

ALTO Incremental Updates Using Server-Sent Events (SSE)

draft-ietf-alto-incr-update-sse-07

W. Roome
Y. Richard Yang

IETF 99
July 20, 2017
Prague

Updates: Overview

- Substantial updates from -05 (Mar. 29, 2017, IETF 98) to -07 (July 3, 2017, IETF 99)
- Main goals of updates
 - Better prepare for document review
 - To handle new use case such as CDNI
- Key types of changes
 - Make more generic statements
 - Make explicit the benefits
 - Restructure the document and make text more friendly
 - Add JSON Patch

More Generic Statements

- Original text tends to focus on maps and the new version makes sure that the design applies to general ALTO information resources

ALTO Incremental Updates Using Server-Sent Events (SSE)
draft-ietf-alto-incr-update-sse-07

Abstract

The Application-Layer Traffic Optimization (ALTO) [RFC7285] protocol provides network related information, called network information resources, to client applications so that clients can make informed decisions in utilizing network resources. For example, an ALTO server can provide network and cost maps so that an ALTO client can use the maps to determine the costs between endpoints when choosing communicating endpoints.

However, the ALTO protocol does not define a mechanism to allow an ALTO client to obtain updates to the information resources, other than by periodically re-fetching them. Because some information resources (e.g., the aforementioned maps) may be large (potentially tens of megabytes), and because only parts of the information resources may change frequently (e.g., only some entries in a cost map), complete re-fetching can be extremely inefficient.

ALTO Incremental Updates Using Server-Sent Events (SSE)
draft-ietf-alto-incr-update-sse-05

Abstract

The Application-Layer Traffic Optimization (ALTO) [RFC7285] protocol provides network related information to client applications so that clients may make informed decisions. To that end, an ALTO Server provides Network and Cost Maps. Using those maps, an ALTO Client can determine the costs between endpoints.

However, the ALTO protocol does not define a mechanism to allow an ALTO client to obtain updates to those maps, other than by periodically re-fetching them. Because the maps may be large (potentially tens of megabytes), and because only parts of the maps may change frequently (especially Cost Maps), that can be extremely inefficient.

Make Explicit the Benefits

- Emphasize two key benefits of incremental updates
 - Immediate (latency reduction)
 - Only incremental changes (traffic volume reduction)

This document presents a mechanism to allow an ALTO server to push updates to ALTO clients, to achieve two benefits: (1) Updates can be immediate, in that the server can send updates as soon as they are available; and (2) Updates are incremental, in that if only a small section of an information resource changes, the server can send just the changes.

Therefore this document presents a mechanism to allow an ALTO Server to provide updates to ALTO Clients. Updates can be both immediate, in that the server can send updates as soon as they are available, and incremental, in that if only a small section of a map changes, the server can send just the changes.

Restructure and Term Changes

2.	Major Changes Since Version -01	4
3.	Terms	5
4.	Background	5
4.1.	Server-Sent Events (SSEs)	5
4.2.	JSON Merge Patch	6
4.2.1.	JSON Merge Patch Encoding	6
4.2.2.	Merge Patch ALTO Messages	7
4.3.	JSON Patch	10
4.3.1.	JSON Patch Encoding	10
4.3.2.	JSON Patch ALTO Messages	10
5.	Overview of Approach	11
6.	Update Stream Messages: Update and Control Events	12
6.1.	ALTO Event Format	12
6.2.	ALTO Update Events	13
6.3.	ALTO Control Events	13
7.	Update Stream	14
7.1.	Media Type	14
7.2.	HTTP Method	14
7.3.	Accept Input Parameters	14
7.4.	Capabilities	16
7.5.	Uses	16
7.6.	Response	17
7.6.1.	Keep-Alive Messages	17
7.6.2.	Event Sequence Requirements	17
7.6.3.	Cross-Stream Consistency Requirements	18
8.	Update Stream Control	18
8.1.	URI	19
8.2.	Media Type	19
8.3.	HTTP Method	19
8.4.	Accept Input Parameters	19
8.5.	Capabilities & Uses	20
8.6.	Response	20
9.	Examples	21
9.1.	Example: Simple Network and Cost Map Updates	21
9.2.	Example: Advanced Network and Cost Map Updates	22
9.3.	Example: Endpoint Property Updates	25
9.4.	IRD Example	28
10.	Client Actions When Receiving Update Messages	30
11.	Design Decisions and Discussions	31
11.1.	HTTP/2 Server-Push	31
11.2.	Not Allowing Stream Restart	32
11.3.	Is Incremental Update Useful for Network Maps?	33
11.4.	Other Incremental Update Message Types	34
11.5.	Requirements on Future ALTO Services to Use this Design	34

2.	Overview of Approach	4
3.	Changes Since Version -01	5
4.	Overview of Server-Sent Events (SSEs)	5
5.	Incremental Update Message Format	7
5.1.	Overview of JSON Merge Patch	7
5.2.	JSON Merge Patch Applied to Network Map Messages	8
5.3.	JSON Merge Patch Applied to Cost Map Messages	10
6.	ALTO Event Stream	11
6.1.	ALTO Event Format	11
6.2.	ALTO Update Events	12
6.3.	ALTO Control Events	12
7.	Update Stream Service	13
7.1.	Media Type	13
7.2.	HTTP Method	13
7.3.	Accept Input Parameters	13
7.4.	Capabilities	15
7.5.	Uses	16
7.6.	Response	16
7.6.1.	Keep-Alive Messages	16
7.6.2.	Event Sequence Requirements	16
7.6.3.	Cross-Stream Consistency Requirements	17
8.	Update Stream Controller	18
8.1.	URI	18
8.2.	Media Type	19
8.3.	HTTP Method	19
8.4.	Accept Input Parameters	19
8.5.	Capabilities & Uses	20
8.6.	Response	20
9.	Examples	20
9.1.	Example: Simple Network and Cost Map Updates	20
9.2.	Example: Advanced Network and Cost Map Updates	22
9.3.	Example: Endpoint Property Updates	24
9.4.	IRD Example	28
10.	Client Actions When Receiving Update Messages	30
11.	Design Decisions and Discussions	31
11.1.	HTTP/2 Server-Push	31
11.2.	Not Allowing Stream Restart	32
11.3.	Is Incremental Update Useful for Network Maps?	33
11.4.	Other Incremental Update Message Types	34

Key Change: Add JSON Patch

- Motivation

- Merge Patch (RFC 7396) is quite elegant in many cases, but cannot handle array edits (can do only full replacement of an array)
- JSON Patch (RFC 6902) defines general operations, in particular manipulation of individual array elements

```
{ "meta" : {  
  "vtag" : {  
    "resource-id" : "my-network-map",  
    "tag" : "da65eca2eb7a10ce8b059740b0b2e3f8eb1d4785"  
  }  
},  
"network-map" : {  
  "PID1" : {  
    "ipv4" : [ "192.0.2.0/24", "198.51.100.0/25" ]  
  },  
  "PID2" : {  
    "ipv4" : [ "198.51.100.128/25" ]  
  },  
  "PID3" : {  
    "ipv4" : [ "0.0.0.0/0" ],  
    "ipv6" : [ "::/0" ]  
  }  
} } }
```

```
[  
  {"op": "replace",  
   "path": "/meta/vtag/tag",  
   "value": "a59740b0b2e3f8eb1acd42231bfe"  
  },  
  {"op": "remove",  
   "path": "/network-map/PID1/ipv4/1"  
  }  
]
```

Impacts on Protocol: IRD

7.4. Capabilities

The capabilities are defined by an object of type `UpdateStreamCapabilities`:

```
object {
  IncrementalUpdateMediaTypes incremental-update-media-types;
} UpdateStreamCapabilities;

object-map {
  ResourceID -> String;
} IncrementalUpdateMediaTypes;
```

If this update stream can provide incremental update events for a resource, the "incremental-update-media-types" field has an entry for that resource-id, and the value is the media-type of the incremental update message. Normally this will be "application/merge-patch+json", "application/json-patch+json", or "application/merge-patch+json,application/json-patch+json", because, as described in Section 6, they are the only incremental update event types defined by this document. However future extensions may define other types of incremental updates.

7.4. Capabilities

The capabilities are defined by an object of type `UpdateStreamCapabilities`:

```
object {
  IncrementalUpdateMediaTypes incremental-update-media-types;
} UpdateStreamCapabilities;

object-map {
  ResourceID -> String;
} IncrementalUpdateMediaTypes;
```

If this Update Stream can provide incremental update events for a resource, the "incremental-update-media-types" field has an entry for that resource-id, and the value is the media-type of the incremental update message. Normally this will be "application/merge-patch+json", because, as described in Section 6, JSON Merge Patch is the only incremental update event type defined by this document.

However future extensions may define other types of incremental updates.

Next Steps

- Main items to be updated
 - Take a full pass and add all terms in the Term section (Section 2)
 - Make a consistent pass on all requirements (MUST, SHOULD, ...)
 - Finish Section 11.5 Requirements on Future ALTO Services to Use this Design
- Will post the final update by Aug. 4
- Issue for WG to think
 - Develop a new patch encoding that can combine the best of Merge Patch and JSON Patch

Backup Slide

- To see the differences
 - <https://tools.ietf.org/rfcdiff>
 - <https://www.ietf.org/id/draft-ietf-alto-incr-update-sse-07.txt>
 - <https://www.ietf.org/archive/id/draft-ietf-alto-incr-update-sse-05.txt>