Multi-Cost ALTO

Updates in
draft-randriamasy-alto-multi-cost-05
S. Randriamasy(ed.), N. Schwan
Outline

• Version presented at Quebec was 03
  – Sketched a Multi-Cost ALTO transaction for the EP Cost Service
  – Possible inclusion of multiple cost types in transactions approved
  – Proposal of time sensitive cost types and use cases

• Main part of 05 - Multi-Cost (MC) Services
  – Extensions of ALTO protocol and ALTO services to include several cost types in 1 ALTO transaction
  – Introduces new Multi-Cost specific ALTO services
    • Specifies MC Service URIs with associated objects and formats
    • Specifies MC transactions for those MC Services
    • Example of transactions
    • Example of IRD with MC services and capabilities

• Separate section on multiple cost values for one Cost Type
  – Introduces «dynamic» Cost Mode and use cases
  – Should be separated from Multi-Cost topic & discussed separately
Objectives of Multi-Cost

• Gain time and resources by
  – Transport information on N Cost Types in 1 ALTO transaction rather than in N transactions

• 1 Multi-Cost Map instead of N Cost Maps
  – Less bulky to store than N Cost Maps
    • At the Client side
    • In an ALTO server
      – (although storage in ALTO Servers is out of ALTO scope)
  – Represents a smaller data volume to transport
  – 1 MC transaction is faster than N single cost
  – Same for Filtered MC Map

• Endpoint Multi-Cost service
  – Faster and easy
On ALTO Multi-Cost services

• Term EP covers
  – Peer, CDN storage location, party in grid computing or on-line gaming or other resources sharing applications.

• Properties have constant values, costs can vary

• Rule1
  – when multiple cost types are requested then the requested Cost Mode MUST be numerical for those Costs Types encoded in JSONNumber
    • Reason: avoid mixing ordinal and numerical costs, requests too complex to handle and ordinal is easy to retrieve from numerical
    • Does not apply to Costs encode with JSONBool, JSONObject

• Rule2 – value order specification
  – The ALTO response, MUST include an array of cost-types, arranged the same way as the values
  – The cost values for Source/Destination pairs are provided in the same order as in the array of cost types
Specified Multi-Cost Services

- Multi-Cost Map Service
- Filtered Multi-Cost Map Service
- Endpoint Multi-Cost Service
- New media-type for
  - MC map services and EP MC service
- New object types describing
  - The resources capabilities,
  - Input parameters
  - The responses
- Example of MC ALTO requests and responses
  - For each of the 3 services
ALTO Multi-Cost transaction

- Multi-costs values are now objects of type DstMultiCosts represented with JSON type JSONArray

- A MC Request contains array of N requested Costs Types
  - and array of associated requested Cost Mode

- A MC Response contains
  - Array of Cost Types
    - Specifies in which order cost values are provided for S/D pairs
  - Array of associated Cost Modes (should come after the Cost Types)
    - To cover Costs that are not numerical e.g. Boolean, …
  - Map of Costs for S/D pairs encoded with the JSONArray type
    - Arrays of elements of different JSON types
      - E.g. [JSONBool, JSONNumber]

- New object MultiCostMapData
  - Contains object DstMultiCosts [PIDName] <0..*> 
    - Contains a JSON Array [PIDName]
Example response – MC Map

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

{
    "meta" : {},
    "data" : {
        "cost-mode" : ["numerical", "numerical"],
        "cost-type" : ["routingcost", "hopcount"],
        "map-vtag" : "1266506139",
        "map" : {
            "PID1" : { "PID1" : [1,6], "PID2" : [5,23], "PID3" : [10,5] },
            "PID2" : { "PID1" : [5,5], "PID2" : [1,11], "PID3" : [15,9] },
            "PID3" : { "PID1" : [20,12], "PID2" : [15,1], "PID3" : [1,18] }
        }
    }
}
Example ALTO Multi-Cost services & capabilities in IRD

```json
{
  "resources" : [ 
    { 
      "uri" : "http://alto.example.com/multi/maps",
      "media-types" : ["application/alto-multicostmap+json"],
      "accepts" : ["application/alto-multicostmapfilter+json"],
      "capabilities" : {
        "cost-constraints" : true,
        "cost-types" : ["routingcost", "hopcount"],
        "cost-modes" : ["numerical", "numerical"]
      }
    }, {
      "uri" : "http://alto.example.com/multi/endpointmulticost/lookup",
      "media-types" : ["application/alto-endpointmulticost+json"],
      "accepts" : ["application/alto-endpointmulticostparams+json"],
      "capabilities" : {
        "cost-constraints" : true,
        "cost-types" : ["routingcost", "hopcount"],
        "cost-modes" : ["numerical", "numerical"]
      }
    }
  ]
}
```
Example request – Filtered MC Map

• Suppose Cost Type « routingcost » = monetary cost.

• Client wants to figure out delay, so it requests Type « hopcount »

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",
  "pids" : {
    "srcs" : [ "PID1" ],
    "dsts" : [ "PID1", "PID2", "PID3" ]
  }
}
Example response – Filtered MC Map

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

{
    "meta": {},
    "data": {
        "cost-mode": ["numerical", "numerical"],
        "cost-type": ["routingcost", "hopcount"],
        "map-vtag": "1266506139",
        "map": {
            "PID1": { "PID1": [1,6], "PID2": [5,23], "PID3": [10,5] }
        }
    }
}

Multi-Cost ALTO - Diffs in v5

November 16th 2011
Representation of varying cost values

• This Section should be separated from the Multi-Cost draft
  – Needs further discussion and specification
• Cost values at different time periods useful
  – E.g. to schedule data transfers across time zones
  – Need appropriate « scope » attributes
  – New Mode = « Dynamic »
    • Appended to existing modes names?
  – Costs values in « Dynamic » mode represented with JSONArray
• Different from case frequently varying values available via repeated
  ALTO transactions that frequently provide « static » values
• Proposed varying cost types: POC, EPOC
• RULE:
  – Arrays of values of the same JSON type
  – These cost values MUST be available as a single value as well
• Applicable cost modes:
  – Numerical, boolean, ordinal (? Not sure…), array?,
• Example of URI and transaction with numerical cost types
  – Restricted to EP Multi-Cost Service
Related discussions

• List discussions related to MC
  – «Opaque» cost types
    • Should be changed to something suggesting it can be interpreted
  – Specific cost values
    • Such as « Not applicable », « not available », …
    • ALTO WG suggestions:
      – numerical indicator such as -1
      – Use « null »
      – 
      – Need to list the « special » values

• Separate discussions needed on « dynamic » costs
Thank you

back-up slides follow
Context

• Current ALTO protocol provides information on cost between source/destination pairs of EPs or PIDs
  – Cost Map Service
  – Filtered Cost Map service
  – Endpoint Cost

• Each ALTO transaction provides information on only one cost at a time
  – ‘routingcost’ = mandatory

• Todays applications require QoE specific Cost information
  ➔ more than one Cost is needed
  – Set of Costs is specific to application

• In addition current ALTO Costs are « static »
  – Routing cost, hop count.
  – ➔ need synthetic metrics reporting on e.g. path bandwidth, delay, availability, loss.
UC3: data transfer scheduling with « dynamic » costs

- CDNs need to regularly transfer their data for dissemination purposes
  - Need to avoid interfering with user peak activity
- Particular groups of users have limited access
  - to network and/or resources in time
- In both cases
  - Fixed/limited choice on target locations
  - Need for bandwidth
    ➜ Need to schedule their transfers
    ➜ Need information at various time periods on e.g.
      - Path occupation
      - Routing cost
Example response MC EP Cost with 1 cost in dynamic mode

« dynamic » should be another attribute rather than a Cost Mode

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

```json
{
   "meta": {},
   "data": {
      "cost-type": ["routingcost", "pathoccupationcost"],
      "cost-mode": ["numerical", "dynamic"],
      "map": {
         "ipv4:192.0.2.2": {
            "ipv4:192.0.2.89": [1, [7, ..., 24 values]],
            "ipv4:198.51.100.34": [2, [4, ..., 24 values]],
            "ipv4:203.0.113.45": [3, [2, ..., 24 values]]
         }
      }
   }
}
```