ALTO WG
Internet-Draft
Intended status: Standards Track
Expires: September 14, 2017

G. Bernstein
Grotto Networking
S. Chen
Tongji University
K. Gao
Tsinghua University
Y. Lee
Huawei
W. Roome
M. Scharf
Nokia
Y. Yang
Yale University
J. Zhang
Tongji University
March 13, 2017

ALTO Extension: Path Vector Cost Mode
draft-yang-alto-path-vector-04.txt

Abstract

The Application-Layer Traffic Optimization (ALTO) Service has defined network and cost maps to provide basic network information, where the cost maps allow only scalar (numerical or ordinal) cost mode values. This document introduces a new cost mode called path-vector to allow ALTO clients to support use cases such as capacity regions for applications. This document starts with a non-normative example called multi-flow scheduling (or capacity region) to illustrate that ALTO cost maps without path vectors cannot provide sufficient information. This document then defines path-vector as a new cost mode.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

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

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

working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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

This Internet-Draft will expire on September 14, 2017.

Table of Contents

1. Introduction

The ALTO base protocol [RFC7285] is designed for exposing network information through services such as the Network Map service and the Cost Map service. These services use the extreme "single-node" abstraction, which represents a whole network with a single node and hosts are connected to the node's access ports.

Although the "single-node" abstraction works well for many settings, new use cases, such as inter-datacenter data transfers and scientific high-performance computing data transfers, require additional network information beyond the single-node abstraction, to support application capabilities, in particular, the ability of application flow scheduling.

Specifically, providing network information to support application flow scheduling introduces multiple complexities. First, the underlying assumption of existing ALTO services is single-commodity flows. Hence, given the flow from a source to a destination, ALTO computes the network metrics of the flow and returns them to the application. The metrics for different flows are independent. Application flow scheduling, however, requires network information to compute application-desirable multi-commodity flows, where multiple flows under the control of the same application may share common network bottlenecks. This requirement is beyond the capability of the single-node abstraction adopted by the base ALTO protocol. Second, some flow scheduling problems may consider end-to-end metrics at the same time and thus require multiple costs to be updated simultaneously. Such a requirement, even though already addressed by [I-D.ietf-alto-multi-cost], still needs to be handled very carefully. Third, flow scheduling can be conducted with several independent sets of flows. Using the cross product of "src" and "dst" lists will introduce a lot of redundancies.

To address these complexities in supporting the new flow scheduling use case, this document specifies a path vector extension to the base ALTO protocol. This extension introduces a new family of cost types, which uses "path-vector" as cost mode and "ne" (network element) or "ane" (abstract network element) as cost metric. It also extends "Unified Property Map" defined in [I-D.roome-alto-unified-props] to address the issue of scalability and consistency in providing path vectors with fine-grained information, and declares "pid-flows" and

"endpoint-flows" to support queries for multiple independent flow sets. This document also registers new domains, entity specification and properties in the ALTO Entity Domain Registry.

The document is organized as follows. Section 2 gives an overview of the path vector extension. Section 4 gives an example of flow scheduling to illustrate the need to introduce cost mode "path-vector" and new cost metrics. Section 5 specifies the new cost mode and cost metrics, new domain types and entity properties, new resource Network Element Property Map, and protocol extensions for (Filtered) Cost Maps and Endpoint Cost Services. Section 6 presents examples of Information Resources, requests and responses. Section 7 discusses the compatibility issues with some other proposed ALTO extensions. Section 8 lists several to-be-determined design decisions. Section 9 discusses about security and Section 10 discusses about IANA Considerations.

2. Overview of Approach

This section presents a non-normative overview of the path vector extension defined in this document. It assumes the readers are familiar with Cost Map and Endpoint Cost Service defined in [RFC7285], and Unified Property Map defined in [I-D.roome-alto-unified-props].

2.1. Graph Model and Path Vector Data Format

In this document, the graph model presented to ALTO clients is represented by path vectors. A path vector between two entities (either PIDs or endpoints) is an array of Network Element Names, where each Network Element Name represents a network element (usually a link) in the path between the two entities.

A specific Network Element Name MUST represent the same network element in the same ALTO Network Element Property Map resource. Thus, ALTO clients can find the flows that share this specific network element by finding the source-destination pairs whose corresponding path vectors contain the Network Element Name.

The cost entries contained in an ALTO Cost Map or Endpoint Cost Map are formally defined in Section 11.2.3.6 [RFC7285] to be any type of JSON value. But the section also suggests that implementations may assume the cost values are numbers unless specifically defined by an extension. This document extends the definition of Cost Map and Endpoint Cost Map to allow the returned cost to be a path vector, which is a JSONArray of Network Element Names.

An example can be found in Section 6.3.

2.2. Network Element Property Map Data Format

An ALTO client may need not know all attributes associated with all network elements. Thus, Network Element Property Map is provided so after an ALTO client obtains the path vectors, it can use Network Element Names to selectively query the associated attributes in the corresponding Network Element Property Map.

2.3. Flow-based Query Data Format

Flow scheduling may involve multiple sets of flows which have different source-destination combinations. Using source and destination lists can lead to a lot of redundancies. To allow more flexible specification of path vector requests, two new filter types for ReqFilteredCostMap and ReqEndpointCostMap are specified in this document.

2.4. Routing State Abstraction Service

For security and scalability considerations, this document specifies an optional feature called Routing State Abstraction (RSA). Routing State Abstraction feature compresses the original path vector information without loss of equivalence. A Routing State Abstraction compression feature MUST be applied only when a (Filtered) Cost Map or Endpoint Cost Service provides the path vector cost type with cost metric being "ane".

3. Changes Since Version -03

- Define "ne" and "ane" as the only cost metrics of the cost mode "path-vector".
- Define new entity domains for network property map and add the "availbw", "delay" property to these domains.
- Use Unified Property service to query properties of network elements.
- Augment the request schema of the (Filtered) Cost Map and Endpoint Cost Service.

4. Use Case: Capacity Region for Multi-Flow Scheduling

Consider the case that routing is given. Then what application-layer traffic optimization will focus on is traffic scheduling among application-layer paths. Specifically, assume that an application has control over a set of flows $F = \{f_1, f_2, ..., f_{|F|}\}$. If routing is given, what the application can control is $x_1, x_2, ...,$

x_|F|, where x_i is the amount of traffic for flow i. Let x = [x_1, ..., x_|F|] be the vector of the flow traffic amounts. Due to shared links, feasible values of x where link capacities are not exceeded can be a complex polytype.

Specifically, consider a network as shown in Figure 1. The network has 7 switches (sw1 to sw7) forming a dumb-bell topology. Switches sw1/sw3 provide access on one side, sw2/sw4 provide access on the other side, 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 each link is 100 Mbps.

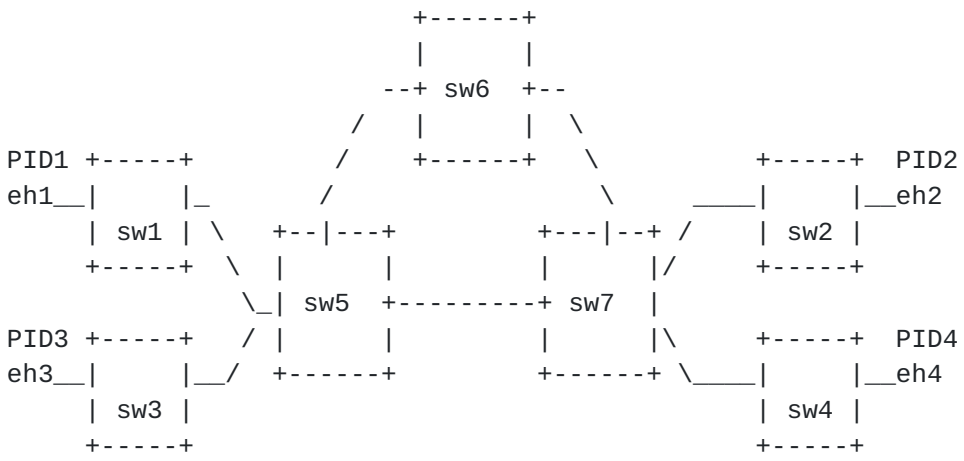

Figure 1: Raw Network Topology.

The single-node ALTO topology abstraction of the network is shown in Figure 2.

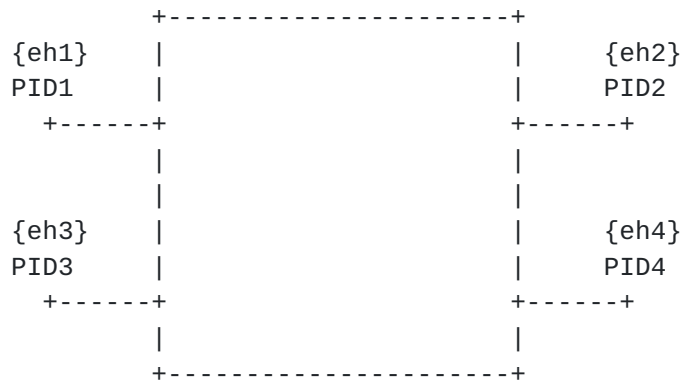

Figure 2: Base Single-Node Topology Abstraction.

Consider an application overlay (e.g., a large data analysis system) which needs to schedule the traffic among a set of end host source-destination pairs, say eh1 -> eh2, and eh3 -> eh4. The application

can request a cost map providing end-to-end available bandwidth, using 'availbw' as cost-metric and 'numerical' as cost-mode.

Assume that the application receives from the ALTO server that the bandwidth of eh1 -> eh2 and eh3 ->eh4 are both 100 Mbps. It cannot determine that if it schedules the two flows together, whether it will obtain a total of 100 Mbps or 200 Mbps. This depends on whether the routing paths of the two flows share a bottleneck in the underlying topology:

- Case 1: If eh1 -> eh2 and eh3 -> eh4 use different paths, for example, when the first uses sw1 -> sw5 -> sw7 -> sw2, and the second uses sw3 -> sw5 -> sw6 -> sw7 -> sw4. Then the application will obtain 200 Mbps.

- Case 2: If eh1 -> eh2 and eh3 -> eh4 share a bottleneck, for example, when both use the direct link sw5 -> sw7, then the application will obtain only 100 Mbps.

To allow applications to distinguish the two aforementioned cases, the network needs to provide more details. The path vector extension defined in this document resolves this issue.

See [I-D.bernstein-alto-topo] for a survey of use-cases where extended network topology information is needed.

5. Protocol Extensions

This section formally specifies the path vector extension.

5.1. Cost Type

The path vector extension defined in this document extends the Cost Types as specified in Section 6.1 of [RFC7285] .

5.1.1. Cost Metric

This document specifies two new cost metrics: "ne" and "ane". Both cost metrics are of type CostMetric as defined in Section 10.6 of [RFC7285]. These cost metrics MUST NOT be used when the cost mode is not "path-vector" unless it is explicitly specified in a future extension. An ALTO server with path vector extension MUST support at least one of these two cost metrics.

Cost metric "ne": This cost metric MUST be encoded as the JSONString "ne". When cost metric is "ne", Network Element Names contained in the path vectors MUST be resource-specific. In this case, different path vector queries to the same (Filtered) Cost Map or

Endpoint Cost Service MUST have the same Network Element Property Map in the responses.

Cost metric "ane": This cost metric MUST be encoded as the JSONString "ane". When cost metric is "ane", Network Element Names contained in the path vector MUST be query-specific. In this case, different path vector queries to the same (Filtered) Cost Map or Endpoint Cost Service MAY have different Network Element Property Maps.

5.1.2. Cost Mode: Path Vector

This document extends the cost mode as defined in Section 10.5 of [RFC7285] by allowing an ALTO server to provide a new cost mode other than "numerical" and "ordinal". The path vector cost mode is of type CostMode and is encoded as the JSONString "path-vector".

A (Filtered) Cost Map resource or Endpoint Cost Service, when queried with this cost mode, MUST return a CostMapData or EndpointCostMapData whose costs are JSONArrays of type NetworkElementName as specified in Section 5.2.

This cost mode MUST be used with either cost metric "ne" or "ane" unless it is explicitly specified by a future extension.

5.2. Network Element Name

This document also extends [RFC7285] with a new basic data type: Network Element Name. A Network Element Name is of type EntityAddr as defined in Section 2.3 of [I-D.roome-alto-unified-props] and is encoded as a JSONString. A Network Element Name MUST be an EntityAddr either of the NE domain (Section 5.3.1) or of the ANE domain (Section 5.3.2).

5.3. Entity Domains

This document specifies two new domains in addition to the ones in [I-D.roome-alto-unified-props].

5.3.1. NE Domain

5.3.1.1. Domain Name

ne

5.3.1.2. Domain-Specific Entity Addresses

Entity address of NE domain MUST be encoded as a JSONString with the same format as PID name defined in Section 10.1 of [RFC7285].

5.3.2. ANE Domain

5.3.2.1. Domain Name

ane

5.3.2.2. Domain-Specific Entity Addresses

The same as Section 5.3.1.2.

5.4. (Filtered) Network Element Property Map Resource

This document extends the base ALTO protocol with a new (filtered) resource: (Filtered) Network Element Property Map.

A Network Element Property Map MUST be a Property Map as defined in Section 4 of [I-D.roome-alto-unified-props] and a Filtered Network Element Property Map MUST be a Filtered Property Map as defined in Section 5 of [I-D.roome-alto-unified-props].

5.4.1. Media Type

The media type of a (Filtered) Network Element Property Map resource is "application/alto-propmap+json".

5.4.2. HTTP Method

An ALTO Network Element Property Map is requested using the HTTP GET method.

An ALTO Filtered Network Element Property Map is requested using the HTTP POST method.

5.4.3. Accept Input Parameters

Network Element Property Map does not accept any input parameters.

The input parameters of a Filtered Network Element Property Map MUST conform to the format in Section 5.3 of [I-D.roome-alto-unified-props]. The EntityAddr in the request MUST have the same format as Network Element Name specified in Section 5.2.

5.4.4. Capabilities

A (Filtered) Network Element Property Map MUST have capabilities "domain-types" and "prop-types" as defined in Section 4.4 of [I-D.roome-alto-unified-props]. The "domain-types" capability MUST contain either "ne" or "ane" but not both at the same time and the "prop-types" capability MUST contain property type "availbw".

5.4.5. Uses

None.

5.4.6. Response

The "vtag" field MUST be included in the "meta" field of a response to a (Filtered) Network Element Map, with the same format as defined in Section 11.2.1.6 of [RFC7285].

The response is the same as for the Property Map, as defined in Section 4.6 of [I-D.roome-alto-unified-props], except that only the requested entities and properties are returned for Filtered Network Element Map. Examples can be found in Section 6.2.

5.5. Cost Map Extensions

5.5.1. Media Types

The same as Section 11.2.3.1 of [RFC7285].

5.5.2. HTTP Method

The same as Section 11.2.3.2 of [RFC7285].

5.5.3. Accept Input Parameters

The same as Section 11.2.3.3 of [RFC7285].

5.5.4. Capabilities

If a Cost Map resource supports the path vector extension defined in this document, its "cost-type-names" capability MUST have exactly one single cost type with the cost mode being "path-vector" and the cost metric being either "ne" or "ane", unless it is explicitly specified by a future extension.

5.5.5. Uses

The same as Section 11.2.3.5 of [RFC7285].

5.5.6. Response

The response has the same format as in Section 4.1.3 of [I-D.ietf-alto-multi-cost], except the follows.

- If cost mode is "path-vector", the costs MUST be JSONArrays of Network Element Names.
- If cost mode is "path-vector", the "meta" field MUST include the "nep-map" field, whose value is of type IRDResourceEntry as defined in Section 9.2.2 of [RFC7285] and represents the Filtered Network Element Property Map associated with this Cost Map as defined in Section 5.4. An ALTO server providing this resource MUST verify the following conditions are met:
- If cost metric of this Cost Map resource is "ne", the "nep-map" entry MUST have "domain-types" capability which includes domain name "ne".
- If cost metric of this Cost Map resource is "ane", the "nep-map" entry MUST have "domain-types" capability which includes domain name "ane".

ALTO servers MUST also verity the conditions in Section 5.1.1 are also met.

5.6. Filtered Cost Map Extensions

5.6.1. Media Type

The same as Section 11.3.2.1 of [RFC7285].

5.6.2. HTTP Method

The same as Section 11.3.2.2 of [RFC7285].

5.6.3. Accept Input Parameters

This document extends the ReqFilteredCostMap as follows:

```
  object {
    [CostType cost-type;]
    [CostType multi-cost-types<1..*>;]
    [CostType testable-cost-types<1..*>;]
    [JSONString constraints<0..*>;]
    [JSONString or-constraints<1..*><1..*>;]
    [PIDFilter pids;]
    [PIDFlow pid-flows<1..*>;]
  } ReqFilteredCostMap;

  object {
    PIDName src;
    PIDName dst;
  } PIDFlow;
```

pid-flows: A list of PID src to PID dst for which path costs are to be returned.

pids: As defined in Section 11.3.2.3 of [RFC7285].

cost-type, multi-cost-types, testable-cost-types, constraints, or-constraints:
As defined in Section 4.1.2 of [I-D.ietf-alto-multi-cost]. A valid query MUST be considered a path vector query, either when "cost-type" of this query exists with "path-vector" cost mode, or when "multi-cost-types" exists and exact one cost type uses "path-vector" cost mode. For a path vector query, the path vector cost type MUST follow the definition in Section 5.1, otherwise the ALTO server SHOULD return an error with error code "E_INVALID_FIELD_VALUE". If "multi-cost-types" contains multiple path vector cost types, ALTO servers SHOULD return an error with the error code "E_INVALID_FIELD_VALUE". If a query is a path vector query and its "constraints" or "or-constraints" field is present, the "testable-cost-types" field MUST be explicitly specified and MUST NOT include any path vector cost type. If a path vector cost type is included, an ALTO server SHOULD return an error with the error code "E_INVALID_FIELD_VALUE".

An ALTO client MUST specify either "pids" or "pid-flows", but MUST NOT specify both in a single query.

5.6.4. Capabilities

A Filtered Cost Map with the path vector extension MAY have the "flow-based-filter" in its IRD capabilities.

```
  object {
    JSONString cost-type-names<1..*>;
    [JSONBool cost-constraints;]
    [JSONNumber max-cost-types;]
    [JSONString testable-cost-type-names<1..*>;]
    [JSONBool flow-based-filter;]
  } FilteredCostMapCapabilities;
```

flow-based-filter: If the value is true, the ALTO server allows ALTO clients to use "pid-flows" to query the Filtered Cost Map. If false or not present, ALTO clients MUST NOT include this field in the queries and the ALTO server SHOULD return an error with the error code "E_INVALID_FIELD_TYPE" for the queries including this field.

cost-type-names and cost-constraints: As defined in Section 11.3.2.4 of [RFC7285].

max-cost-types and testable-cost-type-names: As defined in Section 4.1.1 of [I-D.ietf-alto-multi-cost]. If a cost type with "path-vector" mode is included in "cost-type-names", either the "testable-cost-type-names" field is explicitly specified which MUST NOT include any path vector cost type, or the "testable-cost-type-names" field is empty where ALTO clients MUST interpret this as specified in Section 4.1.1 of [I-D.ietf-alto-multi-cost] except that the resource MUST NOT accept tests on any path vector cost type.

5.6.5. Uses

The same as Section 5.5.5.

5.6.6. Response

The response MUST have the same format as defined in Section 5.5.6 with the supplement that if a query uses the field "pid-flows", the response MUST still conform to the CostMapData format defined in Section 11.2.3.6 of [RFC7285]. Examples can be found in Section 6.3.

5.7. Endpoint Cost Service Extensions

5.7.1. Media Type

The same as Section 11.5.1.1 of [RFC7285].

5.7.2. HTTP Method

The same as Section 11.5.1.2 of [RFC7285].

5.7.3. Accept Input Parameters

This document extends the input parameters of Endpoint Cost Service as follows:

```
  object {
    [CostType cost-type;]
    [CostType multi-cost-types<1..*>;]
    [CostType testable-cost-types<1..*>;]
    [JSONString constraints<0..*>;]
    [JSONString or-constraints<1..*><1..*>;]
    [EndpointFilter endpoints;]
    [EndpointFlow endpoint-flows<1..*>;]
  } ReqEndpointCostMap;

  object {
    TypedEndpointAddr src;
    TypedEndpointAddr dst;
  } EndpointFlow;
```

endpoint-flows: A list of source-destination endpoint pairs for which path costs are to be returned.

endpoints: As defined in Section 11.5.1.3 of [RFC7285].

cost-type, multi-cost-types, testable-cost-types, constraints, or-constraints:
As defined in Section 5.6.3.

ALTO clients MUST specify either "endpoints" or "endpoint-flows", but MUST NOT specify both in the same query.

5.7.4. Capabilities

This document defines EndpointCostMapCpabilities the same as FilteredCostMapCapabilities in Section 5.6.4.

If the value of capability flow-based-filter is true, the ALTO server MUST be able to process "endpoint-flows" in a query. If the value is false or not present, ALTO clients MUST assume the "endpoint-flows" is not supported and ALTO servers SHOULD return an error with the error code "E_INVALID_FIELD_TYPE" if "endpoint-flows" is included in the query.

5.7.5. Uses

The same as Section 11.5.1.5 of [RFC7285].

5.7.6. Response

The response has the same format as in Section 4.2.3 of [I-D.ietf-alto-multi-cost] for compatibility, except the follows.

- If cost mode is "path-vector", the costs MUST be JSONArrays of Network Element Names.
- If cost mode is "path-vector", the "meta" field MUST include the "nep-map" field, whose value is of type IRDResourceEntry as defined in Section 9.2.2 of [RFC7285] and represents the Filtered Network Element Property Map associated with this Endpoint Cost Service as defined in Section 5.4. An ALTO server providing this resource MUST verify the following conditions are met:
- If cost metric of this Endpoint Cost Service is "ne", the "nep-map" entry MUST have "domain-types" capability which includes domain name "ne".
- If cost metric of this Endpoint Cost Service is "ane", the "nep-map" entry MUST have "domain-types" capability which includes domain name "ane".

ALTO servers MUST also verity the conditions in Section 5.1.1 are also met.

Examples can be found in Section 6.5.

5.8. Optional: RSA Extension

5.8.1. Media Type

RSA extension MUST NOT change media types of (Filtered) Cost Map resource or Endpoint Cost Service.

5.8.2. HTTP Method

RSA extension MUST NOT change HTTP method for (Filtered) Cost Map or Endpoint Cost Service.

5.8.3. Accept Input Parameters

RSA extension SHOULD NOT change the input parameters of a Filtered Cost Map or an Endpoint Cost Service unless it is explicitly specified by a future extension.

5.8.4. Capabilities

If a (Filtered) Cost Map or an Endpoint Cost Service supports RSA extension, the "cost-type-names" MUST have one cost type with "path-vector" cost mode and "ane" cost metric, and ALTO clients MUST ignore this field when no such path vector cost type exists. The resource/service MUST also have the field "rsa" explicitly specified to "true" in the "capabilities" field. If the "rsa" field has a value of "false" or is not present, ALTO clients MUST assume this resource/service does not provide RSA compression.

An example can be found in Section 6.1.

5.8.5. Response

RSA extension SHOULD NOT change the output of a (Filtered) Cost Map or an Endpoint Cost Service unless it is explicitly specified in a future extension.

6. Examples

6.1. Information Resource Directory Example

Here is an example of an ALTO server's Information Resource Directory.

```
  "meta" {
    "cost-types": {
      "pv-ne": {
        "cost-mode": "pv",
        "cost-metric": "ne"
      },
      "pv-ane": {
        "cost-mode": "pv",
        "cost-metric": "ane"
      },
      "num-hopcount": {
        "cost-mode": "numerical",
        "cost-metric": "hopcount"
      },
      "num-routingcost": {
        "cost-mode": "numerical",
```

```
        "cost-metric": "routingcost"
      },
      "num-delay": {
          "cost-mode": "numerial",
          "cost-metric": "delay"
      },
      "num-availbw": {
        "cost-mode": "numerical",
        "cost-metric": "availbw"
      }
    }
  },
  "resource": {
    "default-network-map": {
      "uri": "http://alto.example.com/networkmap",
      "media-type": "application/alto-networkmap+json"
    },

    ... Filtered Cost Map Resource ...

    "filtered-multi-cost-map1": {
      "uri": "http://alto.example.com/costmap/multi/filtered1",
      "media-type": "application/alto-costmap+json",
      "accepts": "application/alto-costmapfilter+json",
      "uses": ["default-network-map"],
      "capabilities": {
        "max-cost-types": 3,
        "cost-type-names": ["pv-ne", "num-availbw", "num-hopcount"],
        "testable-cost-types-names": ["num-availbw", "num-hopcount"]
      }
    },

    "filtered-multi-cost-map2": {
      "uri": "http://alto.example.com/costmap/multi/filtered2",
      "media-type": "application/alto-costmap",
      "accepts": "application/alto-costmapfilter+json",
      "uses": ["default-network-map"],
      "capabilities": {
        "max-cost-types":2,
        "cost-type-names": ["pv-ne", "num-routingcost", "num-delay"],
        "cost-constraints": true
      }
    }
    ... Endpoint Cost Map Resource ...

    "default-endpoint-cost-map": {
      "uri": "http://alto.example.com//endpointcost/lookup/ne-ane",
      "media-type": "application/alto-endpointcostmap+json",
```

```
      "accepts": "application/alto-endpointcostparams+json",
      "capabilities": {
        "rsa": true,
        "max-cost-types": 4,
        "cost-type-names": ["pv-ne", "pv-ane", "num-routingcost", "num-
hopcount"],
        "testable-cost-types-names": ["num-hopcount", "num-routingcost"]
      }
    }
  }
```

6.2. Network Element Property Map Example

```
POST /propmap/lookup/nep-map HTTP/1.1
Host: alto.example.com
Accept: application/alto-propmap+json,application/alto-error+json
Content-Length: [TBD]
Content-Type: application/alto-propmapparams+json

{
  "entities" : [ "ne:ne12",
                 "ne:ne23",
                 "ne:ne34" ],
  "properties" : [ "availbw", "delay" ]
}

HTTP/1.1 200 OK
Content-Length: [TBD]
Content-Type: application/alto-propmap+json

{
  "meta": {
      "vtag": {
          "resource-id": "default-network-element-prop-map",
          "tag": "babbc14521772381472bffefff627813909875dd"
      }
  }
  "property-map": {
    "ne:ne12": { "availbw": "90", "delay": "30" },
    "ne:ne23": { "availbw": "80", "delay": "15" },
    "ne:ne34": { "availbw": "70", "delay": "25" }
  }
}
```

6.3. Filtered Cost Map Example #1

Assume that the available bandwidth between PID1 and PID3 is less than 50.

```
POST /costmap/multi/filtered1 HTTP/1.1
Host: alto.example.com
Accept: application/alto-costmap+json,application/alto-error+json
Content-Length: [TBD]
Content-Type: application/alto-costmapfilter+json

{
  "cost-type": {
    "cost-mode": "path-vector",
    "cost-metric": "ne"
  },
  "testable-cost-types": [
    { "cost-mode": "numerical", "cost-metric": "availbw" },
    { "cost-mode": "numerical", "cost-metric": "hopcount" }
  ],
  "or-constraints": [
    ["[0] ge 50", "[1] le 100"]
  ],
  "pid-flows": [
    { "src": "PID1", "dst": "PID2" }
    { "src": "PID1", "dst": "PID3" }
    { "src": "PID3", "dst": "PID4" }
  ]
}

HTTP/1.1 200 OK
Content-Length: [TBD]
Content-Type: application/alto-costmap+json

{
  "meta": {
    "dependent-vtags": [
      {
        "resource-id": "my-default-network-map",
        "tag": "75ed013b3cb58f896e839582504f622838ce670f"
      }
    ],
    "cost-type": {
      "cost-mode": "path-vector",
      "cost-metric": "ne"
    },
    "nep-map": {
      "uri": "http://alto.example.com/propmap/lookup/nep-map",
      "media-type": "application/alto-prompmap+json",
      "accepts": "application/alto-propmapparams+json",
      "capabilities": {
        "domain-types": ["ne"],
        "prop-types": ["availbw", "delay"]
```

```
          }
        }
      },
      "cost-map": {
        "PID1": { "PID2": [ "ne:ne12", "ne:ne23" ] },
        "PID3": { "PID4": [ "ne:ne23", "ne:ne34" ] }
      }
    }
```

6.4. Filtered Cost Map Example #2

Assume that the delay between PID1 and PID2 is greater than 80.

```
  POST /costmap/multi/filtered2 HTTP/1.1
  Host: alto.example.com
  Accept: application/alto-costmap+json,application/alto-error+json
  Content-Length: [TBD]
  Content-Type: application/alto-costmapfilter+json

  {
    "multi-cost-types": [
      { "cost-mode": "path-vector", "cost-metric": "ne" },
      { "cost-mode": "numerical", "cost-metric": "delay" }
    ],
    "testable-cost-types": [
      { "cost-mode": "numerical", "cost-metric": "delay" }
    ],
    "constraints": [ "[0] le 80" ],
    "pid-flows": [
      { "src": "PID1", "dst": "PID2" },
      { "src": "PID1", "dst": "PID3" },
      { "src": "PID3", "dst": "PID4" }
    ]
  }

  HTTP/1.1 200 OK
  Content-Length: [TBD]
  Content-Type: application/alto-costmap+json

  {
    "meta": {
      "dependent-vtags": [
        {
          "resource-id": "my-default-network-map",
          "tag": "75ed013b3cb58f896e839582504f622838ce670f"
        }
      ],
      "cost-type": {},
```

```
        "multi-cost-types": [
          { "cost-mode": "path-vector", "cost-metric": "ne"},
          { "cost-mode": "numerical", "cost-metric": "delay"}
        ],
        "nep-map": {
          "uri": "http://alto.example.com/propmap/lookup/nep-map",
          "media-type": "application/alto-prompmap+json",
          "accepts": "application/alto-propmapparams+json",
          "capabilities": {
            "domain-types": ["ne"],
            "prop-types": ["availbw", "delay"]
          }
        }
      },
      "cost-map": {
        "PID1": { "PID3": [ [ "ne:ne23", "ne:ne45" ], 60 ] },
        "PID3": { "PID4": [ [ "ne:ne23", "ne:ne34" ], 40 ] }
      }
    }
```

6.5. Endpoint Cost Map Example #1

Assume that the delay between ipv4:203.0.113.45 and ipv4:198.51.100.34 is greater than 100.

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

{
  "cost-type": {
    "cost-mode": "path-vector",
    "cost-metric": "ne"
  },
  "testable-cost-types": [
    {
      "cost-mode": "numerical",
      "cost-metric": "delay"
    }
  ],
  "constraints": [
    "[0] le 100"
  ],
  "endpoint-flows": [
    { "src": "ipv4:192.0.2.2", "dst": "ipv4:192.0.2.89" },
    { "src": "ipv4:192.0.2.2", "dst": "ipv4:198.51.100.34" },
```

```
      { "src": "ipv4:203.0.113.45", "dst": "ipv4:198.51.100.34" }
    ]
  }


  HTTP/1.1 200 OK
  Content-Length: [TBD]
  Content-Type: application/alto-endpointcost+json

  {
      "meta": {
      "cost-type": {
        "cost-mode": "path-vector",
        "cost-metric": "ne"
      },
      "nep-map": {
        "uri": "http://alto.example.com/propmap/lookup/nep-map",
        "media-type": "application/alto-prompmap+json",
        "accepts": "application/alto-propmapparams+json",
        "capabilities": {
          "domain-types": ["ne"],
          "prop-types": ["availbw", "delay"]
        }
      }
    },
    "endpoint-cost-map": {
      "ipv4:192.0.2.2": {
        "ipv4:192.0.2.89": [ "ne:ne12", "ne:ne23" ],
        "ipv4:198.51.100.34": [ "ne:ne12", "ne:ne24", "ne:ne45" ]
      }
    }
  }
```

6.6. Endpoint Cost Map Example #2

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

  {
    "multi-cost-types": [
    {
      "cost-mode": "path-vector",
      "cost-metric": "ane"
    },
    {
```

```
      "cost-mode": "numerical",
      "cost-metric": "delay"
    }],
    "endpoint-flows": [
      { "src": "ipv4:192.0.2.2", "dst": "ipv4:192.0.2.89" },
      { "src": "ipv4:192.0.2.2", "dst": "ipv4:198.51.100.34" },
      { "src": "ipv4:203.0.113.45", "dst": "ipv4:198.51.100.34" }
    ]
  }

  HTTP/1.1 200 OK
  Content-Length: [TBD]
  Content-Type: application/alto-endpointcost+json
  {
    "meta": {
      "cost-type": {},
      "multi-cost-types": [
      { "cost-mode": "path-vector", "cost-metric": "ane"},
      { "cost-mode": "numerical", "cost-metric": "delay"}
      ],
      "nep-map": {
        "uri": "http://alto.example.com/propmap/lookup/nep-map/31415926",
        "media-type": "application/alto-prompmap+json",
        "accepts": "application/alto-propmapparams+json",
        "capabilities": {
          "domain-types": ["ane"],
          "prop-types": ["availbw", "delay"]
        }
      }
    },
    "endpoint-cost-map": {
      "ipv4:192.0.2.2": {
        "ipv4:192.0.2.89": [ ["ane:ane12", "ane:ane23"], 45 ],
        "ipv4:198.51.100.34": [ ["ane:ane12", "ane:ane24"], 90 ]
      },
      "ipv4:203.0.113.45": {
        "ipv4:198.51.100.34": [ ["ane:ane34"], 120 ]
      }
    }
  }
```

7. Compatibility

7.1. Compatibility with Legacy ALTO Clients/Servers

Legacy ALTO clients SHOULD NOT send queries with path vector extension and ALTO servers with this extension SHOULD NOT have any compatibility issue. Legacy ALTO servers do not support cost types

with the "path-vector" cost mode and MUST NOT announce the extended cost types in IRD. Thus, ALTO clients MUST NOT send queries specified in this extension to legacy ALTO servers according to Section 11.3.2.3 [RFC7285].

7.2. Compatibility with Multi-Cost Extensions

Path Vector extension SHOULD be fully compatible with Multi-Cost extensions.

7.3. Compatibility with Incremental Update

There is no compatibility issue with incremental update extension.

8. Design Decisions and Discussions

8.1. Path Vector or Path Graph?

When we introduce the "path-vector" as a cost mode in the Cost Map, an unavoidable problem is how to handle multipath. Because a PID is a group of endpoints, it is common that there are multiple paths between two PIDs. The valid routing state information is all of the accessible paths. So in this scenario, the Cost Map Resource SHOULD provide the cost values including of the multiple paths.

A natural solution is to provide an array of path vectors as the cost value. Every path vector in this array means an accessible path between the source PID and the destination PID. It is different from the solution of the path vector extension which provides an array of network elements. So it requires to introduce a different cost mode. This document proposes this new cost mode named "path-graph".

However, the "path-graph" will increase the complexity of the Cost Map Response. Since the applications select ALTO as the protocol to get the network information rather than other topology-based solution such as I2RS, the major reason should be the simplicity. If we provide "path-graph" for each PID pairs, the ALTO client has to handle the complex data structure.

What's more, the "path-vector" is powerful enough to express multiple paths. The simple solution is to list the network elements of all accessible paths in a single path vector. This solution will lose the information about paths. Another solution is to define the path as a new type of network elements. In this way, the path vector can provide an array of paths. Each element of this array contains a path vector of network elements in the Network Element Property Map.

So in this document, we just introduce "path-vector" as the only required cost mode for routing state information.

8.2. Provide More General Calendar Extension?

Cost Calendar is proposed as a useful ALTO extension to provide the historical cost values for Filtered Cost Map Service and Endpoint Cost Service. Since path vector is an extension to these services, it SHOULD be compatible with Cost Calendar extension.

However, the calendar of a path-vector (Endpoint) Cost Map is insufficient for the application which requires the historical data of routing state information. The (Endpoint) Cost Map can only provide the changes of the paths. But more useful information is the history of network element properties which are recorded in the dependent Network Element Property Map.

Before the Unified Property Map is introduced as a new ALTO service, Filtered Cost Map Service and Endpoint Cost Service are the only resources which require the calendar supported. Because other resources don't have to be updated frequently. But Network Element Property Map as a use case of Unified Property Map will collect the real-time information of the network. It SHOULD be updated as soon as possible once the metrics of network elements change.

So the requirement is to provide a general calendar extension which not only meets the Filtered Cost Map and Endpoint Cost Service but also applies to the Property Map Service.

9. Security Considerations

We can identify multiple potential security issues. A main security issue is network privacy, as the path-vector information may reveal more network internal structures than the more abstract single-node abstraction. The network should consider protection mechanisms to reduce information exposure, in particular, in settings where the network and the application do not belong to the same trust domain. On the other hand, in a setting of the same trust domain, a key benefit of the path-vector abstraction is reduced information transfer from the network to the application.

The path-vector query may also reveal more information about the application. In particular, the application may reveal all potential transfers sites (e.g., where the data source is replicated, and where the potential replication sites are). The application should evaluate the potential privacy concerns.

Beyond the privacy issues, the computation of the path-vector is unlikely to be cachable, in that the results will depend on the particular requests (e.g., where the flows are distributed). Hence, this service may become an entry point for denial of service attacks on the availability of an ALTO server. Hence, authenticity and authorization of this ALTO service may need to be better protected.

10. IANA Considerations

10.1. ALTO Cost Mode Registry

This document specifies a new cost mode "path-vector". However, the base ALTO protocol does not have a Cost Mode Registry where new cost mode can be registered. This new cost mode will be registered once the registry is defined either in a revised version of [RFC7285] or in another future extension.

10.2. ALTO Cost Metric Registry

Two new cost metrics need to be registered in the "ALTO Cost Metric Registry", listed in Table 1.

Identifier	Intended Semantics
ne	See Section 5.1.1
ane	See Section 5.1.1

Table 1: ALTO Cost Metrics

10.3. ALTO Entity Domain Registry

As proposed in Section 9.2 of [I-D.roome-alto-unified-props], "ALTO Entity Domain Registry" is requested. Besides, two new domains are to be registered, listed in Table 2.

Identifier	Entity Address Encoding	Hierarchy & Inheritance
ne	See Section 5.3.1.2	None
ane	See Section 5.3.2.2	None

Table 2: ALTO Entity Domain

11. Acknowledgments

The authors would like to thank discussions with Andreas Voellmy, Erran Li, Haibin Son, Haizhou Du, Qiao Xiang, Tianyuan Liu, Xiao Shi, Xin Wang, and Yan Luo.

12. References

12.1. Normative References

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

12.2. Informative References

[I-D.amante-i2rs-topology-use-cases]
Medved, J., Previdi, S., Lopez, V., and S. Amante, "Topology API Use Cases", draft-amante-i2rs-topology-use-cases-01 (work in progress), October 2013.

[I-D.bernstein-alto-topo]
Bernstein, G., Yang, Y., and Y. Lee, "ALTO Topology Service: Uses Cases, Requirements, and Framework", draft-bernstein-alto-topo-00 (work in progress), October 2013.

[I-D.clemm-i2rs-yang-network-topo]
Clemm, A., Medved, J., Tkacik, T., Varga, R., Bahadur, N., and H. Ananthakrishnan, "A YANG Data Model for Network Topologies", draft-clemm-i2rs-yang-network-topo-01 (work in progress), October 2014.

[I-D.ietf-alto-cost-calendar]
Randriamasy, S., Yang, Y., Wu, W., Lingli, D., and N. Schwan, "ALTO Cost Calendar", draft-ietf-alto-cost-calendar-01 (work in progress), February 2017.

[I-D.ietf-alto-incr-update-sse]
Roome, W. and Y. Yang, "ALTO Incremental Updates Using Server-Sent Events (SSE)", draft-ietf-alto-incr-update-sse-03 (work in progress), September 2016.

[I-D.ietf-alto-multi-cost]
Randriamasy, S., Roome, W., and N. Schwan, "Multi-Cost ALTO", draft-ietf-alto-multi-cost-05 (work in progress), February 2017.

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

[I-D.roome-alto-unified-props]
Roome, W., "Extensible Property Maps for the ALTO Protocol", draft-roome-alto-unified-props-01 (work in progress), July 2016.

[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, <http://www.rfc-editor.org/info/rfc7285>.

Authors' Addresses

Greg Bernstein
Grotto Networking
Fremont, CA
USA

Email: gregb@grotto-networking.com

Shiwei Dawn Chen
Tongji University
4800 Caoan Road
Shanghai 201804
China

Email: dawn_chen_f@hotmail.com

Kai Gao
Tsinghua University
Beijing Beijing
China

Email: gaok12@mails.tsinghua.edu.cn

Young Lee
Huawei
TX
USA

Email: leeyoung@huawei.com

Wendy Roome
Nokia/Bell Labs
600 Mountain Ave, Rm 3B-324
Murray Hill, NJ 07974
USA

Phone: +1-908-582-7974
Email: wendy.roome@nokia.com

Michael Scharf
Nokia
Germany

Email: michael.scharf@nokia.com

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

Email: yry@cs.yale.edu

Jingxuan Jensen Zhang
Tongji University
4800 Caoan Road
Shanghai 201804
China

Email: jingxuan.n.zhang@gmail.com