ALTO Extension: Path Vector

draft-ietf-alto-path-vector-04

Presenter: Dawn Chen

IETF 102
July 16, 2018
Montreal
Update Overview

- Three building blocks
  - A new cost type (do not change)
  - A new entity domain (do not change)
  - Combining cost map and property map
    - Remove the multipart response to Multipart Service draft
    - Adopt the extended cost map/endpoint cost service

- Updates on Examples
- Updates on Compatibility
- Updates on Security Considerations
Update 1: Extended Cost Map Service

- Media-Type, HTTP Method, and Uses remain the same

- Capabilities

  object {
    [ ResourceId dependent-property-map; ]
    [ JSONBool allow-compound-response; ]
  } PVFCMCapabilities : FilteredCostMapCapabilities;

- Accept Input Parameters

  object {
    [PropertyName compound-properties;]
  } ReqPVFilteredCostMap : ReqFilteredCostMap;

- Response
  - A “dependent-vtags” field
  - A “property-map” field providing a property map
Update 1: Extended Cost Map Service

- Example of a cost map request and separate responses
  - ALTO client first query for the cost map and get a cost map response

<table>
<thead>
<tr>
<th>Cost Map Request Example</th>
<th>Cost Map Response Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST /costmap/pv HTTP/1.1</td>
<td>HTTP/1.1 200 OK</td>
</tr>
<tr>
<td>Host: alto.example.com</td>
<td>Content-Type: application/alto-costmap+json</td>
</tr>
<tr>
<td>Accept: application/alto-costmap+json, application/alto-error+json</td>
<td>Content-Length: [TBD]</td>
</tr>
<tr>
<td>Content-Length: [TBD]</td>
<td>Content-Type: application/alto-costmap+json</td>
</tr>
</tbody>
</table>

```json
{  
  "cost-type": {
    "cost-mode": "array",
    "cost-metric": "ane-path"
  },
  "pids": {
    "srcs": [ "PID1" ],
    "dsts": [ "PID2", "PID3" ]
  }
}
```

```json
{
  "meta": {
    "dependent-vtags": [{
      "resource-id": "my-default-networkmap",
      "tag": "75ed013b3cb58f896e839582504f622838ce670f"
    }],
    "cost-type": {
      "cost-mode": "array",
      "cost-metric": "ane-path"
    }
  },
  "cost-map": {
    "PID1": {
      "PID2": [ "ane:L001", "ane:L003" ],
      "PID3": [ "ane:L001", "ane:L004" ]
    }
  }
}
```
Update 1: Extended Cost Map Service

- Example of a cost map request and separate responses
  - ALTO client then queries the property map and get a property map response

**Property Map Request Example**

POST /propmap/ane-prop 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

```json
{
    "entities": ["ane:L001", "ane:L003", "ane:L004"],
    "properties": ["delay"]
}
```

**Property Map Response Example**

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

```json
{
    "meta": {
        "dependent-vtags": [
            "resource-id": "cost-map-pv",
            "tag": "a7d57e120ab63124e3c9a82f7a54bc120fc96216"
        ]
    },
    "property-map": {
        "ane:L001": { "delay": 46},
        "ane:L003": { "delay": 50},
        "ane:L004": { "delay": 70}
    }
}
```
Update 1: Extended Endpoint Cost Service

- Media-Type, HTTP Method, and Uses remain the same
- Capabilities

```
object {
    [ResourceId dependent-property-map; ]
    [JSONBool allow-compound-response;]
} PVFCMCapabilities : FilteredCostMapCapabilities;
```

- Accept Input Parameters

```
object {
    [PropertyName compound-properties;]
} ReqPVEndpointCostMap : ReqEndpointCostMap;
```

- Response
  - A “dependent-vtags” field
  - A “property-map” field providing a property map
Update1: Extended Cost Map Service

- Example of a endpoint cost request and a compound response

**Endpoint Cost Request Example**

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

{
    "multi-cost-types": [
        {
            "cost-mode": "array",
            "cost-metric": "ane-path"
        },
        {
            "cost-mode": "numerical",
            "cost-metric": "routingcost"
        }
    ],
    "endpoints": {
        "srcs": [ "ipv4:192.0.2.2" ],
        "dsts": [ "ipv4:192.0.2.89",
                  "ipv4:203.0.113.45",
                  "ipv6:2001:db8::10" ],
        "properties": [ "delay", "availbw" ]
    }
}
```

**Endpoint Cost Response Example**

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

{
    "meta": {
        "dependent-vtags": [{
            "resource-id": "propmap_availbw-delay",
            "tag": "bb6bb72eafe8f9bdc4f335c7ed3b10822a391cef"
        }],
        "cost-type": [{
            "cost-mode": "array",
            "cost-metric": "ane-path"
        }]
    },
    "endpoint-cost-map": {
        "ane:L001": { "availbw": 50, "delay": 46 },
        "ane:L003": { "availbw": 48, "delay": 50 },
        "ane:L004": { "availbw": 55, "delay": 70 },
        "ane:L005": { "availbw": 60, "delay": 100 }
    }
}
```
Update 2: Compatibility

- Compatibility with base ALTO protocol
  - Base ALTO clients will ignore the extended capability fields “property-map” and “allow-compound-response”.
  - Base ALTO servers will ignore the field “properties” in a request.

- Compatibility with Multi-cost Service
  - Not compatible with constraint test on array elements

- Compatibility with Incremental Updates
  - When using compound response, updates on both cost map and property map SHOULD be notified
  - When not using compound response, the ALTO server SHOULD send updates of cost map before sending updates of property map
Update 3: Security Consideration

- Confidentiality of ALTO information
  - Path Vector exposes more network information, the network is more easily exposed to attacks

- Availability of ALTO service
  - ALTO server tend to break down under frequent requests of path vector
Q & A

Thanks