Multi-Cost ALTO

Updates in
draft-randriamasy-alto-multi-cost-07

S. Randriamasy (ed.)
B. Roome
N. Schwan
ALTO Multi-Cost Extension

• Multi-Cost ALTO allows to include several Cost-Types in ALTO requests and responses

• New Cost Types are proposed
  – Path Occupation Cost, EP Occupation Cost EP
  – Nominal Memory, EP Occupied memory

• Specified Multi-Cost Services
  – Multi-Cost Map Service
  – Filtered Multi-Cost Map Service
  – Endpoint Multi-Cost Service
Motivation

• Optimize time and bandwidth
  – 1 MC transaction is faster than N single cost
  – Multi-Cost Map instead of N Cost Maps Lighter to store and transport than N Cost Maps

• Facing unpredictable and/or rapid value changes
  – ALTO Client can get consistent snapshot of several different rapidly varying Cost Type values

• Suitable ALTO Services for multi-cost
  – Endpoint Multi-Cost service
  – Filtered multi-cost map
V7-diffs – any mode allowed

- There is no more incentive rule to require multiple costs in the 'numerical' mode (when applicable).
- A multi-cost request can be done for any purpose, such as updates on one cost type in 'ordinal' combined with 'numerical' information on another cost type.
- The purpose does not restrict to multivariate optimization.
V7-diffs – MC constraints

• New: "4.6. Extended constraints on multi-cost values". It extends the base protocol in two ways:
  – (1) combining constraints on multiple metrics related with a logical 'AND', to cover requirements such as:
    • "select solutions with moderate routingcost" AND "low hopcount",
    'hopcount' values in [20,50] AND 'routingcost' values in [30,100]
V7-diffs – MC constraints

• Combination of constraints on multiple metrics related with both a logical 'AND' and a logical 'OR'.

• To cover constraints such as:

  - 'hopcount' values in [6,20) OR 'routingcost' values in [100,200]

• To allow an application to make compromises such as:

  - select solutions with either moderate 'hopcount' and high 'routingcost' OR higher 'hopcount' and moderate 'routingcost'

\[
\text{[('hopcount' ge 6) AND ('hopcount' lt 20) AND ('routingcost' ge 100) AND ('routingcost' le 200)] OR}\]

\[
\text{[('hopcount' ge 20) AND ('hopcount' le 50) AND ('routingcost' ge 30) AND ('routingcost' le 100)]}
\]
POST multi/multicostmap/filtered HTTP/1.1
Host: alto.example.com
Content-Type: application/alto-multicostmapfilter+json
Accept: application/alto-multicostmap+json, application/alto-error+json
{
  "cost-mode": ["numerical", "numerical"],
  "cost-type": ["routingcost", "hopcount"],
  "or-constraints": [
    ["[0] ge 5", "[0] le 10"],
    ["[1] eq 0"]
  ],
  "pids": {
    "srcs": ["PID1", "PID2"],
    "dsts": ["PID1", "PID2", "PID3"]
  }
}
Thank you

back-up slides follow
Example request – MC ECS

• Cost Typs:: « routingcost » = monetary cost and « hopcount » used to figure out delay

POST multi/endpointmulticost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: [TODO]
Content-Type: application/alto-endpointmulticostparams+json
Accept: application/alto-endpointmulticost+json,application/alto-error+json
{
  "cost-type" : ["routingcost", "hopcount"],
  "cost-mode" : ["numerical", "numerical"],
  "endpoints" : {
    "srcs": [ "ipv4:192.0.2.2" ],
    "dsts": [ 
      "ipv4:192.0.2.89",
      "ipv4:198.51.100.34",
      "ipv4:203.0.113.45"
    ]
  }
}
Example response – MC ECS

HTTP/1.1 200 OK
Content-Length: [TODO]
Content-Type: application/alto-endpointmulticost+json
{
  "meta" : {},
  "data" : {
    "cost-type" : ["routingcost", "hopcount"],
    "cost-mode" : ["numerical", "numerical"],
    "map" : {
      "ipv4:192.0.2.2": {
        "ipv4:192.0.2.89" : [1, 7],
        "ipv4:198.51.100.34" : [2, null],
        "ipv4:203.0.113.45": [3, 2]
      }
    }
  }
}