ALTO Extension: Path Vector

draft-ietf-alto-path-vector-10/11

Kai Gao, Young Lee, Sabine Randriamasy, Yang Richard Yang, Jingxuan Zhang

April 21, 2020

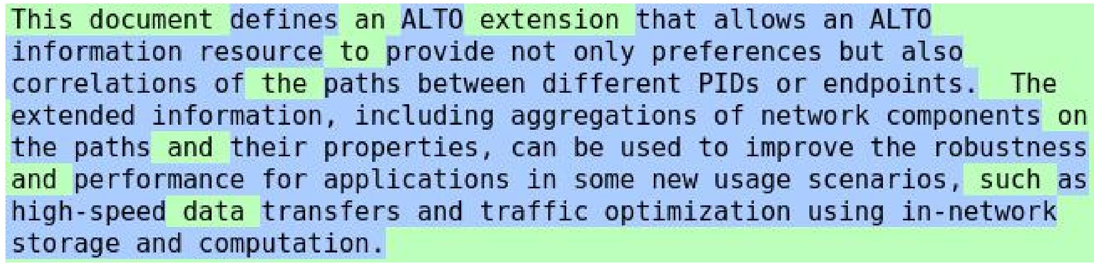
ALTO Interim@IETF 107

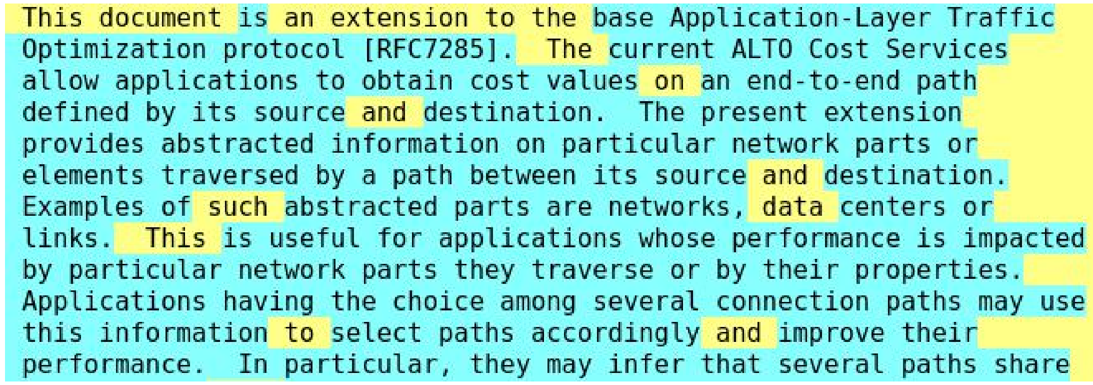
Summary of Changes

- We revise the early parts of the document to clarify
  - the extension provides information on abstract intermediate components of paths between a set of <source, destination> pairs
  - the extension is not limited to a given application (e.g., flow scheduling) but can be applied to multiple use cases
- We revise the specifications to address some potential issues
  - handle “cost-constraints”
  - fix inconsistent naming convention
  - fix spelling errors
  - fix the specification on the multipart response

Issue 1: The Goal of the Extension

- In verstion -09, we use the term "path correlations" as the information conveyed in PV, which limits the scope of the extension and gives the wrong impression that the extension is only for a specific type of applications, e.g., flow scheduling
- Starting from -10, we use "abstract intermediate network parts traversed by a path between …". This definition better clarifies the scope of the extension and also makes the concept of ANE more intuitive
- In -11, we use "components" instead of "parts" as the term "part" is only used in abstract and introduction


This document defines an ALTO extension that allows an ALTO
information resource to provide not only preferences but also
correlations of the paths between different PIDs or endpoints. The
extended information, including aggregations of network components on
the paths and their properties, can be used to improve the robustness
and performance for applications in some new usage scenarios, such as
high-speed data transfers and traffic optimization using in-network
storage and computation.


This document is an extension to the base Application-Layer Traffic
Optimization protocol [RFC7285]. The current ALTO Cost Services
allow applications to obtain cost values on an end-to-end path
defined by its source and destination. The present extension
provides abstracted information on particular network parts or
elements traversed by a path between its source and destination.
Examples of such abstracted parts are networks, data centers or
links. This is useful for applications whose performance is impacted
by particular network parts they traverse or by their properties.
Applications having the choice among several connection paths may use
this information to select paths accordingly and improve their
performance. In particular, they may infer that several paths share

Issue 2: Additional Requirements

- In version -09: We use three use cases but do not explicitly summarize the requirements, which makes it difficult to grasp the ideas and at the same time unnecessarily complex
- In version -10: We bring back the flow scheduling example but the additional requirements are too specific
- In version -11: We first derive the requirements from the flow scheduling example and explicitly summarize the three general additional requirements

(-10)
To allow applications to distinguish the two aforementioned cases, the network needs to provide more details. In particular:

* The network needs to expose more detailed routing information to show the shared bottlenecks.

* The network needs to provide the necessary abstraction to hide the real topology information while providing enough information to applications.

(-11)
AR1: An ALTO server must provide essential information on the intermediate network components on the path of a <source, destination> pair that are critical to the QoE of the overlay application.

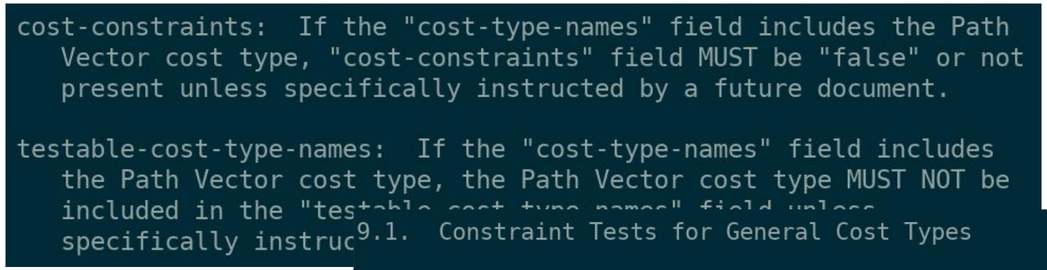
AR2: An ALTO server must provide essential information on how the paths of different <source, destination> pairs share a common network component.

AR3: An ALTO server must provide essential information on the properties associated to the network components.

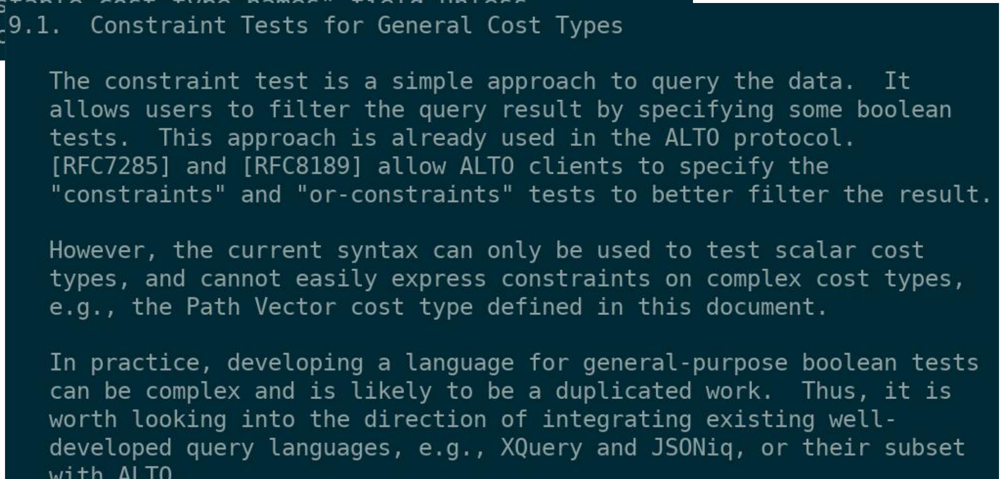
Issue 3: Handling cost constraints

- The same problem is raised to the mailing list on the cost calendar extension
- Values of the fields are explicitly specified in -11

-11

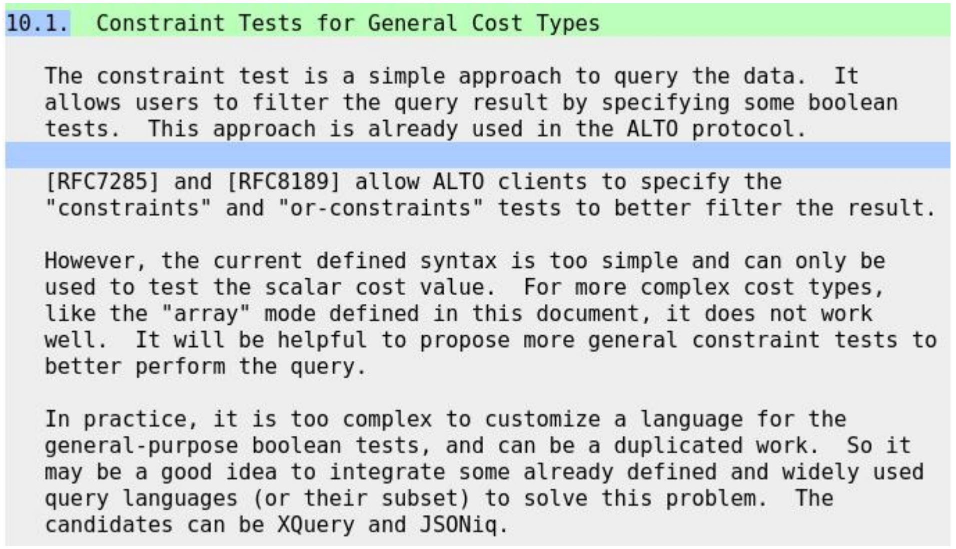


```
cost-constraints:  If the "cost-type-names" field includes the Path
   Vector cost type, "cost-constraints" field MUST be "false" or not
   present unless specifically instructed by a future document.

testable-cost-type-names:  If the "cost-type-names" field includes
   the Path Vector cost type, the Path Vector cost type MUST NOT be
   included in the "testable-cost-type-names" field unless
   specifically instruc
```



```
9.1.  Constraint Tests for General Cost Types

   The constraint test is a simple approach to query the data.  It
   allows users to filter the query result by specifying some boolean
   tests.  This approach is already used in the ALTO protocol.
   [RFC7285] and [RFC8189] allow ALTO clients to specify the
   "constraints" and "or-constraints" tests to better filter the result.

   However, the current syntax can only be used to test scalar cost
   types, and cannot easily express constraints on complex cost types,
   e.g., the Path Vector cost type defined in this document.

   In practice, developing a language for general-purpose boolean tests
   can be complex and is likely to be a duplicated work.  Thus, it is
   worth looking into the direction of integrating existing well-
   developed query languages, e.g., XQuery and JSONiq, or their subset
   with ALTO
```

-09, -10

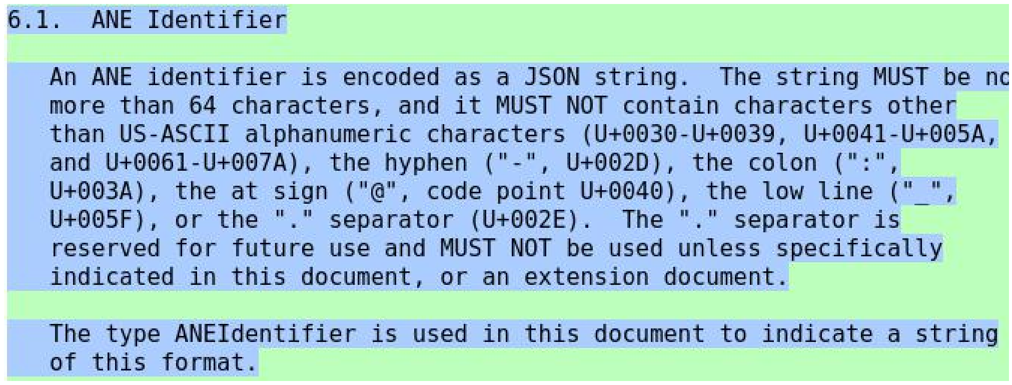


```
10.1.  Constraint Tests for General Cost Types

   The constraint test is a simple approach to query the data.  It
   allows users to filter the query result by specifying some boolean
   tests.  This approach is already used in the ALTO protocol.

   [RFC7285] and [RFC8189] allow ALTO clients to specify the
   "constraints" and "or-constraints" tests to better filter the result.

   However, the current defined syntax is too simple and can only be
   used to test the scalar cost value.  For more complex cost types,
   like the "array" mode defined in this document, it does not work
   well.  It will be helpful to propose more general constraint tests to
   better perform the query.

   In practice, it is too complex to customize a language for the
   general-purpose boolean tests, and can be a duplicated work.  So it
   may be a good idea to integrate some already defined and widely used
   query languages (or their subset) to solve this problem.  The
   candidates can be XQuery and JSONiq.
```

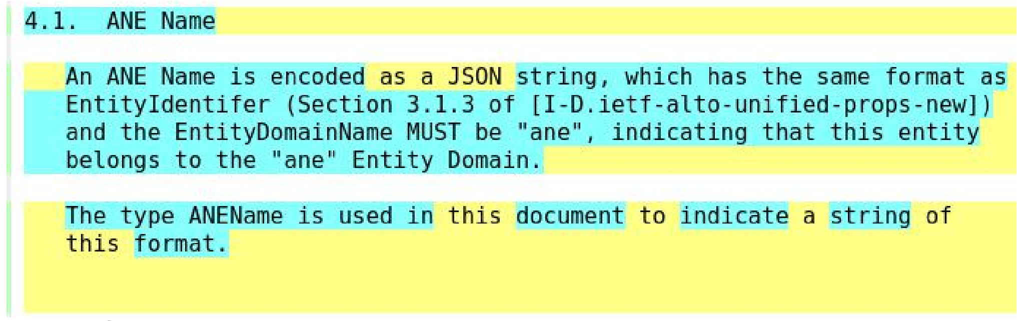
Issue 4: ANE Identifier (-09) --> ANE Name (-10)

- ANEName conveys more semantics
  - ANE is an entity domain type, and the ANE Identifier of an ANE is also the EntityId of the ANE
  - ANE is an aggregation of network components, similar to PID which is the aggregation of endpoints
  - The second seems to convey more information
- ANEIdentifier/ANEId/ANEID are not as easy to parse as ANEName

-09



```
6.1.  ANE Identifier

   An ANE identifier is encoded as a JSON string.  The string MUST be n
   more than 64 characters, and it MUST NOT contain characters other
   than US-ASCII alphanumeric characters (U+0030-U+0039, U+0041-U+005A,
   and U+0061-U+007A), the hyphen ("-", U+002D), the colon (":",
   U+003A), the at sign ("@", code point U+0040), the low line ("_",
   U+005F), or the "." separator (U+002E).  The "." separator is
   reserved for future use and MUST NOT be used unless specifically
   indicated in this document, or an extension document.

   The type ANEIdentifier is used in this document to indicate a string
   of this format.
```



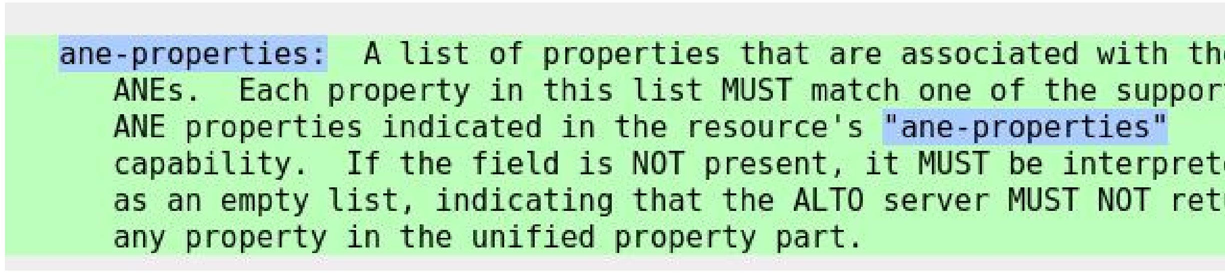
```
4.1.  ANE Name

   An ANE Name is encoded as a JSON string, which has the same format as
   EntityIdentifer (Section 3.1.3 of [I-D.ietf-alto-unified-props-new])
   and the EntityDomainName MUST be "ane", indicating that this entity
   belongs to the "ane" Entity Domain.

   The type ANEName is used in this document to indicate a string of
   this format.
```

-10

Issue 5: ANE Property (-09) --> ANE Property Name (-10)

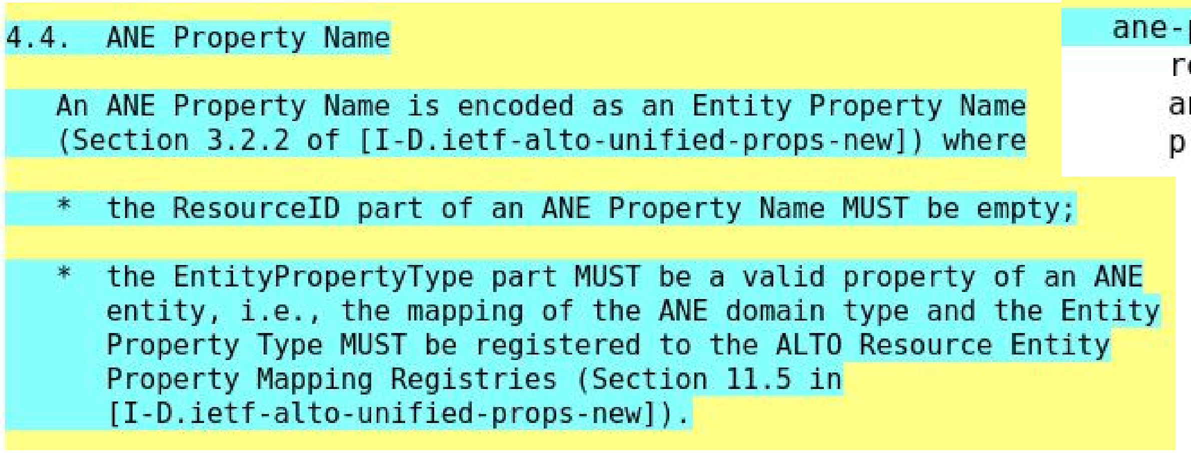
- An ANE Property is a property of the ANE domain, so the ANE Property Name is also the EntityPropertyName of the ANE domain

-09



ane-properties: A list of properties that are associated with th
   ANEs. Each property in this list MUST match one of the suppor
   ANE properties indicated in the resource's "ane-properties"
   capability. If the field is NOT present, it MUST be interpret
   as an empty list, indicating that the ALTO server MUST NOT ret
   any property in the unified property part.

-10

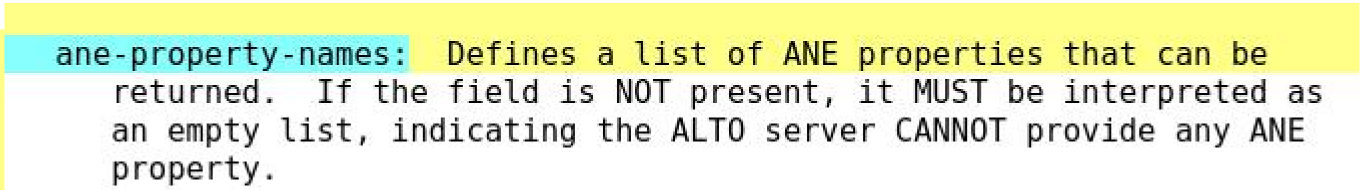


4.4. ANE Property Name

An ANE Property Name is encoded as an Entity Property Name (Section 3.2.2 of [I-D.ietf-alto-unified-props-new]) where

* the ResourceID part of an ANE Property Name MUST be empty;

* the EntityPropertyType part MUST be a valid property of an ANE entity, i.e., the mapping of the ANE domain type and the Entity Property Type MUST be registered to the ALTO Resource Entity Property Mapping Registries (Section 11.5 in [I-D.ietf-alto-unified-props-new]).

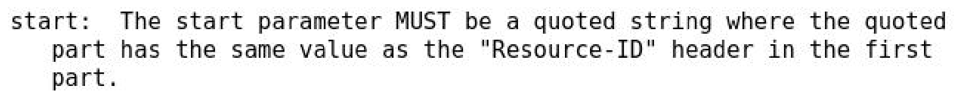


ane-property-names: Defines a list of ANE properties that can be returned. If the field is NOT present, it MUST be interpreted as an empty list, indicating the ALTO server CANNOT provide any ANE property.

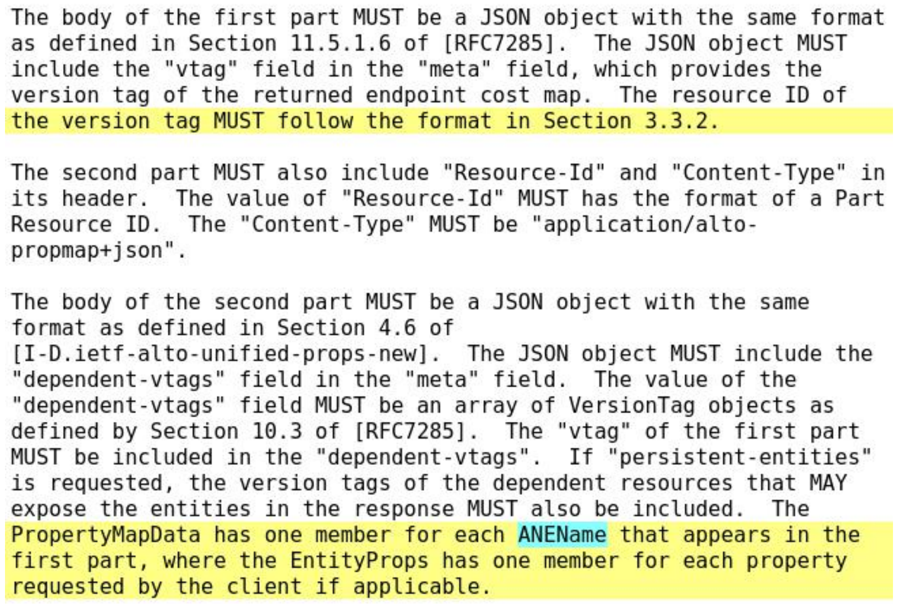
Issue 6: Multipart Message

- Until -10, the specification on the multipart response is contradictory:

-10

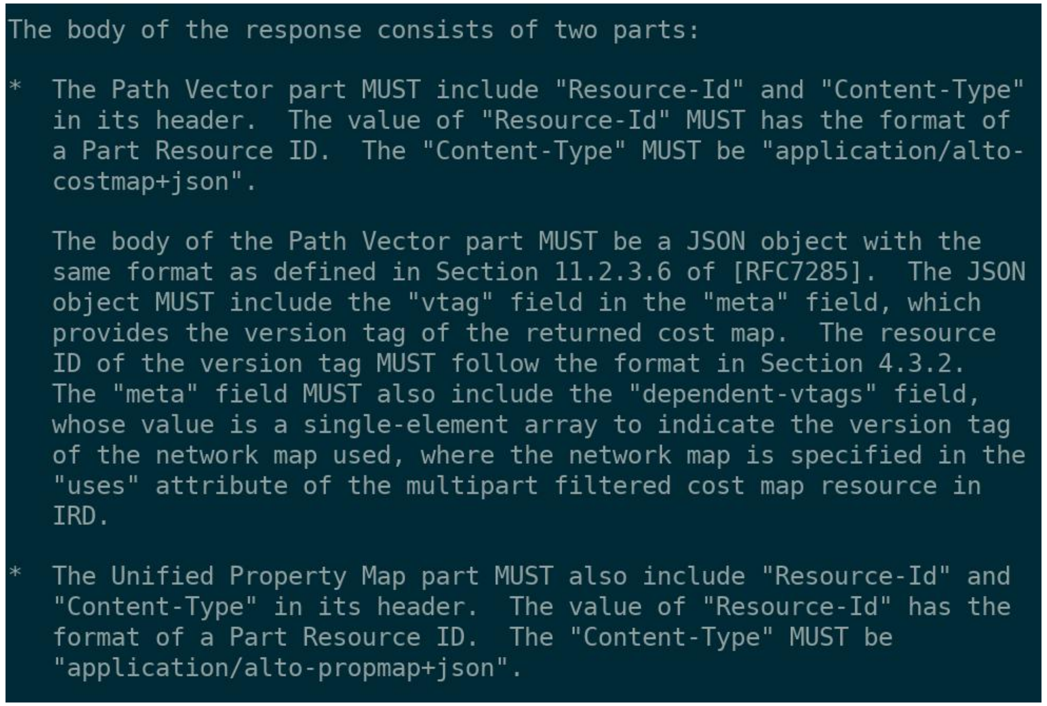


```
start:  The start parameter MUST be a quoted string where the quoted
   part has the same value as the "Resource-ID" header in the first
   part.
```



```
The body of the first part MUST be a JSON object with the same format
as defined in Section 11.5.1.6 of [RFC7285].  The JSON object MUST
include the "vtag" field in the "meta" field, which provides the
version tag of the returned endpoint cost map.  The resource ID of
the version tag MUST follow the format in Section 3.3.2.

The second part MUST also include "Resource-Id" and "Content-Type" in
its header.  The value of "Resource-Id" MUST has the format of a Part
Resource ID.  The "Content-Type" MUST be "application/alto-
propmap+json".

The body of the second part MUST be a JSON object with the same
format as defined in Section 4.6 of
[I-D.ietf-alto-unified-props-new].  The JSON object MUST include the
"dependent-vtags" field in the "meta" field.  The value of the
"dependent-vtags" field MUST be an array of VersionTag objects as
defined by Section 10.3 of [RFC7285].  The "vtag" of the first part
MUST be included in the "dependent-vtags".  If "persistent-entities"
is requested, the version tags of the dependent resources that MAY
expose the entities in the response MUST also be included.  The
PropertyMapData has one member for each ANEName that appears in the
first part, where the EntityProps has one member for each property
requested by the client if applicable.
```

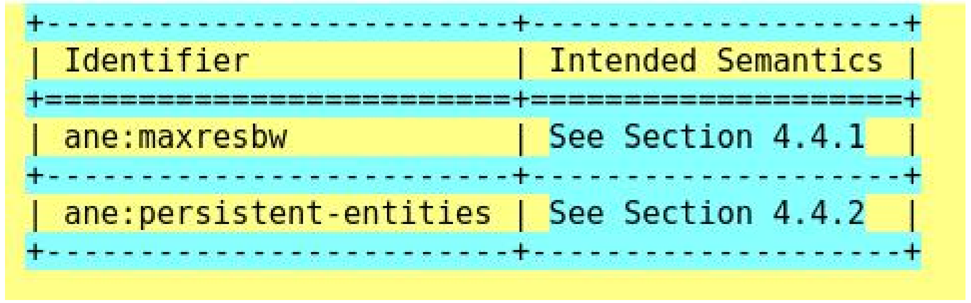
-11



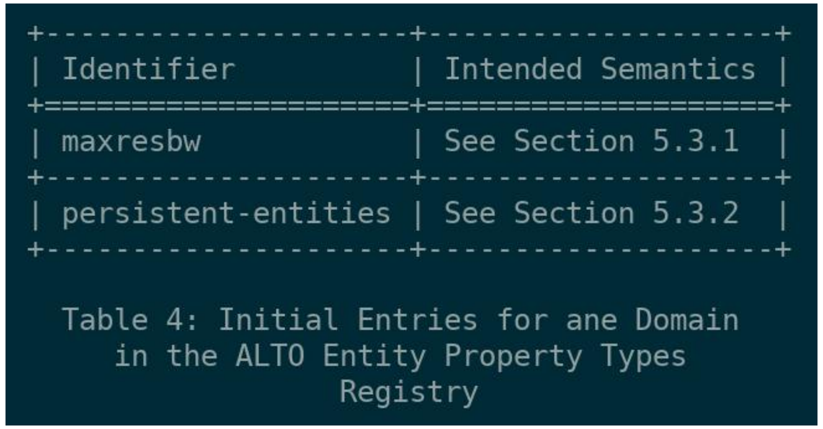
```
The body of the response consists of two parts:

*  The Path Vector part MUST include "Resource-Id" and "Content-Type"
   in its header.  The value of "Resource-Id" MUST has the format of
   a Part Resource ID.  The "Content-Type" MUST be "application/alto-
   costmap+json".

   The body of the Path Vector part MUST be a JSON object with the
   same format as defined in Section 11.2.3.6 of [RFC7285].  The JSON
   object MUST include the "vtag" field in the "meta" field, which
   provides the version tag of the returned cost map.  The resource
   ID of the version tag MUST follow the format in Section 4.3.2.
   The "meta" field MUST also include the "dependent-vtags" field,
   whose value is a single-element array to indicate the version tag
   of the network map used, where the network map is specified in the
   "uses" attribute of the multipart filtered cost map resource in
   IRD.

*  The Unified Property Map part MUST also include "Resource-Id" and
   "Content-Type" in its header.  The value of "Resource-Id" has the
   format of a Part Resource ID.  The "Content-Type" MUST be
   "application/alto-propmap+json".
```

Issue 7: IANA Registry for Property Type

- Until -10, the property types have the “ane:” prefix but in the UP document there is no such prefix
- We remove the prefix in -11



```
+------------------------+-------------------+
| Identifier             | Intended Semantics |
+========================+===================+
| ane:maxresbw           | See Section 4.4.1  |
+------------------------+-------------------+
| ane:persistent-entities | See Section 4.4.2  |
+------------------------+-------------------+
```



```
+--------------------+-------------------+
| Identifier         | Intended Semantics |
+====================+===================+
| maxresbw           | See Section 5.3.1  |
+--------------------+-------------------+
| persistent-entities | See Section 5.3.2  |
+--------------------+-------------------+

  Table 4: Initial Entries for ane Domain
     in the ALTO Entity Property Types
                 Registry
```

Next Step

- Finish WGLC and go to next stage?