draft-ietf-alto-multi-cost-00.txt

Updates since IETF92

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

Revised ToC

- New: section 3 « overview of approach », non normative design
  - Replaces section 3 « Uses Cases For Using Multiple Costs”
- Section 4: specification of protocol extensions
  - Filtered cost map (FCM) extensions
  - Endpoint Cost Service (ECS) extensions
- Section 5: examples
  - IRD,
  - FCM: empty PID filtering, OR-constraints on 1 or 2 metrics
- Section 6: IANA considerations
  - None because no new media-types are introduced
- Section 7: security and privacy
  - Nothing new wrt base protocol

Section 3 – diffs

- Backwards compatibility with RFC 7285
  - MC ALTO Server MUST support legacy Clients
  - RFC 7285 section 8.3.7: *"Extensions may include additional fields within JSON objects defined in this document. ALTO implementations MUST ignore unknown fields when processing ALTO messages.“*
    - if legacy clients GET a full Multi-Cost Map, they are not able to interpret the value array because they ignore the meta explaining them
  - **➔ Filtered Cost Map service only**
    - **For full cost map: use empty SRC & DEST**
- FCM resources in the IRD
  - **`"testable-cost-types": removed`**

Section 4 – highlights - 1

- FCM and ECS extensions
  - New input parameter: **`testable-cost-types`**
  - Allows a client to test on a metric without receiving its values
  - Example: client wants metrics A, B with constaints on metric C
    - Puts metric C: in **`testable-cost-types`** list

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-pathoccupationcost" ],
     "max-cost-types" : 3,
]            }
},
```

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

Indicates that this service is MC compatible

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

Example § 5.4: full MC Map - with testable cost types-1

```
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
{
     "cost-type" : {
       "cost-mode": "numerical", "cost-metric": "routingcost"     },
     "testable-cost-types" : [
       {"cost-mode": "numerical", "cost-metric": "routingcost"},
       {"cost-mode": "numerical", "cost-metric": "hopcount"}
    ],
     "or-constraints": [
           ["[0] le 10", "[1] le 2"],
           ["[0] le 3",  "[1] le 6"]
     ],
    "pids" : {
       "srcs" : [ ],
       "dsts" : [ ]
    }
  }
```

Example § 5.4: full MC Map - with testable cost types -2

```
HTTP/1.1 200 OK
Content-Type: application/alto-costmap+json
{
     "meta" : {
       "dependent-vtags" : [
         {"resource-id": "my-default-network-map",
          "tag":"3ee2cb7e8d63d9fab71b9b34cbf764436315542e"
        }
      ],
       "cost-type" : {
        "cost-mode": "numerical", "cost-metric": "routingcost"
      }
     }
    "cost-map" : {
       "PID1": { "PID1": 1, "PID3": 10 },
       "PID2": { "PID2": 1 },
       "PID3": { "PID3": 1 }
    }
   }
```

Next steps

- Next versions
  - Further explain why full Multi-Cost Maps are not available via GET requests
    - Extend explanation in section 3.2 « compatibility with legacy clients »
  - Integrate WG feedback
  - Clean up text

THANK YOU

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 'routingcost' meeting NP interests and 'bandwidth score' 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