ALTO Incremental Updates Using Server-Sent Events (SSE)

draft-ietf-alto-incr-update-sse-10

W. Roome
Y. Richard Yang
Dawn Chen

IETF 101
March 19, 2018
London

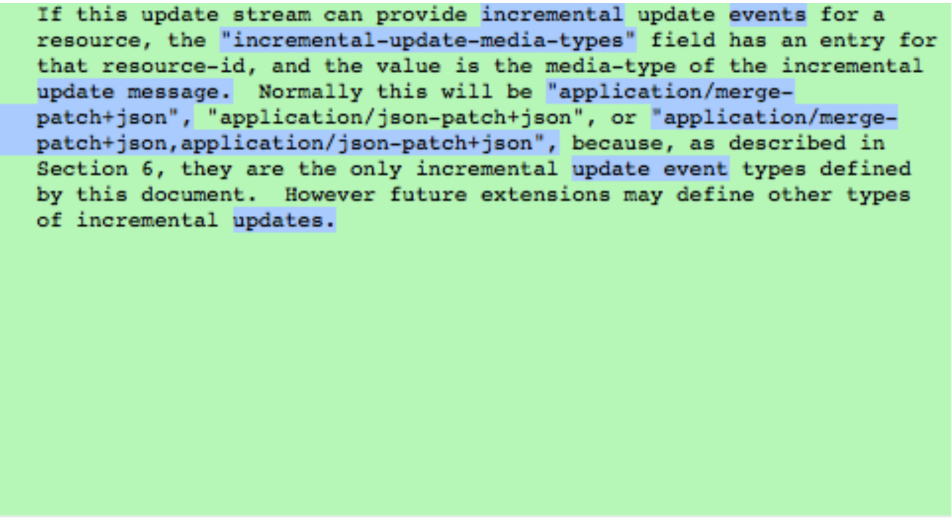
Updates Overview

- Revisions to achieve consistency in
  - Terms:
    - Introduced a new Terms section to define terms precisely and use them consistently
  - Examples:
    - Change to use the same use case across the document
    - Fix small typos in the example, e.g.,
      - add missing vtag field in Section 4.2.2.2 Merge Patch Cost Map Message
  - Section structure:
    - Make the Merge Patch section and JSON Patch section the same
- Clarification on data update choices
- Clarification on requirements
  - “uses”
  - future ALTO services

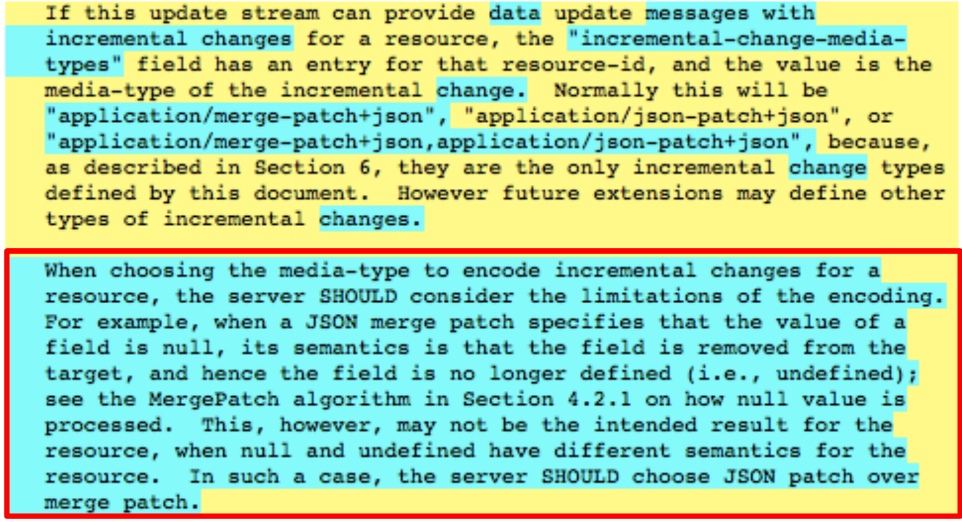
Terms (Consistent Use)

- Update stream
  - data update message
    - full replacement (update message)
    - incremental change (update message)
      - consistent usage: no longer use the term "incremental update"; instead, the document always uses "incremental change"
  - control update message

Clarification on Data Update Choices



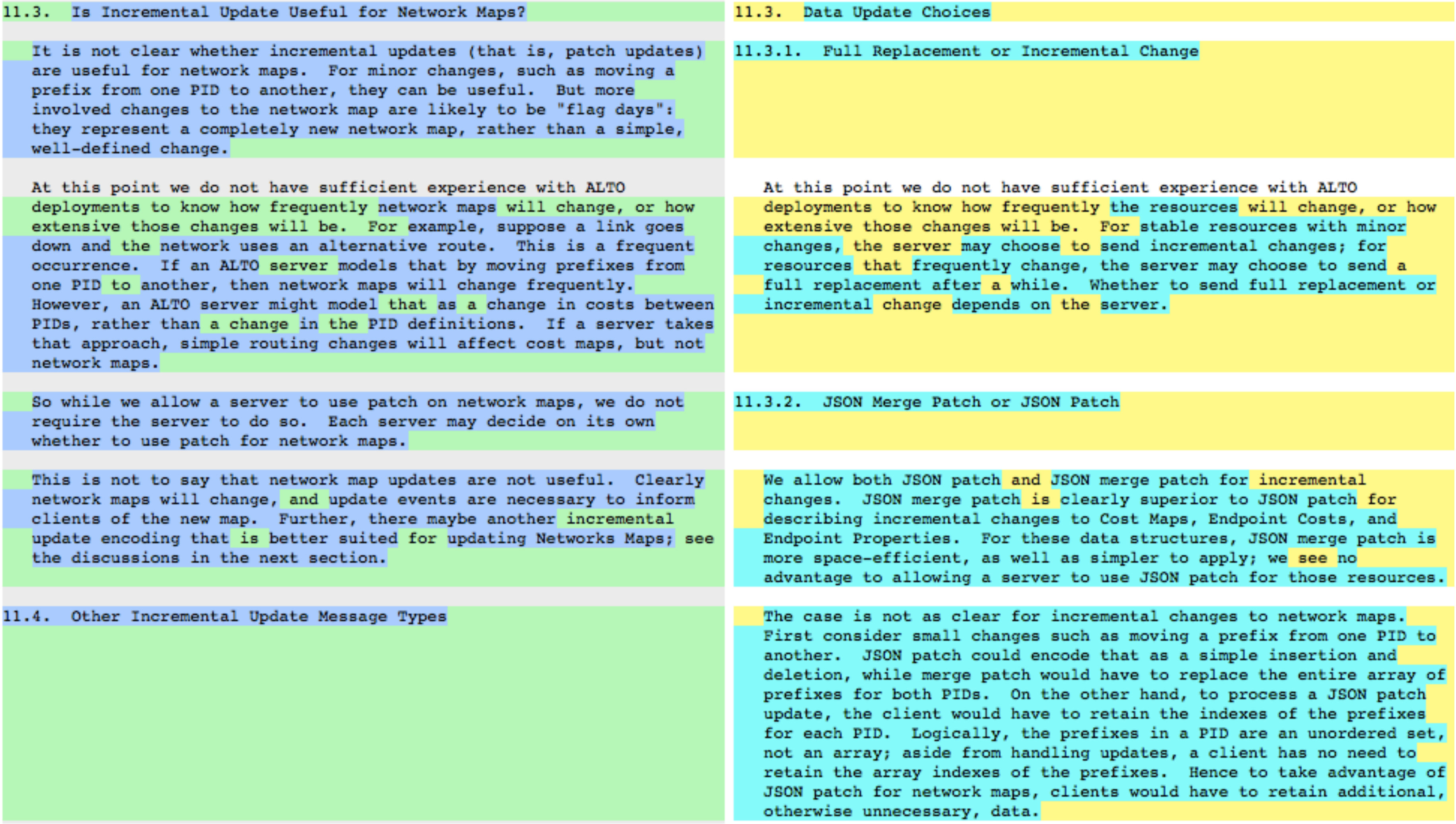
If this update stream can provide incremental update events for a resource, the "incremental-update-media-types" field has an entry for that resource-id, and the value is the media-type of the incremental update message. Normally this will be "application/merge-patch+json", "application/json-patch+json", 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.



If this update stream can provide data update messages with incremental changes for a resource, the "incremental-change-media-types" field has an entry for that resource-id, and the value is the media-type of the incremental change. 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 change types defined by this document. However future extensions may define other types of incremental changes.

When choosing the media-type to encode incremental changes for a resource, the server SHOULD consider the limitations of the encoding. For example, when a JSON merge patch specifies that the value of a field is null, its semantics is that the field is removed from the target, and hence the field is no longer defined (i.e., undefined); see the MergePatch algorithm in Section 4.2.1 on how null value is processed. This, however, may not be the intended result for the resource, when null and undefined have different semantics for the resource. In such a case, the server SHOULD choose JSON patch over merge patch.

Clarification on Data Update Choices



11.3. Is Incremental Update Useful for Network Maps?

It is not clear whether incremental updates (that is, patch updates) are useful for network maps. For minor changes, such as moving a prefix from one PID to another, they can be useful. But more involved changes to the network map are likely to be "flag days": they represent a completely new network map, rather than a simple, well-defined change.

At this point we do not have sufficient experience with ALTO deployments to know how frequently network maps will change, or how extensive those changes will be. For example, suppose a link goes down and the network uses an alternative route. This is a frequent occurrence. If an ALTO server models that by moving prefixes from one PID to another, then network maps will change frequently. However, an ALTO server might model that as a change in costs between PIDs, rather than a change in the PID definitions. If a server takes that approach, simple routing changes will affect cost maps, but not network maps.

So while we allow a server to use patch on network maps, we do not require the server to do so. Each server may decide on its own whether to use patch for network maps.

This is not to say that network map updates are not useful. Clearly network maps will change, and update events are necessary to inform clients of the new map. Further, there maybe another incremental update encoding that is better suited for updating Networks Maps; see the discussions in the next section.

11.4. Other Incremental Update Message Types

11.3. Data Update Choices

11.3.1. Full Replacement or Incremental Change

At this point we do not have sufficient experience with ALTO deployments to know how frequently the resources will change, or how extensive those changes will be. For stable resources with minor changes, the server may choose to send incremental changes; for resources that frequently change, the server may choose to send a full replacement after a while. Whether to send full replacement or incremental change depends on the server.

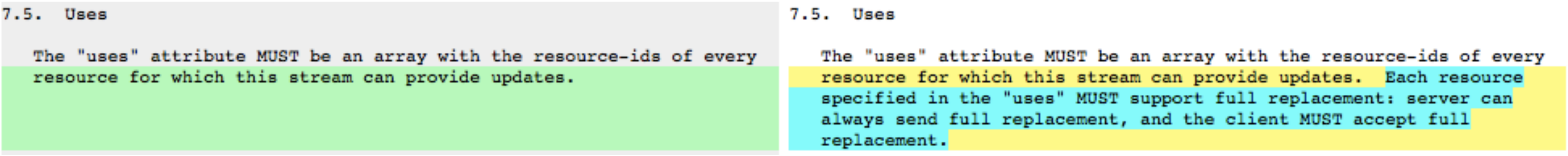
11.3.2. JSON Merge Patch or JSON Patch

We allow both JSON patch and JSON merge patch for incremental changes. JSON merge patch is clearly superior to JSON patch for describing incremental changes to Cost Maps, Endpoint Costs, and Endpoint Properties. For these data structures, JSON merge patch is more space-efficient, as well as simpler to apply; we see no advantage to allowing a server to use JSON patch for those resources.

The case is not as clear for incremental changes to network maps. First consider small changes such as moving a prefix from one PID to another. JSON patch could encode that as a simple insertion and deletion, while merge patch would have to replace the entire array of prefixes for both PIDs. On the other hand, to process a JSON patch update, the client would have to retain the indexes of the prefixes for each PID. Logically, the prefixes in a PID are an unordered set, not an array; aside from handling updates, a client has no need to retain the array indexes of the prefixes. Hence to take advantage of JSON patch for network maps, clients would have to retain additional, otherwise unnecessary, data.

Clarify on Requirements

- On uses



```
7.5.  Uses

   The "uses" attribute MUST be an array with the resource-ids of every
   resource for which this stream can provide updates.
```

```
7.5.  Uses

   The "uses" attribute MUST be an array with the resource-ids of every
   resource for which this stream can provide updates.  Each resource
   specified in the "uses" MUST support full replacement: server can
   always send full replacement, and the client MUST accept full
   replacement.
```

- On requirements on future ALTO services
  - In particular, the key requirements are that (1) each update message is for a single resource; (2) incremental changes can be applied only to a resource that is a single JSON object, as both merge patch and JSON patch can apply only to a single JSON object. Hence, if a future ALTO resource can contain multiple objects, then either each individual object also has a resource-id or an extention to this design is made.

Discussion on a Final Issue: Control Update Design

- Client sends “remove” request to server; response options are:

1. server notifies outcome to the client in the HTTP response by using an HTTP response code.
2. server notifies outcome to the client in an update stream message to indicate the last message.
3. a combination of (1) and (2).

Current draft: 3.

WG decision?

Backup Slides

Specification

- Section 7.3 Accept Input Parameters

```
object {
        String resource-id;
        [String tag;]
        ~~[Boolean incremental-updates;]~~  [Boolean incremental-changes;]
        [Object input;]
} AddUpdateReq;
```

- Section 7.4 Capabilities

```
object {
        ~~IncrementalUpdateMediaTypes incremental-update-media-types;~~
        IncrementalUpdateMediaTypes incremental-change-media-types;
} UpdateStreamCapabilities;
object-map {
        ResourceID -> String;
} IncrementalUpdateMediaTypes;
```