

ALTO Working Group

Internet Draft

Intended status: standard

Young Lee

Dhruv Dhody

Qin Wu

Huawei

Greg Bernstein

Grotto Networking

Tae Sang Choi

ETRI

October 21, 2013

**ALTO Extensions to Support Application and Network Resource
Information Exchange for High Bandwidth Applications in TE networks**

[draft-lee-alto-app-net-info-exchange-04.txt](#)

Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on April 21, 2014.

Copyright Notice

Copyright (c) 2013 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.

Abstract

This draft proposes ALTO information model and protocol extensions to support application and network resource information exchange for high bandwidth applications in partially controlled and controlled environments as part of the infrastructure to application information exposure (i2aex) initiative.

Table of Contents

1.	Introduction.....	3
2.	Problem Statement.....	5
3.	ALTO Constraints Filtering Extension Model.....	8
3.1.	ALTO Query from Application Stratum to Network Stratum....	8
3.2.	ALTO Response from Network Stratum to Application Stratum	10
3.3.	Information Model of ALTO Query from Application Stratum to Network Stratum.....	10
3.4.	Information Model of ALTO Response from Network Stratum to Application Stratum.....	11
3.5.	ALTO Protocol Extension for Constraints Filtering Mechanism	11
3.6.	Multiple Service Class.....	13
3.6.1.	Gold Service.....	13
3.6.2.	Silver Service.....	15
3.6.3.	Bronze Service.....	17
4.	ALTO Protocol Extension for Graph Representation Mechanism....	19
5.	Summary and Conclusion.....	19
6.	Security Considerations.....	19
7.	IANA Considerations.....	19
8.	References.....	19
8.1.	Informative References.....	19
	Author's Addresses.....	21
	Intellectual Property Statement.....	21
	Disclaimer of Validity.....	22

1. Introduction

This draft proposes ALTO information model and protocol extensions to support application and network resource information exchange for high bandwidth applications in partially controlled and controlled environments as part of the infrastructure to application information exposure (i2aex) initiative. The Controlled and partially controlled ALTO environments referred to here are those where general access to a specific ALTO server may be restricted to a qualified list of clients.

This draft is build upon the previously introduced High Bandwidth Use Cases [[HighBW](#)] and assumes that the network type carrying high bandwidth is a Traffic-Engineered (TE) network. In [[HighBW](#)], we have discussed two generic use cases that motivate the usefulness of general interfaces for cross stratum optimization in the network core. In our first use case, network resource usage became significant due to the aggregation of many individually unique client demands. In the second use case where data centers are sending large amount of data with each other, bandwidth usage was already significant enough to warrant the use of traffic engineered "express lanes" (e.g., private line service). We introduce third use case where inter-CDN providers may want to expose controlled network resource usage information so that CDN applications (e.g., request routing server) can utilize such information when appropriate decisions (e.g., request routing redirection) are needed.

These use cases result in optimization problems that trade off computational versus network costs and constraints. Both featured use cases show the usefulness of an ALTO interface between the application and network strata in optimizing the networked applications.

In particular, this draft introduces: (i) enhanced constraints filtering extensions to the ALTO protocol to reduce extraneous information transfer and enhance information hiding from the network's perspective; (ii) constrained cost graph mechanism encoding that enables enhanced application traffic optimization that was introduced by [[HighBW](#)].

In controlled and partially controlled environments in which operators are willing to share additional network stratum resource information such as bandwidth constraints or additional cost types of topology (e.g., graph or summary), it can be useful to reduce the amount of information transferred from the ALTO server to the ALTO client.

In considering information exchange between the application stratum and the network stratum, especially from the network stratum to the application stratum, the degree of information details is one of the key concerns from the network providers' standpoint. On the one hand, the network information has to be useful to the application; on the other hand, the provided network information should hide details about the network. In order to achieve these desired goals, a simple enhancement to ALTO protocol would help significantly both in reducing/filtering the amount of information and in increasing the usefulness of the information from network to application.

Figure 1 shows ALTO Client-Server Architecture for Application-Network information Exchange. Figure 1 shows that ALTO Client in the application stratum can interface with ALTO Server in the network stratum. With this architecture, a simple ALTO query mechanism from application (via ALTO client) to network (via ALTO server) can be implemented. According to this architecture, ALTO Client is assumed to interact with the Application Orchestrator that has the knowledge of the end-user (i.e., source) application requirement, Data Center locations (i.e., destinations) and their resource information.

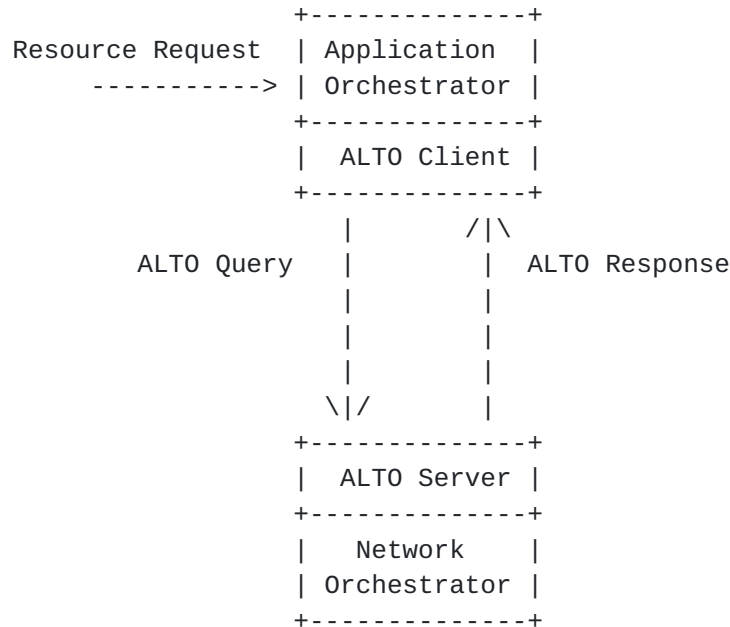


Figure 1 ALTO Client-Server Architecture for Application-Network information Exchange

The Application Orchestration functions depicted in Figure 1 interfacing data centers and end-users are out of the scope of this document. For simplicity purpose, Figure 1 doesn't depict the detailed relationship between ALTO client and server. In fact, both client and server don't need to be in the same administration boundary. It can be inter-operator and one to many relationships. For example, in the cases of inter-CDN environment or generic multi-domain environment, ALTO client represents a request routing server of upstream CDN operator and it interacts with multiple downstream CDN operators for their network resource information to make efficient decisions for desired quality CDN services. Interaction methods can either iterative or recursive. That is, ALTO client can interact with multiple ALTO servers directly or ALTO client can interact with one representative ALTO server and subsequent interaction is done by the representative one to rest of ALTO servers.

The organization of this document is as follows. [Section 2](#) discusses the ALTO architecture in the context of the application and network strata interaction. [Section 3](#) provides ALTO Information model and protocol extension to support ALTO Constraints Filtering Mechanism. [Section 4](#) provides ALTO information model and the protocol extension to support ALTO Constrained Cost Graph Mechanism.

[2. Problem Statement](#)

One critical issue in Application-Network information exchange in ALTO is the amount of information exchanged between the application and network strata. The information provided by network providers can be not so useful to the application stratum unless such information is abstracted into an appropriate level the that application stratum can understand.

In partially controlled and controlled environments, network providers can furnish appropriately abstracted and pruned information to the application stratum with the cooperation of the application stratum that can indicate some level of filtering and pruning in its query.

To reduce extraneous information this draft allows for "filtering" (or "pruning") of the following information in query/response of the ALTO pull model:

- . Topology Filtering - reduction of the results to only those specified set of source(s) and destination(s) instead of the entire network cost map. Note that this mechanism is not new in the current ALTO protocol. In the context of application-network information exchange, this topology filtering can be

- of a tremendous help in reducing the amount of data exchanged between application and network.
- . Multiple Service Class: ALTO server may provide multiple class of service (Gold, Silver, or Bronze) and allow application to request them accordingly.
- . Multiple Cost: ALTO server should be able to provide multiple cost for a end to end path or abstract links in the graph.
- . Optimization Criteria: The optimization criteria that the ALTO server may use. For example, the criteria can be least number of hops, least amount of delay (latency), etc.
- . Constraint Filtering on paths or graphs (e.g., bandwidth, latency, hop count, packet loss, etc.) - reduction of results to only those that meet ALTO client specified cost bounds.

As discussed in [[HighBW](#)], in a controlled environment optimization is significantly enhanced by sharing data related to bandwidth constraints and additional cost measures [[MultiCost](#)], [[TE-cost](#)]. Such information may be considered sensitive to the network provider just as application data, e.g., usage, demand, etc., may be considered sensitive to an application provider. [Section 3](#) provides ALTO information model and protocol extensions to support topology, multiple service class, constraints filtering mechanism.

Multiple Service Class (such as gold, silver and bronze services) MAY be supported by the ALTO server. These service classes could specify how the network is used (for ex exclusively reserved for the application, protection provided etc). The Application should further provision/reserve the network using some mechanisms which are out of scope of this document. Some example of services: _

. Gold Service

This service could be used to specify that an exact path meeting the application needs should be found. This path would be provisioned and resources reserved exclusively for its use. An example could be a private enterprise DC, which wish to offload to a public DC during peak load.

. Silver Service

This service could be used to get the path properties between User regions and DC. It could also specify some basic constraints that all of them should satisfy. These paths would be provisioned and resources maybe reserved. The Application may further assign end user request to a particular DC by using the network information of these paths. An example could be a gaming server geographically dispersed at multiple DC. The end-user (gamer) could be dynamically

assigned to the DC by looking at the past assignment, DC load and network properties.

. Bronze Service

This service could be used to specify that a simple best effort path should be found. This path would not be provisioned and resources will not be reserved. The service could still return the network information to the application which can use this information for DC selection by taking network information into consideration. An example could be a HD video service, which may use the network info to select video source for the end user.

While it is important to reduce and filter the information amount from network to application, for some applications that require stringent QoS objectives (e.g., bandwidth and latency), simple summary source-destination network resource information (i.e., summary form of topology) may not provide sufficient details to the application stratum. For example, suppose that a multiple number of large concurrent flows need to be scheduled from application to network. In such a case, a summary form of network topology that only shows source-destination bandwidth availability may not show the bottleneck links over which more than one flow may compete for the link bandwidth resource. This problem was indicated by [[HighBW](#)]. The following are the excerpts from [[HighBW](#)].

Consider the network shown in Figure 2, where DC indicates a datacenter, ER an end user region (as in the end user aggregation use case), N a switching node of some sort, and L a link. The link capacities and costs are also shown on the figure as well as a cost map between [ER1, ER2] and [DC1, DC2, DC3]. Since the network has a tree structure (very unusual but easier to draw in ASCII art), the cost map is unique.

As an illustration, assume that the maximum available capacity between any individual end region and a data center is 5 units(i.e., $L1=L2=L5=L6=5$). However, link L3 (capacity 8 units) represents a bottle neck to all the data centers (L3 is on all the paths to DC1, DC2, or DC3 from all end regions, ER1 and ER2).

ALTO Cost Map is shown in the lower right corner of Figure 2. This summary cost map does not provide enough details on the bottle necks. The lower left corner shows Link Capacity Cost, from which the bottle necks can be shown such that multi-flow commodity scheduling can be made possible to avoid such bottle necks.

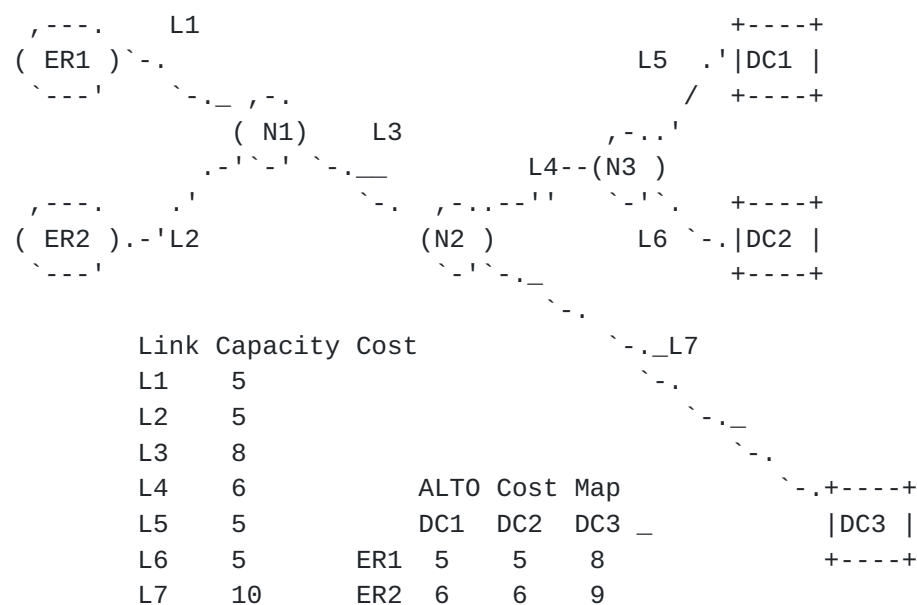


Figure 2. Example network illustrating bottlenecks

With the current ALTO cost map structure, the least cost path from ER1 would be either to DC1 or DC2. However, with the proposed capacitated cost map, the connection from ER1 to DC3 could be a better choice than the rest depending on the relative cost of network resources to data center resources.

A more general and relatively efficient alternative is to provide the requestor with a capacitated and multiply weighted graph that approximates and abstracts the capabilities of the network as seen by the source and destination location sets. This document provides ALTO information model and protocol extensions to support the graph model in [Section 4](#).

3. ALTO Constraints Filtering Extension Model

3.1. ALTO Query from Application Stratum to Network Stratum

In order for the network stratum to provide its resource information, the application stratum needs to provide the End Point Cost Map to the network stratum. The End Point Cost Map should include the following information at a minimum:

- . End Point Source Address(es) /* these are the respective addresses of the nearest PE's to the user/client location */
- . End Point Destination Address(es) /* these are the respective addresses of the nearest PE's to a set of the candidate Data Center locations that can provide service to the user request. */

Note that how ALTO client derives the End Point Source/Destination addresses in terms of the nearest PE's is beyond the scope of this document.

- . Service-Class:= {gold, silver, bronze} /*the service class as described in this document*/
- . Cost Type:= 'routingcost' as defined by base specification. Additional cost (ex. latency, hopcount) are defined in [[MultiCost](#)] and [[TE-cost](#)].
- . Cost Mode :={summary, graph} /* the cost map can be either a summary form or a graph form */

- o Cost Mode: summary

This cost mode is indicated by string 'summary'. This mode indicates that the returned costs contain end-to-end values which can be used by application stratum for better selection of resources.

- o Cost Mode: graph

This cost mode is indicated by string 'graph' in which case an abstract topology is returned to the application.

- . Constraints /* a set of constraints that apply to the requested path summary or graph for filtering. For instance, constraints can be like bandwidth greater than 'x', latency less than 'y', hopcount less than 'z', packetloss less than 'a' etc. */
- . Objective-function (or Optimization Criteria): The summary or the graph should be computed based on optimizing which parameter - IGP cost, latency, residual bandwidth, etc. This is for future use.

3.2. ALTO Response from Network Stratum to Application Stratum

In response to the ALTO Query from the Application Stratum, the Network Stratum needs to provide the filtered Cost Map Data of the feasible path found. The Filtered End Cost Map Data should include the following information at a minimum:

- . The list of feasible Source-Destination pair and its Cost Type
- . For each feasible S-D pair, indicate the following as specified in [Section 3.4](#):
 - o Service Class;
 - o Cost Mode;
 - o Cost Type;
 - o Endpoint Cost Map Data
- . Parameter Values /* indicate the actual values of each constraint requested */

Note that in case of Graph, each S-D pair is the source of the abstract link and the destination of the abstract link.

3.3. Information Model of ALTO Query from Application Stratum to Network Stratum

Alto query:

```
Object{
  TypedEndpointAddr  Src<1...*>; /*atleast one source*/
  TypedEndpointAddr  Dsts<2...*>; /*atleast two destinations*/
}EndpointList;
```

```
Object{
  ServiceClass      service-class;
  CostMode          cost-mode;
  CostType          cost-type;
  [JSONString       constraints<0...*>; ]
  [JSONString       ObjectiveFunction]
  EndpointList      endpoints;
}EndpointCostMapReq;
```

3.4. Information Model of ALTO Response from Network Stratum to Application Stratum

Alto response:

```
Object-map{
    JSONString      costparam;
} EndpointCostParam ;

Object-map{
    TypedEndpointAddr -> EndpointCostParam<1...*>;
} EndpointCosts ;

Object-map{
    TypedEndpointAddr -> EndpointCosts;
} EndpointCostMapData ;

Object{
    ServiceClass      service-class;
    CostMode          cost-mode;
    CostType          cost-type;
    [EndpointCostMapData map;]
}EndpointCostMapRsp;
```

The Alto response consist of map (EndpointCostMapData) which is map containing the S-D pairs information. For each destination, its parameters (rank, cost etc) is included using EndpointCostParam.

3.5. ALTO Protocol Extension for Constraints Filtering Mechanism

This section provides the ALTO protocol extensions based on the information model discussed in Sections [3.3](#). and 3.4. The scenario provided in this section is that the ALTO client in the Application Stratum requests the summary cost map from the network with one source and three destinations.

In this particular example, the ALTO client requests the filtered summary of the network path subject to: bandwidth ≥ 20 , latency < 10 , hop count < 5 and packet loss < 0.03 .

The ALTO server provides the resulted network paths in summary.

POST /endpointcost/lookup HTTP/1.1

Host: alto.example.com

Content-Length: [TODO]

Content-Type: application/alto-csoendpointcostparams+json

Accept: application/alto-csoendpointsummary+json,application/alto-error+json

```
{
  "service-class" : "silver",
  "cost-mode" : "summary",
  "cost-type" : "routingcost",
  "constraints": ["availbw gt 20", "delay lt 10", "hopcount lt 5",
"pktloss lt 0.03"],
  "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: [TODO]

Content-Type: application/alto-csoendpointsummary+json

```
{
  "meta" : {},
  "data" : {
    "service-class" : "silver",
    "cost-mode" : "summary",
    "cost-type" : "routingcost",
    "map" : {
      "ipv4:192.0.2.2": {
        "ipv4:192.0.2.89" : [ "delay eq 5",
"hopcount eq 8", "pktloss eq 0.01", cost eq
100" ],
        "ipv4:18.51.100.34" : [ "delay eq 9",
```

```
        "hopcount eq 10", "pktloss eq 0.02", cost
eq 120" ],
    "ipv4:203.0.113.45" : [ "delay eq 40",
        "hopcount eq 12", "pktloss eq 0.02", cost
eq 50" ]
    }
  }
}
```

3.6. Multiple Service Class

The examples of various class of service is as follows, note that these examples are for illustrative purpose only.

3.6.1. Gold Service

As an example of a Gold service, consider a customer (say an Enterprise Private DC) who pays Top-Dollar to setup network based on the actual demand. The Path (maybe a TE LSP) would not be used by any other customer / application giving guarantee of service and best QoE to the application. The ALTO request/response may be used first to get the network states and later the path may also be provisioned by some mechanism which is out of scope of this document.

In this example, the application may like to find out the ranking of the destinations (DC) from the network point of view. It may further set the filtering constraints for bandwidth (bw), delay etc. The ALTO server first filter the destination that do not meet the constraints, further it provides ranking information based on the requested costtype.

Alto Request:

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: [TODO]
Content-Type: application/alto-csoendpointcostparams+json
Accept: application/alto-
csoendpointsummary+json,application/alto-
error+json
{
  "service-class" : "gold",
```

```
    "cost-mode" : "summary",
    "cost-type" : "routingcost",
    "constraints": ["availbwgt 20", "delay lt 10",
                   "pktloss lt 0.03", "jitter lt 10", "hopcount
lt 5" ],
    "endpoints" : {
      "srcs": [ "ipv4:192.0.2.2" ],
      "dsts": [
        "ipv4:192.0.2.89",
        "ipv4:198.51.100.34",
        "ipv4:203.0.113.45"
      ]
    }
  }
```

ALTO server would factor in the filtering constraints and provide only the ranking information to the application.

Alto Response:

```
HTTP/1.1 200 OK
Content-Length: [TODO]
Content-Type: application/alto-csoendpointsummary+json
{
  "meta" : {},
  "data" : {
    "service-class" : "gold",
    "cost-mode" : "summary",
    "cost-type" : "routingcost",
    "map" : {
      "ipv4:192.0.2.2": {
        "ipv4:192.0.2.89" : [ "rank eq 3" ],
        "ipv4:198.51.100.34" : [ "rank eq 1" ],
        "ipv4:203.0.113.45" : [ "rank eq 2" ]
      }
    }
  }
}
```

Note that above is just an example, a gold service may also choose to get detailed end to end information or an abstract graph.

3.6.2. Silver Service

As an example of a Silver service, consider a customer (say a Online Gaming Company) which will pay flat subscription fees to connect end user-regions to the DC hosting the online gaming servers.. In this case during the setup phase a flat full mesh of paths are established between the User regions and the Data Centers.

The Application gaming load balancer would handle the gaming end user by allocating him to a particular DC (gaming server). The reserved resources during admin setup are allocated to multiple end user requests.

In this example, application may want to know the end to end properties of the path between the user-regions and the DC. It may further set the filtering constraints for bandwidth (bw), delay etc.

Alto Request:

POST /endpointcost/lookup HTTP/1.1

```
Host: alto.example.com
Content-Length: [TODO]
Content-Type: application/alto-csoendpointcostparams+json
Accept: application/alto-csoendpointsummary+json,application/alto-
error+json
{
  "service-class" : "silver",
  "cost-mode" : "summary",
  "cost-type" : "routingcost",
  "constraints": ["availbwgt 20", "delay lt 10",
                  "pktloss lt 0.03", "jitter lt 10", "hopcount
lt 5" ],
  "endpoints" : {
    "srcs": [
      "ipv4:192.0.2.2",
      "ipv4:192.0.2.10"
    ],
    "dsts": [
      "ipv4:192.0.2.89",
      "ipv4:198.51.100.34",
      "ipv4:203.0.113.45"
```

```
    ]  
  }  
}
```

ALTO server would factor in the filtering constraints and provide the end to end cost parameters to the application.

Alto Response:

```
HTTP/1.1 200 OK  
Content-Length: [TODO]  
Content-Type: application/alto-csoendpointsummary+json  
{  
  "meta" : {},  
  "data" : {  
    "service-class" : "silver",  
    "cost-mode" : "summary",  
    "cost-type" : "routingcost",  
    "map" : {  
      "ipv4:192.0.2.2": {  
        "ipv4:192.0.2.89" : [ "delay eq 5", "jitter eq 5",  
                             "pktloss eq 0.01", "hopcount eq 8",  
        "cost eq 100" ],  
        "ipv4:198.51.100.34" : [ "delay eq 9", "jitter eq 3",  
                                "pktloss eq 0.02", "hopcount eq 10",  
        "cost eq 500" ],  
        "ipv4:203.0.113.45" : [ "delay eq 4", "jitter eq 4",  
                                "pktloss eq 0.02", "hopcount eq 12",  
        "cost eq 200" ]  
      }  
  
      "ipv4:192.0.2.10": {  
        "ipv4:192.0.2.89" : [ "delay eq 4", "jitter eq 4",  
                             "pktloss eq 0.03", "hopcount eq 6",  
        "cost eq 300" ],  
        "ipv4:203.0.113.45" : [ "delay eq 6", "jitter eq 6",  
                                "pktloss eq 0.04", "hopcount eq 8",  
        "cost eq 400"]  
      }  
    }  
  }  
}
```

Note that above is just an example, a silver service may also choose to get an abstract graph in response.

3.6.3. Bronze Service

As an example of a Bronze service, consider a customer (say a Video service) doesn't reserve resources but pays a small fee to get an abstract view of the network. Best effort service, use IP best effort path (instead of reserved paths used by gold, silver). The application (global load balancer) could get the network abstract topology and would further handle the end user request by allocating them to a particular DC or CDN.

In this example, application may rely on the basic IP best effort but would like to know the abstract topology that could be used by the application to find out bottleneck etc. Note that no constraints are passed in this example and graph is requested.

Alto Request:

```
POST /endpointcost/lookup HTTP/1.1
Host: alto.example.com
Content-Length: [TODO]
Content-Type: application/alto-csoendpointcostparams+json
Accept: application/alto-
csoendpointsummary+json,application/alto-
error+json
{
  "service-class" : "bronze",
  "cost-mode" : "graph",
  "cost-type" : "routingcost",
  "endpoints" : {
    "srcs": [
      "ipv4:192.0.2.2",
      "ipv4:192.0.2.10"    ],
    "dsts": [
      "ipv4:192.0.2.89",
      "ipv4:198.51.100.34",
      "ipv4:203.0.113.45"
    ]
  }
}
```


ALTO server would prepare an abstract network graph based on the source(s) and destination(s). The graph may also include some internal (maybe abstract) nodes (ex 192.0.2.20 and 192.0.2.30).

Alto Response:

```
HTTP/1.1 200 OK
Content-Length: [TODO]
Content-Type: application/alto-csoendpointsummary+json
{
  "meta" : {},
  "data" : {
    "service-class" : "bronze",
    "cost-mode" : "graph",
    "cost-type" : "routingcost",
    "map": {
      "ipv4:192.0.2.2": {
        "ipv4:192.0.2.20" : [ "delay eq 9", "jitter eq 2",
                             "pktloss eq 0.04", "availbw eq 20",
"cost eq 100" ]
      }

      "ipv4:192.0.2.20": {
        "ipv4:192.0.2.89" : [ "delay eq 5", "jitter eq 2",
                             "pktloss eq 0.02", "availbw eq 30",
"cost eq 100" ],
        "ipv4:198.51.100.34" : [ "delay eq 3", "jitter eq 2",
                             "pktloss eq 0.01", "availbw eq 50",
"cost eq 400" ]
      }

      "ipv4:192.0.2.10": {
        "ipv4:192.0.2.30" : [ "delay eq 4", "jitter eq 2",
                             "pktloss eq 0.01", "availbw eq 60",
"cost eq 300" ]
      }

      "ipv4:192.0.2.30": {
        "ipv4:203.0.113.45" : [ "delay eq 2", "jitter eq 2",
                             "pktloss eq 0.03", "availbw eq 10",
"cost eq 200" ]
      }
    }
  }
}
```

}

Note that above is just an example, a bronze service may also choose to get end to end information instead of an abstract graph in response.

Note that the EndpointCostMapData can be used for both the Graph representation as well as the end to end path.

4. ALTO Protocol Extension for Graph Representation Mechanism

The encoding details for graph representation mechanism are shown in [Section 3.6.3](#) where the use of graph in a Bronze service is described.

5. Summary and Conclusion

TBD

6. Security Considerations

TBD

7. IANA Considerations

TBD

8. Acknowledgements

The authors would like to thank Richard Yang and Sabine Randriamasy for many helpful comments that greatly improved the contents of this draft.

9. References

9.1. Informative References

[HighBW] G. Bernstein and Y. Lee, "Use Cases for High Bandwidth Query and Control of Core Networks," [draft-bernstein-alto-large-bandwidth-cases](#), work in progress.

[MultiCost] S. Randriamasy, Ed., "Multi-Cost ALTO," [draft-randriamasy-alto-multi-cost](#), work in progress.

[TE-cost] Q. Wu, et. al. "JSON Format Extensions for Traffic Engineering (TE) performance metrics in the ALTO Information Resource Directory, [draft-wu-alto-json-te](#), work in progress.

Author's Addresses

Young Lee
Huawei Technologies
1700 Alma Drive, Suite 500
Plano, TX 75075
USA
Phone: (972) 509-5599
Email: leeyoung@huawei.com

Dhruv Dhody
Huawei Technologies, India
Email: dhruv.dhody@huawei.com

Qin Wu
Huawei Technologies, China
Email: bill.wu@huawei.com

Greg M. Bernstein
Grotto Networking
Fremont California, USA
Phone: (510) 573-2237
Email: gregb@grotto-networking.com

Tae-Sang Choi
ETRI
161 Gajong-Dong, Yusong-Gu
Daejeon, Republic of Korea
Phone: (8242) 860-5628
Email: choits@etri.re.kr

Intellectual Property Statement

The IETF Trust takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in any IETF Document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights.

Copies of Intellectual Property disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or

the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement any standard or specification contained in an IETF Document. Please address the information to the IETF at ietf-ipr@ietf.org.

Disclaimer of Validity

All IETF Documents and the information contained therein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION THEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.

