Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: April 25, 2013

V. Gurbani, Ed.
W. Roome
Bell Laboratories, Alcatel-Lucent
R. Varga
Cisco Systems, Inc.
N. Zhang
Neustar

October 22, 2012

Interoperability testing of the Application- Layer Traffic Optimization (ALTO) Protocol

draft-gurbani-alto-interop-cases-02

Abstract

The Application-Layer Traffic Optimization (ALTO) protocol is designed to allow entities with knowledge about the network infrastructure to export such information to applications that need to choose one or more endpoints to connect to among large sets of logically equivalent ones. This document provides a collection of messages that may be used to test the functionality and interoperability of an ALTO client and an ALTO server.

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 April 25, 2013.

Copyright Notice

Table of Contents

1. Overview

The Application-Layer Traffic Optimization (ALTO) protocol is designed to allow entities with knowledge about the network infrastructure to export such information to applications that need to choose one or more endpoints to connect to among large sets of logically equivalent ones.

This document contains a set of messages that may be used test the functionality and interoperability of an ALTO client and an ALTO server.

This document is informational and is NOT NORMATIVE on any aspects of the ALTO protocol. The normative behaviour of ALTO entities is prescribed in [I-D.ietf-alto-protocol].

The test messages are organized into several sections. The key ALTO services, including but not limited to, Information Resource Directory retrieval, retrieving network map and cost map, endpoint cost service, filtered network map service, filtered cost map service are provided as discrete test cases. Also included are errors associating with the ALTO request and the JSON payload.

While every effort has been made to catalogue representative test cases, this document does not attempt to codify every test case that arises in ALTO. The aim of the document is to focus on areas that highlight the key offerings of the ALTO protocol.

2. Maps: Network map and cost map

To uniformly interpret the contents of the ALTO messages, a sample topology is presented below. This topology is divided into a network map and a cost map. The network map contains a series of PIDs, a provider-defined network location identifier as specified in [I-D.ietf-alto-protocol].

PID	IP Address Block
mypid1	10.0.0.0/8, 15.0.0.0/8
mypid2	192.168.0.0/16
mypid3	192.168.10.0/24
peeringpid1	128.0.0.0/16
peeringpid2	130.0.0.0/16, 2001:DB8::/32
transitpid1	132.0.0.0/16
transitpid2	135.0.0.0/16
defaultpid	0.0.0.0/0, ::/0

Figure 1: Sample Network Map

A cost map corresponding to the above network map is shown below. The cost map defines path costs amongst sets of source and destination network locations. Each path cost is the end-to-end cost from the source to the destination.

Source PID	Destination PID	Cost Mode Numerical	Ordinal
mypid1	mypid1	0	1
"	mypid2	0	1
"	mypid3	0	1
"	peeringpid1	0	1
"	peeringpid2	0	1
"	transitpid1	5	3
"	transitpid2	10	7
"	defaultpid	4	2
mypid2	mypid1	0	1
"	mypid2	0	1
"	mypid3	0	1
"	peeringpid1	0	1
"	peeringpid2	0	1
"	transitpid1	7	5
"	transitpid2	8	6
"	defaultpid	4	2
mypid3	mypid1	0	1
"	mypid2	0	1
"	mypid3	0	1
"	peeringpid1	0	1
"	peeringpid2	0	1
"	transitpid1	8	6
"	transitpid2	8	6
"	defaultpid	5.1	4

Figure 2: Corresponding Cost Map

Note that the above represents a sparse cost map, i.e., the ALTO server is not defining a path cost from each source PID to each destination PID. It is only defining the costs that it is interested in serving.

3. Test cases

Exhaustive test cases are described in subsections below. Note that it is not expected that an ALTO client and server implementation generate requests and responses in the same format as shown below.

The syntax related to representing each request or response is left to each individual implementation as long as the payload is syntactically valid and semantically equivalent to any other representation of the same payload.

In the test cases below, an ALTO server is assumed to be available at the URL http://alto.ietf.org. Requests are directed towards it. This document assumes that the ALTO server URL has been discovered by the ALTO discovery protocol, however, it does not provide further details on the discovery protocol itself. Wherever possible, relevant HTTP headers are shown in the test cases, however, for the sake of brevity not all headers are depicted.

3.1. Information Resource Directory

Test-IRD-1: The Information Resource Directory (IRD) enumerates URIs at which an ALTO server offers Information Resources.

```
Client -> Server:
-----------------
GET /directory HTTP/1.1
Host: alto.ietf.org
Accept: application/alto-directory+json,
        application/alto-error+json

Server -> Client:
-----------------
HTTP/1.1 200 OK
Content-Length: 2596
Connection: close
Content-Type: application/alto-directory+json
Date: Wed, 29 Jun 2011 10:52:09 GMT

{
"resources": [
   {
   "capabilities": {
      "cost-modes": [
         "numerical"
      ],
      "cost-types": [
         "routingcost"
      ]
   },
   "media-types": [
      "application/alto-costmap+json"
   ],
```

```
            "uri": "http://alto.ietf.org/costmap/numerical/routingcost"
         },
         {
            "capabilities": {
               "cost-modes": [
                  "ordinal"
               ],
               "cost-types": [
                  "routingcost"
               ]
            },
            "media-types": [
               "application/alto-costmap+json"
            ],
            "uri": "http://alto.ietf.org/costmap/ordinal/routingcost"
         },
         {
            "media-types": [
               "application/alto-networkmap+json"
            ],
            "uri": "http://alto.ietf.org/networkmap"
         },
         {
            "media-types": [
               "application/alto-serverinfo+json"
            ],
            "uri": "http://alto.ietf.org/serverinfo"
         },
         {
            "accepts": [
               "application/alto-costmapfilter+json"
            ],
            "capabilities": {
               "cost-constraints": true,
               "cost-modes": [
                  "numerical",
                  "ordinal"
               ],
               "cost-types": [
                  "routingcost"
               ]
            },
            "media-types": [
               "application/alto-costmap+json"
            ],
            "uri": "http://alto.ietf.org/costmap/filtered"
         },
         {
```

```
            "accepts": [
               "application/alto-endpointcostparams+json"
            ],
            "capabilities": {
               "cost-constraints": true,
               "cost-modes": [
                  "numerical",
                  "ordinal"
               ],
               "cost-types": [
                  "routingcost"
               ]
            },
            "media-types": [
               "application/alto-endpointcost+json"
            ],
            "uri": "http://alto.ietf.org/endpoints/cost"
         },
         {
            "accepts": [
               "application/alto-endpointpropparams+json"
            ],
            "capabilities": {
               "prop-types": [
                  "pid"
               ]
            },
            "media-types": [
               "application/alto-endpointprop+json"
            ],
            "uri": "http://alto.ietf.org/endpoints/property"
         },
         {
            "accepts": [
               "application/alto-networkmapfilter+json"
            ],
            "media-types": [
               "application/alto-networkmap+json"
            ],
            "uri": "http://alto.ietf.org/networkmap/filtered"
         }
      ]
      }
```

3.2. Endpoint property service

Test-EPS-1: An ALTO client retrieves a PID for IPv4 address 192.168.1.23.

```
Client -> Server:
-----------------
POST /endpoints/property HTTP/1.1
Host: alto.ietf.org
Content-Length: ...
Content-Type: application/alto-endpointpropparams+json
Accept: application/alto-endpointprop+json

{
   "properties" : [ "pid" ],
   "endpoints" : [ "ipv4:192.168.1.23" ]
}
```

The server returns the following response (note that the longest prefix match is used to retrieve the corresponding PID property).

```
Server -> Client:
-----------------
HTTP/1.1 200 OK
Content-Length: ...
Content-Type: application/alto-endpointprop+json

{
     "meta" : {},
     "data": {
       "map-vtag" : "1266506139",
       "map" : {
         "ipv4:192.168.1.23" : { "pid": "mypid2" }
       }
     }
}
```

Test-EPS-2: An ALTO client retrieves a PID for IPv4 address 192.168.10.23.

```
        Client -> Server:
        -----------------
      POST /endpoints/property HTTP/1.1
      Host: alto.ietf.or
      Content-Length: ...
      Content-Type: application/alto-endpointpropparams+json
      Accept: application/alto-endpointprop+json

      {
         "properties" : [ "pid" ],
         "endpoints" : [ "ipv4:192.168.10.23" ]
      }
```

The server returns the following response (note that the longest prefix match is used to retrieve the corresponding PID property).

```
       Server -> Client:
       -----------------
       HTTP/1.1 200 OK
       Content-Length: ...
       Content-Type: application/alto-endpointprop+json

       {
            "meta" : {},
            "data": {
              "map-vtag" : "1266506139",
              "map" : {
                "ipv4:192.168.10.23"    : { "pid": "mypid3" }
              }
            }
       }
```

Test-EPS-3: An ALTO client retrieves a PID for IPv4 address 201.1.13.12.

```
       Client -> Server:
       -----------------
      POST /endpoints/property HTTP/1.1
      Host: alto.ietf.org
      Content-Length: ...
      Content-Type: application/alto-endpointpropparams+json
      Accept: application/alto-endpointprop+json

      {
         "properties" : [ "pid" ],
         "endpoints" : [ "ipv4:201.1.13.12" ]
      }
```

The server returns the following response (note that the longest prefix match is used to retrieve the corresponding PID property).

```
Server -> Client:
-----------------
HTTP/1.1 200 OK
Content-Length: ...
Content-Type: application/alto-endpointprop+json

{
     "meta" : {},
     "data": {
       "map-vtag" : "1266506139",
       "map" : {
         "ipv4:201.1.13.12"    : { "pid": "defaultpid" }
       }
}
```

Test-EPS-4: An ALTO client retrieves a PID for an IPv4 and IPv6 address.

```
Client -> Server:
-----------------
POST /endpoints/property HTTP/1.1
Host: alto.ietf.org
Content-Length: 106
Accept: application/alto-endpointprop+json
Content-Type: application/alto-endpointpropparams+json

{
  "properties" : [ "pid" ],
  "endpoints" : [ "ipv6:1234::192.168.1.23",
                  "ipv4:132.0.10.12" ]
}
```

The server returns the following response:

```
Server -> Client:
-----------------
HTTP/1.1 200 OK
Content-Type: application/alto-endpointprop+json;charset=UTF-8
Content-Length: 157

{
     "meta" : { } ,
     "data" : {
       "map-vtag" : "1266506139",
       "map" : {
         "ipv6:1234::192.168.1.23" : { "pid" : "defaultpid " } ,
         "ipv:132.0.10.12" : { "pid" : "transitpid1" }
       }
     }
}
```

3.3. Endpoint cost service

Test-ECS-1: An ALTO client requests cost information between individual endpoints.

```
Client -> Server:
-----------------
POST /endpoints/cost HTTP/1.1
Host: alto.example.com
Content-Length: 429
Content-Type: application/alto-endpointcostparams+json
Accept: application/alto-endpointcost+json,
        application/alto-error+json

{
 "cost-mode" : "numerical",
 "cost-type" : "routingcost",
 "endpoints" : {
   "srcs": [ "ipv4:10.0.0.0", "ipv4:192.168.11.0",
             "ipv4:192.168.10.0"],
   "dsts": [
     "ipv4:10.0.0.0",
     "ipv4:15.0.0.0",
     "ipv4:192.168.11.0",
     "ipv4:192.168.10.0",
     "ipv4:128.0.0.0",
     "ipv4:130.0.0.0",
     "ipv4:0.0.0.0",
     "ipv4:132.0.0.0",
     "ipv4:135.0.0.0"
   ]
 }
}
```

Server responds with the following:

```
Server -> Client:
-----------------
HTTP/1.1 200 OK
Content-Length: 1692
Content-Type: application/alto-endpointcost+json

{
   "meta": { },
   "data": {
          "cost-mode": "numerical",
          "cost-type": "routingcost",
          "map": {
                 "ipv4:10.0.0.0": {
                  "ipv4:10.0.0.0": 0.000000,
                  "ipv4:15.0.0.0": 0.000000,
                  "ipv4:192.168.11.0": 0.000000,
                  "ipv4:192.168.10.0": 0.000000,
                  "ipv4:128.0.0.0": 0.000000,
                  "ipv4:130.0.0.0": 0.000000,
                  "ipv4:0.0.0.0": 4.000000,
                  "ipv4:132.0.0.0": 5.000000,
                   "ipv4:135.0.0.0": 10.000000
               },
               "ipv4:192.168.11.0": {
                    "ipv4:10.0.0.0": 0.000000,
                    "ipv4:15.0.0.0": 0.000000,
                    "ipv4:192.168.11.0": 0.000000,
                    "ipv4:192.168.10.0": 0.000000,
                    "ipv4:128.0.0.0": 0.000000,
                    "ipv4:130.0.0.0": 0.000000,
                    "ipv4:0.0.0.0": 4.000000,
                    "ipv4:132.0.0.0": 7.000000,
                    "ipv4:135.0.0.0": 8.000000
               },
               "ipv4:192.168.10.0": {
                     "ipv4:10.0.0.0": 0.000000,
                     "ipv4:15.0.0.0": 0.000000,
                     "ipv4:192.168.11.0": 0.000000,
                     "ipv4:192.168.10.0": 0.000000,
                     "ipv4:128.0.0.0":0.000000,
                     "ipv4:130.0.0.0": 0.000000,
                     "ipv4:0.0.0.0": 5.100000,
                     "ipv4:132.0.0.0": 8.000000,
                     "ipv4:135.0.0.0": 8.000000
               }
          }
   }
}
```

Test-ECS-2: An ALTO client requests the ranking service for a source host to a set of destination hosts.

Client -> Server:

```
POST /endpoints/cost HTTP/1.1
Host: alto.ietf.org
Accept: application/alto-endpointcost+json,
        application/alto-error+json
Content-Type: application/alto-endpointcostparams+json
Content-Length: 235

{
  "cost-mode" : "ordinal",
  "cost-type" : "routingcost",
  "endpoints" : {
    "srcs": [ "ipv6:2001:DB8::ABCD:6789", "ipv4:192.168.10.1" ],
    "dsts": [
      "ipv6:2001:DB8::2345:5678",
      "ipv4:135.0.29.1",
      "ipv4:192.168.10.23"
    ]
  }
}
```

The server response is shown below. Note that the source IP adress of "ipv6:2001:DB8::ABCD:6789", which occurs in PID "peeringpid2", is omitted in the response. This reflects the fact that the ALTO server does not know the source costs from the "peeringpid2" PID.

```
Server -> Client:
-----------------
HTTP/1.1 200 OK
Content-Type: application/alto-endpointcost+json;charset=UTF-8
Content-Length: 376

{
   "data": {
    "cost-mode": "ordinal",
    "cost-type": "routingcost",
    "map": {
      "ipv4:192.168.10.1": {
        "ipv4:192.168.10.23": 1,
        "ipv6:2001:DB8::2345:5678": 1,
        "ipv4:135.0.29.1": 6
      }
    },
    "map-vtag": "gqcla2l8"
  },
  "meta": {}
}
```

Test-ECS-3: An ALTO client requests the cost service subject to certain constraints.

```
Client -> Server:
-----------------
POST /endpoints/cost HTTP/1.1
Host: alto.example.com
Content-Length: ...
Content-Type: application/alto-endpointcostparams+json
Accept: application/alto-endpointcost+json,
        application/alto-error+json

{
 "constraints": ["le 5", "ge 4"],
 "cost-mode": "numerical",
 "cost-type": "routingcost",
 "endpoints": {
   "dsts": [
     "ipv4:10.0.0.0",
     "ipv4:15.0.0.0",
     "ipv4:192.168.11.0",
     "ipv4:192.168.10.0",
     "ipv4:128.0.0.0",
     "ipv4:130.0.0.0",
     "ipv4:0.0.0.0",
     "ipv4:132.0.0.0",
     "ipv4:135.0.0.0"
   ],
   "srcs": [
     "ipv4:10.0.0.0",
     "ipv4:192.168.11.0",
     "ipv4:192.168.10.0"
   ]
 }
}
```

The server responds with the following:

```
Server -> Client
-----------------
HTTP/1.1 200 OK
Content-Length: 1692
Content-Type: application/alto-endpointcost+json
{
 "data": {
   "cost-mode": "numerical",
   "cost-type": "routingcost",
     "map": {
       "ipv4:10.0.0.0": {
         "ipv4:0.0.0.0": 4, "ipv4:132.0.0.0": 5
       },
       "ipv4:192.168.11.0": {"ipv4:0.0.0.0": 4}
     }
 },
 "meta": {}
}
```

3.4. Retrieving maps

Test-MAPS-1: An ALTO client retrieves a complete network map.

```
Client -> Server:
-----------------
GET /networkmap HTTP/1.1
Host: alto.ietf.org
Accept: application/alto-networkmap+json,
        application/alto-error+json
```

The server returns the following response.

```
Server -> Client:
-----------------
HTTP/1.1 200 OK
Content-Length: 799
Content-Type: application/alto-networkmap+json

{
 "meta" : {},
 "data" : {
   "map-vtag" : "1266506139",
   "map" : {
      "mypid1" : {
         "ipv4" : [ "10.0.0.0/8", "15.0.0.0/8" ]
      },
      "mypid2" : {
         "ipv4" : [ "192.168.0.0/16" ]
      },
      "mypid3" : {
         "ipv4" : [ "192.168.10.0/24" ]
      },
      "peeringpid1" : {
         "ipv4" : [ "128.0.0.0/16" ]
      },
      "peeringpid2" : {
         "ipv4" : [ "130.0.0.0/16" ],
         "ipv6" : [ "2001:DB8::/32"]
      },
      "transitpid1" : {
         "ipv4" : [ "132.0.0.0/16" ]
      },
      "transitpid2" : {
         "ipv4" : [ "135.0.0.0/16" ]
      },
      "defaultpid" : {
         "ipv4" : [ "0.0.0.0/0" ],
         "ipv6" : [ "::/0" ]
      }
   }
 }
}
```

Test-MAPS-2: An ALTO client retrieves a complete cost map for the numerical cost mode.

```
Client -> Server:
-----------------
GET /costmap/numerical/routingcost HTTP/1.1
Host: alto.ietf.org
Accept: application/alto-costmap+json,
        application/alto-error+json
```

The server returns the following response. In the response below, note that the version tag of the cost map ("map-vtag") corresponds to the network map of the same version shown in test case Test-MAPS-1.

```
Server -> Client:
-----------------
HTTP/1.1 200 OK
Content-Length: 787
Content-Type: application/alto-costmap+json

{
  "meta" : {},
  "data" : {
    "cost-mode" : "numerical",
    "cost-type" : "routingcost",
    "map-vtag"  : "1266506139",
    "map" : {
      "mypid1": { "mypid1" : 0,  "mypid2" : 0,  "mypid3" : 0,
                  "peeringpid1" : 0, "peerinpid2" :  0,
                  "transitpid1" : 5, "transitpid2" : 10,
                  "defaultpid" : 4},
      "mypid2": { "mypid1" : 0,  "mypid2" : 0,  "mypid3" : 0,
                  "peeringpid1" : 0, "peerinpid2" :  0,
                  "transitpid1" : 7, "transitpid2" : 8,
                  "defaultpid" : 4},
      "mypid3": { "mypid1" : 0,  "mypid2" : 0,  "mypid3" : 0,
                  "peeringpid1" : 0, "peerinpid2" :  0,
                  "transitpid1" : 8, "transitpid2" : 8,
                  "defaultpid" : 5.1}
    }
  }
}
```

Test-MAPS-3: An ALTO client retrieves a complete cost map for the ordinal cost mode.

```
Client -> Server:
-----------------
GET /costmap/ordinal/routingcost HTTP/1.1
Host: alto.ietf.org
Accept: application/alto-costmap+json,
        application/alto-error+json
```

The server returns the following response. In the response below, note that the version tag of the cost map ("map-vtag") corresponds to the network map of the same version shown in test case Test-MAPS-1.

```
Server -> Client:
-----------------
HTTP/1.1 200 OK
Content-Length: ...
Content-Type: application/alto-costmap+json

{
  "meta" : {},
  "data" : {
    "cost-mode" : "ordinal",
    "cost-type" : "routingcost",
    "map-vtag"  : "1266506139",
    "map" : {
      "mypid1": { "mypid1" : 1,  "mypid2" : 1,  "mypid3" : 1,
                  "peeringpid1" : 1, "peeringpid2" :  1,
                  "transitpid1" : 3, "transitpid2" : 7,
                  "defaultpid" : 2},
      "mypid2": { "mypid1" : 1,  "mypid2" : 1,  "mypid3" : 1,
                  "peeringpid1" : 1, "peeringpid2" :  1,
                  "transitpid1" : 5, "transitpid2" : 6,
                  "defaultpid" : 2},
      "mypid3": { "mypid1" : 1,  "mypid2" : 1,  "mypid3" : 1,
                  "peeringpid1" : 1, "peeringpid2" :  1,
                  "transitpid1" : 6, "transitpid2" : 6,
                  "defaultpid" : 4}
    }
  }
}
```

Test-MAPS-4: This test is designed to detect a change in the network map.

Add a new block of addresses to the network map of Figure 1, thereby updating the network map. For example, add the following new entry to the network map of Figure 1:

peeringpid2 130.0.0.0/16, 2001:DB8::/32, 201.1.2.0/24

The ALTO client retrieves the new network map using the GET request shown in Test-MAPS-1. The expectation is that the retrieved network map has the new PID (peeringpid2) included in the topology, and the version tag on the newly retrieved network map should be different than the response to the ALTO client shown in Test-MAPS-1.

The server should respond with a response that approximates what is shown below:

```
Server -> Client:
-----------------
HTTP/1.1 200 OK
Content-Length: 815
Content-Type: application/alto-networkmap+json

{
"meta" : {},
"data" : {
   "map-vtag" : "1266506155",
   "map" : {
      "mypid1" : {
         "ipv4" : [ "10.0.0.0/8", "15.0.0.0/8" ]
      },
      "mypid2" : {
         "ipv4" : [ "192.168.0.0/16" ]
      },
      "mypid3" : {
         "ipv4" : [ "192.168.10.0/24" ]
      },
      "peeringpid1" : {
         "ipv4" : [ "128.0.0.0/16" ]
      },
      "peeringpid2" : {
         "ipv4" : [ "130.0.0.0/16", "201.1.2.0/24" ],
         "ipv6" : [ "2001:DB8::/32"]
      },
      "transitpid1" : {
         "ipv4" : [ "132.0.0.0/16" ]
      },
      "transitpid2" : {
         "ipv4" : [ "135.0.0.0/16" ]
      },
      "defaultpid" : {
         "ipv4" : [ "0.0.0.0/0" ],
         "ipv6" : [ "::/0" ]
      }
   }
}
}
```

3.5. Filtering

Test-FILTER-1: An ALTO client sends a request to get a filtered map of PID mypid2.

```
Client -> Server:
-----------------
POST /networkmap/filtered HTTP/1.1
Host: alto.ietf.org
Content-Length: 26
Content-Type: application/alto-networkmapfilter+json
Accept: application/alto-networkmap+json,
        application/alto-error+json

{
  "pids": [ "mypid2" ]
}
```

The server responds with the following:

```
Server -> Client:
-----------------
HTTP/1.1 200 OK
Content-Length: 172
Content-Type: application/alto-networkmap+json

{
   "meta" : {},
   "data" : {
      "map-vtag" : "1266506155",
      "map" : {
         "mypid2" : {
            "ipv4" : [ "192.168.0.0/16" ]
         }
      }
   }
}
```

Test-FILTER-2: An ALTO client sends a request to get a filtered map from a source PID to a set of destination PIDs.

```
Client -> Server:
-----------------
POST /costmap/filtered HTTP/1.1
Host: alto.ietf.org
Content-Type: application/alto-costmapfilter+json
Content-Length: 174
Accept: application/alto-costmap+json,
        application/alto-error+json

{
  "cost-mode" : "numerical",
  "cost-type" : "routingcost",
  "pids" : {
    "srcs" : [ "mypid1", "mypid3" ],
    "dsts" : [ "mypid2", "peeringpid1", "transitpid2" ]
  }
}
```

The server responds with the following:

```
Server -> Client:
-----------------
HTTP/1.1 200 OK
Content-Length: 294
Content-Type: application/alto-costmap+json

{
  "meta" : {},
  "data" : {
    "cost-mode" : "numerical",
    "cost-type" : "routingcost",
    "map-vtag" : "1266506155",
    "map" : {
      "mypid1": { "mypid2": 0,  "peeringpid1": 0,
                  "transitpid2": 10 },
      "mypid3": { "mypid2": 0,  "peeringpid1": 0,
                  "transitpid2": 8 }
    }
  }
}
```

3.6. JSON parsing errors

Test-JSON-ERR-1: An ALTO client sends a malformed JSON body in the request --- a missing closing brace ('}').

```
Client -> Server:
-----------------
POST /endpoints/cost HTTP/1.1
Host: alto.ietf.org
Accept: application/alto-endpointcost+json,
        application/alto-error+json
Content-Type: application/alto-endpointcostparams+json
Content-Length: 131

{
  "cost-mode" : "numerical",
  "cost-type" : "routingcost",
  "endpoints": {
          "srcs": [ "ipv4:10.0.0.0"],
          "dsts": [ "ipv4:10.0.0.0" ]
}
```

The server returns an HTTP response code of 400 with ALTO error code of E_SYNTAX (c.f., Table 1 [I-D.ietf-alto-protocol]).

```
Server -> Client:
-----------------
HTTP/1.1 400 Bad Request
Content-Type: application/alto-error+json
Content-Length: ...

{
   "code": "E_SYNTAX"
}
```

Test-JSON-ERR-2: An ALTO client sends a malformed request --- the "dsts" member for the endpoint cost service is missing.

```
Client -> Server:
-----------------
POST /endpoints/cost HTTP/1.1
Host: alto.ietf.org
Accept: application/alto-endpointcost+json,
        application/alto-error+json
Content-Type: application/alto-endpointcostparams+json
Content-Length: 137

{
  "cost-mode" : "numerical",
  "cost-type" : "routingcost",
  "endpoints": {
          "srcs": [ "ipv4:10.0.0.0"] }
}
```

The server returns an HTTP response code of 400 with ALTO error code of E_JSON_FIELD_MISSING (c.f., Table 1 [I-D.ietf-alto-protocol]).

```
Server -> Client:
-----------------
HTTP/1.1 400 Bad Request
Content-Type: application/alto-error+json
Content-Length: ...

{
   "code": "E_JSON_FIELD_MISSING"
}
```

Test-JSON-ERR-3: An ALTO client sends a request with an unexpected type for a JSON value.

```
Client -> Server:
-----------------
POST /endpoints/cost HTTP/1.1
Host: alto.ietf.org
Accept: application/alto-endpointcost+json,
        application/alto-error+json
Content-Type: application/alto-endpointcostparams+json
Content-Length: 176

{
     "cost-mode" : "numerical",
     "cost-type" : "routingcost",
     "endpoints": {
          "srcs":"ipv4:10.0.0.0" ,
          "dsts": [ "ipv4:10.0.0.0" ]
     }
}
```

The server returns an HTTP response code of 400 with ALTO error code of E_JSON_VALUE_TYPE(c.f., Table 1 [I-D.ietf-alto-protocol]).

```
Server -> Client:
-----------------
HTTP/1.1 400 Bad Request
Content-Type: application/alto-error+json
Content-Length: ...

{
     "code": "E_JSON_VALUE_TYPE"
}
```

Test-JSON-ERR-4: An ALTO client sends a request with an invalid JSON code mode.

```
Client -> Server:
-----------------
POST /costmap/filtered HTTP/1.1
Host: alto.ietf.org
Content-Length: 105
Content-Type: application/alto-costmapfilter+json
Accept: application/alto-costmap+json

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

The server returns an HTTP response code of 400 with ALTO error code of E_INVALID_COST_MODE (c.f., Table 1 [I-D.ietf-alto-protocol]).

```
Server -> Client:
-----------------
HTTP/1.1 400 Bad Request
Content-Length: ...
Content-Type: application/alto-error+json

{
  "code": "E_INVALID_COST_MODE"
}
```

Test-JSON-ERR-5: An ALTO client sends a request with an invalid JSON code type.

```
Client -> Server:
-----------------
POST /costmap/filtered HTTP/1.1
Host: alto.ietf.org
Content-Length: 105
Content-Type: application/alto-costmapfilter+json
Accept: application/alto-costmap+json

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

The server returns an HTTP response code of 400 with ALTO error code of E_INVALID_COST_TYPE (c.f., Table 1 [I-D.ietf-alto-protocol]).

```
Server -> Client:
-----------------
HTTP/1.1 400 Bad Request
Content-Length: ...
Content-Type: application/alto-error+json

{
  "code": "E_INVALID_COST_TYPE"
}
```

Test-JSON-ERR-6: An ALTO client sends a request with an invalid JSON endpoint property type.

```
Client -> Server:
-----------------
POST /endpoints/property HTTP/1.1
Host: alto.ietf.org
Content-Length: 66
Content-Type: application/alto-endpointpropparams+json
Accept: application/alto-endpointprop+json

{
    "endpoints": ["ipv4:10.0.0.1"],
    "properties": ["foo"]
}
```

The server returns an HTTP response code of 400 with ALTO error code of E_INVALID_PROPERTYTYPE (c.f., Table 1 [I-D.ietf-alto-protocol]).

```
Server -> Client:
-----------------
HTTP/1.1 400 Bad Request
Content-Length: ...
Content-Type: application/alto-error+json

{
    "code": "E_INVALID_PROPERTY_TYPE"
}
```

Test-JSON-ERR-7: An ALTO client sends a request with multiple errors. In the particular test case below, an invalid cost type and and an invalid cost mode are sent.

```
Client -> Server:
-----------------
POST /costmap/filtered HTTP/1.1
Host: alto.ietf.org
Content-Length: 112
Content-Type: application/alto-costmapfilter+json
Accept: application/alto-costmap+json

{
    "cost-mode": "bar",
    "cost-type": "foo",
    "pids": {
       "dsts": [],
       "srcs": []
    }
}
```

The server must detect at least one of the errors and return the detected error.

```
Server -> Client:
-----------------
HTTP/1.1 400 Bad Request
Content-Length: ...
Content-Type: application/alto-error+json

{
    "code": "E_INVALID_COST_TYPE"
}
```

4. Security considerations

This document does not present any new security considerations above and beyond what is documented in the ALTO protocol [I-D.ietf-alto-protocol].

5. IANA considerations

This document does not require any action from IANA.

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.

6.2. Informative References

[I-D.ietf-alto-protocol]
Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-13 (work in progress), September 2012.

Appendix A. Acknowledgements

The editors will like to thank the ALTO working group participants for reviewing test cases. Richard Alimi and Mikio Hara contributed review cycles to the contents of this document.

Authors' Addresses

Vijay K. Gurbani (editor)
Bell Laboratories, Alcatel-Lucent

Email: vkg@bell-labs.com

William Roome
Bell Laboratories, Alcatel-Lucent

Email: w.roome@alcatel-lucent.com

Robert Varga
Cisco Systems, Inc.

Email: rovarga@cisco.com

Ning Zhang
Neustar

Email: Ning.Zhang@neustar.biz