

ALTO Extension: Path Vector

draft-ietf-alto-path-vector-11

Kai Gao, Young Lee, Sabine Randriamasy, Y. Richard Yang, Jensen Zhang

July 27, IETF 108@Virtual

Summary

- The Path Vector extension aims to provide abstracted network components along the path between a source and a destination
- Use cases: bandwidth reservation for large-scale data analytics, context-aware data transfers (capabilities of access network, etc.), CDN and service edge discovery
- Key abstractions:
 - **Abstract Network Element (ANE)**: A network-agnostic abstraction of network components, associated with different properties
 - **Path Vector (PV)**: An array of ANE names that represents the ANEs on the path between a <src, dst> pair
 - ANE Property Map: A Property Map that maps ANE names to the associated properties
- Key designs:
 - The Path Vector extension extends the Cost Map and Endpoint Cost Map to convey how each <src, dst> pair traverses the ANEs: **a new cost type is defined**
 - The Path Vector extension uses Property Map to convey the association between ANEs and their properties: **a new domain is defined, two initial properties are defined, entries are added to corresponding IANA registries following the procedures defined in the Unified Property draft**
 - For consistency and server security, the two pieces of information are provided together in **a multipart response**.
- Current status: Waiting to issue WGLC

Updates in -11

The problem and proposed solution are better clarified

- We summarize 3 general additional requirements using the flow scheduling example, along other use cases in the broader context
- ... and present how the extension fulfills the requirements
- In particular, we discuss *ephemeral ANE* (returned by the Path Vector response) and *persistent ANE* (defined by another Unified Property Map resource) and how they are handled

* The network needs to expose more detailed routing information to show the shared bottlenecks.

* The network needs to provide the necessary abstraction to hide the real topology information while providing enough information to applications.

The path vector extension defined in this document propose a solution



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 essential information on intermediate network components on the path of a <source, destination> pair that are critical to the QoE of the overlay application.

AR2: An ALTO server must provide essential information on how the paths of different <source, destination> pairs share a common network component.

AR3: An ALTO server must provide essential information on the properties associated to the network components.

The Path Vector extension defined in this document propose a solution to provide these details.

4.2. Recent Use Cases

While the multiple flow scheduling problem is used to help identify the additional requirements, the Path Vector extension can be applied to a wide range of applications. This section highlights some real use cases that are recently reported. See [I-D.bernstein-alto-topo] for a more comprehensive survey of use cases where extended network topology information is needed.

Updates in -11

Path Vector draft is now synchronized with Unified Property Map draft (version -12)

- ANE domain and ANE properties are defined and registered to IANA following the new procedure
- In particular, the defining resource of ANE is now Property Map (with media type `application/alto-propmap+json`) which apply to both ephemeral ANE and persistent ANE
- The entity identifiers used in the ANE Property Map and in the `persistent-entity-id` follow the naming convention of entity identifiers in the Unified Property Map draft

6.4.2. New ANE Property Type: Persistent Entity ID

Identifier: `"persistent-entity-id"`

Intended Semantics: The persistent entity ID property is the entity identifier of the persistent ANE associated with an ephemeral ANE. The value of this property is encoded with the format defined in Section 5.1.3 of [I-D.ietf-alto-unified-props-new]. In this format, the entity ID combines:

6.2.4. Media Type of Defining Resource

When resource specific domains are defined with entities of domain type "ane", the defining resource for entity domain type "ane" MUST be a Property Map. The media type of defining resources for the "ane" domain is:

`application/alto-propmap+json`

Specifically, the defining resource of ephemeral ANEs is the Property Map part of the multipart response. The defining resource of persistent ANEs is the Property Map on which standalone queries for properties of persistent ANEs are made.

in Figure 5. Assume the ALTO server has a Property Map resource called "mec-props" that defines persistent ANEs "MEC1" and "MEC2" that represent the corresponding mobile edge computing (MEC) clusters. The "persistent-entity-id" of the ephemeral ANE that is associated with MEC1 has the value "mec-props.ane:MEC1".

Updates in -11

The specification is better clarified

- Fixed the inconsistent use of the "start" parameter in multipart messages
- Examples are added for request/response
- Fixed inappropriate/inconsistent letter cases (e.g., *CAN*-> *can* in a normal sentence, *path vector*-> *Path Vector* throughout the paper)

Old version is using "first" part and "second" part to refer to parts in a multipart message, which may be incorrect when "start" parameter is used

<p>start: The start parameter MUST be a quoted string where the quoted part has the same value as the "Resource-Id" header in the first part.</p>	<p>start: The start parameter is as defined in [RFC2387]. If present, it MUST have the same value as the "Resource-Id" header of the Path Vector part.</p>
<p>boundary: The boundary parameter is as defined in [RFC2387].</p>	<p>boundary: The boundary parameter is as defined in [RFC2387].</p>
<p>The body of the response consists of two parts.</p>	<p>The body of the response consists of two parts:</p>
<p>The first part MUST include "Resource-Id" and "Content-Type" in its header. The value of "Resource-Id" MUST have the format of a Part Resource ID. The "Content-Type" MUST be "application/alto-costmap+json".</p>	<p>* The Path Vector part MUST include "Resource-Id" and "Content-Type" in its header. The value of "Resource-Id" MUST have the format of a Part Resource ID. The "Content-Type" MUST be "application/alto-costmap+json".</p>
<p>The body of the first part MUST be a JSON object with the same format as defined in Section 11.2.3.6 of [RFC7285]. The JSON object MUST include the "vtag" field in the "meta" field, which provides the version tag of the returned cost map. The resource ID of the version tag MUST follow the format in Section 3.3.2. 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.</p>	<p>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]. The JSON object MUST include the "vtag" field in the "meta" field, which provides the version tag of the returned cost map. The resource ID of the version tag MUST follow the format in Section 5.3.2. 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.</p>
<p>The second part MUST also include "Resource-Id" and "Content-Type" in its header. The value of "Resource-Id" has the format of a Part Resource ID. The "Content-Type" MUST be "application/alto-propmap+json".</p>	<p>* The Unified Property Map part MUST also include "Resource-Id" and "Content-Type" in its header. The value of "Resource-Id" has the format of a Part Resource ID. The "Content-Type" MUST be "application/alto-propmap+json".</p>
<p>The body of the second part MUST be a JSON object with the same format as defined in Section 4.6 of [I-D.ietf.alto-unified-prop-new]. The JSON object MUST include the</p>	<p>The body of the Unified Property Map part MUST be a JSON object with the same format as defined in Section 4.6 of [I-D.ietf.alto-unified-prop-new]. The JSON object MUST include the</p>

<p>5.1.4. Capabilities</p>	<p>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.</p> <pre>POST /costmap/pv HTTP/1.1 Host: alto.example.com Accept: multipart/related;type=application/alto-costmap+json, application/alto-error+json Content-Length: [TBD] 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"] }</pre>
	<p>7.1.4. Capabilities</p>

Conclusion

- A lot of clarification to make the document easier to follow
- PV-11 is fully synchronized with UP-12, and can move ahead as a bundle

- Issue WGLC?

Backup

Updates in -11

- The problem and proposed solution are better clarified
 - We summarize 3 general additional requirements using the flow scheduling example, along other use cases in the broader context
 - ... and present how the extension fulfills the requirements
 - In particular, we discuss *ephemeral ANE* (returned by the Path Vector response) and *persistent ANE* (defined by another Unified Property Map resource) and how they are handled
- Path Vector draft is now synchronized with Unified Property Map draft (version -12)
 - ANE domain and ANE properties are defined and registered to IANA following the new procedure
 - In particular, the defining resource of ANE is now Property Map (with media type `application/alto-propmap+json`) which apply to both ephemeral ANE and persistent ANE
 - The entity identifiers used in the ANE Property Map and in the `persistent-entity-id` follow the naming convention of entity identifiers in the Unified Property Map draft
- The specification is better clarified
 - Fixed the inconsistent use of the “`start`” parameter in multipart messages
 - Examples are added for request/response
 - Fixed inappropriate/inconsistent letter cases (e.g., *CAN* -> *can* in a normal sentence, *path vector* -> *Path Vector* throughout the paper)