ALTO Incremental Updates

draft-ietf-alto-incr-update-sse-01

W. Roome

Alcatel-Lucent/Bell Labs (NJ)

R. Yang

X. Shi

Yale

IETF 94 (Virtual)

October 27, 2015

Summary

- General framework for updates to <u>any</u> ALTO resource
  - <u>Including</u> Endpoint Cost & Property Services
  - Provides *continuous* updates, which may be full or incremental

- Server defines one or more Update Stream resources
  - Continuous stream of update messages for a resource set
  - One stream may update several resources (network + cost maps)

- Update Streams are optional & flexible
  - Server can offer updates for some, all or none of its resources

- Uses Server-Send Events (SSEs) over HTTP/1.1

- Approved as WG draft (IETF 93, March 2015)

- Test server: http://alto.alcatel-lucent.com:8000/directory

Changes Since Version -00

- Added selective stop
  - More graceful than just closing stream

- Update Stream resource accepts <u>two</u> requests:
  - start-updates establishes a stream, as before:
    ```
    "start-updates": {"network-map":{}, "cost-map":{}}
    ```
  - stop-updates tells server to stop some or all updates:
    ```
    "stop-updates": {"stream-id": "XXXXXXX",
                               "resources": ["cost-map"] }
    ```
  - Note: Version -01 uses different syntax; this is clearer
  - Client sends stop-updates request on a <u>new</u> TCP stream (SSE is one-way stream, server -> client)

- Server assigns unique <u>stream-id</u> for each update stream
  - Server returns as first event in stream
  - Client uses in stop-updates to identify stream

Status & Experience

- SSE issues:
  - SSE designed for events with small amounts of line-oriented text
  - Full cost map could be a 10-megabyte "line"
  - ALTO Server: inject new-lines periodically
  - ALTO Client: use SSE library which returns lines as they arrive
- "Updates" may give same value:
  - Example: When network map changes, easier to send full cost map than determine which costs actually changed
- Incremental updates issues:
  - Can be very difficult for server to decide what changed
  - Ordinal costs: You cannot change just one!
  - ECS via PID costs: new network map -> every cost might change
  - Filtered cost map with constraint test
  - Result: Server may send "unnecessary" full-replacement updates

So What's Next?

- Incremental update is a WG charter item – and it's late!

- Next steps?
  - Update to draft -02 (cosmetic, barring substantive comments)
  - Do off-line interop tests with other clients & servers?
  - Interested parties read the draft?

- Or should we just wait for HTTP/2?
  - Would eliminate stream-ids and associated security issues
  - But we would still use SSE; HTTP/2 "server push" does NOT help
  - Disadvantage: HTTP/2 is much more complicated than HTTP/1.1
  - Library support (client & server) is spotty

Backup Slides

Alternatives

- Use HTTP/2 (RFC 7540) instead of SSE
  - Affects transport; does not change update format
- Pluses:
  - IETF Standard
  - Should be widely supported – *eventually*
- Minuses:
  - Much more complex than HTTP/1.1
  - Not yet widely supported in libraries
  - HTTP proxies must support HTTP/2
- Recommendation:
  - We need incremental update *NOW*
  - Offer HTTP/2 version when support becomes common