept Input Parameters	28
7.2.4.	Capabilities	29
7.2.5.	Uses	30
7.2.6.	Response	30
7.3.	Multipart Endpoint Cost Service for Path Vector	34
7.3.1.	Media Type	34
7.3.2.	HTTP Method	34
7.3.3.	Accept Input Parameters	34
7.3.4.	Capabilities	35
7.3.5.	Uses	35
7.3.6.	Response	35
8.	Examples	39
8.1.	Sample Setup	39
8.2.	Information Resource Directory	39
8.3.	Multipart Filtered Cost Map	42
8.4.	Multipart Endpoint Cost Service Resource	43
8.5.	Incremental Updates	48
8.6.	Multi-cost	50
9.	Compatibility with Other ALTO Extensions	52
9.1.	Compatibility with Legacy ALTO Clients/Servers	53
9.2.	Compatibility with Multi-Cost Extension	53
9.3.	Compatibility with Incremental Update	53
9.4.	Compatibility with Cost Calendar	53
10.	General Discussions	54
10.1.	Constraint Tests for General Cost Types	54
10.2.	General Multi-Resource Query	54
11.	Security Considerations	55
12.	IANA Considerations	57
12.1.	ALTO Cost Metric Registry	57
12.2.	ALTO Cost Mode Registry	58
12.3.	ALTO Entity Domain Type Registry	58
12.4.	ALTO Entity Property Type Registry	59
12.4.1.	New ANE Property Type: Maximum Reservable Bandwidth	59
12.4.2.	New ANE Property Type: Persistent Entity ID	60
13.	References	60
13.1.	Normative References	60
13.2.	Informative References	61
Appendix A.	Acknowledgments	64
Appendix B.	Revision Logs (To be removed before publication)	64

B.1.	Changes since -20	64
B.2.	Changes since -19	65
B.3.	Changes since -18	65
B.4.	Changes since -17	65
B.5.	Changes since -16	65
B.6.	Changes since -15	65
B.7.	Changes since -14	65
B.8.	Changes since -13	66
B.9.	Changes since -12	66
B.10.	Changes since -11	66
B.11.	Changes since -10	66
B.12.	Changes since -09	67
B.13.	Changes since -08	67
B.14.	Changes Since Version -06	67
Authors' Addresses	68

1. Introduction

Network performance metrics are crucial to assess the Quality of Experience (QoE) of applications. The ALTO protocol allows Internet Service Providers (ISPs) to provide guidance, such as topological distance between different end hosts, to overlay applications. Thus, the overlay applications can potentially improve the perceived QoE by better orchestrating their traffic to utilize the resources in the underlying network infrastructure.

Existing ALTO Cost Map (Section 11.2.3 of [RFC7285]) and Endpoint Cost Service (Section 11.5 of [RFC7285]) provide only cost information on an end-to-end path defined by its <source, destination> endpoints: The base protocol [RFC7285] allows the services to expose the topological distances of end-to-end paths, while various extensions have been proposed to extend the capability of these services, e.g., to express other performance metrics [I-D.ietf-alto-performance-metrics], to query multiple costs simultaneously [RFC8189], and to obtain the time-varying values [RFC8896].

While the existing extensions are sufficient for many overlay applications, the QoE of some overlay applications depends not only on the cost information of end-to-end paths, but also on particular components of a network on the paths and their properties. For example, job completion time, which is an important QoE metric for a large-scale data analytics application, is impacted by shared bottleneck links inside the carrier network as link capacity may impact the rate of data input/output to the job. We refer to such components of a network as Abstract Network Elements (ANE).

Predicting such information can be very complex without the help of ISPs, for example, [BOXOPT] has shown that finding the optimal bandwidth reservation for multiple flows can be NP-hard without further information than whether a reservation succeeds. With proper guidance from the ISP, an overlay application may be able to schedule its traffic for better QoE. In the meantime, it may be helpful as well for ISPs if applications could avoid using bottlenecks or challenging the network with poorly scheduled traffic.

Despite the claimed benefits, ISPs are not likely to expose raw details on their network paths: first for the sake of topology hiding requirement, second because it may increase volume and computation overhead, and last because applications do not necessarily need all the network path details and are likely not able to understand them.

Therefore, it is beneficial for both ISPs and applications if an ALTO server provides ALTO clients with an "abstract network state" that provides the necessary information to applications, while hiding the network complexity and confidential information. An "abstract network state" is a selected set of abstract representations of Abstract Network Elements traversed by the paths between <source, destination> pairs combined with properties of these Abstract Network Elements that are relevant to the overlay applications' QoE. Both an application via its ALTO client and the ISP via the ALTO server can achieve better confidentiality and resource utilization by appropriately abstracting relevant Abstract Network Elements. Server scalability can also be improved by combining Abstract Network Elements and their properties in a single response.

This document extends [RFC7285] to allow an ALTO server to convey "abstract network state", for paths defined by their <source, destination> pairs. To this end, it introduces a new cost type called "Path Vector" following the cost metric registration specified in [RFC7285] and the updated cost mode registration specified in [I-D.bw-alto-cost-mode]. A Path Vector is an array of identifiers that identifies an Abstract Network Element, which can be associated with various properties. The associations between ANEs and their properties are encoded in an ALTO information resource called Unified Property Map, which is specified in [I-D.ietf-alto-unified-props-new].

For better confidentiality, this document aims to minimize information exposure of an ALTO server when providing Path Vector service. In particular, this document enables and recommends that first ANEs are constructed on demand, and second an ANE is only associated with properties that are requested by an ALTO client. A Path Vector response involves two ALTO Maps: the Cost Map that contains the Path Vector results and the up-to-date Unified Property

Map that contains the properties requested for these ANEs. To enforce consistency and improve server scalability, this document uses the "multipart/related" content type defined in [RFC2387] to return the two maps in a single response.

As a single ISP may not have the knowledge of the full Internet paths between arbitrary endpoints, this document is mainly applicable 1) when there is a single ISP between the requested source and destination PIDs or endpoints, for example, ISP-hosted CDN/edge, tenant interconnection in a single public cloud platform, etc.; or 2) when the Path Vectors are generated from end-to-end measurement data.

2. Requirements Languages

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

When the words appear in lower case, they are to be interpreted with their natural language meanings.

3. Terminology

This document extends the ALTO base protocol [RFC7285] and the Unified Property Map extension [I-D.ietf-alto-unified-props-new]. In addition to the terms defined in these documents, this document also uses the following additional terms:

Abstract Network Element (ANE): An abstract representation for a component in a network that handles data packets and whose properties can potentially have an impact on the end-to-end performance of traffic. An ANE can be a physical device such as a router, a link or an interface, or an aggregation of devices such as a subnetwork or a data center.

The definition of Abstract Network Element is similar to Network Element defined in [RFC2216] in the sense that they both provide an abstract representation of specific components of a network. However, they have different criteria on how these particular components are selected. Specifically, a Network Element requires the components to be capable of exercising QoS control, while Abstract Network Element only requires the components to have an impact on the end-to-end performance.

ANE Name: A string that uniquely identifies an ANE in a specific

scope. An ANE can be constructed either statically in advance or on demand based on the requested information. Thus, different ANEs may only be valid within a particular scope, either ephemeral or persistent. Within each scope, an ANE is uniquely identified by an ANE Name, as defined in Section 6.1. Note that an ALTO client must not assume ANEs in different scopes but with the same ANE Name refer to the same component(s) of the network.

Path Vector: Path Vector, or ANE Path Vector, refers to a JSON array of ANE Names. It is a generalization of BGP path vector. While standard BGP path vector (Section 5.1.2 of [RFC4271]) specifies a sequence of autonomous systems for a destination IP prefix, the Path Vector defined in this extension specifies a sequence of ANEs either for a source Provider-Defined Identifier (PID) and a destination PID as in the CostMapData (11.2.3.6 in [RFC7285]), or for a source endpoint and a destination endpoint as in the EndpointCostMapData object (Section 11.5.1.6 of [RFC7285]).

Path Vector resource: An ALTO information resource (Section 8.1 of [RFC7285]) which supports the extension defined in this document.

Path Vector cost type: A special cost type, which is specified in Section 6.5. When this cost type is present in an IRD entry, it indicates that the information resource is a Path Vector resource. When this cost type is present in a Filtered Cost Map request or an Endpoint Cost Service request, it indicates each cost value must be interpreted as a Path Vector.

Path Vector request: The POST message sent to an ALTO Path Vector resource.

Path Vector response: A Path Vector response refers to the multipart/related message returned by a Path Vector resource.

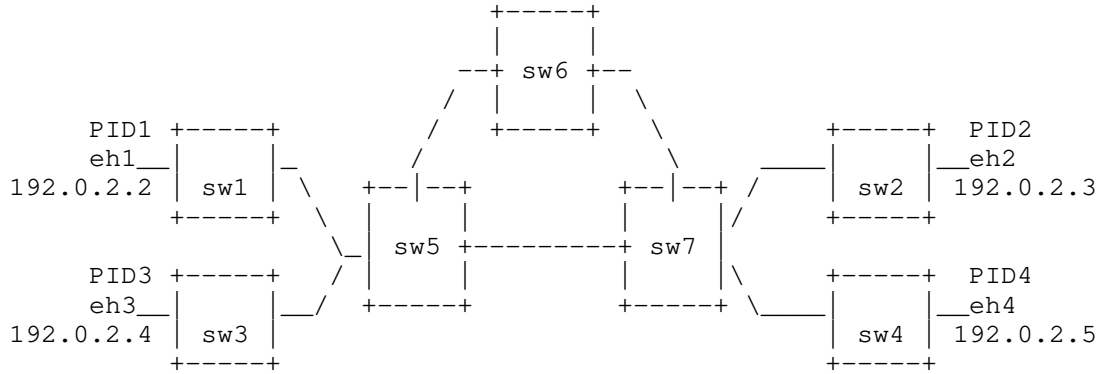
4. Requirements and Use Cases

4.1. Design Requirements

This section gives an illustrative example of how an overlay application can benefit from the extension defined in this document.

Assume that an application has control over a set of flows, which may go through shared links/nodes and share bottlenecks. The application seeks to schedule the traffic among multiple flows to get better performance. The constraints of feasible rate allocations of those flows will benefit the scheduling. However, Cost Maps as defined in [RFC7285] can not reveal such information.

Specifically, consider a network as shown in Figure 1. The network has 7 switches (sw1 to sw7) forming a dumb-bell topology. Switches "sw1", "sw2", "sw3" and "sw4" are access switches, and sw5-sw7 form the backbone. End hosts eh1 to eh4 are connected to access switches sw1 to sw4 respectively. Assume that the bandwidth of link eh1 -> sw1 and link sw1 -> sw5 is 150 Mbps, and the bandwidth of the other links is 100 Mbps.



$\text{bw}(\text{eh1} \rightarrow \text{sw1}) = \text{bw}(\text{sw1} \rightarrow \text{sw5}) = 150 \text{ Mbps}$
 $\text{bw}(\text{eh2} \rightarrow \text{sw2}) = \text{bw}(\text{eh3} \rightarrow \text{sw3}) = \text{bw}(\text{eh4} \rightarrow \text{sw4}) = 100 \text{ Mbps}$
 $\text{bw}(\text{sw1} \rightarrow \text{sw5}) = \text{bw}(\text{sw3} \rightarrow \text{sw5}) = \text{bw}(\text{sw2} \rightarrow \text{sw7}) = \text{bw}(\text{sw4} \rightarrow \text{sw7}) = 100 \text{ Mbps}$
 $\text{bw}(\text{sw5} \rightarrow \text{sw6}) = \text{bw}(\text{sw5} \rightarrow \text{sw7}) = \text{bw}(\text{sw6} \rightarrow \text{sw7}) = 100 \text{ Mbps}$

Figure 1: Raw Network Topology

The base ALTO topology abstraction of the network is shown in Figure 2. Assume the cost map returns an hypothetical cost type representing the available bandwidth between a source and a destination.



Figure 2: Base Topology Abstraction

Now assume the application wants to maximize the total rate of the traffic among a set of <source, destination> pairs, say "eh1 -> eh2" and "eh1 -> eh4". Let "x" denote the transmission rate of "eh1 -> eh2" and "y" denote the rate of "eh1 -> eh4". The objective function is

$$\max(x + y).$$

With the ALTO Cost Map, the cost between PID1 and PID2 and between PID1 and PID4 will both be 100 Mbps. The client can get a capacity region of

$$\begin{aligned} x &\leq 100 \text{ Mbps,} \\ y &\leq 100 \text{ Mbps.} \end{aligned}$$

With this information, the client may mistakenly think it can achieve a maximum total rate of 200 Mbps. However, this rate is infeasible, as there are only two potential cases:

- * Case 1: "eh1 -> eh2" and "eh1 -> eh4" take different path segments from "sw5" to "sw7". For example, if "eh1 -> eh2" uses path "eh1 -> sw1 -> sw5 -> sw6 -> sw7 -> sw2 -> eh2" and "eh1 -> eh4" uses path "eh1 -> sw1 -> sw5 -> sw7 -> sw4 -> eh4", then the shared bottleneck links are "eh1 -> sw1" and "sw1 -> sw5". In this case, the capacity region is:

$$\begin{aligned} x &\leq 100 \text{ Mbps} \\ y &\leq 100 \text{ Mbps} \\ x + y &\leq 150 \text{ Mbps} \end{aligned}$$

and the real optimal total rate is 150 Mbps.

- * Case 2: "eh1 -> eh2" and "eh1 -> eh4" take the same path segment from "sw5" to "sw7". For example, if "eh1 -> eh2" uses path "eh1 -> sw1 -> sw5 -> sw7 -> sw2 -> eh2" and "eh1 -> eh4" also uses path "eh1 -> sw1 -> sw5 -> sw7 -> sw4 -> eh4", then the shared bottleneck link is "sw5 -> sw7". In this case, the capacity region is:

$$\begin{aligned} x &\leq 100 \text{ Mbps} \\ y &\leq 100 \text{ Mbps} \\ x + y &\leq 100 \text{ Mbps} \end{aligned}$$

and the real optimal total rate is 100 Mbps.

Clearly, with more accurate and fine-grained information, the application can gain a better prediction of its traffic and may orchestrate its resources accordingly. However, to provide such information, the network needs to expose abstract information beyond the simple cost map abstraction. In particular:

- * The ALTO server must expose abstract information about the network paths that are traversed by the traffic between a source and a destination beyond a simple numerical value, which allows the overlay application to distinguish between Cases 1 and 2 and to compute the optimal total rate accordingly.
- * The ALTO server must allow the client to distinguish the common ANE shared by "eh1 -> eh2" and "eh1 -> eh4", e.g., "eh1 - sw1" and "sw1 - sw5" in Case 1.
- * The ALTO server must expose abstract information on the properties of the ANEs used by "eh1 -> eh2" and "eh1 -> eh4". For example, an ALTO server can either expose the available bandwidth between "eh1 - sw1", "sw1 - sw5", "sw5 - sw7", "sw5 - sw6", "sw6 - sw7", "sw7 - sw2", "sw7 - sw4", "sw2 - eh2", "sw4 - eh4" in Case 1, or expose 3 abstract elements "A", "B" and "C", which represent the linear constraints that define the same capacity region in Case 1.

In general, we can conclude that to support the multiple flow scheduling use case, the ALTO framework must be extended to satisfy the following additional requirements:

AR1: An ALTO server must provide the ANEs that are important to assess the QoE of the overlay application on the path of a <source, destination> pair.

AR2: An ALTO server must provide information to identify how ANEs are shared on the paths of different <source, destination> pairs.

AR3: An ALTO server must provide information on the properties that are important to assess the QoE of the application for ANEs.

The extension defined in this document specifies a solution to expose such abstract information.

4.2. Sample Use Cases

While the multiple flow scheduling problem is used to help identify the additional requirements, the extension defined in this document can be applied to a wide range of applications. This section highlights some use cases that are reported.

4.2.1. Exposing Network Bottlenecks

An important use case of the Path Vector extension is to expose network bottlenecks. Applications which need to perform large scale data transfers can benefit from being aware of the resource constraints exposed by this extension even if they have different objectives. One such example is the Worldwide LHC Computing Grid (WLCG), the largest example of a distributed computation collaboration in the research and education world.

Figure 3 illustrates an example of using ALTO Path Vector as an interface between the job optimizer for a data analytics system and the network manager. In particular, we assume the objective of the job optimizer is to minimize the job completion time.

In such a setting, the network-aware job optimizer (e.g., [CLARINET]) takes a query and generates multiple query execution plans (QEP). It can encode the QEPs as Path Vector requests that are sent to an ALTO server. The ALTO server obtains the routing information for the flows in a QEP and finds links, routers, or middleboxes (e.g., a stateful firewall) that can potentially become bottlenecks of the QEP (e.g., see [NOVA] and [G2] for mechanisms to identify bottleneck links under different settings). The resource constraint information is encoded in a Path Vector response and returned to the ALTO client.

With the network resource constraints, the job optimizer may choose the QEP with the optimal job completion time to be executed. It must be noted that the ALTO framework itself does not offer the capability to control the traffic. However, certain network managers may offer ways to enforce resource guarantees, such as on-demand tunnels (e.g., [SWAN]), demand vector (e.g., [HUG], [UNICORN]), etc. The traffic control interfaces and mechanisms are out of the scope of this document.

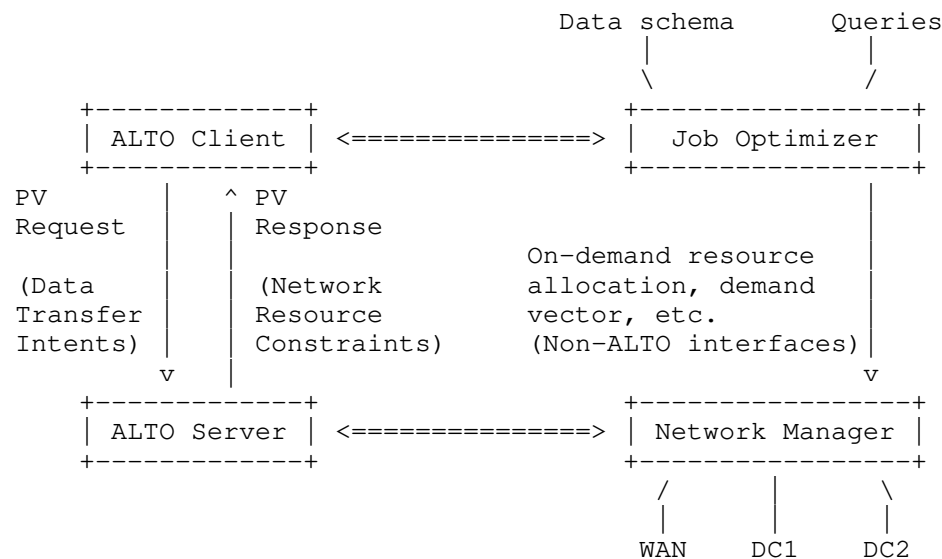


Figure 3: Example Use Case for Data Analytics

Another example is as illustrated in Figure 4. Consider a network consisting of multiple sites and a non-blocking core network, i.e., the links in the core network have sufficient bandwidth that they will not become the bottleneck of the data transfers.

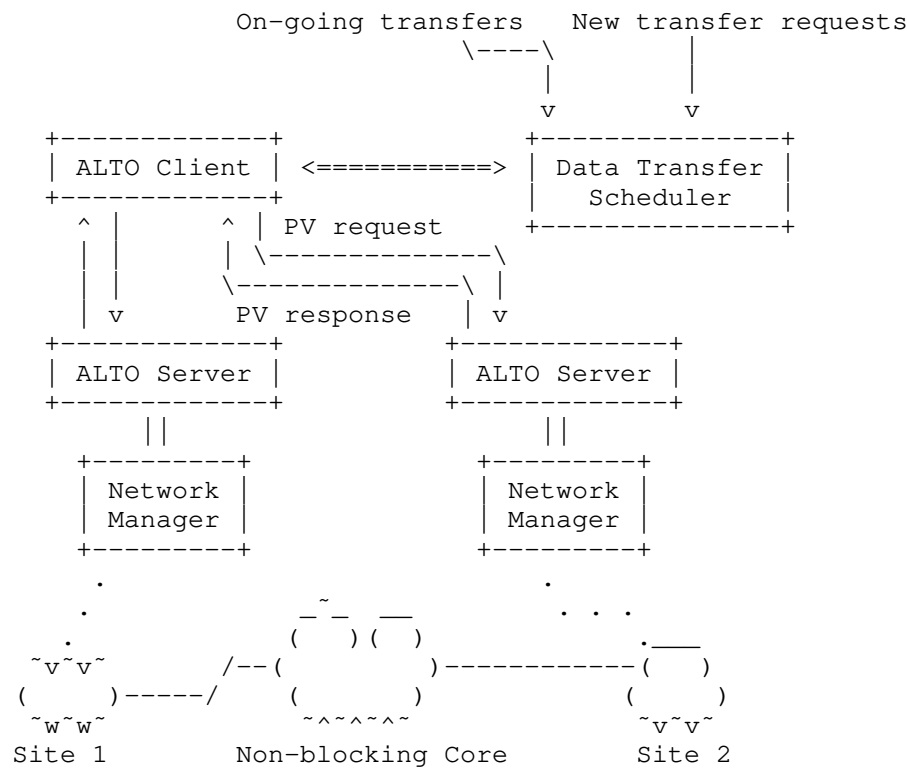
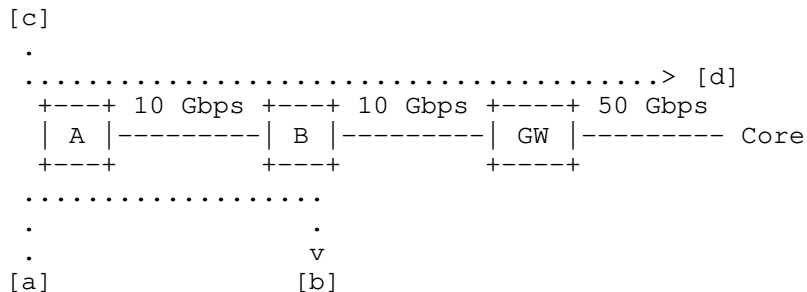


Figure 4: Example Use Case for Cross-site Bottleneck Discovery

Site 1:



Site 2:

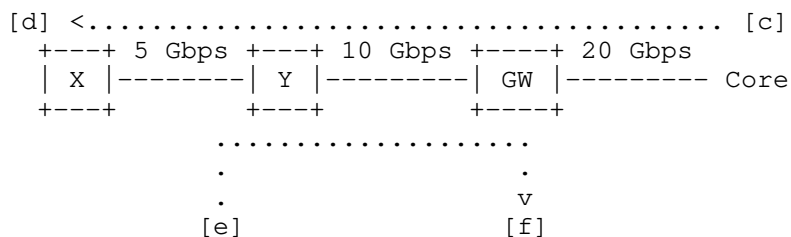


Figure 5: Example: Three Flows in Two Sites

With the Path Vector extension, a site can reveal the bottlenecks inside its own network with necessary information (such as link capacities) to the ALTO client, instead of providing the full topology and routing information, or no bottleneck information at all. The bottleneck information can be used to analyze the impact of adding/removing data transfer flows, e.g., using the [G2] framework. For example, assume hosts "a", "b", "c" are in site 1 and hosts "d", "e", "f" are in site 2, and there are 3 flows in two sites: "a -> b", "c -> d", "e -> f". For these flows, site 1 returns:

```

a: { b: [ane1] },
c: { d: [ane1, ane2, ane3] }

ane1: bw = 10 Gbps (link: A->B)
ane2: bw = 10 Gbps (link: B->GW)
ane3: bw = 50 Gbps (link: GW->Core)

```

and site 2 returns:

```
c: { d: [anei, aneii, aneiii] }  
e: { f: [aneiv] }
```

```
anei: bw = 5 Gbps (link Y->X)  
aneii: bw = 10 Gbps (link GW->Y)  
aneiii: bw = 20 Gbps (link Core->GW)  
aneiv: bw = 10 Gbps (link Y->GW)
```

With the information, the data transfer scheduler can use algorithms such as the theory on bottleneck structure [G2] to predict the potential throughput of the flows.

4.2.2. Resource Exposure for CDN and Service Edge

A growing trend in today's applications (2021) is to bring storage and computation closer to the end users for better QoE, such as Content Delivery Network (CDN), AR/VR, and cloud gaming, as reported in various documents (e.g., [SEREDGE] and [MOWIE]). Internet Service Providers may deploy multiple layers of CDN caches, or more generally service edges, with different latency and available resources including number of CPU cores, memory, and storage.

For example, Figure 6 illustrates a typical edge-cloud scenario where memory is measured in Gigabytes (G) and storage is measured in Terabytes (T). The "on-premise" edge nodes are closest to the end hosts and have the smallest latency, and the site-radio edge node and access central office (CO) have larger latency but more available resources.

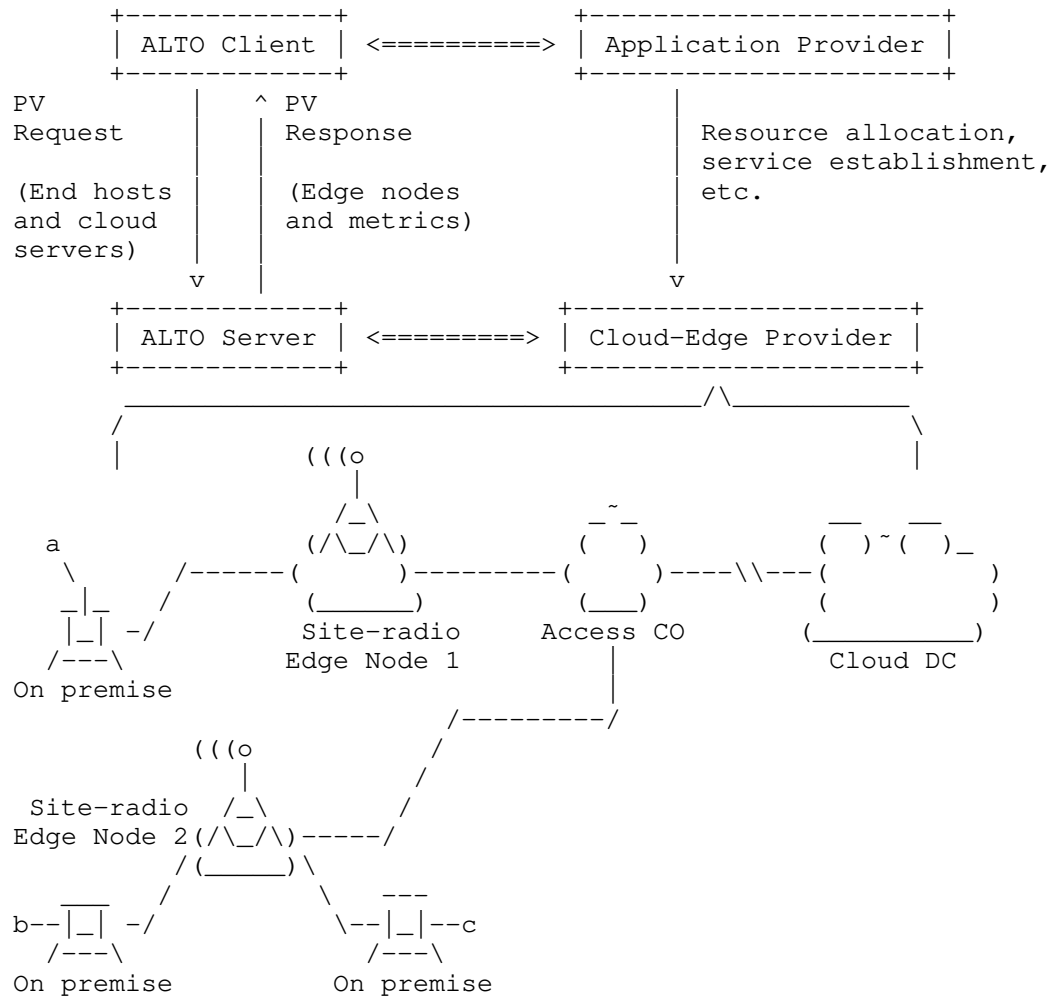


Figure 6: Example Use Case for Service Edge Exposure

```
a: { b: [ane1, ane2, ane3, ane4, ane5],  
      c: [ane1, ane2, ane3, ane4, ane6],  
      DC: [ane1, ane2, ane3] }  
b: { c: [ane5, ane4, ane6], DC: [ane5, ane4, ane3] }  
  
ane1: latency=5ms cpu=2 memory=8G storage=10T  
(on premise, a)  
  
ane2: latency=20ms cpu=4 memory=8G storage=10T  
(Site-radio Edge Node 1)  
  
ane3: latency=100ms cpu=8 memory=128G storage=100T  
(Access CO)  
  
ane4: latency=20ms cpu=4 memory=8G storage=10T  
(Site-radio Edge Node 2)  
  
ane5: latency=5ms cpu=2 memory=8G storage=10T  
(on premise, b)  
  
ane6: latency=5ms cpu=2 memory=8G storage=10T  
(on premise, c)
```

Figure 7: Example Service Edge Query Results

With the extension defined in this document, an ALTO server can selectively reveal the CDNs and service edges that reside along the paths between different end hosts and/or the cloud servers, together with their properties such as capabilities (e.g., storage, GPU) and available Service Level Agreement (SLA) plans. See Figure 7 for an example where the query is made for sources [a, b] and destinations [b, c, DC]. Here each ANE represents a service edge and the properties include access latency, available resources, etc. Note the properties here are only used for illustration purposes and are not part of this extension.

With the service edge information, an ALTO client may better conduct CDN request routing or offload functionalities from the user equipment to the service edge, with considerations on customized quality of experience.

5. Path Vector Extension: Overview

This section provides a non-normative overview of the Path Vector extension defined in this document. It is assumed that the readers are familiar with both the base protocol [RFC7285] and the Unified Property Map extension [I-D.ietf-alto-unified-props-new].

To satisfy the additional requirements listed in Section 4.1, this extension:

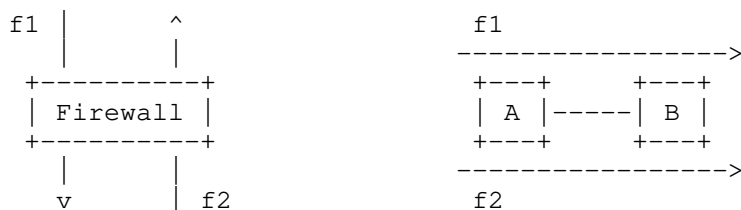
1. introduces the concept of Abstract Network Element (ANE) as the abstraction of components in a network whose properties may have an impact on the end-to-end performance of the traffic handled by those components,
2. extends the Cost Map and Endpoint Cost Service to convey the ANEs traversed by the path of a <source, destination> pair as Path Vectors, and
3. uses the Unified Property Map to convey the association between the ANEs and their properties.

Thus, an ALTO client can learn about the ANEs that are important to assess the QoE of different <source, destination> pairs by investigating the corresponding Path Vector value (AR1), identify common ANEs if an ANE appears in the Path Vectors of multiple <source, destination> pairs (AR2), and retrieve the properties of the ANEs by searching the Unified Property Map (AR3).

5.1. Abstract Network Element (ANE)

This extension introduces ANE as an indirect and network-agnostic way to specify a component or an aggregation of components of a network whose properties have an impact on the end-to-end performance for application traffic between endpoints.

ANEs allow ALTO servers to focus on common properties of different types of network components. For example, the throughput of a flow can be constrained by different components in a network: the capacity of a physical link, the maximum throughput of a firewall, the reserved bandwidth of an MPLS tunnel, etc. See the example below, assume the throughput of the firewall is 100 Mbps and the capacity for link (A, B) is also 100 Mbps, they result in the same constraint on the total throughput of f1 and f2. Thus, they are identical when treated as an ANE.



When an ANE is defined by an ALTO server, it is assigned an identifier by the ALTO server, i.e., a string of type ANEName as specified in Section 6.1, and a set of associated properties.

5.1.1. ANE Entity Domain

In this extension, the associations between ANE and the properties are conveyed in a Unified Property Map. Thus, ANEs must constitute an entity domain (Section 5.1 of [I-D.ietf-alto-unified-props-new]), and each ANE property must be an entity property (Section 5.2 of [I-D.ietf-alto-unified-props-new]).

Specifically, this document defines a new entity domain called "ane" as specified in Section 6.2 and defines two initial properties for the ANE entity domain.

5.1.2. Ephemeral and Persistent ANEs

By design, ANEs are ephemeral and not to be used in further requests to other ALTO resources. More precisely, the corresponding ANE names are no longer valid beyond the scope of a Path Vector response or the incremental update stream for a Path Vector request. Compared with globally unique ANE names, ephemeral ANE has several benefits including better privacy of the ISP's internal structure and more flexible ANE computation.

For example, an ALTO server may define an ANE for each aggregated bottleneck link between the sources and destinations specified in the request. For requests with different sources and destinations, the bottlenecks may be different but can safely reuse the same ANE names. The client can still adjust its traffic based on the information but is difficult to infer the underlying topology with multiple queries.

However, sometimes an ISP may intend to selectively reveal some "persistent" network components which, opposite to being ephemeral, have a longer life cycle. For example, an ALTO server may define an ANE for each service edge cluster. Once a client chooses to use a service edge, e.g., by deploying some user-defined functions, it may want to stick to the service edge to avoid the complexity of state transition or synchronization, and continuously query the properties of the edge cluster.

This document provides a mechanism to expose such network components as persistent ANEs. A persistent ANE has a persistent ID that is registered in a Property Map, together with their properties. See Section 6.2.4 and Section 6.4.2 for more detailed instructions on how to identify ephemeral ANEs and persistent ANEs.

5.1.3. Property Filtering

Resource-constrained ALTO clients (see Section 4.1.2 of [RFC7285]) may benefit from the filtering of Path Vector query results at the ALTO server, as an ALTO client may only require a subset of the available properties.

Specifically, the available properties for a given resource are announced in the Information Resource Directory as a new capability called "ane-property-names". The properties selected by a client as being of interest are specified in the subsequent Path Vector queries using the filter called 'ane-property-names'. The response includes and only includes the selected properties for the ANEs in the response.

The "ane-property-names" capability for Cost Map and for Endpoint Cost Service is specified in Section 7.2.4 and Section 7.3.4 respectively. The "ane-property-names" filter for Cost Map and Endpoint Cost Service is specified in Section 7.2.3 and Section 7.3.3 accordingly.

5.2. Path Vector Cost Type

For an ALTO client to correctly interpret the Path Vector, this extension specifies a new cost type called the Path Vector cost type.

The Path Vector cost type must convey both the interpretation and semantics in the "cost-mode" and "cost-metric" respectively. Unfortunately, a single "cost-mode" value cannot fully specify the interpretation of a Path Vector, which is a compound data type. For example, in programming languages such as C++ where there existed a JSON array type named JSONArray, a Path Vector will have the type of JSONArray<ANENAME>.

Instead of extending the "type system" of ALTO, this document takes a simple and backward compatible approach. Specifically, the "cost-mode" of the Path Vector cost type is "array", which indicates the value is a JSON array. Then, an ALTO client must check the value of the "cost-metric". If the value is "ane-path", it means that the JSON array should be further interpreted as a path of ANENAMES.

The Path Vector cost type is specified in Section 6.5.

5.3. Multipart Path Vector Response

For a basic ALTO information resource, a response contains only one type of ALTO resources, e.g., Network Map, Cost Map, or Property Map. Thus, only one round of communication is required: An ALTO client sends a request to an ALTO server, and the ALTO server returns a response, as shown in Figure 8.

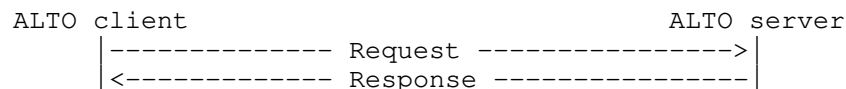


Figure 8: A Typical ALTO Request and Response

The extension defined in this document, on the other hand, involves two types of information resources: Path Vectors conveyed in an `InfoResourceCostMap` (defined in Section 11.2.3.6 of [RFC7285]) or an `InfoResourceEndpointCostMap` (defined in Section 11.5.1.6 of [RFC7285]), and ANE properties conveyed in an `InfoResourceProperties` (defined in Section 7.6 of [I-D.ietf-alto-unified-props-new]).

Instead of two consecutive message exchanges, the extension defined in this document enforces one round of communication. Specifically, the ALTO client must include the source and destination pairs and the requested ANE properties in a single request, and the ALTO server must return a single response containing both the Path Vectors and properties associated with the ANEs in the Path Vectors, as shown in Figure 9. Since the two parts are bundled together in one response message, their orders are interchangeable. See Section 7.2.6 and Section 7.3.6 for details.

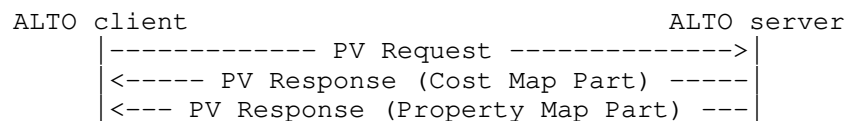


Figure 9: The Path Vector Extension Request and Response

This design is based on the following considerations:

1. ANEs may be constructed on demand, and potentially based on the requested properties (See Section 5.1 for more details). If sources and destinations are not in the same request as the properties, an ALTO server either cannot construct ANEs on-demand, or must wait until both requests are received.

2. As ANEs may be constructed on demand, mappings of each ANE to its underlying network devices and resources can be specific to the request. In order to respond to the Property Map request correctly, an ALTO server must store the mapping of each Path Vector request until the client fully retrieves the property information. The "stateful" behavior may substantially harm the server scalability and potentially lead to Denial-of-Service attacks.

One approach to realize the one-round communication is to define a new media type to contain both objects, but this violates modular design. This document follows the standard-conforming usage of "multipart/related" media type defined in [RFC2387] to elegantly combine the objects. Path Vectors are encoded in an InfoResourceCostMap or an InfoResourceEndpointCostMap, and the Property Map is encoded in an InfoResourceProperties. They are encapsulated as parts of a multipart message. The modular composition allows ALTO servers and clients to reuse the data models of the existing information resources. Specifically, this document addresses the following practical issues using "multipart/related".

5.3.1. Identifying the Media Type of the Root Object

ALTO uses media type to indicate the type of an entry in the Information Resource Directory (IRD) (e.g., "application/alto-costmap+json" for Cost Map and "application/alto-endpointcost+json" for Endpoint Cost Service). Simply putting "multipart/related" as the media type, however, makes it impossible for an ALTO client to identify the type of service provided by related entries.

To address this issue, this document uses the "type" parameter to indicate the root object of a multipart/related message. For a Cost Map resource, the "media-type" field in the IRD entry is "multipart/related" with the parameter "type=application/alto-costmap+json"; for an Endpoint Cost Service, the parameter is "type=application/alto-endpointcost+json".

5.3.2. References to Part Messages

As the response of a Path Vector resource is a multipart message with two different parts, it is important that each part can be uniquely identified. Following the designs of [RFC8895], this extension requires that an ALTO server assigns a unique identifier to each part of the multipart response message. This identifier, referred to as a Part Resource ID (See Section 6.6 for details), is present in the part message's "Content-ID" header. By concatenating the Part Resource ID to the identifier of the Path Vector request, an ALTO server/client can uniquely identify the Path Vector Part or the

Property Map part.

6. Specification: Basic Data Types

6.1. ANE Name

An ANE Name is encoded as a JSON string with the same format as that of the type PIDName (Section 10.1 of [RFC7285]).

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

6.2. ANE Entity Domain

The ANE entity domain associates property values with the Abstract Network Elements in a Property Map. Accordingly, the ANE entity domain always depends on a Property Map.

It must be noted that the term "domain" here does not refer to a network domain. Rather, it is inherited from the "entity domain" defined in Sec 3.2 in [I-D.ietf-alto-unified-props-new] that represents the set of valid entities defined by an ALTO information resource (called the defining information resource).

6.2.1. Entity Domain Type

The Entity Domain Type is "ane".

6.2.2. Domain-Specific Entity Identifier

The entity identifiers are the ANE Names in the associated Property Map.

6.2.3. Hierarchy and Inheritance

There is no hierarchy or inheritance for properties associated with ANEs.

6.2.4. Media Type of Defining Resource

The defining resource for entity domain type "ane" MUST be a Property Map, i.e., the media type of defining resources is:

application/alto-propmap+json

Specifically, for ephemeral ANEs that appear in a Path Vector response, their entity domain names MUST be exactly ".ane" and the defining resource of these ANEs is the Property Map part of the

multipart response. Meanwhile, for any persistent ANE whose defining resource is a Property Map resource, its entity domain name MUST have the format of "PROPMAP.ane" where PROPMAP is the resource ID of the defining resource. Persistent entities are "persistent" because standalone queries can be made by an ALTO client to their defining resource(s) when the connection to the Path Vector service is closed.

For example, the defining resource of an ephemeral ANE whose entity identifier is ".ane:NET1" is the Property Map part that contains this identifier. The defining resource of a persistent ANE whose entity identifier is "dc-props.ane:DC1" is the Property Map with the resource ID "dc-props".

6.3. ANE Property Name

An ANE Property Name is encoded as a JSON string with the same format as that of Entity Property Name (Section 5.2.2 of [I-D.ietf-alto-unified-props-new]).

6.4. Initial ANE Property Types

Two initial ANE property types are specified, "max-reservable-bandwidth" and "persistent-entity-id".

Note that these property types do not depend on any information resource. As such, the EntityPropertyName MUST only have the EntityPropertyType part.

6.4.1. Maximum Reservable Bandwidth

The maximum reservable bandwidth property ("max-reservable-bandwidth") stands for the maximum bandwidth that can be reserved for all the traffic that traverses an ANE. The value MUST be encoded as a non-negative numerical cost value as defined in Section 6.1.2.1 of [RFC7285] and the unit is bit per second (bps). If this property is requested by the ALTO client but not present for an ANE in the server response, it MUST be interpreted as that the property is not defined for the ANE.

This property can be offered in a setting where the ALTO server is part of a network system that provides on-demand resource allocation and the ALTO client is part of a user application. One existing example is [NOVA]: the ALTO server is part of an SDN controller and exposes a list of traversed network elements and associated link bandwidth to the client. The encoding in [NOVA] differs from the Path Vector response defined in this document that the Path Vector part and Property Map part are put in the same JSON object.

In such a framework, the ALTO server exposes resource (e.g., reservable bandwidth) availability information to the ALTO client. How the client makes resource requests based on the information and how the resource allocation is achieved respectively depend on interfaces between the management system and the users or a higher-layer protocol (e.g., SDN network intents or MPLS tunnels), which are out of the scope of this document.

6.4.2. Persistent Entity ID

The persistent entity ID property is the entity identifier of the persistent ANE which an ephemeral ANE presents (See Section 5.1.2 for details). The value of this property is encoded with the format EntityID defined in Section 5.1.3 of [I-D.ietf-alto-unified-props-new].

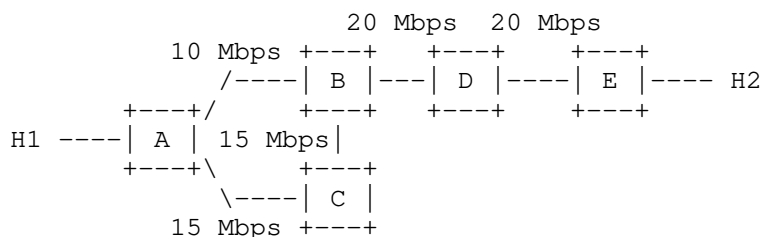
In this format, the entity ID combines:

- * a defining information resource for the ANE on which a "persistent-entity-id" is queried, which is the Property Map resource defining the ANE as a persistent entity, together with the properties;
- * the persistent name of the ANE in that Property Map.

With this format, the client has all the needed information for further standalone query properties on the persistent ANE.

6.4.3. Examples

To illustrate the use of "max-reservable-bandwidth", consider the following network with 5 nodes. Assume the client wants to query the maximum reservable bandwidth from H1 to H2. An ALTO server may split the network into two ANEs: "ane1" that represents the subnetwork with routers A, B, and C, and "ane2" that represents the subnetwork with routers B, D and E. The maximum reservable bandwidth for "ane1" is 15 Mbps (using path A->C->B) and the maximum reservable bandwidth for "ane2" is 20 Mbps (using path B->D->E).



To illustrate the use of "persistent-entity-id", consider the scenario in Figure 6. As the life cycle of service edges are typically long, they may contain information that is not specific to the query. Such information can be stored in an individual unified property map and later be accessed by an ALTO client.

For example, "ane1" in Figure 7 represents the on-premise service edge closest to host a. Assume the properties of the service edges are provided in a unified property map called "se-props" and the ID of the on-premise service edge is "9a0b55f7-7442-4d56-8a2c-b4cc6a8e3aa1", the "persistent-entity-id" of "ane1" will be "se-props.ane:9a0b55f7-7442-4d56-8a2c-b4cc6a8e3aa1". With this persistent entity ID, an ALTO client may send queries to the "se-props" resource with the entity ID ".ane:9a0b55f7-7442-4d56-8a2c-b4cc6a8e3aa1".

6.5. Path Vector Cost Type

This document defines a new cost type, which is referred to as the Path Vector cost type. An ALTO server MUST offer this cost type if it supports the extension defined in this document.

6.5.1. Cost Metric: ane-path

The cost metric "ane-path" indicates the value of such a cost type conveys an array of ANE names, where each ANE name uniquely represents an ANE traversed by traffic from a source to a destination.

An ALTO client MUST interpret the Path Vector as if the traffic between a source and a destination logically traverses the ANEs in the same order as they appear in the Path Vector.

When the Path Vector procedures defined in this document are in use, an ALTO server using the "ane-path" cost metric and the "array" cost mode (see Section 6.5.2) MUST return as the cost value a JSON array of ANEName and the client MUST also check that each element contained in the array is an ANEName (Section 6.1). Otherwise, the client MUST discard the response and SHOULD follow the instructions in Section 8.3.4.3 of [RFC7285] to handle the error.

6.5.2. Cost Mode: array

The cost mode "array" indicates that every cost value in the response body of a (Filtered) Cost Map or an Endpoint Cost Service MUST be interpreted as a JSON array object. While this cost mode can be applied to all cost metrics, additional specifications will be needed to clarify the semantics of the array cost mode when combined with cost metrics other than 'ane-path'.

6.6. Part Resource ID and Part Content ID

A Part Resource ID is encoded as a JSON string with the same format as that of the type ResourceID (Section 10.2 of [RFC7285]).

Even though the client-id assigned to a Path Vector request and the Part Resource ID MAY contain up to 64 characters by their own definition, their concatenation (see Section 5.3.2) MUST also conform to the same length constraint. The same requirement applies to the resource ID of the Path Vector resource, too. Thus, it is RECOMMENDED to limit the length of resource ID and client ID related to a Path Vector resource to 31 characters.

A Part Content ID conforms to the format of msg-id as specified in [RFC2387] and [RFC5322]. Specifically, it has the following format:

"<" PART-RESOURCE-ID "@" DOMAIN-NAME ">"

PART-RESOURCE-ID: PART-RESOURCE-ID has the same format as the Part Resource ID. It is used to identify whether a part message is a Path Vector or a Property Map.

DOMAIN-NAME: DOMAIN-NAME has the same format as dot-atom-text specified in Section 3.2.3 of [RFC5322]. It must be the domain name of the ALTO server.

7. Specification: Service Extensions

7.1. Notations

This document uses the same syntax and notations as introduced in Section 8.2 of RFC 7285 [RFC7285] to specify the extensions to existing ALTO resources and services.

7.2. Multipart Filtered Cost Map for Path Vector

This document introduces a new ALTO resource called multipart Filtered Cost Map resource, which allows an ALTO server to provide other ALTO resources associated with the Cost Map resource in the same response.

7.2.1. Media Type

The media type of the multipart Filtered Cost Map resource is "multipart/related" and the required "type" parameter MUST have a value of "application/alto-costmap+json".

7.2.2. HTTP Method

The multipart Filtered Cost Map is requested using the HTTP POST method.

7.2.3. Accept Input Parameters

The input parameters of the multipart Filtered Cost Map are supplied in the body of an HTTP POST request. This document extends the input parameters to a Filtered Cost Map, which is defined as a JSON object of type ReqFilteredCostMap in Section 4.1.2 of RFC 8189 [RFC8189], with a data format indicated by the media type "application/alto-costmapfilter+json", which is a JSON object of type PVReqFilteredCostMap:

```
object {  
  [EntityPropertyName ane-property-names<0..*>;]  
} PVReqFilteredCostMap : ReqFilteredCostMap;
```

with fields:

ane-property-names: A list of selected ANE properties to be included in the response. Each property in this list MUST match one of the supported ANE properties indicated in the resource's "ane-property-names" capability (Section 7.2.4). If the field is not present, it MUST be interpreted as an empty list.

Example: Consider the network in Figure 1. If an ALTO client wants to query the "max-reservable-bandwidth" between PID1 and PID2, it can submit the following request.

```
POST /costmap/pv HTTP/1.1
Host: alto.example.com
Accept: multipart/related;type=application/alto-costmap+json,
       application/alto-error+json
Content-Length: 201
Content-Type: application/alto-costmapfilter+json

{
  "cost-type": {
    "cost-mode": "array",
    "cost-metric": "ane-path"
  },
  "pids": {
    "srcs": [ "PID1" ],
    "dsts": [ "PID2" ]
  },
  "ane-property-names": [ "max-reservable-bandwidth" ]
}
```

7.2.4. Capabilities

The multipart Filtered Cost Map resource extends the capabilities defined in Section 4.1.1 of [RFC8189]. The capabilities are defined by a JSON object of type PVFilteredCostMapCapabilities:

```
object {
  [EntityTypePropertyName ane-property-names<0..*>;]
} PVFilteredCostMapCapabilities : FilteredCostMapCapabilities;
```

with fields:

ane-property-names: Defines a list of ANE properties that can be returned. If the field is not present, it MUST be interpreted as an empty list, indicating the ALTO server cannot provide any ANE property.

This extension also introduces additional restrictions for the following fields:

cost-type-names: The "cost-type-names" field MUST include the Path Vector cost type, unless explicitly documented by a future extension. This also implies that the Path Vector cost type MUST be defined in the "cost-types" of the Information Resource Directory's "meta" field.

cost-constraints: If the "cost-type-names" field includes the Path Vector cost type, "cost-constraints" field MUST be "false" or not present unless specifically instructed by a future document.

testable-cost-type-names (Section 4.1.1 of [RFC8189]): If the "cost-type-names" field includes the Path Vector cost type and the "testable-cost-type-names" field is present, the Path Vector cost type MUST NOT be included in the "testable-cost-type-names" field unless specifically instructed by a future document.

7.2.5. Uses

This member MUST include the resource ID of the network map based on which the PIDs are defined. If this resource supports "persistent-entity-id", it MUST also include the defining resources of persistent ANEs that may appear in the response.

7.2.6. Response

The response MUST indicate an error, using ALTO protocol error handling, as defined in Section 8.5 of [RFC7285], if the request is invalid.

The "Content-Type" header of the response MUST be "multipart/related" as defined by [RFC2387] with the following parameters:

type: The type parameter is mandatory and MUST be "application/alto-costmap+json". Note that [RFC2387] permits both parameters with and without the double quotes.

start: The start parameter is as defined in [RFC2387] and is optional. If present, it MUST have the same value as the "Content-ID" header of the Path Vector part.

boundary: The boundary parameter is as defined in Section 5.1.1 of [RFC2046] and is mandatory.

The body of the response MUST consist of two parts:

- * The Path Vector part MUST include "Content-ID" and "Content-Type" in its header. The "Content-Type" MUST be "application/alto-costmap+json". The value of "Content-ID" MUST have the same format as the Part Content ID as specified in Section 6.6.

The body of the Path Vector part MUST be a JSON object with the same format as defined in Section 11.2.3.6 of [RFC7285] when the "cost-type" field is present in the input parameters and MUST be a JSON object with the same format as defined in Section 4.1.3 of [RFC8189] if the "multi-cost-types" field is present. The JSON object MUST include the "vtag" field in the "meta" field, which provides the version tag of the returned CostMapData. The resource ID of the version tag MUST follow the format of

resource-id '.' part-resource-id

where "resource-id" is the resource Id of the Path Vector resource, and "part-resource-id" has the same value as the PART-RESOURCE-ID in the "Content-ID" of the Path Vector part. The "meta" field MUST also include the "dependent-vtags" field, whose value is a single-element array to indicate the version tag of the network map used, where the network map is specified in the "uses" attribute of the multipart Filtered Cost Map resource in IRD.

- * The Unified Property Map part MUST also include "Content-ID" and "Content-Type" in its header. The "Content-Type" MUST be "application/alto-propmap+json". The value of "Content-ID" MUST have the same format as the Part Content ID as specified in Section 6.6.

The body of the Unified Property Map part is a JSON object with the same format as defined in Section 7.6 of [I-D.ietf-alto-unified-props-new]. The JSON object MUST include the "dependent-vtags" field in the "meta" field. The value of the "dependent-vtags" field MUST be an array of VersionTag objects as defined by Section 10.3 of [RFC7285]. The "vtag" of the Path Vector part MUST be included in the "dependent-vtags". If "persistent-entity-id" is requested, the version tags of the dependent resources that may expose the entities in the response MUST also be included.

The PropertyMapData has one member for each ANENAME that appears in the Path Vector part, which is an entity identifier belonging to the self-defined entity domain as defined in Section 5.1.2.3 of [I-D.ietf-alto-unified-props-new]. The EntityProps for each ANE has one member for each property that is both 1) associated with the ANE, and 2) specified in the "ane-property-names" in the request. If the Path Vector cost type is not included in the "cost-type" field or the "multi-cost-type" field, the "property-map" field MUST be present and the value MUST be an empty object ({}).

A complete and valid response MUST include both the Path Vector part and the Property Map part in the multipart message. If any part is NOT present, the client MUST discard the received information and send another request if necessary.

According to [RFC2387], the Path Vector part, whose media type is the same as the "type" parameter of the multipart response message, is the root object. Thus, it is the element the application processes first. Even though the "start" parameter allows it to be placed anywhere in the part sequence, it is RECOMMENDED that the parts

arrive in the same order as they are processed, i.e., the Path Vector part is always put as the first part, followed by the Property Map part. When doing so, an ALTO server MAY choose not to set the "start" parameter, which implies the first part is the root object.

Example: Consider the network in Figure 1. The response of the example request in Section 7.2.3 is as follows, where "ANE1" represents the aggregation of all the switches in the network.

```
HTTP/1.1 200 OK
Content-Length: 859
Content-Type: multipart/related; boundary=example-1;
              type=application/alto-costmap+json

--example-1
Content-ID: <costmap@alto.example.com>
Content-Type: application/alto-costmap+json

{
  "meta": {
    "vtag": {
      "resource-id": "filtered-cost-map-pv.costmap",
      "tag": "fb20b76204814e9db37a51151faaaef2"
    },
    "dependent-vtags": [
      {
        "resource-id": "my-default-networkmap",
        "tag": "75ed013b3cb58f896e839582504f6228"
      }
    ],
    "cost-type": { "cost-mode": "array", "cost-metric": "ane-path" }
  },
  "cost-map": {
    "PID1": { "PID2": ["ANE1"] }
  }
}

--example-1
Content-ID: <propmap@alto.example.com>
Content-Type: application/alto-propmap+json

{
  "meta": {
    "dependent-vtags": [
      {
        "resource-id": "filtered-cost-map-pv.costmap",
        "tag": "fb20b76204814e9db37a51151faaaef2"
      }
    ]
  },
  "property-map": {
    ".ane:ANE1": { "max-reservable-bandwidth": 100000000 }
  }
}
```

7.3. Multipart Endpoint Cost Service for Path Vector

This document introduces a new ALTO resource called multipart Endpoint Cost Service, which allows an ALTO server to provide other ALTO resources associated with the Endpoint Cost Service resource in the same response.

7.3.1. Media Type

The media type of the multipart Endpoint Cost Service resource is "multipart/related" and the required "type" parameter MUST have a value of "application/alto-endpointcost+json".

7.3.2. HTTP Method

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

7.3.3. Accept Input Parameters

The input parameters of the multipart Endpoint Cost Service resource are supplied in the body of an HTTP POST request. This document extends the input parameters to an Endpoint Cost Service, which is defined as a JSON object of type ReqEndpointCost in Section 4.2.2 of [RFC8189], with a data format indicated by the media type "application/alto-endpointcostparams+json", which is a JSON object of type PVReqEndpointCost:

```
object {  
  [EntityPropertyName ane-property-names<0..*>;]  
} PVReqEndpointcost : ReqEndpointcostMap;
```

with fields:

ane-property-names: This document defines the "ane-property-names" in PVReqEndpointcost as the same as in PVReqFilteredCostMap. See Section 7.2.3.

Example: Consider the network in Figure 1. If an ALTO client wants to query the "max-reservable-bandwidth" between eh1 and eh2, it can submit the following request.

```
POST /ecs/pv HTTP/1.1
Host: alto.example.com
Accept: multipart/related;type=application/alto-endpointcost+json,
       application/alto-error+json
Content-Length: 227
Content-Type: application/alto-endpointcostparams+json
```

```
{
  "cost-type": {
    "cost-mode": "array",
    "cost-metric": "ane-path"
  },
  "endpoints": {
    "srcs": [ "ipv4:192.0.2.2" ],
    "dsts": [ "ipv4:192.0.2.18" ]
  },
  "ane-property-names": [ "max-reservable-bandwidth" ]
}
```

7.3.4. Capabilities

The capabilities of the multipart Endpoint Cost Service resource are defined by a JSON object of type `PVEndpointcostCapabilities`, which is defined as the same as `PVFilteredCostMapCapabilities`. See Section 7.2.4.

7.3.5. Uses

If this resource supports "persistent-entity-id", it MUST also include the defining resources of persistent ANEs that may appear in the response.

7.3.6. Response

The response MUST indicate an error, using ALTO protocol error handling, as defined in Section 8.5 of [RFC7285], if the request is invalid.

The "Content-Type" header of the response MUST be "multipart/related" as defined by [RFC7285] with the following parameters:

type: The type parameter MUST be "application/alto-endpointcost+json" and is mandatory.

start: The start parameter is as defined in Section 7.2.6.

boundary: The boundary parameter is as defined in Section 5.1.1 of [RFC2046] and is mandatory.

The body MUST consist of two parts:

- * The Path Vector part MUST include "Content-ID" and "Content-Type" in its header. The "Content-Type" MUST be "application/alto-endpointcost+json". The value of "Content-ID" MUST have the same format as the Part Content ID as specified in Section 6.6.

The body of the Path Vector part MUST be a JSON object with the same format as defined in Section 11.5.1.6 of [RFC7285] when the "cost-type" field is present in the input parameters and MUST be a JSON object with the same format as defined in Section 4.2.3 of [RFC8189] if the "multi-cost-types" field is present. The JSON object MUST include the "vtag" field in the "meta" field, which provides the version tag of the returned EndpointCostMapData. The resource ID of the version tag MUST follow the format of

resource-id '.' part-resource-id

where "resource-id" is the resource Id of the Path Vector resource, and "part-resource-id" has the same value as the PART-RESOURCE-ID in the "Content-ID" of the Path Vector part.

- * The Unified Property Map part MUST also include "Content-ID" and "Content-Type" in its header. The "Content-Type" MUST be "application/alto-propmap+json". The value of "Content-ID" MUST have the same format as the Part Content ID as specified in Section 6.6.

The body of the Unified Property Map part MUST be a JSON object with the same format as defined in Section 7.6 of [I-D.ietf-alto-unified-props-new]. The JSON object MUST include the "dependent-vtags" field in the "meta" field. The value of the "dependent-vtags" field MUST be an array of VersionTag objects as defined by Section 10.3 of [RFC7285]. The "vtag" of the Path Vector part MUST be included in the "dependent-vtags". If "persistent-entity-id" is requested, the version tags of the dependent resources that may expose the entities in the response MUST also be included.

The PropertyMapData has one member for each ANENAME that appears in the Path Vector part, which is an entity identifier belonging to the self-defined entity domain as defined in Section 5.1.2.3 of [I-D.ietf-alto-unified-props-new]. The EntityProps for each ANE has one member for each property that is both 1) associated with the ANE, and 2) specified in the "ane-property-names" in the request. If the Path Vector cost type is not included in the "cost-type" field or the "multi-cost-type" field, the "property-map" field MUST be present and the value MUST be an empty object ({}).

A complete and valid response MUST include both the Path Vector part and the Property Map part in the multipart message. If any part is NOT present, the client MUST discard the received information and send another request if necessary.

According to [RFC2387], the Path Vector part, whose media type is the same as the "type" parameter of the multipart response message, is the root object. Thus, it is the element the application processes first. Even though the "start" parameter allows it to be placed anywhere in the part sequence, it is RECOMMENDED that the parts arrive in the same order as they are processed, i.e., the Path Vector part is always put as the first part, followed by the Property Map part. When doing so, an ALTO server MAY choose not to set the "start" parameter, which implies the first part is the root object.

Example: Consider the network in Figure 1. The response of the example request in Section 7.3.3 is as follows.

```
HTTP/1.1 200 OK
Content-Length: 845
Content-Type: multipart/related; boundary=example-1;
             type=application/alto-endpointcost+json
```

```
--example-1
Content-ID: <ecs@alto.example.com>
Content-Type: application/alto-endpointcost+json
```

```
{
  "meta": {
    "vtag": {
      "resource-id": "ecs-pv.ecs",
      "tag": "ec137bb78118468c853d5b622ac003f1"
    },
    "dependent-vtags": [
      {
        "resource-id": "my-default-networkmap",
        "tag": "677fe5f4066848d282ece213a84f9429"
      }
    ],
    "cost-type": { "cost-mode": "array", "cost-metric": "ane-path" },
    "cost-map": {
      "ipv4:192.0.2.2": { "ipv4:192.0.2.18": ["ANE1"] }
    }
  }
}
```

```
--example-1
Content-ID: <propmap@alto.example.com>
Content-Type: application/alto-propmap+json
```

```
{
  "meta": {
    "dependent-vtags": [
      {
        "resource-id": "ecs-pv.ecs",
        "tag": "ec137bb78118468c853d5b622ac003f1"
      }
    ]
  },
  "property-map": {
    ".ane:ANE1": { "max-reservable-bandwidth": 100000000 }
  }
}
```

8. Examples

This section lists some examples of Path Vector queries and the corresponding responses. Some long lines are truncated for better readability.

8.1. Sample Setup

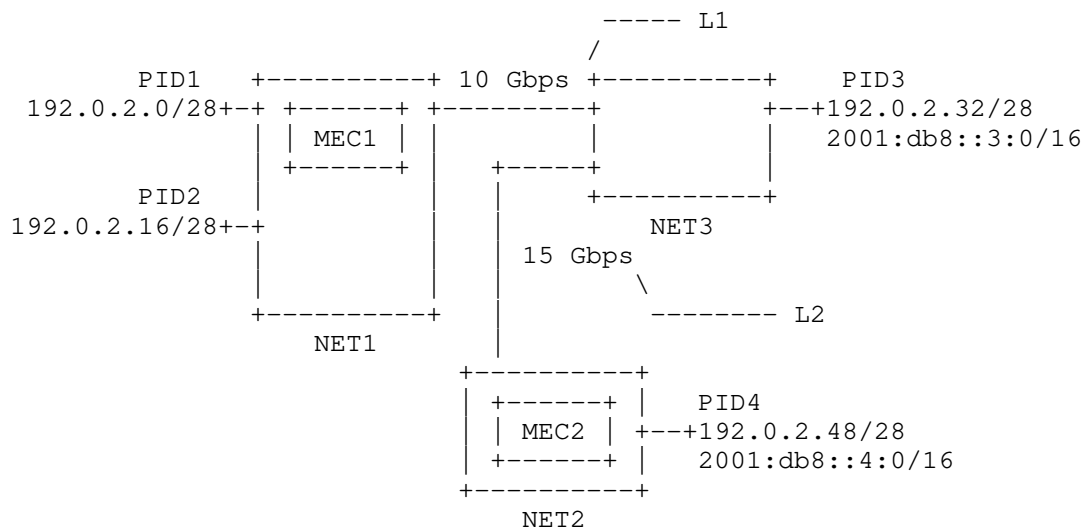


Figure 10: Examples of ANE Properties

In this document, Figure 10 is used to illustrate the message contents. There are 3 sub-networks (NET1, NET2 and NET3) and two interconnection links (L1 and L2). It is assumed that each sub-network has sufficiently large bandwidth to be reserved.

8.2. Information Resource Directory

To give a comprehensive example of the extension defined in this document, we consider the network in Figure 10. Assume that the ALTO server provides the following information resources:

- * "my-default-networkmap": A Network Map resource which contains the PIDs in the network.
- * "filtered-cost-map-pv": A Multipart Filtered Cost Map resource for Path Vector, which exposes the "max-reservable-bandwidth" property for the PIDs in "my-default-networkmap".

- * "ane-props": A filtered Unified Property resource that exposes the information for persistent ANEs in the network.
- * "endpoint-cost-pv": A Multipart Endpoint Cost Service for Path Vector, which exposes the "max-reservable-bandwidth" and the "persistent-entity-id" properties.
- * "update-pv": An Update Stream service, which provides the incremental update service for the "endpoint-cost-pv" service.
- * "multicost-pv": A Multipart Endpoint Cost Service with both Multi-Cost and Path Vector.

Below is the Information Resource Directory of the example ALTO server. To enable the extension defined in this document, the "path-vector" cost type (Section 6.5) is defined in the "cost-types" of the "meta" field, and is included in the "cost-type-names" of resources "filtered-cost-map-pv" and "endpoint-cost-pv".

```
{
  "meta": {
    "cost-types": {
      "path-vector": {
        "cost-mode": "array",
        "cost-metric": "ane-path"
      },
      "num-rc": {
        "cost-mode": "numerical",
        "cost-metric": "routingcost"
      }
    }
  },
  "resources": {
    "my-default-networkmap": {
      "uri": "https://alto.example.com/networkmap",
      "media-type": "application/alto-networkmap+json"
    },
    "filtered-cost-map-pv": {
      "uri": "https://alto.example.com/costmap/pv",
      "media-type": "multipart/related;
        type=application/alto-costmap+json",
      "accepts": "application/alto-costmapfilter+json",
      "capabilities": {
        "cost-type-names": [ "path-vector" ],
        "ane-property-names": [ "max-reservable-bandwidth" ]
      },
      "uses": [ "my-default-networkmap" ]
    }
  }
}
```

```
"ane-props": {
  "uri": "https://alto.example.com/ane-props",
  "media-type": "application/alto-propmap+json",
  "accepts": "application/alto-propmapparams+json",
  "capabilities": {
    "mappings": {
      ".ane": [ "cpu" ]
    }
  }
},
"endpoint-cost-pv": {
  "uri": "https://alto.exmaple.com/endpointcost/pv",
  "media-type": "multipart/related;
    type=application/alto-endpointcost+json",
  "accepts": "application/alto-endpointcostparams+json",
  "capabilities": {
    "cost-type-names": [ "path-vector" ],
    "ane-property-names": [
      "max-reservable-bandwidth", "persistent-entity-id"
    ]
  },
  "uses": [ "ane-props" ]
},
"update-pv": {
  "uri": "https://alto.example.com/updates/pv",
  "media-type": "text/event-stream",
  "uses": [ "endpoint-cost-pv" ],
  "accepts": "application/alto-updatestreamparams+json",
  "capabilities": {
    "support-stream-control": true
  }
},
"multicost-pv": {
  "uri": "https://alto.exmaple.com/endpointcost/mcpv",
  "media-type": "multipart/related;
    type=application/alto-endpointcost+json",
  "accepts": "application/alto-endpointcostparams+json",
  "capabilities": {
    "cost-type-names": [ "path-vector", "num-rc" ],
    "max-cost-types": 2,
    "testable-cost-type-names": [ "num-rc" ],
    "ane-property-names": [
      "max-reservable-bandwidth", "persistent-entity-id"
    ]
  },
  "uses": [ "ane-props" ]
}
}
```

```
}
```

8.3. Multipart Filtered Cost Map

The following examples demonstrate the request to the "filtered-cost-map-pv" resource and the corresponding response.

The request uses the "path-vector" cost type in the "cost-type" field. The "ane-property-names" field is missing, indicating that the client only requests for the Path Vector but not the ANE properties.

The response consists of two parts. The first part returns the array of ANEName for each source and destination pair. There are two ANEs, where "L1" represents the interconnection link L1, and "L2" represents the interconnection link L2.

The second part returns an empty Property Map. Note that the ANE entries are omitted since they have no properties (See Section 3.1 of [I-D.ietf-alto-unified-props-new]).

```
POST /costmap/pv HTTP/1.1
Host: alto.example.com
Accept: multipart/related;type=application/alto-costmap+json,
       application/alto-error+json
Content-Length: 153
Content-Type: application/alto-costmapfilter+json
```

```
{
  "cost-type": {
    "cost-mode": "array",
    "cost-metric": "ane-path"
  },
  "pids": {
    "srcs": [ "PID1" ],
    "dsts": [ "PID3", "PID4" ]
  }
}
```

```
HTTP/1.1 200 OK
Content-Length: 855
Content-Type: multipart/related; boundary=example-1;
             type=application/alto-costmap+json
```

```
--example-1
Content-ID: <costmap@alto.example.com>
Content-Type: application/alto-costmap+json
```

```

{
  "meta": {
    "vtag": {
      "resource-id": "filtered-cost-map-pv.costmap",
      "tag": "d827f484cb66ce6df6b5077cb8562b0a"
    },
    "dependent-vtags": [
      {
        "resource-id": "my-default-networkmap",
        "tag": "c04bc5da49534274a6daeee8ea1dec62"
      }
    ],
    "cost-type": {
      "cost-mode": "array",
      "cost-metric": "ane-path"
    }
  },
  "cost-map": {
    "PID1": {
      "PID3": [ "L1" ],
      "PID4": [ "L1", "L2" ]
    }
  }
}
--example-1
Content-ID: <propmap@alto.example.com>
Content-Type: application/alto-propmap+json

{
  "meta": {
    "dependent-vtags": [
      {
        "resource-id": "filtered-cost-map-pv.costmap",
        "tag": "d827f484cb66ce6df6b5077cb8562b0a"
      }
    ]
  },
  "property-map": {
  }
}

```

8.4. Multipart Endpoint Cost Service Resource

The following examples demonstrate the request to the "endpoint-cost-pv" resource and the corresponding response.

The request uses the Path Vector cost type in the "cost-type" field, and queries the Maximum Reservable Bandwidth ANE property and the Persistent Entity property for two IPv4 source and destination pairs (192.0.2.34 -> 192.0.2.2 and 192.0.2.34 -> 192.0.2.50) and one IPv6 source and destination pair (2001:db8::3:1 -> 2001:db8::4:1).

The response consists of two parts. The first part returns the array of ANENAME for each valid source and destination pair. As one can see in Figure 10, flow 192.0.2.34 -> 192.0.2.2 traverses NET2, L1 and NET1, and flows 192.0.2.34 -> 192.0.2.50 and 2001:db8::3:1 -> 2001:db8::4:1 traverse NET2, L2 and NET3.

The second part returns the requested properties of ANEs. Assume NET1, NET2 and NET3 has sufficient bandwidth and their "max-reservable-bandwidth" values are set to a sufficiently large number (50 Gbps in this case). On the other hand, assume there are no prior reservation on L1 and L2, and their "max-reservable-bandwidth" values are the corresponding link capacity (10 Gbps for L1 and 15 Gbps for L2).

Both NET1 and NET2 have a mobile edge deployed, i.e., MEC1 in NET1 and MEC2 in NET2. Assume the ANENAME for MEC1 and MEC2 are "MEC1" and "MEC2" and their properties can be retrieved from the Property Map "ane-props". Thus, the "persistent-entity-id" property of NET1 and NET3 are "ane-props.ane:MEC1" and "ane-props.ane:MEC2" respectively.

```
POST /endpointcost/pv HTTP/1.1
Host: alto.example.com
Accept: multipart/related;
       type=application/alto-endpointcost+json,
       application/alto-error+json
Content-Length: 362
Content-Type: application/alto-endpointcostparams+json
```

```
{
  "cost-type": {
    "cost-mode": "array",
    "cost-metric": "ane-path"
  },
  "endpoints": {
    "srcs": [
      "ipv4:192.0.2.34",
      "ipv6:2001:db8::3:1"
    ],
    "dsts": [
      "ipv4:192.0.2.2",
      "ipv4:192.0.2.50",
      "ipv6:2001:db8::4:1"
    ]
  },
  "ane-property-names": [
    "max-reservable-bandwidth",
    "persistent-entity-id"
  ]
}
```

```
HTTP/1.1 200 OK
Content-Length: 1432
Content-Type: multipart/related; boundary=example-2;
             type=application/alto-endpointcost+json
```

```
--example-2
Content-ID: <ecs@alto.example.com>
Content-Type: application/alto-endpointcost+json
```

```
{
  "meta": {
    "vtags": {
      "resource-id": "endpoint-cost-pv.ecs",
      "tag": "bb6bb72eafe8f9bdc4f335c7ed3b10822a391cef"
    },
    "cost-type": {
      "cost-mode": "array",
      "cost-metric": "ane-path"
    }
  }
}
```

```
    }
  },
  "endpoint-cost-map": {
    "ipv4:192.0.2.34": {
      "ipv4:192.0.2.2": [ "NET3", "L1", "NET1" ],
      "ipv4:192.0.2.50": [ "NET3", "L2", "NET2" ]
    },
    "ipv6:2001:db8::3:1": {
      "ipv6:2001:db8::4:1": [ "NET3", "L2", "NET2" ]
    }
  }
}
--example-2
Content-ID: <propmap@alto.example.com>
Content-Type: application/alto-propmap+json

{
  "meta": {
    "dependent-vtags": [
      {
        "resource-id": "endpoint-cost-pv.ecs",
        "tag": "bb6bb72eafe8f9bdc4f335c7ed3b10822a391cef"
      },
      {
        "resource-id": "ane-props",
        "tag": "bf3c8c1819d2421c9a95a9d02af557a3"
      }
    ]
  },
  "property-map": {
    ".ane:NET1": {
      "max-reservable-bandwidth": 50000000000,
      "persistent-entity-id": "ane-props.ane:MEC1"
    },
    ".ane:NET2": {
      "max-reservable-bandwidth": 50000000000,
      "persistent-entity-id": "ane-props.ane:MEC2"
    },
    ".ane:NET3": {
      "max-reservable-bandwidth": 50000000000
    },
    ".ane:L1": {
      "max-reservable-bandwidth": 10000000000
    },
    ".ane:L2": {
      "max-reservable-bandwidth": 15000000000
    }
  }
}
```

```
}
```

Under certain scenarios where the traversal order is not crucial, an ALTO server implementation may choose to not follow strictly the physical traversal order and may even obfuscate the order intentionally to preserve its own privacy or conform to its own policies. For example, an ALTO server may choose to aggregate NET1 and L1 as a new ANE with ANE name "AGGR1", and aggregate NET2 and L2 as a new ANE with ANE name "AGGR2". The "max-reservable-bandwidth" of "AGGR1" takes the value of L1, which is smaller than that of NET1, and the "persistent-entity-id" of "AGGR1" takes the value of NET1. The properties of "AGGR2" are computed in a similar way and the obfuscated response is as shown below. Note that the obfuscation of Path Vector responses is implementation-specific and is out of the scope of this document, and developers may refer to Section 11 for further references.

```
HTTP/1.1 200 OK
Content-Length: 1263
Content-Type: multipart/related; boundary=example-2;
              type=application/alto-endpointcost+json
```

```
--example-2
Content-ID: <ecs@alto.example.com>
Content-Type: application/alto-endpointcost+json
```

```
{
  "meta": {
    "vtags": {
      "resource-id": "endpoint-cost-pv.ecs",
      "tag": "bb975862fbe3422abf4dae386b132c1d"
    },
    "cost-type": {
      "cost-mode": "array",
      "cost-metric": "ane-path"
    }
  },
  "endpoint-cost-map": {
    "ipv4:192.0.2.34": {
      "ipv4:192.0.2.2": [ "NET3", "AGGR1" ],
      "ipv4:192.0.2.50": [ "NET3", "AGGR2" ]
    },
    "ipv6:2001:db8::3:1": {
      "ipv6:2001:db8::4:1": [ "NET3", "AGGR2" ]
    }
  }
}
```

```
--example-2
```

Content-ID: <propmap@alto.example.com>
Content-Type: application/alto-propmap+json

```
{
  "meta": {
    "dependent-vtags": [
      {
        "resource-id": "endpoint-cost-pv.ecs",
        "tag": "bb975862fbe3422abf4dae386b132c1d"
      },
      {
        "resource-id": "ane-props",
        "tag": "bf3c8c1819d2421c9a95a9d02af557a3"
      }
    ]
  },
  "property-map": {
    ".ane:AGGR1": {
      "max-reservable-bandwidth": 10000000000,
      "persistent-entity-id": "ane-props.ane:MEC1"
    },
    ".ane:AGGR2": {
      "max-reservable-bandwidth": 15000000000,
      "persistent-entity-id": "ane-props.ane:MEC2"
    },
    ".ane:NET3": {
      "max-reservable-bandwidth": 50000000000
    }
  }
}
```

8.5. Incremental Updates

In this example, an ALTO client subscribes to the incremental update for the multipart Endpoint Cost Service resource "endpoint-cost-pv".

```
POST /updates/pv HTTP/1.1
Host: alto.example.com
Accept: text/event-stream
Content-Type: application/alto-updatestreamparams+json
Content-Length: 112
```

```
{
  "add": {
    "ecspvsub1": {
      "resource-id": "endpoint-cost-pv",
      "input": <ecs-input>
    }
  }
}
```

Based on the server-side process defined in [RFC8895], the ALTO server will send the "control-uri" first using Server-Sent Event (SSE), followed by the full response of the multipart message.

```
HTTP/1.1 200 OK
Connection: keep-alive
Content-Type: text/event-stream
```

```
event: application/alto-updatestreamcontrol+json
data: {"control-uri": "https://alto.example.com/updates/streams/123"}
```

```
event: multipart/related;boundary=example-3;
      type=application/alto-endpointcost+json,ecspvsub1
data: --example-3
data: Content-ID: <ecsmap@alto.example.com>
data: Content-Type: application/alto-endpointcost+json
data:
data: <endpoint-cost-map-entry>
data: --example-3
data: Content-ID: <propmap@alto.example.com>
data: Content-Type: application/alto-propmap+json
data:
data: <property-map-entry>
data: --example-3--
```

When the contents change, the ALTO server will publish the updates for each node in this tree separately, based on Section 6.7.3 of [RFC8895].

```
event: application/merge-patch+json, ecspvsubl.ecsmap@alto.example.com  
data: <Merge patch for endpoint-cost-map-update>
```

```
event: application/merge-patch+json, ecspvsubl.propmap@alto.example.com  
data: <Merge patch for property-map-update>
```

8.6. Multi-cost

The following examples demonstrate the request to the "multicost-pv" resource and the corresponding response.

The request asks for two cost types: the first is the Path Vector cost type, and the second is a numerical routing cost. It also queries the Maximum Reservable Bandwidth ANE property and the Persistent Entity property for two IPv4 source and destination pairs (192.0.2.34 -> 192.0.2.2 and 192.0.2.34 -> 192.0.2.50) and one IPv6 source and destination pair (2001:db8::3:1 -> 2001:db8::4:1).

The response consists of two parts. The first part returns a JSONArray that contains two JSONValue for each requested source and destination pair: the first JSONValue is a JSONArray of ANENames, which is the value of the Path Vector cost type, and the second JSONValue is a JSONNumber which is the value of the routing cost. The second part contains a Property Map that maps the ANEs to their requested properties.

```
POST /endpointcost/mcpv HTTP/1.1
Host: alto.example.com
Accept: multipart/related;
       type=application/alto-endpointcost+json,
       application/alto-error+json
Content-Length: 433
Content-Type: application/alto-endpointcostparams+json

{
  "multi-cost-types": [
    { "cost-mode": "array", "cost-metric": "ane-path" },
    { "cost-mode": "numerical", "cost-metric": "routingcost" }
  ],
  "endpoints": {
    "srcs": [
      "ipv4:192.0.2.34",
      "ipv6:2001:db8::3:1"
    ],
    "dsts": [
      "ipv4:192.0.2.2",
      "ipv4:192.0.2.50",
      "ipv6:2001:db8::4:1"
    ]
  },
  "ane-property-names": [
    "max-reservable-bandwidth",
    "persistent-entity-id"
  ]
}

HTTP/1.1 200 OK
Content-Length: 1350
Content-Type: multipart/related; boundary=example-4;
             type=application/alto-endpointcost+json

--example-4
Content-ID: <ecs@alto.example.com>
Content-Type: application/alto-endpointcost+json

{
  "meta": {
    "vtags": {
      "resource-id": "endpoint-cost-pv.ecs",
      "tag": "84a4f9c14f9341f0983e3e5f43a371c8"
    },
    "multi-cost-types": [
      { "cost-mode": "array", "cost-metric": "ane-path" },
      { "cost-mode": "numerical", "cost-metric": "routingcost" }
    ]
  }
}
```



```

    ]
  },
  "endpoint-cost-map": {
    "ipv4:192.0.2.34": {
      "ipv4:192.0.2.2": [[ "NET3", "AGGR1" ], 3],
      "ipv4:192.0.2.50": [[ "NET3", "AGGR2" ], 2]
    },
    "ipv6:2001:db8::3:1": {
      "ipv6:2001:db8::4:1": [[ "NET3", "AGGR2" ], 2]
    }
  }
}
--example-4
Content-ID: <propmap@alto.example.com>
Content-Type: application/alto-propmap+json

{
  "meta": {
    "dependent-vtags": [
      {
        "resource-id": "endpoint-cost-pv.ecs",
        "tag": "84a4f9c14f9341f0983e3e5f43a371c8"
      },
      {
        "resource-id": "ane-props",
        "tag": "be157afa031443a187b60bb80a86b233"
      }
    ]
  },
  "property-map": {
    ".ane:AGGR1": {
      "max-reservable-bandwidth": 10000000000,
      "persistent-entity-id": "ane-props.ane:MEC1"
    },
    ".ane:AGGR2": {
      "max-reservable-bandwidth": 15000000000,
      "persistent-entity-id": "ane-props.ane:MEC2"
    },
    ".ane:NET3": {
      "max-reservable-bandwidth": 50000000000
    }
  }
}

```

9. Compatibility with Other ALTO Extensions

9.1. Compatibility with Legacy ALTO Clients/Servers

The multipart Filtered Cost Map resource and the multipart Endpoint Cost Service resource has no backward compatibility issue with legacy ALTO clients and servers. Although these two types of resources reuse the media types defined in the base ALTO protocol for the accept input parameters, they have different media types for responses. If the ALTO server provides these two types of resources, but the ALTO client does not support them, the ALTO client will ignore the resources without incurring any incompatibility problem.

9.2. Compatibility with Multi-Cost Extension

The extension defined in this document is compatible with the multi-cost extension [RFC8189]. Such a resource has a media type of either "multipart/related; type=application/alto-costmap+json" or "multipart/related; type=application/alto-endpointcost+json". Its "cost-constraints" field must either be "false" or not present and the Path Vector cost type must be present in the "cost-type-names" capability field but must not be present in the "testable-cost-type-names" field, as specified in Section 7.2.4 and Section 7.3.4.

9.3. Compatibility with Incremental Update

This extension is compatible with the incremental update extension [RFC8895]. ALTO clients and servers MUST follow the specifications given in Sections 5.2 and 6.7.3 of [RFC8895] to support incremental updates for a Path Vector resource.

9.4. Compatibility with Cost Calendar

The extension specified in this document is compatible with the Cost Calendar extension [RFC8896]. When used together with the Cost Calendar extension, the cost value between a source and a destination is an array of Path Vectors, where the k-th Path Vector refers to the abstract network paths traversed in the k-th time interval by traffic from the source to the destination.

When used with time-varying properties, e.g., maximum reservable bandwidth, a property of a single ANE may also have different values in different time intervals. In this case, if such an ANE has different property values in two time intervals, it MUST be treated as two different ANEs, i.e., with different entity identifiers. However, if it has the same property values in two time intervals, it MAY use the same identifier.

This rule allows the Path Vector extension to represent both changes of ANEs and changes of the ANEs' properties in a uniform way. The Path Vector part is calendared in a compatible way, and the Property Map part is not affected by the calendar extension.

The two extensions combined together can provide the historical network correlation information for a set of source and destination pairs. A network broker or client may use this information to derive other resource requirements such as Time-Block-Maximum Bandwidth, Bandwidth-Sliding-Window, and Time-Bandwidth-Product (TBP) (See [SENSE] for details).

10. General Discussions

10.1. Constraint Tests for General Cost Types

The constraint test is a simple approach to query the data. It allows users to filter the query result by specifying some boolean tests. This approach is already used in the ALTO protocol. [RFC7285] and [RFC8189] allow ALTO clients to specify the "constraints" and "or-constraints" tests to better filter the result.

However, the current syntax can only be used to test scalar cost types, and cannot easily express constraints on complex cost types, e.g., the Path Vector cost type defined in this document.

In practice, developing a bespoke language for general-purpose boolean tests can be a complex undertaking, and it is conceivable that there are some existing implementations already (the authors have not done an exhaustive search to determine whether there are such implementations). One avenue to develop such a language may be to explore extending current query languages like XQuery [XQuery] or JSONiq [JSONiq] and integrating these with ALTO.

Filtering the Path Vector results or developing a more sophisticated filtering mechanism is beyond the scope of this document.

10.2. General Multi-Resource Query

Querying multiple ALTO information resources continuously is a general requirement. Enabling such a capability, however, must address general issues like efficiency and consistency. The incremental update extension [RFC8895] supports submitting multiple queries in a single request, and allows flexible control over the queries. However, it does not cover the case introduced in this document where multiple resources are needed for a single request.

This extension gives an example of using a multipart message to encode the responses from two specific ALTO information resources: a Filtered Cost Map or an Endpoint Cost Service, and a Property Map. By packing multiple resources in a single response, the implication is that servers may proactively push related information resources to clients.

Thus, it is worth looking into the direction of extending the SSE mechanism as used in the incremental update extension [RFC8895], or upgrading to HTTP/2 [I-D.ietf-httpbis-http2bis] and HTTP/3 [I-D.ietf-quic-http], which provides the ability to multiplex queries and to allow servers proactively send related information resources.

Defining a general multi-resource query mechanism is out of the scope of this document.

11. Security Considerations

This document is an extension of the base ALTO protocol, so the Security Considerations [RFC7285] of the base ALTO protocol fully apply when this extension is provided by an ALTO server.

The Path Vector extension requires additional scrutiny on three security considerations discussed in the base protocol: confidentiality of ALTO information (Section 15.3 of [RFC7285]), potential undesirable guidance from authenticated ALTO information (Section 15.2 of [RFC7285]), and availability of ALTO service (Section 15.5 of [RFC7285]).

For confidentiality of ALTO information, a network operator should be aware that this extension may introduce a new risk: the Path Vector information, when used together with sensitive ANE properties such as capacities of bottleneck links, may make network attacks easier. For example, as the Path Vector information may reveal more fine-grained internal network structures than the base protocol, an attacker may identify the bottleneck link and start a distributed denial-of-service (DDoS) attack involving minimal flows to conduct the in-network congestion. Given the potential risk of leaking sensitive information, the Path Vector extension is mainly applicable in scenarios where 1) the ANE structures and ANE properties do not impose security risks to the ALTO service provider, e.g., not carrying sensitive information, or 2) the ALTO server and client have established a reliable trust relationship, for example, operated in the same administrative domain, or managed by business partners with legal contracts.

Three risk types are identified in Section 15.3.1 of [RFC7285]: (1) Excess disclosure of the ALTO service provider's data to an unauthorized ALTO client; (2) Disclosure of the ALTO service provider's data (e.g., network topology information or endpoint addresses) to an unauthorized third party; and (3) Excess retrieval of the ALTO service provider's data by collaborating ALTO clients. To mitigate these risks, an ALTO server MUST follow the guidelines in Section 15.3.2 of [RFC7285]. Furthermore, an ALTO server MUST follow the following additional protections strategies for risk types (1) and (3).

For risk type (1), an ALTO server MUST use the authentication methods specified in Section 15.3.2 of [RFC7285] to authenticate the identify of an ALTO client, and apply access control techniques to restrict unprivileged ALTO clients from retrieving sensitive Path Vector information. For settings where the ALTO server and client are not in the same trust domain, the ALTO server should reach agreements with the ALTO client on protecting the confidentiality before granting the access to Path Vector service with sensitive information. Such agreements may include legal contracts or Digital Right Management (DRM) techniques. Otherwise, the ALTO server MUST NOT offer the Path Vector service carrying sensitive information to the clients unless the potential risks are fully assessed and mitigated.

For risk type (3), an ALTO service provider must be aware that persistent ANEs may be used as "landmarks" in collaborative inferences. Thus, they should only be used when exposing public service access points (e.g., API gateways, CDNi) and/or when the granularity is coarse-grained (e.g., when an ANE represents an AS, a data center or a WAN). Otherwise, an ALTO server MUST use dynamic mappings from ephemeral ANE names to underlying physical entities. Specifically, for the same physical entity, an ALTO server SHOULD assign a different ephemeral ANE name when the entity appears in the responses to different clients or even for different request from the same client. A RECOMMENDED assignment strategy is to generate ANE names from random numbers.

Further, to protect the network topology from graph reconstruction (e.g., through isomorphic graph identification [BONDY]), the ALTO server SHOULD consider protection mechanisms to reduce information exposure or obfuscate the real information. When doing so, the ALTO server must be aware that information reduction/obfuscation may lead to potential Undesirable Guidance from Authenticated ALTO Information risk (Section 15.2 of [RFC7285]).

Thus, implementations of ALTO servers involving reduction or obfuscation of the Path Vector information SHOULD consider reduction/obfuscation mechanisms that can preserve the integrity of ALTO information, for example, by using minimal feasible region compression algorithms [NOVA] or obfuscation protocols [RESA][MERCATOR]. However, these obfuscation methods are experimental and their practical applicability of these methods to the generic capability provided by this extension is not fully assessed. The ALTO server MUST carefully verify that the deployment scenario satisfies the security assumptions of these methods before applying them to protect Path Vector services with sensitive network information.

For availability of ALTO service, an ALTO server should be cognizant that using Path Vector extension might have a new risk: frequent requesting for Path Vectors might consume intolerable amounts of the server-side computation and storage, which can break the ALTO server. For example, if an ALTO server implementation dynamically computes the Path Vectors for each request, the service providing Path Vectors may become an entry point for denial-of-service attacks on the availability of an ALTO server.

To mitigate this risk, an ALTO server may consider using optimizations such as precomputation-and-projection mechanisms [MERCATOR] to reduce the overhead for processing each query. Also, an ALTO server may also protect itself from malicious clients by monitoring the behaviors of clients and stopping serving clients with suspicious behaviors (e.g., sending requests at a high frequency).

The ALTO service providers must be aware that providing incremental updates of the "max-reservable-bandwidth" may provide information about other consumers of the network. For example, a change of the value may indicate one or more reservations has been made or changed. To mitigate this risk, an ALTO server can batch the updates and/or add a random delay before publishing the updates.

12. IANA Considerations

12.1. ALTO Cost Metric Registry

This document registers a new entry to the ALTO Cost Metric Registry, as instructed by Section 14.2 of [RFC7285]. The new entry is as shown below in Table 1.

Identifier	Intended Semantics	Security Considerations
ane-path	See Section 6.5.1	See Section 11

Table 1: ALTO Cost Metric Registry

12.2. ALTO Cost Mode Registry

This document registers a new entry to the ALTO Cost Mode Registry, as instructed by Section 4 of [I-D.bw-alto-cost-mode]. The new entry is as shown below in Table 2.

Identifier	Intended Semantics
array	See Section 6.5.2

Table 2: ALTO Cost Mode Registry

12.3. ALTO Entity Domain Type Registry

This document registers a new entry to the ALTO Domain Entity Type Registry, as instructed by Section 12.2 of [I-D.ietf-alto-unified-props-new]. The new entry is as shown below in Table 3.

Identifier	Entity Identifier Encoding	Hierarchy & Inheritance	Media Type of Defining Resource	Mapping to ALTO Address Type
ane	See Section 6.2.2	None	application/alto-propmap+json	false

Table 3: ALTO Entity Domain Type Registry

Identifier: See Section 6.2.1.

Entity Identifier Encoding: See Section 6.2.2.

Hierarchy: None

Inheritance: None

Media Type of Defining Resource: See Section 6.2.4.

Mapping to ALTO Address Type: This entity type does not map to ALTO address type.

Security Considerations: In some usage scenarios, ANE 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 ANEs will be made aware of how (or if) the addressing scheme relates to private information and network proximity, in further iterations of this document.

12.4. ALTO Entity Property Type Registry

Two initial entries "max-reservable-bandwidth" and "persistent-entity-id" are registered to the ALTO Domain "ane" in the "ALTO Entity Property Type Registry", as instructed by Section 12.3 of [I-D.ietf-alto-unified-props-new]. The two new entries are shown below in Table 4 and their details can be found in Section 12.4.1 and Section 12.4.2.

Identifier	Intended Semantics	Media Type of Defining Resource
max-reservable-bandwidth	See Section 6.4.1	application/alto-propmap+json
persistent-entity-id	See Section 6.4.2	application/alto-propmap+json

Table 4: Initial Entries for ane Domain in the ALTO Entity Property Types Registry

12.4.1. New ANE Property Type: Maximum Reservable Bandwidth

Identifier: "max-reservable-bandwidth"

Intended Semantics: See Section 6.4.1.

Media Type of Defining Resource: application/alto-propmap+json

Security Considerations: This property is essential for applications such as large-scale data transfers or overlay network interconnection to make better choice of bandwidth reservation. It may reveal the bandwidth usage of the underlying network and can potentially be leveraged to reduce the cost of conducting

denial-of-service attacks. Thus, the ALTO server MUST consider protection mechanisms including only providing the information to authorized clients, and information reduction and obfuscation as introduced in Section 11.

12.4.2. New ANE Property Type: Persistent Entity ID

Identifier: "persistent-entity-id"

Intended Semantics: See Section 6.4.2.

Media Type of Defining Resource: application/alto-propmap+json

Security Considerations: This property is useful when an ALTO server wants to selectively expose certain service points whose detailed properties can be further queried by applications. The entity IDs may consider sensitive information about the underlying network, and an ALTO server should follow the security considerations in Section 11 of [I-D.ietf-alto-unified-props-new].

13. References

13.1. Normative References

[I-D.bw-alto-cost-mode]

Boucadair, M. and Q. Wu, "A Cost Mode Registry for the Application-Layer Traffic Optimization (ALTO) Protocol", Work in Progress, Internet-Draft, draft-bw-alto-cost-mode-01, 4 March 2022, <<https://datatracker.ietf.org/doc/html/draft-bw-alto-cost-mode-01>>.

[I-D.ietf-alto-unified-props-new]

Roome, W., Randriamasy, S., Yang, Y. R., Zhang, J. J., and K. Gao, "An ALTO Extension: Entity Property Maps", Work in Progress, Internet-Draft, draft-ietf-alto-unified-props-new-24, 28 February 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-alto-unified-props-new-24>>.

[RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, DOI 10.17487/RFC2046, November 1996, <<https://www.rfc-editor.org/rfc/rfc2046>>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/rfc/rfc2119>>.

- [RFC2387] Levinson, E., "The MIME Multipart/Related Content-type", RFC 2387, DOI 10.17487/RFC2387, August 1998, <<https://www.rfc-editor.org/rfc/rfc2387>>.
- [RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <<https://www.rfc-editor.org/rfc/rfc5322>>.
- [RFC7285] Alimi, R., Ed., Penno, R., Ed., Yang, Y., Ed., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, DOI 10.17487/RFC7285, September 2014, <<https://www.rfc-editor.org/rfc/rfc7285>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/rfc/rfc8174>>.
- [RFC8189] Randriamasy, S., Roome, W., and N. Schwan, "Multi-Cost Application-Layer Traffic Optimization (ALTO)", RFC 8189, DOI 10.17487/RFC8189, October 2017, <<https://www.rfc-editor.org/rfc/rfc8189>>.
- [RFC8895] Roome, W. and Y. Yang, "Application-Layer Traffic Optimization (ALTO) Incremental Updates Using Server-Sent Events (SSE)", RFC 8895, DOI 10.17487/RFC8895, November 2020, <<https://www.rfc-editor.org/rfc/rfc8895>>.
- [RFC8896] Randriamasy, S., Yang, R., Wu, Q., Deng, L., and N. Schwan, "Application-Layer Traffic Optimization (ALTO) Cost Calendar", RFC 8896, DOI 10.17487/RFC8896, November 2020, <<https://www.rfc-editor.org/rfc/rfc8896>>.

13.2. Informative References

- [BONDY] Bondy, J.A. and R.L. Hemminger, "Graph reconstructiona survey", Journal of Graph Theory, Volume 1, Issue 3, pp 227-268 , 1977, <<https://doi.org/10.1002/jgt.3190010306>>.
- [BOXOPT] Xiang, Q., Yu, H., Aspnes, J., Le, F., Kong, L., and Y.R. Yang, "Optimizing in the dark: Learning an optimal solution through a simple request interface", Proceedings of the AAAI Conference on Artificial Intelligence 33, 1674-1681 , 2019, <<https://doi.org/10.1609/aaai.v33i01.33011674>>.

- [CLARINET] Viswanathan, R., Ananthanarayanan, G., and A. Akella, "CLARINET: WAN-Aware Optimization for Analytics Queries", In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), USENIX Association, Savannah, GA, 435-450 , 2016, <<https://dl.acm.org/doi/abs/10.5555/3026877.3026911>>.
- [G2] Ros-Giralt, J., Bohara, A., Yellamraju, S., Langston, M.H., Lethin, R., Jiang, Y., Tassiulas, L., Li, J., Tan, Y., and M. Veeraraghavan, "On the Bottleneck Structure of Congestion-Controlled Networks", Proceedings of the ACM on Measurement and Analysis of Computing Systems, Volume 3, Issue 3, pp 1-31 , 2019, <<https://dl.acm.org/doi/10.1145/3366707>>.
- [HUG] Chowdhury, M., Liu, Z., Ghodsi, A., and I. Stoica, "HUG: Multi-Resource Fairness for Correlated and Elastic Demands", 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16) (Santa Clara, CA, 2016), 407-424. , 2016, <<https://dl.acm.org/doi/10.5555/2930611.2930638>>.
- [I-D.ietf-alto-performance-metrics]
Wu, Q., Yang, Y. R., Lee, Y., Dhody, D., Randriamasy, S., and L. M. C. Murillo, "ALTO Performance Cost Metrics", Work in Progress, Internet-Draft, draft-ietf-alto-performance-metrics-26, 2 March 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-alto-performance-metrics-26>>.
- [I-D.ietf-httpbis-http2bis]
Thomson, M. and C. Benfield, "HTTP/2", Work in Progress, Internet-Draft, draft-ietf-httpbis-http2bis-07, 24 January 2022, <<https://datatracker.ietf.org/doc/html/draft-ietf-httpbis-http2bis-07>>.
- [I-D.ietf-quic-http]
Bishop, M., "Hypertext Transfer Protocol Version 3 (HTTP/3)", Work in Progress, Internet-Draft, draft-ietf-quic-http-34, 2 February 2021, <<https://datatracker.ietf.org/doc/html/draft-ietf-quic-http-34>>.
- [JSONiq] "The JSON Query language", 2020, <<https://www.jsoniq.org/>>.

- [MERCATOR] Xiang, Q., Zhang, J., Wang, X., Liu, Y., Guok, C., Le, F., MacAuley, J., Newman, H., and Y.R. Yang, "Toward Fine-Grained, Privacy-Preserving, Efficient Multi-Domain Network Resource Discovery", IEEE/ACM IEEE Journal on Selected Areas of Communication 37(8): 1924-1940, 2019, <<https://doi.org/10.1109/JSAC.2019.2927073>>.
- [MOWIE] Zhang, Y., Li, G., Xiong, C., Lei, Y., Huang, W., Han, Y., Walid, A., Yang, Y.R., and Z. Zhang, "MoWIE: Toward Systematic, Adaptive Network Information Exposure as an Enabling Technique for Cloud-Based Applications over 5G and Beyond", In Proceedings of the Workshop on Network Application Integration/CoDesign, ACM, Virtual Event USA, 20-27. , 2020, <<https://doi.org/10.1145/3405672.3409489>>.
- [NOVA] Gao, K., Xiang, Q., Wang, X., Yang, Y.R., and J. Bi, "An objective-driven on-demand network abstraction for adaptive applications", IEEE/ACM Transactions on Networking (TON) Vol 27, no. 2 (2019): 805-818., 2019, <<https://doi.org/10.1109/IWQoS.2017.7969117>>.
- [RESA] Xiang, Q., Zhang, J., Wang, X., Liu, Y., Guok, C., Le, F., MacAuley, J., Newman, H., and Y.R. Yang, "Fine-grained, multi-domain network resource abstraction as a fundamental primitive to enable high-performance, collaborative data sciences", Proceedings of the Super Computing 2018, 5:1-5:13 , 2019, <<https://doi.org/10.1109/SC.2018.000008>>.
- [RFC2216] Shenker, S. and J. Wroclawski, "Network Element Service Specification Template", RFC 2216, DOI 10.17487/RFC2216, September 1997, <<https://www.rfc-editor.org/rfc/rfc2216>>.
- [RFC4271] Rekhter, Y., Ed., Li, T., Ed., and S. Hares, Ed., "A Border Gateway Protocol 4 (BGP-4)", RFC 4271, DOI 10.17487/RFC4271, January 2006, <<https://www.rfc-editor.org/rfc/rfc4271>>.
- [SENSE] "Software Defined Networking (SDN) for End-to-End Networked Science at the Exascale", 2019, <<https://www.es.net/network-r-and-d/sense/>>.
- [SEREDGE] Contreras, L., Baliosian, J., Martnez-Julia, P., and J. Serrat, "Computing at the Edge: But, what Edge?", In proceedings of the NOMS 2020 - 2020 IEEE/IFIP Network Operations and Management Symposium. pp. 1-9. , 2020, <<https://doi.org/10.1109/NOMS47738.2020.9110342>>.

- [SWAN] Hong, C., Kandula, S., Mahajan, R., Zhang, M., Gill, V., Nanduri, M., and R. Wattenhofer, "Achieving High Utilization with Software-driven WAN", In Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM (SIGCOMM '13), ACM, New York, NY, USA, 15-26. , 2013, <<http://doi.acm.org/10.1145/2486001.2486012>>.
- [UNICORN] Xiang, Q., Chen, S., Gao, K., Newman, H., Taylor, I., Zhang, J., and Y.R. Yang, "Unicorn: Unified Resource Orchestration for Multi-Domain, Geo-Distributed Data Analytics", 2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI) (Aug. 2017), 1-6. , 2017, <<https://doi.org/10.1016/j.future.2018.09.048>>.
- [XQuery] "XQuery 3.1: An XML Query Language", 2017, <<https://www.w3.org/TR/xquery-31/>>.

Appendix A. Acknowledgments

The authors would like to thank discussions with Andreas Voellmy, Erran Li, Haibin Song, Haizhou Du, Jiayuan Hu, Qiao Xiang, Tianyuan Liu, Xiao Shi, Xin Wang, and Yan Luo. The authors thank Greg Bernstein, Dawn Chen, Wendy Roome, and Michael Scharf for their contributions to earlier drafts.

The authors would also like to thank Tim Chown, Luis Contreras, Roman Danyliw, Benjamin Kaduk, Erik Kline, Suresh Krishnan, Murray Kucherawy, Warren Kumari, Danny Lachos, Francesca Palombini, Eric Vyncke, Samuel Weiler, and Qiao Xiang whose feedback and suggestions are invaluable to improve the practicability and conciseness of this document, and Mohamed Boucadair, Martin Duke, Vijay Gurbani, Jan Seedorf, and Qin Wu who provide great support and guidance.

Appendix B. Revision Logs (To be removed before publication)

B.1. Changes since -20

Reivision -21

- * changes the normative requirement on protecting confidentiality of PV information with softer language

B.2. Changes since -19

Revision -20

- * changes the IANA registry information
- * adopts the comments from IESG reviews

B.3. Changes since -18

Revision -19

- * adds detailed examples for use cases
- * clarify terms with ambiguous meanings

B.4. Changes since -17

Revision -18

- * changes the specification for content-id to conform to [RFC2387] and [RFC5322]
- * adds IPv6 examples

B.5. Changes since -16

Revision -17

- * adds items for media type of defining resources in IANA considerations

B.6. Changes since -15

Revision -16

- * resolves the compatibility with the Multi-Cost extension (RFC 8189)
- * adds media types of defining resources for ANE property types (for IANA registration)

B.7. Changes since -14

Revision -15

- * fixes the IDNits warnings,

- * fixes grammar issues,
- * addresses the comments in the AD review.

B.8. Changes since -13

Revision -14

- * addresses the comments in the chair review,
- * fixes most issues raised by IDNits.

B.9. Changes since -12

Revision -13

- * changes the abstract based on the chairs' reviews
- * integrates Richard's responds to WGLC reviews

B.10. Changes since -11

Revision -12

- * clarifies the definition of ANEs in a similar way as how Network Elements is defined in [RFC2216]
- * restructures several paragraphs that are not clear (Sec 3, Path Vector bullet, Sec 4.2, Sec 5.1.3, Sec 6.2.4, Sec 6.4.2, Sec 9.3)
- * uses "ALTO Entity Domain Type Registry"

B.11. Changes since -10

Revision -11

- * replaces "part" with "components" in the abstract;
- * identifies additional requirements (AR) derived from the flow scheduling example, and introduces how the extension addresses the additional requirements
- * fixes the inconsistent use of "start" parameter in multipart responses;
- * specifies explicitly how to handle "cost-constraints";

- * uses the latest IANA registration mechanism defined in [I-D.ietf-alto-unified-props-new];
- * renames "persistent-entities" to "persistent-entity-id";
- * makes "application/alto-propmap+json" as the media type of defining resources for the "ane" domain;
- * updates the examples;
- * adds the discussion on ephemeral and persistent ANEs.

B.12. Changes since -09

Revision -10

- * revises the introduction which
 - extends the scope where the PV extension can be applied beyond the "path correlation" information
- * brings back the capacity region use case to better illustrate the problem
- * revises the overview to explain and defend the concepts and decision choices
- * fixes inconsistent terms, typos

B.13. Changes since -08

This revision

- * fixes a few spelling errors
- * emphasizes that abstract network elements can be generated on demand in both introduction and motivating use cases

B.14. Changes Since Version -06

- * We emphasize the importance of the path vector extension in two aspects:
 1. It expands the problem space that can be solved by ALTO, from preferences of network paths to correlations of network paths.
 2. It is motivated by new usage scenarios from both application's and network's perspectives.

- * More use cases are included, in addition to the original capacity region use case.
- * We add more discussions to fully explore the design space of the path vector extension and justify our design decisions, including the concept of abstract network element, cost type (reverted to -05), newer capabilities and the multipart message.
- * Fix the incremental update process to be compatible with SSE -16 draft, which uses client-id instead of resource-id to demultiplex updates.
- * Register an additional ANE property (i.e., persistent-entities) to cover all use cases mentioned in the draft.

Authors' Addresses

Kai Gao
Sichuan University
No.24 South Section 1, Yihuan Road
Chengdu
610000
China
Email: kaigao@scu.edu.cn

Young Lee
Samsung
South Korea
Email: younglee.tx@gmail.com

Sabine Randriamasy
Nokia Bell Labs
Route de Villejust
91460 Nozay
France
Email: sabine.randriamasy@nokia-bell-labs.com

Yang Richard Yang
Yale University
51 Prospect Street
New Haven, CT
United States of America
Email: yry@cs.yale.edu

Jingxuan Jensen Zhang
Tongji University
4800 Caoan Road
Shanghai
201804
China
Email: jingxuan.n.zhang@gmail.com

ALTO Working Group
Internet-Draft
Intended status: Standards Track
Expires: 22 September 2022

Q. Wu
Huawei
Y. Yang
Yale University
Y. Lee
Samsung
D. Dhody
Huawei
S. Randriamasy
Nokia Bell Labs
L. Contreras
Telefonica
21 March 2022

ALTO Performance Cost Metrics
draft-ietf-alto-performance-metrics-28

Abstract

The cost metric is a basic concept in Application-Layer Traffic Optimization (ALTO), and different applications may use different types of cost metrics. Since the ALTO base protocol (RFC 7285) defines only a single cost metric (namely, the generic "routingcost" metric), if an application wants to issue a cost map or an endpoint cost request in order to identify a resource provider that offers better performance metrics (e.g., lower delay or loss rate), the base protocol does not define the cost metric to be used.

This document addresses this issue by extending the specification to provide a variety of network performance metrics, including network delay, delay variation (a.k.a, jitter), packet loss rate, hop count, and bandwidth.

There are multiple sources (e.g., estimation based on measurements or service-level agreement) to derive a performance metric. This document introduces an additional "cost-context" field to the ALTO "cost-type" field to convey the source of a performance metric.

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 <https://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 22 September 2022.

Copyright Notice

Copyright (c) 2022 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	3
2. Requirements Language	6
3. Performance Metric Attributes	6
3.1. Performance Metric Context: "cost-context"	7
3.2. Performance Metric Statistics	9
4. Packet Performance Metrics	11
4.1. Cost Metric: One-Way Delay (delay-ow)	11
4.1.1. Base Identifier	11
4.1.2. Value Representation	12
4.1.3. Intended Semantics and Use	12
4.1.4. Cost-Context Specification Considerations	14
4.2. Cost Metric: Round-trip Delay (delay-rt)	16
4.2.1. Base Identifier	16
4.2.2. Value Representation	16
4.2.3. Intended Semantics and Use	16
4.2.4. Cost-Context Specification Considerations	17
4.3. Cost Metric: Delay Variation (delay-variation)	18
4.3.1. Base Identifier	18
4.3.2. Value Representation	18
4.3.3. Intended Semantics and Use	18
4.3.4. Cost-Context Specification Considerations	19
4.4. Cost Metric: Loss Rate (lossrate)	20
4.4.1. Base Identifier	20
4.4.2. Value Representation	20
4.4.3. Intended Semantics and Use	20

4.4.4.	Cost-Context Specification Considerations	21
4.5.	Cost Metric: Hop Count (hopcount)	22
4.5.1.	Base Identifier	22
4.5.2.	Value Representation	22
4.5.3.	Intended Semantics and Use	22
4.5.4.	Cost-Context Specification Considerations	23
5.	Throughput/Bandwidth Performance Metrics	24
5.1.	Cost Metric: TCP Throughput (tput)	24
5.1.1.	Base Identifier	24
5.1.2.	Value Representation	24
5.1.3.	Intended Semantics and Use	24
5.1.4.	Cost-Context Specification Considerations	25
5.2.	Cost Metric: Residual Bandwidth (bw-residual)	26
5.2.1.	Base Identifier	26
5.2.2.	Value Representation	26
5.2.3.	Intended Semantics and Use	26
5.2.4.	Cost-Context Specification Considerations	28
5.3.	Cost Metric: Available Bandwidth (bw-available)	28
5.3.1.	Base Identifier	28
5.3.2.	Value Representation	28
5.3.3.	Intended Semantics and Use	29
5.3.4.	Cost-Context Specification Considerations	30
6.	Operational Considerations	30
6.1.	Source Considerations	31
6.2.	Metric Timestamp Consideration	31
6.3.	Backward Compatibility Considerations	31
6.4.	Computation Considerations	32
6.4.1.	Configuration Parameters Considerations	32
6.4.2.	Aggregation Computation Considerations	32
7.	Security Considerations	32
8.	IANA Considerations	33
9.	Acknowledgments	35
10.	References	35
10.1.	Normative References	35
10.2.	Informative References	37
	Authors' Addresses	38

1. Introduction

Application-Layer Traffic Optimization (ALTO) provides a means for network applications to obtain network information so that the applications can identify efficient application-layer traffic patterns using the networks. Cost metrics are used in both the ALTO cost map service and the ALTO endpoint cost service in the ALTO base protocol [RFC7285].

Since different applications may use different cost metrics, the ALTO base protocol introduces an ALTO Cost Metric Registry (Section 14.2 of [RFC7285]) as a systematic mechanism to allow different metrics to be specified. For example, a delay-sensitive application may want to use latency related metrics, and a bandwidth-sensitive application may want to use bandwidth related metrics. However, the ALTO base protocol has registered only a single cost metric, i.e., the generic "routingcost" metric (Section 14.2 of [RFC7285]); no latency or bandwidth related metrics are defined in the base protocol.

This document registers a set of new cost metrics (Table 1) to allow applications to determine "where" to connect based on network performance criteria including delay and bandwidth related metrics.

Metric	Definition in this doc	Semantics Based On
One-way Delay	Section 4.1	Base: [RFC7471,8570,8571] sum Unidirectional Delay
Round-trip Delay	Section 4.2	Base: Sum of two directions from above
Delay Variation	Section 4.3	Base: [RFC7471,8570,8571] sum of Unidirectional Delay Variation
Loss Rate	Section 4.4	Base: [RFC7471,8570,8571] aggr Unidirectional Link Loss
Residual Bandwidth	Section 5.2	Base: [RFC7471,8570,8571] min Unidirectional Residual BW
Available Bandwidth	Section 5.3	Base: [RFC7471,8570,8571] min Unidirectional Avail. BW
TCP Throughput	Section 5.1	[I-D.ietf-tcpm-rfc8312bis]
Hop Count	Section 4.5	[RFC7285]

Table 1. Cost Metrics Defined in this Document.

The first 6 metrics listed in Table 1 (i.e., One-way Delay, Round-trip Delay, Delay Variation, Loss Rate, Residual Bandwidth, and Available Bandwidth) are derived from the set of traffic engineering performance metrics commonly defined in OSPF [RFC3630], [RFC7471]; IS-IS [RFC5305], [RFC8570]; and BGP-LS [RFC8571]. Deriving ALTO cost performance metrics from existing network-layer traffic engineering performance metrics, to expose to application-layer traffic optimization, can be a typical mechanism by network operators to deploy ALTO [RFC7971], [FlowDirector]. This document defines the base semantics of these metrics by extending them from link metrics

to end-to-end metrics for ALTO. The "Semantics Based On" column specifies at a high level how the end-to-end metric is computed from link metrics; the details will be specified in the following sections.

The common metrics Min/Max Unidirectional Delay defined in [RFC8570][RFC8571] and Max Link Bandwidth defined in [RFC3630,RFC5305] are not listed in Table 1 because they can be handled by applying the statistical operators defined in this document. The metrics related with utilized bandwidth and reservable bandwidth (i.e., Max Reservable BW and Unreserved BW defined in [RFC3630,RFC5305]) are outside the scope of this document.

The 7th metric (the estimated TCP-flow throughput metric) provides an estimation of the bandwidth of a TCP flow, using TCP throughput modeling, to support use cases of adaptive applications [Prophet], [G2]. Note that other transport-specific metrics can be defined in the future. For example, QUIC-related metrics [RFC9000] can be considered when the methodology to measure such metrics is more mature (e.g., [I-D.corre-quick-throughput-testing]).

The 8th metric (the hop count metric) in Table 1 is mentioned in the ALTO base protocol [RFC7285], but not defined, and this document defines it to be complete.

These 8 performance metrics can be classified into two categories: those derived from the performance of individual packets (i.e., One-way Delay, Round-trip Delay, Delay Variation, Loss Rate, and Hop Count), and those related to bandwidth/throughput (Residual bandwidth, and Available Bandwidth, and TCP throughput). These two categories are defined in Sections 4 and 5 respectively. Note that all metrics except Round-trip Delay are unidirectional. An ALTO client will need to query both directions if needed.

The purpose of this document is to ensure proper usage of these 8 performance metrics in the context of ALTO. This document follows the guideline defined in Section 14.2 of the ALTO base protocol [RFC7285] on registering ALTO cost metrics. Hence, it specifies the identifier, the intended semantics, and the security considerations of each one of the metrics specified in Table 1.

The definitions of the intended semantics of the metrics tend to be coarse-grained, for guidance only, and they may work well for ALTO. On the other hand, a performance measurement framework, such as the IP Performance Measurement (IPPM) framework, may provide more details in defining a performance metric. This document introduces a mechanism called "cost-context" to provide additional details, when they are available; see Section 3.

Following the ALTO base protocol, this document uses JSON to specify the value type of each defined metric. See [RFC8259] for JSON data type specification. In particular, [RFC7285] specifies that cost values should be assumed by default as JSONNumber. When defining the value representation of each metric in Table 1, this document conforms to [RFC7285], but specifies additional, generic constraints on valid JSONNumbers for each metric. For example, each new metric in Table 1 will be specified as non-negative (≥ 0); Hop Count is specified to be an integer.

An ALTO server may provide only a subset of the metrics described in this document. For example, those that are subject to privacy concerns should not be provided to unauthorized ALTO clients. Hence, all cost metrics defined in this document are optional; not all of them need to be exposed to a given application. When an ALTO server supports a cost metric defined in this document, it announces the metric in its information resource directory (IRD) as defined in Section 9.2 of [RFC7285].

An ALTO server introducing these metrics should consider related security issues. As a generic security consideration on the reliability and trust in the exposed metric values, applications SHOULD rapidly give up using ALTO-based guidance if they detect that the exposed information does not preserve their performance level or even degrades it. Section 7 discusses security considerations in more detail.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119][RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Performance Metric Attributes

The definitions of the metrics in this document are coarse-grained, based on network-layer traffic engineering performance metrics, for guidance only. A fine-grained framework specified in [RFC6390] requires that the fine-grained specification of a network performance metric include 6 components: (i) Metric Name, (ii) Metric Description, (iii) Method of Measurement or Calculation, (iv) Units of Measurement, (v) Measurement Points, and (vi) Measurement Timing. Requiring that an ALTO server provides precise, fine-grained values for all 6 components for each metric that it exposes may not be feasible or necessary for all ALTO use cases. For example, an ALTO server computing its metrics from network-layer traffic-engineering

performance metrics may not have information about the method of measurement or calculation (e.g., measured traffic patterns).

To address the issue and realize ALTO use cases, for metrics in Table 1, this document defines performance metric identifiers which can be used in the ALTO protocol with well-defined (i) Metric Name, (ii) Metric Description, (iv) Units of Measurement, and (v) Measurement Points, which are always specified by the specific ALTO services; for example, endpoint cost service is between the two endpoints. Hence, the ALTO performance metric identifiers provide basic metric attributes.

To allow the flexibility of allowing an ALTO server to provide fine-grained information such as Method of Measurement or Calculation, according to its policy and use cases, this document introduces context information so that the server can provide these additional details.

3.1. Performance Metric Context: "cost-context"

The core additional details of a performance metric specify "how" the metric is obtained. This is referred to as the source of the metric. Specifically, this document defines three types of coarse-grained metric information sources: "nominal", and "sla" (service level agreement), and "estimation".

For a given type of source, precise interpretation of a performance metric value can depend on specific measurement and computation parameters.

To make it possible to specify the source and the aforementioned parameters, this document introduces an optional "cost-context" field to the "cost-type" field defined by the ALTO base protocol (Section 10.7 of [RFC7285]) as the following:

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

object {
  JSONString    cost-source;
  [JSONValue    parameters;]
} CostContext;
```

"cost-context" will not be used as a key to distinguish among performance metrics. Hence, an ALTO information resource MUST NOT announce multiple CostType with the same "cost-metric", "cost-mode" and "cost-context". They must be placed into different information resources.

The "cost-source" field of the "cost-context" field is defined as a string consisting of only US-ASCII alphanumeric characters (U+0030-U+0039, U+0041-U+005A, and U+0061-U+007A). The cost-source is used in this document to indicate a string of this format.

As mentioned above, this document defines three values for "cost-source": "nominal", "sla", and "estimation". The "cost-source" field of the "cost-context" field MUST be one registered in "ALTO Cost Source" registry (Section 8).

The "nominal" category indicates that the metric value is statically configured by the underlying devices. Not all metrics have reasonable "nominal" values. For example, throughput can have a nominal value, which indicates the configured transmission rate of the involved devices; latency typically does not have a nominal value.

The "sla" category indicates that the metric value is derived from some commitment which this document refers to as service-level agreement (SLA). Some operators also use terms such as "target" or "committed" values. For an "sla" metric, it is RECOMMENDED that the "parameters" field provide a link to the SLA definition.

The "estimation" category indicates that the metric value is computed through an estimation process. An ALTO server may compute "estimation" values by retrieving and/or aggregating information from routing protocols (e.g., [RFC7471], [RFC8570], [RFC8571]), traffic measurement management tools (e.g., TWAMP [RFC5357]), and measurement frameworks (e.g., IPPM), with corresponding operational issues. An illustration of potential information flows used for estimating these metrics is shown in Figure 1. Section 6 discusses in more detail the operational issues and how a network may address them.

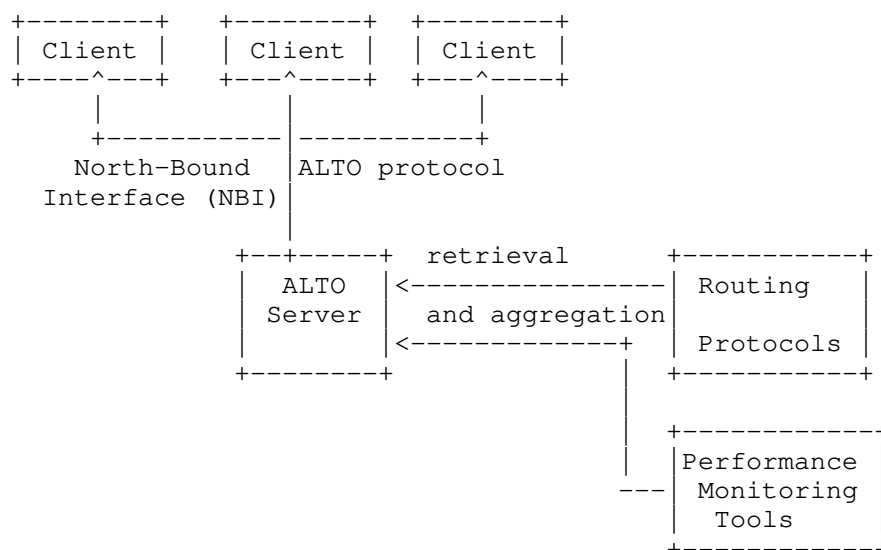


Figure 1. A framework to compute estimation to performance metrics

There can be multiple choices in deciding the cost-source category. It is the operator of an ALTO server who chooses the category. If a metric does not include a "cost-source" value, the application **MUST** assume that the value of "cost-source" is the most generic source, i.e., "estimation".

3.2. Performance Metric Statistics

The measurement of a performance metric often yields a set of samples from an observation distribution ([Prometheus]), instead of a single value. A statistical operator is applied to the samples to obtain a value to be reported to the client. Multiple statistical operators (e.g., min, median, and max) are commonly being used.

Hence, this document extends the general US-ASCII alphanumeric cost metric strings, formally specified as the CostMetric type defined in Section 10.6 of [RFC7285], as follows:

A cost metric string consists of a base metric identifier (or base identifier for short) string, followed by an optional statistical operator string, connected by the ASCII character colon (':', U+003A), if the statistical operator string exists. The total length of the cost metric string **MUST NOT** exceed 32, as required by [RFC7285].

The statistical operator string **MUST** be one of the following:

cur:

the instantaneous observation value of the metric from the most recent sample (i.e., the current value).

percentile, with letter 'p' followed by a number:

gives the percentile specified by the number following the letter 'p'. The number MUST be a non-negative JSON number in the range [0, 100] (i.e., greater than or equal to 0 and less than or equal to 100), followed by an optional decimal part, if a higher precision is needed. The decimal part should start with the '.' separator (U+002E), and followed by a sequence of one or more ASCII numbers between '0' and '9'. Assume this number is y and consider the samples coming from a random variable X. Then the metric returns x, such that the probability of X is less than or equal to x, i.e., $\text{Prob}(X \leq x) = y/100$. For example, delay-ow:p99 gives the 99% percentile of observed one-way delay; delay-ow:p99.9 gives the 99.9% percentile. Note that some systems use quantile, which is in the range [0, 1]. When there is a more common form for a given percentile, it is RECOMMENDED that the common form be used; that is, instead of p0, use min; instead of p50, use median; instead of p100, use max.

min:

the minimal value of the observations.

max:

the maximal value of the observations.

median:

the mid-point (i.e., p50) of the observations.

mean:

the arithmetic mean value of the observations.

stddev:

the standard deviation of the observations.

stdvar:

the standard variance of the observations.

Examples of cost metric strings then include "delay-ow", "delay-ow:min", "delay-ow:p99", where "delay-ow" is the base metric identifier string; "min" and "p99" are example statistical operator strings.

If a cost metric string does not have the optional statistical operator string, the statistical operator SHOULD be interpreted as the default statistical operator in the definition of the base metric. If the definition of the base metric does not provide a definition for the default statistical operator, the metric MUST be considered as the median value.

Note that RFC 7258 limits the overall cost metric identifier to 32 characters. The cost metric variants with statistical operator suffixes defined by this document are also subject to the same overall 32-character limit, so certain combinations of (long) base metric identifier and statistical operator will not be representable. If such a situation arises, it could be addressed by defining a new base metric identifier that is an "alias" of the desired base metric, with identical semantics and just a shorter name.

4. Packet Performance Metrics

This section introduces ALTO network performance metrics on one way delay, round-trip delay, delay variation, packet loss rate, and hop count. They measure the "quality of experience" of the stream of packets sent from a resource provider to a resource consumer. The measures of each individual packet (pkt) can include the delay from the time when the packet enters the network to the time when the packet leaves the network (pkt.delay); whether the packet is dropped before reaching the destination (pkt.dropped); the number of network hops that the packet traverses (pkt.hopcount). The semantics of the performance metrics defined in this section are that they are statistics computed from these measures; for example, the x-percentile of the one-way delay is the x-percentile of the set of delays {pkt.delay} for the packets in the stream.

4.1. Cost Metric: One-Way Delay (delay-ow)

4.1.1. Base Identifier

The base identifier for this performance metric is "delay-ow".

4.1.2. Value Representation

The metric value type is a single 'JSONNumber' type value conforming to the number specification of Section 6 of [RFC8259]. The unit is expressed in microseconds. Hence, the number can be a floating point number to express delay that is smaller than microseconds. The number MUST be non-negative.

4.1.3. Intended Semantics and Use

Intended Semantics: To specify the temporal and spatial aggregated delay of a stream of packets from the specified source to the specified destination. The base semantics of the metric is the Unidirectional Delay metric defined in [RFC8571,RFC8570,RFC7471], but instead of specifying the delay for a link, it is the (temporal) aggregation of the link delays from the source to the destination. A non-normative reference definition of end-to-end one-way delay is [RFC7679]. The spatial aggregation level is specified in the query context, e.g., provider-defined identifier (PID) to PID, or endpoint to endpoint, where PID is defined in Section 5.1 of [RFC7285].

Use: This metric could be used as a cost metric constraint attribute or as a returned cost metric in the response.

Example 1: Delay value on source-destination endpoint pairs

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: 239
Content-Type: application/alto-endpointcostparams+json
Accept:
  application/alto-endpointcost+json,application/alto-error+json

{
  "cost-type": {
    "cost-mode": "numerical",
    "cost-metric": "delay-ow"
  },
  "endpoints": {
    "srcs": [
      "ipv4:192.0.2.2"
    ],
    "dsts": [
      "ipv4:192.0.2.89",
      "ipv4:198.51.100.34"
    ]
  }
}
```

HTTP/1.1 200 OK
Content-Length: 247
Content-Type: application/alto-endpointcost+json

```
{
  "meta": {
    "cost-type": {
      "cost-mode": "numerical",
      "cost-metric": "delay-ow"
    }
  },
  "endpoint-cost-map": {
    "ipv4:192.0.2.2": {
      "ipv4:192.0.2.89": 10,
      "ipv4:198.51.100.34": 20
    }
  }
}
```

Note that since the "cost-type" does not include the "cost-source" field, the values are based on "estimation". Since the identifier does not include the statistical operator string component, the values will represent median values.

Example 1a below shows an example that is similar to Example 1, but for IPv6.

Example 1a: Delay value on source-destination endpoint pairs for IPv6

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: 252
Content-Type: application/alto-endpointcostparams+json
Accept:
  application/alto-endpointcost+json,application/alto-error+json
```

```
{
  "cost-type": {
    "cost-mode": "numerical",
    "cost-metric": "delay-ow"
  },
  "endpoints": {
    "srcs": [
      "ipv6:2001:db8:100::1"
    ],
    "dsts": [
      "ipv6:2001:db8:100::2",
      "ipv6:2001:db8:100::3"
    ]
  }
}
```

```
HTTP/1.1 200 OK
Content-Length: 257
Content-Type: application/alto-endpointcost+json
```

```
{
  "meta": {
    "cost-type": {
      "cost-mode": "numerical",
      "cost-metric": "delay-ow"
    }
  },
  "endpoint-cost-map": {
    "ipv6:2001:db8:100::1": {
      "ipv6:2001:db8:100::2": 10,
      "ipv6:2001:db8:100::3": 20
    }
  }
}
```

4.1.4. Cost-Context Specification Considerations

"nominal": Typically network one-way delay does not have a nominal value.

"sla": Many networks provide delay-related parameters in their application-level SLAs. It is RECOMMENDED that the "parameters" field of an "sla" one-way delay metric include a link (i.e., a field named "link") providing an URI to the specification of SLA details, if available. Such a specification can be either free text for possible presentation to the user, or a formal specification. The format of the specification is out of the scope of this document.

"estimation": The exact estimation method is out of the scope of this document. There can be multiple sources to estimate one-way delay. For example, the ALTO server may estimate the end-to-end delay by aggregation of routing protocol link metrics; the server may also estimate the delay using active, end-to-end measurements, for example, using the IPPM framework [RFC2330].

If the estimation is computed by aggregation of routing protocol link metrics (e.g., OSPF [RFC7471], IS-IS [RFC8570], or BGP-LS [RFC8571]) Unidirectional Delay link metrics, it is RECOMMENDED that the "parameters" field of an "estimation" one-way delay metric include the following information: (1) the RFC defining the routing protocol metrics (e.g., <https://www.rfc-editor.org/info/rfc7471> for RFC7471 derived metrics); (2) configurations of the routing link metrics such as configured intervals; and (3) the aggregation method from link metrics to end-to-end metrics. During aggregation from link metrics to the end-to-end metric, the server should be cognizant of potential issues when computing an end-to-end summary statistic from link statistics. The default end-to-end average one-way delay is the sum of average link one-way delays. If an ALTO server provides the min and max statistical operators for the one-way delay metric, the values can be computed directly from the routing link metrics, as [RFC7471,RFC8570,RFC8571] provide Min/Max Unidirectional Link Delay.

If the estimation is from the IPPM measurement framework, it is RECOMMENDED that the "parameters" field of an "estimation" one-way delay metric includes the following information: the URI to the URI field of the IPPM metric defined in the IPPM performance metric [IANA-IPPM] registry (e.g., https://www.iana.org/assignments/performance-metrics/OWDelay_Active_IP-UDP-Poisson-Payload250B_RFC8912sec7_Seconds_95Percentile). The IPPM metric MUST be one-way delay (i.e., IPPM OWDelay* metrics). The statistical operator of the ALTO metric MUST be consistent with the IPPM statistical property (e.g., 95-th percentile).

4.2. Cost Metric: Round-trip Delay (delay-rt)

4.2.1. Base Identifier

The base identifier for this performance metric is "delay-rt".

4.2.2. Value Representation

The metric value type is a single 'JSONNumber' type value conforming to the number specification of Section 6 of [RFC8259]. The number MUST be non-negative. The unit is expressed in microseconds.

4.2.3. Intended Semantics and Use

Intended Semantics: To specify temporal and spatial aggregated round-trip delay between the specified source and specified destination. The base semantics is that it is the sum of one-way delay from the source to the destination and the one-way delay from the destination back to the source, where the one-way delay is defined in Section 4.1. A non-normative reference definition of end-to-end round-trip delay is [RFC2681]. The spatial aggregation level is specified in the query context (e.g., PID to PID, or endpoint to endpoint).

Note that it is possible for a client to query two one-way delays (delay-ow) and then compute the round-trip delay. The server should be cognizant of the consistency of values.

Use: This metric could be used either as a cost metric constraint attribute or as a returned cost metric in the response.

Example 2: Round-trip Delay of source-destination endpoint pairs

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: 238
Content-Type: application/alto-endpointcostparams+json
Accept:
  application/alto-endpointcost+json,application/alto-error+json
```

```
{
  "cost-type": {
    "cost-mode": "numerical",
    "cost-metric": "delay-rt"
  },
  "endpoints": {
    "srcs": [
      "ipv4:192.0.2.2"
    ],
    "dsts": [
      "ipv4:192.0.2.89",
      "ipv4:198.51.100.34"
    ]
  }
}
```

```
HTTP/1.1 200 OK
Content-Length: 245
Content-Type: application/alto-endpointcost+json
```

```
{
  "meta": {
    "cost-type": {
      "cost-mode": "numerical",
      "cost-metric": "delay-rt"
    }
  },
  "endpoint-cost-map": {
    "ipv4:192.0.2.2": {
      "ipv4:192.0.2.89": 4,
      "ipv4:198.51.100.34": 3
    }
  }
}
```

4.2.4. Cost-Context Specification Considerations

"nominal": Typically network round-trip delay does not have a nominal value.

"sla": See the "sla" entry in Section 4.1.4.

"estimation": See the "estimation" entry in Section 4.1.4. For estimation by aggregation of routing protocol link metrics, the aggregation should include all links from the source to the destination and then back to the source; for estimation using IPPM, the IPPM metric MUST be round-trip delay (i.e., IPPM RTDelay* metrics). The statistical operator of the ALTO metric MUST be consistent with the IPPM statistical property (e.g., 95-th percentile).

4.3. Cost Metric: Delay Variation (delay-variation)

4.3.1. Base Identifier

The base identifier for this performance metric is "delay-variation".

4.3.2. Value Representation

The metric value type is a single 'JSONNumber' type value conforming to the number specification of Section 6 of [RFC8259]. The number MUST be non-negative. The unit is expressed in microseconds.

4.3.3. Intended Semantics and Use

Intended Semantics: To specify temporal and spatial aggregated delay variation (also called delay jitter) with respect to the minimum delay observed on the stream over the one-way delay from the specified source and destination, where the one-way delay is defined in Section 4.1. A non-normative reference definition of end-to-end one-way delay variation is [RFC3393]. Note that [RFC3393] allows the specification of a generic selection function *F* to unambiguously define the two packets selected to compute delay variations. This document defines the specific case that *F* selects as the "first" packet the one with the smallest one-way delay. The spatial aggregation level is specified in the query context (e.g., PID to PID, or endpoint to endpoint).

Note that in statistics, variations are typically evaluated by the distance from samples relative to the mean. In networking context, it is more commonly defined from samples relative to the min. This definition follows the networking convention.

Use: This metric could be used either as a cost metric constraint attribute or as a returned cost metric in the response.

Example 3: Delay variation value on source-destination endpoint pairs

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: 245
Content-Type: application/alto-endpointcostparams+json
Accept:
    application/alto-endpointcost+json,application/alto-error+json
```

```
{
  "cost-type": {
    "cost-mode": "numerical",
    "cost-metric": "delay-variation"
  },
  "endpoints": {
    "srcs": [
      "ipv4:192.0.2.2"
    ],
    "dsts": [
      "ipv4:192.0.2.89",
      "ipv4:198.51.100.34"
    ]
  }
}
```

```
HTTP/1.1 200 OK
Content-Length: 252
Content-Type: application/alto-endpointcost+json
```

```
{
  "meta": {
    "cost-type": {
      "cost-mode": "numerical",
      "cost-metric": "delay-variation"
    }
  },
  "endpoint-cost-map": {
    "ipv4:192.0.2.2": {
      "ipv4:192.0.2.89": 0,
      "ipv4:198.51.100.34": 1
    }
  }
}
```

4.3.4. Cost-Context Specification Considerations

"nominal": Typically network delay variation does not have a nominal value.

"sla": See the "sla" entry in Section 4.1.4.

"estimation": See the "estimation" entry in Section 4.1.4. For estimation by aggregation of routing protocol link metrics, the default aggregation of the average of delay variations is the sum of the link delay variations; for estimation using IPPM, the IPPM metric MUST be delay variation (i.e., IPPM OWPDV* metrics). The statistical operator of the ALTO metric MUST be consistent with the IPPM statistical property (e.g., 95-th percentile).

4.4. Cost Metric: Loss Rate (lossrate)

4.4.1. Base Identifier

The base identifier for this performance metric is "lossrate".

4.4.2. Value Representation

The metric value type is a single 'JSONNumber' type value conforming to the number specification of Section 6 of [RFC8259]. The number MUST be non-negative. The value represents the percentage of packet losses.

4.4.3. Intended Semantics and Use

Intended Semantics: To specify temporal and spatial aggregated one-way packet loss rate from the specified source and the specified destination. The base semantics of the metric is the Unidirectional Link Loss metric defined in [RFC8571,RFC8570,RFC7471], but instead of specifying the loss for a link, it is the aggregated loss of all links from the source to the destination. The spatial aggregation level is specified in the query context (e.g., PID to PID, or endpoint to endpoint).

Use: This metric could be used as a cost metric constraint attribute or as a returned cost metric in the response.

Example 5: Loss rate value on source-destination endpoint pairs

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: 238
Content-Type: application/alto-endpointcostparams+json
Accept:
  application/alto-endpointcost+json,application/alto-error+json
```

```
{
  "cost-type": {
    "cost-mode": "numerical",
    "cost-metric": "lossrate"
  },
  "endpoints": {
    "srcs": [
      "ipv4:192.0.2.2"
    ],
    "dsts": [
      "ipv4:192.0.2.89",
      "ipv4:198.51.100.34"
    ]
  }
}
```

```
HTTP/1.1 200 OK
Content-Length: 248
Content-Type: application/alto-endpointcost+json
```

```
{
  "meta": {
    "cost-type": {
      "cost-mode": "numerical",
      "cost-metric": "lossrate"
    }
  },
  "endpoint-cost-map": {
    "ipv4:192.0.2.2": {
      "ipv4:192.0.2.89": 0,
      "ipv4:198.51.100.34": 0.01
    }
  }
}
```

4.4.4. Cost-Context Specification Considerations

"nominal": Typically packet loss rate does not have a nominal value, although some networks may specify zero losses.

"sla": See the "sla" entry in Section 4.1.4..

"estimation": See the "estimation" entry in Section 4.1.4. For estimation by aggregation of routing protocol link metrics, the default aggregation of the average of loss rate is the sum of the link link loss rates. But this default aggregation is valid only if two conditions are met: (1) it is valid only when link loss rates are low, and (2) it assumes that each link's loss events are uncorrelated with every other link's loss events. When loss rates at the links are high but independent, the general formula for aggregating loss assuming each link is independent is to compute end-to-end loss as one minus the product of the success rate for each link. Aggregation when losses at links are correlated can be more complex and the ALTO server should be cognizant of correlated loss rates. For estimation using IPPM, the IPPM metric MUST be packet loss (i.e., IPPM OWLoss* metrics). The statistical operator of the ALTO metric MUST be consistent with the IPPM statistical property (e.g., 95-th percentile).

4.5. Cost Metric: Hop Count (hopcount)

The hopcount metric is mentioned in Section 9.2.3 of [RFC7285] as an example. This section further clarifies its properties.

4.5.1. Base Identifier

The base identifier for this performance metric is "hopcount".

4.5.2. Value Representation

The metric value type is a single 'JSONNumber' type value conforming to the number specification of Section 6 of [RFC8259]. The number MUST be a non-negative integer (greater than or equal to 0). The value represents the number of hops.

4.5.3. Intended Semantics and Use

Intended Semantics: To specify the number of hops in the path from the specified source to the specified destination. The hop count is a basic measurement of distance in a network and can be exposed as the number of router hops computed from the routing protocols originating this information. A hop, however, may represent other units. The spatial aggregation level is specified in the query context (e.g., PID to PID, or endpoint to endpoint).

Use: This metric could be used as a cost metric constraint attribute or as a returned cost metric in the response.

Example 4: hopcount value on source-destination endpoint pairs

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: 238
Content-Type: application/alto-endpointcostparams+json
Accept:
  application/alto-endpointcost+json,application/alto-error+json
```

```
{
  "cost-type": {
    "cost-mode": "numerical",
    "cost-metric": "hopcount"
  },
  "endpoints": {
    "srcs": [
      "ipv4:192.0.2.2"
    ],
    "dsts": [
      "ipv4:192.0.2.89",
      "ipv4:198.51.100.34"
    ]
  }
}
```

```
HTTP/1.1 200 OK
Content-Length: 245
Content-Type: application/alto-endpointcost+json
```

```
{
  "meta": {
    "cost-type": {
      "cost-mode": "numerical",
      "cost-metric": "hopcount"
    }
  },
  "endpoint-cost-map": {
    "ipv4:192.0.2.2": {
      "ipv4:192.0.2.89": 5,
      "ipv4:198.51.100.34": 3
    }
  }
}
```

4.5.4. Cost-Context Specification Considerations

"nominal": Typically hop count does not have a nominal value.

"sla": Typically hop count does not have an SLA value.

"estimation": The exact estimation method is out of the scope of this document. An example of estimating hopcounts is by importing from IGP routing protocols. It is RECOMMENDED that the "parameters" field of an "estimation" hop count define the meaning of a hop.

5. Throughput/Bandwidth Performance Metrics

This section introduces four throughput/bandwidth related metrics. Given a specified source to a specified destination, these metrics reflect the volume of traffic that the network can carry from the source to the destination.

5.1. Cost Metric: TCP Throughput (tput)

5.1.1. Base Identifier

The base identifier for this performance metric is "tput".

5.1.2. Value Representation

The metric value type is a single 'JSONNumber' type value conforming to the number specification of Section 6 of [RFC8259]. The number MUST be non-negative. The unit is bytes per second.

5.1.3. Intended Semantics and Use

Intended Semantics: To give the throughput of a TCP congestion-control conforming flow from the specified source to the specified destination. The throughput SHOULD be interpreted as only an estimation, and the estimation is designed only for bulk flows.

Use: This metric could be used as a cost metric constraint attribute or as a returned cost metric in the response.

Example 5: TCP throughput value on source-destination endpoint pairs

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: 234
Content-Type: application/alto-endpointcostparams+json
Accept:
  application/alto-endpointcost+json,application/alto-error+json
```

```
{
  "cost-type": {
    "cost-mode": "numerical",
    "cost-metric": "tput"
  },
  "endpoints": {
    "srcs": [
      "ipv4:192.0.2.2"
    ],
    "dsts": [
      "ipv4:192.0.2.89",
      "ipv4:198.51.100.34"
    ]
  }
}
```

```
HTTP/1.1 200 OK
Content-Length: 251
Content-Type: application/alto-endpointcost+json
```

```
{
  "meta": {
    "cost-type": {
      "cost-mode": "numerical",
      "cost-metric": "tput"
    }
  },
  "endpoint-cost-map": {
    "ipv4:192.0.2.2": {
      "ipv4:192.0.2.89": 256000,
      "ipv4:198.51.100.34": 128000
    }
  }
}
```

5.1.4. Cost-Context Specification Considerations

"nominal": Typically TCP throughput does not have a nominal value, and SHOULD NOT be generated.

"sla": Typically TCP throughput does not have an SLA value, and SHOULD NOT be generated.

"estimation": The exact estimation method is out of the scope of this document. It is RECOMMENDED that the "parameters" field of an "estimation" TCP throughput metric include the following information: (1) the congestion-control algorithm; and (2) the estimation methodology. To specify (1), it is RECOMMENDED that the "parameters" field (object) include a field named "congestion-control-algorithm", which provides a URI for the specification of the algorithm; for example, for an ALTO server to provide estimation to the throughput of a Cubic Congestion control flow, its "parameters" includes a field "congestion-control-algorithm", with value being set to [I-D.ietf-tcpm-rfc8312bis]; for an ongoing congestion control algorithm such as BBR, a link to its specification. To specify (2), the "parameters" includes as many details as possible; for example, for TCP Cubic throughput estimation, the "parameters" field specifies that the throughput is estimated by setting `_C_` to 0.4, and the Equation in Figure 8 of [I-D.ietf-tcpm-rfc8312bis] is applied; as an alternative, the methodology may be based on the NUM model [Prophet], or the G2 model [G2]. The exact specification of the parameters field is out of the scope of this document.

5.2. Cost Metric: Residual Bandwidth (bw-residual)

5.2.1. Base Identifier

The base identifier for this performance metric is "bw-residual".

5.2.2. Value Representation

The metric value type is a single 'JSONNumber' type value that is non-negative. The unit of measurement is bytes per second.

5.2.3. Intended Semantics and Use

Intended Semantics: To specify temporal and spatial residual bandwidth from the specified source and the specified destination. The base semantics of the metric is the Unidirectional Residual Bandwidth metric defined in [RFC8571,RFC8570,RFC7471], but instead of specifying the residual bandwidth for a link, it is the residual bandwidth of the path from the source to the destination. Hence, it is the minimal residual bandwidth among all links from the source to the destination. When the max statistical operator is defined for the metric, it typically provides the minimum of the link capacities along the path, as the default value of the residual bandwidth of a link is its link capacity [RFC8571,8570,7471]. The spatial aggregation unit is specified in the query context (e.g., PID to PID,

or endpoint to endpoint).

The default statistical operator for residual bandwidth is the current instantaneous sample; that is, the default is assumed to be "cur".

Use: This metric could be used either as a cost metric constraint attribute or as a returned cost metric in the response.

Example 7: bw-residual value on source-destination endpoint pairs

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: 241
Content-Type: application/alto-endpointcostparams+json
Accept:
  application/alto-endpointcost+json,application/alto-error+json
```

```
{
  "cost-type": {
    "cost-mode": "numerical",
    "cost-metric": "bw-residual"
  },
  "endpoints": {
    "srcs": [
      "ipv4:192.0.2.2"
    ],
    "dsts": [
      "ipv4:192.0.2.89",
      "ipv4:198.51.100.34"
    ]
  }
}
```

HTTP/1.1 200 OK
Content-Length: 255
Content-Type: application/alto-endpointcost+json

```
{
  "meta": {
    "cost-type": {
      "cost-mode": "numerical",
      "cost-metric": "bw-residual"
    }
  },
  "endpoint-cost-map": {
    "ipv4:192.0.2.2": {
      "ipv4:192.0.2.89": 0,
      "ipv4:198.51.100.34": 2000
    }
  }
}
```

5.2.4. Cost-Context Specification Considerations

"nominal": Typically residual bandwidth does not have a nominal value.

"sla": Typically residual bandwidth does not have an "sla" value.

"estimation": See the "estimation" entry in Section 4.1.4 on aggregation of routing protocol link metrics. The current ("cur") residual bandwidth of a path is the minimal of the residual bandwidth of all links on the path.

5.3. Cost Metric: Available Bandwidth (bw-available)

5.3.1. Base Identifier

The base identifier for this performance metric is "bw-available".

5.3.2. Value Representation

The metric value type is a single 'JSONNumber' type value that is non-negative. The unit of measurement is bytes per second.

5.3.3. Intended Semantics and Use

Intended Semantics: To specify temporal and spatial available bandwidth from the specified source to the specified destination. The base semantics of the metric is the Unidirectional Available Bandwidth metric defined in [RFC8571,RFC8570,RFC7471], but instead of specifying the available bandwidth for a link, it is the available bandwidth of the path from the source to the destination. Hence, it is the minimal available bandwidth among all links from the source to the destination. The spatial aggregation unit is specified in the query context (e.g., PID to PID, or endpoint to endpoint).

The default statistical operator for available bandwidth is the current instantaneous sample; that is, the default is assumed to be "cur".

Use: This metric could be used either as a cost metric constraint attribute or as a returned cost metric in the response.

Example 8: bw-available value on source-destination endpoint pairs

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: 244
Content-Type: application/alto-endpointcostparams+json
Accept:
  application/alto-endpointcost+json,application/alto-error+json

{
  "cost-type": {
    "cost-mode": "numerical",
    "cost-metric": "bw-available"
  },
  "endpoints": {
    "srcs": [
      "ipv4:192.0.2.2"
    ],
    "dsts": [
      "ipv4:192.0.2.89",
      "ipv4:198.51.100.34"
    ]
  }
}
```

HTTP/1.1 200 OK
Content-Length: 255
Content-Type: application/alto-endpointcost+json

```
{
  "meta": {
    "cost-type": {
      "cost-mode": "numerical",
      "cost-metric": "bw-available"
    }
  },
  "endpoint-cost-map": {
    "ipv4:192.0.2.2": {
      "ipv4:192.0.2.89": 0,
      "ipv4:198.51.100.34": 2000
    }
  }
}
```

5.3.4. Cost-Context Specification Considerations

"nominal": Typically available bandwidth does not have a nominal value.

"sla": Typically available bandwidth does not have an "sla" value.

"estimation": See the "estimation" entry in Section 4.1.4 on aggregation of routing protocol link metrics. The current ("cur") available bandwidth of a path is the minimum of the available bandwidth of all links on the path.

6. Operational Considerations

The exact measurement infrastructure, measurement condition, and computation algorithms can vary from different networks, and are outside the scope of this document. Both the ALTO server and the ALTO clients, however, need to be cognizant of the operational issues discussed in the following sub-sections.

Also, the performance metrics specified in this document are similar, in that they may use similar data sources and have similar issues in their calculation. Hence, this document specifies common issues unless one metric has its unique challenges.

6.1. Source Considerations

The addition of the "cost-source" field is to solve a key issue: An ALTO server needs data sources to compute the cost metrics described in this document, and an ALTO client needs to know the data sources to better interpret the values.

To avoid too fine-grained information, this document introduces "cost-source" to indicate only the high-level type of data sources: "estimation", "nominal" or "lsa", where "estimation" is a type of measurement data source, "nominal" is a type of static configuration, and "lsa" is a type that is more based on policy.

For estimation, for example, the ALTO server may use log servers or the OAM system as its data source as recommended by [RFC7971]. In particular, the cost metrics defined in this document can be computed using routing systems as the data sources.

6.2. Metric Timestamp Consideration

Despite the introduction of the additional cost-context information, the metrics do not have a field to indicate the timestamps of the data used to compute the metrics. To indicate this attribute, the ALTO server SHOULD return HTTP "Last-Modified", to indicate the freshness of the data used to compute the performance metrics.

If the ALTO client obtains updates through an incremental update mechanism [RFC8895], the client SHOULD assume that the metric is computed using a snapshot at the time that is approximated by the receiving time.

6.3. Backward Compatibility Considerations

One potential issue introduced by the optional "cost-source" field is backward compatibility. Consider that an IRD which defines two cost-types with the same "cost-mode" and "cost-metric", but one with "cost-source" being "estimation" and the other being "lsa". Then an ALTO client that is not aware of the extension will not be able to distinguish between these two types. A similar issue can arise even with a single cost-type, whose "cost-source" is "lsa": an ALTO client that is not aware of this extension will ignore this field and consider the metric estimation.

To address the backward-compatibility issue, if a "cost-metric" is "routingcost" and the metric contains a "cost-context" field, then it MUST be "estimation"; if it is not, the client SHOULD reject the information as invalid.

6.4. Computation Considerations

The metric values exposed by an ALTO server may result from additional processing on measurements from data sources to compute exposed metrics. This may involve data processing tasks such as aggregating the results across multiple systems, removing outliers, and creating additional statistics. There are two challenges on the computation of ALTO performance metrics.

6.4.1. Configuration Parameters Considerations

Performance metrics often depend on configuration parameters, and exposing such configuration parameters can help an ALTO client to better understand the exposed metrics. In particular, an ALTO server may be configured to compute a TE metric (e.g., packet loss rate) in fixed intervals, say every T seconds. To expose this information, the ALTO server may provide the client with two pieces of additional information: (1) when the metrics are last computed, and (2) when the metrics will be updated (i.e., the validity period of the exposed metric values). The ALTO server can expose these two pieces of information by using the HTTP response headers Last-Modified and Expires.

6.4.2. Aggregation Computation Considerations

An ALTO server may not be able to measure the performance metrics to be exposed. The basic issue is that the "source" information can often be link level. For example, routing protocols often measure and report only per link loss, not end-to-end loss; similarly, routing protocols report link level available bandwidth, not end-to-end available bandwidth. The ALTO server then needs to aggregate these data to provide an abstract and unified view that can be more useful to applications. The server should consider that different metrics may use different aggregation computation. For example, the end-to-end latency of a path is the sum of the latency of the links on the path; the end-to-end available bandwidth of a path is the minimum of the available bandwidth of the links on the path; in contrast, aggregating loss values is complicated by the potential for correlated loss events on different links in the path

7. Security Considerations

The properties defined in this document present no security considerations beyond those in Section 15 of the base ALTO specification [RFC7285].

However, concerns addressed in Sections 15.1, 15.2, and 15.3 of [RFC7285] remain of utmost importance. Indeed, Traffic Engineering (TE) performance is highly sensitive ISP information; therefore, sharing TE metric values in numerical mode requires full mutual confidence between the entities managing the ALTO server and the ALTO client. ALTO servers will most likely distribute numerical TE performance to ALTO clients under strict and formal mutual trust agreements. On the other hand, ALTO clients must be cognizant on the risks attached to such information that they would have acquired outside formal conditions of mutual trust.

To mitigate confidentiality risks during information transport of TE performance metrics, the operator should address the risk of ALTO information being leaked to malicious Clients or third parties, through attacks such as the person-in-the-middle (PITM) attacks. As specified in "Protection Strategies" (Section 15.3.2 of [RFC7285]), the ALTO Server should authenticate ALTO Clients when transmitting an ALTO information resource containing sensitive TE performance metrics. "Authentication and Encryption" (Section 8.3.5 of [RFC7285]) specifies that "ALTO Server implementations as well as ALTO Client implementations MUST support the "https" URI scheme of [RFC7230] and Transport Layer Security (TLS) of [RFC8446]".

8. IANA Considerations

IANA has created and now maintains the "ALTO Cost Metric" registry, listed in Section 14.2, Table 3 of [RFC7285]. This registry is located at <<https://www.iana.org/assignments/alto-protocol/alto-protocol.xhtml#cost-metrics>>. This document requests to add the following entries to the "ALTO Cost Metric" registry.

Identifier	Intended Semantics
delay-ow	Section 4.1 of [RFCXXX]
delay-rt	Section 4.2 of [RFCXXX]
delay-variation	Section 4.3 of [RFCXXX]
lossrate	Section 4.4 of [RFCXXX]
hopcount	Section 4.5 of [RFCXXX]
tput	Section 5.1 of [RFCXXX]
bw-residual	Section 5.2 of [RFCXXX]
bw-available	Section 5.3 of [RFCXXX]

* [Note to the RFC Editor]: Please replace RFCXXX with the RFC number assigned to this document.

This document requests the creation of the "ALTO Cost Source" registry. This registry serves two purposes. First, it ensures uniqueness of identifiers referring to ALTO cost source types. Second, it provides references to particular semantics of allocated cost source types to be applied by both ALTO servers and applications utilizing ALTO clients.

A new ALTO cost source can be added after IETF Review [RFC8126], to ensure that proper documentation regarding the new ALTO cost source and its security considerations have been provided. The RFC(s) documenting the new cost source 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 source should be interpreted. Updates and deletions of ALTO cost source follow the same procedure.

Registered ALTO address type identifiers MUST conform to the syntactical requirements specified in Section 3.1. Identifiers are to be recorded and displayed as strings.

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

- * Identifier: The name of the desired ALTO cost source type.
- * Intended Semantics: ALTO cost source type carry with them semantics to guide their usage by ALTO clients. Hence, a document defining a new type should provide guidance to both ALTO service providers and applications utilizing ALTO clients as to how values of the registered ALTO endpoint property should be interpreted.
- * Security Considerations: ALTO cost source types expose information to ALTO clients. ALTO service providers should be made aware of the security ramifications related to the exposure of a cost source type.

This specification requests registration of the identifiers "nominal", "sla", and "estimation" listed in the table below. Semantics for these are documented in Section 3.1, and security considerations are documented in Section 7.

Identifier	Intended Semantics	Security Considerations
nominal	Values in nominal cases; Section 3.1 of [RFCXXX]	Section 7 of [RFCXXX]
sla	Values reflecting service level agreement; Section 3.1 of [RFCXXXX]	Section 7 of [RFCXXX]
estimation	Values by estimation; Section 3.1 of [RFCXXX]	Section 7 of [RFCXXX]

9. Acknowledgments

The authors of this document would also like to thank Martin Duke for the highly informative, thorough AD reviews and comments. We thank Christian Amsuess, Elwyn Davies, Haizhou Du, Kai Gao, Geng Li, Lili Liu, Danny Alex Lachos Perez, and Brian Trammell for the reviews and comments. We thank Benjamin Kaduk, Eric Kline, Francesca Palombini, Lars Eggert, Martin Vigoureux, Murraray Kucherawy, Roman Danyliw, Zaheduzzaman Sarker, Eric Vyncke for discussions and comments that improve this document.

10. References

10.1. Normative References

- [I-D.ietf-tcpm-rfc8312bis]
 Xu, L., Ha, S., Rhee, I., Goel, V., and L. Eggert, "CUBIC for Fast and Long-Distance Networks", Work in Progress, Internet-Draft, draft-ietf-tcpm-rfc8312bis-07, 4 March 2022, <<https://www.ietf.org/archive/id/draft-ietf-tcpm-rfc8312bis-07.txt>>.
- [IANA-IPPM]
 IANA, "Performance Metrics Registry, <https://www.iana.org/assignments/performance-metrics/performance-metrics.xhtml>".
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <<https://www.rfc-editor.org/info/rfc3630>>.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <<https://www.rfc-editor.org/info/rfc5305>>.
- [RFC6390] Clark, A. and B. Claise, "Guidelines for Considering New Performance Metric Development", BCP 170, RFC 6390, DOI 10.17487/RFC6390, October 2011, <<https://www.rfc-editor.org/info/rfc6390>>.
- [RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <<https://www.rfc-editor.org/info/rfc7230>>.
- [RFC7285] Alimi, R., Ed., Penno, R., Ed., Yang, Y., Ed., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, DOI 10.17487/RFC7285, September 2014, <<https://www.rfc-editor.org/info/rfc7285>>.
- [RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", RFC 7471, DOI 10.17487/RFC7471, March 2015, <<https://www.rfc-editor.org/info/rfc7471>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <<https://www.rfc-editor.org/info/rfc8259>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.

- [RFC8570] Ginsberg, L., Ed., Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", RFC 8570, DOI 10.17487/RFC8570, March 2019, <<https://www.rfc-editor.org/info/rfc8570>>.
- [RFC8571] Ginsberg, L., Ed., Previdi, S., Wu, Q., Tantsura, J., and C. Filsfils, "BGP - Link State (BGP-LS) Advertisement of IGP Traffic Engineering Performance Metric Extensions", RFC 8571, DOI 10.17487/RFC8571, March 2019, <<https://www.rfc-editor.org/info/rfc8571>>.
- [RFC8895] Roome, W. and Y. Yang, "Application-Layer Traffic Optimization (ALTO) Incremental Updates Using Server-Sent Events (SSE)", RFC 8895, DOI 10.17487/RFC8895, November 2020, <<https://www.rfc-editor.org/info/rfc8895>>.

10.2. Informative References

- [FlowDirector] Pujol, E., Poese, I., Zerwas, J., Smaragdakis, G., and A. Feldmann, "Steering Hyper-Giants' Traffic at Scale", ACM CoNEXT 2020, 2020.
- [G2] Ros-Giralt, J., Bohara, A., Yellamraju, S., and et. al., "On the Bottleneck Structure of Congestion-Controlled Networks", ACM SIGMETRICS 2019, 2020.
- [I-D.corre-quick-throughput-testing] Corre, K., "Framework for QUIC Throughput Testing", Work in Progress, Internet-Draft, draft-corre-quick-throughput-testing-00, 17 September 2021, <<https://www.ietf.org/archive/id/draft-corre-quick-throughput-testing-00.txt>>.
- [Prometheus] Volz, J. and B. Rabenstein, "Prometheus: A Next-Generation Monitoring System", 2015.
- [Prophet] Gao, K., Zhang, J., and YR. Yang, "Prophet: Fast, Accurate Throughput Prediction with Reactive Flows", ACM/IEEE Transactions on Networking July, 2020.
- [RFC2330] Paxson, V., Almes, G., Mahdavi, J., and M. Mathis, "Framework for IP Performance Metrics", RFC 2330, DOI 10.17487/RFC2330, May 1998, <<https://www.rfc-editor.org/info/rfc2330>>.

- [RFC2681] Almes, G., Kalidindi, S., and M. Zekauskas, "A Round-trip Delay Metric for IPPM", RFC 2681, DOI 10.17487/RFC2681, September 1999, <<https://www.rfc-editor.org/info/rfc2681>>.
- [RFC3393] Demichelis, C. and P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", RFC 3393, DOI 10.17487/RFC3393, November 2002, <<https://www.rfc-editor.org/info/rfc3393>>.
- [RFC5357] Hedayat, K., Krzanowski, R., Morton, A., Yum, K., and J. Babiarz, "A Two-Way Active Measurement Protocol (TWAMP)", RFC 5357, DOI 10.17487/RFC5357, October 2008, <<https://www.rfc-editor.org/info/rfc5357>>.
- [RFC7679] Almes, G., Kalidindi, S., Zekauskas, M., and A. Morton, Ed., "A One-Way Delay Metric for IP Performance Metrics (IPPM)", STD 81, RFC 7679, DOI 10.17487/RFC7679, January 2016, <<https://www.rfc-editor.org/info/rfc7679>>.
- [RFC7971] Stiemerling, M., Kiesel, S., Scharf, M., Seidel, H., and S. Previdi, "Application-Layer Traffic Optimization (ALTO) Deployment Considerations", RFC 7971, DOI 10.17487/RFC7971, October 2016, <<https://www.rfc-editor.org/info/rfc7971>>.
- [RFC9000] Iyengar, J., Ed. and M. Thomson, Ed., "QUIC: A UDP-Based Multiplexed and Secure Transport", RFC 9000, DOI 10.17487/RFC9000, May 2021, <<https://www.rfc-editor.org/info/rfc9000>>.

Authors' Addresses

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing
Jiangsu, 210012
China
Email: bill.wu@huawei.com

Y. Richard Yang
Yale University
51 Prospect St
New Haven, CT 06520
United States of America
Email: yry@cs.yale.edu

Young Lee
Samsung
Email: young.lee@gmail.com

Dhruv Dhody
Huawei
Leela Palace
Bangalore 560008
Karnataka
India
Email: dhruv.ietf@gmail.com

Sabine Randriamasy
Nokia Bell Labs
Route de Villejust
91460 Nozay
France
Email: sabine.randriamasy@nokia-bell-labs.com

Luis Miguel Contreras Murillo
Telefonica
Madrid
Spain
Email: luismiguel.contrerasmurillo@telefonica.com

ALTO WG
Internet-Draft
Intended status: Standards Track
Expires: 1 September 2022

W. Roome
S. Randriamasy
Nokia Bell Labs
Y. Yang
Yale University
J. Zhang
Tongji University
K. Gao
Sichuan University
28 February 2022

An ALTO Extension: Entity Property Maps
draft-ietf-alto-unified-props-new-24

Abstract

This document specifies an extension to the base Application-Layer Traffic Optimization (ALTO) protocol that generalizes the concept of "endpoint properties", which were so far tied to IP addresses, to entities defined by a wide set of objects. Further, these properties are presented as maps, similar to the network and cost maps in the base ALTO protocol. While supporting the endpoints and related endpoint property service defined in RFC7285, the ALTO protocol is extended in two major directions. First, from endpoints restricted to IP addresses to entities covering a wider and extensible set of objects; second, from properties on specific endpoints to entire entity property maps. These extensions introduce additional features allowing entities and property values to be specific to a given information resource. This is made possible by a generic and flexible design of entity and property types.

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 <https://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 1 September 2022.

Copyright Notice

Copyright (c) 2022 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	4
1.1. Terminology and notation	6
2. Requirements Language	6
3. Basic Features of the Entity Property Map Extension	7
3.1. Entity	7
3.2. Entity Domain	8
3.2.1. Entity Domain Type	8
3.2.2. Entity Domain Name	8
3.3. Entity Property Type	9
3.4. New Information Resource and Media Type: ALTO Property Map	10
4. Advanced Features of the Entity Property Map Extension	10
4.1. Entity Identifier and Entity Domain Name	10
4.2. Resource-Specific Entity Domain Name	11
4.3. Resource-Specific Entity Property Value	12
4.4. Entity Hierarchy and Property Inheritance	13
4.4.1. Entity Hierarchy	13
4.4.2. Property Inheritance	14
4.4.3. Property Value Unicity	14
4.5. Supported Properties on Entity Domains in Property Map Capabilities	15
4.6. Defining Information Resource for Resource-Specific Entity Domains	15
4.6.1. Defining Information Resource and its Media Type	16
4.6.2. Examples of Defining Information Resources and Their Media Type	17
4.7. Defining Information Resource for Resource-Specific Property Values	18
5. Protocol Specification: Basic Data Types	19
5.1. Entity Domain	19
5.1.1. Entity Domain Type	19
5.1.2. Entity Domain Name	20

5.1.3.	Entity Identifier	22
5.1.4.	Hierarchy and Inheritance	23
5.2.	Entity Property	23
5.2.1.	Entity Property Type	23
5.2.2.	Entity Property Name	24
5.2.3.	Format for Entity Property Value	25
6.	Entity Domain Types Defined in this Document	25
6.1.	Internet Address Domain Types	25
6.1.1.	Entity Domain Type: IPv4	25
6.1.2.	Entity Domain Type: IPv6	26
6.1.3.	Hierarchy and Inheritance of Internet Address Domains	26
6.1.4.	Defining Information Resource Media Type for domain types IPv4 and IPv6	28
6.2.	Entity Domain Type: PID	28
6.2.1.	Entity Domain Type Identifier	28
6.2.2.	Domain-Specific Entity Identifiers	28
6.2.3.	Hierarchy and Inheritance	28
6.2.4.	Defining Information Resource Media Type for Domain Type PID	28
6.2.5.	Relationship To Internet Addresses Domains	29
6.3.	Internet Address Properties vs. PID Properties	29
7.	Property Map	29
7.1.	Media Type	30
7.2.	HTTP Method	30
7.3.	Accept Input Parameters	30
7.4.	Capabilities	30
7.5.	Uses	30
7.6.	Response	30
8.	Filtered Property Map	32
8.1.	Media Type	32
8.2.	HTTP Method	32
8.3.	Accept Input Parameters	32
8.4.	Capabilities	34
8.5.	Uses	34
8.6.	Filtered Property Map Response	34
8.7.	Entity Property Type Defined in This Document	36
8.7.1.	Entity Property Type: pid	36
9.	Impact on Legacy ALTO Servers and ALTO Clients	37
9.1.	Impact on Endpoint Property Service	37
9.2.	Impact on Resource-Specific Properties	37
9.3.	Impact on Other Properties	37
10.	Examples	38
10.1.	Network Map	38
10.2.	Property Definitions	38
10.3.	Information Resource Directory (IRD)	39
10.4.	Full Property Map Example	42
10.5.	Filtered Property Map Example #1	43

10.6.	Filtered Property Map Example #2	44
10.7.	Filtered Property Map Example #3	45
10.8.	Filtered Property Map Example #4	47
10.9.	Filtered Property Map for ANEs Example #5	47
11.	Security Considerations	48
12.	IANA Considerations	50
12.1.	application/alto-propmap+json Media Type	51
12.2.	alto-propmapparams+json Media Type	52
12.3.	ALTO Entity Domain Type Registry	53
12.3.1.	Consistency Procedure between ALTO Address Type Registry and ALTO Entity Domain Type Registry	54
12.3.2.	ALTO Entity Domain Type Registration Process	56
12.4.	ALTO Entity Property Type Registry	57
13.	Acknowledgments	58
14.	References	59
14.1.	Normative References	59
14.2.	Informative References	60
Appendix A.	Features introduced with the Entity Property Maps extension	61
Authors' Addresses	62

1. Introduction

The ALTO protocol [RFC7285] introduces the concept of "properties" attached to "endpoint addresses". It also defines the Endpoint Property Service (EPS) to allow ALTO clients to retrieve those properties. While useful, the EPS, as defined in [RFC7285], has at least three limitations that are further elaborated hereafter.

First, the EPS allows properties to be associated with only endpoints that are identified by individual communication addresses like IPv4 and IPv6 addresses. It is reasonable to think that collections of endpoints or Provider-Defined Identifiers (PIDs), may also have properties. Furthermore, recent ALTO use cases show that properties of entities such as abstracted network elements as defined in [I-D.ietf-alto-path-vector] are also useful. However, the current EPS is restricted to individual endpoints and cannot be applied to those entities.

Second, the EPS only allows endpoints identified by global communication addresses. However, an endpoint address may be a local IP address or an anycast IP address that may not be globally unique. Additionally, an entity such as a PID may have an identifier that is not globally unique. That is, a same PID may be used in multiple network maps, while in each network map, this PID points to a different set of addresses.

Third, in section 11.4 of [RFC7285], the EPS is only defined as a POST-mode service. ALTO clients must request the properties for an explicit set of endpoint addresses. By contrast, [RFC7285], in section 11.2.3, defines a GET-mode cost map resource which returns all available costs, so an ALTO Client can retrieve a full set of costs once, and then process cost lookups without querying the ALTO server. [RFC7285] does not define a similar service for endpoint properties. At first, a map of endpoint properties might seem impractical, because it could require enumerating the property value for every possible endpoint. In particular, the number of endpoint addresses involved by an ALTO server can be quite large. To avoid enumerating a large number of endpoint addresses inefficiently, the ALTO server might define properties for a sufficiently large subset of endpoints and uses an aggregation representation to reference endpoints to allow efficient enumeration. This is particularly true if blocks of endpoint addresses with a common prefix have the same value for a property. Entities in other domains may very well allow aggregated representation and hence be enumerable as well.

To address these three limitations, this document specifies an ALTO protocol extension for defining and retrieving ALTO properties:

- * The first limitation is addressed by introducing a generic concept called ALTO Entity, which generalizes an endpoint and may represent a PID, a network element, a cell in a cellular network, an abstracted network element [I-D.ietf-alto-path-vector], or other physical or logical objects involved in a network topology. Each entity is included in a collection called an ALTO entity domain. Since each ALTO entity domain includes only one type of entities, each entity domain can be classified by the type of enclosed entities.
- * The second limitation is addressed by using resource-specific entity domains. A resource-specific entity domain contains entities that are defined and identified with respect to a given ALTO information resource, which provides scoping. For example, an entity domain containing PIDs is identified with respect to the network map in which these PIDs are defined. Likewise, an entity domain containing local IP addresses may be defined with respect to a local network map.
- * The third limitation is addressed by defining two new types of ALTO information resources: Property Map (Section 7) and Filtered Property Map (Section 8). The former is a resource that is requested using the HTTP GET method, returns the property values for all entities in one or more entity domains, and is analogous to a network map or a cost map in Section 11.2 of [RFC7285]. The latter is a resource that is requested using the HTTP POST

method, returns the values for sets of properties and entities requested by the client, and is analogous to a filtered network map or a filtered cost map.

The Entity Property Maps extension described in this document introduces a number of features that are summarized in Appendix A, where Table 4 lists the features and references the sections in this document that give their high-level and their normative description.

The protocol extension defined in this document is augmentable. New entity domain types can be defined without revising the present specification. Similarly, new cost metrics and new endpoint properties can be defined in other documents without revising the protocol specification defined in [RFC7285].

1.1. Terminology and notation

This document uses the following terms and abbreviations, that will be further defined in the document. While this document introduces the feature "entity property map", it will use both the term "property map" and "entity property map" to refer to this feature.

- * Transaction: A request/response exchange between an ALTO client and an ALTO server.
- * Client: When used with a capital "C", this term refers to an ALTO client. Note that expressions "ALTO client", "ALTO Client" and "Client" are equivalent.
- * Server: When used with a capital "S", this term refers to an ALTO server. Note that expressions "ALTO server", "ALTO Server" and "Server" are equivalent.
- * EPS: An abbreviation for Endpoint Property Service.

This document uses the semi-formal notation defined in Section 8.2 of [RFC7285].

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here. When the words appear in lower case, they are to be interpreted with their natural language meanings.

3. Basic Features of the Entity Property Map Extension

This section gives a high-level overview of the basic features involved in ALTO Entity Property Maps. It assumes the reader is familiar with the ALTO protocol [RFC7285]. The purpose of this extension is to convey properties on objects that extend ALTO Endpoints and are called ALTO Entities, or entities for short.

The features introduced in this section can be used as standalone. However, in some cases, these features may depend on particular information resources and need to be defined with respect to them. To this end, Section 4 introduces additional features that extend the ones presented in the present section.

3.1. Entity

The concept of an ALTO Entity generalizes the concept of an ALTO Endpoint defined in Section 2.1 of [RFC7285]. An entity is an object that can be an endpoint defined by its network address, but can also be an object that has a defined mapping to a set of one or more network addresses or an object that is not even related to any network address. Thus, whereas all endpoints are entities, not all entities are endpoints.

Examples of entities are:

- * an ALTO endpoint that represents an application or a host identified by a communication address (e.g., an IPv4 or IPv6 address) in a network,
- * a PID, defined in [RFC7285], that has a provider defined human-readable identifier specified by an ALTO network map, which maps a PID to a set of IPv4 and IPv6 addresses,
- * an Autonomous System (AS), that has an AS number (ASN) as its identifier and maps to a set of IPv4 and IPv6 addresses, that is defined in [I-D.ietf-alto-cdni-request-routing-alto],
- * a country with a code specified in [ISO3166-1], to which applications such as CDN providers associate properties and capabilities, that is defined in [I-D.ietf-alto-cdni-request-routing-alto],
- * a TCP/UDP network flow, that is identified by a TCP/UDP 5-tuple specifying its source and destination addresses and port numbers, and the IP protocol,

- * a routing element, that is specified in [RFC7921] and is associated with routing capabilities information,
- * an abstract network element, that is specified in [I-D.ietf-alto-path-vector] and that represents an abstraction of a network part such as a router, one or more links, a network domain or their aggregation.

Some of the example entities listed above have already been documented as ALTO entities. The other examples are provided for illustration as potential entities.

3.2. Entity Domain

An entity domain defines a set of entities of the same semantic type. An entity domain is characterized by a type and identified by a name.

In this document, an entity is owned by exactly one entity domain name. An entity identifier points to exactly one entity. If two entities in two different entity domains refer to the same physical or logical object, they are treated as different entities. For example, if an end host has both an IPv4 and an IPv6 address, these two addresses will be treated as two entities, defined respectively in the "ipv4" and "ipv6" entity domains.

3.2.1. Entity Domain Type

The type of an entity domain type defines the semantics of a type of entity. Entity domain types can be defined in different documents. For example: the present document defines entity domain types "ipv4", "ipv6" and "pid" in Section 6.1 and Section 6.2. The entity domain type "ane", that defines Abstract Network Elements (ANEs), is introduced in [I-D.ietf-alto-path-vector]. The entity domain type that defines country codes is introduced in [I-D.ietf-alto-cdni-request-routing-alto]. An entity domain type MUST be registered at the IANA, as specified in Section 12.3.2.

3.2.2. Entity Domain Name

In this document, the identifier of an entity domain is mostly called "entity domain name". The identifier of an entity domain is defined in the scope of an ALTO server. An entity domain identifier can sometimes be identical to the identifier of its relevant entity domain type. This is the case when the entities of a domain have an identifier that points to the same object throughout all the information resources of the Server that provide entity properties for this domain. For example, a domain of type "ipv4" containing entities that are identified by a public IPv4 address can be named

"ipv4" because its entities are uniquely identified by all the Server resources.

In some cases, the name of an entity domain cannot be simply its entity domain type. Indeed, for some domain types, entities are defined relative to a given information resource. This is the case for entities of domain type "pid". A PID is defined relative to a network map. For example, an entity "mypid10" of domain type "pid" may be defined in a given network map and be undefined in other network maps. Or "mypid10" may even be defined in two different network maps and map, in each of these network maps, to a different set of endpoint addresses. In this case, naming an entity domain only by its type "pid" does not guarantee that its set of entities is owned by exactly one entity domain.

Section 4.2 and Section 5.1.2 describe how a domain is uniquely identified, across the ALTO server, by a name that associates the domain type and the related information resource.

3.3. Entity Property Type

An entity property defines a property of an entity. This is similar to the endpoint property defined in Section 7.1 of [RFC7285]. An entity property can convey either network-aware or network-agnostic information. Similar to an entity domain, an entity property is characterized by a type and identified by a name. An entity property type MUST be registered at the IANA, as specified in Section 12.4.

Below are listed some examples with real and fictitious entity domain and property names:

- * an entity in the "ipv4" domain type may have a property whose value is an Autonomous System (AS) number indicating the AS to which this IPv4 address belongs and another property named "countrycode" indicating a country code mapping to this address,
- * an entity identified by its country code in the entity domain type "countrycode", defined in [I-D.ietf-alto-cdni-request-routing-alto] may have a property indicating what delivery protocol is used by a CDN,
- * an entity in the "netmap1.pid" domain may have a property that indicates the central geographical location of the endpoints it includes.

It should be noted that some identifiers may be used for both an entity domain type and a property type. For example:

- * the identifier "countrycode" may point to both the entity domain type "countrycode" and the fictitious property type "countrycode".
- * the identifier "pid" may point to both the entity domain type "pid" and the property type "pid".

Likewise, the same identifier may point to both a domain name and a property name. For example: the identifier "netmap10.pid" may point to either the domain defined by the PIDs of network map "netmap10" or to a property that returns, for an entity defined by its IPv4 address, the PID of netmap10 that contains this entity. Such cases will be further explained in Section 4.

3.4. New Information Resource and Media Type: ALTO Property Map

This document introduces a new ALTO information resource named Property Map. An ALTO property map provides a set of properties on one or more sets of entities. A property may apply to different entity domain types and names. For example, an ALTO property map may define the "ASN" property for both "ipv4" and "ipv6" entity domains.

The present extension also introduces a new media type.

This document uses the same definition of an information resource as Section 9.1 of [RFC7285]. ALTO uses media types 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. In the present case, an ALTO property map resource is defined by the media type "application/alto-propmap+json".

A Property Map can be queried as a GET-mode resource, thus conveying all properties on all entities indicated in its capabilities. It can also be queried as a POST-mode resource, thus conveying a selection of properties on a selection of entities.

4. Advanced Features of the Entity Property Map Extension

This section gives a high-level overview of the advanced features involved in ALTO Entity Property Maps. Most of these features are defined to extend the ones defined in Section 3.

4.1. Entity Identifier and Entity Domain Name

In [RFC7285], an endpoint has an identifier that is explicitly associated with the "ipv4" or "ipv6" address domain. Examples are "ipv4:192.0.2.14" and "ipv6:2001:db8::12".

In this document, example IPv4 and IPv6 addresses and prefixes are taken from the address ranges reserved for documentation by [RFC5737] and [RFC3849].

In this document, an entity must be owned by exactly one entity domain name and an entity identifier must point to exactly one entity. To ensure this, an entity identifier is explicitly attached to the name of its entity domain and an entity domain type characterizes the semantics and identifier format of its entities.

The encoding format of an entity identifier is further specified in Section 5.1.3 of this document.

For instance:

- * if an entity is an endpoint with IPv4 address "192.0.2.14", its identifier is associated with entity domain name "ipv4" and is "ipv4:192.0.2.14",
- * if an entity is a PID named "mypid10" in network map resource "netmap2", its identifier is associated with entity domain name "netmap2.pid" and is "netmap2.pid:mypid10".

4.2. Resource-Specific Entity Domain Name

Some entities are defined and identified uniquely and globally in the context of an ALTO server. This is the case for instance when entities are endpoints that are identified by a reachable IPv4 or IPv6 address. The entity domain for such entities can be globally defined and named "ipv4" or "ipv6". Those entity domains are called resource-agnostic entity domains in this document, as they are not associated with any specific ALTO information resources.

Some other entities and entity types are only defined relative to a given information resource. This is the case for entities of domain type "pid", that can only be understood with respect to the network map where they are defined. For example, a PID named "mypid10" may be defined to represent a set S1 of IP addresses in a network map resource named "netmap1". Another network map "netmap2" may use the same name "mypid10" and define it to represent another set S2 of IP addresses. The identifier "pid:mypid10" may thus point to different objects because the information on the originating information resource is lost.

To solve this ambiguity, the present extension introduces the concept of resource-specific entity domain. This concept applies to domain types where entities are defined relative to a given information resource. It can also apply to entity domains that are defined locally, such as local networks of objects identified with a local IPv4 address.

In such cases, an entity domain type is explicitly associated with an identifier of the information resource where these entities are defined. Such an information resource is referred to as the "specific information resource". Using a resource-aware entity domain name, an ALTO property map can unambiguously identify distinct entity domains of the same type, on which entity properties may be queried. Examples of resource-specific entity domain names may look like: "netmap1.pid" or "netmap2.pid". Thus, a name association such as "netmap1.pid:mypid10" and "netmap2.pid:mypid10" allows to distinguish the two abovementioned PIDs that are both named "mypid10" but in two different resources, "netmap1" and "netmap2".

An information resource is defined in the scope of an ALTO Server and so is an entity domain name. The format of a resource-specific entity domain name is further specified in Section 5.1.2.

4.3. Resource-Specific Entity Property Value

Like entity domains, some types of properties are defined relative to an information resource. That is, an entity may have a property of a given type, whose values are associated to different information resources.

For example, suppose entity "192.0.2.34" defined in the "ipv4" domain has a property of type "pid", whose value is the PID to which address "192.0.2.34" is attached in a network map. The mapping of network addresses to PIDs is specific to a network map and probably different from one network map resource to another one. Thus, if a property "pid" is defined for entity "192.0.2.34" in two different network maps "netmap1" and "netmap2", the value for this property can be a different value in "netmap1" and "netmap2".

To support information resource dependent property values, this document uses the same approach as in Section 10.8.1 of [RFC7285] entitled "Resource-Specific Endpoint Properties". When a property value depends on a given information resource, the name of this property MUST be explicitly associated with the information resource that defines it.

For example, the property "pid" queried on entity "ipv4:192.0.2.34" and defined in both "netmap1" and "netmap2", can be named "netmap1.pid" and "netmap2.pid". This allows a Client to get a property of the same type but defined in different information resources with a single query. Specifications on the property name format are provided in Section 5.2.

4.4. Entity Hierarchy and Property Inheritance

For some domain types, there is an underlying structure that allows entities to efficiently be grouped into a set and be defined by the identifier of this set. This is the case for domain types "ipv4" and "ipv6", where individual Internet addresses can be grouped in blocks. When the same property value applies to a whole set, a Server can define a property for the identifier of this set instead of enumerating all the entities and their properties. This allows a substantial reduction of transmission payload both for the Server and the Client. For example, all the entities included in the set defined by the address block "ipv6:2001:db8::1/64" share the same properties and values defined for this block.

Additionally, entity sets sometimes are related by inclusion, hierarchy or other relations. This allows defining inheritance rules for entity properties that propagate properties among related entity sets. The Server and the Client can use these inheritance rules for further payload savings. Entity hierarchy and property inheritance rules are specified in the documents that define the applicable domain types. The present document defines these rules for the "ipv4" and "ipv6" domain types.

This document introduces, for applicable domain types, "Entity Property Inheritance rules", with the following concepts: Entity Hierarchy, Property Inheritance and Property Value Unicity. A detailed specification of entity hierarchy and property inheritance rules is provided in Section 5.1.4.

4.4.1. Entity Hierarchy

An entity domain may allow using a single identifier to identify a set of related individual entities. For example, a CIDR block can be used to identify a set of IPv4 or IPv6 entities. A CIDR block is called a hierarchical entity identifier, as it can reflect inclusion relations among entity sets. That is, in an entity hierarchy, "supersets" are defined at upper levels and include "subsets" defined at lower levels." For example, the CIDR "ipv4:192.0.1.0/24" includes all the individual IPv4 entities identified by the CIDR "ipv4:192.0.1.0/26". This document will sometimes use the term "hierarchical address" to refer to a hierarchical entity identifier.

4.4.2. Property Inheritance

A property may be defined for a hierarchical entity identifier, while it may be undefined for individual entities covered by this identifier. In this case, these individual entities inherit the property value defined for the identifier that covers them. For example, suppose a property map defines a property P for which it assigns value V1 only for the hierarchical entity identifier "ipv4:192.0.1.0/24" but not for individual entities in this block. Suppose also that inheritance rules are specified for CIDR blocks in the "ipv4" domain type. When receiving this property map, a Client can infer that entity "ipv4:192.0.1.1" inherits the property value V1 of block "ipv4:192.0.1.0/24" because the address "ipv4:192.0.1.1" is included in the CIDR block "ipv4:192.0.1.0/24".

Property value inheritance rules also apply among entity sets. A property map may define values for an entity set belonging to a hierarchy but not for "subsets" that are covered by this set identifier. In this case, inheritance rules must specify how entities in "subsets" inherit property values from their "superset". For instance, suppose a property P is defined only for the entity set defined by address block "ipv4:192.0.1.0/24". We know that entity set "ipv4:192.0.1.0/30" is included in "ipv4:192.0.1.0/24". Therefore, the entities of "ipv4:192.0.1.0/30" may inherit the value of property P from set "ipv4:192.0.1.0/24", if an inheritance rule from "ipv4" CIDR blocks to included "ipv4" CIDR blocks, is specified.

4.4.3. Property Value Unicity

The inheritance rules must ensure that an entity belonging to a hierarchical set of entities inherits no more than one property value, for the sake of consistency. Indeed, a property map may define a property on a hierarchy of entity sets that inherit property values from one or more supersets (located at upper levels). On the other hand, a property value, defined on a subset (located at a lower level) may be different from the value defined on a superset. In such a case, subsets may potentially end up with different property values. This may be the case for address blocs with increasing prefix length, on which a property value gets increasingly accurate and thus may differ. For example, a fictitious property such as "geo-location" or "average transfer volume" may be defined at a progressively finer grain for lower level subsets of entities, defined with progressively longer CIDR prefixes. It seems more interesting to have property values of progressively higher accuracy. A unicity rule, applied to the entity domain type must specify an arbitration rule among the different property values for an entity. An example illustrating the need for such rules is provided in Section 6.1.3.

4.5. Supported Properties on Entity Domains in Property Map Capabilities

A property type is not necessarily applicable to any domain type, or an ALTO Server may choose not to provide a property on all applicable domains. For instance, a property type reflecting link bandwidth is likely not defined on entities of a domain of type "countrycode". Therefore, an ALTO server providing Property Maps needs to specify the properties that can be queried on the different entity domains it supports.

This document explains how the Information Resources Directory (IRD) capabilities of a Property Map resource unambiguously expose what properties a Client can query on a given entity domain:

- * a field named "mappings" lists the names of the entity domains supported by the Property Map,
- * for each listed entity domain, a list of the names of the applicable properties is provided.

An example is provided in Section 10.3. The "mappings" field associates entity domains and properties that can be resource-agnostic or resource-specific. This allows a Client to formulate compact and unambiguous entity property queries, possibly relating to one or more information resources. In particular:

- * it prevents a Client from querying a property on entity domains on which it is not defined,
- * it allows a Client to query, for an entity E, values for a property P that are defined in several information resources,
- * it allows a Client to query a property P on entities that are defined in several information resources.

Further details are provided in Section 7.4.

4.6. Defining Information Resource for Resource-Specific Entity Domains

A Client willing to query properties on entities belonging to a domain needs to know how to retrieve these entities. To this end, the Client can look up the "mappings" field exposed in IRD capabilities of a property map, see Section 4.5. This field, in its keys, exposes all the entity domains supported by the property map. The syntax of the entity domain identifier specified in Section 5.1.2 allows the client to infer whether the entity domain is resource-specific or not. The Client can extract, if applicable, the

identifier of the specific resource, query the resource and retrieve the entities. For example:

- * an entity domain named "netmap1.ipv4" includes the IPv4 addresses that appear in the "ipv4" field of the endpoint address group of each PID in the network map "netmap1", and that have no meaning outside "netmap1" because, for instance, these are local addresses not reachable outside some private network,
- * an entity domain named "netmap1.pid" includes the PIDs listed in network map "netmap1".
- * an entity domain named "ipv4" is resource-agnostic and covers all the reachable IPv4 addresses.

Besides, it is not possible to prevent a Server from mistakenly exposing inappropriate associations of information resources and entity domain types. To prevent failures due to invalid queries, it is necessary to inform the Client about which associations are allowed. An informed Client will just ignore inappropriate associations exposed by a Server and avoid error-prone transactions with the Server.

For example, the association "costmap3.pid" is not allowed for the following reason: although a cost map exposes PID identifiers, it does not define the set of addresses included in this PID. Neither does a cost map list all the PIDs on which properties can be queried, because a cost map only exposes PID pairs on which a queried cost type is defined. Therefore, the resource "costmap3" does not enable a Client to extract information on the existing PID entities or on the addresses they contain.

Instead, the cost map uses a network map, where all the PIDs used in a cost map are defined together with the addresses contained by the PIDs. This network map is qualified in this document as the Defining Information Resource for the entity domain of type "pid" and this concept is explained in Section 4.6.1.

4.6.1. Defining Information Resource and its Media Type

For the reasons explained in Section 4.6, this document introduces the concept of "Defining Information Resource and its Media Type".

A defining information resource for an entity domain D is the information resource where entities of D are defined. That is, all the information on the entities of D can be retrieved in this resource. A defining information resource is defined for resource-specific entity domains. It does not exist for entity domains that

are not resource-specific such as "ipv4" or "ipv6". Neither does it exist for entity domains that are covering entity identifiers already defined in other standardization documents, at it is the case for country code identifiers standardized in [ISO3166-1] or AS numbers allocated by the IANA. This is useful for entity domain types that are by essence domain-specific, such as the "pid" domain type. It is also useful for resource-specific entity domains constructed from resource-agnostic domain types, such as network map specific domains of local IPv4 addresses.

The defining information resource of a resource-specific entity domain D, when it exists, is unique and has the following specificities:

- * it has an entry in the IRD,
- * it defines the entities of D,
- * it does not use another information resource that defines these entities,
- * it defines and exposes entity identifiers that are all persistent,
- * its media type is equal to the one that is specified for the defining information resource of an entity domain type.

A fundamental characteristic of a defining information resource is its media type. There is a unique association between an entity domain type and the media type of its defining information resource. When an entity domain type allows associations with defining information resources, the media type of the potential defining information resource MUST be specified:

- * in the document that defines this entity domain type,
- * in the IANA ALTO Entity Domain Type Registry and related information.

When the Client wants to use a resource-specific entity domain, it needs to be cognizant of the media-type of its defining information resource. If the Server exposes a resource-specific entity domain with a non-compliant media type for the defining resource, the Client MUST ignore the entities from that entity domain to avoid errors.

4.6.2. Examples of Defining Information Resources and Their Media Type

Here are examples of defining information resource types and their media types associated to different entity domain types:

- * For entity domain type "pid": the media type of the specific resource is "application/alto-networkmap+json", because PIDs are defined in network map resources.
- * For entity domain types "ipv4" and "ipv6": the media type of the specific resource is "application/alto-networkmap+json", because IPv4 and IPv6 addresses covered by the Server are defined in network map resources.
- * For entities of domain type "ane": [I-D.ietf-alto-path-vector] defines entities named "ANE", where ANE stands for Abstracted Network Element, and the entity domain type "ane". An ANE may have a persistent identifier, say, "entity-4", that is provided by the Server as a value of the "persistent-entity-id" property of this ANE. Further properties may then be queried on an ANE by using its persistent entity ID. These properties are available from a persistent property map, that defines properties on a specific "ane" domain. Together with the persistent identifier, the Server also provides the property map resource identifier where the "ane" domain containing "entity-4" is defined. The definition of the "ane" entity domain containing "entity-4" is thus specific to the property map. Therefore, for entities of domain type "ane" that have a persistent identifier, the media type of the defining information resource is "application/alto-propmap+json".
- * Last, the entity domain types "asn" and "countrycode" defined in [I-D.ietf-alto-cdni-request-routing-alto] do not have a defining information resource. Indeed, the entity identifiers in these two entity domain types are already standardized in documents that the Client can use.

4.7. Defining Information Resource for Resource-Specific Property Values

As explained in Section 4.3, a property type may take values that are resource-specific. This is the case for property type "pid", whose values are by essence defined relative to a specific network map. That is, the PID value returned for an IPv4 address is specific to the network map defining this PID and may differ from one network map to another one.

Another example is provided in [I-D.ietf-alto-cdni-request-routing-alto] that defines property type "cdni-capabilities". The value of this property is specific to a CDNI advertisement resource, that provides a list of CDNI capabilities. The property is provided for entity domain types "ipv4", "ipv6", "asn" and "countrycode". A CDNI Advertisement resource does however not define PID values for IPv4 addresses while a network map does not define CDNI capabilities for IPv4 addresses.

Similar to resource-specific entity domains, the Client needs to be cognizant of appropriate associations of information resource and property types. Therefore, when specifying and registering a property type whose values are resource-specific, the media type of its defining information resource needs to be specified. For example:

- * The media type of the defining information resource for property type "pid" is "application/alto-networkmap+json".
- * The media type of the defining information resource for property type "cdni-capabilities" defined in [I-D.ietf-alto-cdni-request-routing-alto] is "application/alto-cdni+json".

5. Protocol Specification: Basic Data Types

5.1. Entity Domain

5.1.1. Entity Domain Type

An entity domain has a type, which is uniquely identified by a string that MUST be no more than 64 characters, and MUST NOT contain characters other than US-ASCII alphanumeric characters (U+0030-U+0039, U+0041-U+005A, and U+0061-U+007A), the hyphen ('-', U+002D), the colon (':', U+003A), or the low line ('_', U+005F).

The usage of colon (':', U+003A) MUST obey the rules below:

- * The colon (':', U+003A) character MUST NOT appear more than once,
- * The colon character MUST NOT be used unless within the string "priv:",
- * The string "priv:" MUST NOT be used unless it starts the string that identifies an entity domain type,

- * For an entity domain type identifier with the "priv:" prefix , an additional string (e.g., company identifier or random string) MUST follow "priv:", to reduce potential collisions.

For example, the strings "ipv4", "ipv6", "pid" and "priv:example-test-edt", are valid entity domain types. "ipv4.anycast", "pid.local" and "priv:" are invalid.

Although "_", "-", "__--" are valid entity domain types, it is desirable to add characters such as alphanumeric ones, for better intelligibility.

The type EntityDomainType is used in this document to denote a JSON string meeting the preceding requirements.

An entity domain type defines the semantics of a type of entity, independently of any specifying resource. All entity domain types that are not prefixed with "priv:" MUST be registered with the IANA, in the "ALTO Entity Domain Type Registry", defined in Section 12.3, following the procedure specified in Section 12.3.2 of this document. The format of the entity identifiers (see Section 5.1.3) in that entity domain type, as well as any hierarchical or inheritance rules (see Section 5.1.4) for those entities, MUST be specified in the IANA registration.

Entity domain type identifiers prefixed with "priv:" are reserved for Private Use (see [RFC8126]) without a need to register with IANA. The definition of a private use entity domain type MUST apply the same way in all property maps of an IRD where it is present.

5.1.2. Entity Domain Name

As discussed in Section 3.2, an entity domain is characterized by a type and identified by a name.

This document distinguishes three categories of entity domains: resource-specific entity domains, resource-agnostic entity domains and self-defined entity domains. Their entity domain names are constructed as specified in the following sub-sections.

Each entity domain is identified by a unique entity domain name. Borrowing the symbol "::=" from the Backus-Naur Form notation [RFC5511], the format of an entity domain name is defined as follows:

EntityDomainName ::= [[ResourceID] '.'] EntityDomainType

The presence and construction of the component

"[[ResourceID] '.']"

depends on the category of entity domain.

Note that the '.' separator is not allowed in EntityDomainType and hence there is no ambiguity on whether an entity domain name refers to a resource-agnostic entity domain or a resource-specific entity domain.

Note also that Section 10.1 of [RFC7285] specifies the format of the PID name which is the format of the resource ID including the following specification: "the '.' separator is reserved for future use and MUST NOT be used unless specifically indicated in this document, or an extension document". The present extension keeps the format specification of [RFC7285], hence the '.' separator MUST NOT be used in an information resource ID.

5.1.2.1. Resource-specific Entity Domain

A resource-specific entity domain is identified by an entity domain name constructed as follows. It MUST start with a resource ID using the ResourceID type defined in Section 10.2 of [RFC7285], followed by the '.' separator (U+002E), followed by a string of the type EntityDomainType specified in Section 5.1.1.

For example, if an ALTO server provides two network maps "netmap-1" and "netmap-2", these network maps can define two resource-specific domains of type "pid", respectively identified by "netmap-1.pid" and "netmap-2.pid".

5.1.2.2. Resource-agnostic Entity Domain

A resource-agnostic entity domain contains entities that are identified independently of any information resource. The identifier of a resource-agnostic entity domain is simply the identifier of its entity domain type. For example, "ipv4" and "ipv6" identify the two resource-agnostic Internet address entity domains defined in Section 6.1.

5.1.2.3. Self-defined Entity Domain

A property map can define properties on entities that are specific to a unique information resource, which is the property map itself. This may be the case when an ALTO Server provides properties on a set of entities that are defined only in this property map, are not relevant to another one and do not depend on another specific resource.

For example: a specialised property map may define a domain of type "ane", defined in [I-D.ietf-alto-path-vector], that contains a set of ANEs representing data centers, that each have a persistent identifier and are relevant only to this property map.

In this case, the entity domain is qualified as "self-defined". The identifier of a self-defined entity domain can be of the format:

EntityDomainName ::= '.' EntityDomainType

where '.' indicates that the entity domain only exists within the property map resource using it.

A self-defined entity domain can be viewed as a particular case of resource-specific entity domain, where the specific resource is the current resource that uses this entity domain. In that case, for the sake of simplification, the component "ResourceID" MUST be omitted in its entity domain name.

5.1.3. Entity Identifier

Entities in an entity domain are identified by entity identifiers (EntityID) of the following format:

EntityID ::= EntityDomainName ':' DomainTypeSpecificEntityID

Examples from the Internet address entity domains include individual IP addresses such as "net1.ipv4:192.0.2.14" and "net1.ipv6:2001:db8::12", as well as address blocks such as "net1.ipv4:192.0.2.0/26" and "net1.ipv6:2001:db8::/48".

The format of the second part of an entity identifier, DomainTypeSpecificEntityID, depends on the entity domain type, and MUST be specified when defining a new entity domain type and registering it with the IANA. Identifiers MAY be hierarchical, and properties MAY be inherited based on that hierarchy. The rules defining any hierarchy or inheritance MUST be defined when the entity domain type is registered.

The type EntityID is used in this document to denote a JSON string representing an entity identifier in this format.

Note that two entity identifiers with different valid textual representations may refer to the same entity, for a given entity domain. For example, the strings "net1.ipv6:2001:db8::1" and "net1.ipv6:2001:db8:0:0:0:0:1" refer to the same entity in the "ipv6" entity domain. Such equivalences should be established by the object represented by DomainTypeSpecificEntityID, for example, [RFC5952] establishes equivalence for IPv6 addresses, while [RFC4632] does so for IPv4 addresses.

5.1.4. Hierarchy and Inheritance

To simplify the representation, some types of entity domains allow the ALTO Client and Server to use a hierarchical entity identifier format to represent a block of individual entities. For instance, in an IPv4 domain "net1.ipv4", a CIDR "net1.ipv4:192.0.2.0/26" covers 64 individual IPv4 entities. In this case, the corresponding property inheritance rule MUST be defined for the entity domain type. The hierarchy and inheritance rule MUST have no ambiguity.

5.2. Entity Property

Each entity property has a type to indicate the encoding and the semantics of the value of this entity property, and has a name to identify it.

5.2.1. Entity Property Type

The type EntityPropertyType is used in this document to indicate a string denoting an entity property type. The string MUST be no more than 32 characters, and it MUST NOT contain characters other than US-ASCII alphanumeric characters (U+0030-U+0039, U+0041-U+005A, and U+0061-U+007A), the hyphen ('-', U+002D), the colon (':', U+003A), or the low line ('_', U+005F). Note that the '.' separator is not allowed because it is reserved to separate an entity property type and an information resource identifier when an entity property is resource-specific.

While, in Section 5.1.1, character ":" is allowed with restrictions on entity domain identifiers, it can be used without restrictions on entity property type identifiers. This relates to [RFC7285], where a Server can define properties on endpoints "ipv4" and "ipv6". In the present extension, there is a mapping of ALTO entity domain types "ipv4" and "ipv6", to ALTO address types "ipv4" and "ipv6". Properties defined on "ipv4" and "ipv6" endpoints should be re-usable on "ipv4" and "ipv6" entities. Forbidding the usage of ":" in a non-private entity property type identifier would not allow to use properties previously defined on "ipv4" and "ipv6" endpoints because their identifiers would be invalid.

Although ":" or "::-" are valid entity domain types, it is desirable to add characters such as alphanumeric ones, for better intelligibility.

Identifiers prefixed with "priv:" are reserved for Private Use [RFC8126] without a need to register with IANA. All other identifiers for entity property types MUST be registered in the "ALTO Entity Property Type Registry", defined in Section 12.4. The intended semantics of the entity property type MUST be specified in the IANA registration.

For an entity property identifier with the "priv:" prefix, an additional string (e.g., company identifier or random string) MUST follow the prefix to reduce potential collisions, that is, the string "priv:" alone is not a valid entity property identifier. The definition of a private use entity property type must apply the same way in all property maps of an IRD where it is present.

To distinguish from the endpoint property type, the entity property type has the following characteristics:

- * Some entity property types are applicable to entities in particular entity domain types only. For example, the property type "pid" is applicable to entities in the entity domain types "ipv4" or "ipv6" while is not applicable to entities in an entity domain of type "pid".
- * The intended semantics of the value of an entity property may also depend on the entity domain type. For example, suppose that a property named "geo-location" is defined as the coordinates of a point, encoded as: "latitude longitude [altitude]". When applied to an entity that represents a specific host computer, identified by an address in an entity domain of type "ipv4" or "ipv6", the "geo-location" property would define the host's location. However, when applied to an entity in a "pid" domain type, the property would indicate a location representative of all hosts in this "pid" entity.

5.2.2. Entity Property Name

Each entity property is identified by an entity property name, which is a string of the following format:

EntityPropertyName ::= [[ResourceID] '.'] EntityPropertyType

Similar to the endpoint property type defined in Section 10.8 of [RFC7285], each entity property may be defined by either the property map itself (self-defined) or some other specific information resource (resource-specific).

The entity property name of a resource-specific entity property starts with a string of the type ResourceID defined in [RFC7285], followed by the '.' separator (U+002E) and a EntityDomainType typed string. For example, the "pid" properties of an "ipv4" entity defined by two different maps "net-map-1" and "net-map-2" are identified by "net-map-1.pid" and "net-map-2.pid" respectively.

The specific information resource of an entity property may be the current information resource itself, that is, the property map defining the property. In that case, the ResourceID in the property name SHOULD be omitted. For example, the property name ".asn" applied to an entity identified by its IPv4 address, indicates the AS number of the AS that "owns" the entity, where the returned AS number is defined by the property map itself.

5.2.3. Format for Entity Property Value

Section 11.4.1.6 of [RFC7285] specifies that an implementation of the Endpoint Property Service specified in [RFC7285] SHOULD assume that the property value is a JSONString and fail to parse if it is not. This document extends the format of a property value by allowing it to be a JSONValue instead of just a JSONString.

6. Entity Domain Types Defined in this Document

The definition of each entity domain type MUST include (1) the entity domain type name and (2) domain-specific entity identifiers, and MAY include (3) hierarchy and inheritance semantics optionally. This document defines three initial entity domain types as follows.

6.1. Internet Address Domain Types

The document defines two entity domain types (IPv4 and IPv6) for Internet addresses. Both types are resource-agnostic entity domain types and hence define corresponding resource-agnostic entity domains as well. Since the two domains use the same hierarchy and inheritance semantics, we define the semantics together, instead of repeating for each.

6.1.1. Entity Domain Type: IPv4

6.1.1.1. Entity Domain Type Identifier

The identifier for this Entity Domain Type is "ipv4".

6.1.1.2. Domain-Specific Entity Identifiers

Individual addresses are strings as specified by the IPv4address rule in Section 3.2.2 of [RFC3986]; Hierarchical addresses are strings as specified by the prefix notation in Section 3.1 of [RFC4632]. To define properties, an individual Internet address and the corresponding full-length prefix are considered aliases for the same entity on which to define properties. Thus, "ipv4:192.0.2.0" and "ipv4:192.0.2.0/32" are equivalent.

6.1.2. Entity Domain Type: IPv6

6.1.2.1. Entity Domain Type Identifier

The identifier for this Entity Domain Type is "ipv6".

6.1.2.2. Domain-Specific Entity Identifiers

Individual addresses are strings as specified by Section 4 of [RFC5952]; Hierarchical addresses are strings as specified by IPv6 address prefixes notation in Section 2.3 of [RFC4291]. To define properties, an individual Internet address and the corresponding 128-bit prefix are considered aliases for the same entity. That is, "ipv6:2001:db8::1" and "ipv6:2001:db8::1/128" are equivalent, and have the same set of properties.

6.1.3. Hierarchy and Inheritance of Internet Address Domains

Both Internet address domains allow property values to be inherited. Specifically, if a property P is not defined for a specific Internet address I, but P is defined for a hierarchical Internet address C which represents a set of addresses containing I, then the address I inherits the value of P defined for the hierarchical address C. If more than one such hierarchical addresses define a value for P, I inherits the value of P in the hierarchical address with the longest prefix. Note that this longest prefix rule ensures no multiple value inheritances, and hence no ambiguity.

Hierarchical addresses can also inherit properties: if a property P is not defined for the hierarchical address C, but is defined for a set of hierarchical addresses, where each address C' in the set contains all IP addresses in C, and C' has a shorter prefix length than C, then C MUST inherit the property P from the C' having the longest prefix length.

As an example, suppose that a server defines a property P for the following entities:

```
ipv4:192.0.2.0/26: P=v1
ipv4:192.0.2.0/28: P=v2
ipv4:192.0.2.0/30: P=v3
ipv4:192.0.2.0:    P=v4
```

Figure 1: Defined Property Values.

Then the following entities have the indicated values:

```
ipv4:192.0.2.0:    P=v4
ipv4:192.0.2.1:    P=v3
ipv4:192.0.2.16:   P=v1
ipv4:192.0.2.32:   P=v1
ipv4:192.0.2.64:   (not defined)
ipv4:192.0.2.0/32: P=v4
ipv4:192.0.2.0/31: P=v3
ipv4:192.0.2.0/29: P=v2
ipv4:192.0.2.0/27: P=v1
ipv4:192.0.2.0/25: (not defined)
```

Figure 2: Inherited Property Values.

An ALTO server MAY explicitly indicate a property as not having a value for a particular entity. That is, a server MAY say that property P of entity X is "defined to have no value", instead of "undefined". To indicate "no value", a server MAY perform different behaviors:

- * If entity X would inherit a value for property P, and if the ALTO server decides to say that "X has no value for P", then the ALTO server MUST return a "null" value for that property on X. In this case, the ALTO client MUST recognize the JSON "null" value as "no value" and interpret it as "do not apply the inheritance rules for this property on X".
- * If the entity would not inherit a value, then the ALTO server MAY return "null" or just omit the property. In this case, the ALTO client cannot infer the value for this property of this entity from the Inheritance rules. So, the client MUST interpret that this property has no value.

If the ALTO server does not define any properties for an entity, then the server MAY omit that entity from the response.

6.1.4. Defining Information Resource Media Type for domain types IPv4 and IPv6

Entity domain types "ipv4" and "ipv6" both allow to define resource specific entity domains. When resource specific domains are defined with entities of domain type "ipv4" or "ipv6", the defining information resource for an entity domain of type "ipv4" or "ipv6" MUST be a Network Map. The media type of a defining information resource is therefore:

application/alto-networkmap+json

6.2. Entity Domain Type: PID

The PID entity domain associates property values with the PIDs in a network map. Accordingly, this entity domain always depends on a network map.

6.2.1. Entity Domain Type Identifier

The identifier for this Entity Domain Type is "pid"

6.2.2. Domain-Specific Entity Identifiers

The entity identifiers are the PID names of the associated network map.

6.2.3. Hierarchy and Inheritance

There are no hierarchy or inheritance for properties associated with PIDs.

6.2.4. Defining Information Resource Media Type for Domain Type PID

The entity domain type "pid" allows to define resource specific entity domains. When resource specific domains are defined with entities of domain type "pid", the defining information resource for entity domain type "pid" MUST be a Network Map. The media type of a defining information resource is therefore:

application/alto-networkmap+json

6.2.5. Relationship To Internet Addresses Domains

The PID domain and the Internet address domains are completely independent; the properties associated with a PID have no relation to the properties associated with the prefixes or endpoint addresses in that PID. An ALTO server MAY choose to assign all the properties of a PID to the prefixes in that PID or only some of these properties.

For example, suppose "PID1" consists of the prefix "ipv4:192.0.2.0/24", and has the property "P" with value "v1". The Internet address entities "ipv4:192.0.2.0" and "ipv4:192.0.2.0/24" in the IPv4 domain MAY have a value for the property "P", and if they do, it is not necessarily "v1".

6.3. Internet Address Properties vs. PID Properties

Because the Internet address and PID domains relate to completely distinct domain types, the question may arise as to which entity domain type is the best for a property. In general, the Internet address domain types are RECOMMENDED for properties that are closely related to the Internet address, or are associated with, and inherited through, hierarchical addresses.

The PID domain type is RECOMMENDED for properties that arise from the definition of the PID, rather than from the Internet address prefixes in that PID.

For example, because Internet addresses are allocated to service providers by blocks of prefixes, an "ISP" property would be best associated with Internet address domain types. On the other hand, a property that explains why a PID was formed, or how it relates to a provider's network, would best be associated with the PID domain type.

7. Property Map

A property map returns the properties defined for all entities in one or more domains, e.g., the "location" property of entities in "pid" domain, and the "ASN" property of entities in "ipv4" and "ipv6" domains. Section 10.4 gives an example of a property map request and its response.

Downloading the whole property map is a way for the Client to obtain the Entity IDs that can be used as input for a Filtered Property Map request. However, a whole property map may be too voluminous for a Client that only wants the list of applicable Entity IDs. How to obtain the list of entities of a filtered property map in a simplified response is specified in Section 8.

7.1. Media Type

The media type of a property map is "application/alto-propmap+json".

7.2. HTTP Method

The property map is requested using the HTTP GET method.

7.3. Accept Input Parameters

A Property Map has no Accept Input parameters.

7.4. Capabilities

The capabilities are defined by an object of type `PropertyMapCapabilities`:

```
object {  
  EntityPropertyMapping mappings;  
} PropertyMapCapabilities;  
  
object-map {  
  EntityDomainName -> EntityPropertyName<1..*>;  
} EntityPropertyMapping
```

with fields:

`mappings`: A JSON object whose keys are names of entity domains and values are the supported entity properties of the corresponding entity domains.

7.5. Uses

The "uses" field of a property map resource in an IRD entry specifies the resources in this same IRD on which this property map directly depends. It is an array of resource ID(s). This array identifies the defining information resources associated with the resource-specific entity domains and properties that are indicated in this resource.

7.6. Response

If the entity domains in this property map depend on other resources, the "dependent-vtags" field in the "meta" field of the response MUST be an array that includes the version tags of those resources, and the order MUST be consistent with the "uses" field of this property map resource. The data component of a property map response is named "property-map", which is a JSON object of type `PropertyMapData`,

where:

```
object {  
  PropertyMapData property-map;  
} InfoResourceProperties : ResponseEntityBase;  
  
object-map {  
  EntityID -> EntityProps;  
} PropertyMapData;  
  
object {  
  EntityPropertyName -> JSONValue;  
} EntityProps;
```

The ResponseEntityBase type is defined in Section 8.4 of [RFC7285].

Specifically, a PropertyMapData object has one member for each entity in the property map. The entity's properties are encoded in the corresponding EntityProps object. EntityProps encodes one name/value pair for each property, where the property names are encoded as strings of type PropertyName. A protocol implementation SHOULD assume that the property value is either a JSONString or a JSON "null" value, 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.

For each entity in the property map:

- * If the entity is in a resource-specific entity domain, the ALTO server MUST only return self-defined properties and resource-specific properties which depend on the same resource as the entity does. The ALTO client MUST ignore any resource-specific property for this entity if the mapping between this resource-specific property and this entity is not indicated, in the IRD, in the "mappings" capability of the property map resource.
- * If the entity identifier is resource-agnostic, the ALTO server SHOULD return the self-defined properties and all the resource-specific properties that are defined in the property defining information resources indicated, in the IRD, in the "mappings" capability of the property map resource, unless property values can be omitted upon some inheritance rules.

The ALTO server MAY omit property values that are inherited rather than explicitly defined, in order to achieve more compact encoding. As a consequence, the ALTO Client MUST NOT assume inherited property values will all be present. If the Client needs inherited values, it MUST use the entity domain's inheritance rules to deduce those values.

8. Filtered Property Map

A filtered property map returns the values of a set of properties for a set of entities selected by the client.

Section 10.5, Section 10.6, Section 10.7 and Section 10.8 give examples of filtered property map requests and responses.

While the IRD lists all the names of the supported properties, it only lists the names of the supported entity domains and not the entity IDs. A client, sometimes, may only want to know what entity IDs it can provide as input to a filtered property map request but wants to avoid the burden of downloading the full property map. Or it may want to check whether some given entity IDs are eligible for a query. To support such a case, the filtered property map supports a light weight response, with empty property values.

8.1. Media Type

The media type of a property map resource is "application/alto-propmap+json".

8.2. HTTP Method

The filtered property map is requested using the HTTP POST method.

8.3. Accept Input Parameters

The input parameters for a filtered property map request 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-propmapparams+json", which is a JSON object of type ReqFilteredPropertyMap. ReqFilteredPropertyMap is designed to support the following cases of client requests:

- * The client wants the value of a selected set of properties on a selected set of entities,
- * The client wants all properties values on all the entities,

- * The client wants all entities on which a property is defined but is not interested in their property values,
- * The Client wants to cross-check whether some entity IDs are present in the Filtered Property Map but is not interested in their property values.

The third case is equivalent to querying the whole unfiltered property map, which can also be achieved with a GET request. Some Clients however, may prefer to systematically make filtered property map queries, where filtering parameters may sometimes be empty.

The JSON object `ReqFilteredPropertyMap` is specified as follows:

```
object {  
  EntityID          entities<0..*>;  
  [EntityPropertyName  properties<0..*>;]  
} ReqFilteredPropertyMap;
```

with fields:

entities: List of entity identifiers for which the specified properties are to be returned. If the list is empty, the ALTO Server MUST interpret the list as if it contained a list of all entities currently defined in the filtered property map. The domain of each entity MUST be included in the list of entity domains in this resource's "capabilities" field (see Section 8.4). The ALTO server MUST interpret entries appearing multiple times as if they appeared only once.

properties: List of properties to be returned for each entity. If the list is empty, the ALTO Server MUST interpret the list as if it contained a list of all properties currently defined in the filtered property map. Each specified property MUST be included in the list of properties in this resource's "capabilities" field (see Section 8.4). The ALTO server MUST interpret entries appearing multiple times as if they appeared only once. This field is optional. If it is absent, the Server returns a property value equal to the literal string "{}" for all the entity IDs of the "entities" field on which at least one property is defined.

Note that the field "properties" is optional. When in addition, the "entities" field is an empty list, it corresponds to a query for all applicable entity IDs of the filtered property map, with no current interest on any particular property. When the "entities" field is not empty, it allows the Client to check whether the listed entity IDs can be used as input to a filtered property map query.

8.4. Capabilities

The capabilities are defined by an object of type `PropertyMapCapabilities`, as defined in Section 7.4.

8.5. Uses

Same to the "uses" field of the Property Map resource (see Section 7.5).

8.6. Filtered Property Map Response

The response MUST indicate an error, using ALTO protocol error handling, as defined in Section 8.5 of [RFC7285], if the request is invalid.

Specifically, a filtered property map request can be invalid in the following cases:

- * The input field "entities" is absent from the Client request. In this case, the Server MUST return an "E_MISSING_FIELD" error as defined in Section 8.5.2 of [RFC7285].
- * An entity identifier in the "entities" field of the request is invalid. This occurs when:
 - The domain of this entity is not defined in the "entity-domains" capability of this resource in the IRD,
 - The entity identifier is not valid for the entity domain.

A valid entity identifier does never generate an error, even if the filtered property map resource does not define any properties for it.

If an entity identifier in the "entities" field of the request is invalid, the ALTO server MUST return an "E_INVALID_FIELD_VALUE" error defined in Section 8.5.2 of [RFC7285], and the "value" field of the error message SHOULD indicate the provided invalid entity identifier.

- * A property name in the "properties" field of the request is invalid. This occurs when this property name is not defined in the "properties" capability of this resource in the IRD.

When a filtered property map resource does not define a value for a property requested on a particular entity, it is not an error. In this case, the ALTO server MUST omit that property from the response for that endpoint.

If a property name in "properties" in the request is invalid, the ALTO server MUST return an "E_INVALID_FIELD_VALUE" error defined in Section 8.5.2 of [RFC7285]. The "value" field of the error message SHOULD indicate the property name.

Some identifiers can be interpreted as both an entity name and a property name, as it is the case for "pid" if it would be erroneously used alone. In such a case, the Server SHOULD follow Section 8.5.2 of [RFC7285], that says: "For an E_INVALID_FIELD_VALUE error, the server may include an optional field named "field" in the "meta" field of the response, to indicate the field that contains the wrong value."

The response to a valid request is the same as for the Property Map (see Section 7.6), except that:

- * If the requested entities include entities with a resource-agnostic identifier, the "dependent-vtags" field in its "meta" field MUST include version tags of all dependent resources appearing in the "uses" field.
- * If the requested entities only include entities in resource-specific entity domains, the "dependent-vtags" field in its "meta" field MUST include the version tags of the resources on which the requested resource-specific entity domains and the requested resource-specific properties are dependent on.
- * The response only includes the entities and properties requested by the client. If an entity in the request is identified by a hierarchical identifier (e.g., a "ipv4" or "ipv6" prefix), the response MUST return all properties that are present on any address covered by the prefix, even though some of those properties may not be present on all addresses covered by the prefix.
- * When the input member "properties" is absent from the client request, the Server returns a property map containing all the requested entity identifiers on which one or more properties are defined. For all the entities of the returned map, the returned property value is equal to '{}'.

The filtered property map response MUST include all the inherited property values for the requested entities and all the entities which are able to inherit property values from the requested entities. To achieve this goal, the ALTO server MAY follow two rules:

- * If a property for a requested entity is inherited from another entity not included in the request, the response MUST include this property for the requested entity. For example, A full property map may skip a property P for an entity A (e.g., ipv4:192.0.2.0/31) if P can be derived using inheritance from another entity B (e.g., ipv4:192.0.2.0/30). A filtered property map request may include only A but not B. In such a case, the property P MUST be included in the response for A.
- * If there are entities covered by a requested entity but having different values for the requested properties, the response MUST include all those entities and the different property values for them. For example, considering a request for property P of entity A (e.g., ipv4:192.0.2.0/31), if P has value v1 for A1=ipv4:192.0.2.0/32 and v2 for A2=ipv4:192.0.2.1/32, then, the response SHOULD include A1 and A2.

For the sake of response compactness, the ALTO server SHOULD obey the following rule:

- * If an entity identifier in the response is already covered by other entities identifiers in the same response, it SHOULD be removed from the response. In the previous example, the entity A = ipv4:192.0.2.0/31 SHOULD be removed because A1 and A2 cover all the addresses in A.

An ALTO client should be aware that the entities in the response may be different from the entities in its request.

8.7. Entity Property Type Defined in This Document

This document defines the entity property type "pid". This property type extends the ALTO Endpoint Property Type "pid" defined in section 7.1.1 of [RFC7285] as follows: the property has the same semantics and applies to IPv4 and IPv6 addresses; the difference is that the IPv4 and IPv6 addresses have evolved from the status of endpoints to the status of entities.

The defining information resource for property type MUST be a network map. This document requests a IANA registration for this property

8.7.1. Entity Property Type: pid

1. Identifier: pid
2. Semantics: the intended semantics are the same as in [RFC7285] for the ALTO Endpoint Property Type "pid"
3. Media type of defining information resource: application/alto-networkmap+json
4. Security considerations: for entity property type "pid" are the same as documented in [RFC7285] for the ALTO Endpoint Property Type "pid".

9. Impact on Legacy ALTO Servers and ALTO Clients

9.1. Impact on Endpoint Property Service

Since the Property Map and the Filtered Property Map defined in this document provide a functionality that covers the EPS defined in Section 11.4 of [RFC7285], ALTO servers may prefer to provide Property Map and Filtered Property Map in place of EPS. However, for the legacy endpoint properties, it is recommended that ALTO servers also provide EPS so that legacy clients can still be supported.

9.2. Impact on Resource-Specific Properties

Section 10.8 of [RFC7285] defines two categories of endpoint properties: "resource-specific" and "global". Resource-specific property names are prefixed with the ID of the resource they depend on, while global property names have no such prefix. The property map and the filtered property map defined in this document define similar categories of entity properties. The difference is that entity property maps do not define "global" entity properties. Instead, they define "self-defined" entity properties as a special case of "resource-specific" entity properties, where the specific resource is the property map itself. This means that "self-defined" properties are defined within the scope of the property map.

9.3. Impact on Other Properties

In the present extension, properties can be defined on sets of entity addresses, rather than just individual endpoint addresses as initially defined in [RFC7285]. This might change the semantics of a property. These sets can be for example hierarchical IP address blocks. For instance, a property such as fictitious "geo-location", defined on a set of IP addresses would have a value corresponding to a location representative of all the addresses in this set.

10. Examples

In this document, the HTTP message bodies of all the examples use Unix-style line-ending character (%x0A) as the line separator.

10.1. Network Map

The examples in this section use a very simple default network map:

```
defaultpid:  ipv4:0.0.0.0/0  ipv6:::/0
pid1:         ipv4:192.0.2.0/25
pid2:         ipv4:192.0.2.0/27
pid3:         ipv4:192.0.3.0/28
pid4:         ipv4:192.0.3.16/28
```

Figure 3: Example Default Network Map

And another simple alternative network map:

```
defaultpid:  ipv4:0.0.0.0/0  ipv6:::/0
pid1:         ipv4:192.0.2.0/27
pid2:         ipv4:192.0.3.0/27
```

Figure 4: Example Alternative Network Map

10.2. Property Definitions

Beyond "pid", the examples in this section use four additional fictitious property types for entities of domain type "ipv4": "countrycode", "ASN", "ISP", and "state". These properties are assumed to be resource-agnostic so their name is identical to their type. The entities have the following values:

	ISP	ASN	countrycode	state
ipv4:192.0.2.0/23:	BitsRus	-	us	-
ipv4:192.0.2.0/28:	-	65543	-	NJ
ipv4:192.0.2.16/28:	-	65543	-	CT
ipv4:192.0.2.1:	-	-	-	PA
ipv4:192.0.3.0/28:	-	65544	-	TX
ipv4:192.0.3.16/28:	-	65544	-	MN

Figure 5: Example Property Values for Internet Address Domains

And the examples in this section use the property "region" for the PID domain of the default network map with the following values:

	region
pid:defaultpid:	-
pid:pid1:	us-west
pid:pid2:	us-east
pid:pid3:	us-south
pid:pid4:	us-north

Figure 6: Example Property Values for Default Network Map's PID Domain

Note that "-" means the value of the property for the entity is "undefined". So the entity would inherit a value for this property by the inheritance rule if possible. For example, the value of the "ISP" property for "ipv4:192.0.2.1" is "BitsRus" because of "ipv4:192.0.2.0/24". But the "region" property for "pid:defaultpid" has no value because no entity from which it can inherit.

Similar to the PID domain of the default network map, the examples in this section use the property "ASN" for the PID domain of the alternative network map with the following values:

	ASN
pid:defaultpid:	-
pid:pid1:	65543
pid:pid2:	65544

Figure 7: Example Property Values for Alternative Network Map's PID Domain

10.3. Information Resource Directory (IRD)

The following IRD defines ALTO Server information resources that are relevant to the Entity Property Service. It provides a property map for the "ISP" and "ASN" properties. The server could have provided a single property map for all four properties, but does not, presumably because the organization that runs the ALTO server believes that a client is not necessarily interested in getting all four properties.

The server provides several filtered property maps. The first returns all four properties, and the second returns only the "pid" property for the default network map and the "alt-network-map".

The filtered property maps for the "ISP", "ASN", "countrycode" and "state" properties do not depend on the default network map (it does not have a "uses" capability), because the definitions of those properties do not depend on the default network map. The Filtered Property Map providing the "pid" property does have a "uses" capability for the default network map because the default network map defines the values of the "pid" property.

Note that for legacy clients, the ALTO server provides an Endpoint Property Service for the "pid" property defined on the endpoints of the default network map and the "alt-network-map".

The server provides another filtered Property map resource, named "ane-dc-property-map", that returns fictitious properties named "storage-capacity", "ram" and "cpu" for ANEs that have a persistent identifier. The entity domain to which the ANEs belong is "self-defined" and valid only within the property map.

The other property maps in the returned IRD are here for purposes of illustration.

```
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: 2713
Content-Type: application/alto-directory+json
```

```
{
  "meta" : {
    "default-alto-network-map" : "default-network-map"
  },
  "resources" : {
    "default-network-map" : {
      "uri" : "http://alto.example.com/networkmap/default",
      "media-type" : "application/alto-networkmap+json"
    },
    "alt-network-map" : {
      "uri" : "http://alto.example.com/networkmap/alt",
      "media-type" : "application/alto-networkmap+json"
    },
    "ia-property-map" : {
      "uri" : "http://alto.example.com/propmap/full/inet-ia",
      "media-type" : "application/alto-propmap+json",
      "capabilities" : {
        "mappings": {
          "ipv4": [ ".ISP", ".ASN" ],

```

```
        "ipv6": [ ".ISP", ".ASN" ]
      }
    },
    "iacs-property-map" : {
      "uri" : "http://alto.example.com/propmap/lookup/inet-iacs",
      "media-type" : "application/alto-propmap+json",
      "accepts": "application/alto-propmapparams+json",
      "capabilities" : {
        "mappings": {
          "ipv4": [ ".ISP", ".ASN", ".countrycode", ".state" ],
          "ipv6": [ ".ISP", ".ASN", ".countrycode", ".state" ]
        }
      }
    },
    "region-property-map": {
      "uri": "http://alto.example.com/propmap/lookup/region",
      "media-type": "application/alto-propmap+json",
      "accepts": "application/alto-propmapparams+json",
      "uses" : [ "default-network-map", "alt-network-map" ],
      "capabilities": {
        "mappings": {
          "default-network-map.pid": [ ".region" ],
          "alt-network-map.pid": [ ".ASN" ]
        }
      }
    },
    "ip-pid-property-map" : {
      "uri" : "http://alto.example.com/propmap/lookup/pid",
      "media-type" : "application/alto-propmap+json",
      "accepts" : "application/alto-propmapparams+json",
      "uses" : [ "default-network-map", "alt-network-map" ],
      "capabilities" : {
        "mappings": {
          "ipv4": [ "default-network-map.pid",
                    "alt-network-map.pid" ],
          "ipv6": [ "default-network-map.pid",
                    "alt-network-map.pid" ]
        }
      }
    },
    "legacy-endpoint-property" : {
      "uri" : "http://alto.example.com/legacy/eps-pid",
      "media-type" : "application/alto-endpointprop+json",
      "accepts" : "application/alto-endpointpropparams+json",
      "capabilities" : {
        "properties" : [ "default-network-map.pid",
                         "alt-network-map.pid" ]
      }
    }
  },
  "legacy-endpoint-property" : {
    "uri" : "http://alto.example.com/legacy/eps-pid",
    "media-type" : "application/alto-endpointprop+json",
    "accepts" : "application/alto-endpointpropparams+json",
    "capabilities" : {
      "properties" : [ "default-network-map.pid",
                       "alt-network-map.pid" ]
    }
  }
}
```

```

    }
  },
  "ane-dc-property-map": {
    "uri" : "http://alto.example.com/propmap/lookup/ane-dc",
    "media-type" : "application/alto-propmap+json",
    "accepts": "application/alto-propmapparams+json",
    "capabilities": {
      "mappings": {
        ".ane" : [ "storage-capacity", "ram", "cpu" ]
      }
    }
  }
}
}
}

```

Figure 8: Example IRD

10.4. Full Property Map Example

The following example uses the properties and IRD defined in Section 10.3 to retrieve a Property Map for entities with the "ISP" and "ASN" properties.

Note that, to be compact, the response does not include the entity "ipv4:192.0.2.1" because values of all those properties for this entity are inherited from other entities.

Also note that the entities "ipv4:192.0.2.0/28" and "ipv4:192.0.2.16/28" are merged into "ipv4:192.0.2.0/27", because they have the same value of the "ASN" property. The same rule applies to the entities "ipv4:192.0.3.0/28" and "ipv4:192.0.3.16/28". Both of "ipv4:192.0.2.0/27" and "ipv4:192.0.3.0/27" omit the value for the "ISP" property, because it is inherited from "ipv4:192.0.2.0/23".

```

GET /propmap/full/inet-ia HTTP/1.1
Host: alto.example.com
Accept: application/alto-propmap+json,application/alto-error+json

```

```
HTTP/1.1 200 OK
Content-Length: 418
Content-Type: application/alto-propmap+json

{
  "meta": {
    "dependent-vtags": [
      {"resource-id": "default-network-map",
       "tag": "3ee2cb7e8d63d9fab71b9b34cbf764436315542e"},
      {"resource-id": "alt-network-map",
       "tag": "c0ce023b8678a7b9ec00324673b98e54656d1f6d"}
    ]
  },
  "property-map": {
    "ipv4:192.0.2.0/23": {".ISP": "BitsRus"},
    "ipv4:192.0.2.0/27": {".ASN": "65543"},
    "ipv4:192.0.3.0/27": {".ASN": "65544"}
  }
}
```

10.5. Filtered Property Map Example #1

The following example uses the filtered property map resource to request the "ISP", "ASN" and "state" properties for several IPv4 addresses.

Note that the value of "state" for "ipv4:192.0.2.1" is the only explicitly defined property; the other values are all derived by the inheritance rules for Internet address entities.

```
POST /propmap/lookup/inet-iacs HTTP/1.1
Host: alto.example.com
Accept: application/alto-propmap+json,application/alto-error+json
Content-Length: 158
Content-Type: application/alto-propmapparams+json
```

```
{
  "entities" : [ "ipv4:192.0.2.0",
                 "ipv4:192.0.2.1",
                 "ipv4:192.0.2.17" ],
  "properties" : [ ".ISP", ".ASN", ".state" ]
}
```

```
HTTP/1.1 200 OK
Content-Length: 540
Content-Type: application/alto-propmap+json
```

```
{
  "meta": {
    "dependent-vtags": [
      {"resource-id": "default-network-map",
       "tag": "3ee2cb7e8d63d9fab71b9b34cbf764436315542e"},
      {"resource-id": "alt-network-map",
       "tag": "c0ce023b8678a7b9ec00324673b98e54656d1f6d"}
    ]
  },
  "property-map": {
    "ipv4:192.0.2.0":
      {".ISP": "BitsRus", ".ASN": "65543", ".state": "NJ"},
    "ipv4:192.0.2.1":
      {".ISP": "BitsRus", ".ASN": "65543", ".state": "PA"},
    "ipv4:192.0.2.17":
      {".ISP": "BitsRus", ".ASN": "65543", ".state": "CT"}
  }
}
```

10.6. Filtered Property Map Example #2

The following example uses the filtered property map resource to request the "ASN", "countrycode" and "state" properties for several IPv4 prefixes.

Note that the property values for both entities "ipv4:192.0.2.0/26" and "ipv4:192.0.3.0/26" are not explicitly defined. They are inherited from the entity "ipv4:192.0.2.0/23".

Also note that some entities like "ipv4:192.0.2.0/28" and "ipv4:192.0.2.16/28" in the response are not explicitly listed in the request. The response includes them because they are refinements of the requested entities and have different values for the requested properties.

The entity "ipv4:192.0.4.0/26" is not included in the response, because there are neither entities which it is inherited from, nor entities inherited from it.

```
POST /propmap/lookup/inet-iacs HTTP/1.1
Host: alto.example.com
Accept: application/alto-propmap+json,application/alto-error+json
Content-Length: 174
Content-Type: application/alto-propmapparams+json
```

```
{
  "entities" : [ "ipv4:192.0.2.0/26",
                 "ipv4:192.0.3.0/26",
                 "ipv4:192.0.4.0/26" ],
  "properties" : [ ".ASN", ".countrycode", ".state" ]
}
```

```
HTTP/1.1 200 OK
Content-Length: 774
Content-Type: application/alto-propmap+json
```

```
{
  "meta": {
    "dependent-vtags": [
      {"resource-id": "default-network-map",
       "tag": "3ee2cb7e8d63d9fab71b9b34cbf764436315542e"},
      {"resource-id": "alt-network-map",
       "tag": "c0ce023b8678a7b9ec00324673b98e54656d1f6d"}
    ]
  },
  "property-map": {
    "ipv4:192.0.2.0/26": {".countrycode": "us"},
    "ipv4:192.0.2.0/28": {".ASN": "65543",
                        ".state": "NJ"},
    "ipv4:192.0.2.16/28": {".ASN": "65543",
                          ".state": "CT"},
    "ipv4:192.0.2.1": {".state": "PA"},
    "ipv4:192.0.3.0/26": {".countrycode": "us"},
    "ipv4:192.0.3.0/28": {".ASN": "65544",
                        ".state": "TX"},
    "ipv4:192.0.3.16/28": {".ASN": "65544",
                          ".state": "MN"}
  }
}
```

10.7. Filtered Property Map Example #3

The following example uses the filtered property map resource to request the "default-network-map.pid" property and the "alt-network-map.pid" property for a set of IPv4 addresses and prefixes.

Note that the entity "ipv4:192.0.3.0/27" is decomposed into two entities "ipv4:192.0.3.0/28" and "ipv4:192.0.3.16/28", as they have different "default-network-map.pid" property values.

```
POST /propmap/lookup/pid HTTP/1.1
Host: alto.example.com
Accept: application/alto-propmap+json,application/alto-error+json
Content-Length: 222
Content-Type: application/alto-propmapparams+json
```

```
{
  "entities" : [
    "ipv4:192.0.2.128",
    "ipv4:192.0.2.0/27",
    "ipv4:192.0.3.0/27" ],
  "properties" : [ "default-network-map.pid",
    "alt-network-map.pid" ]
}
```

```
HTTP/1.1 200 OK
Content-Length: 774
Content-Type: application/alto-propmap+json
```

```
{
  "meta": {
    "dependent-vtags": [
      {"resource-id": "default-network-map",
        "tag": "3ee2cb7e8d63d9fab71b9b34cbf764436315542e"},
      {"resource-id": "alt-network-map",
        "tag": "c0ce023b8678a7b9ec00324673b98e54656d1f6d"}
    ]
  },
  "property-map": {
    "ipv4:192.0.2.128": { "default-network-map.pid": "defaultpid",
      "alt-network-map.pid": "defaultpid"},
    "ipv4:192.0.2.0/27": { "default-network-map.pid": "pid2",
      "alt-network-map.pid": "pid1"},
    "ipv4:192.0.3.0/28": { "default-network-map.pid": "pid3",
      "alt-network-map.pid": "pid2"},
    "ipv4:192.0.3.16/28": { "default-network-map.pid": "pid4",
      "alt-network-map.pid": "pid2"}
  }
}
```

10.8. Filtered Property Map Example #4

Here is an example of using the filtered property map to query the regions for several PIDs in "default-network-map". The "region" property is specified as a "self-defined" property, i.e., the values of this property are defined by this property map resource.

```
POST /propmap/lookup/region HTTP/1.1
Host: alto.example.com
Accept: application/alto-propmap+json,application/alto-error+json
Content-Length: 132
Content-Type: application/alto-propmapparams+json
```

```
{
  "entities" : ["default-network-map.pid:pid1",
               "default-network-map.pid:pid2"],
  "properties" : [ ".region" ]
}
```

```
HTTP/1.1 200 OK
Content-Length: 326
Content-Type: application/alto-propmap+json
```

```
{
  "meta" : {
    "dependent-vtags" : [
      { "resource-id": "default-network-map",
        "tag": "7915dc0290c2705481c491a2b4ffbec482b3cf62" }
    ]
  },
  "property-map": {
    "default-network-map.pid:pid1": {
      ".region": "us-west"
    },
    "default-network-map.pid:pid2": {
      ".region": "us-east"
    }
  }
}
```

10.9. Filtered Property Map for ANEs Example #5

The following example uses the filtered property map resource "ane-dc-property-map" to request properties "storage-capacity" and "cpu" on several ANEs defined in this property map.


```
POST /propmap/lookup/ane-dc HTTP/1.1
Host: alto.example.com
Accept: application/alto-propmap+json,application/alto-error+json
Content-Length: 155
Content-Type: application/alto-propmapparams+json
```

```
{
  "entities" : [".ane:dc21",
                ".ane:dc45.srv9",
                ".ane:dc6.srv-cluster8"],
  "properties" : [ "storage-capacity", "cpu"]
}
```

```
HTTP/1.1 200 OK
Content-Length: 295
Content-Type: application/alto-propmap+json
```

```
{
  "meta" : {
  },
  "property-map": {
    ".ane:dc21":
      {"storage-capacity" : 40000, "cpu" : 500},
    ".ane:dc45.srv9":
      {"storage-capacity" : 100, "cpu" : 20},
    ".ane:dc6.srv-cluster8":
      {"storage-capacity" : 6000, "cpu" : 100}
  }
}
```

11. Security Considerations

Both Property Map and Filtered Property Map defined in this document fit into the architecture of the ALTO base protocol, and hence the Security Considerations (Section 15 of [RFC7285]) of the base protocol fully apply: authenticity and integrity of ALTO information (i.e., authenticity and integrity of Property Maps), potential undesirable guidance from authenticated ALTO information (e.g., potentially imprecise or even wrong value of a property such as geo-location), confidentiality of ALTO information (e.g., exposure of a potentially sensitive entity property such as geo-location), privacy for ALTO users, and availability of ALTO services should all be considered.

ALTO clients using this extension should in addition be aware that the entity properties they require may convey more details than the endpoint properties conveyed by using [RFC7285]. Client requests may reveal details on their activity or plans thereof, that a malicious

Server, that is in a position to do so, may monetize or use for attacks or undesired surveillance. Likewise, ALTO Servers expose entities and properties related to specific parts of the infrastructure that reveal details on capabilities, locations, or resource availability. These details may be maliciously used for competition purposes, or to cause resource shortage or undesired publication.

To address these concerns, the Property Maps provided by this extension require additional attention on two security considerations discussed in [RFC7285]: "potential undesirable guidance from authenticated ALTO information" (Section 15.2 of [RFC7285]) and "confidentiality of ALTO information" (Section 15.3 of [RFC7285]). Threats to the availability of the ALTO Service caused by highly demanding queries should be addressed as specified in Section 15.5 of [RFC7285].

- * Potential undesirable guidance from authenticated ALTO information: it can be caused by Property values that change over time and thus lead to performance degradation or system rejection of application requests.

To avoid these consequences, a more robust ALTO client should adopt and extend protection strategies specified in Section 15.2 of [RFC7285]. For example, to be notified immediately when a particular ALTO value that the Client depends on changes, it is RECOMMENDED that both the ALTO Client and ALTO Server using this extension implement "Application-Layer Traffic Optimization (ALTO) Incremental Updates Using Server-Sent Events (SSE)" [RFC8895].

- * Confidentiality of ALTO information: as discussed in Section 15 of [RFC7285], properties may have sensitive customer-specific information. If this is the case, an ALTO Server may limit access to those properties by providing several different property maps. For non-sensitive properties, the ALTO Server would provide a URI which accepts requests from any client. Sensitive properties, on the other hand, would only be available via a secure URI which would require client authentication. Another way is to expose highly abstracted coarse-grained property values to all Clients while restricting access to URIs exposing more fine-grained values to authorized Clients. Restricted access URIs may be gathered in delegate IRDs as specified in Section 9.2.4 of [RFC7285].

Also, while technically this document does not introduce any security risks not inherent in the Endpoint Property Service defined by [RFC7285], the GET-mode property map resource defined in this document does make it easier for a client to download large numbers of property values. Accordingly, an ALTO Server should limit GET-mode property maps to properties that do not contain sensitive data.

Section 12 of this document specifies that the ALTO service provider **MUST** be aware of the potential sensitivity of exposed entity domains and properties. Section 12.2.2. (ALTO Entity Domain Type Registration Process) of this document specifies that when the registration of an entity domain type is requested at the IANA, the request **MUST** include security considerations that show awareness of how the exposed entity addresses may be related to private information about an ALTO client or an infrastructure service provider. Likewise, Section 12.3. (ALTO Entity Property Type Registry) of this document specifies that when the registration of a property type is requested at the IANA, the request **MUST** include security considerations that explain why this property type is required for ALTO-based operations.

The risk of ALTO information being leaked to malicious Clients or third parties is addressed similarly to Section 7 of [RFC8896]. ALTO clients and servers **SHOULD** support TLS 1.3 [RFC8446].

12. IANA Considerations

This document defines additional application/alto-* media types, that are listed in Table 1. It defines an ALTO Entity Domain Type Registry that extends the ALTO Address Type Registry defined in [RFC7285]. It also defines an ALTO Entity Property Type Registry that extends the ALTO endpoint property registry defined in [RFC7285].

Type	Subtype	Specification
application	alto-propmap+json	Section 7.1
application	alto-propmapparams+json	Section 8.3

Table 1: Additional ALTO Media Types.

12.1. application/alto-propmap+json Media Type

Type name:

application

Subtype name:

alto-propmap+json

Required parameters:

n/a

Optional parameters:

n/a

Encoding considerations:

Encoding considerations are identical to those specified for the "application/json" media type. See [RFC8259].

Security considerations:

Security considerations related to the generation and consumption of ALTO Protocol messages are discussed in Section 15 of [RFC7285] and Section 11 of this document.

Interoperability considerations:

n/a

Published specification:

This document is the specification for this media type. See Section 7.1.

Applications that use this media type:

ALTO servers and ALTO clients [RFC7285], either stand alone or embedded within other applications, when the queried resource is a property map, whether filtered or not.

Fragment identifier considerations:

n/a

Additional information:

Magic number(s): n/a

File extension(s): n/a

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).

12.2. alto-propmapparams+json Media Type

Type name:
application

Subtype name:
alto-propmapparams+json

Required parameters:
n/a

Optional parameters:
n/a

Encoding considerations:
Encoding considerations are identical to those specified for the "application/json" media type. See [RFC8259].

Security considerations:
Security considerations related to the generation and consumption of ALTO Protocol messages are discussed in Section 15 of [RFC7285] and Section 11 of this document.

Interoperability considerations:
n/a

Published specification:
This document is the specification for this media type. See Section 8.3.

Applications that use this media type:
ALTO servers and ALTO clients [RFC7285], either stand alone or embedded within other applications, when the queried resource is a filtered property map. This media-type indicates the data format used by the ALTO client to supply the property map filtering parameters.

Fragment identifier considerations:
n/a

Additional information:
Magic number(s): n/a

File extension(s): n/a

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).

12.3. ALTO Entity Domain Type Registry

This document requests IANA to create and maintain the "ALTO Entity Domain Type Registry", listed in Table 2. The first line lists information items that must be provided with each registered entity domain type. Section 12.3.2 specifies how to document these items and provides guidance on the security considerations item that must be documented in addition.

Identifier	Entity Identifier Encoding	Hierarchy & Inheritance	Media Type of Defining Resource	Mapping to ALTO Address Type
ipv4	See Section 6.1.1	See Section 6.1.3	application/alto-networkmap+json	true
ipv6	See Section 6.1.2	See Section 6.1.3	application/alto-networkmap+json	true
pid	See Section 6.2	None	application/alto-networkmap+json	false

Table 2: ALTO Entity Domain Types

This registry serves two purposes. First, it ensures uniqueness of identifiers referring to ALTO entity domain types. Second, it states the requirements for allocated entity domain types.

As specified in Section 5.1.1, identifiers prefixed with "priv:" are reserved for Private Use without a need to register with IANA

12.3.1. Consistency Procedure between ALTO Address Type Registry and ALTO Entity Domain Type Registry

One potential issue of introducing the "ALTO Entity Domain Type Registry" is its relationship with the "ALTO Address Types Registry" already defined in Section 14.4 of [RFC7285]. In particular, the entity identifier of a type of an entity domain registered in the "ALTO Entity Domain Type Registry" MAY match an address type defined in "ALTO Address Type Registry". It is necessary to precisely define and guarantee the consistency between "ALTO Address Type Registry" and "ALTO Entity Domain Registry".

We define that the ALTO Entity Domain Type Registry is consistent with ALTO Address Type Registry if two conditions are satisfied:

- * When an address type is already or able to be registered in the ALTO Address Type Registry [RFC7285], the same identifier MUST be used when a corresponding entity domain type is registered in the ALTO Entity Domain Type Registry.
- * If an ALTO entity domain type has the same identifier as an ALTO address type, their addresses encoding MUST be compatible.

To achieve this consistency, the following items MUST be checked before registering a new ALTO entity domain type in a future document:

- * Whether the ALTO Address Type Registry contains an address type that can be used as an identifier for the candidate entity domain type identifier. This has been done for the identifiers "ipv4" and "ipv6" of Table 2.
- * Whether the candidate entity domain type identifier can potentially be an endpoint address type, as defined in Sections 2.1 and 2.2 of [RFC7285].

When a new ALTO entity domain type is registered, the consistency with the ALTO Address Type Registry MUST be ensured by the following procedure:

- * Test: Do corresponding entity domain type identifiers match a known "network" address type?
 - If yes (e.g., cell, MAC or socket addresses):
 - o Test: Is such an address type present in the ALTO Address Type Registry?
 - + If yes: Set the new ALTO entity domain type identifier to be the found ALTO address type identifier.
 - + If no: Define a new ALTO entity domain type identifier and use it to register a new address type in the ALTO Address Type Registry following Section 14.4 of [RFC7285].
 - o Use the new ALTO entity domain type identifier to register a new ALTO entity domain type in the ALTO Entity Domain Type Registry following Section 12.3.2 of this document.
 - If no (e.g., pid name, ane name or country code): Proceed with the ALTO Entity Domain Type registration as described in Section 12.3.2.

12.3.2. ALTO Entity Domain Type Registration Process

New ALTO entity domain types are assigned after IETF Review [RFC8126] to ensure that proper documentation regarding the new ALTO entity domain types and their security considerations has been provided. RFCs defining new entity domain types MUST indicate how an entity in a registered type of domain is encoded as an EntityID, and, if applicable, the rules defining the entity hierarchy and property inheritance. Updates and deletions of ALTO entity domains types follow the same procedure.

Registered ALTO entity domain type identifiers MUST conform to the syntactical requirements specified in Section 5.1.2. Identifiers are to be recorded and displayed as strings.

Requests to the IANA to add a new value to the Entity Domain Type registry MUST include the following information:

- * Identifier: The name of the desired ALTO entity domain type.
- * Entity Identifier Encoding: The procedure for encoding the identifier of an entity of the registered domain type as an EntityID (see Section 5.1.3). If corresponding entity identifiers of an entity domain type match a known "network" address type, the Entity Identifier Encoding of this domain identifier MUST include both Address Encoding and Prefix Encoding of the same identifier registered in the ALTO Address Type Registry [RFC7285]. To define properties, an individual entity identifier and the corresponding full-length prefix MUST be considered aliases for the same entity.
- * Hierarchy: If the entities form a hierarchy, the procedure for determining that hierarchy.
- * Inheritance: If entities can inherit property values from other entities, the procedure for determining that inheritance.
- * Media type of defining information resource: Some entity domain types allow an entity domain name to be combined with an information resource name to define a resource-specific entity domain. Such an information resource is called "defining information resource", defined in Section 4.6. For each entity domain type, the potential defining information resources have one common media type. This unique common media type is specific to the entity domain type and MUST be specified.
- * Mapping to ALTO Address Type: A boolean value to indicate if the entity domain type can be mapped to the ALTO address type with the same identifier.

- * **Security Considerations:** In some usage scenarios, entity identifiers 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 cognizant of how (or if) the addressing scheme relates to private information and network proximity.

This specification requests registration of the identifiers "ipv4", "ipv6" and "pid", as shown in Table 2.

12.4. ALTO Entity Property Type Registry

This document requests IANA to create and maintain the "ALTO Entity Property Type Registry", listed in Table 3.

This registry extends the "ALTO Endpoint Property Type Registry", defined in [RFC7285], in that a property type is defined on one or more entity domains, rather than just on IPv4 and IPv6 Internet address domains. An entry in this registry is an ALTO entity property type defined in Section 5.2.1. Thus, a registered ALTO entity property type identifier MUST conform to the syntactical requirements specified in that section.

As specified in Section 5.2.1, identifiers prefixed with "priv:" are reserved for Private Use without a need to register with IANA.

The first line of Table 3 lists information items that must be provided with each registered entity property type.

Identifier	Intended Semantics	Media Type of Defining Resource
pid	See Section 7.1.1 of [RFC7285]	application/alto-networkmap+json

Table 3: ALTO Entity Property Types.

New ALTO entity property types are assigned after IETF Review [RFC8126] to ensure that proper documentation regarding the new ALTO entity property types and their security considerations has been provided. RFCs defining new entity property types SHOULD indicate how a property of a registered type is encoded as a property name. Updates and deletions of ALTO entity property types follow the same procedure.

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

- * Identifier: The identifier for the desired ALTO entity property type. The format MUST be as defined in Section 5.2.1 of this document.
- * Intended Semantics: ALTO entity properties carry with them semantics to guide their usage by ALTO clients. Hence, a document defining a new type SHOULD provide guidance to both ALTO service providers and applications utilizing ALTO clients as to how values of the registered ALTO entity property should be interpreted.
- * Media type of defining information resource: when the property type allows values to be defined relative to a given information resource, the latter is referred to as the "defining information resource", see description in Section 4.7. For each property type, the potential defining information resources have one common media type. This unique common media type is specific to the property type and MUST be specified.
- * Security Considerations: ALTO entity properties expose information to ALTO clients. ALTO service providers should be cognizant of the security ramifications related to the exposure of an entity property.

In security considerations, the request should also discuss the sensitivity of the information, and why it is required for ALTO-based operations. Regarding this discussion, the request SHOULD follow the recommendations of Section 14.3. ALTO Endpoint Property Type Registry in [RFC7285].

This document requests registration of the identifier "pid", listed in Table 3. Semantics for this property are documented in Section 7.1.1 of [RFC7285]. No security issues related to the exposure of a "pid" identifier are considered, as it is exposed with the Network Map Service defined and mandated in [RFC7285].

13. Acknowledgments

The authors would like to thank Dawn Chen, and Shenshen Chen for their contributions to earlier drafts. Thank you also to Qiao Xiang, Shawn Lin, Xin Wang and Vijay Gurbani for fruitful discussions. Last, big thanks to Danny Perez and Luis Contreras for their substantial Working Group review feedback and suggestions to improve this document, to Vijay Gurbani, ALTO WG Chair and Martin Duke, Transport Area Director, for their thorough review, discussions, guidance and shepherding, that further helped to enrich this

document.

14. References

14.1. Normative References

- [ISO3166-1] ISO (International Organization for Standardization), ., "ISO 3166-1: Codes for the representation of names of countries and their subdivisions -- Part 1: Country codes", 2020.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <<https://www.rfc-editor.org/info/rfc3986>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", February 2006.
- [RFC4632] Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", BCP 122, RFC 4632, DOI 10.17487/RFC4632, August 2006, <<https://www.rfc-editor.org/info/rfc4632>>.
- [RFC5952] Kawamura, S. and M. Kawashima, "A Recommendation for IPv6 Address Text Representation", RFC 5952, DOI 10.17487/RFC5952, August 2010, <<https://www.rfc-editor.org/info/rfc5952>>.
- [RFC7285] Alimi, R., Ed., Penno, R., Ed., Yang, Y., Ed., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, DOI 10.17487/RFC7285, September 2014, <<https://www.rfc-editor.org/info/rfc7285>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.

- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.
- [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <<https://www.rfc-editor.org/info/rfc8259>>.
- [RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <<https://www.rfc-editor.org/info/rfc8446>>.
- [RFC8895] Roome, W. and Y. Yang, "Application-Layer Traffic Optimization (ALTO) Incremental Updates Using Server-Sent Events (SSE)", RFC 8895, DOI 10.17487/RFC8895, November 2020, <<https://www.rfc-editor.org/info/rfc8895>>.

14.2. Informative References

- [I-D.ietf-alto-cdni-request-routing-alto]
Seedorf, J., Yang, Y., Ma, K., Peterson, J., and J. Zhang, "Content Delivery Network Interconnection (CDNI) Request Routing: CDNI Footprint and Capabilities Advertisement using ALTO", Work in Progress, Internet-Draft, draft-ietf-alto-cdni-request-routing-alto-16, 12 January 2021, <<http://www.ietf.org/internet-drafts/draft-ietf-alto-cdni-request-routing-alto-16.txt>>.
- [I-D.ietf-alto-path-vector]
Gao, K., Lee, Y., Randriamasy, S., Yang, Y., and J. Zhang, "ALTO Extension: Path Vector", Work in Progress, Internet-Draft, draft-ietf-alto-path-vector-13, 20 November 2020, <<http://www.ietf.org/internet-drafts/draft-ietf-alto-path-vector-13.txt>>.
- [RFC3849] Huston, G., Lord, A., and P. Smith, "IPv6 Address Prefix Reserved for Documentation", July 2004.
- [RFC5511] Farrel, A., "Routing Backus-Naur Form (RBNF): A Syntax Used to Form Encoding Rules in Various Routing Protocol Specifications", April 2009.
- [RFC5737] Arkko, J., Cotton, M., and L. Vegoda, "IPv4 Address Blocks Reserved for Documentation", January 2010.

- [RFC7921] Atlas, A., Halpern, J., Hares, S., Ward, D., and T. Nadeau, "An Architecture for the Interface to the Routing System", RFC 7921, DOI 10.17487/RFC7921, June 2016, <<https://www.rfc-editor.org/info/rfc7921>>.
- [RFC8896] Randriamasy, S., Yang, R., Wu, Q., Deng, L., and N. Schwan, "Application-Layer Traffic Optimization (ALTO) Cost Calendar", RFC 8896, DOI 10.17487/RFC8896, November 2020, <<https://www.rfc-editor.org/info/rfc8896>>.

Appendix A. Features introduced with the Entity Property Maps extension

The Entity Property Maps extension described in this document introduces a number of features that are summarized in table below. The first column provides the name of the feature. The second column provides the section number of this document that gives a high level description of the feature. The third column provides the section number of this document that gives a normative description relating to the feature, when applicable.

Feature	High-level description	Related normative description
Entity	Section 3.1	Section 5.1.3
Entity domain (ED)	Section 3.2	
Entity domain type	Section 3.2.1	Section 5.1.1
Entity domain name	Section 3.2.2	Section 5.1.2
Entity property (EP) type	Section 3.3	Section 5.2, Section 5.2.1, Section 5.2.2, Section 5.2.3
Entity property map	Section 3.4	Section 7, Section 8
Resource-specific ED name	Section 4.2	Section 5.1.2, Section 5.1.2.1
Resource-specific EP value	Section 4.3	Section 5.2.3
Entity Hierarchy and property inheritance	Section 4.4	Section 5.1.4
Defining information resource	Section 4.6, Section 4.7	Section 12.3.2, Section 12.4

Table 4: Features introduced with ALTO Entity Property Maps

Authors' Addresses

Wendy Roome
 Nokia Bell Labs (Retired)
 124 Burlington Rd
 Murray Hill, NJ 07974
 United States of America
 Phone: +1-908-464-6975
 Email: wendy@wdroome.com

Sabine Randriamasy
Nokia Bell Labs
Route de Villejust
91460 NOZAY
France
Email: Sabine.Randriamasy@nokia-bell-labs.com

Y. Richard Yang
Yale University
51 Prospect Street
New Haven, CT 06511
United States of America
Phone: +1-203-432-6400
Email: yry@cs.yale.edu

Jingxuan Jensen Zhang
Tongji University
4800 Cao'An Hwy
Shanghai
201804
China
Email: jingxuan.n.zhang@gmail.com

Kai Gao
Sichuan University
No.24 South Section 1, Yihuan Road
Chengdu
610000
China
Email: kaigao@scu.edu.cn