

A Unified Approach for ALTO Properties

draft-roome-alto-unified-props-00

Wendy Roome
Alcatel-Lucent/Bell Labs (NJ)

IETF 93
July 21, 2015

The Problem

- In the beginning there were Endpoint Properties (EPs)
- Some properties are server-specific
 - Bandwidth, location,
- But other properties really apply to CIDRs – endpoint sets:
 - ASN, ISP,
- Endpoint *inherit* properties from CIDRs
- Other other entities may have properties:
 - PIDs, Abstract Network Element Properties (topology draft),

Let's unify those Property Services
into a common extensible framework
that can handle new entity domains

Another Problem

- ALTO's Endpoint Property Service is POST-mode only
 - Client cannot get a full property map
 - Unlike network & cost maps
- Made sense for endpoints: a full map would be enormous
- But what if only a small set of endpoints have properties?
- Or if properties are defined on CIDRs, and inherited?
- Or if the entity domain is small?

As with Network & Cost Maps, we need
full (GET-mode) & filtered (POST-mode)
Property Map resources

Entity Naming

- Every entity has a unique name:

entity-name := domain-name : domain-specific-name

domain-name := ipv4 | ipv6 | mac48 | pid | ane |

- Domain-specific names can be hierarchical

- Examples:

ipv4:1.2.3.4

ipv4:1.2.0.0/16

pid:mypid1

ane:datacenter-14.rack-37.rack-router

ane:datacenter-14

Property Naming

- Common property name space, independent of entity domain
 - Same value format for all domains
 - Interpretation may vary, but basic meaning stays the same
 - If a property does not make sense for an entity domain, skip it!
- Good example:
 - geo-location property is “latitude longitude [height]”
 - For PIDs, it’s the centroid of endpoints in PID
- Bad example:
 - For endpoints, geo-location is “lat long [height]”
 - For PIDs, geo-location is “nw-lat nw-long se-lat se-long”

Property Maps & Network Maps

- In RFC 7285, Endpoint Properties were independent of Network Maps
 - Holdover from early single Network Map versions of the protocol
 - Illusion, because the “pid” property depends on the Network Map
 - Led to “resource-specific property” kludge (mea culpa!)
- Complication: Some entity domains (e.g., ANEs) are defined in the context of a Network Map resource
- Conceptual change in the Network Map dependency:
 - Property names are NOT qualified by a Network Map resource id
 - Instead, as with Cost Maps, a Property Map *resource* depends on a Network Map
 - All properties & entities in the Property Map depend on that Network Map
 - Indicated by the “uses” attribute of the Property Map resource

Internet Address Domain

- Domain names: ipv4 & ipv6
- Entities can be prefixes (CIDRs) as well as endpoints
 - E.g: ipv4:1.2.3.4 ipv4:1.2.0.0/16 ipv4:1.0.0.0/8
- Endpoints *inherit* properties from the longest matching CIDR
- CIDRs can also inherit properties
- There are several separate spaces of internet address entities:
 - One for each Network Map resource
 - One unnamed space, not associated with a Network Map
 - A Property Map which “uses” a Network Map returns properties from that map’s space
 - A Property Map with an empty “uses” attribute returns properties from the unnamed space

Property Map Services

- Two new services, modeled on Full & Filtered Network Maps:
 - GET-mode Full Property Map
 - POST-mode Filtered Property Map
- IRD entry gives property names and entity domains in that Property Map
 - Implicit cross product of entity domains & property names
 - Server omits meaningless combinations
 - Server can define multiple maps to avoid meaningless combinations
- A Full Property Map for Endpoint Properties???
 - Yes, there are billions of endpoints
 - But the server might define properties only for a few thousand
 - And CIDRs may have properties, which endpoints inherit
 - If a full map is too big, don't define the resource

IRD Entries: Full Property Maps

```
"full-property-1" : {  
  "uri" : "http://-----",  
  "media-type" : "application/alto-propmap+json", (new type)  
  "uses" : [ "my-default-network-map" ],  
  "capabilities" : {  
    "prop-types" : [ "geo-location", "asn" ],  
    "domain-types" : [ "ipv4", "ipv6" ]  
  }  
},  
"full-property-2" : {  
  "uri" : "http://-----",  
  "media-type" : "application/alto-propmap+json",  
  "uses" : [ "my-default-network-map" ],  
  "capabilities" : {  
    "prop-types" : [ "bandwidth", "type" ],  
    "domain-types" : [ "ane" ]  
  }  
}
```

IRD Entries: Filtered Property Maps

```
"filtered-property-1" : {
  "uri" : "http://-----",
  "media-type" : "application/alto-propmap+json",
  "accepts" : "application/alto-propmapfilter+json", (new type)
  "uses" : [ "my-default-network-map" ],
  "capabilities" : {
    "prop-types" : [ "pid", "location", "asn" ]
    "domain-types" : [ "ipv4", "ipv6" ]
  },
},
"filtered-property-2" : {
  "uri" : "http://-----",
  "media-type" : "application/alto-propmap+json",
  "accepts" : "application/alto-propmapfilter+json",
  "uses" : [ "my-default-network-map" ],
  "capabilities" : {
    "prop-types" : [ "bandwidth", "type" ]
    "domain-types" : [ "ane" ]
  }
}
```

Filtered Request

Client gives property names & entity names:

```
POST /----- HTTP/1.1
Host: alto.example.com
Content-Length: ###
Content-Type: application/alto-propmapfilter+json
Accept: application/alto-propmap+json,application/alto-error+json
```

```
{
  "properties" : [ "geo-location", "asn" ],
  "entities" : [ "ipv4:1.2.3.4" ]
}
```

Response

Similar to current Endpoint Property service:

```
HTTP/1.1 200 OK
```

```
Content-Length: ###
```

```
Content-Type: application/alto-propmap+json
```

```
{  
  "property-map": {  
    "ipv4:1.2.3.4" : { "geo-location": "40.1205,-74.2519",  
                     "asn": 65000 }  
  }  
}
```

Effect On Current Documents

RFC 7285:

- Deprecate the current Endpoint Property Service
- Do not define any new resource-specific properties
 - But keep “resource-id.pid” for legacy clients

PID Properties Draft:

- Drop; the Property Map draft defines a simpler version
- Inheritance happens via CIDR properties, not PID properties

New Properties Drafts:

- Define the entity domains for those properties

What Next?

- Do you like this approach?