

ALTO Extension: Path Vector

draft-yang-alto-path-vector-04

Presenter: Dawn Chen

IETF 98

March 31, 2017

Overview

- Draft goal: address the network graph milestone
- Status at the last IETF
 - Three key remaining issues and potential design choices for each, but no choices were made
- Progress made after last IETF
 - Investigated the design choices, and made design choice for each

Recall: Three Key Remaining Issues at IETF 96

- **Issue 1:** How to encode path vector (PV) in cost maps?
- **Issue 2:** What is the query format?
- **Issue 3:** How to provide PV network element properties (nep)?

Issue 1	Issue 2	Issue 3
Define a specific cost type for path vector	Native FCM/ECS query	Inline
A unifying scheme supporting multi-cost, and cost calendar	A new flow query format	Reference

Design Choice Made

- **Issue 1:** How to encode path vector (PV) in cost maps?
- **Issue 2:** What is the query format?
- **Issue 3:** How to provide PV network element properties (nep)?

Issue 1	Issue 2	Issue 3
<u>Define a specific cost type for path vector</u>	<u>Native FCM/ECS query</u>	Inline
A unifying scheme supporting multi-cost, and cost calendar	<u>New flow query format</u>	<u>Reference</u>

Detail: Design Choice for Issue 1 (Encode Path Vector)

- **Introduce new cost type:**
 - "cost-mode": "path-vector"
 - "cost-metric": "ane"
- **Introduce new cost-metric "ane":** an abstract network element, which can be a device, an aggregation of network links...
- **Make cost into an array (i.e., path vector):** each cost value (in Cost Maps and Endpoint Cost Maps) is a JSONArray of abstract network elements, e.g.,

```
object {  
  CostMapData cost-map;  
} InfoResourceCostMap : ResponseEntityBase;  
object-map {  
  PIDName -> DstCosts;  
} CostMapData;  
object-map {  
  PIDName -> JSONValue; // allowed to be an array of ane names  
} DstCosts;
```

Example: ["ane:L001", "ane:L002", "ane:L003", ...]

Detail: Design Choice for Issue 2 (Query Format)

- Still support legacy FCM/ECS query format

```
object {  
  CostType cost-type;  
  [JSONString constraints<0..*>];  
  [PIDFilter pids;]  
} ReqFilteredCostMap;  
  
object {  
  PIDName srcs<0..*>;  
  PIDName dsts<0..*>;  
} PIDFilter;
```

- Comment: cross-product query is limited

Detail: Design Choice for Issue 2 (Query Format)

- Introduce a new field for flows (no new media type), e.g.,

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

```
object {  
  PIDName srcs<0..*>;  
  PIDName dsts<0..*>;  
} PIDFilter;
```

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

- Comment: acceptable backward compatibility.

Detail: Issue 3 (Provide PV Element Properties)

Decided to not use inline mode:

Inline mode: embedded in the same cost map/ endpoint cost map

```
// Example of filtered cost map resource entry
```

```
"filtered-cost-map": {  
  "capabilities": {  
    "cost-type-names": ["pv-ane"],  
    "prop-types": ["delay"]  
  }  
}
```

```
// Example of filtered cost map response
```

```
{  
  "cost-map": {  
    "PID1": {"PID2": ["ane:L001", "ane:L002"], ..... }  
  }  
  "nep-map": {  
    "ane:L001": {"delay": "10"},  
    "ane:L002": {"delay": "30"} .....  
  }  
}
```

Limitations:

- Capabilities field depends on value of cost-type
- Need a new FCM/ECS response format

Detail: Design Choice for Issue 3 (Provide PV Element Properties)

Reference Mode:

// Resource 1: Resource providing path vector

```
"pv-map1" : {  
  "media-type" : "application/alto-costmap+json",  
  "accepts" : "application/alto-costmapfilter+json",  
  "capabilities" : {"cost-type-names": ["pv-cost-type"] },  
  "uses": [ "my-default-network-map"],  
}
```

// Resource 2: Resource providing network element property

```
"nep-map1": {  
  "media-type": "application/alto-propmap+json",  
  "capabilities" : {  
    "domain-types": ["ane"],  
    "prop-types": ["delay"]  
  }  
  ...  
}
```

// Resource 3: Resource providing network element property

```
"nep-map2": {  
  "media-type": "application/alto-propmap+json",  
  "capabilities" : {  
    "domain-types": ["ane"],  
    "prop-types": ["bandwidth" ]  
  }  
  ...  
}
```

Requirements:

1. Client needs to know which nep-map to get the ane properties of a response
2. Since response can depend on query, the set of ane's can be dynamically generated. Hence need to handle query specific ane's.

Detail: Design Choice for Issue 3 (Provide PV Element Properties)

Reference Mode:

// Resource 1: Resource providing path vector

```
"pv-map1" : {  
  "media-type" : "application/alto-costmap+json",  
  "accepts" : "application/alto-costmapfilter+json",  
  "capabilities" : {"cost-type-names": ["pv-cost-type"] },  
  "uses": [ "my-default-network-map" ],  
  "propertymap": "nep-map1"  
}
```

// Resource 2: Resource providing network element property

```
"nep-map1": {  
  "media-type": "application/alto-propmap+json",  
  "capabilities" : {  
    "domain-types": ["ane"],  
    "prop-types": ["delay"]  
  }  
  ...  
}
```

// Resource 3: Resource providing network element property

```
"nep-map2": {  
  "media-type": "application/alto-propmap+json",  
  "capabilities" : {  
    "domain-types": ["ane"],  
    "prop-types": ["delay", "bandwidth" ]  
  }  
  ...  
}
```

Requirements:

1. Client needs to know which nep-map to get the ane properties of a response
2. Since response can depend on query, the set of ane's can be dynamically generated. Hence need to handle query specific ane's.

Detail: Design Choice for Issue 3 (Provide PV Element Properties)

- PV-map1 query & response

```
// pv-map1 request:
{
  "cost-type": {
    "cost-mode": "path-vector",
    "cost-metric": "ane"
  },
  "pids": {
    "srcs": ["PID1", "PID3"],
    "dsts": ["PID2", "PID4"]
  }
}

// pv-map1 response:
{
  "meta": {
    "vtag": [{
      "resource-id": "pv-map1",
      "tag": "<sha256>",
      "query-id": "query_0"
    }], // Means this response is associated with query_0.
    "dependent-vtags": [{
      "resource-id": "default-network-map",
      "tag": "<sha256>"
    }],
    "cost-type": {
      "cost-mode": "path-vector",
      "cost-metric": "ane"
    }
  },
  "cost-map": {
    "PID1": {"PID2": ["ane:L01", "ane:L02"],
            "PID4": ["ane:L01", "ane:L03"]},
    "PID3": {"PID2": ["ane:L04", "ane:L02"],
            "PID4": ["ane:L05", "ane:L03"]}
  }
}
```

Introduce
query ID

- nep-map1 query & response

```
// nep-map1 request:
{
  "query-id": "query_0",
  "entities": ["ane:L01", "ane:L02", "ane:L03", "ane:L04",
              "ane:L05"],
  "properties": ["availbw"]
}

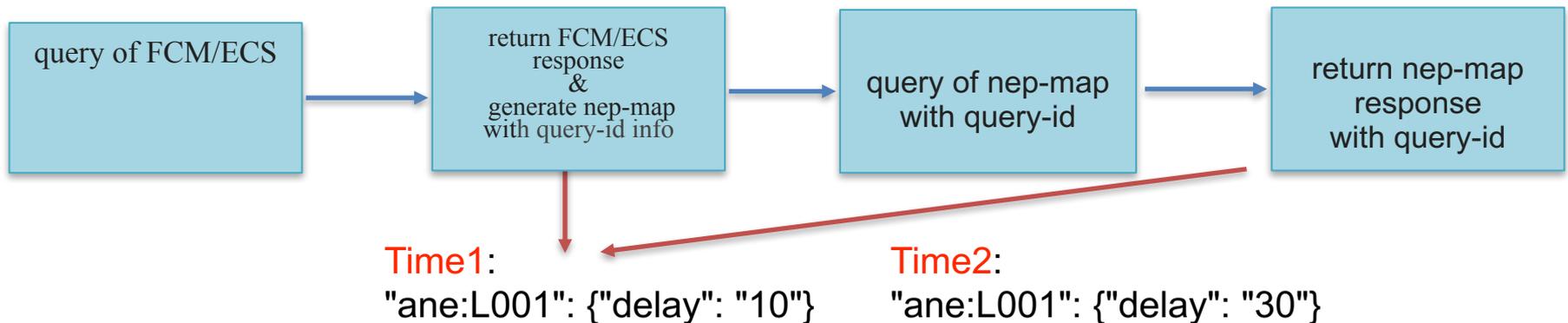
// nep-map1 response:
{
  "property-map": {
    "ane:L01": {"availbw": "30"},
    "ane:L02": {"availbw": "40"},
    "ane:L03": {"availbw": "50"},
    "ane:L04": {"availbw": "40"},
    "ane:L05": {"availbw": "70"}
  }
}
```

Other Considerations

- Compatibility with multi-cost
 - Path Vector is not a testable cost type: path vector **MUST NOT** be included in testable-cost-type-names or testable-cost-types.
 - Fields "constraints" and "or-constraints" on path vector SHOULD be regarded as error.
- Time-to-live (TTL)
 - Query specific ane's can consume resources when cached. Need a TTL to remove outdated entries.

Discussion

- Cost calendar for PV:
 - Calendar for FCM/ECS
 - Calendar for property map
- Snapshot and real-time update: query specific ane properties are likely to be snapshot, and hence no longer updates in realtime.



Summary

- Draft goal: address the network graph milestone
- Status at the last IETF
 - Three key remaining issues and potential design choices for each, but no choices were made
- Good progress made after last IETF
 - Investigated the design choices, and made design choice for each

Q & A

Thanks