draft-randriamasy-alto-cost-calendar-06

Recap and updates since v06

July 21st, 2016 @ IETF 96 - Berlin

Sabine Randriamasy

Y. Richard Yang

Qin Wu

Lingli Deng

Nico Schwan

Updates since v05

- IPv6 examples for the Endpoint Cost Map service
- Author affiliation

ALTO Cost Calendar in a nutshell

- ALTO Calendar: allows deciding where to connect ***and when***
  - Array of time-dependent cost values for a given metric,
  - Set of attributes describing time scope of the calendar
- Allows Delay tolerant applications to schedule their connections
  - Optimal time for data transfers
- Allows ALTO Clients to schedule their Calendar requests
  - ALTO servers may save transactions on repeated value arrays
- Applicable to
  - time-sensitive ALTO metrics
  - Filtered Cost Map (FCM)
    - for full Cost Map: use empty SRC & DEST
  - Endpoint Cost Map (ECM)
- Addresses target WG item: cost extensions (May 2014)

ALTO Calendar design

- Backwards compatibility with legacy Clients and Multi-Cost Map
  - Calendars associated to ALTO information resources
  - Calendar attributes specified in
    - IRD information resources of IRD
    - "meta" member of ALTO Server responses
- Does not introduce a new mode
- Does not introduce new media types
- Compatible with **all** cost-modes
  - numerical, string, …

Simple extension design

- IRD for calendared resources
  - Exposes attributes allowing to understand the calendar
    - "cost-type-names“, "time-interval-size“, "number-of-intervals"

- ALTO request for FCM and ECM
  - 1 member added: **"calendared" : [true]**

- ALTO Server responses for FCM and ECM
  - Add calendar attributes and their value
  - May OPTIONALLY use attribute "repeated"
    - When ALTO value arrays are repeated for N periods
    - To avoid useless processing of requests for unchanged values

- 3 RULES to be included in draft on Calendar information updates
  - RULE 1: Calendar start and duration VS request date
  - RULE 2: “HTTP Last-Modified” VS Calendar start and duration
  - RULE 3: “HTTP Last-Modified” VS Calendar start and duration for repeated values

ALTO Calendar v05- example IRD - §3.3

```
"endpoint-cost-calendar-map" : {
        "uri" : "http://custom.alto.example.com/calendar/endpointcost/calendar/lookup",
        "media-types" : [ "application/alto-endpointcost+json" ],
        "accepts" : [ "application/alto-endpointcostparams+json" ],
        "capabilities" : {
          "cost-constraints" : true,
          "cost-type-names" : [ "num-routingcost", "num-latency",
                                "num-pathbandwidth", "string-service-status" ],
          "calendar-attributes" : [
            {"cost-type-names" : "num-routingcost",
             "time-interval-size" : "1 hour",
             "number-of-intervals" : 24
            },

            // … calendar attributes for "num-latency", "num-pathbandwidth“ …

            {"cost-type-names" : "string-service-status",
             "time-interval-size" : "2 minute",
             "number-of-intervals" : 30
            },
           ]
          "uses": [ "my-default-network-map" ]
        } // ECM capab
```

Calendar-aware clients understand text in blue.
Others ignore it

ALTO Calendar v05- example ECM - § 4.2.3

```
POST /calendar/endpointcost/lookup HTTP/1.1
Host: alto.example.com  Content-Length: [TODO]
Content-Type: application/alto-endpointcostparams+json
Accept: application/alto-endpointcost+json,application/alto-error+json

{    "cost-type" : {"cost-mode" : "numerical", "cost-metric" : "routingcost"},
     "calendared" : [true],
     "endpoints" : {
       "srcs": [ "ipv4:192.0.2.2" ],
       "dsts": [
         "ipv4:192.0.2.89",
         "ipv4:198.51.100.34",
         "ipv4:203.0.113.45"      ]
   }
}
```

ALTO Calendar v05- examples ECM - §4.2.3

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

{    "meta" : {
     "cost-type" : {"cost-mode" : "numerical", "cost-metric" : "routingcost"},
     "calendar-response-attributes" : [
        { "calendar-start-time" : Mon, 30 Jun 2014 00:00:00 GMT,
          "time-interval-size" : "1 hour",
          "numb-intervals" : 24,
          "repeated": 4 }        ], // means: same value array for Monday, Tuesday, Wednesday, Thursday
        } // end meta

     "endpoint-cost-map" : {
       "ipv4:192.0.2.2": {
         "ipv4:192.0.2.89"    : [v1, v2, ... v24],
         "ipv4:198.51.100.34" : [v1, v2, ... v24],
         "ipv4:203.0.113.45"  : [v1, v2, ... v24]
       }
     }
 }
```

Next steps

- Draft is at the ALTO WG Item document adoption stage
- Additional comments and suggestions are welcome

Thank you

Back-up follows

Calendar rules

- **RULE 1: Calendar start and duration VS request date**

  an ALTO Server indicating Calendars for a given cost-type in its IRD resources MUST provide one

  - That begins at TS = “calendar-start-time” and
  - with values for a duration DU = ("time-interval-size" * "number-of-intervals")
  - Such that: if TR is the date of the client request, TR lies in the interval [TS, TS+DU]

- **RULE 2: “HTTP Last-Modified” VS Calendar start and duration**

  we should not have values HL of “HTTP Last-Modified” such that HL < TS-DU since the design assumes that the Calendar values are updated periodically at intervals equal to DU.

  - If the Server does not provide a Calendar on the next period for a cost-type, it MUST NOT list this Cost-Type in the “cost-type-names” member of calendared IRD resources.

- **RULE 3: “HTTP Last-Modified” VS Calendar start and duration for repeated values**

  IF THE SERVER USES MEMBER “repeated” in its responses and if “repeated” has a value n>1 then we can have HL < TS-DU and RULES 1 and 2 are replaced by RULE 3, see examples of section 4.2.3

  - we MUST have TR is the date of the client request, TR lies in the interval [TS, TS+n*DU]

FCM and ECS specifications in v05

- FCM and ECS request must add 1 input parameter
  - `JSONBoolean    calendared<1..*>`
    - //list size = number of requested cost types

- FCM and ECS responses have 1 additional field in « meta »
  - `CalendarResponseAttributes calendar-response-attributes <1..*>;`

```
 object{
    JSONString    calendar-start-time;
    JSONString    time-interval-size;
    JSONNumber    number-of-intervals;
    [JSONNumber   repeated;] [OPTIONAL]
 // for «periodic» calendar-start-time: number of calendar iterations with
same values
}CalendarResponseAttributes;
```

- Calendared Cost values are JSONArrays of time-dependent JSONValues