ALTO Protocol

draft-ietf-alto-protocol-14

Richard Alimi (Ed.), Reinaldo Penno (Ed.), Stefano Previdi, Stanislav Shalunov, Richard Woundy, Y. Richard Yang (Ed.)

Grateful to contributions from large number of collaborators; see draft for complete list.

Outline

- Changes between -13 and -14
- Additional changes (between -14 and to be published -15)

Changes: -13 → -14

- Revisions according to reviews by AD Martin and others
- Split a single section (Section 6 of -13) into multiple sections:

  - Protocol Specification: General Processing (Section 7 of -14);

  - Protocol Specification: Basic ALTO Data Types (Section 8 of -14), and

  - Protocol Specification: Service Information Resources (Section 9 of -14).

  The goal of the split is to better show the general structure of the ALTO protocol, to allow the possibility (or evaluation of the possibility) of extensions.

- IANA considerations

  - Added a table (Table 3) to summarize Cost Types.

  - Added a table (Table 4) to summarize Endpoint Property Types.

  - Added a table (Table 5) to summarize Address Types.

Changes: -13 → -14

- Defined the meaning of that two Version Tags match; added a sentence on potential collision probability (Sec. 5.3 of -14)
- Added that an ALTO Server MUST define the 'pid' Endpoint Property Type and that the Version Tag of the Network Map used to return the pid property MUST be included. (Sec. 6.1 of -14 and the example in Sec. 9.3.1.6)
- Added a constraint on Cost Type: "For an identifier with the 'priv:' or 'exp:' prefix, an additional string (e.g., company identifier or random string) MUST follow to reduce potential collisions. For example, a short string after 'exp:' to indicate the starting time of a specific experiment is recommended." (Sec. 8.5 of -14)
- Revised the discussion on Hosts with multiple endpoint addresses (Sec. 11.2 of -14)
- Added a suggestion on the possibility of a monitoring service (Sec. 14.1.4 of -14).

Consensus Status on WG Discussions

- Item: Specify behaviors of degenerated map filtering service
  - Consensus: No objection if Default to complete map
  - Revision: Already in Sec. 14.1.1 of -14 by adding a clarification that filtering service may degenerate into a full map and hence becomes an operation issue

- Item: Endpoint property
  - Consensus: Introduced an endpoint property registry
  - Revision: Already in Table 4 -14

- Item: Unifying cost-mode and cost-type to a single type
  - Consensus: Not unifying but simplify
  - Proposed revision: see next slide

Proposed Change on Handling cost-mode and cost-type

- Introduce a new (syntax sugar) type combining mode and type to disallow arbitrary combinations:

```
object {
   CostMode mode;
   CostType type;
} CostID;
```

Proposed Change on Handling cost-mode and cost-type

```
// IRD
  …{
   "uri" : "http://alto.example.com/costmap/num/hopcount",
   "media-types" : [ "application/alto-costmap+json" ],
   "capabilities" : {
       // OLD:
       // "cost-mode" : [ "numerical" ],
       // "cost-type" : [ ″routingcost" ]
       "cost-id" : {"mode" : "numerical", "type": ″routingcost"}
 }, {
   "uri" : "http://alto.example.com/endpointcost/lookup",
   "media-types" : [ "application/alto-endpointcost+json" ],
   "accepts" : [ "application/alto-endpointcostparams+json" ],
   "capabilities" : {
       "cost-constraints" : true,
       // OLD:
       // "cost-modes" : [ "ordinal", "numerical" ],
       // "cost-types" : [ "routingcost", "hopcount" ]
       // Assume server will not reveal numerical of hopcount
       "cost-ids" : [ {"mode":"ordinal",  "type":"routingcost"},
                      {"mode":"numerical","type":"routingcost"},
                      {"mode":"ordinal",  "type":"hopcount"}
       ]
    }
```

Proposed Change on Handling cost-mode and cost-type

```
// Example
  HTTP/1.1 200 OK
  Content-Length: 262
  Content-Type: application/alto-costmap+json

  {
    "meta" : {},
    "data" : {
      "cost-id" : {"mode":"numerical", "type":"routingcost"},
      "map-vtag"  : "1266506139",
      "map" : {
        "PID1": { "PID1": 1,  "PID2": 5,  "PID3": 10 },
        "PID2": { "PID1": 5,  "PID2": 1,  "PID3": 15 },
        "PID3": { "PID1": 20, "PID2": 15   }
      }
    }
  }
```

Handling TLS/SSL

- Previous: An ALTO Server MUST support SSL/TLS [RFC5246] to implement server and/or client authentication ... (Sec. 7.3.5 in -14; Sec. 6.3.5 in -13)
  - Proposed change:
    - A written version sent to the mailing list
    - An ALTO Server MUST support both HTTP/HTTPS schemes.
    - If client authentication is desired, the operator can use either HTTP Digest or Client Certificate.

Discussion: Interpreting HTTP Status Code and/or ALTO Status Code

- Current wording: An ALTO Client MUST interpret both HTTP Status Code and ALTO Error Code. If the ALTO Error Code indicates an error, the ALTO Client should consider that the request has failed." (Sec. 7.7 of -14)

Other Change (to be published)

- Revision of the ALTO JSON schema to allow easier parsing (for some implementations)
  - Will not change the data format on the wire but allows implementation to use either HashMap or object to represent ALTO JSON data
  - Will change if no objection