Charter Update

Where We Are

- Base ALTO protocol finally done
- Extensions have been proposed for the last two years (and postponed till now)
  - Protocol optimizations
  - New usages / use cases
- Extensive discussion since IETF88
  - Identified four tentative new items
  - (Reasonably) achievable in a short time

At a Glance

1. Protocol optimizations
   - » draft-roome-alto-incr-updates
   - » draft-marocco-alto-ws
   - » draft-alto-caching-subscription
2. Server discovery extensions
   - » draft-kiesel-alto-ip-based-srv-disc
   - » draft-kist-alto-3pdisc
   - » draft-kiesel-alto-3pdisc-impl
3. Endpoint property and e2e cost extensions
   - » draft-roome-alto-pid-properties
   - » draft-seedorf-cdni-fci-alto
   - » draft-scharf-alto-vpn-service
   - » draft-seedorf-lmap-alto
   - » draft-wu-alto-te-metrics
   - » draft-randriamasy-alto-cost-schedule
   - » draft-randiamasy-alto-multi-cost
4. Graph representation extensions
   - » draft-bernstein-alto-topo
   - » draft-yang-alto-topology

Protocol Optimizations

Protocol extensions for reducing the volume of on-the-wire data exchange required to align the ALTO server and clients.

Extensions under consideration are mechanisms for delivering server-initiated notifications and partial updates of maps.

Efforts developed in other working groups such as Websockets and JSON-patch will be considered, as well as bespoke mechanisms specific to the ALTO protocol.

Server Discovery Extensions

Extensions to the base ALTO server discovery mechanism (RFC-to-be) for deployment in heterogeneous network environments.

Mechanisms under consideration are extensions for third-party and anycast-based server discovery.

Endpoint Property and e2e Cost Extensions

Protocol extensions to convey a richer set of attributes to allow applications to determine not only "where" to connect but also "when" to connect.

Such additional information will be related both to endpoints (e.g. conveying server load and cache geo-location information for CDN use cases) and to endpoint-to-endpoint costs (e.g. bandwidth calendaring to represent time-averaged cost values in datacenter networks).

The working group will specify such extension in coordination with other working groups that are also working on the related use cases (e.g. cdni, i2rs, lmap).

Graph Representation Extensions

A survey of techniques to formalize the structure of a network graph (that can derived from a set of related ALTO network and cost maps) in a format that would facilitate advanced graph computation.

Such survey will cover both models used in popular open-source software (e.g. NetworkX, tinkerpop blueprints) and models being considered in other working groups (e.g. netmod, i2rs).

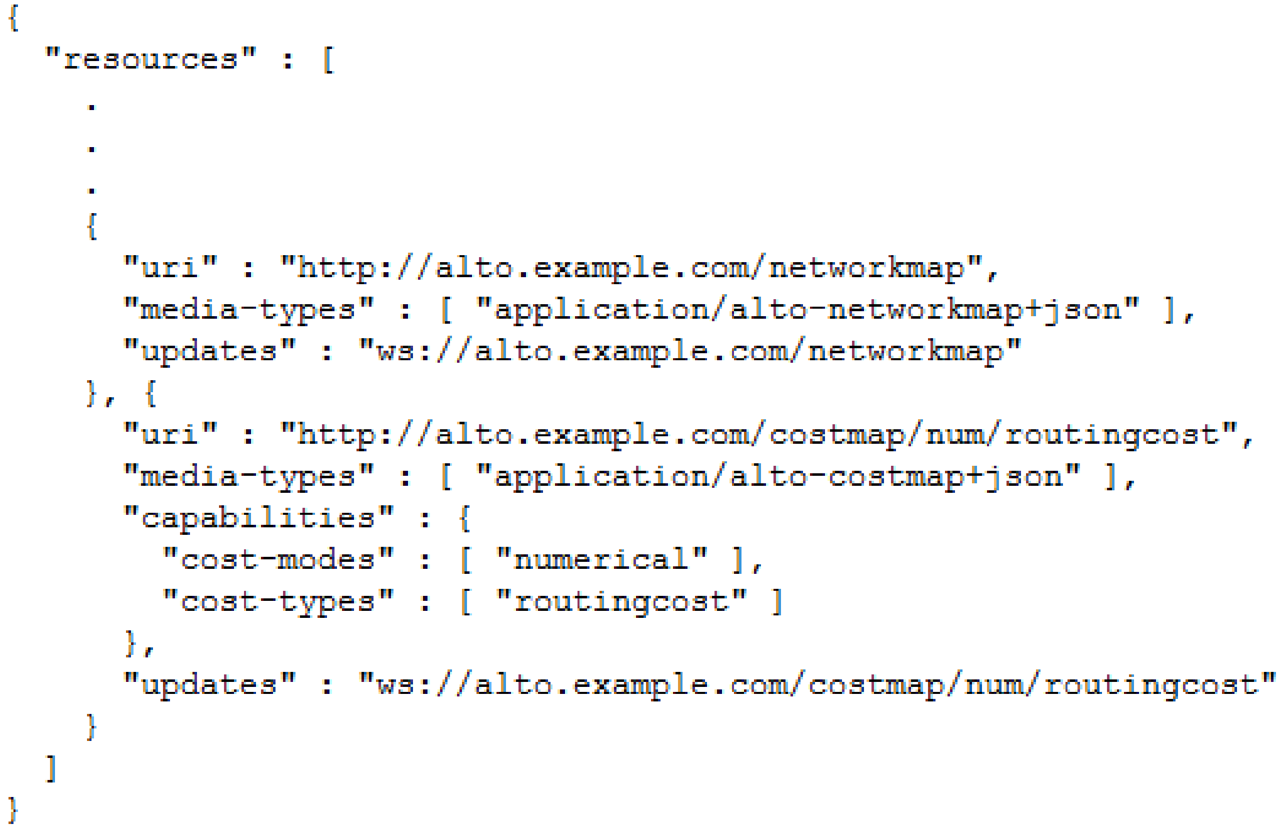
WebSocket-based server-to-client notifications for the Application-Layer Traffic Optimization (ALTO) Protocol

draft-marocco-alto-ws-02

Enrico Marocco, Jan Seedorf

IETF 89, London
ALTO WG
March 2014

Example



```
{
  "resources" : [
    .
    .
    .
    {
      "uri" : "http://alto.example.com/networkmap",
      "media-types" : [ "application/alto-networkmap+json" ],
      "updates" : "ws://alto.example.com/networkmap"
    }, {
      "uri" : "http://alto.example.com/costmap/num/routingcost",
      "media-types" : [ "application/alto-costmap+json" ],
      "capabilities" : {
        "cost-modes" : [ "numerical" ],
        "cost-types" : [ "routingcost" ]
      },
      "updates" : "ws://alto.example.com/costmap/num/routingcost"
    }
  ]
}
```

Information Resource Directory returned by an ALTO Server

- IRD contains the WebSocket URI of the update notification service associated to the specific resource
- ALTO Server provides both a network map and a cost map with corresponding update notification services

Incremental Updates

Wendy Roome & Nico Schwan

Our suggestion: Client polls at interval suggested by server. Use standard HTTP requests, so no additional client libraries are required.

ALTO Server assigns tags to Cost Map versions, just like Network Map tags. When a Cost Map changes, its tag must change. For Incremental Update Service, the client sends the tag of a previous Cost Map, and the server responds with all changes since that version.

We suggest using existing Cost Map message for incremental updates. Just re-interpret the semantics as: “replace existing src-dest costs with these, and leave the other costs as is.” To delete a cost, the server sets cost to -1 or null.

JSON Patch vs. ALTO Cost Map Response

JSON Patch is less efficient. To update costs from pid1 to pid2 & pid3, and delete the cost to pid4, JSON Patch requires:

```
{"replace": "cost-map.pid1.pid2", "value": 123,
 "replace": "cost-map.pid1.pid3", "value": 42,
 "delete": "cost-map.pid1.pid4"}
```

while an ALTO Cost Map response is just:

```
{"pid1": {"pid2": 123, "pid3": 42, "pid4": null}}
```

Incremental Update clients want fast access to large maps. Those clients will not store a map as a JSON object defined by a JSON library. Those clients will extract the cost-points from JSON, and save them in a sparse matrix or hash table. So clients cannot rely on a JSON library to automatically apply JSON Patch updates.

Similarly, servers will not store cost data as JSON objects; they will use a more compact format, and create JSON as needed. So servers cannot rely on a JSON library to automatically create a JSON patch by diff'ing two JSON objects.

JSON is an interchange format, not a storage format.

Extensions to ALTO Server Discovery:

IP Anycast based ALTO Server Discovery

draft-kiesel-alto-ip-based-srv-disc-02

Third-party ALTO Server Discovery

draft-kist-alto-3pdisc-05

IETF 89, ALTO session, London, 2014-03-06

Sebastian Kiesel

Martin Stiemerling

Reinaldo Penno

ALTO Server Discovery

- **`draft-ietf-alto-server-discovery-10`**
  - DHCP + manual config. In the RFC editor queue!
  - Useful but with certain limitations
    - DHCP is widely deployed, but not everywhere
    - Writing a portable application using DHCP opts. is not trivial (API)
    - Issues with middleboxes (forwarding DHCP opts WAN → LAN)

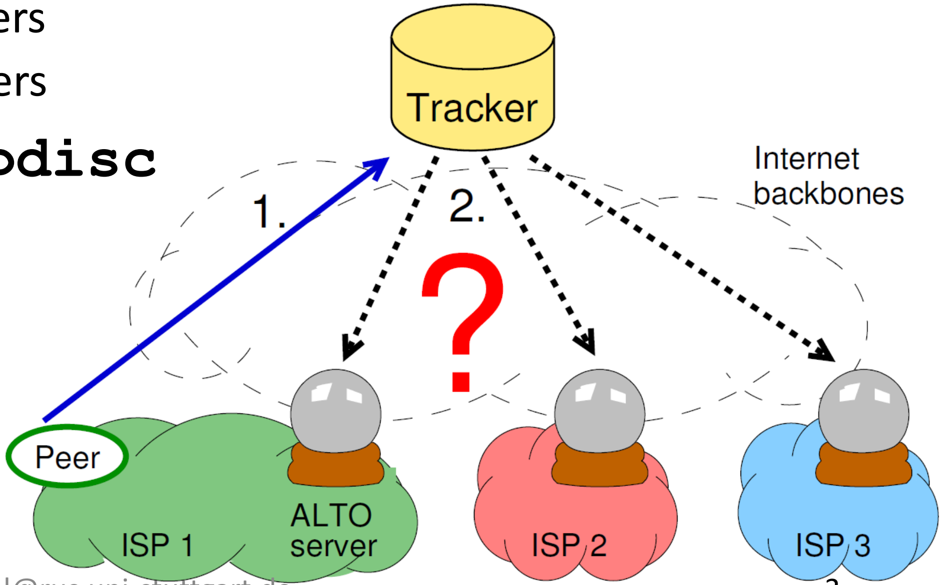
- New: **`draft-kiesel-alto-ip-based-srv-disc`**
  - Send queries for an IRD to a special **anycast address**
  - Network operator will setup routing tables to ensure that query is directed to an appropriate ALTO server
    (ALTO is quite much related to routing (costs), so routing folks will have the ALTO server on their radar anyway)

Third-party ALTO Server Discovery

- Conceptually, the ALTO protocol allows to deliver a “full” NxN network/cost map to ALTO clients. However, the information sources (e.g., ISPs) may not have this global knowledge

→ Several ALTO servers, each with partial knowledge, in the network

- If knowledge is partitioned, client needs to find the “right” server
- Alternatives that were considered and abandoned:
  - Data replication between ALTO servers
  - Request routing between ALTO servers
- New: **`draft-kist-alto-3pdisc`**
  - ALTO client finds “right” ALTO server by means of DNS lookups
  - IP address ↦ ALTO IRD URI
  - Allows gradual deployment without central coordination
  - Fulfills RFC 6708, AR-33



PID Properties

Wendy Roome & Richard Yang

Presumably the endpoints in a PID have something in common, so why not extend the Endpoint Property concept to PIDs?

Example PID properties:

> Geo: Continent code, country code, state code, lat/long bounding box, …
> Network: ISP codes, ASNs, …
> Endpoint type: server farm, residential customers, mobile devices, …

If PID2 is a subset of PID1, then PID2 inherits properties of PID1, unless overridden. The PID Property extension will define rules for such inheritance.

PID Properties and Endpoint Property Service share the same set of property definitions. EPS may return properties for the endpoint's PID, if not overridden.

As with Full & Filtered Cost Maps, the extension defines a GET-mode Full PID Property service that returns properties for all PIDs, and a POST-mode Filtered PID Property service that returns selected properties for selected PIDs. An ALTO server can choose to provide one or both.

Example: Filtered PID Property Service

The “country” and “state” properties are blank-separated lists of codes (a PID can span political borders). PID1a & PID1b are subsets of PID1, and inherit the “country” property from PID1.

Request (POST-mode):

```
{ "properties": [ "country", "state" ],
  "pids": [ "PID1", "PID1a", "PID1b" ] }
```

Response:

```
{ "meta": {
    "dependent-vtags":
       { "resource-id": "my-network-map", "tag": "314159265359" }
  }
  "pid-properties": {
    "PID1":  { "country": "USA" },
    "PID1a": { "country": "USA", "state": "NY" },
    "PID1b": { "country": "USA", "state": "NJ PA" }
  }
}
```

ALTO PID Property Extension is needed for an ALTO-based CDNI FCI Solution (draft-seedorf-cdni-request-routing-alto-06)

Jan Seedorf, Richard Yang

IETF 89, London
ALTO WG
March 2014

Example ALTO PID Properties for CDNI FCI “Footprints”

```
HTTP/1.1 200 OK
 Content-Length: TBA
 Content-Type: application/alto-pidprop+json

 {
  "meta" : {
   "dependent-vtags" : [
    {"resource-id": "my-eu-netmap",
     "tag": "1266506139"
    }
   ]
  },
  ”properties": {
   ”pid:south-france” : { “delivery-protocol": [”HTTP”], … },
   ”pid:germany” : { “delivery-protocol": [”HTTP”, “HTTPS”], … },
   “pid:rest” : {}
  }
 }
```

PID properties would allow to assign the same capabilities to all endpoints in a given CDNI footprint (footprint=ALTO PID defined in an ALTO network map)

PID Property concept allows clean separation between footprint & capabilities:

- **PID gives name to a footprint**
- **Can then easily change separately either**
  - □ **Capabilities (=PID properties) for a given footprint**
  - □ **Composition of the footprint itself**

Alternative Service (CDNI)

- CDNs have always been a major use case for ALTO.
  - Footprint sharing and request routing.
  - Progress on getting ALTO incorporated in CDNI has been difficult.
    - Asynchronous updates
    - Regions instead of strict prefix topology
    - Capabilities
  - We could try to twist the ALTO network map services into something suitable for CDNI.
    - But why not just make a new ALTO information service?
- Proposal: new information service for CDNI FCI.
  - Based on a new CDNI object format (pending)
  - In the future consider other new services as well

VPN Service PID Properties (M. Scharf)

Network map

```
HTTP/1.1 200 OK
…
Content-Type: application/alto-
  networkmap+json

{ "meta" : { … },
  "network-map" : {
    "PID11" : {
      "id" : [ "SITE-SANFRANCISCO" ]
    },
    "PID14" : {
      "id" : [ "SITE-CHICAGO" ]
    },
    "PID21" : {
      "id" : [ "SITE-OTTAWA" ]
    },
    "PID27" : {
      "id" : [ "SITE-PARIS" ]
    }
  }
}
```

Topology example taken from draft-scharf-alto-vpn-service-02

PID properties

```
HTTP/1.1 200 OK
…
Content-Type: application/alto-pidprop+json

{
  "meta" : { … },
  "properties": {
    "pid:PID11" : {
      "type": "L3VPN site",
      "location": [ "37.75 N, 122.28 W" ],
      "oper_status": "up", …
    },
    "pid:PID14" : {
      "type": "L3VPN site",
      "location": [ "41.85 N, 87.65 W" ],
      "oper_status": "down", …
    },
    "pid:PID21" : {
      "type": "L3VPN site",
      "location": [ "45.24 N, 75.43 W" ],
      "oper_status": "up", …
    },
     "pid:PID27" : {
      "type": "L3VPN site",
      "location": [ "48.86 N, 2.35 E" ],
      "oper_status": "up", …
    }
  }
}
```

- ALTO benefit: Get topology even w/o connectivity (e.g., before scale-out)
- Important ALTO extensions: General PID properties, non-IP PID identifiers
- Further requirements: Query for properties, and other requirements in draft-scharf-alto-vpn-service-02

ALTO Traffic Engineering Cost Metrics

draft-wu-alto-te-metrics-01
Qin Wu(bill.wu@huawei.com)
Y. Richard Yang (yry@cs.yale.edu)
Young Lee(leeyoung@huawei.com)
Dhruv Dhody(dhruv.ietf@gmail.com)
Sabine Randriamasy (sabine.randriamasy@alcatel-lucent.com)

JSON format example
request on ‘routingcost’ metric with constraint on ‘latency’

```
•   POST /endpointcost/lookup HTTP/1.1
•    Host: alto.example.com
•    Content-Length: TBA
•    Content-Type: application/alto-endpointcostparams+json
•    Accept: application/alto-endpointcost+json,application/alto-error+json
•   {
•     "cost-type": {"cost-mode" : "numerical",
•                 "cost-metric" : “routingcost"},
•                  "constraints" : {"delay ls 15"},
•     "endpoints" : {
•       "srcs": [ "ipv4:192.0.2.2" ],
•       "dsts": [
•         "ipv4:192.0.2.89",
•         "ipv4:198.51.100.34",
•         "ipv4:203.0.113.45"
•       ]
•     }
•   }
```

```
•   HTTP/1.1 200 OK
•   Content-Length: TBA
•   Content-Type: application/alto-endpointcost+json
•   {
•     "meta" :{
•       "cost-type": {"cost-mode" : "numerical",
•                     "cost-metric" : “routingcost“
•                     "constraints" : {"delay "},
•        }
•      },
•       "endpoint-cost-map" : {
•         "ipv4:192.0.2.2": {
•           "ipv4:192.0.2.89"    : 10, ["delay eq 0"],
•           "ipv4:198.51.100.34" : 20, ["delay eq 2"],
•           "ipv4:203.0.113.45"  : 30, ["delay eq 3"],
•       }
•     }
•   }
```

JSON format example request on ‘Bandwidth Calendaring ‘

```
•       GET /directory HTTP/1.1
•       Host: alto.example.com
•       Accept: application/alto-directory+json,application/alto-error+json

•       HTTP/1.1 200 OK
•       Content-Length: 2333
•       Content-Type: application/alto-directory+json

•       {
•         "meta" : {
•           "cost-types": {
•             "calendaring-bw": {
•               "cost-mode"  : "calendaring",
•               "cost-metric": "Availbandwidth",
•               "description": {"interval":mm/hh,"duration":mm/hh/dd/mm,
•            "start":mm/hh/dd/mm}
•             },

•        "resources" : {
•           "endpoint-cost" : {
•             "uri" : "http://alto.example.com/endpointcost/lookup",
•             "media-type" : "application/alto-endpointcost+json",
•             "accepts" : "application/alto-endpointcostparams+json",
•             "capabilities" : {
•               "cost-constraints" : true,
•               "cost-type-names" : [ "calendaring-bw"]
•             }


•        }
```

```
•  POST /endpointcost/lookup HTTP/1.1
•   Host: alto.example.com
•   Content-Length: TBA
•  Content-Type: application/alto-endpointcostparams+json
•  Accept: application/alto-endpointcost+json,application/alto-error+json
•  {
•  "cost-type": {"cost-mode" : “calendaring",
•            "cost-metric" : “Availbandwidth"},
•   "endpoints" : {
•    "srcs": [ "ipv4:192.0.2.2" ],
•     "dsts": [
•     "ipv4:192.0.2.89",
•     "ipv4:198.51.100.34",
•     "ipv4:203.0.113.45"
•    ]
•  }
• }
•       HTTP/1.1 200 OK
•       Content-Length: TBA
•       Content-Type: application/alto-endpointcost+json
•       {
•         "meta" :{
•          "cost-type": {"cost-mode" : “calendaring",
•                  "cost-metric" : “Availbandwidth"
•            }
•          },
•          "endpoint-cost-map" : {
•           "ipv4:192.0.2.2": {
•             "ipv4:192.0.2.89"    : [6,5,7,8,4,10,7,6],
•             "ipv4:198.51.100.34" : [7,4,6,8,5,9,6,7],
•             "ipv4:203.0.113.45"  : [7,6,8,5,7,9,6,8],
•          }
•         }
```

ALTO Graph Representation Extension for ALTO

M. Scharf, G. Bernstein, Y. Lee, R. Yang

Motivating Example (Revealing path coupling):

- Current ALTO maps do not convey coupling among E2E paths: e.g., bw(A –> B) = 1 && bw(C -> D) but bw(A->B, C->D) ≠ 2, if there is a shared bottleneck.
- Such information can be helpful for application scheduling, reliability analysis.
- Graph representation allows disclosure of such information.

Basic Graph Representation Schemes:

- **Path-vector:** Allows flexible network policy routing, but at the price of large representation overhead (e.g., enumeration of E2E paths among N nodes has complexity N*N*M, where M is average path length)
- **Node-edge**: Compact representation (e.g., only N * K links, where K is avg node degree), but network routing policy may not be conveyed.

Proposal:

- Simple extensions generalizing ALTO nw/cost maps to provide both path-vector and node-edge representations.
- Abstract graph for a given client (usually not a NMS); not full RIB data.
- Maintains ALTO privacy enforcement by operator.

Graph Model

```
{
 "nodes": [
   “n0“ : {
    },
   “n1“ : {
   },
   ...
 ],
 "edges": [
   { "src": “node:n0",
     "dst": “node:n1",
     "type": "directed",
     "cost" : [ {
        "cost-metric" : "delay",
        "value" : "3"
      }, {
        "cost-metric" : "availbw",
        "value" : "50"
      }, {
        "cost-metric" : "risk-group",
        "value" : ["SLRG3“]
     } ]
   },
   ...
 ]
}
```

Abstract network nodes in graph

Abstract network edges in graph

Node/edge properties, such as multiple TE metrics from draft-wu-alto-te-metrics

Node-edge

```
GET /costmap/pathvec HTTP/1.1
Host: alto.example.com
Accept: application/alto-costmap
     +json,application/alto-error+json
```

```
HTTP/1.1 200 OK
Content-Length: TBA
Content-Type: application/alto-costmap+json

 {
   "meta" : {
    ...
     "cost-type" : {"cost-mode"  : ”path-vector”}
   },
   "cost-map" : {
     "PID1": { "PID1": [ ],
                 "PID2": [“n0”,  “n1”]
                "PID3":  [“n0”] },
     "PID2": { …},
     "PID3": { … }}
  }
```

Vector of network nodes traversed.

Path-vector Cost Map