Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: April 5, 2013

S. Amante
Level 3 Communications, Inc.
J. Medved
Cisco Systems, Inc.
T. Nadeau
Juniper Networks

October 2, 2012

Topology API Use Cases
draft-amante-irs-topology-use-cases-00

Abstract

This document describes use cases for gathering routing, forwarding and policy information, (hereafter referred to as topology information), about the network and reflecting changes to the topology back into the network and related systems. It describes several applications that need to view or change the topology of the underlying physical or logical network. This document further demonstrates a need for a "Topology Manager" and related functions that collects topology data from network elements and other data sources, coalesces the collected data into a coherent view of the overall network topology, and normalizes the network topology view for use by clients -- namely, applications that consume or want to change topology 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."

This Internet-Draft will expire on April 5, 2013.

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 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

In today's networks, a variety of applications, such as Traffic Engineering, Capacity Planning, Security Auditing or Services Provisioning (for example, Virtual Private Networks), have a common need to acquire and consume network topology information. Unfortunately, all of these applications are (typically) vertically integrated: each uses its own proprietary normalized view of the network and proprietary data collectors, interpreters and adapters, which speak a variety of protocols, (SNMP, CLI, SQL, etc.) directly to network elements and to back-office systems. While some of the topological information can be distributed using routing protocols, unfortunately it is not desirable for some of these applications to understand or participate in routing protocols.

This approach is incredibly inefficient for several reasons. First, developers must write duplicate 'network discovery' functions, which then become challenging to maintain over time, particularly as/when new equipment are first introduced to the network. Second, since there is no common "vocabulary" to describe various components in the network, such as physical links, logical links, or IP prefixes, each application has its own data model. To solve this, some solutions have distributed this information in the normalized form of routing distribution. However, this information still does not contain "inactive" topological information, thus not containing information considered to be part of a network's inventory.

These limitations lead to applications being unable to easily exchange information with each other. For example, applications cannot share changes with each other that are (to be) applied to the physical and/or logical network, such as installation of new physical links, or deployment of security ACL's. Each application must frequently poll network elements and other data sources to ensure that it has a consistent representation of the network so that it can carry out its particular domain-specific tasks. In other cases, applications that cannot speak routing protocols must use proprietary CLI or other management interfaces which represent the topological information in non-standard formats or worse, semantic models.

Overall, the software architecture described above at best results in incredibly inefficient use of both software developer resources and network resources, and at worst, it results in some applications simply not having access to this information.

Figure 1 is an illustration of how individual applications collect data from the underlying network. Applications retrieve inventory, network topology, state and statistics information by communicating directly with underlying Network Elements as well as with

intermediary proxies of the information. In addition, applications transmit changes required of a Network Element's configuration and/or state directly to individual Network Elements, (most commonly using CLI or Netconf). It is important to note that the "data models" or semantics of this information contained within Network Elements are largely proprietary with respect to most configuration and state information, hence why a proprietary CLI is often the only choice to reflect changes in a NE's configuration or state. This remains the case even when standards-based mechanisms such as Netconf are used which provide a standard syntax model, but still often lack due to the proprietary semantics associated with the internal representation of the information.

```
                               +----------------+
                            +-----------------+ |
                            |  Applications   |-+
                            +-----------------+
                                  ^  ^  ^
              SQL, RPC, ReST      #  |  *    SQL, RPC, ReST ...
           ########################  |  ************************
           #                         |                         *
     +------------+                  |                   +------------+
     | Statistics |                  |                   | Inventory  |
     | Collection |                  |                   | Collection |
     +------------+                  |                   +------------+
           ^                         | NETCONF, SNMP,          ^
           |                         | CLI, TL1, ...           |
           +-------------------------+-------------------------+
           |                         |                         |
           V                         V                         V
   +----------------+        +----------------+       +----------------+
   | Network Element|        | Network Element|       | Network Element|
   | +------------+ |<-LLDP->| +------------+ |<-LMP->| +------------+ |
   | | Data Model | |        | | Data Model | |       | | Data Model | |
   | +------------+ |        | +------------+ |       | +------------+ |
   +----------------+        +----------------+       +----------------+
```

Figure 1: Applications getting topology data

Figure 1 shows how current management interfaces such as NETCONF, SNMP, CLI, etc. are used to transmit or receive information to/from various Network Elements. The figure also shows that protocols such as LLDP and LMP participate in topology discovery, specifically to discover adjacent network elements.

The following sections describe the "Statistics Collection" and "Inventory Collection" functions.

1.1. Statistics Collection

In Figure 1, "Statistics Collection" is a dedicated infrastructure that collects statistics from Network Elements. It periodically polls Network Elements (for example, every 5-minutes) for octets transferred per interface, per LSP, etc. Collected statistics are stored and collated, (for example, to provide hourly, daily, weekly 95th-percentile figures), within the statistics data warehouse. Applications typically query the statistics data warehouse rather than poll Network Elements directly to get the appropriate set of link utilization figures for their analysis.

1.2. Inventory Collection

"Inventory Collection" is a network function responsible for collecting network element component and state (i.e.: interface up/down, SFP/XFP optics inserted into physical port, etc.) information directly from network elements, as well as storing inventory information about physical network assets that are not retrievable from network elements, (hereafter referred to as a inventory asset database). Inventory Collection from network elements commonly use SNMP and CLI to acquire inventory information. The information housed in the Inventory Manager is retrieved by applications using a variety of protocols: SQL, RPC, etc. Inventory information, retrieved from Network Elements, is updated in the Inventory Collection system on a periodic basis to reflect changes in the physical and/or logical network assets. The polling interval to retrieve updated information is varied depending on scaling constraints of the Inventory Collection systems and expected intervals at which changes to the physical and/or logical assets are expected to occur.

Examples of changes in network inventory that need be learned by the Inventory Collection function are as follows:

- o Discovery of new Network Elements. These elements may or may not be actively used in the network (i.e.: provisioned but not yet activated).

- o Insertion or removal of line cards or other modules, i.e.: optics modules, during service or equipment provisioning.

- o Changes made to a specific Network Element through a management interface by a field technician.

- o Indication of an NE's physical location and associated cable run list, at the time of installation.

o Insertion of removal of cables that result in dynamic discovery of a new or lost adjacent neighbor, etc.

1.3. 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. Terminology

The following briefly defines some of the terminology used within this document.

Inventory Manager: Describes a function of collecting network element inventory and state information directly from network elements, and potentially associated offline inventory databases, via standards-based data models. Components contained in this super set might be visible or invisible to a specific network layer, i.e.: a physical link is visible within the IGP, however the Layer-2 switch through which the physical link traverses is unknown to the Layer-3 IGP. .

Policy Manager: Describes a function of attaching metadata to network components/attributes. Such metadata is likely to include security, routing, L2 VLAN ID, IP numbering, etc. policies that enable the Topology Manager to: a) assemble a normalized view of the network for clients to access; b) allow clients (or, upper-layer applications) read-only vs. read-write access to various network layers and/or network components, etc. The Policy Manager function may be a sub-component of the Topology Manager or it may be a standalone. This will be determined as the work with IRS evolves.

Topology Manager: Network components (inventory, etc.) are retrieved from the Inventory Manager and synthesized with information from the Policy Manager into cohesive, normalized views of network layers. The Topology Manager exposes normalized views of the network via standards-based data models to Clients, or higher-layer applications, to act upon in a read-only and/or read-write fashion. The Topology Manager may also push information back into the Inventory Manager and/or Network Elements to execute changes to the network's behavior, configuration or state.

Orchestration Manager: Describes a function of stitching together resources (i.e.: compute, storage) and/or services with the network or vice-versa. The Orchestration Manager relies on the capabilities provided by the other "Managers" listed above in order to realize a complete service.

Normalized Topology Data Model: A data model that is constructed and represented using an open, standards-based model that is consistent between implementations.

Data Model Abstraction: The notion that one is able to represent the same set of elements in a data model at different levels of "focus" in order to limit the amount of information exchanged in order to convey this information.

Multi-Layer Topology: Topology is commonly referred to using the OSI protocol layering model. For example, Layer 3 represents routed topologies that typically use IPv4 or IPv6 addresses. It is envisioned that, eventually, multiple layers of the network may be represented in a single, normalized view of the network to certain applications, (i.e.: Capacity Planning, Traffic Engineering, etc.)

Network Element (NE): refers to a network device that typically is addressable (but not always), and hosts. It is sometimes referred to as Nodes.

Links: Every NE contains at least 1 link. These are used to connect the NE to other NEs in the network. Links may be in a variety of states including up, down, administratively down, internally testing, or dormant. Links are often synonymous with network ports on NEs.

3. Orchestration, Collection & Presentation Framework

3.1. Overview

Section 1 demonstrates the need for a network function that would provide a common, standard-based topology view to applications. Such topology collection/management/presentation function would be a part wider framework, that would also include policy management and orchestration. The framework is shown in Figure 2.

```
                               +---------------+
                            +----------------+ |
                            |  Applications  |-+
                            +----------------+
      Websockets, ReST, XMPP, ...   ^    Websockets, ReST, XMPP, ...
          +-------------------------+-------------------------+
          |                         |                         |
     +------------+     +-------------------------+     +-------------+
     |   Policy   |<----|    Topology Manager     |---->|Orchestration|
     |   Manager  |     | +---------------------+ |     |   Manager   |
     +------------+     | | Topology Data Model | |     +-------------+
                        | +---------------------+ |
                        +-------------------------+
                                  ^  ^  ^
       Websockets, ReST, XMPP     #  |  *    Websockets, ReST, XMPP
           ########################  |  ************************
           #                         |                       *
     +------------+                  |                 +------------+
     | Statistics |                  |                 | Inventory  |
     | Collection |                  |                 | Collection |
     +------------+                  |                 +------------+
           ^                         | IRS, NETCONF, SNMP,     ^
           |                         | TL1 ...                 |
           +-------------------------+-------------------------+
           |                         |                         |
           V                         V                         V
   +----------------+        +----------------+       +----------------+
   | Network Element|        | Network Element|       | Network Element|
   | +------------+ |<-LLDP->| +------------+ |<-LMP->| +------------+ |
   | | Data Model | |        | | Data Model | |       | | Data Model | |
   | +------------+ |        | +------------+ |       | +------------+ |
   +----------------+        +----------------+       +----------------+
```

Figure 2: Topology Manager

The following sections describe in detail the Topology Manager, Policy Manager and Orchestration Manager functions.

3.2. Topology Manager

The Topology Manager is responsible for retrieving topological information from the network via a variety of sources. The first most obvious source is the "live" IGP or an equivalent mechanism. "Live" IGP provides information about links that are components of the active topology, in other words links that are present in the Link State Database (LSDB) and are eligible for forwarding. The second source of topology information is the Inventory Collection system, which provides information for network components not visible

within the IGP's LSDB, (i.e.: links or nodes, or properties of those links or nodes, at lower layers of the network).

The Topology Manager would synthesize retrieved information into cohesive, abstracted views of the network using a standards-based, normalized topology data model. The Topology Manager can then expose these data models to Clients, or higher-layer applications using a northbound interface, which would be a protocol/API commonly used by higher-layer applications to retrieve and update information. Examples of such protocols are ReST, Websockets, or XMPP. Topology Manager's clients would be able to act upon the information in a read-only and/or read-write fashion, (based on policies defined within the Policy Manager).

Clients may request changes to the network topology by publishing changes within data models and sending those to the Topology Manager. The Topology Manager internally validates the requested changes against various constraints and, if the changes are permitted, the Topology Manager updates associated Managers (Policy or Inventory Managers), communicates those changes to the individual network elements and, finally, verifies that those configurations were properly received and executed by the network elements.

It is envisioned that the Topology Manager will ultimately contain topology information for multiple layers of the network: Transport, Ethernet and IP/MPLS as well as multiple (IGP) areas and/or multiple Autonomous Systems (ASes). This allows the Topology Manager to stitch together a holistic view of several layers of the network, which is an important requirement, particularly for upper-layer Traffic Engineering, Capacity Planning and Provisioning Clients (applications) used to design, augment and optimize IP/MPLS networks that require knowledge of underlying Shared Risk Link Groups (SRLG) within the Transport and/or Ethernet layers of the network.

The Topology Manager must have the ability to discover and communicate with network elements who are not only active and visible within the Link State Database (LSDB) of an active IGP, but also network elements who are active, but invisible to a LSDB (e.g.: L2 Ethernet switches, ROADM's, etc.) that are part of the underlying Transport network. This requirement will influence the choice of protocols needed by the Topology Manager to communicate to/from network elements at the various network layers.

It is also important to recognize that the Topology Manager will be gleaning not only (relatively) static inventory information from the Inventory Manager, i.e.: what linecards, interface types, etc. are actively inserted into network elements, but also dynamic inventory information, as well. With respect to the latter, network elements

are expected to rely on various Link Layer Discovery Protocols (i.e.: LLDP, LMP, etc.) that will aid in automatically identifying an adjacent node, port, etc. at the far-side of a link. This information is then pushed to or pulled by the Topology Manager in order for it to have an accurate representation of the physical topology of the network.

3.3. Policy Manager

The Policy Manager is the function used to enforce and program policies applicable to network component/attribute data. Policy enforcement is a network-wide function that can be consumed by various network elements and services including the Inventory Manager, Topology Manager or other network elements. Such policies are likely to encompass the following.

- Logical Identifier Numbering Policies
    * Correlation of IP prefix to link based on type of link (P-P, P-PE, PE-CE, etc.)
    * Correlation of IP Prefix to IGP Area
    * Layer-2 VLAN ID assignments, etc.
- Routing Configuration Policies
    * OSPF Area or IS-IS Net-ID to Node (Type) Correlation
    * BGP routing policies, i.e.: nodes designated for injection of aggregate routes, max-prefix policies, AFI/SAFI to node correlation, etc.
- Security Policies
    * Access Control Lists
    * Rate-limiting
- Network Component/Attribute Data Access Policies. Client's (upper-layer application) read-only or read-write access to Network Components/Attributes contained in the "Inventory Manager" as well as Policies contained within the "Policy Manager" itself.

The Policy Manager function may be a sub-component of the Topology or Orchestration Manager or it may be a standalone. This will be determined as the work with IRS evolves.

3.4. **Orchestration Manager**

The Orchestration Manager provides the ability to stitch together resources (i.e.: compute, storage) and/or services with the network or vice-versa. Examples of 'generic' services may include the following:

- Application-specific Load Balancing

- Application-specific Network (Bandwidth) Optimization

- Application or End-User specific Class-of-Service

- Application or End-User specific Network Access Control

The above services could then enable coupling of resources with the network to realize the following:

- Network Optimization: Creation and Migration of Virtual Machines (VM's) so they are adjacent to storage in the same DataCenter.

- Network Access Control: Coupling of available (generic) compute nodes within the appropriate point of the data-path to perform firewall, NAT, etc. functions on data traffic.

The Orchestration Manager is expected to exchange data models with the Topology Manager, Policy Manager and Inventory Manager functions. In addition, the Orchestration Manager is expected to support publish and subscribe capabilities to those functions, as well as to Clients, to enable scalability with respect to event notifications.

The Orchestration Manager may receive requests from Clients (applications) for immediate access to specific network resources. However, Clients may request to schedule future appointments to reserve appropriate network resources when, for example, a special event is scheduled to start and end.

Finally, the Orchestration Manager should have the flexibility to determine what network layer(s) may be able to satisfy a given Client's request, based on constraints received from the Client as well as those constraints learned from the Policy and/or Topology Manager functions. This could allow the Orchestration Manager to, for example, satisfy a given service request for a given Client using the optical network (via OTN service) if there is insufficient IP/MPLS capacity at the specific moment the Client's request is received.

The operational model is shown in the following figure.

TBD.

Figure 3: Overall Reference Model

4. Use Cases

4.1. Virtualized Views of the Network

4.1.1. Capacity Planning and Traffic Engineering

When performing Traffic Engineering and/or Capacity Planning of an IP/MPLS network, it is important to account for SRLG's that exist within the underlying physical, optical and Ethernet networks. Currently, it's quite common to create and/or take "snapshots", at infrequent intervals, that comprise the inventory data of the underlying physical and optical layer networks. This inventory data then needs to be massaged or normalized to conform to the data import requirements of sometimes separate Traffic Engineering and/or Capacity Planning tools. This process is error-prone and inefficient, particularly as the underlying network inventory information changes due to introduction of, for example, new network element makes or models, linecards, capabilities, etc. at the optical and/or Ethernet layers of the underlying network.

This is inefficient with respect to the time and expense consumed by software developer, Capacity Planning and Traffic Engineering resources to normalize and sanity check underlying network inventory information, before it can be consumed by IP/MPLS Capacity Planning and Traffic Engineering applications. Due to this inefficiency, the underlying physical network inventory information, (containing SRLG and corresponding critical network asset information), used by the IP/MPLS Capacity Planning and TE applications is not updated frequently, thus exposing the network to, at minimum, inefficient utilization and, at worst, critical impairments.

An Inventory Manager function is required that will, first, extract inventory information from network elements -- and potentially associated offline inventory databases to acquire physical cross-connects and other information that is not available directly from network elements -- at the physical, optical, Ethernet and IP/MPLS layers of the network via standards-based data models. Data models and associated vocabulary will be required to represent not only components inside or directly connected to network elements, but also to represent components of a physical layer path (i.e.: cross-connect panels, etc.) The aforementioned inventory will comprise the complete set of inactive and active network components.

A Statistics Collection Function is also required. As stated above, it will collect utilization statistics from Network Elements, archive and aggregate them in a statistics data warehouse. Summaries of these figures then need to be exposed in normalized data models to the Topology Manager so it can easily acquire historical link and LSP utilization figures that can be used to, for example, build trended utilization models to forecast expected changes to the physical and/or logical network components to accommodate network growth.

The Topology Manager function may then augment the Inventory Manager information by communicating directly with Network Elements to reveal the IGP-based view of the active topology of the network. This will allow the Topology Manager to include dynamic information from the IGP, such as Available Bandwidth, Reserved Bandwidth, etc. Traffic Engineering (TE) attributes associated with links, contained with the Traffic Engineering Database (TED) on Network Elements.

It is important to recognize that extracting topology information from the network solely via an IGP, (such as IS-IS TE or OSPF TE), is inadequate for this use case. First, IGP's only expose the active components (e.g. vertices of the SPF tree) of the IP network; unfortunately, they are not aware of "hidden" or inactive interfaces within IP/MPLS network elements, (e.g.: unused linecards or unused ports), or components that reside at a lower layer than IP/MPLS, e.g. Ethernet switches, Optical transport systems, etc. This occurs frequently during the course of maintenance, augment and optimization activities on the network. Second, IGP's only convey SRLG information that have been first applied within the router's configurations, either manually or programatically. As mentioned previously, this SRLG information in the IP/MPLS network is subject to being infrequently updated and, as a result, may inadequately account for critical, underlying network fate sharing properties that are necessary to properly design resilient circuits and/or paths through the network.

In this use case, the Inventory Manager will need to be capable of using a variety of existing protocols such as: NETCONF, CLI, SNMP, TL1, etc. depending on the capabilities of the network elements. The Topology Manager will need to be capable of communicating via an IGP from a (set of) Network Elements. It is important to consider that to acquire topology information from Network Elements will require read-only access to the IGP. However, the end result of the computations performed by the Capacity Planning Client may require changes to various IGP attributes, (e.g.: IGP metrics, TE link-colors, etc.) These may be applied directly by devising a new capability to either: a) inject information into the IGP that overrides the same information injected by the originating Network Element; or, b) allowing the Topology and/or Inventory Manager the

ability to write changes to the Network Element's configuration in order to have it adjust the appropriate IGP attribute(s) and re-flood them throughout the IGP. It would be desirable to have a single mechanism (data model or protocol) that allows the Topology Manager to read and write IGP attributes.

Once the Topology Manager function has assembled a normalized view of the topology and synthesized associated metadata with each component of the topology (link type, link properties, statistics, intra-layer relationships, etc.), it can then expose this information via its northbound API to Clients. In this use case that means Capacity Planning and Traffic Engineering applications, which are not required to know innate details of individual network elements, but do require generalized information about the node and links that comprise the network, e.g.: links used to interconnect nodes, SRLG information (from the underlying network), utilization rates of each link over some period of time, etc. In this case, it is important that any Client that understands both the web services API and the normalized data model can communicate with the Topology Manager in order to understand the network topology information that was provided by network elements from potentially different vendors, all of which likely represent that topology information internally using different models. If the Client had gone directly to the network elements themselves, it would have to translate and then normalize these different representations for itself. However, in this case, the Topology Manager has done that for it.

When this information is consumed by the Traffic Engineering application, it may run a variety of CSPF algorithms the result of which is likely a list of RSVP LSP's that need to be (re-)established, or torn down, in the network to globally optimize the packing efficiency of physical links throughout the network. The end result of the Traffic Engineering application is "pushing" out to the Topology Manager, via a standard data model to be defined here, a list of RSVP LSP's and their associated characteristics, (i.e.: head and tail-end LSR's, bandwidth, priority, preemption, etc.). The Topology Manager then would consume this information and carry out those instructions by speaking directly to network elements, perhaps via PCEP Extensions for Stateful PCE [I-D.ietf-pce-stateful-pce], which in turn initiates RSVP signaling through the network to establish the LSP's.

After this information is consumed by the Capacity Planning application, it may run a variety of algorithms the result of which is a list of new inventory that is required to be purchased (or, redeployed) as well as associated work orders for field technicians to augment the network for expected growth. It would be ideal if this information was also "pushed" back into the Topology and, in

turn, Inventory Manager as "inactive" links and/or nodes, so that as new equipment is installed it can be automatically correlated with original design and work order packages associated with that augment.

4.1.2. Services Provisioning

Beyond Capacity Planning and Traffic Engineering applications, having a normalized view of just the IP/MPLS layer of the network is still very important for other mission critical applications such as Security Auditing and IP/MPLS Services Provisioning, (e.g.: L2VPN, L3VPN, etc.). With respect to the latter, these types of applications should not need a detailed understanding of, for example, SRLG information, assuming that the underlying MPLS Tunnel LSP's are known to account for the resiliency requirements of all services that ride over them. Nonetheless, for both types of applications it is critical that they have a common and up-to-date normalized view of the IP/MPLS network in order to easily instantiate new services at the appropriate places in the network, in the case of VPN services, or validate that ACL's are configured properly to protect associated routing, signaling and management protocols on the network, with respect to Security Auditing.

For this use case, what is most commonly needed by a VPN Service Provisioning application is as follows. First, Service PE's need to be identified in all markets/cities where the customer has identified they want service. Next, does their exist one, or more, Servies PE's in each city with connectivity to the access network(s), e.g.: SONET/ TDM, used to deliver the PE-CE tail circuits to the Service's PE. Finally, does the Services PE have available capacity on both the PE-CE access interface and its uplinks to terminate the tail circuit? If this were to be generalized, this would be considered an Resource Selection function. Namely, the VPN Provisioning application would iteratively query the Topology Manager to narrow down the scope of resources to the set of Services PE's with the appropriate uplink bandwidth and access circuit capability plus capacity to realize the requested VPN service. Once the VPN Provisioning application has a candidate list of resources it then requests the Topology Manager to go about configuring the Services PE's and associated access circuits to realize the customer's VPN service.

4.1.3. Rapid IP Renumbering, AS Migration

A variety of reasons exist for the "rapid renumbering" of IPv4/IPv6 prefixes and ASN's in an IP/MPLS network. Perhaps the most common reason is as a result of mergers, acquisitions or divestitures of companies, organizations or divisions.

Inside the network of an Enterprise or Service Provider, there

already exist protocols such as DHCP or SLAAC to support rapid renumbering of hosts, (i.e.: servers, laptops, tablets, etc.). These are outside the scope of this document. However, there still exists a critical need to quickly renumber network infrastructure, namely: router interfaces, management interfaces, etc. in order to: a) avoid overlapping RFC 1918 addresses in previously separate domains; b) allow for (better) aggregation of IP prefixes within areas/domains of an IGP; c) allow for more efficient utilization of globally unique IPv4 addresses, which are in limited supply; d) realize business synergies of combining two different AS'es into one, etc.

The set of IPv4 and IPv6 prefixes that have been configured on point-to-point, LAN, Loopback, Tunnel, Management, PE-CE and other interfaces would be gathered from all network elements by the Inventory Manager function. Similarly, the set of ASN's that have been configured on individual NE's, as the global BGP Autonomous System Number, and the PE-CE interfaces is also acquired from the Inventory Manager. Afterward, an "inventory" report of the total number, based on type, of IPv4/IPv6 prefixes could be quickly assembled to understand how much address space is required to accommodate the existing network, but also future growth plans. Next, a new IP prefix and ASN would be assigned to the overall network. An operator may then decide to manually carve up the IP prefix into sub-prefixes that are assigned to various functions or interface types in the network, i.e.: all Loopback interface addresses are assigned from a specific GUA IPv4/IPv6 prefix. Other rules may be crafted by the operator so that, for example, GUA IPv4/IPv6 prefixes for interfaces within each IGP area are assigned out of contiguous address space so that they may be (easily) summarized within the IGP configuration. Finally, the set of ASN's, IP prefixes, rules and/or policies governing how their are to be assigned are encoded in a data model/schema an sent to a Topology Manager (TM). The Topology Manager is then responsible for communicating changes to the Inventory Manager and/or Network Elements in a proper sequence, or order of operations, so as to not lose network connectivity from the Topology Manager to the network elements.

This function could be extended further whereby the Orchestration Manager would be used in order to automatically create a list of IP addresses and their associated DNS names, which would then be "pushed" to Authoritative DNS servers so that interface names would get updated in DNS automatically. In addition, the Orchestration Manager function could notify a "Infrastructure Security" application that IP prefixes on the network has changed so that it then updates ACL's used to, for example, protect IP/MPLS routing and signaling protocols used on the network.

4.1.4. Troubleshooting & Monitoring

Once the Topology Manager has a normalized view of several layers of the network, it's then possible to more easily expose a richer set of data to network operators when performing diagnosis, troubleshooting and repairs on the network. Specifically, there is a need to (rapidly) assemble a current, accurate and comprehensive network diagram of a L2VPN or L3VPN service for a particular customer when either: a) attempting to diagnose a service fault/error; or, b) attempting to augment the customer's existing service. Information that may be assembled into a comprehensive picture could include physical and logical components related specifically to that customer's service, i.e.: VLAN's or channels used by the PE-CE access circuits, CoS policies, historical PE-CE circuit utilization, etc. The Topology Manager would assemble this information, on behalf of each of the network elements and other data sources in and associated with the network, and could present this information in a vendor-independent data model to applications to be displayed allowing the operator (or, potentially, the customer through a SP's Web portal) to visualize the information.

4.2. Path Computation Element (PCE)

As described in [RFC4655] a PCE can be used to compute MPLS-TE paths within a "domain" (such as an IGP area) or across multiple domains (such as a multi-area AS, or multiple ASes).

o Within a single area, the PCE offers enhanced computational power that may not be available on individual routers, sophisticated policy control and algorithms, and coordination of computation across the whole area.

o If a router wants to compute a MPLS-TE path across IGP areas its own TED lacks visibility of the complete topology. That means that the router cannot determine the end-to-end path, and cannot even select the right exit router (Area Border Router - ABR) for an optimal path. This is an issue for large-scale networks that need to segment their core networks into distinct areas, but which still want to take advantage of MPLS-TE.

The PCE presents a computation server that may have visibility into more than one IGP area or AS, or may cooperate with other PCEs to perform distributed path computation. The PCE needs access to the topology and the Traffic Engineering Database (TED) for the area(s) it serves, but [RFC4655] does not describe how this is achieved. Many implementations make the PCE a passive participant in the IGP so that it can learn the latest state of the network, but this may be sub-optimal when the network is subject to a high degree of churn, or

when the PCE is responsible for multiple areas.

The following figure shows how a PCE can get its TED information using a Topology Server.

```
                  +----------+
                  |  -----   | TED synchronization via Topology API
                  | | TED |<-+----------------------------------+
                  |  -----   |                                  |
                  |    |     |                                  |
                  |    |     |                                  |
                  |    v     |                                  |
                  |  -----   |                                  |
                  | | PCE |  |                                  |
                  |  -----   |                                  |
                  +----------+                                  |
                       ^                                        |
                       | Request/                               |
                       | Response                               |
                       v                                        |
         Service  +----------+   Signaling  +----------+   +----------+
         Request  | Head-End |   Protocol   | Adjacent |   | Topology |
         -------->|  Node    |<------------>|   Node   |   | Manager  |
                  +----------+              +----------+   +----------+
```

Figure 4: Topology use case: Path Computation Element

4.3. ALTO Server

An ALTO Server [RFC5693] is an entity that generates an abstracted network topology and provides it to network-aware applications over a web service based API. Example applications are p2p clients or trackers, or CDNs. The abstracted network topology comes in the form of two maps: a Network Map that specifies allocation of prefixes to PIDs, and a Cost Map that specifies the cost between PIDs listed in the Network Map. For more details, see [I-D.ietf-alto-protocol].

ALTO abstract network topologies can be auto-generated from the physical topology of the underlying network. The generation would typically be based on policies and rules set by the operator. Both prefix and TE data are required: prefix data is required to generate ALTO Network Maps, TE (topology) data is required to generate ALTO Cost Maps. Prefix data is carried and originated in BGP, TE data is originated and carried in an IGP. The mechanism defined in this document provides a single interface through which an ALTO Server can retrieve all the necessary prefix and network topology data from the underlying network. Note an ALTO Server can use other mechanisms to get network data, for example, peering with multiple IGP and BGP

Speakers.

The following figure shows how an ALTO Server can get network topology information from the underlying network using the Topology API.

```
+--------+
| Client |<--+
+--------+   |
             |    ALTO    +--------+                 +----------+
+--------+   |  Protocol  |  ALTO  | Network Topology | Topology |
| Client |<--+------------| Server |<-----------------| Manager  |
+--------+   |            |        |                 |          |
             |            +--------+                 +----------+
+--------+   |
| Client |<--+
+--------+
```

Figure 5: Topology use case: ALTO Server

5. Acknowledgements

The authors wish to thank Alia Atlas, Dave Ward, Hannes Gredler, Stafano Previdi for their valuable contributions and feedback to this draft.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

At the moment, the Use Cases covered in this document apply specifically to a single Service Provider or Enterprise network. Therefore, network administrations should take appropriate precautions to ensure appropriate access controls exist so that only internal applications and end-users have physical or logical access to the Topology Manager. This should be similar to precautions that are already taken by Network Administrators to secure their existing Network Management, OSS and BSS systems.

As this work evolves, it will be important to determine the appropriate granularity of access controls in terms of what individuals or groups may have read and/or write access to various types of information contained with the Topology Manager. It would

be ideal, if these access control mechanisms were centralized within the Topology Manager itself.

8. References

8.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

8.2. Informative References

[I-D.atlas-irs-problem-statement]
Atlas, A., Nadeau, T., and D. Ward, "Interface to the Routing System Problem Statement", draft-atlas-irs-problem-statement-00 (work in progress), July 2012.

[I-D.ietf-alto-protocol]
Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-13 (work in progress), September 2012.

[I-D.ietf-pce-stateful-pce]
Crabbe, E., Medved, J., Varga, R., and I. Minei, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce-01 (work in progress), July 2012.

[I-D.ward-irs-framework]
Atlas, A., Nadeau, T., and D. Ward, "Interface to the Routing System Framework", draft-ward-irs-framework-00 (work in progress), July 2012.

[RFC4655] Farrel, A., Vasseur, J., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, August 2006.

[RFC5693] Seedorf, J. and E. Burger, "Application-Layer Traffic Optimization (ALTO) Problem Statement", RFC 5693, October 2009.

Authors' Addresses

Shane Amante
Level 3 Communications, Inc.
1025 Eldorado Blvd
Broomfield, CO 80021
USA

Email: shane@level3.net

Jan Medved
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134
USA

Email: jmedved@cisco.com

Thomas D. Nadeau
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
USA

Email: tnadeau@juniper.net