ALTO Integration in Rucio
Summer 2023 Work Plan

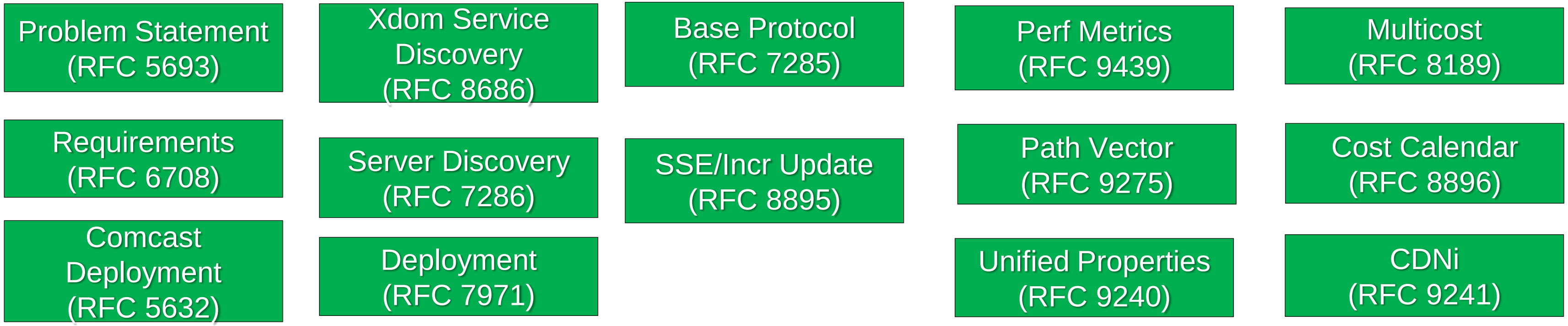
June 29, 2023

Context

- Relatively new to Rucio/FTS/LHCONE, but found them to be fascinating designs
- Initial focus is on integrating infrastructure visibility capability (i.e., ALTO) into Rucio/FTS, using existing Rucio transfer/FTS scheduling algorithms
  - Focus on both application of Internet standards and also extensions to ALTO driven by real issues revealed by Rucio/FTS application
- Then evaluate, model and optimize existing algorithms
  - Compare w/ existing algorithms with approaches taken by other systems with similar problems
  - Focus on global modeling and optimization

Background: Application-Layer Traffic Optimization (ALTO)

- Defines an Internet standard for networks to expose its state to applications to optimize both network and application performance
- Defined by the Transport Area of Internet Engineering Task Force (IETF)
- Two core components:
  - Abstractions of network state/services
  - Transport and discovery of abstractions



Problem Statement (RFC 5693)

Requirements (RFC 6708)

Comcast Deployment (RFC 5632)

Xdom Service Discovery (RFC 8686)

Server Discovery (RFC 7286)

Deployment (RFC 7971)

Base Protocol (RFC 7285)

SSE/Incr Update (RFC 8895)

Perf Metrics (RFC 9439)

Path Vector (RFC 9275)

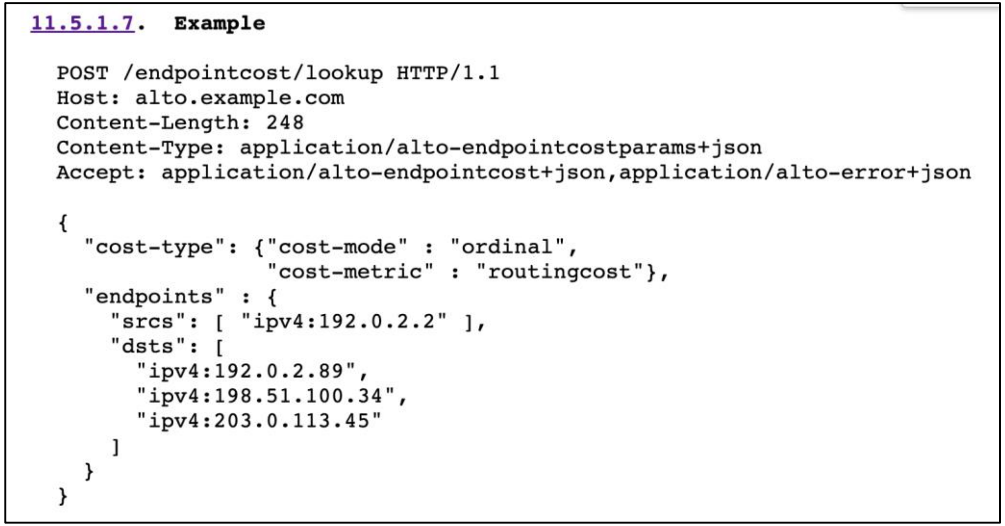
Unified Properties (RFC 9240)

Multicost (RFC 8189)

Cost Calendar (RFC 8896)

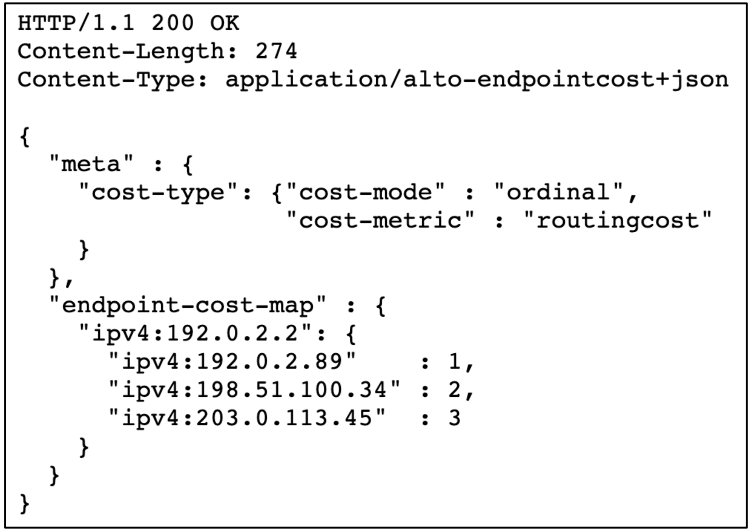
CDNi (RFC 9241)

ALTO Abstraction Example: Endpoint Cost Service (ECS)



11.5.1.7. Example

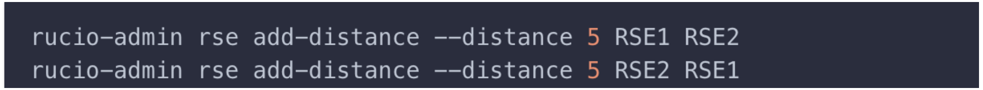
```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: 248
Content-Type: application/alto-endpointcostparams+json
Accept: application/alto-endpointcost+json,application/alto-error+json

{
  "cost-type": {"cost-mode" : "ordinal",
                "cost-metric" : "routingcost"},
  "endpoints" : {
    "srcs": [ "ipv4:192.0.2.2" ],
    "dsts": [
      "ipv4:192.0.2.89",
      "ipv4:198.51.100.34",
      "ipv4:203.0.113.45"
    ]
  }
}
```



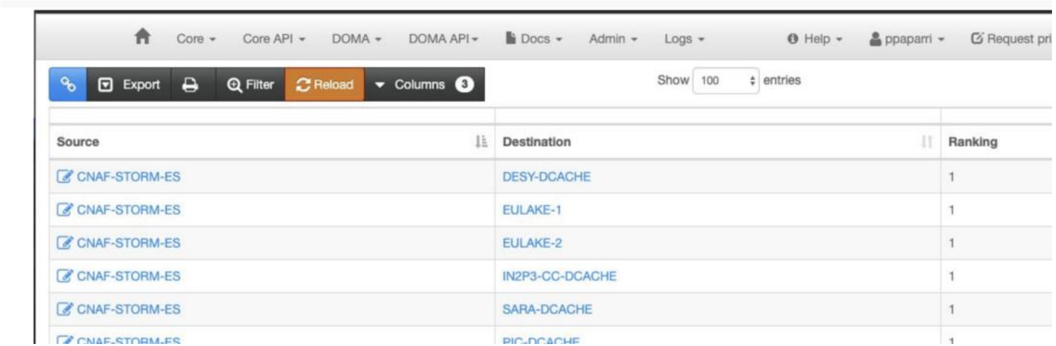
```
HTTP/1.1 200 OK
Content-Length: 274
Content-Type: application/alto-endpointcost+json

{
  "meta" : {
    "cost-type": {"cost-mode" : "ordinal",
                  "cost-metric" : "routingcost"
    }
  },
  "endpoint-cost-map" : {
    "ipv4:192.0.2.2": {
      "ipv4:192.0.2.89"    : 1,
      "ipv4:198.51.100.34" : 2,
      "ipv4:203.0.113.45"  : 3
    }
  }
}
```

- More details see [RFC 7285].

ALTO Cost Services and Rucio Distance

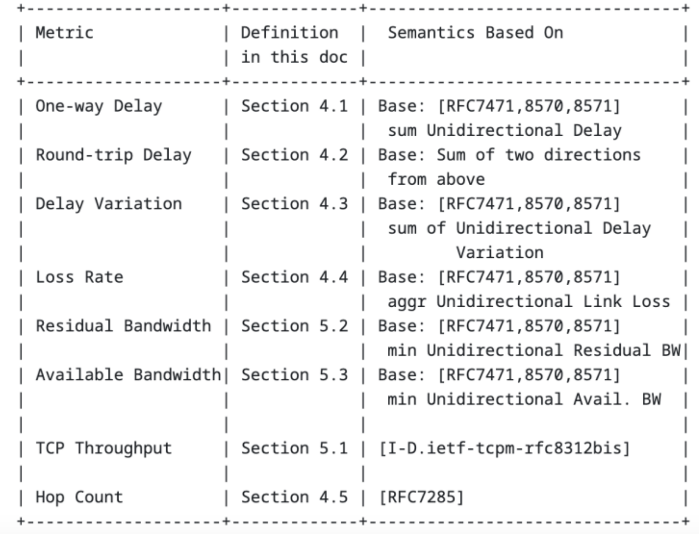
- Excellent match between Rucio distance and ALTO costs
  - ALTO endpoint cost service (ECS) and Cost Map service provide the distances between any source/destination pairs, for a set of performance metrics
- 

```
rucio-admin rse add-distance --distance 5 RSE1 RSE2
rucio-admin rse add-distance --distance 5 RSE2 RSE1
```



Source	Destination	Ranking
CNAF-STORM-ES	DESY-DCACHE	1
CNAF-STORM-ES	EULAKE-1	1
CNAF-STORM-ES	EULAKE-2	1
CNAF-STORM-ES	IN2P3-CC-DCACHE	1
CNAF-STORM-ES	SARA-DCACHE	1
CNAF-STORM-ES	PIC-DCACHE	1

Src: https://indico.cern.ch/event/867913/contributions/3769387/attachments/2001400/3341196/CRIC_-_Rucio_Workshop_5.pdf



Metric	Definition in this doc	Semantics Based On
One-way Delay	Section 4.1	Base: [RFC7471,8570,8571] sum Unidirectional Delay
Round-trip Delay	Section 4.2	Base: Sum of two directions from above
Delay Variation	Section 4.3	Base: [RFC7471,8570,8571] sum of Unidirectional Delay Variation
Loss Rate	Section 4.4	Base: [RFC7471,8570,8571] aggr Unidirectional Link Loss
Residual Bandwidth	Section 5.2	Base: [RFC7471,8570,8571] min Unidirectional Residual BW
Available Bandwidth	Section 5.3	Base: [RFC7471,8570,8571] min Unidirectional Avail. BW
TCP Throughput	Section 5.1	[I-D.ietf-tcpm-rfc8312bis]
Hop Count	Section 4.5	[RFC7285]

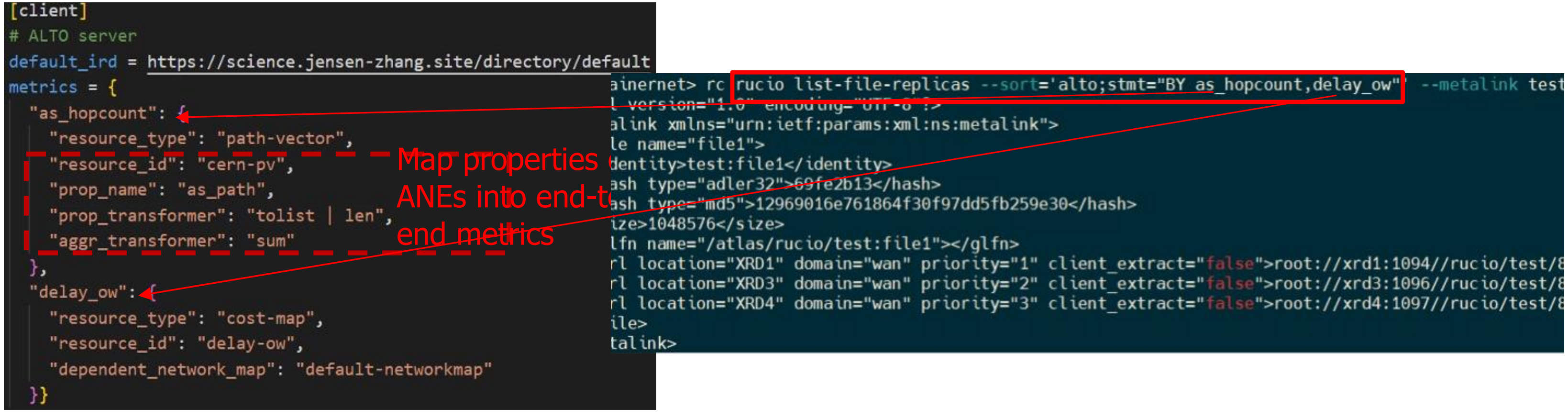
Table 1. Cost Metrics Defined in this Document.

Src: https://datatracker.ietf.org/doc/draft-ietf-alto-performance-metrics/28/

  - Benefits of ALTO based: automated, dynamic, according to network state

Thread 1: Rucio + ALTO Integration

- Three components to be added
  - T1.1 Develop and deploy ALTO servers obtaining infrastructure visibility
  - T1.2 Declare visibility to Rucio deployment by operator
  - **T1.3 Introduce unified distance expression** (UDE) in Rucio API to allow user specifying sorting expression using a combination of distances and other properties (Kai, Jensen, Lauren)

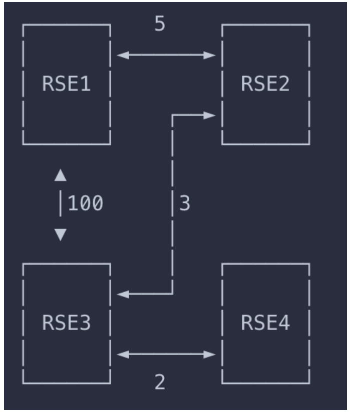


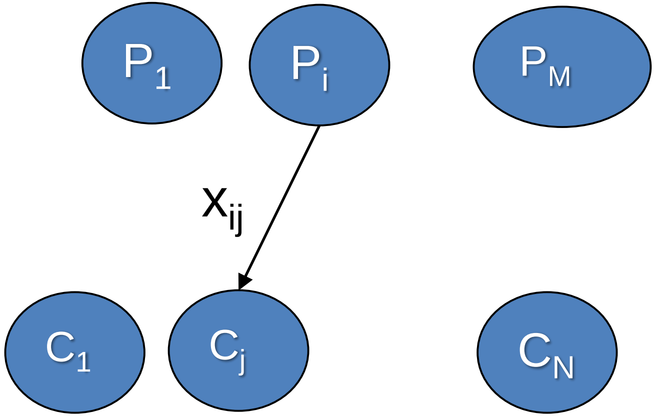
```
[client]
# ALTO server
default_ird = https://science.jensen-zhang.site/directory/default
metrics = {
  "as_hopcount": {
    "resource_type": "path-vector",
    "resource_id": "cern-pv",
    "prop_name": "as_path",
    "prop_transformer": "tolist | len",
    "aggr_transformer": "sum"
  },
  "delay_ow": {
    "resource_type": "cost-map",
    "resource_id": "delay-ow",
    "dependent_network_map": "default-networkmap"
  }}
```

```
ainernet> rc rucio list-file-replicas --sort='alto;stmt="BY as_hopcount,delay_ow"' --metalink test
l version="1.0" encoding="UTF-8"?>
alink xmlns="urn:ietf:params:xml:ns:metalink">
le name="file1">
dentity>test:file1</identity>
ash type="adler32">69fe2b13</hash>
ash type="md5">12969016e761864f30f97dd5fb259e30</hash>
ize>1048576</size>
lfn name="/atlas/rucio/test:file1"></glfn>
rl location="XRD1" domain="wan" priority="1" client_extract="false">root://xrd1:1094//rucio/test/8
rl location="XRD3" domain="wan" priority="2" client_extract="false">root://xrd3:1096//rucio/test/8
rl location="XRD4" domain="wan" priority="3" client_extract="false">root://xrd4:1097//rucio/test/8
ile>
talink>
```

Thread 2: Rucio Transfer Algorithms Modeler and Optimizer

- Formalization and Optimization of Rucio Transfer Algorithms, e.g.,
  - The algorithm framework of Rucio source selection for a downloader [1] is defined by ordering vector <source pri, path cost>
  - More direct control is to compute $\{x_{ij}\}$, which is the amount of load assigned to be sent using path P_i to client cluster C_j
  - Task: Transfer Modeler and Optimizer compute $\{x_{ij}\}$ and find config resulting in better $\{x_{ij}\}$





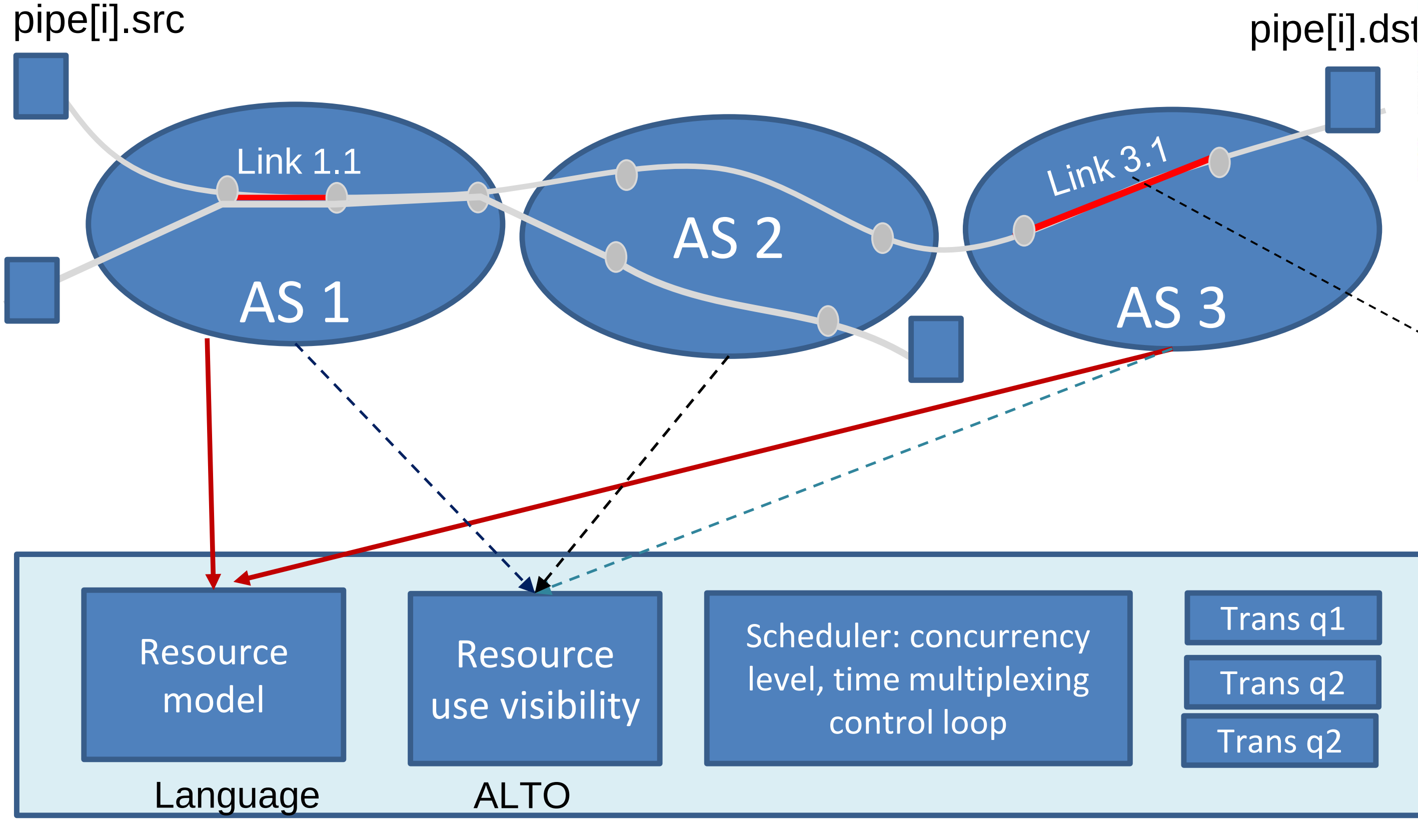
[1] https://rucio.cern.ch/documentation/operator/transfers/transfers-preparer/

Thread 3: Rucio Transfer Algorithms in Context

- Comparison: source selection ~ load balancer (LB) for server selection; traffic engineering (TE) in networking; Content multihoming in cloud cost optimization
  - LB:
    - Local LB, e.g., nginx alg such as round robin, load (example see http://nginx.org/en/docs/http/load_balancing.html), and more complex such as consistent hash, virtual servers, Maglev
    - Global LB (GLB), e.g., Akamai, Netflix
  - Internet TE: ECMP, fast rerouting, …

- Main task: comparison of Rucio design and other designs and discussions on implications

Bigger Picture (Thread 4): ALTO+FTS



Resource model

Language

Resource use visibility

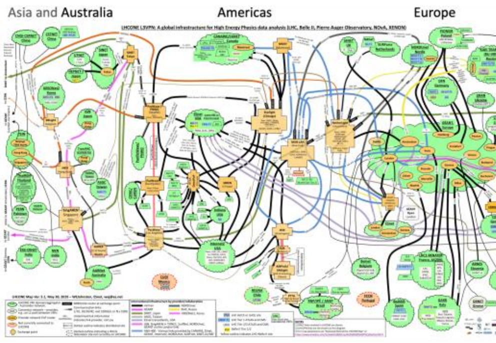
ALTO

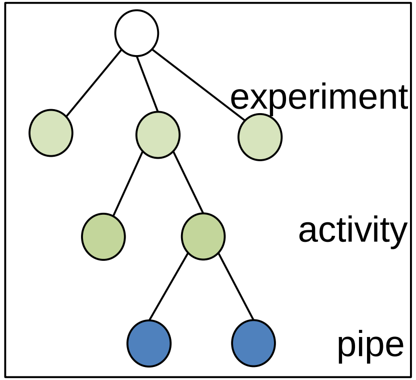
Scheduler: concurrency level, time multiplexing control loop

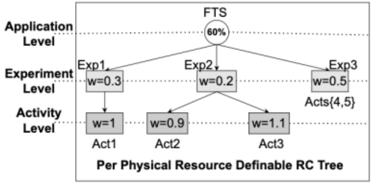
Trans q1

Trans q2

Trans q2



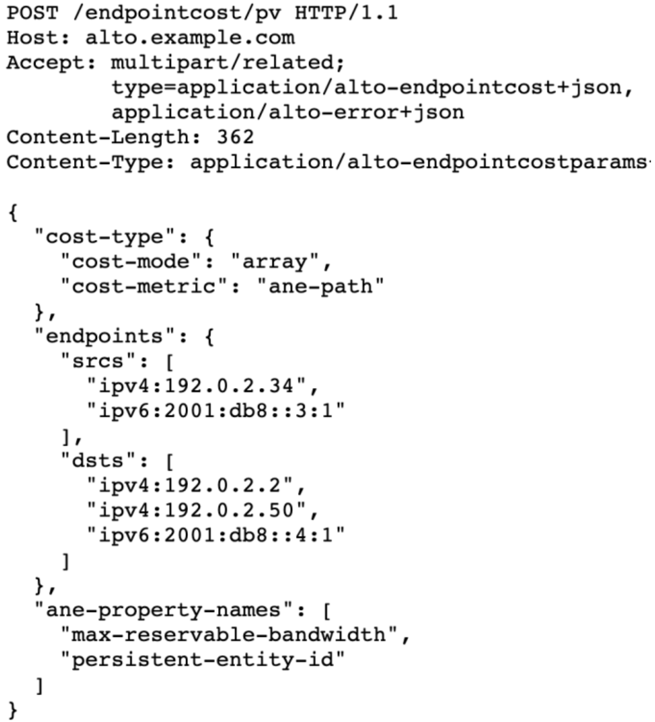




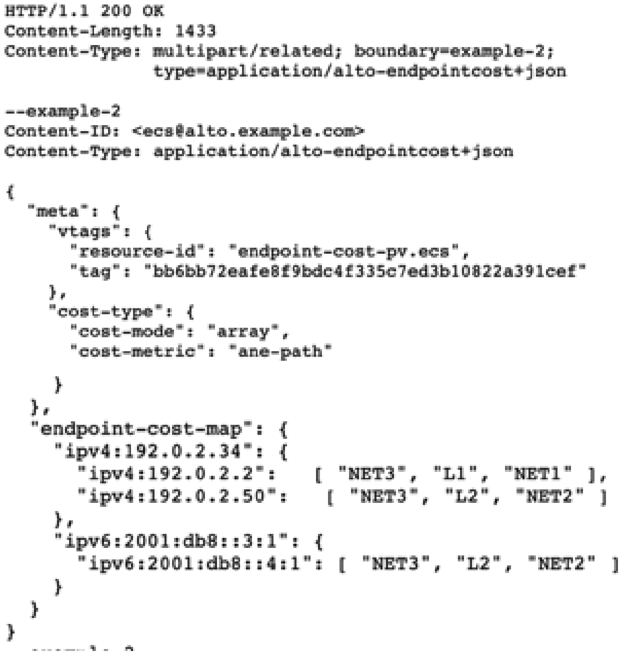
Per Physical Resource Definable RC Tree

Backup Slides

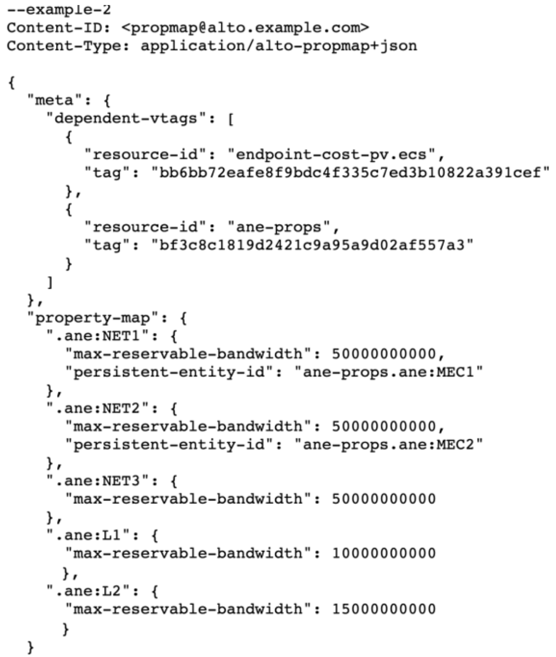
ALTO Abstraction Example: Path Vector



```
POST /endpointcost/pv HTTP/1.1
Host: alto.example.com
Accept: multipart/related;
        type=application/alto-endpointcost+json,
        application/alto-error+json
Content-Length: 362
Content-Type: application/alto-endpointcostparams

{
  "cost-type": {
    "cost-mode": "array",
    "cost-metric": "ane-path"
  },
  "endpoints": {
    "srcs": [
      "ipv4:192.0.2.34",
      "ipv6:2001:db8::3:1"
    ],
    "dsts": [
      "ipv4:192.0.2.2",
      "ipv4:192.0.2.50",
      "ipv6:2001:db8::4:1"
    ]
  },
  "ane-property-names": [
    "max-reservable-bandwidth",
    "persistent-entity-id"
  ]
}
```



```
HTTP/1.1 200 OK
Content-Length: 1433
Content-Type: multipart/related; boundary=example-2;
              type=application/alto-endpointcost+json

--example-2
Content-ID: <ecs@alto.example.com>
Content-Type: application/alto-endpointcost+json

{
  "meta": {
    "vtags": {
      "resource-id": "endpoint-cost-pv.ecs",
      "tag": "bb6bb72eafe8f9bdc4f335c7ed3b10822a391cef"
    },
    "cost-type": {
      "cost-mode": "array",
      "cost-metric": "ane-path"

    }
  },
  "endpoint-cost-map": {
    "ipv4:192.0.2.34": {
      "ipv4:192.0.2.2":   [ "NET3", "L1", "NET1" ],
      "ipv4:192.0.2.50":   [ "NET3", "L2", "NET2" ]
    },
    "ipv6:2001:db8::3:1": {
      "ipv6:2001:db8::4:1": [ "NET3", "L2", "NET2" ]
    }
  }
}
```



```
--example-2
Content-ID: <propmap@alto.example.com>
Content-Type: application/alto-propmap+json

{
  "meta": {
    "dependent-vtags": [
      {
        "resource-id": "endpoint-cost-pv.ecs",
        "tag": "bb6bb72eafe8f9bdc4f335c7ed3b10822a391cef"
      },
      {
        "resource-id": "ane-props",
        "tag": "bf3c8c1819d2421c9a95a9d02af557a3"
      }
    ]
  },
  "property-map": {
    ".ane:NET1": {
      "max-reservable-bandwidth": 50000000000,
      "persistent-entity-id": "ane-props.ane:MEC1"
    },
    ".ane:NET2": {
      "max-reservable-bandwidth": 50000000000,
      "persistent-entity-id": "ane-props.ane:MEC2"
    },
    ".ane:NET3": {
      "max-reservable-bandwidth": 50000000000
    },
    ".ane:L1": {
      "max-reservable-bandwidth": 10000000000
     },
    ".ane:L2": {
      "max-reservable-bandwidth": 15000000000
     }
  }
```

- More details see https://datatracker.ietf.org/doc/html/draft-ietf-alto-path-vector-21#section-8.1

ALTO+FTS Visibility Mapping Example

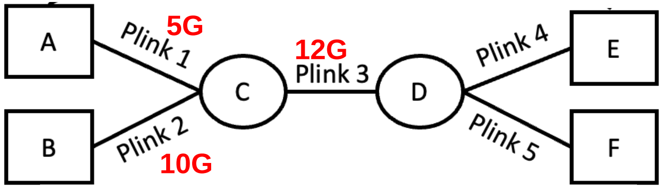
Resource Model:

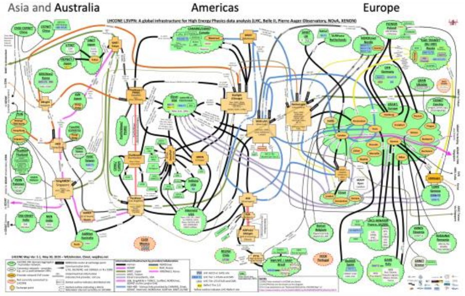
Experiment X:

R1: <Plink 1> <= 5G

R2: <Plink 2> <= 10G

R3: <Plink 3> <= 12G





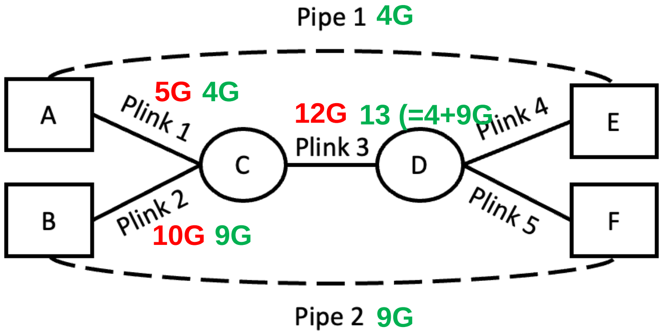
Experiment X Uses 2 pipes:

Pipe1.traffic = 4G, Pipe2.traffic = 9G

Resource Use Visibility (ALTO):

Pipe 1: {Plink 1, Plink 3, Plink 4}.

Pipe 2: {Plink 2, Plink 3, Plink 5}.



Since usage on Plink 3 is over resource model, controller reduces concurrency levels of Pipe 1 and Pipe 2.