Multi-Cost ALTO

draft-randriamasy-alto-multi-cost-10

S. Randriamasy

W. Roome

N. Schwan

Multi-Cost ALTO in a nutshell

- Returns ***array*** of costs instead of *scalar* cost
- Defines 'OR‘ constraints,
  - Supports **trade-offs** such as:
  - *"give me costs among {those PIDs/Endpoints} with either moderate ‘routingcost’ or 0 ‘hopcount’*
    - For example: 'hopcount' = 0 **OR** routingcost in [5, 10]"
- Proposes additional abstract cost metrics
- Applicable service information resources:
  - Cost Map (CM),
  - Filtered Cost Map (FCM),
  - Endpoint Cost Service (ECS)
- Same media types for MC service information resources
- MC Server supports both Single and Multi-Cost clients

Example Filtered multi-cost map resource in IRD

```
"filtered-multicost-map" : {
  "uri" : "http://alto.example.com/multi/costmap/filtered",
   "media-types" : ["application/alto-costmap+json" ],
   "accepts" : ["application/alto-costmapfilter+json" ],
   "uses" : [ "my-default-network-map" ],
   "capabilities" : {
      "cost-constraints" : true,
      "cost-type-names" : [ "num-routingcost",
                            "num-hopcount",
                            "num-pathoccupationc           ],
      "max-cost-types" : 3,
      "testable-cost-types": ["num-routingcost",
                              "num-hopcount",
                              "num-pathoccupationcost" ]
},
```

Base ALTO clients « see » fields in black and ignore others

Indicates that this service is MC compatible

Supports constraints on cost-types not present in the value request

MC ALTO Clients see fields in slanted blue pick in « cost-type-name »

Example filtered endpoint-multicost map resource in IRD

```
"endpoint-multicost-map" : {
  "uri" : "http://alto.example.com/multi/endpointcost/lookup",
  "media-types" : [ "application/alto-endpointcost+json" ],
  "accepts" : [ "application/alto-endpointcostparams+json" ],
  "uses" : [ "my-default-network-map" ],
  "capabilities" : {
    "cost-constraints" : true,
    "cost-type-names" : [ "num-routingcost",
                          "num-hopcount",
                          "str-status" ],
    "max-cost-types" : 3,
    "testable-cost-types": ["num-routingcost",
                            "num-hopcount",
                            "num-pathoccupationcost" ]
  }
}
```

Base ALTO clients « see » fields in black and ignore others

Supports constraints on cost-types not present in the value request

MC ALTO Clients see fields in slanted blue

Protocol updates for Multi-cost ALTO

New fields are introduced

- IRD capability field: “multi-cost-type-names”
  - Array of cost type names.
  - Presence means this resource can return a multi-cost.

- **RULE**: A filtered cost map resource can have either cost-type-names or multi-cost-type-names or both. Former means it can return a single cost, latter a multi cost. Client selects which.

- **RULE**: A full cost map resource has cost-type-names or multi-cost-type-names, but not both. Former means it returns a single cost, latter means it returns a multi cost. Since it is GET mode, the resource returns what it returns; the client has no choice

- ➔ **DISCUSSION**: to define full MC map without inventing new MIME or breaking Clients we propose POST requests on Filtered MC Maps with no inputs

Protocol updates for Multi-cost ALTO

- IRD: “testable-cost-types”:
  - Array of cost-type names that can be used in Multi- Cost and Logical Operator (MCLO) constraints.
  - Presence means this resource supports MCLO constraints.

- Server response fields: “multi-cost-types”:
  - array of cost-types used in meta of multi-cost response, instead of “cost-type”.
  - **RULE**: **cost-types MUST be in the order in which they appear in the multi-cost values array.**

- Client request fields: “multi-cost-types”:
  - For multi-cost aware filtered cost map resources,
  - to get a multi-cost response, client provides “multi-cost-types”, instead of “cost-types”,
  - **RULE: the array of cost-types MUST be in the order in which the server should return them.**

Example request for filtered MC Map - § 6.3.8

Trade-off filtering:

- OR-Constraints: (‘routingcost’ > 5 **AND** <10) **OR** (‘hopcount’ = 0)
- From ["PID1", "PID2"] to ["PID1", "PID2", "PID3"]

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

{
    "multi-cost-types" : [
      {"cost-mode": "numerical", "cost-metric": "routingcost"},
      {"cost-mode": "numerical", "cost-metric": "hopcount"}
    ],
    "or-constraints" : [ ["[0] ge 5", "[0] le 10"],
                        ["[1] eq 0"] ]
    "pids" : {
      "srcs" : [ "PID1", "PID2" ],
      "dsts" : [ "PID1", "PID2", "PID3" ]
    }
}
```

Example response – filtered MC Map § 6.3.8

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

{
"meta" : {
  "dependent-vtags" : […],
  "multi-cost-types" : [
    {"cost-mode": "numerical", "cost-metric": "routingcost"},
    {"cost-mode": "numerical", "cost-metric": "hopcount"}]
  }

"cost-map" : {
  "PID1": { "PID2": [5,23], "PID3": [10,5] },
  "PID2": { "PID2": [1,0]                  }
 }
}
```

RULE: Source/Destination pairs for which the Path Costs do not meet the constraints MUST NOT be included in the returned Cost Map

Next steps

- Request adoption as WG item
- Next versions
  - Make full Multi-Cost Maps available only via POST request for filtered MC Maps with no constraints
  - Clean up text
  - Integrate WG feedback

THANK YOU

Back-up

Motivation – use cases

- Use multiple selection metrics for endpoints and e2e paths
  - To jointly meet application needs while keeping network awareness
    - E.g. by ***jointly*** getting ʻroutingcostʼ meeting NP interests and ʻbandwidth scoreʼ meeting app interests

- Save time and bandwidth on ALTO requests
  - 1 Multi-Cost transaction on N metrics rather than N on 1 metric
  - 1 Multi-Cost Map is smaller than N Cost Maps

- Consistency of metric values
  - Different cost-types may change at different paces
  - For multi-variate optimization

- Enrich filtering constraints to represent compromises, e.g.
  - *select paths with moderate 'routingcost' OR null 'hopcount'*

Multi-Cost transactions

- Multi-Cost Requests and responses convey an *Array* of costs
  - Array may contain any Cost Mode combination
    - Requested Cost-types array

      `["num-routingcost", "ord-hopcount", "string-status"]`
    - Taking values:

      `[23, 6, "medium"]`
  - **RULE:** cost values for each Source/Destination pair MUST be provided in the same order as in the array of Multi-Cost Types

Design

- Suggested new properties and costs
  - Aggregate values with or without units
    - EP-Nominal Memory, EP-Nominal Bandwidth
    - EP Occupied memory, EP Occupied bandwidth,
    - Path Occupation Cost, // or Bandwidth Score,

- Multi-Cost filtering constraints
  - Combine AND and OR operators
  - Are applied to cost-types present in value request
    - NOTE: [draft-lee-alto-app-net-info-exchange] proposes to use constraints on metrics not present in value request

Extensions for Multi-cost ALTO

- New Capability of applicable information resources in the IRD:
  - "multi-cost-type-names": a list of cost type names that can be used in Multi-Cost requests

- MC-aware ALTO client requests
  - have one different "meta" field called "multi-cost-types", which is the array of cost-types provided in the MC aware Client requests for MC Services.

- MC-aware ALTO server responses to multi-cost requests
  - have one different "meta" field called "multi-cost-types", which is the array of cost-types provided in the MC aware Client requests for MC Services.

- Multi- Cost and Logical Operator (MCLO) constraints:
  - capabilities of FCM and ECM have one additional field called "testable-cost-types" listing the cost types that can be used in MCLO constraints.

  ➔ **can have constraints on cost-types not present in the value request**