

Workgroup: ALTO
Internet-Draft:
draft-contreras-alto-service-edge-07
Published: 13 March 2023
Intended Status: Informational
Expires: 14 September 2023
Authors: L. M. Contreras S. Randriamasy
 Telefonica Nokia Bell Labs
 J. Ros-Giralt D. Lachos C. Rothenberg
 Qualcomm Europe, Inc. Benocs Unicamp
Use of ALTO for Determining Service Edge

Abstract

Service providers are starting to deploy computing capabilities across the network for hosting applications such as AR/VR, vehicle networks, IoT, and AI training, among others. In these distributed computing environments, knowledge about computing and communication resources is necessary to determine the proper deployment location of each application. This document proposes an initial approach towards the use of ALTO to expose such information to the applications and assist the selection of their deployment locations.

About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at <https://giralt.github.io/draft-contreras-alto-service-edge/#go.draft-contreras-alto-service-edge.html>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-contreras-alto-service-edge/>.

Discussion of this document takes place on the WG Working Group mailing list (<mailto:alto@ietf.org>), which is archived at <https://mailarchive.ietf.org/arch/browse/alto/>. Subscribe at <https://www.ietf.org/mailman/listinfo/alto/>.

Source for this draft and an issue tracker can be found at <https://github.com/giralt/draft-contreras-alto-service-edge>.

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 14 September 2023.

Copyright Notice

Copyright (c) 2023 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](#)
- [2. Conventions and Definitions](#)
- [3. Computing Needs](#)
- [4. Usage of ALTO for Service Placement](#)
 - [4.1. Integrating Compute Information in ALTO](#)
 - [4.2. Association of Compute Capabilities to Network Topology](#)
 - [4.3. ALTO Architecture for Determining Service Edge](#)
- [5. ALTO Design Considerations for Determining Service Edge](#)
 - [5.1. Example of Entity Definition in Different Domains](#)
 - [5.2. Definition of Flavors in ALTO Property Map](#)
- [6. Use Cases](#)
 - [6.1. Open Abstraction for Edge Computing](#)
 - [6.2. Optimized placement of microservice components](#)
- [7. Security Considerations](#)
- [8. IANA Considerations](#)
- [9. Conclusions](#)
- [10. References](#)
 - [10.1. Normative References](#)
 - [10.2. Informative References](#)
- [Acknowledgments](#)
- [Authors' Addresses](#)

1. Introduction

With the advent of network virtualization, operators can make use of 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 as demanded by the users.

Operators are starting to deploy distributed computing environments in different parts of the network with the objective of addressing different service needs including latency, bandwidth, processing capabilities, storage, etc. This is translated in the emergence of a number of data centers (both in the cloud and at the edge) 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 of the corresponding data center.

The proliferation of the edge computing paradigm further increases the potential footprint and heterogeneity of the environments where a function or application can be deployed, resulting in different unitary cost per CPU, memory, and storage. This increases the complexity of deciding the location where a given function or application should be best deployed, as this decision should be influenced not only by the available resources in a given computing environment, but also by the network capacity of the path connecting the traffic source with the destination.

It is then essential for a network operator to have mechanisms assisting this 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 the transport network substrate. This would enable the integration of network capabilities in the function placement decision and further optimize performance of the deployed application.

This document proposes the use of ALTO [[RFC7285](#)] for assisting such a decision.

2. Conventions and Definitions

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. 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, and flavors that can be consumed in an ephemeral fashion, during the actual lifetime of the required function or application.

This same approach is being taken for characterizing bundles of resources on the so-called Network Function Virtualization Infrastructure Points of Presence (NFVI-PoPs) being deployed by the telco operators. For instance, the Common Network Function Virtualization Infrastructure Telecom Taskforce (CNTT) [[CNTT](#)], jointly hosted by GSMA [[GSMA](#)] and the Linux Foundation, intends to harmonize the definition of instances and flavors for abstracting capabilities of the underlying NFVI, facilitating a more efficient utilization of the infrastructure and simplifying the integration and certification of functions. (Here certification means the assessment of the expected behavior for a given function according to the level of resources determined by a given flavor.) An evolution of this initiative is Anuket [[Anuket](#)], which works to consolidate different architectures for well-known tools such as OpenStack and Kubernetes.

Taking CNTT as an example, the flavors or instances can be characterized according to:

**Type of instance (T):* Used to specify the type of instances, which are characterized as B (Basic), or N (Network Intensive). The latter includes network acceleration extensions for offloading network intensive operations to hardware.

**Interface Option (I):* Used to specify the associated bandwidth of the network interface.

**Compute flavor (F):* Used to specify a given predefined combination of resources in terms of virtual CPU, RAM, disk, and bandwidth for the management interface.

**Optional storage extension (S):* Used to specify additional storage capacity.

**Optional hardware acceleration characteristics (A):* Used to specify acceleration capabilities for improving the performance of the application.

The naming convention of an instance is thus encoded as TIFSA.

4. Usage of ALTO for Service Placement

ALTO can assist the deployment of a service 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 must also be interconnected outside the computing environment where it is deployed, therefore requiring the necessary network resources to satisfy application performance requirements such as bandwidth or latency.

The objective then is to leverage 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.

4.1. 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 a catalogue could be communicated to ALTO or even incorporated to it.

ALTO server queries could support TIFSA encoding in order to retrieve proper maps from ALTO. Additionally, filtered queries for particular characteristics of a flavor could also be supported.

4.2. 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 that the management of the network and cloud capabilities will be performed by the same entity, producing an integrated map to handle both network and compute abstractions jointly. While this can be straightforward when an ISP owns both the network and the cloud infrastructure, it can in general require collaboration between multiple administrative domains. Details on potential scenarios will be provided in future versions of this document.

At this stage, four potential solutions could be considered:

*To leverage (and possibly extend)
[\[I-D.ietf-teas-sf-aware-topo-model\]](#) for disseminating topology information together with the 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 [\[I-D.llc-teas-dc-aware-topo-model\]](#).

*To extend BGP-LS [[RFC7752](#)], which is already considered as a mechanism for feeding topology information in ALTO, in order to also advertise computing capabilities as well.

*To combine information from the infrastructure profiles catalogue with topological information by leveraging the IP prefixes allocated to the gateway providing connectivity to the NFVI PoP.

*To integrate with Cloud Infrastructure Managers that could expose cloud infrastructure capabilities as in [[CNTT](#)] and [[GSMA](#)].

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

4.3. ALTO Architecture for Determining Serve Edge

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

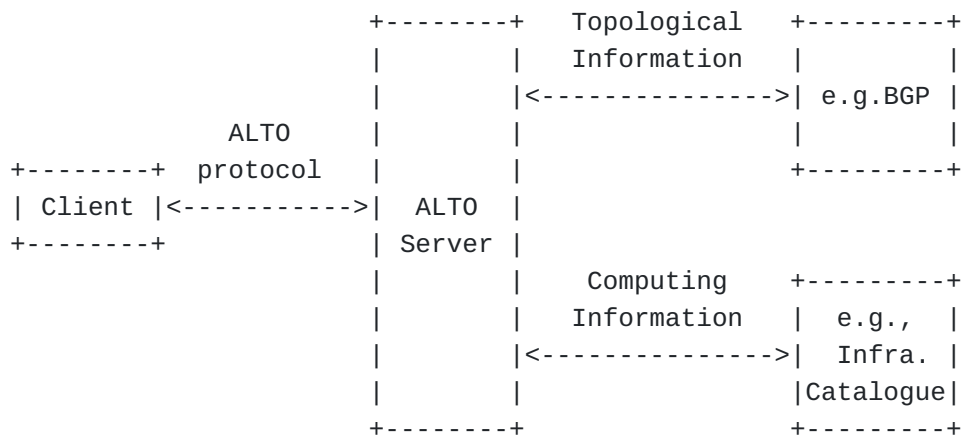


Figure 1: Service Edge Information Exchange.

In order to select the optimal edge server from both the network (e.g., the path with lower latency and/or higher bandwidth) and the cloud perspectives (e.g., number of CPUs/GPUs, available RAM and storage capacity), there is a need to see the edge server as both an IP entity (as in [[RFC7285](#)]) and an Abstract Network Element (ANE) entity (as in [[RFC9275](#)]).

Currently there is no mechanism (neither in [[RFC9275](#)] nor [[RFC9240](#)]) to see the same edge server as an entity in both domains. The design of ALTO, however, allows extensions that could be used to identify that an entity can be defined in several domains. These different domains and their related properties can be specified in extended ALTO property maps, as proposed in the next sections.

5. ALTO Design Considerations for Determining Service Edge

This section is in progress and gathers the ALTO features that are needed to support the exposure of both networking capabilities and compute capabilities in ALTO Maps.

In particular, ALTO Entity Property Maps defined in [[RFC9240](#)] can be extended. [[RFC9240](#)] generalizes the concept of endpoint properties to domains of other entities through property maps. Entities can be defined in a wider set of entity domains such as IPv4, IPv6, PID, AS, ISO3166-1 country codes or ANE. In addition, RFC 9240 specifies how properties can be defined on entities of each of these domains.

5.1. Example of Entity Definition in Different Domains

As there can be applications that do not necessarily need both compute and networking information, it is fine to keep the entity domains separate, each with their own native properties. However, some applications need information on both topics, and a scalable and flexible solution consists in defining an ALTO property type, that:

- *Indicates that an entity can be defined in several domains;

- *Specifies, for an entity, the other domains where this entity is defined.

For instance, one possible approach is to introduce entity properties that list other entity domains where an edge server is identified. This property type should be usable in all entity domains types. The following provides an example where the property "entity domain mapping" lists the other domains in which an entity is defined.

Suppose an edge server is identified as follows:

- *In the IPV4 domain, with an IP address, e.g., ipv4:1.2.3.4;

- *In the ANE domain, with an identifier, e.g., ane:DC10-HOST1.

To get information on this edge server as an entity in the "ipv4" entity domain, an ALTO client can query the properties "pid" and "entity-domain-mappings" and obtain a response as follows:

```
POST /propmap/lookup/dc-ip HTTP/1.1
Host: alto.example.com
Accept: application/alto-propmap+json,application/alto-error+json
Content-Length: TBC
Content-Type: application/alto-propmapparams+json
{
  "entities" : ["ipv4:1.2.3.4"],
  "properties" : [ "pid", "entity-domain-mappings"]
}
```

```
HTTP/1.1 200 OK
Content-Length: TBC
Content-Type: application/alto-propmap+json
{
  "meta" : {},

  "property-map": {
    "ipv4:1.2.3.4" :
      {"pid" : DC10,
       "entity-domain-mappings" : ["ane"]}
  }
}
```

To get information on this edge server as an entity in the "ane" entity domain, an ALTO client can query the properties "entity-domain-mappings" and "network-address" and obtain a response as follows:


```

POST /propmap/lookup/dc-ane HTTP/1.1
Host: alto.example.com
Accept: application/alto-propmap+json,application/alto-error+json
Content-Length: TBC
Content-Type: application/alto-propmapparams+json
{
  "entities" : ["ane:DC10-HOST1"],
  "properties" : [ "entity-domain-mappings", "network-address"]
}

HTTP/1.1 200 OK
Content-Length: TBC
Content-Type: application/alto-propmap+json
{
  "meta" : {},

  "property-map": {
    "ane:DC10-HOST1"
      {"entity domain mappings" : ["ipv4"]",
        "network-address" : ipv4:1.2.3.4}
      }
  }
}

```

Thus, if the ALTO Client sees the edge server as an entity with a network address, it knows that it can see the server as an ANE on which it can query relevant properties.

Further elaboration will be provided in future versions of this document.

5.2. Definition of Flavors in ALTO Property Map

The ALTO Entity Property Maps [[RFC9240](#)] generalize the concept of endpoint properties to domains of other entities through property maps. The term "flavor" or "instance" refers to an abstracted set of computing resources, with well-specified properties such as CPU, RAM and Storage. Thus, a flavor can be seen as an ANE with properties defined in terms of TIFSA. A flavor or instance is a group of 1 or more elements that can be reached via one or more network addresses. So an instance can also be seen as a PID that groups one or more IP addresses. In a context such as the one defined in CNTT, an ALTO property map could be used to expose TIFSA information of potential candidate flavors, i.e., potential NFVI-PoPs where an application or service can be deployed.

[Figure 2](#) below shows an example of an ALTO property map with property values grouped by flavor name.

flavor -name	type (T)	inter face (I)	f-c pu (F)	f-ra m (F)	f-di sk (F)	f-b w (F)	S	A
small-1	basic	1 Gbps	1	512 MB	1 GB	1 G bps		
small-2	network- intensive	9 Gbps	1	512 MB	1 GB	1 G bps		
medium-1	network- intensive	25 Gbps	2	4 GB	40 GB	1 G bps		
large-1	compute- intensive	50 Gbps	4	8 GB	80 GB	1 G bps		
large-2	compute- intensive	100 Gbps	8	16 GB	160 GB	1 G bps		

Figure 2: ALTO Property Map.

The following example uses ALTO's filtered property map to request properties "type", "cpu", "ram", and "disk" on five ANE flavors named "small-1", "small-2", "medium-1", "large-1", "large-2" defined in the example before.

```

POST /propmap/lookup/ane-flavor-name 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" : ["small-1",
                "small-2",
                "medium-1",
                "large-1"],
                "large-2"],
  "properties" : [ "type", "cpu", "ram", "disk"]
}

HTTP/1.1 200 OK
Content-Length: 295
Content-Type: application/alto-propmap+json
{
  "meta" : {
  },
  "property-map": {
    "small-1":
      {"type" : "basic", "cpu" : 1,
        "ram" : "512MB", "disk" : 1GB},
    "small-2":
      {"type" : "network-intensive", "cpu" : 1,
        "ram" : "512MB", "disk" : 1GB},
    "medium-1":
      {"type" : "compute-intensive", "cpu" : 2,
        "ram" : "4GB", "disk" : 40GB},
    "large-1":
      {"type" : "compute-intensive", "cpu" : 4,
        "ram" : "8GB", "disk" : 80GB},
    "large-2":
      {"type" : "compute-intensive", "cpu" : 8,
        "ram" : "16GB", "disk" : 160GB},
  }
}

```

Figure 3: Filtered Property Map query example.

6. Use Cases

6.1. Open Abstraction for Edge Computing

As shown in this document, modern applications such as AR/VR, V2X, or IoT, require bringing compute closer to the edge in order to meet strict bandwidth, latency, and jitter requirements. While this deployment process resembles the path taken by the main cloud

providers (notably, AWS, Facebook, Google and Microsoft) to deploy their large-scale datacenters, the edge presents a key difference: datacenter clouds (both in terms of their infrastructure and the applications run by them) are owned and managed by a single organization, whereas edge clouds involve a complex ecosystem of operators, vendors, and application providers, all striving to provide a quality end-to-end solution to the user. This implies that, while the traditional cloud has been implemented for the most part by using vertically optimized and closed architectures, the edge will necessarily need to rely on a complete ecosystem of carefully designed open standards to enable horizontal interoperability across all the involved parties. This document envisions ALTO playing a role as part of the ecosystem of open standards that are necessary to deploy and operate the edge cloud.

As an example, consider a user of an XR application who arrives at his/her home by car. The application runs by leveraging compute capabilities from both the car and the public 5G edge cloud. As the user parks the car, 5G coverage may diminish (due to building interference) making the home local Wi-Fi connectivity a better choice. Further, instead of relying on computational resources from the car and the 5G edge cloud, latency can be reduced by leveraging computing devices (PCs, laptops, tablets) available from the home edge cloud. The application's decision to switch from one domain to another, however, demands knowledge about the compute and communication resources available both in the 5G and the Wi-Fi domains, therefore requiring interoperability across multiple industry standards (for instance, IETF and 3GPP on the public side, and IETF and LF Edge [[LF-EDGE](#)] on the private home side). ALTO can be positioned to act as an abstraction layer supporting the exposure of communication and compute information independently of the type of domain the application is currently residing in.

Future versions of this document will elaborate further on this use case.

6.2. Optimized placement of microservice components

Current applications are transitioning from a monolithic service architecture towards the composition of microservice components, following cloud-native trends. The set of microservices can have associated SLOs which impose constraints not only in terms of required compute resources (CPU, storage, ...) dependent on the compute facilities available, but also in terms of performance indicators such as latency, bandwidth, etc, which impose restrictions in the networking capabilities connecting the computing facilities. Even more complex constraints, such as affinity among certain microservices components could require complex calculations

for selecting the most appropriate compute nodes taken into consideration both network and compute information.

Thus, service/application orchestrators can benefit from the information exposed by ALTO at the time of deciding the placement of the microservices in the network.

7. Security Considerations

TODO Security

8. IANA Considerations

This document has no IANA actions.

9. Conclusions

Telco networks will increasingly contain a number of interconnected data centers and edge clouds of different sizes and characteristics, allowing flexibility in the dynamic deployment of functions and applications for advanced 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 these distributed computing environments. The result of these discussions will be reflected in future versions of this draft.

10. References

10.1. 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>>.
- [RFC9275] Gao, K., Lee, Y., Randriamasy, S., Yang, Y., and J. Zhang, "An Extension for Application-Layer Traffic Optimization (ALTO): Path Vector", RFC 9275, DOI

10.17487/RFC9275, September 2022, <<https://www.rfc-editor.org/info/rfc9275>>.

[RFC9240] Roome, W., Randriamasy, S., Yang, Y., Zhang, J., and K. Gao, "An Extension for Application-Layer Traffic Optimization (ALTO): Entity Property Maps", RFC 9240, DOI 10.17487/RFC9240, July 2022, <<https://www.rfc-editor.org/info/rfc9240>>.

[I-D.ietf-teas-sf-aware-topo-model]

Bryskin, I., Liu, X., Lee, Y., Guichard, J., Contreras, L. M., Ceccarelli, D., Tantsura, J., and D. Shyti, "SF Aware TE Topology YANG Model", Work in Progress, Internet-Draft, draft-ietf-teas-sf-aware-topo-model-11, 12 March 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-teas-sf-aware-topo-model-11>>.

[I-D.llc-teas-dc-aware-topo-model] Lee, Y., Liu, X., and L. M. Contreras, "DC aware TE topology model", Work in Progress, Internet-Draft, draft-llc-teas-dc-aware-topo-model-02, 11 July 2022, <<https://datatracker.ietf.org/doc/html/draft-llc-teas-dc-aware-topo-model-02>>.

[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>>.

[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>>.

10.2. Informative References

[CNTT] "Cloud Infrastructure Telco Taskforce Reference Model", <https://github.com/cntt-n/CNTT/wiki> , June 2022.

[GSMA] "Cloud Infrastructure Reference Model, Version 2.0", <https://www.gsma.com/newsroom/wp-content/uploads//NG.126-v2.0-1.pdf> , October 2021.

[Anuket] "Anuket Project", <https://wiki.anuket.io/> , October 2022.

[LF-EDGE] "Linux Foundation Edge", <https://www.lfedge.org/> , March 2023.

Acknowledgments

TODO acknowledge.

Authors' Addresses

Luis M. Contreras
Telefonica

Email: luismiguel.conterasmurillo@telefonica.com

Sabine Randriamasy
Nokia Bell Labs

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

Jordi Ros-Giralt
Qualcomm Europe, Inc.

Email: jros@qti.qualcomm.com

Danny Lachos
Benocs

Email: dlachos@benocs.com

Christian Rothenberg
Unicamp

Email: chesteve@dca.fee.unicamp.br