A Unified Approach for ALTO Properties

(no draft yet)

Wendy Roome
Alcatel-Lucent/Bell Labs (NJ)

IETF 92
March 26, 2015

Motivation

- In the beginning there were Endpoint Properties (EPs).
- EPs were independent of the Network Map, but there was only one Network Map, so it was moot.
- And then we added multiple Network Maps and "resource-specific" EPs, and EPs became more complicated.
- And then we proposed PID Properties.
- And Abstract Network Element Properties (topology draft).
- And Foo Properties, and Bar Properties, and ….

Let's unify all those Property Services into a common framework that can be extended for new entity classes

Entity Naming

- Extend typed endpoint addresses:

  entity-name := *entity-class* : *entity-specific-name*

  entity-class := ipv4 | ipv6
  cidrv4 | cidrv6 |
  mac48 |
  pid |
  ane | ….

- Examples:

```
ipv4:1.2.3.4
cidrv4:1.2.0.0/16
pid:mypid1
ane:link42
ane:datacenter-14.rack-37.rack-router
```

Property Naming

- Common property name space, independent of entity type
  - Same value format for all entity types
  - Interpretation may vary, but basic meaning stays the same
  - If a property does not make sense for an entity type, 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"
- Only applies to IANA registered properties. For "priv:" properties, do whatever you want.

Property Map Services

- Two new services, modeled on Full & Filtered Network Maps:
  - GET-mode Full Property Map
  - POST-mode Filtered Property Map
- IRD gives property names and entity types each map returns
  - Implicit cross product of entity types & 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
  - If a Full Map is too big, don't define the resource

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!)
- Conceptual change:

  Each Property Map resource depends on a Network Map

- Many entity types are defined by the Network Map, so this provides necessary context
- Use the default Network Map for any properties that really are independent of the network

IRD Entries: Full Property Maps

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

(new type)

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" ]
      "entity-types" : [ "ipv4", "ipv6", "pid" ]
    },
 },
"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" ]
      "entity-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", "pid:mypid2" ]
}
```

Response

Similar to current Endpoint Property service:

```
HTTP/1.1 200 OK
Content-Length: ###
Content-Type: application/alto-propmap+json
{
  "meta" : {
    "dependent-vtags" : [
      {"resource-id": "my-default-network-map",
       "tag": "7915dc0290c2705481c491a2b4ffbec482b3cf62"
      }
    ]
  },
  "property-map": {
    "ipv4:1.2.3.4" : { "geo-location": "40.1205,-74.2519",
                       "asn": 65000 }
    "pid:mypid2" :   { "geo-location": "40.0,-74.0",
                       "asn": 65000 }
  }
```

ALTO Properties Simplify Access To …

DNS:

- Properties for (say) "dns:ietf.org":
  - "address" is preferred address
  - "addresses" is list of alternate addresses
  - Properties for the various DNS resource records?
  - Resolved at ALTO server

WHOIS:

- Properties for (say) "whois:ietf.org":
  - "registrant", "admin" and "tech" could be JSON dictionaries
  - "name-servers" could be list of registered name servers

Effect On Current Documents

RFC 7285:

- Deprecate the current Endpoint Property Service
- Do not define any new resource-specific properties

PID Properties Draft:

- Revise to use this Property Map service
- Define the “pid” and “cidr” entity types
- Define inheritance between pids, cidrs and endpoints

New Properties Drafts:

- Define the entity types for those properties

What Next?

- Do you like this approach?
- If so, write draft & circulate via mailing list