

ALTO WG  
Internet-Draft  
Intended status: Standards Track  
Expires: January 4, 2015

W. Roome  
Alcatel-Lucent  
Y. Yang  
Yale  
July 3, 2014

PID Property Extension for ALTO Protocol  
draft-roome-alto-pid-properties-02

Abstract

This document extends the Application-Layer Traffic Optimization (ALTO) Protocol [I-D.ietf-alto-protocol] by defining PID-based properties in much the same way that the original ALTO Protocol defines endpoint-based properties.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 4, 2015.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. The Consistency and Inheritance Design Views . . . . .	3
3. A Hierarchical View of a Network Map . . . . .	3
3.1. Default Containment Hierarchy . . . . .	3
3.2. Extension: Implicit Inheritance Via Nested PIDs . . . . .	4
4. Services . . . . .	5
4.1. PID Properties Announcement . . . . .	5
4.2. Full PID Property Map Service . . . . .	6
4.3. Filtered PID Property Map Service . . . . .	7
4.4. Endpoint Property Service . . . . .	7
5. Security Considerations . . . . .	7
6. IANA Considerations . . . . .	8
7. References . . . . .	8
Authors' Addresses . . . . .	8

## 1. Introduction

A key abstraction introduced by the ALTO Protocol [I-D.ietf-alto-protocol] is PIDs (Provider-defined Identifiers), where each PID is defined as a name and a set of associated endpoint addresses. For IPv4/IPv6 networks, a PID's address set is defined by one or more endpoint address prefixes called CIDRs [RFC.4632]. This extension focuses on IPv4/IPv6 networks.

An ALTO Server uses PIDs when defining one or more Network Maps, each of which is defined by a set of PIDs. Each Network Map defines a logical partition of a network address space, where similar endpoints are grouped in the same PID, specified by the addresses contained in the definition of the PID. An ALTO Server may publish multiple Network Maps when there are multiple ways to partition networks. For example, one Network Map may partition endpoints according to geographical locations, and hence each PID defined in the Network Map represents the set of endpoints at a given location. Another Network Map may partition endpoints according to the capabilities (e.g., CDN delivery protocols such as HTTP or HTTPS) that the network can provide. In this case, each PID defined in the Network Map represents the endpoints with similar capabilities.

A major missing component of the base ALTO Protocol is that the common properties are not specified. In particular, in the base ALTO Protocol, each PID has only a name and a set of endpoint addresses. The objective of this document is to allow PIDs to have properties. Example PID properties include "country code", "continent code", "ISP", "lat/long bounding box", "endpoint type" (server farm, end users, cell data connections, etc). We identify use cases (e.g., VPN

selection and CDN Capability Advertisement) where PID properties can provide value.

## 2. The Consistency and Inheritance Design Views

When we define PID properties, we follow a key consistency design guideline that PID properties should be consistent with and generalize the endpoint properties already defined in the base ALTO Protocol. Specifically, in the base ALTO Protocol, for each selected endpoint address, there can be a set of (prop-type, value) pairs associated with the endpoint address. These are called the endpoint properties of the selected endpoint. The ALTO Protocol allows an ALTO Client to obtain defined endpoint properties.

Consider a given endpoint property  $p$  and all endpoints defined in a PID named  $pid1$ . If all of the endpoints have the same value  $v$  for  $p$ , then it is natural and consistent that when we define the value for  $p$ , as a PID property, the value should be  $v$ . For the more general case, let  $ip1.p$  denote the value of property  $p$  for endpoint  $ip1$ . Assume that  $pid1$  consists of a set of  $n$  IP addresses,  $ip1, ip2, \dots, ipn$ . Let  $pid1.p$  denote the value of property  $p$  for  $pid1$ . Then we can consider that  $pid1.p$  is from an aggregation function of  $ip1.p, ip2.p, \dots, ipn.p$ . Example aggregation functions include average/mean, mode, geo-center, union, bounding box, where meaningful aggregations depend on the specific property  $p$ .

Complementing the bottom-up aggregation view, we also adopt a top-down inheritance view, by considering that when  $ip1$  is in  $pid1$ ,  $ip1.p$  inherits the value of  $pid1.p$ , if the value of  $ip1.p$  is not defined; otherwise,  $ip1.p$  overrides the value of  $pid1.p$ . The concept of inheritance is a simple, but powerful concept to reduce information redundancy.

## 3. A Hierarchical View of a Network Map

### 3.1. Default Containment Hierarchy

A Network Map defined in the base ALTO Protocol can be considered as a default three-level hierarchy: with the highest (1st) level being a root, the next (2nd) level being the PIDs, and the lowest (3rd or leaf) level being the individual endpoint addresses. An issue that the base ALTO Protocol needs to resolve is that PID definitions can overlap, and hence we must determine the PID to which an endpoint address belongs. For example, consider a Network Map with two PIDs:  $PID1$  is  $10.0.0.0/8$ , and  $PID2$  is  $10.0.1.0/24$ . Then all addresses in  $PID2$  are also in  $PID1$ . The base ALTO Protocol requires that an endpoint address be in one, and only one, PID, among the set of PIDs defined in the same Network Map. ALTO achieves this by specifying

that if an address matches several CIDR, the address is in the PID with the CIDR with the longest prefix. We refer to this PID as the home PID of the endpoint. Thus, for the example, 10.0.1.5 is in PID2, and 10.0.2.6 in in PID1.

### 3.2. Extension: Implicit Inheritance Via Nested PIDs

We would like to use the PID hierarchy to inherit property values. That is, if all endpoints in *px*, *py* and *pz* are also in *pa*, then unless otherwise overridden, PIDs *px*, *py* and *pz* should inherit all properties defined in PID *pa*.

Unfortunately overlapping PID definitions result in the usual issues with multiple inheritance. Consider the following example:

```
PID p1: [1.0.0.0/8]
PID p2a: [1.0.0.0/16]
PID p2b: [1.1.0.0/16]
PID p3: [1.0.255.0/24, 1.1.0.0/24]
```

All endpoints in *p2a* and *p2b* are also in *p1*, so those two PIDs should inherit any properties defined in *p1*. However, the endpoints in *p3* are split between *p2a* and *p2b*, so *p3* cannot simply inherit values from *p2a* or *p2b*. On the other hand, all endpoints in *p3* are in *p1*, so we would expect *p3* to inherit any properties defined in *p1* that are not overridden in *p2a* or *p2b*.

Hence we will define inheritance as follows.

Definition: The immediate parent of CIDR *C* is the CIDR *C'* with the longest prefix of all CIDRs, in the set of all CIDRs in all PIDs in the Network Map, which contain all endpoints in *C*. The immediate parent CIDR might not exist, but if it does, it is unique.

Definition: A CIDR *C* inherits the value *V* for property *PR* if the PID containing its immediate parent CIDR *C'* defines the value *V* for property *PR*, or if its immediate parent CIDR *C'* inherits the value *V* for property *P*.

Definition: A PID *P* has the value *V* for property *PR* if that value is explicitly defined for *PR* in *P*, or if all CIDRs *C* in *P* inherit the same value *V* for property *PR*.

Suppose the following properties are defined for PIDs described above:

```
PID p1:  ISP="Verizon" country-code="us"  
PID p2a: state-code="NJ"  
PID p2b: state-code="NY"
```

Then p2a, p2b, and p3 would all inherit the ISP and country-code properties from p1. However, p3 would not inherit the state-code property, because it has different values in p2a and p2b.

#### 4. Services

In the interests of simplicity, we will give an overview of the proposed services, rather than detailed descriptions.

##### 4.1. PID Properties Announcement

Given the consistency and inheritance design guideline, we require that PID Properties and Endpoint Properties use the same property name space. Such property names must be registered with IANA.

To allow an ALTO Client to know the set of PID Properties associated with a PID Property Resource, we use the same approach as that of endpoint properties: announcement in IRD. An example is shown below.

```

...
"resources" : {
  "my-default-network-map" : {
    "uri" : "http://alto.example.com/networkmap",
    "media-type" : "application/alto-networkmap+json"
  },
  "endpoint-property" : {
    "uri" : "http://alto.example.com/endpointprop/lookup",
    "media-type" : "application/alto-endpointprop+json",
    "accepts" : "application/alto-endpointpropparams+json",
    "capabilities" : {
      "prop-types" : [ "my-default-network-map.pid",
                       "priv:ietf-example-prop" ]
    },
  },
  "my-pid-property" : {
    "uri" : "http://alto.example.com/pidprop/netmap1/pidp1",
    "media-type" : "application/alto-pidprop+json",
    "uses" : [ "my-default-network-map" ]
    "capabilities" : {
      "prop-types" : [ "country-code",
                       "asn" ]
    },
  },
}
}

```

#### 4.2. Full PID Property Map Service

Analogous to ALTO's Full Cost Map Service, a Full PID Map Service returns properties defined for all PIDs in a Network Map.

This is a GET request. The response message is similar to that of ALTO's Endpoint Property Service, but with PID names instead of endpoint addresses. The IRD entry for the service defines a "prop-types" capability with the names of the properties that this service returns, and specifies a "uses" attribute for the Network Map defining the PIDs.

In the interests of limiting the response message size, the Full PID Property Map Service would NOT enumerate inherited property values. Thus if PID1 defines PROP1, and if PID2 is contained within PID1 and does not override the value for PROP1, then the response message gives a value for PROP1 in PID1, but not in PID2. In this case the client is expected to deduce the inheritance. That is feasible because the client has all information needed to do that.

#### 4.3. Filtered PID Property Map Service

Analogous to ALTO's Filtered Cost Map Service, a Filtered PID Map Service returns a subset of the Full PID Property Map. The client specifies the desired property and PID names.

This is a POST request. The response message is the same as for the Full PID Property Map Service. The request message is similar to the request message for ALTO's Endpoint Property Service, except with PID names instead of endpoint addresses. The IRD entry for the service defines a "prop-types" capability with the names of the properties this service returns, and specifies a "uses" attribute for the Network Map defining the PIDs.

Unlike the Full Filtered PID Property Service, the Filtered PID Property Service would explicitly enumerate inherited property values. Thus if PID1 defines PROP1, and if PID2 is contained within PID1 and does not override the value for PROP1, then the response message includes PID1's value for PROP1 in PID2's properties. This is necessary because the Filtered PID Property Map response does not give the client enough information to deduce the inherited properties. For consistency, the Filtered PID Property Service would enumerate inherited properties for a PID even if the client also requested properties for all PIDs that containing that PID.

#### 4.4. Endpoint Property Service

As described in Section 10.8 of the ALTO protocol specification, endpoint property names may be prefixed with the Resource ID of a Network Map. For such resource-specific properties, if a value is not explicitly defined for an endpoint, the Endpoint Cost Service MUST return the value that the Filtered PID Property Map Service would return for the PID containing that endpoint.

For properties that are not prefixed by a Network Map Resource ID, if a value is not defined for an endpoint, the Endpoint Property Service MAY return the value defined for that property in one of the ALTO Server's PID Property Maps for the PID containing the endpoint.

#### 5. Security Considerations

Some properties may have sensitive customer-specific information. If this is the case, an ALTO Server may limit access to those properties by providing several different PID property services. For non-sensitive properties, the ALTO Server would provide a uri which accepts requests from any client. Sensitive properties, on the other hand, would only be available via a secure uri which would require client authentication.

## 6. IANA Considerations

No actions are required from IANA as result of the publication of this document.

## 7. References

[I-D.ietf-alto-protocol]

Almi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-20 (work in progress), October 2013.

[RFC.4632]

Fuller, V. and T. Li, "Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan", RFC 4632, BCP 122, August 2006.

## Authors' Addresses

Wendy Roome  
Alcatel-Lucent/Bell Labs  
600 Mountain Ave, Rm 3B-324  
Murray Hill, NJ 07974  
USA

Phone: +1-908-582-7974  
Email: w.roome@alcatel-lucent.com

Y. Richard Yang  
Yale University  
51 Prospect St.  
New Haven, CT  
USA

Email: yry@cs.yale.edu