ALTO Protocol

draft-ietf-alto-protocol-10

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

- Functional Changes
- Editorial Changes
- Protocol Extensions
- Discussion Items

Functional Changes: -09 → -10

- PID format
  - Significantly widened character set [0x21, 0x7E]
  - Addresses TODO item of allowing PID names be Base64-encoded data
- Map Version Tag format
  - 64 characters
  - Same character set as PID names [0x21, 0x7E], except '.'
- The “I don't know” costs
  - Cost Maps (and Endpoint Cost Maps) may have missing entries
  - No “authoritativeness” bit (more on that later)
- Add filtering for Address Types in Filtered Network Map
- Add Map Version Tag to Endpoint Property response containing PIDs

Editorial Changes: -09 → -10

- Simplified text for response codes to Information Resource Directory
  - Basically: clients need to follow HTTP 1.1 (RFC2616)
  - Server MAY reply HTTP OPTIONS with Information Resource Directory
  - If server replies with HTTP 300, it SHOULD be an Information Resource Directory
- Cautionary text for Client to check Version Tag in Filtered Cost Map response
- Revised description of numerical costs
  - Replaced “summation” with “normalization” (more accurate use case)

Discussion Items

- Adding constraint operators
  - Operators:
    - {greater,less}-than-or-equal-to
    - equal to
  - Plan to add these in next revision (pending objections/discussion/etc)
- Constraints in Ordinal cost mode
  - Possible interpretations (courtesy of Bill Roome)
    1) Constraints apply to Ordinal numbers
    2) Constraints apply to underlying numerical costs
    3) Constraints only apply to Numerical cost mode

Discussion Item: Non-number Costs

- Use Case
  - String or other structured information with a source-destination pair
  - Better extensibility for Cost Types down the line
- Basic Idea
  - Add a new Cost Mode, and it may be used by registered Cost Types
- Approach 1: Add a Cost Mode called “string”
  - Costs with this mode are JSON String. We don't care what goes in it.
  - Advantages: simple specification, data types independent of ALTO encoding
  - Disadvantages: less efficient
- Add a Cost Mode called “typed” (or similar)
  - Costs with this mode is any JSON Value
  - Advantages: more efficient (reuse same JSON parser)
  - Disadvantages: more complex spec, types and their format bound JSON

Protocol Extensions

- We're starting to draw the line between what goes in the base protocol and what is an extension (yay!)

- On list discussion about possible extensions
  - PID Properties
  - Indication of “authoritativeness”
    - An “I'm guessing” bit, or
    - Continuous value in [0, 1]
  - Maps with Multiple Cost Types
- Split out Redistribution into separate document (extension)
  - Specification may make use of output from JOSE WG