Toward a Unified Resource Representation in ALTO

draft-xiang-alto-unified-representation-01

Qiao Xiang[1,2], Franck Le[3], Y. Richard Yang[1,2], Jensen Zhang[1,2]

[1] Tongji University, [2] Yale University,

[3] IBM Watson Research Center,

July 16, 2018, IETF 102 ALTO

Background

- The ALTO protocol [RFC7285] provides network information to applications. However, the "one-big-switch" abstraction cannot reveal the resource sharing (i.e., the bottleneck) among a set of endpoint pairs. Such information is needed to support emerging applications, e.g., multi-flow scheduling.

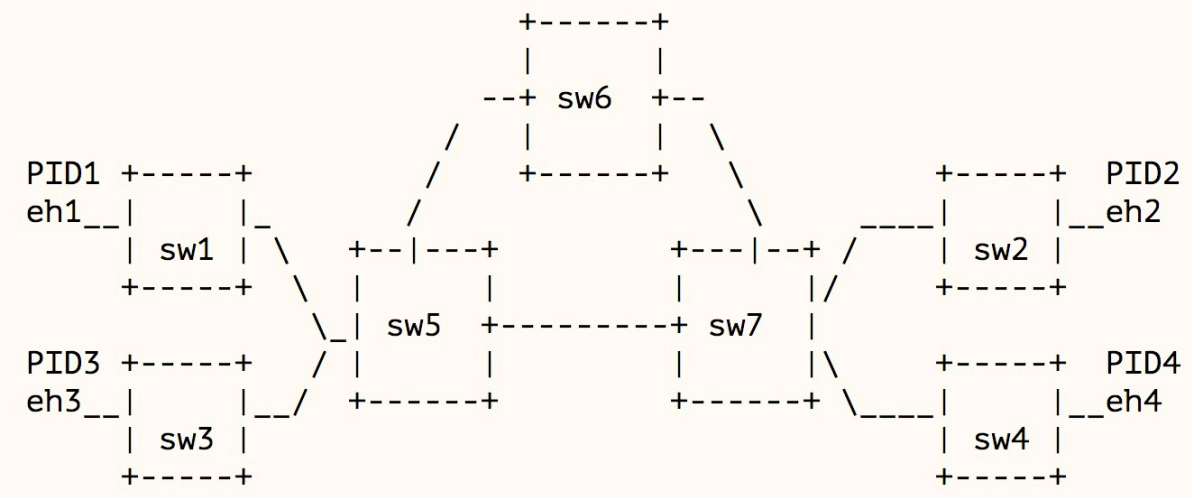


Figure source: the ALTO Path Vector draft.

Motivation

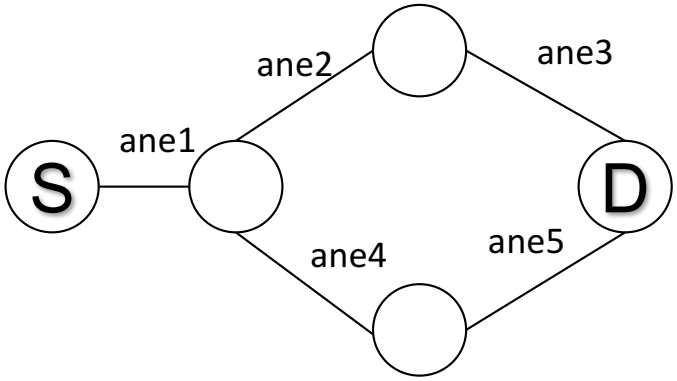
- To provide such information, the ALTO path vector extension is proposed to represent the capacity region for a set of endpoint pairs in a set of linear inequalities.
  - In the ALTO PV extension, new cost mode (array), new cost metric (ane-path), new entity domain (ane) are introduced, and extensions to cost map/endpoint cost service are proposed.
- However, the ALTO PV extension cannot provide accurate, compact information of resource sharing of flows:
  1. when network is using multi-path/multicast/load-balancing;
  2. when network is using on-demand routing (e.g., PCE); and
  3. when the application wants to get the shared risked link group (SRLG) information.
- These use cases are pointed out by scientists and engineers from an important ALTO use case.

A Proposal to Support Multipath/Multicast/LB in PV

- **Observation**: Consider an endpoint pair (e.g., a flow). Its route (no matter single-path route or multipath route) is essentially a set of *route segments*.
  - In addition, the "vector" of anes does not have to provide semantics like BGP AS-path.
- **Basic idea**: When the ALTO client submits a PV query about a set of flow sto the ALTO server. Instead of returning an array of ANEs, the ALTO server returns a set of arrays of ANEs, where *each array represents a route segment* in the route of this flow and is assigned a unique ID.
  - Motivated by RFC7911 ("Advertisement of Multiple Paths in BGP").

Support Multipath in PV: Example

- Assume the network uses ECMP to forward traffic from S to D.
  - route1: ane1 -> ane2 -> ane3
  - route2: ane1 -> ane4 -> ane5
- The capacity region of ECMP from S to D is:



Each link is 100 Mbps.

route1.bw+route2.bw <= 100
route1.bw <=100
route2.bw <=100
route1.bw = route2.bw
SD.bw = route1.bw+route2.bw

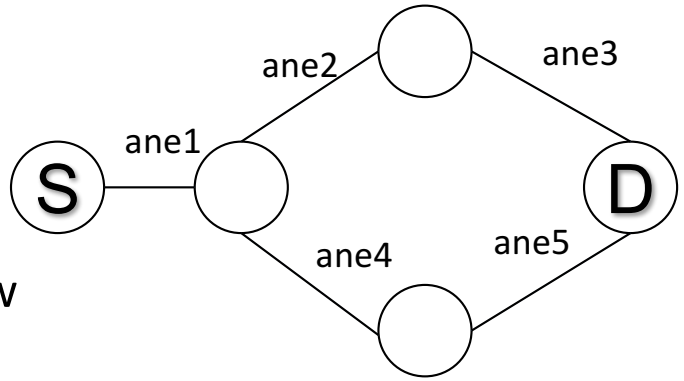
- We transform two routes into three route segments：
  - rs1: ane1
  - rs2: ane2 -> ane3
  - rs3: ane4 -> ane5
- Then the capacity region can be expressed as:

rs1.bw <= 100
rs2.bw <= 100
rs3.bw <= 100
SD.bw = rs1.bw = rs2.bw+rs3.bw

Support Multipath in PV: Example (Cont'd)

rs1: ane1
rs2: ane2 -> ane3
rs3: ane4 -> ane5

rs1.bw <= 100
rs2.bw <= 100
rs3.bw <= 100
SD.bw = rs1.bw = rs2.bw+rs3.bw



Each link is 100 Mbps.

- If we define a new entity domain "rs" (for route segment), a new cost mode "set", and a new metric "rs-set", such information can be encoded in one cost map and two property maps (one for ane, and one for rs).

```
"endpoint-cost-map": {
  "S": {
    "D": {
      "rs:1": ["ane1"],
      "rs:2": ["ane2", "ane3"],
      "rs:3": ["ane4", "ane5"]
    }
  }
}
```

```
"property-map1": {
  "ane:1": { "availbw": 50 },
  "ane:2": { "availbw": 50 },
  "ane:3": { "availbw": 50 },
  "ane:4": { "availbw": 50 },
  "ane:5": { "availbw": 50 }
}
```

```
"property-map2": {
  "rs:1": { "trafficpercentage": 1},
  "rs:2": { " trafficpercentage ": 0.5 },
  "rs:3": { " trafficpercentage ": 0.5 }
}
```

What is Next?

- This design handles multipath/multicast/load balancing.
- But it is an extension to an extension of ALTO.
- If we keep doing this for each important use case, we may end up with many extensions, with a chaos of dependency and compatibility issues.

- **Driving question**: Can we design a unified resource representation framework in ALTO to provides accurate, compact resource information to applications, who may have a wide range of requirements / objectives?

A First Step

- In the -00 version, we make a first step to tackle this question.
- **Basic idea**: Use mathematical programming constraints to represent the capacity region for a set of flows.
  - Introduce a new cost type (cost mode: "array", cost metric: "variable-list") to allow the ALTO server to send a cost map:

    (source, destination) pair -> a list of decision variables related to this pair
  - Introduce a new entity domain "cstr" (short for constraint), and use a cstr property map to send the set of mathematical constraints.

```
"cost-type": {
   "cost-mode": "array",
   "cost-metric": "variable-list"
},
```

```
"property-map": {
  "cstr:001": { "bw-cstr": "[0][0] add [0][1] leq 100"},
  "cstr:002": { "bw-cstr": "[0][0] eq [0][1]"},
  "cstr:003": { "bw-cstr": "[1][0] add [0][0] leq 100"},
  "cstr:004": { "srlg-cstr": "[0][2] intersect [1][1] eq {2, 3, 4}"},
}
```

```
"cost-map": {
  "PID1": {
    "PID2": ["f1:b2:p1", "f2:bw:p2", "f1:srlg"],
    "PID3": [ "f2:bw:p1", "f2:srlg" ]
  }
}
```

f1:bw:p1 + f1:bw:p2 <= 100

f1:bw:p1 = f1:bw:p2

f2:bw:p1 + f1:bw:p1 <= 100

f1.srlg ∩ f2.srlg = {2, 3, 4}

Discussion

- This is a first and very early step toward an ALTO unified resource representation service.
- Lots of issues need to be addressed
  - Obviously, this design is generic, but it may be too generic …
  - Security/privacy …
- We start with the returned representation sent by ALTO server, but **another important missing piece** is: how an ALTO client express the requirements on what information is needed?
  - For example, if an ALTO client wants to ask: for a set of flows, what is the network capacity region when the size of SRLG of every two flows is smaller than or equal to 2?
  - Questions like this cannot be expressed in current filtered cost/property map services.
  - For starter, the grammar for ALTO client to specify the requirements will be specified in the next version.