Use Cases for High Bandwidth Query and Control of Core Networks

draft-bernstein-alto-large-bandwidth-cases-01.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 September 12, 2011.

Copyright Notice

Copyright (c) 2012 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.
Internet-Draft   Cross Stratum Optimization Use-cases        March 2012

carefully, as they describe your rights and restrictions with respect to this document.

Abstract

This draft describes two generic use-cases that illustrate application layer traffic optimization applied to high bandwidth core networks. The type of information and interactions needed to perform various optimizations is described. In addition, extensions to the existing ALTO protocol are suggested that provide this functionality.

Table of Contents

1. Introduction...................................................3
1.1. Computing Clouds, Data Centers, and End Systems.............4
2. End System Aggregate Networking................................5
2.1. Aggregated Bandwidth Scaling................................5
2.2. Cross Stratum Optimization Example..........................6
2.3. Data Center and Network Faults and Recovery................7
2.4. Cross Stratum Control Interfaces................................8
3. Data Center to Data Center Networking..........................9
3.1. Cross Stratum Optimization Examples........................9
3.2. Network and Data Center Faults and Reliability.............10
4. Potential ALTO Protocol Extensions............................11
4.1. High Bandwidth Network Information..........................12
4.1.1. Maximum Reservable Bandwidth................................13
4.1.2. Latency Information......................................14
4.1.3. Endpoint Access Bandwidth Capacity.......................14
4.2. Network Information via Constraint and Cost Graph...........14
4.3. Network Updates and Notifications............................17
4.3.1. Notification Interface....................................17
4.4. Application-Network Reservation Interface..................18
4.4.1. IP Bypass/Traffic Engineering.............................18
4.4.2. High Bandwidth Reservation/Recovery Interface...........19
5. Conclusion....................................................19
6. Security Considerations........................................20
7. IANA Considerations............................................20
8. References....................................................20
8.1. Informative References.....................................20
Author’s Addresses................................................22
Intellectual Property Statement..................................22
Disclaimer of Validity............................................22
1. Introduction

Cloud Computing, network applications, software as a service (SaaS), Platform as a service (PaaS), and Infrastructure as a Service (IaaS), are just a few of the terms used to describe situations where multiple computation entities interact with one another across a network. When the communication resources consumed by these interacting entities is significant compared with link or network capacity then opportunities may exist for more efficient utilization of available computation and network resources if both computation and network stratums cooperate in some way. The application layer traffic optimization (ALTO) working group is tackling the similar problem of "better-than-random peer selection" for distributed applications based on peer to peer (P2P) or client server architectures [1]. In addition, such optimization is important in content distribution networks (CDNs) as illustrated in [2].

In the network stratum, particularly at the lower layers such as MPLS and optical, there are many restoration and recovery mechanisms to deal with network faults. The emergence of network based applications or cloud based disaster recovery/business recovery brings a new dimension to fault management, but also opportunities to more efficiently deliver higher levels of reliability. For example, the reliability requirements for mission critical applications are typically quantified by two key time parameters. The first is the Recovery Time Objective (RTO) which is the time to get the application back up and functioning and is similar to network recovery time notions. The second is the Recovery Point Objective (RPO) which quantifies in terms of time the amount of data loss that can be tolerated when a disaster occurs. Different applications and organizations can have greatly different demands from milliseconds to 12 hours. In addition, the amount of data that may need to be transferred to meet these objectives can vary greatly amongst different application types. With recover point objectives of, say an hour or more, a dynamic optical network layer could be very efficiently shared so as to reduce the overall cost to achieve a given layer of reliability. However, to do so requires cooperation between application and network stratum.

General multi-protocol label switching (GMPLS) [3] can and is being applied to various core networking technologies such as SONET/SDH and wavelength division multiplexing (WDM) [4]. GMPLS provides dynamic network topology and resource information, and the capability to dynamically allocate resources (provision label switched paths). Furthermore, the path computation element (PCE) [5] provides for traffic engineered path optimization.
However, neither GMPLS nor PCE provide interfaces that are appropriate for an application layer entity to use for the following reasons:

. GMPLS routing exposes full network topology information which tends to be proprietary to a carrier or require specialized knowledge and techniques to make use of, e.g., the routing and wavelength assignment (RWA) problem in WDM networks [4].

. Core networks typically consist of two or more layers, while applications are typically only know about the IP layer and above. Hence applications would not be able to make direct use of PCE capabilities.

. GMPLS signaling interfaces are defined for either peer GMPLS nodes or via a user network interface (UNI) [6]. Neither of these are appropriate for direct use by an application entity.

In this paper we discuss two general use-cases that can generate core network flows with significant bandwidth and may vary significantly over time. The "cross stratum optimization" problems generated by these use cases are discussed. Finally, we look at interfaces between the application and network "stratums" that can enable these types of optimizations and how they can be created via extensions to the current ALTO protocol[7].

1.1. Computing Clouds, Data Centers, and End Systems

While the definition of cloud computing or compute clouds is somewhat nebulous (or "foggy" if you will) [8], the physical instantiation of compute resources with network connectivity is very real and bounded by physical and logical constraints. For the purposes of this draft, we will call any network connected compute resources a data center if its network connectivity is significant compared either to the bandwidth of an individual WDM wavelength or with respect to the network links in which it is located. Hence we include in our definition very large data centers that feature multiple fiber access and consume more than 10MW of power, moderate to large content distribution network (CDN) installations located in or near major internet exchange points, medium sized business centers, etc...

We will refer to those computational entities that don’t meet our bandwidth criteria for a data center as an "end system".
2. End System Aggregate Networking

In this section we consider the fundamental use case of end systems communicating with data centers as shown in Figure 1. In this figure the "clients" are end systems with relatively small access bandwidth compared to a WDM wavelength, e.g., under 100Mbps. We show these clients roughly partitioned into three network related end user regions ("A", "B", and "C"). Given a particular network application, in a static network application situation, each client in a region would be associated with a particular data center.

Figure 1. End system to data center communications.

2.1. Aggregated Bandwidth Scaling

One of the simplest examples where the aggregation of end system bandwidth can quickly become significant to the "network" is for video on demand (VoD) streaming services. Unlike a live streaming service where IP or lower layer multicast techniques can be generally applied, in VoD the transmissions are unique between the data center and clients. For regular quality VoD we’ll use an estimate of 1.5Mbps per stream (assuming H.264 coding), for HD VoD we’ll use an estimate of 10Mbps per stream. To fill up a 10Gbps capacity optical wavelength requires either 6,666 or 1,000 clients.
for regular or high definition respectively. Note that special
multicasting techniques such as those discussed in [9] and peer
assistance techniques such as provided in some commercial systems
[10] can reduce the overall network bandwidth requirements.

With current high speed internet deployment such numbers of clients
are easily achieved; in addition demand for VoD services can vary
significantly over time, e.g., new video releases, inclement weather
(increases number of viewers), etc...

2.2. Cross Stratum Optimization Example

In an ideal world both data centers and networks would have
unlimited capacity, however in actuality both can have constraints
and possibly varying marginal costs that vary with load or time of
day. For example suppose that in Figure 1 that Data Center 3 has
been primarily serving VoD to region "C" but that it has, at a
particular period in time, run out of computation capacity to serve
all the client requests coming from region "C". At this point we
have a fundamental cross stratum optimization (CSO) problem. We want
to see if we can accommodate additional client request from region
"C" by using a different data center than the fully utilized data
center #3. To answer this questions we need to know (a) available
capacity on other data centers to meet a request, (b) the marginal
(incremental) cost of servicing the request on a particular data
center with spare capacity, (c) the ability of the network to
provide bandwidth between region "C" to a data center, and (d) the
incremental cost of bandwidth from region "C" to a data center.
Figure 2. Aggregated flows between end systems and data centers.

In Figure 2 we show a possible result of solving the previously mentioned CSO problem. Here we show the additional client requests from region "C" being serviced by data center #2 across the network. Figure 2 also illustrates the possibility of setting up "express" routes across the network at the MPLS level or below. Such techniques, known as "optical grooming" or "optical bypass"[11],[12] at the optical layer, can result in significant equipment and power savings for the network by "bypassing" higher level routers and switches.

2.3. Data Center and Network Faults and Recovery

Data center failures, whether partial or complete, can have a major impact on revenues in the VoD example previously described. If there is excess capacity in other data centers within the network associated with the same application then clients could be redirected to those other centers if the network has the capacity. Moreover, MPLS and GMPLS controlled networks have the ability to reroute traffic very quickly while preserving QoS. As with general network recovery techniques [13] various combinations of pre-
planning and "on the fly" approaches can be used to tradeoff between recovery time and excess network capacity needed for recovery.

In the case of network failures there is the potential for clients to be redirected to other data centers to avoid failed or over utilized links.

2.4. Cross Stratum Control Interfaces

Two types of load balancing techniques are currently utilized in cloud computing. The first is load balancing within a data center and is sometimes referred to as local load balancing. Here one is concerned with distributing requests to appropriate machines (or virtual machines) in a pool based on the current machine utilization. The second type of load balancing is known as global load balancing and is used to assign clients to a particular data center out of a choice of more than one within the network and is our concern here. A number of commercial vendors offer both local and global load balancing products. Currently global load balancing systems have very little knowledge of the underlying network. To make better assignments of clients to data centers many of these systems use geographic information based on IP addresses. Hence we see that current systems are attempting to perform cross stratum optimization albeit with very coarse network information. A more elaborate interface for CSO in the client aggregation case would be:

1. A Network Query Interface - Where the global load balancer can inquire as to the bandwidth availability between "client regions" and data centers.

2. A Network Resource Reservation Interface - Where the global load balancer can make explicit requests for bandwidth between client regions and data centers.

3. A Fault Recovery Interface - For the global load balancer to make requests for expedited bulk rerouting of client traffic from one data center to another. Or for the network layer to make requests to the application to help deal with network faults.

The network query interface can be considered a superset of the functionality supported by the current ALTO protocol [7]. Potential extensions are detailed in section 4.
3. Data Center to Data Center Networking

There are a number of motivations for data center to data center communications: on demand capacity expansion ("cloud bursting"), cooperative exchanges between business partners, offsite data backup, "rent before building", etc... In Figure 3 we show an example where a number of businesses each with an "internal data center" contracts with a large external data center for additional computational (which may include storage) capacity. The data centers may connect to each other via IP transit type services or more typically via some type of Ethernet virtual private line or LAN service.

![Diagram of data center to data center networking](image)

**Figure 3. Basic data center to data center networking.**

3.1. Cross Stratum Optimization Examples

In the DC-to-DC example of Figure 3 we can have computational constraints/limits at both local and remote data centers; fixed and marginal computational costs at local and remote data centers; and
network bandwidth costs and constraints between data centers. Note that computing costs could vary by the time of day along with the cost of power and demand. Some cloud providers have quite sophisticated compute pricing models including: reserved, on demand, and spot (auction) variants.

In addition, to possibly dynamically changing pricing, traffic loads between data centers can be quite dynamic. In addition, data movement between data centers is another source of large network usage variation. Such peaks can be due to scheduled daily or weekly offsite data backup, bulk VM migration to a new data center, periodic virtual machine migration, etc...

3.2. Network and Data Center Faults and Reliability

For networked applications that require high levels of reliability/availability the network diagram of Figure 4 could be enhanced with redundant business locations and external data centers as shown in Figure 4. For example cell phone subscriber databases and financial transactions generally require what is called geographic database replication and results in extra communication between sites supporting high availability. For example if business #1 in Figure 4 required a highly available database related service then there would be an additional communication flows from the data center "1a" to data center "1b". Furthermore, if business #1 has outsourced some of its computation and storage needs to independent data center X then for resilience it may want/need to replicate (hot-hot redundancy) this information at independent data center Y.
4. Potential ALTO Protocol Extensions

This section discusses the applicability of the ALTO protocol and necessary extensions to support the high bandwidth consuming use cases previously covered. Before doing so we discuss general properties of the high bandwidth scenarios that may differ significantly from other uses of the ALTO protocol.

The first has to do with scope and scale. The consumer of high bandwidth alto extensions is typically some type of application controller within a data center, as opposed to an individual end user. The number of such entities with a need for the high bandwidth related information is orders of magnitude smaller than, say, peer to peer networking users, or applications closer to the end user. Since a network provider may consider this information sensitive, there may be a desire to limit its distribution to a "pre-registered" set of entities.
Secondly, there is the notion of time scales. In cloud services we already see variants such as "on demand" compute instances and "reserved" compute instances. For network resource queries we may be concerned with (a) current bandwidth availability, (b) bandwidth availability at a future time, or (c) bandwidth for a bulk data transfer of a given amount that must take place within a given time window.

Time-dependent bandwidth information can be and typically are considered in network planning and provisioning systems. For example, a VoD provider knows ahead of time when the latest "blockbuster" film will be available via its service and can make estimates based on historical data on the bandwidth that it will need to deal with the subsequent demand. The following discussions, however, are restricted to "current time" for now.

Finally another goal in the design of an interface between the application and networking strata is to minimize the need for either stratum to know too much about the inner workings of the other. Hence as much as possible it is desired to insulate the applications stratum from technology specifics of the network. That said, data centers providing IaaS may prefer to specify flows and connectivity at a layer below IP such as Ethernet.

4.1. High Bandwidth Network Information

ALTO’s network map and cost map concepts can be used to support the aforementioned high bandwidth use cases. In this section we will explore both how they could be used in high bandwidth "core" networks and how they might be extended to better support large bandwidth optimization.

The ALTO concept of provider defined network location identifier, (PID), is a powerful network abstraction mechanism that is also appropriate for optical/high bandwidth scenarios. For example, a network provider could assign PIDs to WDM ROADMs or OTN switches providing access to an optical core network. All subtending datacenters or hosts would have their IP addresses grouped with such a PID. The collection of these would form an ALTO network map. Furthermore, a corresponding ALTO cost map can be used by the network to indicate preferred connectivity. Since not all these entities necessarily connect directly to an edge WDM ROADM or OTN switch, ALTO’s Endpoint Property Service can be used to denote the type of interface supported by an end system or data center and its bandwidth capabilities.
4.1.1. Maximum Reservable Bandwidth

The amount of bandwidth of available between two sites or subnetworks can be of prime interest to large bandwidth consuming applications. Unlike "unused" IP bandwidth, sub-IP bandwidth such as that from SDH, OTN, and WDM cannot be probed from a network edge or application. The only way to find out if such bandwidth could be allocated to a particular application data flow is to query the network.

One may want to query the network as to the reservable bandwidth in a number of different cases:

(a) Bandwidth available between a single source destination pair (two PIDs)

(b) Bandwidth between one particular source (PID) and several other destinations (PIDs)

(c) Bandwidth between one set of sources (PIDs) and another set of destinations (PIDs).

Case (a), bandwidth between two points, is well defined, however, in cases (b) and (c) there is some ambiguity. For example in (c) are we considering multiple sources communicating with multiple destinations at the same time? Do some of these pairs interfere with each other? To fully understand such constraints some type of constrained graph abstraction would be needed.

However, if we restrict the question in cases (b) and (c) to what is the maximum reservable bandwidth between each source and destination pair within the sets considered individually, then the question is unambiguous, useful, and can fit within ALTO’s existing cost map structure (section 5.2 [7]). A new ALTO cost type of "reservable bandwidth" can be defined for this purpose. This would be a "numeric" cost type that represents the actual bandwidth in the unit of, say, Mbps.

From the point of view of an optical network, an extended ALTO request would arrive at our extended ALTO server asking for the "reservable bandwidth" between multiple Source Network Locations, say [Src_1, Src_2, ..., Src_m], and a list of multiple Destination Network Locations, say [Dst_1, Dst_2, ..., Dst_n]. The network computing entity would calculate the "reservable bandwidth" between all of these individual source destination pairs. The extended ALTO Server would then return the "reservable bandwidth" as an ALTO Path Cost for each communicating pair (i.e., Src_1 -> Dst_1, ..., Src_1 -> Dst_n, ..., Src_m -> Dst_1, ..., Src_m -> Dst_n).
4.1.2. Latency Information

Latency information, either fixed due to propagation delay times, or statistical measures due to queuing induced delays can be similarly represented via ALTO’s cost map structure.

When choosing amongst flows between multiple data centers utilizing significant amounts of bandwidth, alternative routes with differing latency may need to be considered. In such a situation, a simple latency cost map, may need to be replaced by an abstract graph model to allow for more effective optimization of resources.

4.1.3. Endpoint Access Bandwidth Capacity

There are a number of standard sized pipes used to access high bandwidth networks and these can either be larger or smaller than the bandwidth availability within various portions of the network. Hence to make good use of network resources it is desirable to advertise and endpoints access bandwidth capacity. Typically this would be a number in terms of Mbps or Gbps and would reflect the true bandwidth available to the endpoint after upstream bottlenecks or overhead is taken into account. This information could be advertised via ALTO’s endpoint property service.

4.2. Network Information via Constraint and Cost Graph

As discussed in the previous section, as the desired connectivity between locations becomes more complex (rather than exclusive point to point), the basic ALTO cost map structure can be insufficient to reveal network bottlenecks and hence optimization decision points.

Consider the network shown in Figure 5, where DC indicates a data center, 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). In a similar way, link L4 (capacity 6 units) represents a bottle neck to data centers DC1 and DC2 from all end regions, ER1 and ER2. A simple "cost map" like structure misses these bottle necks.
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.

The creation of an approximate graph model to represent the network for cross layer optimization purposes is similar to the well-known topology aggregation problem [14] and [15], but different in a number of respects. First, the goal is not the approximation of the network structure for general route computation use, but a view of only a portion of the network relevant to the participating locations that approximates the costs and constraints amongst these locations. Second, the specific technologies underlying the costs and constraints are of no interest to the application layer and hence much technology specific layer information that one sees in GMPLS link state routing databases would be absent in such a graph.
Like the current ALTO filtered cost map, a request for a
capacitated, weighted graph would take source and destination PIDs
as inputs. In JSON notation we could represent the resulting graph
as an JSON object containing link objects. A first cut encoding
could be something like:

```json
object {
  LinkEntry [LinkName]<0..*>;
} CostConstraintGraphData;

object {
  PIDName:    a-end; // Node name at one side of the link
  PIDName:    z-end; // Node name at the other side of the link
  Weight:     wt;
  JSONNumber: latency;
  Capacity:   r-cap; // Reservable capacity
} LinkEntry;
```

Where a link name is formatted like a PIDName (but names a link),
and PID names are used for both provider defined location and
provider defined internal model node identification. A graph
representation of the network of 0 might look like:

```json
{
  "meta": {},
  "data": {
    "graph": {
      "L1": {"a-end":"ER1", "z-end":"N1", "wt":1,"r-cap":5},
      "L2": {"a-end":"ER2", "z-end":"N1", "wt":2,"r-cap":5},
      "L3": {"a-end":"N1", "z-end":"N2", "wt":1,"r-cap":8},
      "L4": {"a-end":"N2", "z-end":"N3", "wt":2,"r-cap":6},
      "L5": {"a-end":"N3", "z-end":"DC1", "wt":1,"r-cap":5},
      "L6": {"a-end":"N3", "z-end":"DC2", "wt":1,"r-cap":5},
      "L7": {"a-end":"N2", "z-end":"DC3", "wt":6,"r-cap":10}
    }
  }
}
```
4.3. Network Updates and Notifications

Changing conditions in the network such as costs or capacity may need to be relayed to the application layer in suitable form and in a time frame relative to their importance to service QoS, service delivery, or cross layer optimization.

Network fault conditions can affect service QoS in a number of ways. The most obvious being a significant reduction in capacity to current application flows. In such a case the application would want to be notified as soon as possible and take remedial action. In other cases a network fault may only be observable as an increase in latency (due to increased length of recovered optical path) such an increase may not immediately result in breach of a service level agreement (SLA) but could cumulatively over time. Hence notification of such a change in condition would need to be timely and the network may qualify if the change of state is relatively permanent or what the duration may be.

Some applications, such as those involving bulk file transfer, may have flexible time windows, with the exact time the service is rendered dictated by network availability. In particular, the network takes advantage of application flexibility in the exact scheduling of the network resources to be used. Such occurrences may be non-recurring, e.g. a one off bulk file transfer, or recurring as would be common in cloud based system backup and restore applications. In this case the notification from the network needs to relatively timely (but most likely on the order of seconds rather than milliseconds), is specific to a particular network service instance rather than raw network cost or capacity, and the entire notification process may require a non-repudiation security assurance.

Changes in the network that only affect costs but not QoS can affect the cross layer optimization of an existing application. The time frame for such notifications would typically be in terms of fractions of an hour to days.

4.3.1. Notification Interface

With the exception of the "notification of network service instance availability", all other notifications can be made via modifications or updates to suitably extended network or cost maps, or graphs.

Since the high bandwidth uses cases deal with a rather restricted user group, a number of implementation mechanisms may be possible, that may not be viable in a more general ALTO deployment. For example, with a capacitated graph representation we may selectively
update specific links of the graph for particular application entities. Note that in order to do this the network layer would need to keep track of the graph models in use by specific application entities and update them as appropriate.

4.4. Application-Network Reservation Interface

The network query interfaces previously discussed allows the application layer to find out about the options, costs, and capabilities available from the network layer in a suitably high level but actionable format. However, it remains to specify an interface for the application layer to communicate its usage intent to the network layer and possibly make firm commitments for scarce network resources. Before delving into this interface we first look a bit at what happens behind the scenes in high bandwidth networks.

4.4.1. IP Bypass/Traffic Engineering

There are various ways to alter the path that IP flows take through a network. Two IETF standard ways are via DiffServe [16] and MPLS-TE [17]. Both mechanism start with IP packet classification but in MPLS-TE a packet belonging to a flow matching an MPLS forwarding equivalence class (FEC) will be "pulled" from normal IP packet forwarding and place in a MPLS tunnel, known as a label switched path. It will then be forwarded on via MPLS mechanisms bypassing the IP layer until it "pops" out of its MPLS tunnel and rejoins the IP forwarding world (hopefully much closer to its intended destination and making better use of network resources along the way).

In the SONET, SDH, G.709, and WDM world a similar process can take place, but is known by the term "grooming" [11],[12]. In both cases network resources including bandwidth, equipment, and power can be significantly optimized by essentially setting up "express lanes" at a lower layer in the network's protocol stack. Note that with optical transport networks there can many layers below "layer 2", i.e., one can think of the "physical" layer as possibly consisting of a number of different sub-layers.

If the application layer by knowing its usage patterns or required network usage can let the network its needs then IP/Optical bypass can be more readily be performed on a dynamic basis, particularly if the network has a GMPLS infrastructure. The application layer should not need to know the specifics of how the IP bypass occurs, e.g., via MPLS, OpenFlow, or other standard or proprietary techniques.
4.4.2. High Bandwidth Reservation/Recovery Interface

As previously stated the application layer should not be exposed to the details of networking mechanisms that will provide the bandwidth and QoS guarantees. Hence the application layer would specify its demands in terms of IP flows such as when specifying an MPLS FEC. It is for further study whether some IaaS applications may want to deal with layer 2 (Ethernet) flows rather than IP. In either case the basic principles would be the same. Note that a bandwidth reservation interface such as this could also be used when application layer is seeking network help in dealing with disaster recovery and business continuity.

A number of current protocols come close to the features desired of such an interface, but none are completely appropriate. A short summary follows:

(a) PCE: The PCE interface takes requests for connections with various optimization conditions supported. PCEs though return the computed paths to the requester, something of which is undesired in our reservation interface. Note that PCE is built directly on TCP.

(b) UNIs (GMPLS and OIF): UNIs provide RSVP-TE based signaling interfaces for connection requests at a particular layer. Such interfaces expect the requester to know something about the network layers being utilized. Typically, if these are used, they are used between access and core network equipment.

(c) Cloud IaaS interfaces for reserving instances: These are typically RESTful or XML-RPC type interfaces. With these interfaces compute, storage and other IaaS related resources are requested (setup/teardown).

We note that currently such an interface is currently out of the scope of ALTO or any current IETF working group. One reason to consider this within ALTO is the tight coupling between the network information (PIDs, network map, cost map, capacitated graph) and requests that would be made by the application layer. In the high bandwidth case both query and reservation have similar security/privacy requirements.

5. Conclusion

In this draft 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. While in the second use case where data
centers were communicating with each other bandwidth usage was already significant enough to warrant the use of private line/LAN type of network services.

Both use cases result in optimization problems that trade off computational versus network costs and constraints. Both featured scenarios where advanced reservation, on demand, and recovery type service interfaces could prove beneficial. In the later section of this document we showed how ALTO concepts [1] and the ALTO protocol could be used and extended to support joint application network optimization for large network bandwidth consuming applications.

6. Security Considerations

TBD

7. IANA Considerations

This informational document does not make any requests for IANA action.

8. References

8.1. Informative References

Author’s Addresses

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

Young Lee  
Huawei Technologies  
5340 Legacy Drive, Building 3  
Plano, TX 75024  
USA  
Phone: (469) 277-5838  
Email: leeyoung@huawei.com

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, THE IETF, AND THEIR CONTRIBUTORS DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, ANY WARRANTY THAT THE USE OF THE DOCUMENT WILL NOT INFRINGE A THIRD PARTY’S RIGHTS.

Bernstein & Lee         Expires September 12, 2012 [Page 22]
Acknowledgment

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

This draft considers an approach to the optimization of the traffic generated by the overlay (peer-to-peer) applications, where some information on inter-AS (Autonomous System) paths is obtained with the usage of dedicated communication scheme known as inter-ALTO communication.

The large amount of network traffic generated by overlay applications requires effective management. This traffic traverses inter-AS links and thus generates substantial costs for the operators and poses problems with overload on the external and internal links. The traffic is not time-stable as the peers connect and disconnect very often. Additionally, when the overlay traffic is observed on inter-AS links, it can be seen that sources and destinations change in a very short period of time. The ALTO (Application-Layer Traffic Optimization) service provides the information that enables more efficient management of the overlay traffic. Such applications can use the information to perform better-than-random peer selection. The ALTO servers are responsible for a pre-selection procedure; the final selection is done by overlay clients and then the ALTO protocol conveys network information to applications. To be credible, this information should be as close to real network situation as possible. However, some types of data are not hold by an operator, but they should be gained on the basis of the additional information exchange with other AS operators. This document presents rationale for the need of introduction of the inter-ALTO communication.
Table of Contents

1. Introduction .................................................. 4

2. Definitions .................................................. 5

3. The Problem and Motivation ................................. 6
   3.1. Route Asymmetry ....................................... 8
   3.2. Many ASes within One ISP ............................. 8
   3.3. Different Types of Business Relations ............... 9
   3.4. Congestion Avoidance ................................ 9
   3.5. Proximity Awareness .................................. 9
   3.6. Remote ISP’s Preference .............................. 10
   3.7. Coordination of ISPs’ Policies ....................... 10
   3.8. Sensitivity of Topology Information ................. 11
   3.9. Outdated Information ................................. 11
   3.10. Mobile Networks .................................... 12
   3.11. Route Aggregation .................................. 12

4. Usage of the Mechanisms Offered by the ALTO Protocol .... 12

5. Security Considerations ...................................... 13

6. IANA Considerations ......................................... 14

7. Acknowledgements ............................................ 14

8. References .................................................. 15
   8.1. Normative References ................................. 15
   8.2. Informative References ............................... 15

Authors’ Addresses ............................................. 15
1. Introduction

This document describes the rationale for a communication to be used between ALTO servers located in different autonomous systems (AS). Such an inter-ALTO communication extends the ALTO service capabilities and provides additional information on remote peers, i.e., peers located in other ASes. To make the consideration more clear we distinguish local AS and remote ASes. Local AS is the one from which perspective we describe the communication. Local peers are located in the local AS and are served by a local ALTO server. On the contrary, all other peers are located in remote ASes. Those peers are referred to as remote and are served by remote ALTO server. This basic terminology adheres to majority of considerations in this document.

The motivation for the ALTO service as discussed in the ALTO problem statement [RFC5693] focuses on the overlay traffic optimization based on information gathered from the Internet Service Provider (ISP) domain, i.e., an Autonomous System (AS). Due to the suggested approach, information on the AS internal topology and some routing information obtained from the global Internet (the BGP routing tables) may be used for the peer selection procedures. The data transfer cost can be also incorporated. However, there are some parameters which can be used for the better peer selection mechanism, but they are not available in the local AS and must be obtained from outside, i.e., from a remote AS. For example, the BGP routing information available in the AS identifies only the upstream traffic. The information about the downstream traffic is not present or is incomplete and the full BGP information for this traffic could be obtained from the remote AS containing the subnetwork where the peer sending downstream traffic is attached. In order to obtain such data, we propose the inter-ALTO communication.

It is assumed that the ALTO servers are deployed in the local and remote ASes. The ALTO server located in the client AS can request desired information from the ALTO server which is located in the remote AS. Each server is managed by a respective ISP. Each ISP decides what type of information can be exposed to the requesting party. The ALTO server responds with the type of information that was previously agreed to exchange. Each local ALTO server has to discover the address of the remote ALTO server before starting the communication. The discovery procedure may use the IP addresses of remote peers for the establishment of IP addresses of remote ALTO servers. The detailed analysis of this functionality is out of scope of this document.

The information delivered by remote ALTO servers may be used by a local ALTO server to perform advanced rating/ranking procedure of
peers. The general idea is as follows.

1. A peer receives a list of other peers from a tracker, i.e., a list of potential candidates to communicate with.

2. A peer uses the ALTO protocol [I-D.ietf-alto-protocol] to send the list of peers to its local ALTO server.

3. Local ALTO server obtains additional information on remote peers by communicating respective remote ALTO servers. If sufficient information is available locally and the inter-ALTO communication is not needed, this step is omitted.

4. Using ISP specific policies and values of parameters associated with remote peers the local ALTO server performs rating/ranking procedure.

5. Sorted/rated list of peers is sent back to the peer with usage of the ALTO protocol.

The requirements, syntax and detailed operation of the inter-ALTO communication as well as the rating/ranking procedure is out of scope of this document.

2. Definitions

In the scope of this document we use all the definitions specified in the Section 2 of ALTO problem statement [RFC5693]. Moreover, the following terms have the special meaning.

Local Peer: A peer which belongs to the same Autonomous System to which the ALTO client belongs.

Remote Peer: A peer which belongs to another Autonomous System than the one to which the ALTO client belongs.

Local AS: The Autonomous System to which the ALTO client belongs.

Remote AS: An Autonomous System to which a remote peer belongs.

Local ALTO Server: The ALTO server serving the ALTO client and the local peers.

Remote ALTO Server: An ALTO server serving remote peers.
3. The Problem and Motivation

ALTO server optimization capabilities are limited by the fact that they use information available locally only. It can be shown that more information on remote peers, a routing path, or remote ISP preferences would be useful. The data from remote ASes obtained by the external interface as shown in Figure 1 of the ALTO protocol draft [I-D.ietf-alto-protocol] may have a substantial significance for the management of overlay traffic (e.g. with respect to peer rating, ranking, or the choice of the best peers). The suggested approach to deliver these types of information is defined in the inter-ALTO communication discussed in this document.

The ALTO service may also be used by the client-server applications, supporting the best choice of the mirror servers. If some mirror servers are in other ASes than the client’s AS, some information about the network conditions in the remote ASes may be delivered only by the ALTO servers localized in these ASes. Both clients and mirror servers may communicate with their local ALTO servers for delivering traffic optimization parameters. As an ALTO client communicates only with its local ALTO server, the information from remote ASes is accessible only via inter-ALTO communication.

The ALTO-based traffic optimization may be also used in the context of the Content Delivery Networks (CDNs) [I-D.jenkins-alto-cdn-use-cases]. The draft by Niven-Jenkins et al. shows how CDNs may benefit from the information provided by the ALTO protocol. Local information, however, may be not sufficient to optimize the request routing process. The information gathered from ALTO servers in other domains may be needed.

The basic ALTO architecture presented in Figure 1 of the ALTO protocol draft [I-D.ietf-alto-protocol] defines the external interface used to communicate with other information sources like remote ALTO servers. The ALTO Protocol draft, however, does not define what information should be exchanged between ALTO servers to optimize the traffic. The inter-ALTO communication proposed by the current document implements the external interface as defined by the ALTO protocol draft.
The architecture of the Inter-ALTO communication is shown in Figure 1. Both ALTO servers gather the information from their information sources like routing protocols, provisioning policies, or dynamic network information sources. The local ALTO server needs to communicate with a remote ALTO server to obtain information which is available only at the entities in the remote AS.

In particular, the following key aspects motivate the proposal of the inter-ALTO communication.

- Route asymmetry.
- Different types of business relations.
- Congestion avoidance.
- Proximity awareness (distance to the remote AS), e.g.:
  - number of inter-AS hops;
* delay (RTT).
- Remote ISP’s preference.
- Coordination of ISPs’ policies.
- Outdated information.

3.1. Route Asymmetry

The communication between two ASes does not need to follow the same path in the upstream and downstream direction. It was shown that about 29% of paths between AS pairs in the Internet are fully symmetric, i.e., upstream and downstream traffic follows exactly the same path [ICC.optimal]. In 51% of cases the number of inter-AS hops is different for the upstream and downstream direction. Additionally, in 50.5% of all path pairs a neighbor AS for upstream and downstream paths are different.

The ALTO server can obtain routing information locally (e.g. from BGP routers) and can determine the upstream path. Information about the downstream path is usually not easily available. Some additional routing information can be obtained from Looking Glass Servers, but not all ASes provide them. The inter-ALTO communication provides the ability to exchange the relevant information between ALTO servers. Especially, the downstream path can be reliably determined using the information provided by remote ALTO server. In the light of route asymmetry in the Internet such information appears to be necessary for a better optimization of a peer rating/ranking algorithm, as assumption that the inter-AS routes follow symmetrical paths can give not only sub-optimal, but misleading and, in effect, harmful results.

3.2. Many ASes within One ISP

An ISP may possess a complex topology network composed of many autonomous systems. Current ALTO specification allows for deployment of independent ALTO servers in each AS. In such a case the overlay traffic management performed by the ALTO server is restricted to a single AS since cost maps have a local meaning. An ISP operating a multi-AS network may be interested in managing the traffic in the whole administrative domain in a consistent and coordinated manner. The information possessed by a single ALTO server is insufficient. To obtain a complete knowledge on the multi-AS network a communication between ALTO servers is needed. As a result, local cost maps originating from different autonomous systems may be coordinated. A uniform cost map reflecting the whole network structure may be created and distributed between ALTO servers.
3.3. Different Types of Business Relations

Two basic business relations between ISPs may be distinguished.

When two ISPs agree to exchange the traffic without any charge, such a relation is called peering. The inter-domain link between the respective ASes is also called a peering link. Usually, there is no charge if the difference between traffic volumes passing such a link in different directions does not exceed a previously agreed limit.

The other case occurs when one ISP serves as a network provider to another ISP (e.g. relation between tier 2 and tier 3 ISPs). In such a case one ISP (acting as a customer) has to pay the other ISP (acting as provider) for the traffic sent over the inter-AS link connecting them. The real monetary cost of the traffic volume exchanged on such a link depends on agreements between ISPs. In general, some links may be considered as cheaper or more expensive.

AS may be connected to many other ASes with various agreements. The cost of the inter-AS traffic transfer may differ depending on which neighbor AS the path passes. For this reason an ISP may prefer that its own customers exchange data with remote peers located in such ASes that the path directed to them passes cheaper links. The ALTO server may sort peers taking into account these criteria. To receive almost complete information on routing paths to and from different remote domains the information provided by remote ALTO server using inter-AS communication may be helpful.

3.4. Congestion Avoidance

A peer rating/ranking procedure may also take into account the congestions on inter-AS links. An ISP is able to monitor queues on its inter-domain links and assign metrics indicating the buffer occupancy or bandwidth utilization. These metrics can express percentage use of buffers or bandwidth on a particular inter-AS link. If one inter-domain link is congested it is desirable to promote peers reachable through lightly loaded links. Again, information provided by the remote ALTO server would support such optimization. The aim of the inter-ALTO communication is not to replace the existing congestion avoidance mechanisms. The idea is to support the present mechanism by the exchange of parameters describing the load on the inter-AS links.

3.5. Proximity Awareness

For a set of reasons (e.g. the performance of an application) the ALTO server may suggest its customers to connect to remote peers located in its proximity. The simplest measure of proximity is the...
number of inter-AS hops. As indicated above, due to the route asymmetry, the number of hops may significantly differ between the upstream and downstream paths. Such information for the downstream path may be provided by the remote ALTO server. A more advanced metric of proximity can be found in the delay that can be approximated by exchanging messages between ALTO servers. The ALTO servers can be equipped with an application-layer ping functionality which only operates between ALTO servers. By exchanging special packets prepared by the ALTO servers, these servers can estimate delay and packet loss.

3.6. Remote ISP’s Preference

If two ISPs agree on a cooperation, the remote ALTO server may provide its preference parameters (remote preference parameters) indicating which peers are better from the point of view of the remote ISP. For instance, the AS in which the remote ALTO server is located may possess two subnetworks connected to the operator’s core network by distinct links. It may happen that a connection to one of the subnetworks is cheaper than the other. The remote operator may prefer connections through cheaper link, so peers located in the subnetwork transferring data via this cheaper link are preferred.

The remote preference parameter may be also used when a remote ISP wants to suggest peers which are connected to the Internet through access links of higher capacity. This way, the remote ALTO server, without exposing the exact values of access link bandwidth, may indicate peers with higher throughput. The remote preference parameters have only local meaning, i.e., their values are comparable for peers located in the same AS only.

If a remote ISP does not want to reveal numerical values of network parameters related to its peers (such information might be considered as confidential) the remote ALTO server may perform a rating/ranking procedure and assign priority parameter to its peers. The rating/ranking criteria may remain hidden for the requesting local ALTO server.

3.7. Coordination of ISPs’ Policies

Operators may have an incentive to coordinate their efforts in order to decrease transfer costs on inter-AS links or improve quality experienced by peers, i.e., coordinate their peer rating/ranking strategies. This way, operators may avoid contradictory strategies resulting in inefficiency of rating/ranking algorithms. Operators may agree to promote each other’s peers.

For example, it may happen that operator A wanting to decrease
traffic on one of its links discourages its own peers from communicating with peers located in operator B’s domain. On the other hand, operator B would consider peers located in a domain of operator A as very attractive for its own peers. As a result, rating/ranking procedures performed by respective ALTO servers give contradictory results what may decrease the effectiveness of these procedures. To avoid such a situation, the inter-ALTO communication is needed.

Another example of a usefulness of coordination of policies is clustering of ASes. Recent studies [IJNM.unfairness] have shown that locality promotion might be ineffective or even harmful if used in AS with small number of peers. A proposed solution is to create a cluster of two or more ASes. Then ALTO servers serving different ASes in the cluster treat all peers located in the cluster as if they were in a single AS. In other words, from a point of view of locality promotion algorithm all peers located in the cluster are local, regardless of their home AS.

3.8. Sensitivity of Topology Information

The minimum information that the remote AS provides to the local ALTO server via the inter-ALTO communication may be the number of inter-AS hops and the number of the local AS’s neighbor in the downstream path (the full downstream AS_PATH may be not exchanged). Such information does not reveal any sensitive information neither on the ISP internal topology details nor remote AS connections with other ASes, but does provide basic and very useful information for the local ALTO server.

3.9. Outdated Information

It is expected that some information (parameters) from routing protocols that will be used in the rating/ranking procedures may outdate. Also information related to the network performance is constantly changing. Therefore, the information obtained from the remote AS requires updates. This updates may be generated on request (pull mechanism), on event base schema or periodically (push mechanism). The inter-ALTO communication should be equipped with mechanisms for updates. The need for the present information describing network conditions and some routing parameters are arguments for introducing specific protocol for the communication between ALTO servers.

3.10. Mobile Networks

The inter-ALTO communication may be very useful for mobile network operators and content providers serving mobile clients. An ALTO server may recognize mobile clients and properly assign them to PIDs.
Some information about the mobile network resources gathered from mobile network nodes located in different networks should be exchanged between operators for better than random peer selection. ALTO servers should possess information which allows to make proper peer selection, taking into account, e.g., the mobile network load (including the load in the radio access network and in the circuit- and packet-switched domains).

After collecting the load information, the ALTO server may assign priorities. These priorities may exemplify the load in some parts of the radio access network. Via the inter-ALTO communication, the priorities may be passed to the other operator's networks where other clients are located. Relying on this information, the ALTO server may optimize the connections between clients.

3.11. Route Aggregation

The BGP protocol allows the aggregation of specific routes into one route. In such a case the aggregate route is advertised. The full path is either lost completely or the AS set information is available. In the latter case only the set of ASes behind the aggregating router is known but the detailed information about the routing path, including AS sequence and AS-hop count, is lost. From the overlay traffic optimization point of view the knowledge on ASes located behind aggregating router and the number as well as sequence of inter-AS hops may be useful, e.g., because of route asymmetry problem described earlier (Section 3.1). The solution for this problem is information exchange between ALTO servers located in ASes ahead and behind the router aggregating routes.

4. Usage of the Mechanisms Offered by the ALTO Protocol

The basic ALTO protocol architecture allows an ALTO server to communicate with a third party through the external interface. The inter-ALTO communication may use some functionalities offered by the ALTO protocol [I-D.ietf-alto-protocol].

Server Information Service: This service defined by the ALTO protocol may be extended in order to provide information about server’s ability to cooperate with other ALTO servers. Thanks to this service, the other ALTO servers may acquire the information about available parameters and their definitions. These parameters may be used by cooperating ALTO servers for the peer rating/ranking procedures. The access for this service may be restricted. Some information may be accessible only by the privileged ALTO servers after the successful authentication.
ALTO Information Services: These services have been defined to provide the query information services for ALTO clients. All the information delivered by these services has local meaning. This information is related to the locally defined parameters describing a particular ISP’s network. Some part of this information managed by a remote ALTO server may be useful for the requesting local ALTO server. The requesting ALTO server obtains this information via inter-ALTO communication. After receiving the response, the local ALTO server has to perform some calculations, scaling, merging, or adaptation of the received parameters. In this way the local ALTO server may conform to both its internal network topology and measurements, and the external ones. However, it should be stressed that the ALTO Information Services is designed for communication between ALTO clients and servers, not for the inter-ALTO communication.

Network Map: This structure is defined by the ISP and reflects the internal structure of the ISP network. This structure has only a local meaning and, generally, it is not unique for all entities within the Internet. A particular network map can be used by different operators. The requesting ALTO server usually has to perform some prediction of the external topology on its own. The ALTO server has to apply its own rules and definitions. The PIDs, defined in the remote ALTO server, have to be mapped on the PID structure defined in the local AS.

Cost Map: This structure also has the local meaning. The local ALTO server may receive the network map and the cost map from a remote ALTO server. These costs may require recalculation in order to unify the cost measures in the local AS. After these operations, if it is needed, the rating/ranking procedure can be performed.

5. Security Considerations

The communication between ALTO servers requires authentication and authorization procedures. In some cases it may require establishment of the secured tunnels between the partner ALTO servers. The minimum security requirements for the inter-ALTO communication is out of scope of this document.

The inter-ALTO communication allows ALTO servers to exchange any parameters which improve the performance of the overlay traffic, or, generally, allows them to manage overlay traffic. In order to achieve this results a group ISP may exchange sensitive data, the exchanged parameters may be confidential. They should not be
accessible by a third party, e.g., some other ISPs or peers.

An ISP may have its own policy how organize the overlay traffic and this policy may use a number of parameters during the evaluation procedure. The policy result may be delivered to peers in many ways. It can take the form of a sorted peer list without any parameters, a sorted list with some parameters which are derived from the parameters exchanged in the inter-ALTO communication, or raw exchanged parameters. ISPs may have an incentive not to expose these parameters in the raw form to peers. The mentioned sensitive parameters require applying a higher level of the security procedures.

In order to keep the exchanged parameters confidential it may be reasonable to keep the communications between peers and ALTO server from communication among ALTO servers by the protocol differentiation separated. Different security procedures may be easier to manage if these communication procedures take the form of two distinct protocols. This protocol separation allows to define mechanisms which are specific for the inter-ALTO communication only. The protocol should not allow to use this mechanism by overlay peers. The set of procedures for the inter-ALTO communication is expected to be separated from the client ALTO communication and this can be achieved by distinct protocols.

6. IANA Considerations

This document has no actions for IANA.

7. Acknowledgements

The authors would like to thank following people for the valuable discussions (in alphabetical order):

- Piotr Cholda (AGH University of Science and Technology)
- Miroslaw Kantor (AGH University of Science and Technology)
- Jan Medved (Juniper)
- Stefano Previdi (Cisco)
- Robert Varga (Pantheon)

This work has been partially supported by the EU through the ICT FP7 Project SmoothIT (FP7-2007-ICT-216259).
8. References

8.1. Normative References

[I-D.ietf-alto-protocol]
Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol",
draft-ietf-alto-protocol-09 (work in progress), June 2011.

[ICC.optimal]


8.2. Informative References

[I-D.jenkins-alto-cdn-use-cases]
Niven-Jenkins, B., Watson, G., Bitar, N., Medved, J., and S. Previdi, "Use Cases for ALTO within CDNs",
draft-jenkins-alto-cdn-use-cases-01 (work in progress), June 2011.

[IJNM.unfairness]

Authors’ Addresses

Zbigniew Dulinski
Jagiellonian University
ul. Reymonta 4
Krakow 30-059
Poland

Phone: +48 12 663 5571
Fax: +48 12 633 4079
Email: dulinski@th.if.uj.edu.pl
Use Cases for ALTO within CDNs
draft-jenkins-alto-cdn-use-cases-02

Abstract

For some time, Content Distribution Networks (CDNs) have been used in the delivery of some Internet services (e.g. delivery of websites, software updates and video delivery) as they provide numerous benefits including reduced delivery cost for cacheable content, improved quality of experience for end users and increased robustness of delivery.

In order to derive the optimal benefit from a CDN it is preferable to deliver content from the servers (caches) that are "closest" to the End User requesting the content, where "closest" may be as simple as "geographical or network distance" combined with CDN server load within a location, but may also consider other more complex combinations of metrics and CDN or Network Service Provider (NSP) policies.

There are a number of different ways in which a CDN may obtain the necessary network topology and/or cost information to allow it to serve End Users from the most optimal servers/locations, such as static configuration, passively listening to routing protocols directly, active probing of underlying network(s), or obtaining topology and cost by querying an information service such as the ALTO map & cost services.

This document describes the use cases for a CDN to be able to obtain network topology and cost information from an ALTO server(s).

Status of this Memo

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

1. Introduction ...................................................... 4
1.1. Terminology .................................................... 5
2. CDN overview ..................................................... 5
3. CDN & ALTO Use Cases ............................................. 7
   3.1. Exposing NSP End User Reachability to a CDN .......... 8
   3.2. Exposing CDN End User Reachability to CSPs .......... 9
   3.3. CDN deployed within a Broadband network ............ 10
   3.4. CDN delivering Over-The-Top of a NSP’s network .... 11
   3.5. CDN acquiring content from multiple upstream sources (Origins) ......................... 11
   3.6. Additional Use Cases ....................................... 12
4. IANA Considerations .............................................. 13
5. Security Considerations ......................................... 13
6. Contributing Authors ............................................ 13
7. Acknowledgements ................................................ 13
8. Normative References ............................................ 13
Authors’ Addresses .................................................. 14
1. Introduction

For some time, Content Distribution Networks (CDNs) have been used in the delivery of some Internet services (e.g. delivery of websites, software updates and video delivery) as they provide numerous benefits including reduced delivery cost for cacheable content, improved quality of experience for end users and increased robustness of delivery.

A CDN typically consists of a network of servers often attached to Network Service Provider (NSP) networks. The point of attachment is often as close to content consumers and peering points as economically or operationally feasible in order to decrease traffic load on the NSP backbone and to provide better user experience measured by reduced latency and higher throughput.

As the volume of video and multimedia content delivered over the Internet is rapidly increasing and expected to continue doing so in the future, existing CDN providers are scaling up their infrastructure and many NSPs are deploying their own CDNs. The result of such deployments is typically that more CDN servers are being deployed within NSP networks and those CDN servers are being deployed in locations that are "deeper" (i.e. geographically closer to the NSP’s End Users) than was previously the case.

In order to derive the optimal benefit from a CDN it is preferable to deliver content from the servers (caches) that are "closest" to the End User requesting the content, where "closest" may be as simple as "geographical or network distance" combined with CDN server load within a location, but may also consider other more complex combinations of metrics and CDN or NSP policies.

When CDN servers are deployed outside of an NSP’s network or in a small number of central locations within an NSP’s network a simplified view of the NSP’s topology or an approximation of proximity is typically sufficient to enable the CDN to serve End Users from the optimal server/location. As CDN servers are deployed deeper within NSP networks it becomes necessary for the CDN to have more detailed knowledge of the underlying network topology and costs between network locations in order to enable the CDN to serve End Users from the most optimal servers for the NSP.

There are a number of different ways in which a CDN may obtain the necessary network topology and/or cost information to allow it to serve End Users from the most optimal servers/locations, such as static configuration, passively listening to routing protocols directly, active probing of underlying network(s), or obtaining topology and cost by querying an information service such as the ALTO
The rest of this document describes the use cases for a CDN to be able to obtain network topology and cost information from an ALTO server(s).

1.1. Terminology

The following terms are taken from [I-D.ietf-cdni-problem-statement] and repeated here for completeness.

Content Distribution Network (CDN) / Content Delivery Network (CDN): Network infrastructure in which the network elements cooperate at layers 4 through layer 7 for more effective delivery of Content to User Agents. Typically a CDN consists of a Request Routing system, a Distribution System (that includes a set of Surrogates), a Logging System and a CDN control system.

Content Service Provider (CSP): Provides a Content Service to End Users (which the End Users access via a User Agent). A CSP may own the Content made available as part of the Content Service, or may license content rights from another party.

End User (EU): The 'real' user of the system, typically a human but maybe some combination of hardware and/or software emulating a human (e.g. for automated quality monitoring etc.)

Network Service Provider (NSP): Provides network-based connectivity/services to Users.

Surrogate: A device/function that interacts with other elements of the CDN for the control and distribution of Content within the CDN and interacts with User Agents for the delivery of the Content.

User Agent (UA): Software (or a combination of hardware and software) through which the End User interacts with the Content Service. The User Agent will communicate with the CSP's Service for the selection of content and one or more CDNs for the delivery of the Content. Such communication is not restricted to HTTP and may be via a variety of protocols. Examples of User Agents (non-exhaustive) are: Browsers, Set Top Boxes (STB), Dedicated content applications (e.g. media players), etc.

2. CDN overview

This section provides a high level and simplified overview of the operation of a CDN to help put the ALTO & CDN use cases into context.
A typical CDN consists of a number of functional components, however in the context of ALTO three functional components are of interest: The Request Routing function, the Surrogate (i.e. caching) function and the Origin function.

The Request Routing function within a CDN is responsible for receiving content requests from User Agents, obtaining and maintaining necessary information about a set of candidate Surrogates, and for selecting and redirecting the User Agent to the appropriate Surrogate.

The Surrogate function interacts with other elements of the CDN for the control and distribution of Content within the CDN and interacts with User Agents for the delivery of the Content.

The figure below shows a high level call flow showing the interaction between a User Agent, Request Router and Surrogate for the delivery of content in a single CDN.

1. The User Agent makes an initial request to the CDN. Depending on the type of content being delivered and the configuration of the CDN this request may be an application (e.g. HTTP, RTMP, etc.) level request directly from the User Agent or may be a DNS request via the User Agent’s assigned DNS proxy.
2. The Request Router selects an appropriate Surrogate (or set of Surrogates) based on the User Agent’s (or its proxy’s) IP address, the Request Router’s knowledge of the network topology and reachability cost between CDN caches and end users, and any additional CDN policies.
3. The Request Router responds to the UA’s initial request with an appropriate response containing a redirection to the selected cache, for example by returning an appropriate DNS A/AAAA record, a HTTP 302 redirect, etc.

4. The User Agent uses the information provided in the Redirection Response to connect directly to the Surrogate and request the desired content.

5. If CDN policy allows the User Agent to receive the requested content, the Surrogate delivers the content to the User Agent.
   A. [Not Shown] If the Surrogate does not have a copy of the requested content then it obtains it from the appropriate Origin Server.

   Note: A Surrogate may not communicate with the Origin directly and instead obtain the requested content from other surrogates or caching layers in the CDN hierarchy. The details of how content requests filter through the CDN hierarchy to the Origin are internal to a specific CDN and are out of scope of this document.

3. CDN & ALTO Use Cases

   The primary use case for ALTO in a CDN context is to improve the selection of a CDN Surrogate or Origin. The CDN makes use of an ALTO server to choose a better CDN Surrogate or Origin than would otherwise be the case. In its simplest form an ALTO server would provide an NSP with the capability to offer a service to a CDN which provides network map and cost information that the CDN can use to enhance its surrogate and/or Origin selection.

   Although it is possible to obtain raw network map and cost information in other ways, for example passively listening to the NSP’s routing protocols, the use of an ALTO service to expose that information may provide additional control to the NSP over how their network map/cost is exposed. Additionally it may enable the NSP to maintain a functional separation between their routing plane and network map computation functions. This may be attractive for a number of reasons, for example:

   o The ALTO service could provide a filtered view of the network and/or cost map that relates to CDN locations and their proximity to end users, for example to allow the NSP to control the level of topology detail they are willing to share with the CDN.

   o The ALTO service could apply additional policies to the network map and cost information to provide a CDN-specific view of the network map/cost, for example to allow the NSP to encourage the CDN to use network links that would not ordinarily be preferred by a Shortest Path First routing calculation.
The routing plane may be operated and controlled by a different operational entity (even within a single NSP) to the CDN and the ALTO service could provide a layer of separation because:
* The CDN is not able to passively listen to routing protocols.
* The NSP is not willing to allow the CDN to passively listen to routing protocols, e.g. because the NSP is concerned the CDN may inadvertently interfere with the routing plane or because the routing plane and the CDN are operated by different operational entities/groups (including different entities within the same NSP).

The use cases in this document are not necessarily specific as to the relationship between the commercial/operational entity that "owns" the ALTO service and the commercial/operational entity that "owns" the CDN service as it is assumed that such relationships will be deployment specific. Although the ownership of each service may affect the level of topology detail that the ALTO service will be permitted to expose, it is assumed that the general requirements a CDN places on the ALTO service should not change provided that the ALTO server is able to expose sufficient topology for the CDN to make appropriate surrogate and/or Origin selection decisions.

In general, the ALTO service is expected to be operated by an entity or entities that wish to optimize or otherwise influence request routing decisions. Some, non-exhaustive, examples of such entities are:

* The entity that operates the CDN’s underlying network (e.g. the "CDN deployed within a Broadband network" described in Section 3.3).
* An NSP that wishes to optimize over-the-top content delivery from a CDN that is deployed outside of its network (e.g. the "CDN delivering Over-The-Top of a NSP" described in Section 3.4).
* An NSP (that may or may not operate a CDN) or a CDN that wishes to advertise which End Users are reachable via its network/CDN (e.g. the exposing "End User reachability" use cases described in Section 3.1 and Section 3.2).

The following sections outline some specific, non-exhaustive, example use cases, which are subsets of the primary use case outlined above but applied to specific usage examples to demonstrate how a CDN could make use of ALTO services.

### 3.1. Exposing NSP End User Reachability to a CDN

In order for a Request Router to be able to make surrogate selection decisions, the Request Router needs to have information on which End User IP subnets are reachable via its network or network locations. The granularity of location information required depends
on the specific deployment of the CDN relative to the End Users. For example, an Over-The-Top CDN whose surrogates are deployed only within the Internet "backbone" may only require knowledge of which End User IP subnets are reachable via which NSPs’ networks, whereas a CDN deployed within a particular NSP’s network requires a finer granularity of knowledge, i.e. which End User IP subnets are reachable via which regions within that NSP’s network.

Such reachability information is often available via dynamic routing protocols, however it is likely that in a number of deployment scenarios that peering of the routing plane of the network with a CDN would be deemed unacceptable (e.g. where the CDN is operated by an entity other than the NSP(s) operating the underlying network).

Provided that some common mapping between ALTO PIDs and network locations (or entire networks) is known to both the NSP and the CDN, the network map services offered by ALTO could be used to expose which End User IP subnets are reachable via a particular network or network locations in order to export End User reachability to a Request Router to enable the NSP to expose End User reachability while also giving the NSP the ability to control the granularity of any End User reachability to network location mapping while also avoiding routing plane peering between the NSP and the CDN.

3.2. Exposing CDN End User Reachability to CSPs

This use case is similar to the use case described in Section 3.1 however in this case it is the CDN that wishes to expose which End User IP subnets the CDN is capable of delivering services to.

In some deployments a particular CDN may not have reachability to (or may not wish to offer services to) every End User IP subnet reachable via the global Internet, for example because the CDN is only deployed within certain networks or geographic regions and the CDN is either unable (due to lack of reachability) or unwilling (due to cost or policy) to serve all End Users reachable via the global Internet.

The reachability offered by a particular CDN may not include all the End User IP subnets that a particular CSP requires in order to serve all of that CSP’s customers and therefore if the CSP wishes to make use of the services offered by a CDN that can only serve a subset of their customers the CSP must have knowledge of which End User IP subnets a particular CDN is able to serve, so that they can select an appropriate CDN to use to deliver their service to particular subsets of their customers.

In such cases, the network map services offered by ALTO could be used to expose to a CSP which End User IP subnets are reachable via a
particular CDN. In the case where the CDN is operated by an NSP using ALTO in this way could also enable the NSP to separate the exposure of End User subnets reachable via their CDN from those reachable via their underlying network.

3.3. CDN deployed within a Broadband network

In this use case an NSP is providing Broadband services to its customers and has deployed a CDN within its Broadband network to alleviate the cost and/or improve the User Experience of content services for its Broadband customers.

The topology of Broadband access/backhaul networks is often much more constrained than metro/core networks. If CDN surrogates are deployed within the access/backhaul network, for a given set of End Users, the NSP is likely to want to utilise the surrogates deployed in the same access/backhaul region as those End Users in preference to surrogates deployed within the metro/core or within other access/backhaul regions.

It is common for Broadband subscribers to obtain their IP addresses dynamically and in many deployments the IP subnets allocated to a particular access/backhaul region can change relatively frequently. For example new IP subnets are added as the subscriber base grows, IP subnets are moved from one Broadband product in the NSP’s portfolio to another as customers migrate in order to optimise the NSP’s IP address utilisation, or they are simply moved as part of IP address management, etc.

Additionally, in certain cases, CDN surrogates deployed in a particular network region may become overloaded, leading to the CDN selecting alternative surrogates in a different region of the network for content delivery. If this occurs, an NSP may wish to influence such a decision, for example because the NSP would prefer a surrogate to be selected that is deployed in the the next best (cost-wise to the NSP) location.

In order to meet the NSP’s objective of utilising their CDN to constrain access/backhaul costs and/or improve User Experience it is important that the CDN is able to select the most appropriate surrogate for a given set of End User IP subnets. Although the network topology is often reasonably static, in networks where the IP subnets allocated to a Broadband region are changing relatively frequently, static configuration of End User IP Subnets to CDN surrogates is possible but some NSPs may consider the operational burden of having to update such static configuration too high and would prefer the CDN to be able to dynamically obtain network map and cost information.
The NSP could make use of an ALTO service to expose a cost mapping/ranking between End User IP subnets (within that NSP’s network) and CDN surrogate IP subnets/locations to meet its requirements while avoiding static configuration or direct integration of the CDN into its IP routing plane and to avoid the CDN being required to implement network layer routing computations.

3.4. CDN delivering Over-The-Top of a NSP’s network

In this use case a CDN is deployed within one or more NSPs’ networks but is delivering content "Over-The-Top" into another NSP’s network (which we will call NSP Z) where the CDN is not deployed.

The CDN is unlikely to have direct visibility of NSP Z’s network topology and may have a choice of entry points into NSP Z’s network from which it could serve content to NSP Z’s End Users. For example because NSP Z has direct peering links with the CDN in a number of locations or NSP Z has transit and/or peering relationships with several other NSPs where the CDN is deployed. NSP Z may wish to influence the locations from which the CDN serves content based on some factor(s) that it does not wish to expose directly or that might change over time. For example the available transit/peering capacity in different locations, the cost of connectivity to different locations, etc.

For example, a CSP is using NSP A’s CDN and another NSP (NSP Z) has peering with NSP A in Los Angeles and New York. NSP Z would like to influence which peering location NSP A’s CDN delivers content out of for NSP Z’s end users by using their knowledge of the peering capacity they have deployed in LA & NY and the capacity they have between those peering locations and groups of end users without directly exposing their internal topology to NSP A.

An NSP could make use of an ALTO service to expose a cost mapping/ranking between End User IP subnets (within that NSP’s network) and entry points into that NSP’s network in order to try to influence the locations from which the CDN serves content into that NSP’s network.

3.5. CDN acquiring content from multiple upstream sources (Origins)

Before a surrogate within a CDN is able to deliver content to an End User it must first have a copy of the content that the End User is requesting. Content may be obtained by surrogates in advance of it being requested (pre-positioned) by End Users or it may be obtained by surrogates dynamically in response to End User requests for the content (on-demand).

The ultimate source of the content (i.e. where the 'master' copy is
permanently stored) is typically referred to as the content’s Origin (or Origin Server), however CDNs often employ an internal hierarchy of caching layers so that surrogates do not necessarily obtain content directly from the Origin. Such a hierarchy provides a number of benefits, for example reducing the number of requests for content received by the Origin (and therefore reducing the scaling requirements on the Origin), more efficient use of the underlying network as fewer copies of the content is required to traverse the same network links, etc.

For a particular CSP’s content service multiple, possibly independently addressable, Origins may be used for resiliency and the Origin(s) may be deployed in a distributed manner across multiple geographic locations.

For the rest of this use case "upstream source" is used to mean either the Origin itself as well as other sources of the content, for example another caching layer within the CDN that has (or will obtain on demand) a copy of the content but is not the actual Origin.

Therefore, for a particular item of content, a surrogate may have a choice of upstream sources (both internal to the CDN and external Origins) from which it could obtain the content.

When presented with a choice of upstream sources, a surrogate may utilise some combination of policy and heuristics to decide which upstream sources (and in which order) it should attempt to use to obtain the content. A CDN may wish to utilise network topology & cost information as one of the inputs into such a content source selection process, for example to weight upstream sources that are topologically close to the surrogate that requires the content.

Additionally, where the CDN is deployed within one or more NSP networks, an NSP may want to try to influence the choice of upstream sources, for example the NSP may prefer the CDN to use content sources that are deployed within that NSP’s network or within networks with which it has direct peering agreements with over other content sources.

An NSP (or a CSP) could provide an ALTO service which a CDN could use to obtain network topology and/or cost/ranking information to use as an input into surrogates’ selection decisions for content sources.

3.6. Additional Use Cases

The following additional use case may be relevant to ALTO and will be described in more detail in a future version of this document:
4. IANA Considerations

This document makes no specific request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

5. Security Considerations

TBD.

6. Contributing Authors

Reinaldo Penno
Juniper Networks
Email: rpenno@juniper.net

Richard Alimi
Google
Email: ralimi@google.com

Richard Yang
Yale University
Email: ryr@yale.edu

7. Acknowledgements

The authors would like to thank Vijay Gurbani and Volker Hilt for their review comments and contributions to the text.

8. Normative References

[I-D.ietf-cdni-problem-statement]

Authors’ Addresses

Ben Niven-Jenkins (editor)
Velocix (Alcatel-Lucent)
326 Cambridge Science Park
Milton Road, Cambridge CB4 0WG
UK
Email: ben@velocix.com

Grant Watson
BT
Email: grant.watson@bt.com

Nabil Bitar
Verizon
40 Sylvan Road
Waltham, MA 02145
USA
Email: nabil.bitar@verizon.com

Jan Medved
Juniper Networks
Email: jmedved@juniper.net

Stefano Previdi
Cisco Systems
Email: sprevidi@cisco.com
Abstract

The Application-Layer Traffic Optimization (ALTO) protocol is designed to allow entities with knowledge about the network infrastructure to export such information to applications that need to choose one or more endpoints to connect to among large sets of logically equivalent ones. The primary use case for the ALTO protocol was peer-to-peer applications for file sharing, video streaming and realtime communications, usually running on end-user devices. However, a number of other applications executing in more controlled environments may also benefit from the information that can be exported through the ALTO protocol. The use cases that have received significant attention include Content Delivery Networks (CDNs), distributed applications running in large datacenters, as well as systems made of inter-communicating ALTO servers.

To apply ALTO to these new use cases, this document aims to foster a discussion to determine if, and how, the ALTO protocol could be extended to provide a simple yet useful view of a computational environment that goes beyond the static (or near static) network topology and cost map information.

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."
Table of Contents

1. Introduction .................................................. 4
   1.1. Requirements Language .................................. 5
2. Use Cases ..................................................... 5
   2.1. Content Delivery Networks (CDNs) ....................... 5
   2.2. Virtualized Applications in Datacenters ............... 6
   2.3. ALTO Server-to-server Communications .................. 6
3. New Protocol Features ......................................... 6
   3.1. Server-initiated Notifications ......................... 7
   3.2. ALTO Information Extensions ........................... 8
      3.2.1. Bandwidth Availability Between Hosts ............. 9
      3.2.2. Resource Availability on Hosts ................... 9
      3.2.3. Content Availability on Hosts ................... 9
4. Security Considerations ....................................... 9
5. References .................................................... 10
   5.1. Normative References ................................... 10
   5.2. Informative References .................................. 10
Authors’ Addresses .................................................. 10
1. Introduction

The Application-Layer Traffic Optimization (ALTO) protocol is designed to allow entities with knowledge about the network infrastructure to export such information to applications that need to choose one or more endpoints to connect to among large sets of logically equivalent ones. The primary use case for the ALTO protocol was peer-to-peer applications for file sharing, video streaming and realtime communications, usually running on end-user devices. However, a number of other applications executing in more controlled environments may also benefit from the information that can be exported through the ALTO protocol. The use cases that have received significant attention include Content Delivery Networks (CDNs), distributed applications running in large datacenters, as well as systems made of inter-communicating ALTO servers.

Such applications require information about the underlying infrastructure that goes beyond network topology and associated costs. We believe that the ALTO protocol can be easily extended to provide this information.

The basic idea is to use the ALTO protocol to present a simplified view of a computational environment, aggregating with some level of abstraction and approximation information that at a fine-grained level may be conveyed by protocols like OSPF, ISIS, BGP, SNMP, ECN, and ConEx.

To provide such kind of information the ALTO protocol need to be extended on several axes:

- a means for providing incremental updates to optimize for frequently changing information;
- a means for providing integrity protection for the information provided by an ALTO server, in order to enable information redistribution;
- a server-initiated notification mechanism, for promptly informing applications of status changes;
- different types of information, not only related to network costs, such as:
  * network link load;
  * server load;
availability of resources such as storage memory, content and installed applications.

Detail-level and timescale of the additional information that can be provided are an open topic of discussion. If on the one hand applications may not take any advantage of too coarse-grained information, on the other hand ALTO protocol extensions cannot satisfy all the requirements of the mechanisms that today make full use of such low level information and therefore must not be intended in any way as a replacement for them. The goal of this document is to frame the discussion of what could be reasonable compromises for exporting information of the underlying network and computational infrastructure to applications that need to make best use of it.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

2. Use Cases

2.1. Content Delivery Networks (CDNs)

CDNs consist of systems of caching servers that cooperate in the distribution of frequently requested content. When a client wants to access some content, the request is directed by the CDN routing logic to the most appropriate caching server. The criteria for selecting the most appropriate server can be arbitrary complex and depend of information such as:

- network distance between the querying client and the caching server;
- load of the caching server, e.g. in terms of number of concurrent requests or available serving capacity measured over a recent timeframe;
- content availability;
- storage capacity availability, for deciding whether to replicate some content on servers that do not have it yet.
2.2. Virtualized Applications in Datacenters

Applications running on virtual servers in large datacenters require dynamic allocation of resources such as computation power, storage capacity and network bandwidth. Datacenter management logic allocate the resources of physical servers to such applications based on information such as:

- resource availability on the physical servers;
- application code and configurations availability on the physical servers;
- network connectivity quality (i.e. delay and expected throughput) between the physical servers the virtual server is already running on, and the new physical servers the additional resources may be allocated from.

2.3. ALTO Server-to-server Communications

ALTO servers can improve the guidance they provide by aggregating information distributed by other servers (see [I-D.medved-alto-svr-apis] and [I-D.dulinski-alto-inter-problem-statement]). In such scenarios, for the model to be effective, at any point in time all servers need to have a fresh version of the information distributed by the servers they are communicating with, regardless of the type of information distributed. However, the frequency of changes increases with the number of communicating servers, and the faster the information changes, the less the pull-based approach of the base ALTO protocol [I-D.ietf-alto-protocol] is suitable for maintaining an updated representation of the environment status.

3. New Protocol Features

This section discusses some extensions to the ALTO protocol that can be used to cover the use cases described in Section 2. Such extensions include:

- incremental updates for the network and cost maps defined in the base ALTO protocol [I-D.ietf-alto-protocol], for the CDN, datacenter and server-to-server use cases, to enable efficient transmission of status changes;

- integrity protection, for the server-to-server use case, to enable servers assert the authenticity of data re-distributed by other servers;
o a mechanism for server-initiated notifications, for the CDN, datacenters and server-to-server use cases, to enable a fast propagation of the status changes;

o new types of information for the ALTO protocol, for the CDN and datacenters use cases, to provide a representation of the computational environment that goes beyond the network topology.

Incremental updates and integrity protection are easily defined on the basis of existing (ongoing) work, namely [I-D.pbryan-json-patch] and [I-D.jones-json-web-signature]. The remainder of this section discusses the other, perhaps more controversial extensions.

3.1. Server-initiated Notifications

The base ALTO protocol [I-D.ietf-alto-protocol] defines a JSON-based syntax to be conveyed statelessly over HTTP. Such a lightweight approach has several advantages and is considered most appropriate for the use case of peer-to-peer applications, where the information is likely to be retrieved and consumed by huge numbers of clients. However, in more controlled environment, the same information, with the same or an equivalent syntax, can also be conveyed by different protocols, such as XMPP, SIP, or by any protocol with publish/subscribe capabilities that would allow servers to send updates to subscribed clients.
As an example, if an ALTO service provider wanted to make cost maps available also through XMPP (assuming some kind of specification for ALTO-over-XMPP exists), it could simply advertise the proper URI in the information resource directory along with the basic HTTP one:

```
{
  "resources" : [
    {
      "uri" : "http://alto.example.com/serverinfo",
      "media-types" : [ "application/alto-serverinfo+json" ]
    }, {
      "uri" : "http://alto.example.com/networkmap",
      "media-types" : [ "application/alto-networkmap+json" ]
    }, {
      "uri" : "http://alto.example.com/costmap/num/routingcost",
      "media-types" : [ "application/alto-costmap+json" ],
      "additional-uris" : [ "xmpp:routingcost@alto.example.com" ],
      "capabilities" : {
        "cost-modes" : [ "numerical" ],
        "cost-types" : [ "routingcost" ]
      }
    }, {
      "uri" : "http://alto.example.com/costmap/num/hopcount",
      "media-types" : [ "application/alto-costmap+json" ],
      "additional-uris" : [ "xmpp:hopcount@alto.example.com" ],
      "capabilities" : {
        "cost-modes" : [ "numerical" ],
        "cost-types" : [ "hopcount" ]
      }
    }
  ]
}
```

3.2. ALTO Information Extensions

The base ALTO protocol [I-D.ietf-alto-protocol] has been designed to be easily extended, in terms of both endpoint properties and path cost types. The remainder of this section discusses the types of information that are required by the use cases described in Section 2 and that would allow an ALTO servers to expose an abstract representation of a computational environment beyond the simple network topology.
3.2.1. Bandwidth Availability Between Hosts

Bandwidth availability is a kind of information that changes instantaneously and strictly depends on applications behavior. For such (and other) reasons, conveying it for congestion control other than in-band within the data flows may result useless at best, if not the cause of detrimental feedback loops.

However, some notion of link bandwidth availability averaged over a reasonable timeframe may be effectively used by CDN or datacenter applications to select well-connected pairs or groups of hosts that have to perform bandwidth-demanding tasks.

Information about bandwidth availability can be defined for encoding in the ALTO protocol as a new path cost type.

3.2.2. Resource Availability on Hosts

Information about storage and computational capacity availability averaged over a reasonable timeframe may be effectively used by CDN and datacenter applications as one of the criteria for selecting hosts for serving content or performing tasks.

Information about resource availability can be defined for encoding in the ALTO protocol as a new endpoint property.

3.2.3. Content Availability on Hosts

Information about content availability can be expressed as lists of URIs (e.g. for identifying stored files in CDN caching servers), URNs or other kinds of identifiers (e.g. for identifying installed applications on physical servers in a datacenter).

Information about content availability can be defined for encoding in the ALTO protocol as a new endpoint property.

4. Security Considerations

The information types discussed in this document are likely to be privacy critical in many environments and therefore must be protected, restricting or controlling access to the servers that export them.

Server initiated notification requires more resources than the stateless retrieval model adopted by the base ALTO protocol [I-D.ietf-alto-protocol] and is more thus more vulnerable to denial of service attacks.
Access control mechanisms, including HTTP’s, may be valid options for addressing the security issues related to both privacy critical information types and resource-consuming server notifications.

5. References

5.1. Normative References


5.2. Informative References

[I-D.dulinski-alto-inter-problem-statement]

[I-D.ietf-alto-protocol]

[I-D.jones-json-web-signature]

[I-D.medved-alto-svr-apis]

[I-D.pbryan-json-patch]

Authors’ Addresses

Enrico Marocco
Telecom Italia
Via Reiss Romoli, 274
Torino, 10148
Italy

Phone:
Email: enrico.marocco@telecomitalia.it

Vijay K. Gurbani
Bell Laboratories, Alcatel-Lucent

Email: vkg@bell-labs.com
Abstract

ALTO servers require automated operation, where the topology of the underlying networks is incorporated into network maps automatically. In addition to the Client-to-Server API defined in the ALTO protocol document, two more standardized API are required: an API between the ALTO Server and networking nodes (e.g. routers), through which the ALTO Server can get topology information from the network, and an API between the ALTO Servers, through which they can exchange topology and status information between themselves.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119]

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 September 5, 2011.
Copyright Notice

Copyright (c) 2011 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 .................................................. 3
2. Scope ........................................................ 3
3. Terminology ................................................... 4
4. ALTO Server API Reference ................................. 4
   4.1. The ALTO Server-to-Network Interface ............... 5
      4.1.1. Requirements ........................................ 5
      4.1.2. BGP with TE Extensions ................................. 6
   4.2. The ALTO Server-to-Server Interface ............ 8
5. Conclusion .................................................... 9
6. IANA Considerations .......................................... 9
7. Security Considerations ...................................... 9
8. Acknowledgements .............................................. 10
9. References ..................................................... 10
   9.1. Normative References .................................... 10
   9.2. Informative References ................................ 10
Authors’ Addresses .............................................. 11
1. Introduction

ALTO Servers are becoming increasingly important technology for finding "the best" or "most preferred" content or server. For example, an ALTO Server can be used to facilitate the selection of the best cache in a CDN, the best set of peers for a P2P node ([RFC5632] or [I-D.lee-alto-chinatelecom-trial]), or the best service instance in a cloud. These use cases will require that network and cost map information accurately reflects the actual network topology and utilization. Static configuration of network and cost maps is not feasible even for moderately sized networks. Therefore, creation of network and cost maps in the ALTO Server should be automated and policy driven.

The ALTO Server can use multiple sources of information to generate the network and cost maps. Network topology data coming directly from routers is required. Additionally, traffic engineering data, geo location data, or network resource utilization data could also be used to further refine the maps, or to generate different maps for different clients. The ALTO Server should use well defined APIs to get the data required to generate maps, since the data will be obtained from different sources provided by a multitude of vendors, and vendor inter-operability will be critical for adoption of ALTO-based solutions. For network topology data, this draft proposes BGP with TE extensions as the ALTO Server-to-Network API.

The ALTO Server will typically only have partial topology data, which will depend on the Server's location and the sources from which it obtains data to generate the network and cost maps. To obtain a full view of the network topology, the ALTO Server will have to exchange topology data with other ALTO Servers, or redirect Endpoint Cost ranking requests to the best possible ALTO Server. Therefore, a standard Server-to-Server API is also required.

2. Scope

The scope of this draft are the ALTO Server-to-Network APIs and Server-to-Server API that are required for automated operation of the ALTO Service. The Server-to-Network API is used to obtain network topology information from the underlying network. Server-to-Server API is used to exchange topology information between ALTO servers, or to redirect ranking requests from one ALTO Server to another. The ALTO Client-to-Server protocol [I-D.ietf-alto-protocol] itself may be used as the ALTO Server-to-Server protocol; in other words, one ALTO Server may request maps or status from other servers.
3. Terminology

We use the following terms defined in ALTO Problem Statement [RFC5693]: Application, ALTO Service, ALTO Server, ALTO Client, ALTO Query, ALTO Reply, ALTO Transaction.

4. ALTO Server API Reference

In addition to the ALTO protocol, which constitutes the API between the ALTO Server and its clients, the ALTO Server needs several other APIs to get data that are required to generate the network and cost maps. The reference diagram of possible ALTO Server APIs is shown in Figure 1.

![Figure 1: ALTO Server API reference](image)

The ALTO Server interfaces shown in Figure 1 are as follows:

1. CS: The Client-to-Server interface has been the focus of the ALTO WG, and is defined in [I-D.ietf-alto-protocol].

2. SS: The Server-to-Server interface is required to exchange topology data and status between ALTO servers in different networks or administrative domains. For Endpoint Cost queries, the interface can be used to direct the client’s request to the...
peer ALTO Server that has the best data to respond to the query. The interface may also facilitate other functions, such as ALTO Server discovery.

3. SN: The Server-to-Network interface is used to get the network topology data from the network.

4. NN: The Network-to-Network routing and data interfaces are well-defined in a number of standards (for example, BGP [RFC4271]), and they are not in scope of this draft.

4.1. The ALTO Server-to-Network Interface

4.1.1. Requirements

The Server-to-Network interface should satisfy the following requirements:

- Enable automation of the operation of the ALTO server with minimal human intervention
- Leverage existing sources of network topology data; don’t introduce new (routing) protocols; don’t force un-natural deployment of routing protocols within the ISP network
- Leverage scalable mechanisms for (near real-time) network topology acquisition; don’t use fragile mechanisms to obtain data (e.g. screen-scraping information from looking glass servers)
- Enable centralized and/or distributed deployments of ALTO servers
- Provide network topology information from within the ISP network (intra-AS) as well as from outside the ISP network (inter-AS), as well as from different intra-domain routing areas. (Note that some ISPs use multiple AS’s for different components of the overall network topology.)
- Enable automated ALTO server policy controls above and beyond mere routing metrics
- Provide origin security for network topology information
- Provide the right balance between frequency of updates and accuracy /timeliness of the data. Topology updates from the network should be throttled. For ALTO application, a 15 minute time interval between topology updates from the network should be sufficient.
In addition to having a standardized Server-to-Network interface, the algorithms for generation of ALTO network / cost maps and for endpoint ranking should be normalized as well, to facilitate interoperability of different ALTO Server implementations.

4.1.2. BGP with TE Extensions

Network topology is best conveyed through routing protocols. BGP carries information about all subnets in the network, and subnet / prefix data from BGP is required to generate ALTO network maps. Intra-AS topology information that is carried in link-state IGPs and inter-AS topology information carried in BGP is required to generate ALTO cost maps. IGP TE data is required if costs in the cost maps have a link utilization component.

This draft proposes to use BGP with TE extensions [I-D.gredler-bgp-te] as the ALTO Server-to-Network API that can carry both the subnet/prefix data for network map generation and the topology data for cost map generation. A BGP Speaker can learn a part or the entire intra-AS topology by participating in the IGP and then distribute the learned topology to other BGP Speakers in the AS. The ALTO Server establishes an iBGP session with a BGP speaker within the AS, typically a Route Reflector, and learns the intra-AS topology from its peer BGP speaker, along with the inter-AS topology and the subnet/prefix data.

Using BGP with TE extensions as the ALTO Server-to-Network API has several advantages:

- Avoid peering with IGP routers, which is more challenging than BGP peering. Moreover, IS-IS, OSPF and EIGRP implementations would be required, although only one IGP peering implementation would typically be used at any given time.

- Unified interface to the network (single protocol), which carries all the network information required to generate the topological component of network and cost maps. The alternative would be for the ALTO Server to interface - in addition to BGP - with IS-IS, OSPF and EIGRP routing protocols.

- Simplified handling of multi-area IGP topologies: if the ALTO Server wants to see the entire multi-area IGP topology, it would need to peer with at least one IGP router in each area. Since the ALTO Server would have to reside in one of the areas, it would have to peer with IGP routers in other areas over GRE tunnels, which is complex and potentially error prone. Alternatively, an ALTO Server would have to be placed in each area, and the ALTO Servers would have to exchange topology information between
themselves via the Server-to-Server API.

- The ALTO Server can peer with a BGP Route Reflector. Route Reflectors are widely deployed, and the Route Reflector control architecture dovetails nicely with the desired ALTO Server control architecture.

- BGP policy and marking capabilities allow the operator to modify or filter / adjust both the prefix and the connectivity information specifically for the ALTO Server’s use. This capability is important if the BGP Speaker and the ALTO Server are in different administrative domains.

- BGP has some origin security. This capability is important if the BGP Speaker and the ALTO Server are in different administrative domains.

- BGP carries multicast for future enhancements, where the ALTO Server will be creating multicast network and cost maps.

- Using BGP with TE extensions means that there only needs to be one BGP speaker in each area (or two for redundancy) that gets the area’s topology from local IGP routers. The topology information is then distributed throughout the AS and relayed to all interested ALTO Servers. The topology information can be appropriately tagged so that is only stored by those Route Reflectors that talk to ALTO Servers. BGP Input and Output filtering could ensure that only the minimum set of BGP Speakers would need to store the topology information.

- The ALTO Server only needs to peer with a single BGP Speaker to get the entire network topology.

- BGP with TE extensions can be used between eBGP peers to advertise intra-AS topology information between peers in different ASes. Intra-AS topology information from multiple ASes can then be used by an ALTO Server to create more detailed network and cost maps for the combined network.

Due to policy and security considerations, it is assumed that an ALTO Server speaks via the Server-to-Network APIs only to a BGP Speaker in the same Administrative Domain (that may encompass multiple IGP areas and ASes). Any other use cases are for further study.

Note that the network topology received by the ALTO Server must not be summarized beyond what is expressed by the IGP in each area. This is because the network (router) does not understand the application-specific constraints of the ALTO Server for suitable summarization.
Also, where different scaling of metrics and different policies exist inside an Administrative Domain, the Alto Server is instructed via management on how to compare or normalize the data received from the network. The network is not expected to provide translation or normalization.

4.2. The ALTO Server-to-Server Interface

The ALTO Server-to-Server API is required because each ALTO Server will likely have only a partial view of the overall network. The ALTO Server’s view of the network depends on which routers are the sources of its topology data. Each router’s topology data depends on the administrative domain (Autonomous System) where the router is deployed. In order to generate a combined network/cost map that covers the network beyond its own Autonomous System, an ALTO Server needs to exchange its map information with other ALTO Servers in other network locations and/or administrative domains. To allow generation of combined maps, costs in partial cost maps must be normalized.

The network and cost maps defined in the Client-to-Server ALTO interface provide sufficient semantics to be considered a good candidate for the Server-to-Server information exchange. In other words, the ALTO Client-to-Server interface can be used for communication between ALTO Servers as well.

Note that the idea of sharing information directly between ALTO clients has already been anticipated, as stated in Section 3.1.4 in ALTO Requirements [I-D.ietf-alto-reqs]:

REQ. ARv07-31: The ALTO client protocol SHOULD allow the ALTO server to add information about appropriate modes of re-use to its ALTO responses. Re-use may include redistributing an ALTO response to other parties, as well as using the same ALTO information in a resource directory to improve the responses to different resource consumers, within the specified lifetime of the ALTO response...

Also, although not a formal part of the ALTO protocol, support for redistribution of ALTO data between clients has been anticipated in the ALTO Protocol specification [I-D.ietf-alto-protocol] – see Sections 6.2 and 8. Sharing data between ALTO Servers is similar, but not the same.

Typically, an ALTO Server will handle requests for different services. Moreover, the level of trust between different ALTO Servers can vary. Therefore, topology passed via the Server-to-Server API may be summarized, aggregated, or incomplete as long as they are sufficient to meet the requirements implied by the client’s
request.

5. Conclusion

Having well-defined standard APIs will facilitate inter-operation between ALTO Servers and the different sources of information that are required to put together the maps. It will also facilitate inter-operation between the ALTO Servers themselves. Multiple ALTO Servers in different administrative domains may be required to combine partial network maps / cost maps into an overall set of maps that cover a larger multi-provider network or the whole internet. Altogether, having standardized APIs will facilitate interoperability between ALTO Servers from different vendors.

6. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

7. Security Considerations

ALTO offers advice to applications on the optimality of various possible Internet destinations for acquiring a resource or service. An attacker who subverts or impersonates an ALTO service might be able to trick many application on the Internet into contacting the same host as a part of a distributed denial of service attack, for example. Interfaces that provision the back-end of ALTO servers are therefore a potentially attractive to attackers, as attackers might attempt to corrupt the ALTO database in order to launch such an attack.

For an ALTO server back-end interface to accept topology data from BGP, the server must trust the source of the information. The ALTO server must peer with a known route reflector, and must authenticate that entity, especially if it is outside the administrative domain of the ALTO server. Any origin security mechanisms will also increase the assurance of the ALTO server. Integrity protection for the channel between the ALTO server and the BGP speaker will also prevent malicious parties from inserting problem information.

Similarly, the ALTO server-to-server mechanism also requires an authentication and data integrity mechanism. If ALTO servers share network maps between one another, for example, assuring the
authenticity and source of data is essential. If ALTO servers share network maps with one another over a public network, a confidentiality mechanism will also be desirable in order to prevent eavesdropping.

8. Acknowledgements

Hannes Gredler from Juniper Networks made significant contributions to concepts presented in this draft. We would like to thank Alia Atlas from Juniper Networks for her input and comments.

9. References

9.1. Normative References

[I-D.gredler-bgp-te]

[I-D.ietf-alto-protocol]

[I-D.ietf-alto-reqs]


9.2. Informative References

[I-D.lee-alto-chinatelecom-trial]
Li, K. and G. Jian, "ALTO and DECADE service trial within China Telecom", draft-lee-alto-chinatelecom-trial-01 (work in progress),

Authors’ Addresses

Jan Medved
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
US

Email: jmedved@juniper.net

David Ward
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
US

Email: dward@juniper.net

Jon Peterson
Neustar

Email: jon.peterson@neustar.biz

Richard Woundy
Comcast Corporation
27 Industrial Avenue
Chelmsford, MA 01824
US

Email: Richard_Woundy@cable.comcast.com
Abstract

Networking applications can request through the ALTO protocol information about the underlying network topology from the ISP or Content Provider (henceforth referred as Provider) point of view. In other words, information about what a Provider prefers in terms of traffic optimization -- and a way to distribute it. The ALTO Service provides information such as preferences of network resources with the goal of modifying network resource consumption patterns while maintaining or improving application performance.

One of the main use cases of the ALTO Service is its integration with Content Delivery Networks (CDN). The purpose of this draft is twofold: first, to describe how ALTO can be used in existing and new CDNs, both within an ISP and in separate organizational entities from the ISP; second, to collect requirements for ALTO usage in CDNs and to provide recommendations into the development of the ALTO protocol for better support of CDNs.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

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-

Table of Contents

1. Introduction ............................................. 5
2. Scope ..................................................... 5
3. Terminology .............................................. 5
4. Request Routing as an Integration Point of ALTO into CDN ... 6
   4.1. HTTP Redirect ......................................... 7
   4.2. DNS Request Routing .................................... 7
5. Basic Scheme of CDN/ALTO Integration .......................... 8
   5.1. Basic Integration Scheme ............................... 8
      5.1.1. ALTO for HTTP Redirect .............................. 9
      5.1.2. ALTO for DNS Resolution ............................ 10
   5.2. Multi-hop Redirection ................................... 10
6. Request Routing using ALTO Services ........................... 11
   6.1. ALTO Topology vs. Network Topology ...................... 11
   6.2. CDN Node Discovery and Status Notification ................ 11
      6.2.1. CDN Node Status Updates received by Request
             Routing Function .................................... 12
      6.2.2. CDN Node Status Updates received by ALTO ........... 12
   6.3. Request Routing using the Map Service .................... 13
   6.4. Request Routing using the Endpoint Cost Service ........... 14
      6.4.1. Topology Computation and ECS Delivery ............. 15
      6.4.2. Ranking Service .................................... 15
   6.5. Update, Redirection of ALTO Info to CDN Request Routing . 16
      6.5.1. ALTO Update and Network Events .................... 16
      6.5.2. Caching and Lifetime ............................... 16
      6.5.3. ALTO Redirection ................................... 16
      6.5.4. Groups and Costs ................................... 17
7. Multiple Administrative Domains ................................ 17
   7.1. CDN nodes/Request Router in a separate administrative
        domain from that of ISP ................................ 18
   7.2. Managed DNS Domain with Three Administrative Domains ... 21
      7.2.1. Managed DNS Redirect to Local CDN .................. 21
      7.2.2. Managed DNS with CDN-Provided Request Routing ......... 22
8. Protocol Recommendations .................................... 23
   8.1. Necessary Additions ................................... 23
      8.1.1. NA1: PID Attributes ................................ 23
      8.1.2. NA2: PID Attributes and Query ....................... 24
   8.2. Helpful Additions ..................................... 24
      8.2.1. HA1: Push Mechanism ................................ 24
      8.2.2. HA2: Incremental Map Updates ....................... 24
      8.2.3. HA3: ALTO Border Router PID attribute ............... 24
      8.2.4. HA4: CDN ALTO Server Discovery ..................... 24
      8.2.5. HA5: Extensible ALTO Cost Maps ..................... 25
      8.2.6. NA4: Federated Deployment of ALTO Servers ......... 25
9. IANA Considerations ....................................... 25
10. Security Considerations .................................... 25
11. Acknowledgements ......................................... 25

12. References ................................................... 25
  12.1. Normative References ................................. 25
  12.2. Informative References ................................ 26
  Authors’ Addresses ........................................... 27
1. Introduction

Content Delivery Networks are becoming increasingly important in the Internet [ARBOR] and many CDNs today already use some form of proximity such as latency-based proximity [GoogleCDN]. But in many cases the content provider/distributor and the Internet Service Provider (ISP) are disjoint entities. Consequently, even if content servers are co-located into the ISP’s networks, there is not a standardized way to share server location and/or network topology information. Therefore a natural step forward would be to use ALTO to share this information.

Another key aspect of ALTO in the context of CDNs deployments is that it is desirable that no changes to the hosts are needed (or that changes to hosts would be transparent to the user). In other words, a traditional web browser using standard HTTP flow is all there is needed to take advantage of ALTO information. This is a significant difference from the P2P applications where a special client is typically needed and ALTO is normally used as a way to reduce operational expense.

2. Scope

This document discusses how Content Delivery Networks can benefit from ALTO through integration of the ALTO Service with the main request routing techniques. There are two objectives:

- Present basic integration schemes of ALTO into CDNs.
- Provide protocol recommendations to ALTO: Whenever a new requirement on protocol functionality is identified to achieve integration with CDNs, it will be enumerated with ‘REQ-<N>’. Each requirement is documented in a section of its own in order to foster parallel discussions and possible adoption.

3. Terminology

We use the following terms defined in ALTO Problem Statement [RFC5693]: Application, ALTO Service, ALTO Server, ALTO Client, ALTO Query, ALTO Reply, ALTO Transaction.

In addition to the above, the following terms are defined:

Content-aware Proximity Request Routing Function: The Request Routing function knows about locations and presence of content & media objects in the network. Therefore the redirection to a CDN
node is made based on both the availability of content and/or content-type in that CDN node and the proximity of the CDN node to the requesting user.

Service-aware Proximity Request Routing Function: The Request Routing function knows about locations of CDN nodes in the network and redirects user to the closest CDN node. A redirection is made irrespective of content presence in the CDN node; if content is not present, the node will be populated with the content while the content is being served to the user.

HTTP Request Routing Function: A Content-aware or Service-aware Proximity Request Routing function for HTTP. It embeds an HTTP Server that performs HTTP Redirections, an ALTO client that retrieves network mapping from the ALTO Server, and a Location Database which stores network mappings received from the ALTO Client. The HTTP Server consults the Location Database when making redirection decisions.

4. Request Routing as an Integration Point of ALTO into CDN

Content Distribution is a rich and evolving field. New architectures and approaches (e.g., a hybrid architecture using both servers and P2P) continue to be developed in the research community and industry. Several CDN architectures are being deployed in production. While we would like to provide a survey of each possible CDN architecture and show how it may be integrated with ALTO, it would be a daunting task to track such a rapidly-changing field.

One scheme that is out of the scope of this document is P2P-only CDNs, where the application tracker takes the role of the ALTO Client, fetching the Network and Cost Maps from the ALTO Server and integrating them with its peer database. The result is a peer database that takes into account both the current peer metrics, such as peer availability or content availability, and network metrics, such as topological localization. This architecture, in the context of file sharing, has been studied extensively and trialed by ISPs such as Comcast [RFC5632] and China Telecom [I-D.lee-alto-chinatelecom-trial] under the ALTO/P4P [P4P] protocol. Thus, P2P-only CDNs are not discussed in this document.

The Request Routing Component of a CDN directs a request to a serving CDN node, and thus is the major integration point to utilize information available through ALTO. Today, multiple request routing schemes have been used even in CDNs with purely server-based infrastructure. The specific schemes include HTTP Redirect, DNS name resolution, and anycast. We focus on HTTP Redirect and DNS name resolution.
resolution.

Though anycast is a request routing technique that has been used in deployed CDNs, we do not discuss it in this document. Even though one may be able to integrate ALTO with anycast, we do not believe that this is a proper use of ALTO’s capabilities. In particular, ALTO has been developed to improve selection amongst multiple content providers at the application level. In contrast, anycast operates by adjusting the routing layer to match content consumers with the desired content providers. Applying ALTO to routing layer decisions introduces additional complexity because it directly adjusts the routing layer from which the ALTO information is typically generated, creating a tight feedback loop. We leave a more detailed study of integrating ALTO with anycast-based CDNs as future work.

We next briefly review the two mechanisms presented in this document, HTTP Redirect and DNS Request Routing.

4.1. HTTP Redirect

In this mechanism, an HTTP GET request from a host is received by an HTTP Request Routing Function which sends back an HTTP response with Status-Code 302 (Redirect) informing the host of the most preferred location to fetch the content. The HTTP Redirection method is already commonly used in production CDNs as described in RFC3568. ALTO integration provides localization services where the device that performs the redirection becomes an ALTO client.

4.2. DNS Request Routing

In this mechanism, the DNS server handling host requests provides the Request Routing Component. When the host performs a DNS query/lookup, the IP address(es) in the DNS response will indicate the selected location to serve the request.

DNS queries can be either iterative or recursive. Iterative queries can be used with ALTO if the host itself queries the DNS Servers, or if the DNS Proxy used by the host is topologically close to the host. If the Host directly queries the DNS Servers, the authoritative DNS Server can see directly the host’s IP address. If the DNS Proxy is topologically close to the Host, its IP address is a good approximation for the host’s location. In recursive queries, the authoritative DNS Server sees the IP address of the previous DNS Server in the resolution chain, and the IP address of the host is unknown. DNS-based request routing does not work well with recursive DNS queries.

In an iterative DNS lookup with a DNS Proxy (say for cdn.com), the
host queries the Proxy, which in turn first queries one of the root servers to find the server authoritative for the top-level domain (com in our example). The Proxy then queries the obtained top-level-domain DNS server for the address of the DNS server authoritative for the CDN domain. Finally, the Proxy queries the DNS server that is authoritative for the cdn.com domain. The authoritative DNS Server for cdn.com will perform the request routing to the most appropriate CDN node, based on the source IP address of the requestor. The host will then request the content directly from the CDN Node.

Recently, an EDNS0 option in DNS query has been proposed in [I-D.vandergaast-edns-client-subnet] that will provide a mechanism to carry sufficient network information about the client for the authoritative DNS server to tailor DNS response based on the client’s subnet. Using this mechanism, the authoritative DNS server can achieve the same request routing accuracy as that of the HTTP Request Routing Function, and both recursive and iterative queries can be supported.

5. Basic Scheme of CDN/ALTO Integration

Although HTTP Redirect and DNS are quite different mechanisms to direct a request to a serving CDN node, as we will see, the basic structure of integrating ALTO with them can be quite similar. Thus, we first present common structures. We refer to the HTTP Redirect component or the DNS component of a CDN as a CDN Request Routing Function.

5.1. Basic Integration Scheme

Figure 1 shows a general structure to embed an ALTO Client into a CDN Request Routing Function.

An ALTO Server may aggregate information from multiple sources, such as routing protocols, traffic engineering policies, and monitoring systems. Thus, ALTO is complementary to existing infrastructure. For further detail, see Figure 1 of [I-D.ietf-alto-protocol].

5.1.1. ALTO for HTTP Redirect

To make the basic scheme more concrete, Figure 2 shows the case that the Request Routing Function is HTTP Redirect.
5.1.2. ALTO for DNS Resolution

Figure 3 shows the case that the Request Routing Function uses DNS Resolution.

```
2      +----------------+
   | +----------------- |       root     |
   | |   Name Server   +----------------+      | Content  |
   | |                                          | Provider |
   | |           3      +----------------+      +----------+
   | |   +------------> |       com      |
   | |   | +----------- |   Name Server  |
   | |   | |     5      +----------------+
   | |   V   | V
   | V   +---------+    6      +----------------+
   | ^ |               |     cdn.com    |
   | | 8             |   Name Server  |
   | V               | +------------+ |
1 | | 8             | |ALTO Client |
   | ^               | +------------+ |
   |                  |   ALTO Protocol
   |                  |       V
   |                  |           V
   |                  |   CDN Node   |
   | 1 |               | +----------++
   | V |               | |ALTO Server |
   |   |               | +----------+
   |   +------------+ |
```

Figure 3: ALTO for DNS Resolution.

5.2. Multi-hop Redirection

The preceding examples show the logical flow for redirection. It is important to state that there maybe multiple redirection hops.

For HTTP Redirect, the requestor may be redirected again by the first CDN node. For DNS, the first DNS server may direct, using aggregated ALTO information (e.g., from multiple ALTO Servers of multiple ISPs), the DNS resolution to a second level DNS server, which then may use more specific ALTO information as well as CDN node status.
6. Request Routing using ALTO Services

Either the Map Service or the Endpoint Cost Service of ALTO can be used by the Request Routing Function. We first discuss two common issues: how to configure ALTO topology at ALTO servers; and how to achieve CDN node discovery and status notification. Then we give specific details on using the Map Service or the Endpoint Cost Service.

6.1. ALTO Topology vs. Network Topology

To answer queries from CDN Request Routing Functions, the ALTO server builds a ALTO-specific network topology that represents the network as it should be understood and utilized by the application layer (the CDN). Besides the security requirements that consist of not delivering any confidential or critical information about the infrastructure, there are efficiency requirements in terms of what visibility of the network, and at which level of granularity, is required by the CDN and more in general by the application layer.

The ALTO server builds topology (for either Map and ECS services) based on multiple sources that may include routing protocols, network policies, state and performance information, geo-location, etc. In all cases, the ALTO topology will not contain any details that would endanger the network integrity and security (for example, there will be no leaking of OSPF/ISIS/BGP databases to ALTO clients).

6.2. CDN Node Discovery and Status Notification

A design issue of integrating ALTO into Request Routing is how CDN Request Routing discovers the available CDN nodes and their locations. The exact mechanism is outside the scope of this document.

It is desirable that not only CDN node locations, but also real-time CDN node status (like health, load, cache utilization, CPU, etc.) is communicated to the Request Routing Function.

Specifically, CDN node status can be retrieved from the existing Load Balancer infrastructure. Most Load Balancers today have mechanisms to poll caches/servers via ping, HTTP Get, traceroute, etc. Most LBs have SNMP trap capabilities to let other devices know about these thresholds. Specification of a particular mechanism or API used to fetch load status information into an ALTO Server is out of scope of this document.

Note that in addition to the CDN node status, network status can also be retrieved from TE/RP databases. The Request Routing Function may
also need to be configured with a proper set of policies and business rules that control routing of requests. For example, it may be desirable to set up a rule that within a CDN certain requests have higher priority.

We see two approaches that CDN node status can be communicated to the Request Routing Function.

6.2.1. CDN Node Status Updates received by Request Routing Function

In the first approach, the Request Routing Function receives CDN Status updates directly.

For example, the Request Routing Function can implement an SNMP agent and get to know whatever is needed.

```plaintext
+-----------------+       |
| Request Routing |<--- Real-time CDN |
|     Function    |status updates
|<-------          |
| Requestor       |<--------  |
|<--- Business     |and Policies |
|     rules and    |<--- ALTO Protocol |
|     Policies    |v           |
|<-------          |
| +-------------+   |
| | ALTO Client  |
| +-------------+   |
|<-------          |
| +-----------------+       |
|   ALTO Server    |
<-----------------+       |
```

Figure 4: CDN Node Status to Request Routing Function

6.2.2. CDN Node Status Updates received by ALTO

In the second approach, the Request Routing Function receives CDN Status from ALTO instead of CDN nodes.

This model generally simplifies the Request Routing Function. It allows an easier distribution of the Request Routing Function, and to keep real time CDN status data updates in a logically centralized ALTO Server or in an ALTO Server Cluster. It allows for the Request Routing Function and the ALTO Server to be in different administrative domains. For example, the Request Routing Function can be in a Content Provider’s domain; the ALTO Server and CDN Nodes
in a Network Service Provider’s domain.

Specifically, ALTO Server could provide an API (for example, a Web Service or XMPP-based API) that could be used by CDN nodes to communicate their status to the ALTO server directly.

![Diagram](image)

Figure 5: CDN Node Status to ALTO

6.3. Request Routing using the Map Service

The ALTO client embedded in the Request Routing Function fetches the Network and Cost Maps from the ALTO Server and provides that information to the Request Router.

As an illustrative example, we consider the case of HTTP Redirect. A simple Request Router may be given (from an external source) the list of available CDN nodes. The Request Router precomputes a redirection table indexed by source PID with values being the closest CDN nodes. This redirection table can be built based on information from Network and Cost Maps. Then when the Request Router receives an HTTP GET request, it looks up the PID of the source IP address on the request, indexes the redirection table using the request PID to select a CDN node, and finally returns a response that is an HTTP redirect with the URL of the selected CDN node. The URL in 302 Redirect may contain the IP address of the selected CDN node or a domain name instead of IP address due to virtual hosting. Therefore the IP addresses contained in the cost maps may need to be correlated to
domain names a priori. In practice, the redirection table may be indexed by both source and content to provide better redirection.

The illustrative example can also be extended to DNS.

The Network Maps generated by the ALTO Server will contain both Host PIDs and CDN Node PIDs, i.e., Host PIDs contain host subnets; CDN PIDs contain IP addresses of available CDN nodes. Cost Maps may contain only cost from each host PID to each CDN PID and not the full matrix across all PIDs. The reason is that the Request Router may redirect a host only to a CDN node, not to another host as in the P2P case. Moreover, there is no generic way to disambiguate PIDs containing only hosts from PIDs containing CDN nodes.

It is possible that a Request Router may be designated as being responsible only for a fixed set of Host PIDs. This information can be made available to the Request Router before it receives requests from hosts. If the set of Host PIDs is not known ahead of time, the latency for serving requests will be impacted by the capabilities of the ALTO server.

With such information ahead of time, a Request Router that uses the Network Maps Service may pre-download the Network Map for the interesting Host PIDs and the CDN PIDs. It can also start periodically pulling Cost Map for relevant PID 2-tuples.

The Request Router can rely on the ALTO Server generated Cache-Control headers to decide how often to fetch CDN PID network map and Host PID network maps.

For Alto protocol requirements related to request routing with the Map Service see Section 8.1.1 and Section 8.1.2.

6.4. Request Routing using the Endpoint Cost Service

Alternatively, the Request Router may request the Endpoint service from the ALTO client.

Specifically, the Request Router requests the Endpoint Cost Service to rank/rate the content locations (i.e., IP addresses of CDN nodes) based on their distance/cost (by default the Endpoint Cost Service operates based on Routing Distance) from/to the user address.

Once the Request Router obtains from the ALTO Server the ranked list of locations (for the specific user) it can incorporate this information into its selection mechanisms in order to point the user to the most appropriate location.
A Request Router that uses the Endpoint Cost Service may query the ALTO Server for rankings of CDN Node IP addresses for each requesting host and cache the results for later usage.

Maps Services and ECS deliver similar ALTO service by allowing the Request Routing Function to optimize internal selection mechanisms. Both services deliver similar level of security, confidentiality of layer-specific information (i.e.: application and network) however, Maps and ECS differ in the way the ALTO service is delivered and address a different set of requirements in terms of topology information and network operations.

6.4.1. Topology Computation and ECS Delivery

ECS allows the Request Routing Function to not have to implement any specific algorithm or mechanism in order to retrieve, maintain and process network topology information (of any kind). The complexity of the network topology (computation, maintenance and distribution) is kept in the ALTO server and ECS is delivered on demand. Thus ECS is used in order to implement a lightweight integration of ALTO services in the CDN layer. ECS implies an ALTO and CDN implementation with the necessary scalability in order to cope with the amount of transactions that CDN and ALTO server will have to handle (knowing that the CDN is able to cache ALTO ECS results for further use).

6.4.2. Ranking Service

When a user requests a given content, the Request Routing Function locates the content in one or more caches and executes a selection algorithm to redirect the user to the ‘best’ cache. In order to achieve that, the CDN issues an ECS request with the endpoint address (IPv4/IPv6) of the user (content requester) and the set of endpoint addresses of the content caches (content targets). The ALTO server, receives the request and ranks the list of content targets addresses based on their distance from the content requester. By default, according to [I-D.ietf-alto-protocol], the distance represents the routing cost as computed by the routing layer (OSPF, ISIS, BGP) and may take into consideration other routing criteria such as MPLS-VPN (MP-BGP) and MPLS-TE (RSVP), policy and state & performance information.

Once the ALTO server has computed the distance it replies with the ranked list of content target addresses. The list being ranked by distance, the CDN is capable of integrating the rankings into its selection process (that will also incorporate other criteria) and redirect the user accordingly.
6.5. Update, Redirection of ALTO Info to CDN Request Routing

The information provided by an ALTO server to Request Routing is based on topology information of the network. The different methods and algorithms through which the ALTO server computes topology information and rankings is out of the scope of this document. However, update and redirection of such information may have an impact on the integration of ALTO into CDN Request Routing.

6.5.1. ALTO Update and Network Events

In the case that ALTO information is based on routing (IP/MPLS) topology, it is obvious that network events may impact the ALTO computation. The scope of the ALTO information delivered to Request Routing is not to maintain the CDN aware of any possible network topology changes since, due to redundancy of current networks, most of the network events happening in the infrastructure will have limited impact on the CDN. However, catastrophic events such as main trunks failures or backbone partition will have to take into account by the ALTO server so to redirect traffic away from the failure impacted area.

6.5.2. Caching and Lifetime

Each reply sent back by the ALTO server to the ALTO client running in the Request Routing Function has a validity in time so that the CDN can cache the results in order to re-use it and hence reducing the number of transactions between CDN and ALTO server. The ALTO server may indicate in the reply message how long the content of the message is to be considered reliable and insert a lifetime value that will be used by the Request Routing Function in order to cache (and then flush or refresh) the entry.

An ALTO server implementation may want to keep state about ALTO clients so to inform and signal to these clients when a major network event happened so to clear the ALTO cache in the client. In a CDN/ALTO interworking architecture, where there are only a few CDN components interacting with the ALTO server, there are no scalability issues in maintaining state about clients in the ALTO server.

6.5.3. ALTO Redirection

When ALTO server receives a request from a CDN Request Routing Function, it may not have the most appropriate topology information to reply. In such case, the ALTO server, may want to adopt the following strategies:
o Reply with available information (best effort).

o Redirect the request to another ALTO server presumed to have better topology information (redirection).

o Doing both (best effort and redirection). In this case, the reply message contains both the rankings and the indication of another ALTO server where more accurate information may be delivered.

The decision process that is used to determine if redirection is necessary (and which mode to use) is out of the scope of this document. As an example, an ALTO server may decide to redirect any request having addresses that are located into a remote Autonomous System. In such case the redirection message includes the ALTO server to be used and that resides in the remote AS. Redirection implies communication between ALTO servers so to be able to signal their identity, location and type of visibility (AS number).

6.5.4. Groups and Costs

An automated ALTO implementation may use dynamic algorithms to aggregate network topology. However, it is often desirable to have a mechanism through which the network operator can control the level and details of network aggregation based on a set of requirements and constraints. IP/MPLS networks make use of a common mechanism to aggregate and group prefixes that is called BGP Communities. BGP is the protocol all ISP networks use in order to exchange information about their prefix reachability. BGP Community us an attribute used to tag a prefix so to group prefixes based on mostly any criteria (as an example, most SP networks originate BGP prefixes with communities identifying the Point of Presence (PoP) where the prefix has been originated).

The ALTO server may leverage the BGP information that is available in the ISP network layer and compute group of prefixes. By policy, the ALTO server operator may decide an arbitrary cost to set between groups. Alternatively, there are algorithms that allow dynamic computation of cost between groups.

7. Multiple Administrative Domains

The preceding discussion works well in a single administrative domain setting: the CDN nodes are in the administrative domain of the ISP. However, the CDN nodes, the ISP, and the Request Router can be in different administrative domains. In this section, we consider a few such deployment cases. We use DNS as an example.
7.1. CDN nodes/Request Router in a separate administrative domain from that of ISP

In many situations, the CDN nodes and the Request Router are in a separate network managed by an entity that is distinct from the ISP. Consequently, the CDN nodes belong to a network with its own ALTO server that is distinct from the ALTO server of the ISP where the subscribers belong to.
The ALTO server in the CDN provider network is assumed to be initialized with information about the ISP networks it serves. For every such ISP network, it consults the routing plane to find the set of Border routers. The CDN network ALTO server computes the cost of reaching each Border router from every CDN node (say, \( C_{cdn} \)).
Next, the CDN ALTO server contacts the ISP network’s ALTO server and downloads the network map. In order to help the CDN ALTO server compute the cost from a CDN node to a subscriber’s PID, we break it down into two parts - the cost from the CDN node to the Border Router \( (C_{cdn}) \) and the cost from the Border Router to the subscriber’s PID (say, \( C_{isp} \)). Note that for any chosen exit point, \( C_{cdn} \) may be computed locally by the CDN ALTO Server. However, the fundamental issue is that \( C_{isp} \) depends on the exit point (Border outer) chosen by the CDN. There are multiple ways for the CDN ALTO Server to compute \( C_{isp} \) given the Network Map and Cost Map from the ISP’s ALTO Server.

One possibility is for the ISP ALTO Server to define a special Border Router PID (denoted by a PID attribute) which also indicates the corresponding Border Router PID in the CDN. The attributes and values may be agreed-upon by the ISP and CDN when the ALTO Services are configured. For example, in the example shown in Figure 5, the ISP ALTO Server indicates that its PID4 and PID5 are Border PIDs, with corresponding PIDs in the CDN as PID6, and PID7, respectively. Then, CDN ALTO Server can locally compute \( C_{isp} = \text{cost}(\text{ISP Border Router PID, Subscriber PID}) \).

A second possibility for computing \( C_{isp} \) is to make use of Border Router IP addresses. The CDN’s Border Router can locally determine the IP address of the connected border router in the ISP. In this approach, neither the CDN ALTO Server nor the ISP ALTO Server define PID attributes. The ISP ALTO Server is not required to define special PIDs for Border Routers - it only needs to ensure that Border Router IP addresses are aggregated appropriately in its Network Map.

Specifically, we identify two scenarios for the CDN ALTO Server to compute \( C_{isp} \) and \( C_{cdn} \).

In the first scenario, the CDN does not conduct CDN-level multi-path routing from the CDN nodes to the subscriber hosts. Thus, the routing path from a CDN IP address to a subscriber host IP address is typically uniquely (if no ECMP) determined by the network routing system. In this scenario, for a given CDN node IP address to a subscriber host IP address, the CDN ALTO Server uses the routing system to compute the Border Egress router inside the CDN, and the corresponding Border Ingress router inside the ISP. Then the CDN ALTO Server has \( C_{cdn}(\text{CDN node IP, Border Egress router IP inside the CDN}), \) and \( C_{isp}(\text{Border Ingress router IP inside the ISP, Subscriber IP}) \). The computation of \( C_{cdn} \) and \( C_{isp} \) can be done using ALTO in the traditional way through either the Network Map and Cost Map or the Endpoint Cost Service.

In the second scenario, the CDN may support CDN-level multi-path
routing from the CDN nodes to the subscriber hosts. In particular, from each CDN node, the CDN has a capability (e.g., through tunneling) to send to a subscriber host IP through multiple Border Egress routers (e.g., through any Egress router that receives an announcement from the ISP of the subscriber host IP). In this case, the cost of reaching a host PID from a given CDN node is then determined as the minimum cost among all possible intermediate Border Routers.

If the network is homogeneous, then a good approximation of the cost between each host PID and a given CDN node can be given as: \( C_{cdn}(\text{CDN Node}, \text{Border router}) + C_{isp}(\text{Border router, Subscriber PID}) \). In this computation, the Border Router is the one that is on the best path from the CDN node to the Subscriber PID.

The CDN ALTO server now has a cost map that provides the cost from each CDN node to all known Subscriber PIDs. The ALTO client in the CDN DNS server downloads this cost map in preparation for subscriber DNS requests.

When a subscriber DNS request arrives at the CDN provider’s DNS server, it looks up the network map and maps the source IP address to a Subscriber PID. It then uses the cost map to pick the best CDN node for this Subscriber PID.

7.2. Managed DNS Domain with Three Administrative Domains

Many organizations / content providers outsource DNS management to the external vendors for various reasons like reliability, performance improvement, DNS security etc. Managed DNS service could be used either with caches owned by the organization itself (section 6.3.1) OR with external CDNs (section 6.3.2).

7.2.1. Managed DNS Redirect to Local CDN

One of the common functions offered by managed DNS service vendor is DNS traffic management where DNS resolver can load balance traffic dynamically across CDN servers.

Typically managed DNS service provider has DNS resolvers spread across geographical locations to improve performance. This also makes easier for DNS resolver to redirect host to the nearest cache. Such a DNS resolver would be an ideal candidate to implement ALTO client where it can fetch network map and cost map from ALTO servers located in the same geographical area only. Load balancing implemented with the knowledge of network and cost map would be more efficient than other mechanisms like round robin.
In the figure above, there exists 2 possibilities:

Case 1: Domain 1 and Domain 2 are connected to the same service provider network. This case is similar to section 6.1

Case 2: Domain 1 and Domain 2 are connected to different service provider network. This case is similar to section 6.2

7.2.2. Managed DNS with CDN-Provided Request Routing

It is also possible to utilize a Managed DNS service and still rely on a CDN’s request routing. For example, this could be done if a network provider wishes to utilize a Managed DNS provider, but also wishes to integrate its own CDN using ALTO with DNS-based request routing.
To support this, the network provider may submit any necessary configuration files (e.g., indicating necessary CNAME records) to redirect CDN requests to the CDN’s DNS Request Routing mechanism. Requests for the CDN (e.g., ‘cdn.isp.com’) will then be directed by DNS request routing, while requests for other hosts are handled by the Managed DNS solution.

8. Protocol Recommendations

In the previous sections, this document has taken the approach of providing information on existing CDN approaches and possible benefits of utilizing ALTO. However, in developing the taxonomy, use cases, and deployment scenarios, we have identified cases where the ALTO Protocol [I-D.ietf-alto-protocol] and Server Discovery [I-D.kiesel-alto-3pdisc] [I-D.song-alto-server-discovery] [I-D.stiemerling-alto-dns-discovery] may be lacking capabilities that may be helpful and/or necessary for usage with CDNs. We now focus on detailing these gaps with the goal of providing feedback and recommendations. Note that some protocol changes may be necessary in the core protocol, while others may be implemented as extensions.

This section will be updated to track changes in the ALTO Protocol, ALTO Server Discovery, and accompanying protocols.

8.1. Necessary Additions

This section details changes to the ALTO protocols that would be necessary to make use of ALTO within CDN infrastructures. We classify a change as "necessary" if there is a core feature of a CDN/ALTO integration that is not possible to implement with the existing protocols.

8.1.1. NA1: PID Attributes

In order to disambiguate between PIDs that contain endpoints of a specific class, a PID property is needed. A PID can be classified as containing "CDN nodes", "Mobile Hosts", "Wireline Hosts", etc. This mechanism can be used to provide an ALTO Client a list of nodes of a particular type, along with the ALTO Costs to each node. In the context of CDNs, the attributes could describe a type of CDN node. For example an Origin would have one type of attribute while an edge cache would have another. This would allow for more intelligent routing.
8.1.2.  NA2: PID Attributes and Query

PID attributes can be used by the ALTO Client to select an appropriate host and also passed as a constraint in the map filtering service.

8.2.  Helpful Additions

This section details changes to the ALTO Protocol that would be helpful to make use of ALTO within CDN infrastructures. We classify a change as "helpful" if there is a compelling extension to existing CDNs that would be possible with additional functionality within ALTO, or if there is a component of CDN/ALTO integration that could be made more efficient or otherwise improved with additional ALTO functionality.

8.2.1.  HA1: Push Mechanism

It is important for the ALTO Service through the ALTO protocol or a companion protocol to provide a push mechanism from server to client. The push mechanism can be a notification that new data is available or the data itself.

8.2.2.  HA2: Incremental Map Updates

A natural evolution to the protocol if maps are large and change often is to allow for incremental map updates. In this sense the map contained in the reply would be considered the delta from the previous version.

8.2.3.  HA3: ALTO Border Router PID attribute

In order for administrative domains to collate costs across domain boundaries, the border routers may be placed in their own PIDs. Such PIDs may be identified by a Border Router attribute.

8.2.4.  HA4: CDN ALTO Server Discovery

In certain deployment scenarios, it may be beneficial for an ALTO client to directly query a CDN’s ALTO Server (instead of the CDN’s ALTO Server only being consulted as a backend process). For example, this can provide more accurate guidance than DNS Request Routing since the client’s IP address may be directly used by the CDN in order to select a cache node. This would require an ALTO Client (e.g., an ISP subscriber) to be able to discover an ALTO Server owned and/or managed by a CDN. This could be done by an extension to the discovery protocol, or it could be done by allowing an ISP’s ALTO Server to redirect certain queries to a CDN ALTO Server.
8.2.5. HA5: Extensible ALTO Cost Maps

Certain deployment scenarios may benefit from additional information being carried within ALTO information. For example, a trusted neighboring ISP B may be able to help ISP A optimize multihoming costs. To provide an extensible way to communicate additional data, the ALTO Protocol could be extended to include opaque data strings (in addition to numeric and ordinal values) in an ALTO Cost Map.

8.2.6. NA4: Federated Deployment of ALTO Servers

There is a need to define how ALTO servers may communicate with each other in a federated model.

9. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

10. Security Considerations

When the ALTO Server and Client are operated by different entities the issue of trust and security comes forward. The exchange of information could be done using the encryption methods already present in HTTP but preventing unauthorized redistribution comes into play. A further issue is if the ALTO information is transitive, which modifications are allowed.

11. Acknowledgements

We would like to thank Satish Raghunath and Mayuresh Bakshi for valuable input and contributions to this draft. We would also like to thank Nabil Bitar, Manish Bhardwaj, Michael Korolyov, Steven Luong and Ferry Sutanto for their comments.

12. References

12.1. Normative References

[I-D.ietf-alto-protocol]
Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol",
draft-ietf-alto-protocol-06 (work in progress),

12.2. Informative References


[I-D.vandergaast-edns-client-subnet] Contavalli, C., Gaast, W., Leach, S., and D. Rodden, "Client subnet in DNS requests", draft-vandergaast-edns-client-subnet-00 (work in


Authors’ Addresses

Reinaldo Penno
Juniper Networks
Email: rpenno@juniper.net

Jan Medved
Juniper Networks
Email: jmedved@juniper.net

Richard Alimi
Google
Email: ralimi@google.com

Richard Yang
Yale University
Email: yry@yale.edu

Stefano Previdi
Cisco Systems
Email: sprevidi@cisco.com
Abstract

Network Service Providers (NSPs) are currently considering to deploy Content Delivery Networks (CDNs) within their networks. As a consequence of this development, there is a need for interconnecting these local CDNs. The necessary interfaces for inter-connecting CDNs are currently being defined in the Content Delivery Networks Interconnection (CDNi) WG. This document focusses on the Request Routing Interface of CDNi, and more specifically on how the solutions currently being defined in the Application Layer Traffic Optimization (ALTO) WG can improve CDN request routing. The overall intention behind this document is to foster discussions (in the CDNi as well as in the ALTO WG) regarding a) if, b) how, and c) under what conditions ALTO can be useful to optimize CDNi request routing. As basis for this discussion, this document provides concrete examples of how ALTO can be integrated within CDNi request routing. The examples in this document are based on the use cases and examples currently being discussed in the CDNi WG.

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 April 21, 2012.

Copyright Notice
Table of Contents

1.  Introduction ................. 3
2.  CDNi Request Routing ........ 4
3.  Using ALTO within CDNi Request Routing .... 5
   3.1.  ALTO to simplify DNS-based Request Routing Redirection ... 5
   3.2.  ALTO to simplify http-Redirection for Request Routing ... 7
   3.3.  ALTO to support Selection of Downstream CDN .......... 9
4.  Security Considerations ........ 10
5.  Summary and Outlook ........... 11
6.  Acknowledgements ............... 12
7.  Informative References ........ 13
   Author’s Address ................ 14
1. Introduction

Many Network Service Providers (NSPs) are currently considering or have already started to deploy Content Delivery Networks (CDNs) within their networks. As a consequence of this development, there is a need for interconnecting these local CDNs. Content Delivery Networks Interconnection (CDNi) has the goal of standardizing protocols to enable such interconnection of CDNs [refs.cdniproblemstatement].

The CDNi problem statement envisions four interfaces to be standardized within the IETF for CDN interconnection [refs.cdniproblemstatement]:

- CDNI Request Routing Interface
- CDNI Metadata Interface
- CDNI Logging Interface
- CDNI Control Interface

This document focuses solely on the CDNI Request Routing Interface. In particular, this document shows concrete examples of how ALTO [RFC5693] can be integrated in CDNI request routing. The goal of this document is to show in what cases ALTO can benefit CDNi request routing, giving concrete examples and explaining how ALTO improves CDNi request routing in each of these examples. The examples used in this document are based on the use cases and request routing proposals currently being discussed in the CDNi WG [refs.cdniiusecases] [refs.cdnistrawman] and in the ALTO WG [refs.altocdn]. The overall rationale of this document is to foster discussions (in the CDNi as well as in the ALTO WG) regarding a) if, b) how, and c) under what conditions ALTO can be useful to optimize CDNi request routing.

Throughout this document, we use the terminology for CDNi defined in [refs.cdniproblemstatement].
2. CDNi Request Routing

The main purpose of the CDNI Request Routing Interface is described in [refs.cdniproblemstatement] as follows: "The CDNI Request Routing interface enables a Request Routing function in an upstream CDN to query a Request Routing function in a downstream CDN to determine if the downstream CDN is able (and willing) to accept the delegated content request and to allow the downstream CDN to control what the upstream Request Routing function should return to the User Agent in the redirection message". On a high level, the scope of the CDNI Request Routing Interface therefore contains two main tasks:

- A) Determining if the downstream CDN is willing to accept a delegated content request
- B) Redirecting the content request coming from an upstream CDN to the proper entry point or entity in the downstream CDN
3. Using ALTO within CDNi Request Routing

Application Layer Traffic Optimization (ALTO) is an approach for guiding the resource provider selection process in distributed applications that can choose among several candidate resources providers to retrieve a given resource. By conveying network layer (topology) information, an ALTO server can provide important information to "guide" the resource provider selection process in distributed applications. Usually, it is assumed that an ALTO server conveys information these applications cannot measure themselves [RFC5693].

Originally, ALTO was motivated by the huge amount of cross-ISP traffic generated by P2P applications [RFC5693]. Recently, however, ALTO is also being considered for improving the request routing in CDNs [refs.altocdn]. In this context, it has also been proposed to use ALTO for selecting an entry-point in a downstream NSP’s network (see section 3.4 "CDN delivering Over-The-Top of a NSP’s network" in [refs.altocdn]). Also, the CDNi problem statement explicitly mentions ALTO as a candidate protocol for "algorithms for selection of CDN or Surrogate by Request-Routing systems" [refs.cdniproblemstatement]. Yet, there have not been concrete proposals so far on how to use ALTO in the context of CDN interconnection. This document tries to close this gap by giving some examples on how ALTO could be used within CDNi request routing.

As explicitly being out-of-scope for CDNi [refs.cdniproblemstatement], the examples used in this document assume that ingestion of content or acquiring content across CDNs is not part of request routing as considered within CDNi standardization work. The focus of using ALTO (as considered in this document) is hence on request routing only, assuming that the content (desired by the end user) is available in the downstream CDN (or can be acquired by the downstream CDN by some means).

3.1. ALTO to simplify DNS-based Request Routing Redirection

If CDNi request routing is based on DNS, ALTO can potentially help to avoid one or more DNS resolution steps. For instance, Figure 1 shows a modified version of the high-level message sequence chart from Figure 5 of [refs.cdniframework] (note that this figure is similar to the high-level message sequence chart shown in Figure 3 of [refs.cdnistrawman]). In the original figure (i.e. Figure 5 in [refs.cdniframework]), the DNS server hosted by the upstream CDN (assumed to be the authoritative DNS server for the requested content), returns a DNS CNAME and NS record which essentially directs the end user to the request router of the downstream CDN. However, this redirection involves another DNS resolution for the request
router of the downstream CDN to be performed by the end user.

In the example using ALTO provided in Figure 1 of this document, the
downstream CDN provides the upstream CDN an ALTO network (and
corresponding cost) map by means of the ALTO protocol
[refs.altoprotocol] (0). This ALTO map provides sufficient
information for the upstream CDN to directly return a suitable IP-
address for the CDN entry point in the downstream CDN (2).

In principle, using ALTO this way the downstream CDN provider would
provide the decision on which delivery node is best by means of an
ALTO network map to the upstream CDN provider. This enables the
upstream CDN provider to directly return a suitable delivery node in
the downstream CDN to the end user as a response to the initial DNS
request received by the upstream CDN provider.

An implicit assumption for the example in Figure 1 to work is that the
CDN entry point for the downstream CDN only depends on the location
of the end user. Using the cost map, the upstream CDN can determine
the "best" entry point in the downstream CDN. If the "best" entry
point depends also on the target domain ("cdn.csp.com" in the
example), it becomes more tricky to make such information available
to the upstream CDN by means of ALTO network map and cost map. One
possible way to still use ALTO in this case would be for the
downstream CDN to provide a different cost map for each Content
Service Provider (CSP).
3.2. ALTO to simplify http-Redirection for Request Routing

Similar to the example given in the previous subsection, ALTO can also help to reduce intermediate http redirection steps. Figure 2 shows a modified version of the the high-level message sequence chart from Figure 3 of [refs.cdniframework] (note that this figure is similar to the high-level message sequence chart shown in Figure 1 of [refs.cdnistrawman]). In this case, the ALTO maps provided in step (0) by the downstream CDN potentially enable the upstream CDN to directly return - as a response to an http request - the hostname of the suitable cache (delivery node) in the downstream CDN by means of 302-redirection (4). This use of ALTO would enable to avoid several http-302 redirections and DNS resolutions by the end user (compare with figure 3 of [refs.cdniproblemstatement], steps (2-4)).

Depending on how dynamically the actual "best" delivery node changes (from the downstream CDN’s request routing perspective), it might only be meaningful to return to the end user the "best" entry point

Figure 1 - ALTO within DNS-based redirection of request routing

<table>
<thead>
<tr>
<th>End-User</th>
<th>Operator B</th>
<th>Operator A</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ALTO network/cost map</td>
<td>(0)</td>
</tr>
<tr>
<td>DNS cdn.csp.com</td>
<td></td>
<td>(1)</td>
</tr>
<tr>
<td>IPaddr of B’s CDN Entry Point</td>
<td></td>
<td>(2)</td>
</tr>
<tr>
<td>HTTP cdn.csp.com</td>
<td></td>
<td>(3)</td>
</tr>
<tr>
<td>DNS op-b-acq.op-a.net</td>
<td></td>
<td>(4)</td>
</tr>
<tr>
<td>IPaddr of A’s Delivery Node</td>
<td></td>
<td>(5)</td>
</tr>
<tr>
<td>HTTP op-b-acq.op-a.net</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
or cluster within the downstream CDN. This would require an additional http-redirect by the downstream request router to the "best" actual cache. However, even in such a case ALTO can save one http-redirect and one DNS resolution at the end user, consequently speeding up the overall process of CDNi request routing.

End-User | Operator B | Operator A
---|---|---
| DNS cdn.csp.com | ALTO network/cost map | Operator A
| IPaddr of A’s Request Router | | ALTO network/cost map
| HTTP cdn.csp.com | | Operator A
| 302 node1.peer-a.op-b.net/cdn.csp.com | | ALTO network/cost map
| DNS node1.peer-a.op-b.net | | Operator A
| IPaddr of B’s Delivery Node | | ALTO network/cost map
| HTTP node1.peer-a.op-b.net/cdn.csp.com | | Operator A
| DNS op-b-acq.op-a.net | | ALTO network/cost map
| IPaddr of A’s Request Router | | Operator A
| HTTP op-b-acq.op-a.net | | ALTO network/cost map
| 302 node2.op-b.acq.op-A.net | | ALTO network/cost map
| DNS node2.op-b.acq.op-a.net | | Operator A
| IPaddr of A’s Delivery Node | | ALTO network/cost map
| Data | | Operator A

Data
3.3. ALTO to support Selection of Downstream CDN

ALTO could also help for the upstream CDN provider to select a proper downstream CDN provider for a given end user request. For instance, a network map provided by each of several candidate downstream CDNs could provide information to the upstream CDN provider regarding the geographical coverage, the location of "surrogates", or similar. Future versions of this document will discuss this use case in more detail, and provide concrete examples how ALTO can be used for downstream CDN selection by the upstream CDN provider.
4. Security Considerations

Security Considerations will be discussed in a future version of this document.
5. Summary and Outlook

This document presented some examples on how ALTO can be used within CDNi Request Routing and argued why such use of ALTO is meaningful in certain cases. The intention of this document is to arouse discussions in the CDNi WG as well as the ALTO WG in order to find consensus on scenarios where ALTO is beneficial to CDNi request routing and to form agreement on how ALTO should be used in such cases within the CDNi request routing protocol. It is the intention to capture the outcome of such continuing discussions in future versions of this document.
6. Acknowledgements

Jan Seedorf is partially supported by the COAST project (COntent Aware Searching, retrieval and sTreaming, http://www.coast-fp7.eu), a research project supported by the European Commission under its 7th Framework Program (contract no. 248036). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the COAST project or the European Commission.
7. Informative References

[refs.cdniusecases]

[refs.cdniproblemstatement]

[refs.altocdn]


[refs.cdnistrawman]

[refs.cdniframework]

[refs.altoprotocol]
Author’s Address

Jan Seedorf
NEC Laboratories Europe, NEC Europe Ltd.
Kurfuersten-Anlage 36
Heidelberg  69115
Germany

Phone: +49 (0) 6221 4342 221
Email: jan.seedorf@neclab.eu
URI: http://www.neclab.eu
Abstract

Recently, there has been a discussion on "Infrastructure-to-Application Information Exposure" and the related communications requirements in fully controlled (e.g. data centers) or partially controlled environments (e.g. CDN). One possibility to expose infrastructure information to applications in the aforementioned environments is to use yet-to-be-defined ALTO extensions. This draft discusses requirements for such ALTO extensions for a specific use case: Request Routing in CDN Interconnection (CDNI).

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 September 6, 2012.

Copyright Notice

Copyright (c) 2012 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.
Table of Contents

1. Introduction ........................................... 3
2. Useful Extensions to ALTO to facilitate CDNI Request
   Routing with ALTO ....................................... 4
3. Security Considerations ................................. 6
4. Conclusion ............................................... 7
5. Acknowledgements ....................................... 8
6. Informative References .................................. 9
Author’s Address ........................................... 10
1. Introduction

Recently, there has been a discussion on "Infrastructure-to-Application Information Exposure" and the related communications requirements in fully controlled (e.g. data centers) or partially controlled environments (e.g. CDNs) [I-D.marocco-alto-next]. One possibility to expose infrastructure information to applications in the aforementioned environments is to use ALTO [I-D.ietf-alto-protocol], or rather extension to ALTO which are yet to be investigated and specified [I-D.marocco-alto-next].

Network Service Providers (NSPs) are currently considering to deploy Content Delivery Networks (CDNs) within their networks. As a consequence of this development, there is a need for interconnecting these local CDNs. The necessary interfaces for inter-connecting CDNs are currently being defined in the Content Delivery Networks Interconnection (CDNI) WG [I-D.ietf-cdni-requirements] [I-D.ietf-cdni-problem-statement] [I-D.ietf-cdni-use-cases]. Among other protocols, ALTO has been suggested to facilitate downstream CDN selection in the context of CDNI request routing [refs.altocdni], i.e. ALTO is currently a candidate protocol for "Footprint and Capabilities Advertisement" within CDNI Request Routing.

This document discusses what extensions to ALTO would be useful in case ALTO is used as a protocol within CDNI request routing, and in particular within the "Footprint and Capabilities Advertisement" part of the CDNI request routing interface. The discussion is based on the suggested extensions to ALTO proposed in [I-D.marocco-alto-next] and the examples of how ALTO could be used for downstream CDN selection within CDNI request routing in [refs.altocdni].
2. Useful Extensions to ALTO to facilitate CDNI Request Routing with ALTO

In the following, we assume the following scenario of using ALTO for selecting a downstream CDN [refs.altocdni]: Each downstream CDN provider hosts an ALTO server which provides ALTO information (i.e. ALTO network maps and ALTO cost maps [I-D.ietf-alto-protocol]) to an ALTO client at the upstream CDN provider. A network map provided by each of several candidate downstream CDNs can provide information to the upstream CDN provider regarding the geographical coverage, the location of "surrogates", or similar. In addition, an ALTO cost map can provide an upstream CDN provider information about the "cost" of delivering certain content via the downstream CDN which provided such a cost map. "Cost" in this context is a generic term; many types of costs are possible and can be useful in the context of CDNI request routing, e.g. average link load, expected delay, or monetary costs.

As an example, an upstream CDN (uCDN) receives a request from an end user. Based on the IP-address of the end user, uCDN determines that it is possible to deliver the content from one of several candidate downstream CDNs (dCDN-a, dCDN-b, and dCDN-c). Assume that only the ALTO network maps provided by dCDN-a and dCDN-c indicate that these downstream CDNs can deliver content for the location of the end user requesting content. In this case, the ALTO costs maps provide useful information to the upstream CDN, uCDN, in order to make a selection decision regarding either dCDN-a or dCDN-c.

From the perspective of the given scenario and example, the following proposed extensions to ALTO would be beneficial to facilitate CDNI request routing with ALTO:

- **Server-initiated Notifications and Incremental Updates:** In case the footprint or the capabilities of a downstream CDN change abruptly (i.e. unexpectedly from the perspective of an upstream CDN), server initiated notifications would enable a dCDN to directly inform an upstream CDN about such changes. Consider the case where - due to failure - part of the footprint of the dCDN is not functioning, i.e. the CDN cannot serve content to such clients with reasonable QoS. Without server-initiated notifications, the uCDN might still use a very recent network and cost map from dCDN, and therefore redirect request to dCDN which it cannot serve. Similarly, the possibility for incremental updates would enable efficient conveyance of the aforementioned (or similar) status changes by the dCDN to the uCDN.

- **Content Availability on Hosts:** A dCDN might want to express CDN capabilities in terms of certain content types (e.g. codecs/
formats, or content from certain content providers). A new endpoint property for ALTO that would be able to express such "content availability" would enable a dCDN to make available such information to an upstream CDN. This would enable a uCDN to determine if a given dCDN actually has the capabilities for a given request with respect to the type of content requested.

- Resource Availability on Hosts or Links: The capabilities on links (e.g. maximum bandwidth) or caches (e.g. average load) might be useful information for an upstream CDN for optimized downstream CDN selection. For instance, if a uCDN receives a streaming request for content with a certain bitrate, it needs to know if it is likely that a dCDN can fulfill such stringent application-level requirements (i.e. can be expected to have enough consistent bandwidth) before it redirects the request. In general, if ALTO could convey such information via new endpoint properties, it would enable more sophisticated means for downstream CDN selection with ALTO.
3. Security Considerations

Security Considerations will be discussed in a future version of this document.
4. Conclusion

Recently, there has been a discussion on "Infrastructure-to-Application Information Exposure" and the related communications requirements in fully controlled (e.g. data centers) or partially controlled environments (e.g. CDN). One possibility to expose infrastructure information to applications in the aforementioned environments is to use yet-to-be-defined ALTO extensions. This draft considered requirements for such ALTO extensions for a specific use case: Request Routing in CDN Interconnection (CDNI). This document discussed useful extensions to ALTO in this context, i.e. what extensions would be beneficial from the CDNI perspective and why.
5. Acknowledgements

Jan Seedorf is partially supported by the COAST project (COntent Aware Searching, retrieval and sTreaming, http://www.coast-fp7.eu), a research project supported by the European Commission under its 7th Framework Program (contract no. 248036). The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the COAST project or the European Commission.
6. Informative References

[I-D.ietf-alto-protocol]  
Penno, R., Alimi, R., and Y. Yang, "ALTO Protocol",  
draft-ietf-alto-protocol-10 (work in progress),  
October 2011.

[I-D.ietf-cdni-problem-statement]  
Niven-Jenkins, B., Faucheur, F., and N. Bitar, "Content  
Distribution Network Interconnection (CDNI) Problem  
Statement", draft-ietf-cdni-problem-statement-03 (work in  
progress), January 2012.

[I-D.ietf-cdni-requirements]  
Leung, K. and Y. Lee, "Content Distribution Network  
Interconnection (CDNI) Requirements",  
draft-ietf-cdni-requirements-02 (work in progress),  
December 2011.

[I-D.ietf-cdni-use-cases]  
Gilles, B., Watson, G., Ma, K., Eardley, P., Emile, S.,  
and T. Burbridge, "Use Cases for Content Delivery Network  
Interconnection", draft-ietf-cdni-use-cases-03 (work in  
progress), January 2012.

[I-D.marocco-alto-next]  
Marocco, E. and V. Gurbani, "Extending the Application-  
Layer Traffic Optimization (ALTO) Protocol",  
draft-marocco-alto-next-00 (work in progress),  
January 2012.

[refs.altocdni]  
Seedorf, J., "ALTO for CDNi Request Routing",  
draft-seedorf-cdni-request-routing-alto-00 (work in  
progress), March 2012.
Author’s Address

Jan Seedorf
NEC Laboratories Europe, NEC Europe Ltd.
Kurfuersten-Anlage 36
Heidelberg  69115
Germany

Phone: +49 (0) 6221 4342 221
Email: jan.seedorf@neclab.eu
URI: http://www.neclab.eu