

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: January 7, 2016

P. Wydrych  
AGH University  
Q. Fu  
China Mobile  
July 6, 2015

Paths Extension for the ALTO Protocol  
draft-wydrych-alto-paths-00

Abstract

The Application-Layer Traffic Optimization (ALTO) protocol has been standardized in RFC7285 to ease better-than-random overlay connection management. However, through the base protocol it is not possible to differentiate paths from a given source to a given destination and provide an ALTO client with a detailed information of sending traffic through these paths.

This document describes an extension to the ALTO protocol allowing for representation of paths in the network maps, cost maps, and endpoint cost calculation. Moreover, this document defines a new path computation service.

Status of This Memo

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

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

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

This Internet-Draft will expire on January 7, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Use-cases . . . . .	4
1.1.1. MPLS and RSVP-TE . . . . .	4
1.1.2. Locator/ID Separation . . . . .	4
1.1.3. Content Delivery Networks . . . . .	5
1.1.4. Service Function Chaining . . . . .	5
1.2. Requirements Language . . . . .	6
2. Extension Specification . . . . .	6
2.1. Information Resource Directory . . . . .	6
2.1.1. Example . . . . .	6
2.2. Provider-Defined Path Identifier (PPID) . . . . .	9
2.2.1. PPID Nesting . . . . .	9
2.3. Map Service: Network Map . . . . .	9
2.3.1. Example . . . . .	9
2.4. Map Service: Cost Map . . . . .	10
2.4.1. Example . . . . .	11
2.5. Map-Filtering Service: Filtered Network Map . . . . .	12
2.5.1. Example . . . . .	12
2.6. Map-Filtering Service: Filtered Cost Map . . . . .	13
2.6.1. Example . . . . .	14
2.7. Endpoint Cost Service . . . . .	14
2.7.1. Example . . . . .	15
2.8. Path Computation Service . . . . .	16
2.8.1. Example . . . . .	17
3. IANA Considerations . . . . .	18
4. Security Considerations . . . . .	18
5. Compatibility Considerations . . . . .	18
6. References . . . . .	18
6.1. Normative References . . . . .	18
6.2. Informative References . . . . .	19
Appendix A. Acknowledgments . . . . .	19
Authors' Addresses . . . . .	19

## 1. Introduction

As stated in [RFC5693], information on network topologies and routing policies in today Internet are not generally available to the application layer. At the same time, a lot of new overlay

applications creating their own topologies on top of the physical one emerge. As both network operators and users of the overlay applications may benefit from better-than-random overlay connection management, the ALTO protocol has been standardized in [RFC7285].

In the current networks, a number of unique paths from a given source to a given destination often exist. Even for the same pair of source and destination, the routing cost may be significantly different for each of the paths. Especially in case of transit traffic, cost may depend on ingress and egress link policies. Other factors that may influence the routing cost are, e.g., load of the links traversed throughout the network or capabilities of used routers.

Moreover, a number of tunneling techniques are used currently to handle parts of the traffic separately. To ensure that services of high quality are provided to customers, Internet Service Providers (ISPs) set up tunnels using specified paths. Then, network devices are configured to take into account the tunnel parameters and, e.g., prioritize traffic sent through a tunnel by using separate queues on routers.

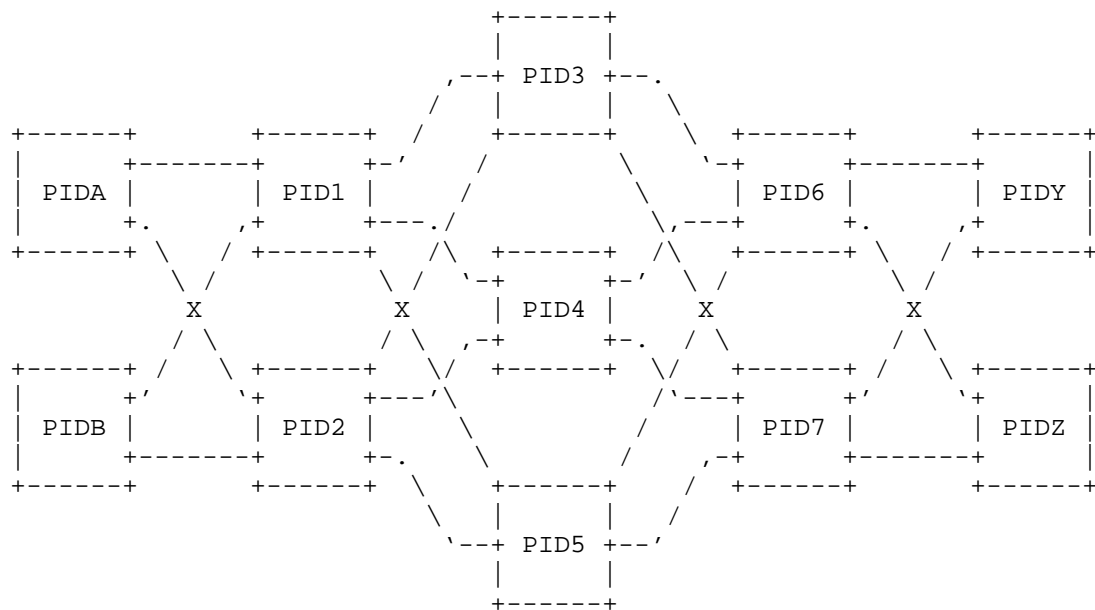


Figure 1: A redundant network topology.

In Figure 1, a redundant network topology has been presented. Due to the abovementioned observations, the cost of sending data from PIDA via PID1, PID4, and PID7 to PIDZ (using a known path) does not have

to be equal to the sum of costs of sending independent flows from PIDA to PID1, from PID1 to PID4, from PID4 to PID7, and from PID7 to PIDZ in a best-effort way.

Therefore, information on the cost of data transfer from a source to a destination provided by the network and cost maps using a base ALTO Protocol may be not accurate.

### 1.1. Use-cases

There are a number of situations in which would be optimal if a network device selected a path to a destination based not only on the information from the routing information databases. In this document, the following are shortly described:

- o establishments tunnels in MPLS networks,
- o providing optimal EID-to-RLOC mapping in LISP sites,
- o storing content replicas in CDNs,
- o forwarding data between virtualized network functions.

#### 1.1.1. MPLS and RSVP-TE

Currently, a lot of core networks use Multi-Protocol Label Switching (MPLS) [RFC3031] to forward IP packets and Resource Reservation Protocol (RSVP) to establish tunnels [RFC3209] between ingress and egress MPLS nodes. Due to the link redundancy, there may be a number of paths that packet may flow through for each ingress/egress nodes pair. The ALTO Protocol may be used during the optimal path establishment process, taking into account various factors like link policies or device and link load.

For instance, PIDA and PIDB (in Figure 1) may aggregate the traffic sources, while PIDY and PIDZ - the traffic destinations. If a MPLS network is established between nodes in PID1 to PID9, PID1 and PID2 are ingress MPLS nodes, and PID6 and PID7 are the egress MPLS nodes, the ALTO Protocol with a paths extension may be used by the control plane to assist tunnel establishment from the ingress to the egress nodes. Then, ISP may take into account source and destination of the packets to direct the traffic into a specific tunnel.

#### 1.1.2. Locator/ID Separation

The recently standardized Locator/ID Separation Protocol (LISP) [RFC6830] separates the IP addresses into two numbering spaces: Endpoint Identifiers (EIDs) and Routing Locators (RLOCs). Due to the

multihoming, one EID may be possible to be mapped to more than one RLOC. The ALTO Protocol may be used by the mapping service and/or by the ingress nodes select the optimal EID-to-RLOC mapping and, thus, the path used by the encapsulated packets.

For instance, PIDA and PIDB (in Figure 1) may aggregate the traffic sources, while PIDY and PIDZ - the traffic destinations. If a LISP network is established between nodes in PID1 to PID9, PID1 and PID2 are Ingress Tunnel Routers, and PID6 and PID7 are the Egress Tunnel Routers, the ALTO Protocol with a paths extension may be used by the EID-to-RLOC mapping service to assist LISP tunnel establishment from the ingress to the egress nodes.

#### 1.1.3. Content Delivery Networks

The content-delivery networks (CDNs) use replica servers to take benefit of caching in providing high quality service to the end-users [CDNs]. One of the crucial parts of each CDN is the algorithm deciding which data should be cached on which replica server and for how long. Especially, when an end-user requests for a relatively new content that is available only at the origin servers, it may be cached on the replica servers while it is being delivered to the end-user.

For instance, PIDA and PIDB (in Figure 1) may aggregate the origin servers, i.e., the authoritative content sources, while PIDY and PIDZ - the end-users' devices. If a CDN network design is tiered, PID1 and PID2 may aggregate data centers (DCs) with global replica servers, PID3 to PID5 - country-wide caches, and PID6 and PID7 - regional ones. The ALTO protocol with a paths extension may assist the CDN controller in selecting through which DCs traffic from the content source to the end-users should be forwarded to both provide high-quality service and store content replicas in correct caches at the same time.

#### 1.1.4. Service Function Chaining

As described in [I-D.fu-alto-nfv-usecase], the ALTO protocol can be used in the service function chaining (SFC) scenario, in which the SFC control plane may act as an ALTO client and ask ALTO server to provide a cost of a service function path (SFP). SFP is a sequence of network functions which the traffic flow should travel through. Utilizing the ALTO protocol, the SFC control plane can get the cost of each different paths and make the decision of which path to choose. In such a scenario, an extension of path is needed to define the SFP.

For instance, PIDA and PIDB (in Figure 1) may aggregate the service consumers, while PIDY and PIDZ are the service providers. If there is a requirement that the traffic from the service consumer has to be processed by a sequence of service functions, e.g., a firewall (PID1 or PID2), a deep packet inspector (PID3, PID4, or PID5), and a load-balancer (PID6 or PID7) in the correct order before reaching a service provider, the ALTO protocol with a paths extension may be used to optimize forwarding of the data between the service functions.

## 1.2. Requirements Language

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

## 2. Extension Specification

This document specifies an extension to the ALTO Protocol, as defined in RFC 7285, by adding ability to provide path-specific information through map and endpoint cost services. Moreover, this extension defines a new service called "path computation service".

An ALTO server that is compliant with the paths extension MUST implement at least one of the services:

- o paths-enhanced map service (both path-enabled network map and path-enabled cost map MUST be provided),
- o paths-enhanced endpoint cost service,
- o path computation service.

### 2.1. Information Resource Directory

This extension defines a new Boolean ALTO server capability: "paths". The default value of the "paths" capability is false. Each resource that provides information using syntax defined in this document MUST be assigned "paths": true capability. For clarity, an ALTO server implementing this extension MAY explicitly assign "path": false capability to the resources that are not providing any path information and are not using the syntax defined in this document.

#### 2.1.1. Example

```
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: 2477  
Content-Type: application/alto-directory+json

```
{
  "meta": {
    "cost-types": {
      "num-routing": {
        "cost-mode": "numerical",
        "cost-metric": "routingcost"
      },
      "ord-routing": {
        "cost-mode": "ordinal",
        "cost-metric": "routingcost"
      }
    },
    "default-alto-network-map": "network-map"
  },
  "resources": {
    "my-default-network-map": {
      "uri": "http://alto.example.com/networkmap",
      "media-type": "application/alto-networkmap+json"
    },
    "my-default-paths-network-map": {
      "uri": "http://alto.example.com/paths/networkmap",
      "media-type": "application/alto-networkmap+json",
      "capabilities": {
        "paths": true
      }
    },
    "numerical-routing-cost-map": {
      "uri": "http://alto.example.com/costmap/num/routingcost",
      "media-type": "application/alto-costmap+json",
      "capabilities": {
        "cost-type-names": [
          "num-routing"
        ]
      },
      "uses": [
        "my-default-network-map"
      ]
    },
    "paths-numerical-routing-cost-map": {
      "uri": "http://alto.example.com/paths/costmap/num/routingcost",
      "media-type": "application/alto-costmap+json",
      "capabilities": {
        "paths": true,
        "cost-type-names": [
```

```
        "num-routing"
      ]
    },
    "uses": [
      "my-default-network-map"
    ]
  },
  "endpoint-cost": {
    "uri": "http://alto.example.com/endpointcost/lookup",
    "media-type": "application/alto-endpointcost+json",
    "accepts": "application/alto-endpointcostparams+json",
    "capabilities": {
      "cost-constraints": true,
      "cost-type-names": [
        "num-routing",
        "ord-routing"
      ]
    }
  },
  "paths-endpoint-cost": {
    "uri": "http://alto.example.com/paths/endpointcost/lookup",
    "media-type": "application/alto-endpointcost+json",
    "accepts": "application/alto-endpointcostparams+json",
    "capabilities": {
      "paths": true,
      "cost-constraints": true,
      "cost-type-names": [
        "num-routing",
        "ord-routing"
      ]
    }
  },
  "path-computation": {
    "uri": "http://alto.example.com/paths/compute",
    "media-type": "application/alto-pathcompute+json",
    "accepts": "application/alto-pathcomputeparams+json",
    "capabilities": {
      "paths": true,
      "cost-constraints": true,
      "cost-type-names": [
        "num-routing",
        "ord-routing"
      ]
    }
  }
}
```



## 2.2. Provider-Defined Path Identifier (PPID)

This extension introduces Provider-defined Path Identifiers (PPIDs) to provide a way to specify a sequence of endpoints traversed by the traffic. A PPID is a string of type PPIDName and its associated list of PIDs. The list of PIDs associated to a PPID MUST NOT be empty. The list of PIDs MUST NOT contain an undefined PID.

A PPID Name MUST conform to all PID Name requirements specified by Section 10.1 of RFC 7285. A PPID Name MUST begin with 'path.', i.e., the word 'path' encoded in lower-case US-ASCII followed by the dot separator ('.', U+002E). A PPID Name MUST contain at least one character after the dot separator.

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

### 2.2.1. PPID Nesting

The Provider-defined Path Identifiers may be nested in case one path contains another. For instance, in case PIDs "PID1", "PID3", and "PID6" are defined, and a PPID "path.1-3" is defined as ["PID1", "PID3"], a PPID "path.1-3-6" consisting of "PID1", "PID3", and "PID6" may be defined both as ["PID1", "PID3", "PID6"] or ["path.1-3", "PID6"].

Nested PPIDs MUST NOT create circular dependencies. I.e., "path.A" MUST NOT contain "path.B" if "path.B" contains (directly or indirectly) "path.A".

## 2.3. Map Service: Network Map

Through this extension, a set of PPIDs is added to the network map. A network map MUST define all PIDs that PPIDs being defined comprise of. A network map SHOULD define all needed PIDs before defining a PPID. A network map SHOULD define a nested path before defining an outer one.

### 2.3.1. Example

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

```
HTTP/1.1 200 OK
Content-Length: 988
Content-Type: application/alto-networkmap+json
```

```
{
  "meta": {
    "vtag": {
      "resource-id": "paths-network-map",
      "tag": "e65e696925e7cc350b562b6a7d5f2540"
    }
  },
  "network-map": {
    "PIDA": { "ipv4": [ "192.0.2.0/25" ] },
    "PIDB": { "ipv4": [ "192.0.2.128/25" ] },
    "PID1": { "ipv4": [ "198.51.100.1/32" ] },
    "PID2": { "ipv4": [ "198.51.100.2/32" ] },
    "PID3": { "ipv4": [ "198.51.100.3/32" ] },
    "PID4": { "ipv4": [ "198.51.100.4/32" ] },
    "PID5": { "ipv4": [ "198.51.100.5/32" ] },
    "PID6": { "ipv4": [ "198.51.100.6/32" ] },
    "PID7": { "ipv4": [ "198.51.100.7/32" ] },
    "PIDY": { "ipv4": [ "203.0.113.0/25" ] },
    "PIDZ": { "ipv4": [ "203.0.113.128/25" ] },
    "path.1-3-6": [ "PID1", "PID3", "PID6" ],
    "path.1-4-7": [ "PID1", "PID4", "PID7" ],
    "path.2-5-7": [ "PID2", "PID5", "PID6" ],
    "path.1-3-6-Y": [ "path.1-3-6", "PIDY" ],
    "path.1-3-6-Z": [ "path.1-3-6", "PIDZ" ],
    "path.1-4-7-Z": [ "path.1-4-7", "PIDZ" ],
    "path.2-5-7-Z": [ "path.2-5-7", "PIDZ" ]
  }
}
```

#### 2.4. Map Service: Cost Map

Through a cost map, an ALTO server implementing this extension lists the path costs from sources to destinations via paths defined in the network map.

For each source/destination pair defined by a cost map, where a destination is a PPID, the cost value corresponds to the traffic originated in the source, traversing all-but-last PIDs of the path, and directed to the endpoint belonging to the last PID in the path.

A cost map MUST NOT define costs source/destination pair where source is a PPID. In other words, a PPIDName MUST NOT be a key of a CostMapData dictionary map object.

#### 2.4.1. Example

```
GET /paths/costmap/num/routingcost HTTP/1.1
Host: alto.example.com
Accept: application/alto-costmap+json,application/alto-error+json
```

```
HTTP/1.1 200 OK
Content-Length: 572
Content-Type: application/alto-costmap+json
```

```
{
  "meta": {
    "dependent-vtags": [
      {
        "resource-id": "paths-network-map",
        "tag": "e65e696925e7cc350b562b6a7d5f2540"
      }
    ],
    "cost-type": {
      "cost-mode": "numerical",
      "cost-metric": "routingcost"
    }
  },
  "cost-map": {
    "PIDA": { "PIDY": 50, "PIDZ": 100,
      "path.1-3-6-Y": 10, "path.1-3-6-Z": 15,
      "path.1-4-7-Z": 10, "path.2-5-7-Z": 20 },
    "PIDB": { "PIDY": 75, "PIDZ": 125,
      "path.1-3-6-Y": 20, "path.1-3-6-Z": 30,
      "path.1-4-7-Z": 25, "path.2-5-7-Z": 20 }
  }
}
```

In the above example, the routing cost of sending data from 192.0.2.0/25 to 203.0.113.128/25 using a best-effort service is 125, while the routing cost of sending data from 192.0.2.0/25 via 198.51.100.1, 198.51.100.4, and 198.51.100.7 to 203.0.113.128/25 using a dedicated path (e.g., a tunnel) is 25.

## 2.5. Map-Filtering Service: Filtered Network Map

An ALTO client requesting for network map filtering may specify both PIDs and PPIDs in the "pids" field of the request. Through this extension, PIDs and PPIDs may be requested implicitly. The behavior is different for PIDs and PPIDs requested.

If a PID name was specified, an ALTO server MUST return the requested PID as well as:

- o all PPIDs that have the specified PID at the last entry (i.e., PPIDs for all paths ending in the specified PID);
- o all nested PPID, recursively;
- o all PIDs that are needed to define implicitly requested paths.

If a PPID name was specified, an ALTO server MUST return the requested PPID as well as:

- o all nested PPID, recursively;
- o all PIDs that are needed to define explicitly and implicitly requested paths.

If the list of PIDs is empty, the ALTO server MUST interpret the list as if it contained a list of all currently defined PIDs and PPIDs.

### 2.5.1. Example

```
POST /paths/networkmap/filtered HTTP/1.1
Host: alto.example.com
Content-Length: 33
Content-Type: application/alto-networkmapfilter+json
Accept: application/alto-networkmap+json,application/alto-error+json
```

```
{
  "pids": [ "PIDA", "PIDY" ]
}
```

```
HTTP/1.1 200 OK
Content-Length: 477
Content-Type: application/alto-networkmap+json
```

```
{
  "meta": {
    "vtag": {
      "resource-id": "paths-network-map",
      "tag": "e65e696925e7cc350b562b6a7d5f2540"
    }
  },
  "network-map": {
    "PIDA": { "ipv4": [ "192.0.2.0/25" ] },
    "PID1": { "ipv4": [ "198.51.100.1/32" ] },
    "PID3": { "ipv4": [ "198.51.100.3/32" ] },
    "PID6": { "ipv4": [ "198.51.100.6/32" ] },
    "PIDY": { "ipv4": [ "203.0.113.0/25" ] },
    "path.1-3-6": [ "PID1", "PID3", "PID6" ],
    "path.1-3-6-Y": [ "path.1-3-6", "PIDY" ],
  }
}
```

## 2.6. Map-Filtering Service: Filtered Cost Map

An ALTO client requesting for cost map filtering may specify both PIDs and PPIDs in the "pids"."dsts" field of the request. Through this extension, PPIDs may be requested implicitly. If a PID name was specified as a destination, an ALTO server MUST return the cost map for all source/destination pairs in which the requested PID is a destination as well as all source/destination pairs in which a PPID that have the specified PID at the last entry is a destination.

If the list of destination PIDs is empty, the ALTO server MUST interpret the list as if it contained a list of all currently defined PIDs and PPIDs.

The source PID list MUST NOT contain any PPIDs.

### 2.6.1. Example

```
POST /paths/costmap/filtered HTTP/1.1
Host: alto.example.com
Content-Type: application/alto-costmapfilter+json
Content-Length: 145
Accept: application/alto-costmap+json,application/alto-error+json
```

```
{
  "cost-type": {
    "cost-mode": "numerical",
    "cost-metric": "routingcost"
  },
  "pids": {
    "srcs": [ ],
    "dsts": [ "PIDY" ]
  }
}
```

```
HTTP/1.1 200 OK
Content-Length: 370
Content-Type: application/alto-costmap+json
```

```
{
  "meta": {
    "dependent-vtags": [
      {
        "resource-id": "paths-network-map",
        "tag": "e65e696925e7cc350b562b6a7d5f2540"
      }
    ],
    "cost-type": {
      "cost-mode": "numerical",
      "cost-metric": "routingcost"
    }
  },
  "cost-map": {
    "PIDA": { "PIDY": 50, "path.1-3-6-Y": 10 },
    "PIDB": { "PIDY": 75, "path.1-3-6-Y": 20 }
  }
}
```

### 2.7. Endpoint Cost Service

An ALTO client requesting for cost map filtering may specify both destinations and paths in the "endpoints"."dsts" field of the request.

A cost for a path is requested by specifying an array of typed endpoint addresses as a destination entry.

For each source endpoint, the "endpoint-cost-map" field of the response contains a tree denoting costs for requested paths. The subsequent endpoints a path comprise of are represented as internal nodes of a cost tree.

#### 2.7.1. Example

```
POST /paths/endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: 468
Content-Type: application/alto-endpointcostparams+json
Accept: application/alto-endpointcost+json,application/alto-error+json
```

```
{
  "cost-type": {
    "cost-mode": "ordinal",
    "cost-metric": "routingcost"
  },
  "endpoints": {
    "srcs": [ "ipv4:192.0.2.2" ],
    "dsts": [
      "ipv4:203.0.113.128.129",
      [
        "ipv4:198.51.100.1",
        "ipv4:198.51.100.3",
        "ipv4:198.51.100.6",
        "ipv4:203.0.113.128.129"
      ],
      [
        "ipv4:198.51.100.1",
        "ipv4:198.51.100.4",
        "ipv4:198.51.100.7",
        "ipv4:203.0.113.128.129"
      ]
    ]
  }
}
```

```
HTTP/1.1 200 OK
Content-Length: 495
Content-Type: application/alto-endpointcost+json
```

```
{
  "meta": {
    "cost-type": {
```

```

        "cost-mode": "ordinal",
        "cost-metric": "routingcost"
    },
    "endpoint-cost-map": {
        "ipv4:192.0.2.2": {
            "ipv4:203.0.113.128.129": 3,
            "ipv4:198.51.100.1": {
                "ipv4:198.51.100.3": {
                    "ipv4:198.51.100.6": {
                        "ipv4:203.0.113.128.129": 1
                    }
                }
            },
            "ipv4:198.51.100.4": {
                "ipv4:198.51.100.7": {
                    "ipv4:203.0.113.128.129": 2
                }
            }
        }
    }
}

```

## 2.8. Path Computation Service

This document defines a new service called path computation service. This service provides information on best paths composed of the endpoints specified by a client.

The path computation resource is requested using the HTTP POST method. The media type of the path computation resource is "application/alto-pathcompute+json". An ALTO client supplies the path computation parameters through a media type "application/alto-pathcompute+params+json", with an HTTP POST entity body of a JSON object with fields:

**cost-type:** The cost type (Section 10.7 of <RFC7285>) to use for returned costs. The "cost-metric" and "cost-mode" fields MUST match one of the supported cost types indicated in this resource's "capabilities" fields. The ALTO client SHOULD omit the "description" field, and if present, the ALTO server MUST ignore the "description" field.

**endpoints:** A list of lists of endpoints from which paths may be composed. The ALTO server MUST compose a path taking exactly one element from each list, preserving the order.



The response comprises a list of suggested paths. Each path is an object containing a list of endpoints forming the path and the path's cost. The server MAY return more than one path. The server MAY return no paths if it cannot compute an optimal one.

#### 2.8.1. Example

```
POST /paths/compute HTTP/1.1
Host: alto.example.com
Content-Length: 296
Content-Type: application/alto-pathcomputeparams+json
Accept: application/alto-pathcompute+json,application/alto-error+json
```

```
{
  "cost-type": {
    "cost-mode": "ordinal",
    "cost-metric": "routingcost"
  },
  "endpoints": [
    [ "ipv4:192.0.2.2" ],
    [ "ipv4:198.51.100.1" ],
    [ "ipv4:198.51.100.3", "ipv4:198.51.100.4" ],
    [ "ipv4:198.51.100.6", "ipv4:198.51.100.7" ],
    [ "ipv4:203.0.113.128.129" ]
  ]
}
```

```
HTTP/1.1 200 OK
Content-Length: 536
Content-Type: application/alto-pathcompute+json
```

```
{
  "meta": {
    "cost-type": {
      "cost-mode": "ordinal",
      "cost-metric": "routingcost"
    }
  },
  "computed-paths": [
    {
      "path": [
        "ipv4:192.0.2.2",
        "ipv4:198.51.100.1",
        "ipv4:198.51.100.3",
        "ipv4:198.51.100.6",
        "ipv4:203.0.113.128.129"
      ],
      "cost": 1
    }
  ]
}
```

```
    },
    {
      "path": [
        "ipv4:192.0.2.2",
        "ipv4:198.51.100.1",
        "ipv4:198.51.100.4",
        "ipv4:198.51.100.7",
        "ipv4:203.0.113.128.129"
      ],
      "cost": 2
    }
  ]
}
```

### 3. IANA Considerations

This document registers two application/alto-\* media types: application/alto-pathcomputeparams+json and application/alto-pathcompute+json.

### 4. Security Considerations

This document does not introduce any privacy or security issues not already present in the ALTO protocol.

### 5. Compatibility Considerations

The extension defined in this document is compatible with the multi-cost extension [I-D.ietf-alto-multi-cost]. Whenever a cost value is considered in the server response, an array of multiple costs may be used if a server implements both paths and multi-cost extensions.

The extension defined in this document is compatible with the incremental updates using server-sent events [I-D.ietf-alto-incr-update-sse].

### 6. References

#### 6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC7285] Alimi, R., Penno, R., Yang, Y., Kiesel, S., Previdi, S., Roome, W., Shalunov, S., and R. Woundy, "Application-Layer Traffic Optimization (ALTO) Protocol", RFC 7285, September 2014.

## 6.2. Informative References

- [CDNs]      Pathan, A. and R. Buyya, "A Taxonomy and Survey of Content Delivery Networks", 2007,  
            <<http://www.cloudbus.org/reports/CDN-Taxonomy.pdf>>.
- [I-D.fu-alto-nfv-usecase]  
            Fu, Q., Cao, Z., and H. Song, "What's the Impact of Virtualization on Application-Layer Traffic Optimization (ALTO)?", draft-fu-alto-nfv-usecase-05 (work in progress), June 2015.
- [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-00 (work in progress), May 2015.
- [I-D.ietf-alto-multi-cost]  
            Randriamasy, S., Roome, W., and N. Schwan, "Multi-Cost ALTO", draft-ietf-alto-multi-cost-00 (work in progress), May 2015.
- [RFC3031]    Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.
- [RFC3209]    Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.
- [RFC5693]    Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, October 2009.
- [RFC6830]    Farinacci, D., Fuller, V., Meyer, D., and D. Lewis, "The Locator/ID Separation Protocol (LISP)", RFC 6830, January 2013.

## Appendix A. Acknowledgments

P. Wydrych's work on this draft were performed under contract 15.11.230.190.

## Authors' Addresses

Piotr Wydrych  
AGH University of Science and Technology  
Mickiewicza 30  
Krakow 30-059  
Poland

Phone: +48 12 617 48 17  
Email: [piotr.wydrych@agh.edu.pl](mailto:piotr.wydrych@agh.edu.pl)  
URI: <http://wydrych.net/>

Fu Qiao  
China Mobile  
Xuanwumenxi Ave. No.32  
Beijing  
China

Email: [fuqiao1@outlook.com](mailto:fuqiao1@outlook.com)