ALTO Incremental Updates
draft-schwan-incr-updates-01

Nico Schwan <nico.schwan@alcatel-lucent.com>
Bill Roome <w.roome@alcatel-lucent.com>

Presented by Vijay Gurbani

IETF-83
Paris, France
Thursday, March 29, 2012
Outline

• Problem Statement
• Existing HTTP Mechanisms
• Incremental Change Messages
• Our Proposal
• Conclusion
Problem Statement

• ALTO Network and Cost Maps can be large, e.g.
  – Network Map: 5000 PIDs, 10 CIDRs per PID \(\rightarrow\) \(\sim\) 1.25 MB
  – Cost Map: 5000 PIDs \(\rightarrow\) 5000x5000 matrix \(\rightarrow\) \(\sim\) 417 MB

• Estimated update frequency:
  – Network Map: Maybe once every day or two
  – Cost Map: *Something* changes every few minutes

→ Need conditional *and* incremental updates to *avoid retransmission of full map* for every change

• Draft lists options where client polls ALTO server and server decides based on client request
Existing HTTP Mechanisms

• Conditional retrieval:
  – **If-Modified-Since** header:
    • Use **Last-Modified** date returned by ALTO server in previous full map response.
  – **If-None-Match** header:
    • Use **Etag** returned by ALTO server in previous full map response.

• Partial retrieval:
  – **Range** header:
    • Fetch a byte range. Fine if a GET stopped 25 megs into a 50 meg file.
    • Not useful for fetching changes when updates change the byte offsets.

• Compression:
  – Even at 10:1 compression, 400 megs is still a lot of data.

• Conclusion:
  – HTTP might work for conditional retrieval of a full map, but not for incremental retrieval of changes.
Incremental Change Messages

- **JSON Patch:**
  - `{ "replace": "/data/map/PID1/ipv4",  
  "value": ["192.0.2.0/24", "198.51.200.0/25"] }  
  - `{ "delete": "/data/map/PID2" }`

- But the existing Filtered Network Map and Cost Map response messages work just as well!
  - **Network Map:**
    - For each PID in the message, replace previous CIDRs with new CIDRs
    - To delete a PID, use “delete” as the value (or an empty array)
    - PIDs not in the message stay the same
  - **Cost Map:**
    - Costs in the message replace the previous costs for those source/dest PIDs
    - To delete a cost, use the value “delete” (or “-1”, or “NaN”, or …)
    - Costs not in the message stay the same
Our Proposal (Overview)

• Two new requests: **Get Network Map Updates** and **Get Cost Map Updates**
  
  – Response is Filtered Network Map or Filtered Cost Map message
  
  – Add a cost map version tag to full Cost Map responses:

    ```json
    {"data": {"cost-vtag": "1266506140", "map": ... }}
    ```
  
  – Update requests are POST, with a simple input message of MIME type “application/alto-update-param+json”. The input gives the tag for the client’s current network map or cost map:

    ```json
    {"reference-tag": "1266506140"}
    ```
Our Proposal (Continued)

- New optional field in response:
  
  ```json
  {"full-map": true, ...}
  ```

  If true, the response is the full map, not an incremental update.
  
  • E.g., the client’s map is so old that the server can no longer provide incremental changes relative to that version.

- The Cost Map Update request returns the cost-vtag for the new version

- Update URIs are identified in IRD with a capability:

  ```json
  { "uri" : "http://alto.example.com/incrementalupdate/networkmap",
    "media-types" : [ "application/alto-networkmap+json" ],
    "accepts" : [ "application/alto-update-param+json" ],
    "capabilities" : { "cost-type" : "routingcost",
                        "cost-mode" : "numerical",
                        "incremental-update" : true } }
  ```
Example

• Request:

POST /incrementalupdate/costmap
Content-Type: application/alto-update-param+json
Accept: application/alto-costmap+json

{ "reference-tag": "1266506140" }

• Response:

HTTP/1.1 200 OK
Content-Type: application/alto-costmap+json

{ "meta": {},
  "data": {
    "cost-mode": "numerical",
    "cost-type": "routingcost",
    "map-vtag": "314159",
    "cost-vtag": "1266506141",
    "full-map": false,
    "map": { "PID1": { "PID2": 1, "PID3": 2 } }
  }
}
Conclusion

• Current draft identifies partial update options
  – Identify Map Version
    • HTTP
    • ALTO Extension
  – Partial Update Options
  – Information Resource Capability

• Next steps:
  – Other options?
  – Identify most suitable option