Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: July 13, 2013

H. Xie
Huawei & USTC
T. Tsou
Huawei (USA)
D. Lopez
Telefonica I+D
H. Yin
Huawei (USA)

January 9, 2013

# Use Cases for ALTO with Software Defined Networks
# draft-xie-alto-sdn-extension-use-cases-01

Abstract

The introduction of SDN fundamentally changes the way that the application layer traffic optimization (ALTO) works. This draft describes two architectures, the Vertical Architecture and the Horizontal Architecture, allowing coherent coexistence of ALTO and software defined network (SDN). Unique requirements for design and operations are identified and summarized, suggesting that the Vertical Architecture allows better division, management, flexibility, privacy control and long-term evolution of the network. We also define the main interactions and information flows, and present a set of use cases to illustrate how we extend ALTO to support SDN, in the Vertical Architecture.

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 July 13, 2013.

Table of Contents

## 1. Introduction

The concept of Software Defined Network (SDN) has emerged and become one of the fundamental, promising networking primitives that allow flexibility of control, separation of functional planes and continuous evolution in networks.

One of the key features of SDN is the full separation of two functional planes: the control plane and the data-forwarding plane. SDN requires that networking devices support such separation, implementing the data plane mechanisms, while the control plane is provided by an external entity called the "controller". The other key feature of SDN is the new interaction process between the separated control plane and data-forwarding plane. This interaction is mandated to be performed specific open protocols, allowing for a free combination of networking devices and controllers, as well as supporting the controller to take decisions on information additional to the networking device status.

There could be numerous benefits as a result of the above features in SDN, e.g., just to name a few, network virtualization, better flexible control and utilization of the network, networks customized for scenarios with specific requirements. For instance, some SDN technologies have started to find their ways into Data Center Networks (DCNs). Furthermore, in order to allow efficient and flexible implementation of the above separation and interactions, a significant portion of the SDN system could typically be implemented in software, as opposed to the hardware-based implementation adopted by most of today's networking devices.

Due to the great potentials of SDN and the unique requirements of DCNs, Data Centers are likely to become a first place where SDN could be deployed. We envision that SDN could be gradually adopted by enterprise networks and then by carrier networks due to its unique, attractive features. When deploying SDN in large-scale distributed networks, we expect to see a collection of deployments limited in relatively small segments of a bigger network. In other words, it is likely that the operator of a large-scale enterprise / carrier network prefers to divide the whole networks into multiple smaller segments and put each of there segments in an SDN domain. These smaller network segments (therefore their corresponding SDN domains) are connected and jointly form the larger whole network. Such a divide-and-conquer methodology not only allows gradual deployment and continuous evolution, but also enables flexible provisioning of the network.

With the deployment of SDN, application layer traffic optimization (ALTO) faces new challenges including, but not limited to, privacy

preservation, granularity of information collection and exchange, join optimization, etc. We shall elaborate these challenges and their impacts on the design of ALTO extensions for SDN in this draft.

### 1.1. Terminology

While the definition of software defined networks is still "nebulous" to some extent, we refer to as SDN the networks that implement the separation of the control and data-forwarding planes and software defined interactions between these two separated planes; such interactions are characterized by open interfaces which allow programming the network equipment's forwarding plane by external agents, e.g., SDN controllers. However, how the two separated planes interact is not a focus of this draft; instead, the ALTO extension for SDN recommended in this draft is independent of how such interactions would be.

An SDN domain is a portion of a network infrastructure, consisting of numerous connected networking devices that are SDN capable (i.e., SDN capable devices implement the control/forwarding plane separation and the open interfaces allowing external agents to program the devices) and a domain controller which implements the SDN control plane functionalities for these devices. An SDN domain can be as small as a sub-network of a dozen devices or as large as a medium/large data center network.

A software defined network is a network that has one or multiple SDN domains. Due to an SDN domain typically has limited coverage, an SDN may be comprised of networking devices under control of some SDN domains (i.e., SDN managed devices) and devices under no control of any SDN domain (i.e., normal devices). Note that such normal devices could still be SDN capable and their control/forwarding planes are managed as in normal networks today. This implies that a network with both normal devices and SDN capable devices (managed by SDN domains) needs both normal and SDN capable control/forwarding plane management.

## 2. An Overview of Software Defined Network

This section provides a high level and conceptual overview of software defined network in order to help illustrate its unique requirements for ALTO.

### 2.1. Software Defined Network and Applications

We refer to as an "SDN" a carrier's or an enterprise's network which deploys or implements software defined networking technologies.

Since SDN separates the control plane and data-forwarding plane, we refer to the entity that implements the control-plane functionalities as the "SDN controller".

In order for a network to be classified as an SDN, it is unnecessary that all networking devices have to be SDN capable. Some of devices in a network may be managed by an SDN controller while the remaining ones may not; such a network is still qualified as an SDN.

There are two types of applications in software defined networks:

- SDN-aware applications: applications prefer direct communication with SDN controllers, which implies that there must exist mechanism(s) for SDN-aware applications to locate and communication with SDN controllers. If the application prefers indirect communication with SDN controllers, then it follows the case of SDN-unaware applications (see below). Applications that are SDN-aware may be able to better utilize the SDN capable network due to its knowledge about SDN and its capability of proactive, direct interaction with SDN.

- SDN-unaware applications: applications indirectly communicate with SDN controllers by sending application protocol datagrams with specific formats, via which the application can specify its requirements for the network resources. Legacy applications (applications for the current IP networks) are largely SDN-unaware, and such applications may not be able to utilize the SDN capable networks as efficiently as SDN-aware applications.

Whether and how applications should/can be SDN-aware or SDN-unaware is beyond the scope of this draft. However, the framework we described in this draft is applicable to both SDN-aware and SDN-unaware cases.

## 2.2. Division of Network

A network operator may decide to divide the network into multiple sub-networks, some of which are SDN capable and thus are managed by corresponding SDN controllers.

There could be numerous reasons for such division of network. Below we summarize a few of them:

- Scalability.

  The number of devices an SDN controller can feasibly manage is likely to be limited. Therefore, in order to manage a many devices that cannot be put under control of a single SDN

controller, multiple controllers have to be deployed. Hence, the network is divided into multiple sub-networks; if a sub-network has SDN capable devices, it should be managed by an SDN controller.

- Manageability.

  At the network level, in order to reduce the complexity of management, a carrier may decide to divide the network into multiple sub-networks and put some of them under control of some SDN controllers (provided that the devices in such sub-networks are SDN capable); each of the sub-networks can be managed independently to some extent, thus reducing the overall complexity of managing the whole network. Even at the sub-network level, a carrier may still decide to further divide the sub-network in order to further reduce complexity of management. For instance, a sub-network under control of an SDN controller may span across a large geographical area or cover a large number of devices; in this case it is reasonable for the carrier to further divide it into two or even more sub-networks, such that the complexity of managing each individual sub-network plus the overall complexity of managing all divided sub-networks are reduced.

- Privacy.

  When a carrier divides its network into multiple sub-networks and put them under control of SDN, the carrier may choose to implement difference privacy policies in different sub-networks. For example, a carrier may dedicate a part of its infrastructure to some certain customers, dividing the whole network and dedicate some of the sub-networks is a convenient and scalable way to manage the network resources for these customers. Another example is that a sub-network in a carrier's network may be dedicated to some certain customers and such information as network topology may not be disclosed to any external entity.

- Deployment

  The deployment of network infrastructures, especially new network infrastructure and technologies, has to be incremental. This means that at any time, a carrier's network may consist of a portion of legacy and a portion of non-legacy infrastructure. When deploying new infrastructure or technologies, it is highly preferable to limit the scope of new deployment and do it in an incremental way. In such cases, it is very favorable to divide the network into multiple individually manageable sub-networks and choose some of them to deploy the new infrastructure / technologies.

### 2.3. SDN Domain

With the introduction of SDN, we believe that it is inevitable for carriers to divide their networks for many reasons as illustrated in the preceding subsection. Therefore, we believe that to better suit this need, it should be recommended that SDN domains are defined and applied in division of the networks.

An SDN domain is a sub-network, resulted from division of a network which is determined by the network operator. Each domain typically consists of numerous connected networking devices, each of which is SDN capable. Each domain also has a domain controller which implements SDN control plane functionalities for the devices in this domain. Another important task such a domain controller is responsible for includes fine-grain network information collection (for devices in this domain). The collected information is necessary for the controller to make data-forwarding plane decisions. Note that such a domain controller may be integrated as a part of a so-called "network operating system" (NOS). An example of such a network operating system is illustrated in [1].

Any networking device, if under the control of certain SDN domains, should belong to only one SDN domain and should be under the control of the SDN domain's controller. In other words, the intersection of two domains is always empty.

Furthermore, SDN domains are connected (via the connectivity among underlying devices provided by the underlying network; such devices belong to different SDN domains) to form the whole network. For a large-scale distributed networks owned by a national/multi-national carrier or enterprise, it is natural to adopt the "divide-and-conquer" strategy and divide the whole network into multiple SDN domains. Even for small or medium networks, multiple SDN domains may be necessary in order to virtualize the network resources (e.g., set up multiple SDN domains for a large data center network to instantiate multiple virtual data centers for many content providers). Note that how multiple SDN domains inside a carrier's/ enterprise' network work together is beyond the scope of this draft and is handled by other working groups.

Inside each SDN domain, its controller may define domain-specific policies on information importing from devices, aggregating, and exporting to external entities. Such policies may not be made public; therefore, other domain controllers or ALTO may not know the existence of such policies for any given SDN domain.

The natural network division implemented by SDN domains impose significantly new and challenging requirement for shaping the

interactions between SDN and ALTO, and therefore designing the protocols to enable such interactions.

## 3. Architectures for Co-existing SDN and ALTO

In this section, we first compare the ALTO scopes without and with the existence of SDN, and then describe two architectures for co-existing SDN and ALTO.

### 3.1. ALTO Changes Due to SDN

SDN incurs significant changes to ALTO scopes and clients. We describe the major differences below.

#### 3.1.1. ALTO Scopes

The existence of SDN differentiates two scopes of ALTO, namely,

- The current scope of ALTO without SDN (referred to as the SDN-unfriendly Scope).

  In the current scope of ALTO, there exist interactions between ALTO clients and ALTO servers. Such interactions are one-way interaction, meaning that ALTO information flows in one direction, i.e., from the server to the clients. More specifically, ALTO clients submit ALTO requests to (and subsequently receive ALTO responses from) an ALTO server. Additionally, ALTO clients in the current scope of ALTO are network applications who would like to consume the network resources. ALTO clients are typically managed by network applications rather than by network carriers. However, ALTO servers are owned and managed by network carriers.

- The new scope of ALTO with coherent coexistence of SDN (referred to as the SDN-friendly Scope).

  With the introduction of SDN, the interactions between ALTO clients and ALTO servers become more complex. A more careful examination as well as important ALTO extensions are necessary to make ALTO work in the context of SDN. It is important to note that if the network does not implement network division (i.e., does not implement SDN domains), the interactions are "compressed" into a compact set of interactions; specifically, both the SDN controller (recall that there exists only one single controller for the whole network) and the ALTO server could be integrated in many equally efficient fashions. For instance, ALTO server could be put underneath the controller, i.e., ALTO server provides information to the controller, as suggested by [2]. However, if

the network implements division via SDN domains (i.e., there exists multiple SDN domains), we essentially "unfold" the compressed interaction space and need more complex structures that allow efficient design and implementation, due to the facts that we listed in the preceding sections. Furthermore, the design originally for the compressed space could be instantiated for the unfolded space but could not be as efficient.

### 3.1.2. ALTO clients

We next focus on the SDN-friendly Scope and highlight the complex structures and the important differences.

With the existence of SDN and SDN controllers, ALTO clients could be not only legacy network applications (or proxies for these network applications), but also SDN controllers.

In SDN, SDN controllers have similar needs as the legacy ALTO clients which communicate with ALTO servers, because ALTO clients would like to better understand the network and thus improve the application performance.

There are multiple reasons for this similarity. The most important reason is that each SDN controller is only responsible for managing a sub-network in a carrier's network; therefore, it may not understand well other sub-networks in the same carrier network. However, in order to allocate the network resources to satisfy application requirements (note that the end points of such applications may well span across multiple SDN domains), an SDN controller may have to involve other SDN controllers because the network paths needed may traverse multiple SDN domains. Thus, obtaining global information from the ALTO server is a significantly more efficient and more preferable than otherwise from SDN interconnection protocols; plus, such protocols do not exist yet today.

Although SDN controllers have similar needs as legacy ALTO applications do, the fundamental properties of SDN controllers as ALTO clients are significantly different. One of the differences is the ownership and management, as most SDN controllers require additional (and more likely complex) access privileges. Specifically, SDN controllers are typically owned by the network carriers who also own their ALTO servers, while legacy ALTO clients are network applications typically not owned by network carriers (there are cases where owned collaboratively amongst operators).

In terms of management, the entity managing SDN controller is the same as that managing the ALTO server. We emphasize that when an SDN domain is dedicated to some customers of a network carrier, the use

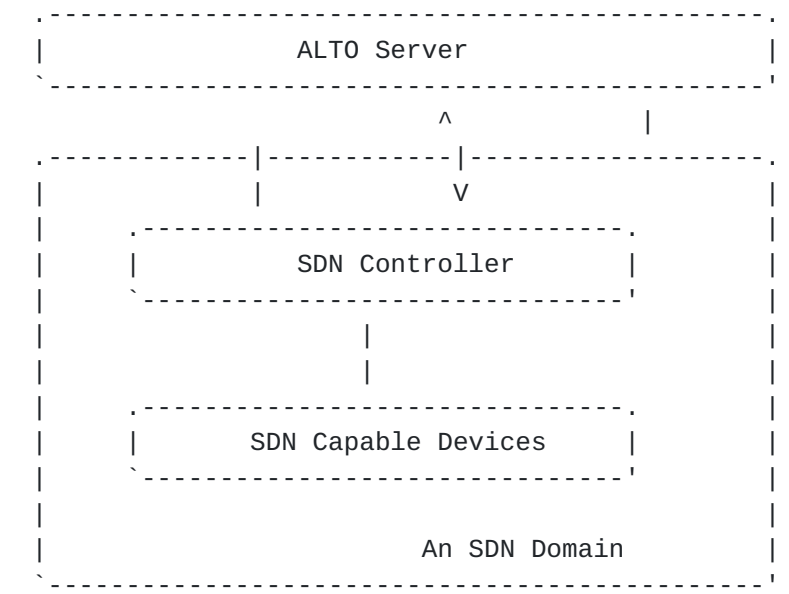
of the SDN domain is owned by these customers and so is the management. In this case, the SDN controllers as ALTO clients are slightly more like legacy ALTO clients. However, there still exist fundamental differences which we will illustrate later.

## 3.2. The Vertical and Horizontal Architectures

We now introduce two architectures that allow coherent co-existence of SDN and ALTO in this section:

- the Vertical Architecture (or the V Architecture for short) allows better division, management, flexibility, privacy control and long-term evolution of the network.

  The Vertical Architecture is illustrated in the following figure. The network has one or multiple SDN domains and an ALTO server. The interactions between the SDN controllers and the SDN capable devices fall in the scope of SDN and therefore is out of the scope of this draft; instead, the interactions between the SDN domains (more specifically, SDN controllers) and the ALTO server is what this draft focuses on.

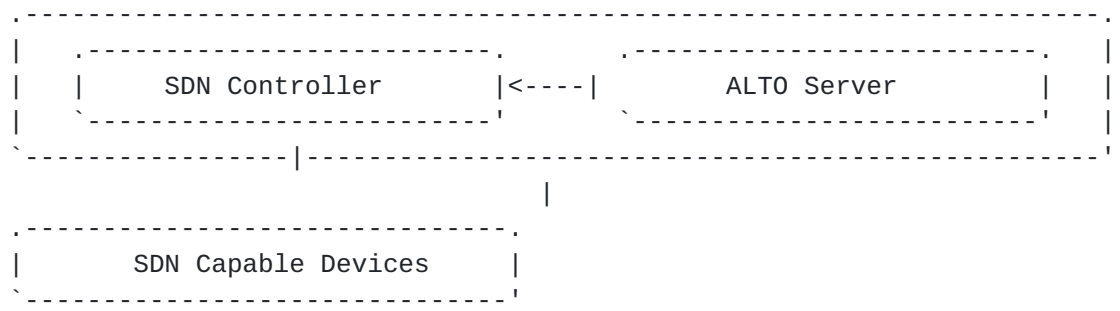



  The Vertical Architecture is a hierarchical architecture consisting of three tiers. In the first tier, the only entity is the ALTO server. In the second tier, the only entities are the SDN domain controllers. In the third tier, the only entities are SDN domains (each domain consists of numerous networking devices).

  In this architecture, all entities are owned by the same carrier. However, some of the SDN domains (and the networking devices in

them) may be dedicated to certain customers of the carrier (the carrier gives the customers privileges access). This is subject to a collaboration agreement between them.

- the Horizontal Architecture (or the H Architecture for short) simplifies the implementation of ALTO extensions for SDN. The Horizontal Architecture is illustrated in the following figure. Each SDN controller is integrated with an ALTO server. The interactions between the SDN controllers and the ALTO server is represented by the horizontal line in the figure. An example of this architecture can be found in [2].




In the Horizontal Architecture, the SDN controller can act as an ALTO client and query the network information of the networking devices from the ALTO server. However, such network information may not meet the SDN controllers' granularity requirement (i.e., the information provided by the ALTO server may not be as fine-grained as needed by the SDN controllers). In addition, there may exist redundant information collection, as SDN controllers are inevitably collecting various fine-grain information from the devices they manage; the information collection by the ALTO server, either mannully or automatically, is likely to be redundant. Furthermore, when the carrier decides to divide its network into multiple SDN domains, it can be difficult, if not impossible at all, for the Horizontal Architecture to adapt to the network division.

The ALTO server covers a carrier's network as a whole; however, if the carrier divide the network into multiple SDN domains, each SDN domain covers only a segment of the network. Additionally, the ALTO server has only relatively coarse-grained information, while SDN domain controllers could easily collect more fine-grained information. More importantly, different SDN domains may implement different information aggregation and privacy policies (e.g., when such domains are dedicated to certain customers of the carrier).

These observations suggest that the Vertical Architecture is

advantageous over the Horizontal Architecture. With the Vertical Architecture, SDN and ALTO are explicitly separated and as a result the logic is cleaner and this allows them to evolve independently. Furthermore, the Vertical Architecture makes automated information collect possible for ALTO, which could make ALTO deployment and management easier and more attractive. Therefore, in the remainder of this draft, we mainly focus on the Vertical Architecture. We will describe the interactions and the information flows in further details for the Vertical Architecture.

### 3.3. Interactions between SDN and ALTO

The interactions between SDN controllers (as ALTO clients) and ALTO servers are significantly different. Legacy ALTO clients retrieve information from ALTO servers. However, SDN controllers may also need to push information to ALTO servers. In a carrier's network, SDN controllers and the ALTO server are owned by the same carrier. Interactions between them could be significantly more complex.

The interactions between the SDN controllers and the ALTO server can be divided into two categories. We refer to as the "upward flow" the information flow from the second tier (SDN controllers as ALTO clients) to the first tier (ALTO server), and refer to as the "downward flow" the information flow in the reverse direction, i.e., from the first tier (ALTO server) to the second tier (SDN controllers as ALTO clients).

#### 3.3.1. Downward Interaction

The downward interaction is the information flow from ALTO servers to ALTO clients (i.e., SDN controllers). Each SDN controller is also an ALTO client and retrieves relevant network information from the ALTO server. This is similar to the current scope of ALTO without the existence of SDN; however, the differences are that with the existence of SDN, the network information is generally specific to SDN and SDN domains; SDN controllers as ALTO clients could query the ALTO server for either inter-domain or intra-domain network information (provided that intra-domain information is reported and made available).

The fundamental difference is a result of SDN and SDN domain division, which do not exist in legacy network application scenarios. For instance, an SDN controller for a specific SDN domain may be interested in obtaining internal information of other SDN domains (provided that other domains allow to do so), or obtaining domain-level information such as inter-SDN-domain connectivity. None of these is applicable to legacy ALTO client scenarios. As a result, ALTO server and its protocol should be extended to support such

scenarios.

### 3.3.2. Upward Interaction

The upward interaction is the information flow from ALTO clients (i.e., SDN controllers) to ALTO servers. SDN controllers open up the possibilities of conveniently collecting network information and exporting such information to ALTO servers. SDN controllers are at the best position to collect network information.

More importantly, it is an inevitable requirement that SDN controllers collect the information of the networking devices in its domain. Each SDN controller may collect network information from the devices managed by it and information from other SDN controllers), and report such information to the ALTO server, subject to the information aggregation and privacy policies defined for the corresponding individual SDN domain. Such network information is referred to the inter-domain network information. The network information could include key information such as domain-level network cost, bandwidth, domain-specific connectivity, etc. The upward interactions could be implemented in either the push model or the pull model.

For instance, an SDN domain could be dedicated to some of a carrier's certain customers; the usage of such a domain gives privileged client access. However, such a domain is an integral sub-network of the carrier's network. In such a case, the ALTO server for the carrier's network is not able to collect necessary information in a scalable, manageable way. Even if we assume that the ALTO server can automatically pull necessary information directly from networking devices, the dedicated domain may disallow the ALTO server to do so, because customers who own and manage this domain may enforce stringent privacy policies and disallow exporting information externally. The SDN controller is the best entity that can facility the automation of information collection while at the same time enforce the specific privacy policy.

It is worth noting that network information collection has not been explored, and that network information collection could introduce significant overhead and complexity, in the current scope of ALTO. However, automated network information collection is a key to the success of ALTO. With the help of SDN and the Vertical Architecture, such automated network information collection becomes feasible and appealing. Note that this does not exclude the possibility of network operators manually or automatically update the ALTO server with the network information (e.g., the network cost map). It is also worth noting that an SDN controller may choose to report its domain-specific network information only (referred to as the intra-

domain information), with or without privacy policies. In this case, SDN controllers become an automated information collector for the ALTO server.

### 3.4. Interactions between Legacy ALTO Clients and ALTO Servers

With the existence of SDN, the way that legacy network applications (i.e., as legacy ALTO clients) interact with ALTO servers is also different.

In legacy ALTO client/server scenarios, ALTO clients obtain cost maps from ALTO servers, with the implicit assumption that ALTO servers understand how the underlying network routes packets, which allows ALTO servers to define or compute a cost metric associated with a given route.

However, with the introduction of SDN, such assumption may no longer hold, as SDN controllers may dynamically negotiate and determine a route between two end points (which may belong to two different SDN domains), especially when applications have specific requirements for network resources (e.g.bandwidth, delay, etc). Thus, in order for applications to best utilize the network resources, the way that legacy ALTO clients communicate with ALTO servers should be adapted to SDN.

## 4. Information Flows

We now further describe the two different information flows through two sets of use cases, one for the information flow from ALTO servers to ALTO clients, the other for the information flow from SDN controllers to ALTO servers.

### 4.1. Information Flow of SDN Controller

A network may consist of multiple SDN domains. Note that due to operational or deployment concerns, there may exist networking devices that do not belong to any SDN domain. In each SDN domain, the SDN controller is responsible for the following tasks (only ALTO related tasks are included below):

- Collect fine-grain information from the networking devices it manages. Such information could include, but not limited to, SDN domain topology, link capacity, available bandwidth on each link, links connected to external devices belonging to other SDN domains.

- Implement pre-defined domain-specific policies. Such policies could include, but not limited to, how resources should be allocated, how the collected information should combined and presented.

- Optionally aggregate the collected information for external view purposes per its policies.

- Obtain cost maps at the granularity of SDN domains or obtain internal cost maps for specific domains (if available), consult for cross-domain data-forwarding plane recommendations from ALTO.

- Make (ALTO recommended) data/forwarding plane decisions based on the cost maps obtained from ALTO.

### 4.2. Information Flow of Applications, SDN and ALTO

We now give three examples to describe a complete work flow, which connects all key elements in an SDN.

#### 4.2.1. SDN-aware Applications

- An application's end point sends a request for network resources to the SDN controller it belongs to (i.e., the SDN controller for the SDN domain where this application's end point belongs to). The request should include the destination end point or the set of destination end points, and a set of requirements on network resources (e.g., bandwidth)

- The SDN controller obtains an SDN-specific cost map from the ALTO server (this step may occur independent of remaining steps)

- The SDN controller uses the cost map and negotiate one or many path(s) with other SDN controllers (since the path may span across multiple SDN domains, thus all SDN controllers of the involved domains should participate in setting up the paths)

- The SDN controller responds to the requesting application's end point.

- If the requested path(s) are successfully set up, the application's end point starts to communicate with the destination end points.

### 4.2.2. SDN-unaware Applications

SDN-unaware applications do not directly communicate with SDN controllers. Instead, they follow special packet formatting rules to encode the SDN-specific requests, and the SDN capable networking devices pick up these requests and forward them the SDN controllers.

The remaining work flow is similar to the work flow of SDN-aware applications, except that SDN controllers do not respond to the requesting applications. Thus, when the requests cannot be satisfied, SDN-unaware applications may suffer from packet losses, due to networking devices process these applications' packets in a best effort fashion.

### 4.2.3. Legacy Applications

Legacy applications can be greatly simplified, as it is unnecessary and is not helpful for them to directly communicate with ALTO servers any more:

- An end point of a legacy application sends a packet to a known destination
- A SDN capable networking device picks up the packet; however, if the path for the two end points has not been set up yet, the SDN controller will be consulted
- The SDN controller obtains a cost map from the ALTO server (this step may occur independent of remaining steps).
- The SDN controller negotiate with other SDN controllers to set up a best-effort path for the requesting end point.
- The forwarding rules for this path are pushed to all networking devices that are on the chosen path
- Communications between the two end points continue; the forwarding rules may expire if the communication is tore down

In this case, legacy applications are relieved from the complexity of dealing with the ALTO server using the ALTO protocol. ALTO-related intelligence, which fundamentally belongs to the network intelligence, is implemented in the network, rather than partly outside the network.

### 4.3. Summary

It is worth noting that this architecture is fundamentally different from common ALTO use cases such as ALTO in CDN or data center (DC). The differences lie in that in the latter cases the components in question (e.g., CDN or DC) are largely consumers of ALTO services, while in the former case SDN domains are not only making decisions that may affect ALTO and generating/aggregating information that ALTO needs, but also the consumers of ALTO services. Furthermore, in the former case, SDN domains are an integral part of the underlying network infrastructure where their decisions could be treated as constraints for ALTO; however, in the latter cases, the components in question (e.g., CDN or DC) are apparently not necessarily integral parts of the underlying network and their decisions could be treated as recommended outcomes suggested by ALTO.

## 5. Messaging

The information exchanged between the SDN domain controllers and the ALTO server is encoded and implemented by specific messaging mechanisms. Below we describe a preferred messaging mechanism where we focus mainly on the semantics of the messages.

Based on the ALTO services, there are two-way message exchanges between the SDN controller and the ALTO server. NBI is used and should be adapted to accommodate such message exchanges. The concept of SDN domain is enforced by the controller if its policy defines so; therefore, the controller can opt to export the relevant information at policy-specific granularities.

### 5.1. Service Negotiation

SDN Domain controllers can communicate with the ALTO server to negotiate any or all of the service information described in the next two subsections. After negotiation, such information can be pulled from or pushed to ALTO server depending further on the communication mechanism provided by NBI. Further, the detail mechanism of consuming the above information will depend on the types of ALTO services being offered and not be covered by this use case.

### 5.2. Status Report (Upward Information Flow)

- network "node" information (its granularity is specified by controller's policy), mainly including network and/or geographical location, services, etc

- network "topology" information (at a granularity specified by controller's policy), mainly including SDN-domain-level (interdomain) topology and an abstract SDN intradomain topology if any; if the policy allows, controller can also export detailed intradomain topology (the granularity should be specified by the policy).

- network "link" information, similar to "node" and "topology", such information (e.g., link usage and state like congestion, delay, cost etc) is policed by the controller's policy and could be exported at different levels of granularity

- network "routing" information, for flows defined in flow tables, at the policy-specified granularity

- path information, about the path initiation and status policed by controller's policy.

### 5.3. ALTO Message Dissemination (Downward Information Flow

It is important to note that the vanilla ALTO service (i.e., cost map or path cost information) is no longer directly applicable to the context of co-existing SDN and ALTO.

In vanilla ALTO service scenarios, paths (i.e., routing between any pair of routers) are deterministic a prior, regardless of ALTO recommendations. However, in the context of co-existing SDN and ALTO, routing is to be determined based on many factors including ALTO. For instance, the routing between any pair of two SDN capable routers may not be fully determined when the SDN domain controller(s) query ALTO service for recommendations.

- network path-cost map at the granularity of SDN domains (keep in mind that the routing path may not be finalized when ALTO is consulted, as the flow table may not be propagated for the given flows).

- selection or preferences of one or multiple paths among a set of paths at the granularity of SDN domains; selected/preferred paths can have defined priority and/or failover definitions;

## 6. Use Case for Co-existing SDN and ALTO

### 6.1. Use Case for Upward Flow

The upward flow delivers SDN domains' network information by SDN controllers to the ALTO server. Each SDN controller is responsible

for collecting, aggregating, and submitting its own domain's network information to the ALTO server. Due to the possibility of some SDN domain being dedicated to certain customers, we illustrate the upward flow in two use cases.

#### 6.1.1. Unrestrictive Information Exporting

SDN domain controllers have to collect various network information from the networking devices they manage no matter if ALTO exists or not. The reason is that an SDN controller may have to make decisions for allocating resources within its domain, and making such decisions need various network information. Since such information is readily collected and available, an SDN controller could submit such information as is (or after simple processing) to the ALTO server.Take the available link bandwidth as an example (available link bandwidth could be used as a measure of "distance"). An SDN controller could periodically collect the available bandwidth on all links in its domain and submit it to the ALTO server. However, such information should be annotated with the domain information (e.g., domain ID). By submitting such information, later other SDN controllers may request for this domain's available link bandwidth information.

#### 6.1.2. Restrictive Information Exporting

An SDN domain belonging to a carrier may be dedicated to certain customers of that carrier. In this case, the dedicated users of an SDN domain manage not only how resources should be allocated but also what information should be exported.

A carrier may dedicate a set of small data centers (on multiple sites) to its certain customer. These data centers are put under a single SDN domain. The customer can manage the dedicated multi-site, small data centers via the SDN controller. Periodically the SDN controller collects network information from all data centers.

However, different than the former unrestrictive case, the customer may have stringent privacy policies and therefore decide to aggregate the collected information before submitting to the ALTO server.

For instance, the customer may aggregate the information for a data center network in the same site such that the data center network is shrunk into a single node; by doing so, the multi-site data center network is aggregated into a multi-node network topology, each node in the topology actually corresponds to a small data center in reality. The aggregated network topology could be annotated with available link bandwidth information or other information that is collected and allowed to be exported.

The customer's information aggregation policy defines how the information should be pre-processed before exporting to the ALTO server. The main purpose of aggregation is to protect privacy. As a result of information aggregation, the exported network information could be a logical topology (annotated with various network information, e.g., distance or cost) which is totally different from the physical topology.

### 6.1.3. Information Aggregation

Without SDN, ALTO defines cost maps for an aggregated view of the network topology, where the granularity of aggregation is determined by the network carrier and could be either coarse-grain or fine-grain.

However, with the introduction of SDN, such information aggregation could be greatly simplified and should be policed based on the policies defined for each SDN domain. For instance, ALTO only needs to collect information from a pre-defined set of SDN domain controllers, where the controllers determines at what granularity they would like to aggregate the information and export them. In addition, such aggregation is governed by the domain-specific policies, which defines not only the granularity of aggregation but also to whom such aggregated information may be exposed.

More specifically, an illustrative use case is as follows. SDN controllers collect fine-grain information and aggregate it periodically per their policies. ALTO is configured to obtain the aggregated information from a set of SDN domain controllers and obtain possibly raw information from networking devices (or the network operation center). ALTO then constructs a complete view of the overall network (an aggregated view of the network). SDN controllers obtain cost maps from ALTO and apply such maps when making data/forwarding plane decisions.

Another illustrative use case is as follows. SDN controllers may choose to export fine-grain information to ALTO. After it obtains the cost maps from ALTO, it could leverage the cost maps with greater details about their own domains and make informed decisions. However, SDN controllers should not overload ALTO by exporting too much fine-grain information.

## 6.2. Use Case for Downward Flow

We illustrate the use of downward flow through several use cases as follows.

Note that when the originating SDN domain's controller make decisions

for choosing path(s) and set up the path(s), each involved SDN domain controller should map the overall decision to scoped decisions specifically for their responsible domains.

### 6.2.1. SDN-Aware QoS Metrics

We use two use cases to describe SDN-aware QoS. When aggregating QoS information, SDN controllers or the information aggregation policies should understand the semantics of each QoS metrics. For instance, some metrics (e.g., delay) are additive, while some others are multiplicative (e.g., packet loss rate). The information aggregation policy should be flexible enough to specify such details.

An SDN capable application / source end-point may request for a certain amount of end-to-end bandwidth to a destination end-point on the fly. The two end points in question should be in the same administrative domain, but they are not in the same SDN domain. The path(s) set up for such a request span across multiple SDN domains.

The SDN controller of the source domain (i.e., the SDN domain where the source end-point is located), referred to as the source SDN controller, should first obtain the cost maps from the ALTO server. Such cost maps are SDN-domain-specific, namely, the costs are defined for pairs of SDN domains, rather than for pairs of end points as in the legacy ALTO case.

The source SDN domain controller should then determine path(s) for the two end points based on the cost maps and associated information obtained from ALTO. More specifically, the controller should:

- Compute a lowest-cost path at the SDN domain level using the obtained SDN-domain-specific cost map.

- Contact the controllers of those SDN domains on the selected path, probing for the available bandwidth that could be dedicated to the requested session.

- Check if all of the selected path have sufficient combined bandwidth that matches the required bandwidth

- if the combined bandwidth of all selected paths cannot match the requirement, then go back to step 1 and select another lowest-cost path (different than the already selected ones)

    * if no path can be selected and the combined bandwidth does not match the requirement, the request cannot be satisfied.

- if the combined bandwidth of all selected paths match the requirement, then set up all selected paths by signaling all involved SDN domain controllers. Note that the signaling protocol and how to set up paths are beyond the scope of this document.

Data backup and migration among data centers, which typically require bulk data transfers, is an example of on-demand bandwidth use case. Data centers may be managed by one or multiple SDN domains; thus bulk data transfer could be thought of as bulk data transfer among multiple SDN domains.

Similar to the preceding use case, applications may request for paths satisfying some certain QoS metrics, e.g., VoIP applications may ask for paths with delay being lower than certain thresholds. This requires that ALTO cost maps embed such information, and that SDN controller should export such information to ALTO.

### 6.2.2. Content Delivery Networks (CDN)

Content Delivery Network (CDN) has been widely deployed to help dissemination of content at the Internet scale. Network carries are also deploying CDNs inside their own networks to improve the user experiences of their customers. With the introduction of SDN, not only legacy CDN but also a new SDN-based CDN can be seamlessly implemented and integrated with the current network infrastructure.

Here is an example of the flow of SDN-enabled CDN. Suppose that there are a set of CDN servers deployed in a carrier's network and they are willing to be managed by SDN. An equivalent class for each of the CDN server is defined by either the CDN carrier or the network carrier (these two carriers can be the same). An equivalent class is a set of IP addresses, one for a CDN server, where if one can be used to fulfill requests for a specific content, then any server in this class can also be used to serve the same requests. In the extreme case, there is only one equivalent class for all CDN servers.

Then the pre-defined equivalent classes are pushed to the SDN controllers, which leverage such information to select CDN servers and set up paths for any end point to any such servers.

- A network client (e.g., an HTTP-based Web client) obtains the IP address, referred to as A, of one of the CDN servers in the carrier's network (e.g., by DNS queries)

- The client sends a first packet destined for A (for HTTP requests, this packet is a TCP SYN packet)

- An SDN capable networking device picks up the packet
- If there are forwarding rules already set up for the communication between the requesting client and the destination A, then follow the rules to forward the request packet
- Otherwise, forward the request packet to the SDN controller of this domain
- Once receiving a forwarded packet from a networking device, the SDN controller takes the following actions:
  - Retrieves the equivalent class for the given destination A
  - Obtains a cost map from the ALTO server (this step could take place asynchronously)
  - Ranks all CDN servers in the equivalent class according to the cost map obtained from the ALTO server
  - Selects the best CDN server, referred to as B, based on the above ranking
  - Negotiates and sets up a best-effort path to the selected CDN server with other controllers
  - Sets up forwarding rules for the path, and rewriting rules for replacing the IP address of A with the IP address of B (so that the client is actually communicating with B, although it may think that it is communicating with A; however, which server it communicates is not important)
- The request packet is forwarded to the chosen CDN server B, subject to the forwarding rules and rewriting rules
- The client communicates with the CDN server B
- The path and associated forwarding/rewriting rules are expired whe n the communication is torn down (this step is irrelevant to the ALTO extension for SDN, therefore, it is out of scope)

However, the above use case has two limitations. First, it violates the TCP semantics; namely, the client intends to and believes that it is communicating with server A, but actually it is communicating with server B. Second, it has to rely on the capability of devices being able to rewrite forwarding rules (e.g., use one IP address to replace another one in a packet).

If the above two limitations become concerns, e.g., either TCP semantics should not be violated or rewriting is not available or both, the above SDN-enabled CDN use case can be implemented in similar way, with the help of a redirection server.

Below we describe the steps that are different:

- A redirection server (or server farm), referred to as R, is set up for redirecting client requests
- Each SDN controller sets up path(s) to the given redirection server R
- Note that the redirection server could be an integral component of an SDN controller (either collocated or integrated), in which path(s) are not necessary
- Once receiving a forwarded packet from a networking device, the SDN controller takes the following actions:
  - Retrieves the equivalent class for the given destination A
  - Obtains a cost map from the ALTO server (this step could take place asynchronously)
  - Ranks all CDN servers in the equivalent class according to the cost map obtained from the ALTO server
  - Selects the best CDN server, referred to as B, based on the above ranking
  - Sends the information of the chosen CDN server, i.e., its IP address B, to the redirection server R
  - Negotiates and sets up a best-effort path to the redirection server R (if R is not integrated with the SDN controller)
  - Sets up forwarding rules for the path to R
  - Negotiates and sets up a best-effort path to the CDN server B
  - Sets up forwarding rules for the path to B
- The client communicates with the redirection server R
- R sends an HTTP redirection packet to the client, redirecting future requests to the CDN server B (which is notified by the SDN controller)

- The client communicates with the chosen CDN server B (note that the path to B has been already set up)

### 6.2.3. Information-Centric Content Delivery Networks (IC-CDN)

Information-Centric Networking (ICN) is a "host-to-information" communication model, different from the legacy "host-to-host" model implemented by the Internet. Content Delivery Network (CDN) is more of a "host-to-information" model (i.e., CDNs can be treated as a special instance of ICN), but implemented in the "host-to-host" model, due to the fact that the current semantics provided by the Internet only support the "host-to-host" model.

With the introduction of SDN, CDNs can be converted into an information-centric networking implementation:

- A CDN client sends a request for a specific content
- The request packet is formatted per the CDN in SDN specification (beyond the scope of this draft), containing
  - the requested content name in the packet
  - destination (a specific anycast IP address) which is reserved for legacy applications to invoke SDN capabilities
  - (optional) QoS requirements (e.g., prefer fast/local servers vs. slow/remote servers, demands are elastic or not)
- An SDN capable networking device picks up the request packet
- If there are forwarding rules set up for this content request already, then follow the rules to forward the request packet, and terminate this.
- Otherwise, forward the request packet to the SDN controller for this domain.
- The SDN controller communicates with the CDN's name directory to look up possible CDN servers that can satisfy the request
- The SDN controller obtains a cost map from the ALTO server
- The SDN controller applies the cost map to select the best CDN server per the QoS requirements if specified in the request
- The SDN controller negotiate the path to the selected CDN server with other controllers

- The SDN controllers that along the chosen path set up the path, and push the forwarding rule(s) for this chosen path to all networking devices that are involved

- The request packet is forwarded to the chosen CDN server

- Data packets flow back to the CDN client

In this use case, the CDN clients could be modified to send the "information-centric" request. However, in a realistic implementation, neither the CDN clients nor the CDN servers have to be significantly modified (e.g., CDN redirection could be leveraged to implement the above work flow).

## 7. Conclusions

In this draft, we identify the fundamental differences between legacy ALTO client/server and ALTO client/server with the existence of SDN. The introduction of SDN fundamentally changes the way that the ALTO works. We present the Vertical Architecture and the Horizontal Architecture to allow coherent coexistence of SDN and ALTO. We believe that the Vertical Architecture allows better division, management, flexibility, privacy control and long-term evolution of the network. Therefore we mainly focus on the Vertical Architecture in this draft. We also define the main interactions and information flows, and present a set of use cases to illustrate how we extend ALTO to support SDN, in the Vertical Architecture.

## 8. Contributors

The authors would like to thank Vijay K. Gurbani for his many detailed reviews and helpful assistance on this draft.

Vijay K. Gurbani
Bell Laboratories, Alcatel-Lucent
1960 Lucent Lane, Rm. 9C-533
Naperville, IL 60566
USA

Email: vkg AT (acm.org,bell-labs.com)

## 9. Acknowlegements

The authors would like to thank Tom Taylor and Aditi Vira for editing the draft.

This memo is based upon work supported in part by the National Science Foundation of China (NSFC) under Grant No. 61073192 and the China 973 Program under Grant No. 2011CB302905. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF.

## 10. Security Considerations

TBD.

## 11. IANA Considerations

This document makes no specific request of IANA.

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

## 12. Informative References

[1] Koponen, T., Casado, M., Gude, N., Stribling, J., Poutievski, L., Zhu, M., Ramanathan, R., Iwata, Y., Inoue, H., Hama, T., and S. Shenker, "Onix: A Distributed Control Platform for Large-scale Production Networks", Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI 10), Vancouver, Canada, pp. 351-364", October 2010.

[2] Gurbani, V., Scharf, M., Lakshman, T., and V. Hilt, "Abstracting network state in Software Defined Networks (SDN) for rendezvous services, IEEE International Conference on Communications (ICC) Workshop on Software Defined Networks (SDN)", June 2012.

Authors' Addresses

Haiyong Xie
Huawei & USTC
2330 Central Expy
Santa Clara, CA 95050
USA

Phone:
Email: Haiyong.xie@huawei.com

Tina Tsou
Huawei (USA)
2330 Central Expy
Santa Clara, CA 95050
USA

Phone:
Email: Tina.Tsou.Zouting@huawei.com

Diego R. Lopez
Telefonica I+D
Don Ramon de la Cruz, 84
Madrid, 28006
Spain

Phone:
Email: diego@tid.es

Hongtao Yin
Huawei (USA)
2330 Central Expy
Santa Clara, CA 95050
USA

Phone:
Email: Hongtao.yin@huawei.com