Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: April 23, 2014

J. Medved
S. Previdi
Cisco Systems, Inc.
V. Lopez
Telefonica I+D
S. Amante

October 20, 2013

Topology API Use Cases
draft-amante-i2rs-topology-use-cases-01

Abstract

This document describes use cases for gathering routing, forwarding and policy information, (hereafter referred to as topology information), about the network. It describes several applications that need to view 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 topology information.

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 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.

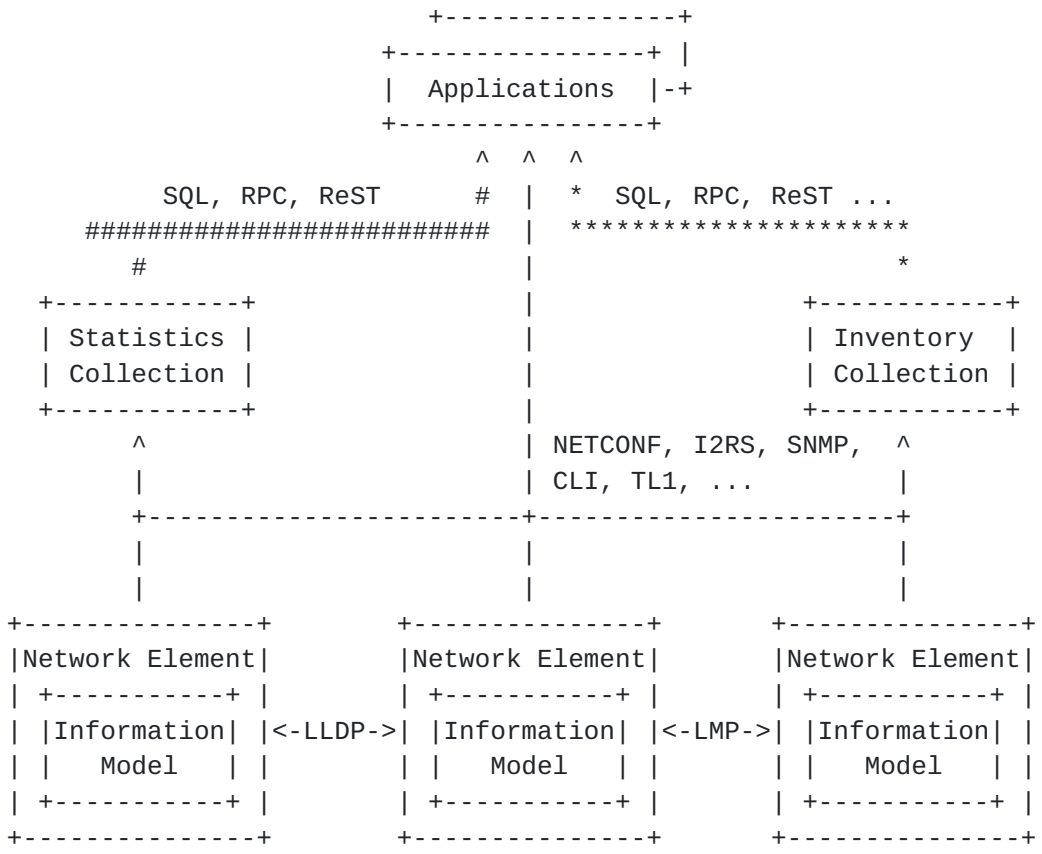

Figure 1: Applications getting topology data

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 (for example, every 5-minutes) polls Network Elements for octets transferred per interface, per LSP, etc. Collected statistics are stored and collated within a 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 component and state information directly from Network Elements, as well as for storing inventory information about physical network assets that are not retrievable from Network Elements. The collected data is hereafter referred to as the 'Inventory Asset Database. Examples of information collected from Network Elements are: interface up/down status, the type of SFP/XFP optics inserted into physical port, etc.

The Inventory Collection function may use SNMP and CLI to acquire inventory information from Network Elements. The information housed in the Inventory Manager is retrieved by applications via a variety of protocols: SQL, RPC, REST etc. Inventory information, retrieved from Network Elements, is periodically updated in the Inventory Collection system 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:

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

- Insertion or removal of line cards or other modules, such as optics modules, during service or equipment provisioning.

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

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

- 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 is a function that collects Network Element inventory and state information directly from Network Elements and from associated offline inventory databases. Inventory information may only be visible at a specific network layer; for example, a physical link is visible within the IGP, but a Layer-2 switch through which the physical link traverses is unknown to the Layer-3 IGP.

Policy Manager is a function that attaches metadata to network components/attributes. Such metadata may include security, routing, L2 VLAN ID, IP numbering, etc. policies, which enable the Topology Manager to:

* Assemble a normalized view of the network for clients (or upper-layer applications

* Allow clients (or upper-layer applications) access to information collected from 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 function.

Topology Manager is a function that collects topological information from a variety of sources in the network and provides a normalized view of the network topology to clients and/or higher-layer applications.

Orchestration Manager is a function that stitches together resources, such as compute or storage, and/or services with the network or vice-versa. To realize a complete service, the Orchestration Manager relies on capabilities provided by the other "Managers" listed above.

Normalized Topology Information Model is an open, standards-based information model of the network topology.

Information Model Abstraction: The notion that one is able to represent the same set of elements in an information 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), or a host. It is sometimes referred to as a 'Node'.

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, such as up, down, administratively down, internally testing, or dormant. Links are often synonymous with network ports on NEs.

3. The Orchestration, Collection & Presentation Framework

3.1. Overview

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

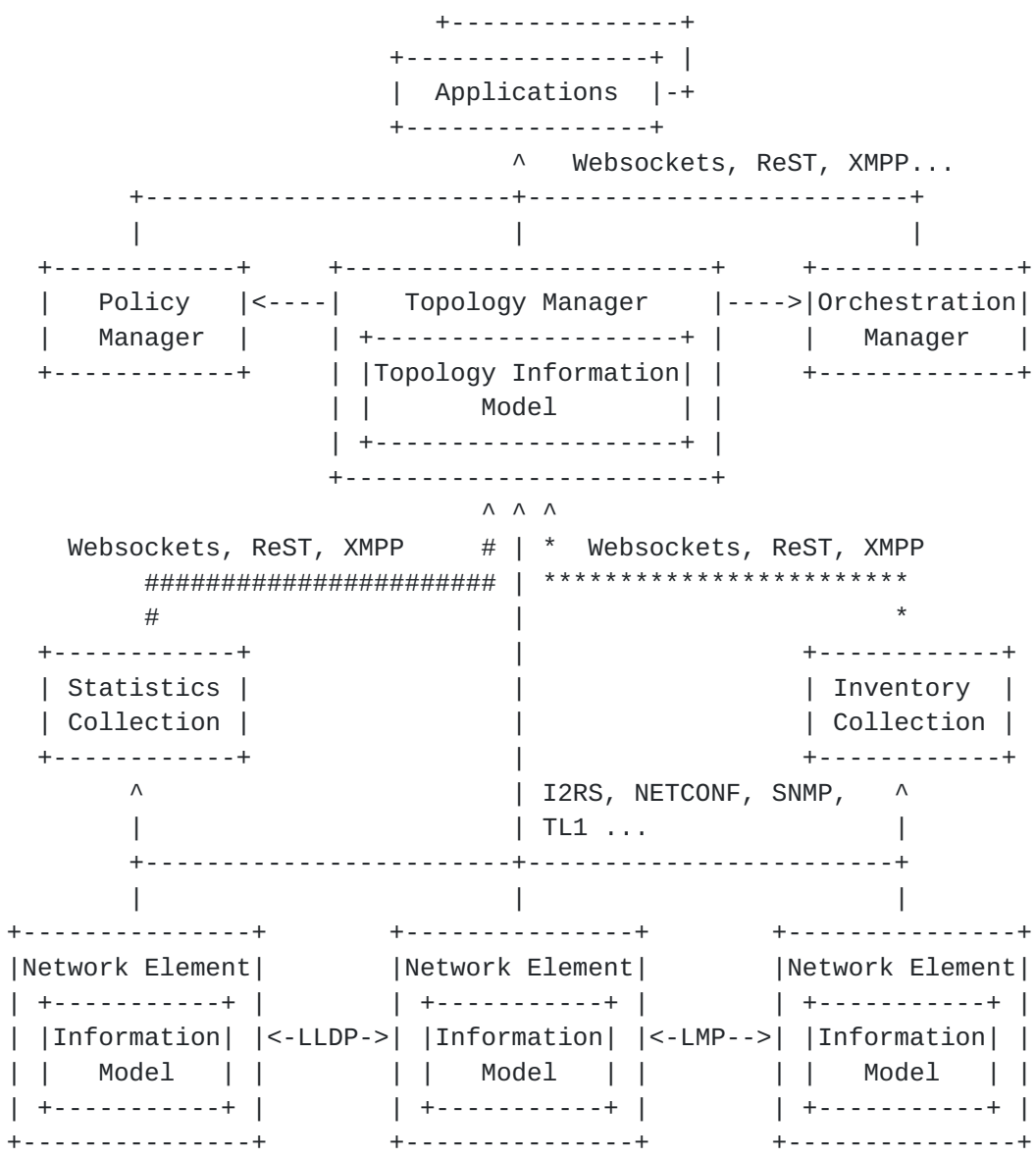

Figure 2: Topology Manager

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

3.2. The Topology Manager

The Topology Manager is a function that collects topological information from a variety of sources in the network and provides a cohesive, abstracted view of the network topology to clients and/or higher-layer applications. The topology view is based on a standards-based, normalized topology information model.

Topology information sources can be:

- The "live" Layer 3 IGP or an equivalent mechanism that provides information about links that are components of the active topology. Active topology links are present in the Link State Database (LSDB) and are eligible for forwarding. Layer 3 IGP information can be obtained by listening to IGP updates flooded through an IGP domain, or from Network Elements.

- The Inventory Collection system that provides information for network components not visible within the Layer 3 IGP's LSDB, (i.e.: links or nodes, or properties of those links or nodes, at lower layers of the network).

- The Statistics Collection system that provides traffic information, such as traffic demands or link utilizations.

The Topology Manager provides topology information to Clients or higher-layer applications via a northbound interface, such as ReST, Websockets, or XMPP.

The Topology Manager will contain topology information for multiple layers of the network: Transport, Ethernet and IP/MPLS, as well as information for multiple Layer 3 IGP areas and multiple Autonomous Systems (ASes). The topology information can be used by higher-level applications, such as Traffic Engineering, Capacity Planning and Provisioning. Such applications are typically used to design, augment and optimize IP/MPLS networks, and require knowledge of underlying Shared Risk Link Groups (SRLG) within the Transport and/or Ethernet layers of the network.

The Topology Manager must be able to discover Network Elements that are not visible in the "live" L3 IGP's Link State Database (LSDB). Such Network Elements can either be inactive, or active but invisible in the L3 LSDB (e.g.: L2 Ethernet switches, ROADM's, or Network Elements that are in an underlying transport network).

In addition static inventory information collected from the Inventory Manager, the Topology Manager will also collect dynamic inventory information. For example, Network Elements utilize various Link Layer Discovery Protocols (i.e.: LLDP, LMP, etc.) to automatically identify adjacent nodes and ports. This information can be pushed to or pulled by the Topology Manager in order to create an accurate representation of the physical topology of the network

3.3. The 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, the Topology Manager and other Network Elements. Such policies are likely to encompass the following:

- Logical Identifier Numbering Policies
  - Correlation of IP prefix to link based on link type, such as P-P, P-PE, or PE-CE.
  - 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, such as nodes designated for injection of aggregate routes, max-prefix policies, or AFI/SAFI to node correlation..
- Security Policies
  - Access Control Lists
  - Rate-limiting
- Network Component/Attribute Data Access Policies. Client's (upper-layer application) 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 either a sub-component of the Topology or Orchestration Manager or a standalone component.

3.4. Orchestration Manager

The Orchestration Manager provides the ability to stitch together resources (such as compute or 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 will exchange information models with the Topology Manager, the Policy Manager and the Inventory Manager. In addition, the Orchestration Manager must support publish and subscribe capabilities to those functions, as well as to Clients.

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 constraints learned from the Policy and Topology Managers. 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

4.1.1.1. Present Mode of Operation

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 take "snapshots" at infrequent intervals that comprise the inventory data of the underlying physical and optical layer networks. This inventory data is then normalized to conform to 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 new network element makes or models, line cards, capabilities, etc..

The present mode of operation is inefficient with respect to Software Development, Capacity Planning and Traffic Engineering resources. Due to this inefficiency, the underlying physical network inventory information (containing SRLG and corresponding critical network assets information) is not updated frequently, thus exposing the network to, at minimum, inefficient utilization and, at worst, critical impairments.

4.1.1.2. Proposed Mode of Operation

First, the Inventory Manager will extract inventory information from network elements and associated inventory databases. Information extracted from inventory databases will include physical cross-connects and other information that is not available directly from network elements. Standards-based information 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 inventory data will comprise the complete set of inactive and active network components.

Second, the Topology Manager will augment the inventory information with topology information obtained from Network Elements and other sources, and provide an IGP-based view of the active topology of the network. The Topology Manager will also include non-topology dynamic information from IGPs, such as Available Bandwidth, Reserved Bandwidth, Traffic Engineering (TE) attributes associated with links, etc.

Finally, the Statistics Collector will collect utilization statistics from Network Elements, and archive and aggregate them in a statistics data warehouse. Selected statistics and other dynamic data may be distributed through IGP routing protocols ([I-D.ietf-isis-te-metric-extensions] and

[I-D.ietf-ospf-te-metric-extensions]) and then collected at the Statistics Collection Function via BGP-LS ([I-D.ietf-idr-ls-distribution]). Statistics summaries then will be exposed in normalized information models to the Topology Manager, which can use them to, for example, build trended utilization models to forecast expected changes to physical and logical network components.

It is important to recognize that extracting topology information from the network solely from Network Elements and IGPs (IS-IS TE or OSPF TE), is inadequate for this use case. First, IGPs only expose the active components (e.g. vertices of the SPF tree) of the IP network, and are not aware of "hidden" or inactive interfaces within IP/MPLS network elements, such as unused line cards or ports. IGPs are also not aware of components that reside at a layer lower than IP /MPLS, such as Ethernet switches, or Optical transport systems. Second, IGP's only convey SRLG information that have been first applied within a 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.

Once the Topology Manager has assembled a normalized view of the topology and metadata associated with each component of the topology, it can expose this information via its northbound API to the Capacity Planning and Traffic Engineering applications. The applications only require generalized information about nodes 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.

Note that any client/application that understands the Topology Manager's northbound API and its topology information model can communicate with the Topology Manager. Note also that topology information may be provided by Network Elements from different vendors, which may use different information models. If a Client wanted to retrieve topology information directly from Network Elements, it would have to translate and normalize these different representations.

A Traffic Engineering application may run a variety of CSPF algorithms that create a list of TE tunnels that globally optimize the packing efficiency of physical links throughout the network. The TE tunnels are then programmed into the network either directly or through a controller. Programming of TE tunnels into the network is outside the scope of this document.

A Capacity Planning application may run a variety of algorithms the result of which is a list of new inventory that is required for purchase or redeployment, as well as associated work orders for field technicians to augment the network for expected growth.

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 to have a common and up-to-date normalized view of the IP/MPLS network to, for example, instantiate new services at optimal locations in the network, or to validate proper ACL configuration to protect associated routing, signaling and management protocols on the network.

A VPN Service Provisioning application must perform the following resource selection operations:

- Identify Service PE's in all markets/cities where the customer has identified they want service

- Identify one or more existing 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.

- Determine that the Services PE have available capacity on both the PE-CE access interface and its uplinks to terminate the tail circuit

The VPN Provisioning application would iteratively query the Topology Manager to narrow down the scope of resources to the set of Services PEs 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 requests programming of the Service PE's and associated access circuits to set up a customer's VPN service into the network. Programming of Service PEs is outside the scope of this document.

4.1.3. Troubleshooting & Monitoring

Once the Topology Manager has a normalized view of several layers of the network, it can expose a rich set of data to network operators who are 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 would 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. Virtual Network Topology Manager (VNTM)

Virtual Network Topology Manager (VNTM) is in charge of managing the Virtual Network Topology (VNT), as defined in [RFC5623]. VNT is defined in [RFC5212] as a set of one or more LSPs in one or more lower-layer networks that provides information for efficient path handling in an upper-layer network.

The maintenance of virtual topology is a complicated task. VNTM have to decide which are the nodes to be interconnected in the lower-layer to fulfill the resource requirements of the upper-layer. This means to create a topology to cope with all demands of the upper layer without wasting resources in the underlying network. Once the decision is made, some actions have to be taken in the network elements of the layers so the new LSPs are provisioned. Moreover, VNT has to release unwanted resources, so they can be available in the lower-layer network for other uses.

VNTM does not have to solve all previous problems in all scenarios. As defined in [RFC5623] in the PCE-VNTM cooperation model, PCEis computing the paths in the higher layer and when there is not enough resources in the VNT, PCE requests to the VNTM for a new path in the VNT. VNTM checks PCE request using internal policies to check whether this request can be take into account or not. VNTM requests the egress node in the upper layer to set-up the path in the lower layer. However, the VNTM can actively modify the VNT based on the policies and network status without waiting to an explicit PCE request.

Regarding the provisioning phase, VNTM may have to directly talk with an NMS to set-up the connection [RFC5623] or it can delegate this function to the provisioning manager [I-D.farrkingel-pce-abno-architecture].

The aim of this document is not to categorize all implementation options for VNTM, but to present the necessity to retrieve topological information to perform its functions. The VNTM may require the topologies of the lower and/or upper layer and even the inter-layer relation between the upper and lower layer nodes, to decide which is the optimal VNT.

4.3. 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).

- 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.

- 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.

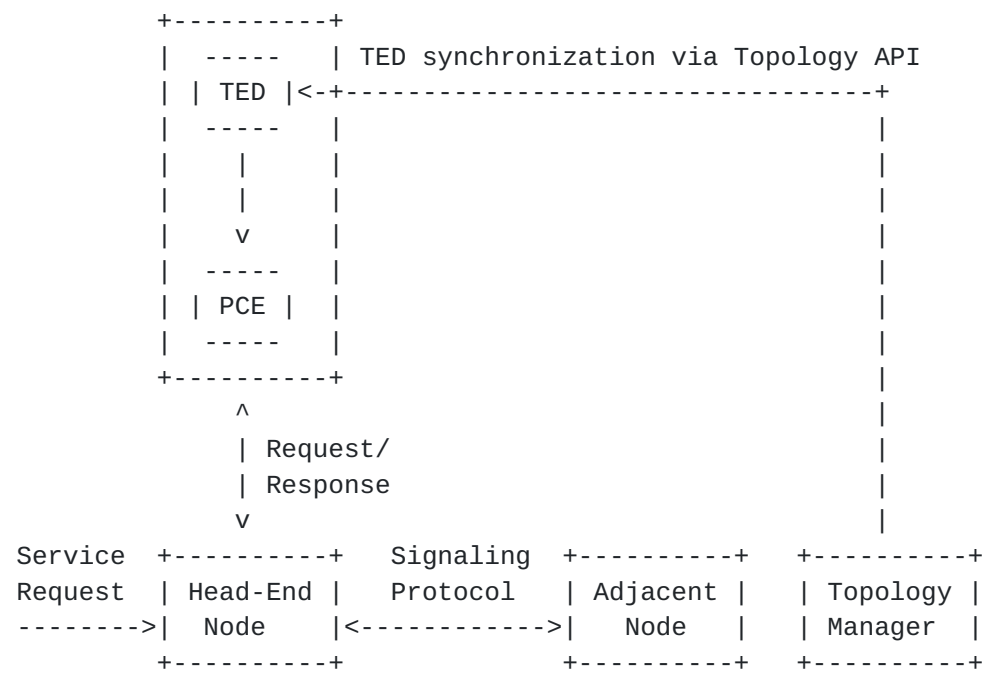

Figure 4: Topology use case: Path Computation Element

4.4. 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.

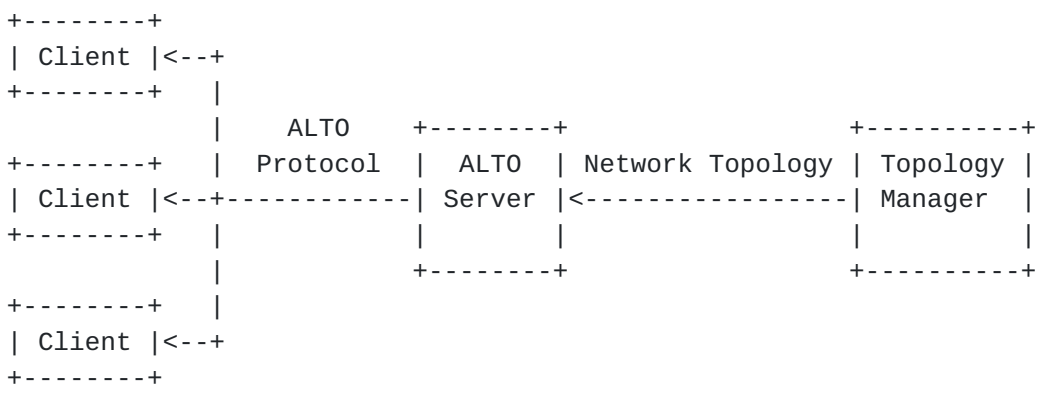

Figure 5: Topology use case: ALTO Server

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.farrkingel-pce-abno-architecture]
King, D. and A. Farrel, "A PCE-based Architecture for Application-based Network Operations", draft-farrkingel-pce-abno-architecture-05 (work in progress), July 2013.

[I-D.ietf-alto-protocol]
Alimi, R., Penno, R., and Y. Yang, "ALTO Protocol", draft-ietf-alto-protocol-17 (work in progress), July 2013.

[I-D.ietf-idr-ls-distribution]
Gredler, H., Medved, J., Previdi, S., Farrel, A., and S. Ray, "North-Bound Distribution of Link-State and TE Information using BGP", draft-ietf-idr-ls-distribution-03 (work in progress), May 2013.

[I-D.ietf-isis-te-metric-extensions]
Previdi, S., Giacalone, S., Ward, D., Drake, J., Atlas, A., and C. Filsfils, "IS-IS Traffic Engineering (TE) Metric Extensions", draft-ietf-isis-te-metric-extensions-00 (work in progress), June 2013.

[I-D.ietf-ospf-te-metric-extensions]
Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", draft-ietf-ospf-te-metric-extensions-04 (work in progress), June 2013.

[I-D.ietf-pce-stateful-pce]
Crabbe, E., Medved, J., Minei, I., and R. Varga, "PCEP Extensions for Stateful PCE", draft-ietf-pce-stateful-pce-05 (work in progress), July 2013.

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

[RFC5212] Shiomoto, K., Papadimitriou, D., Le Roux, JL., Vigoureux, M., and D. Brungard, "Requirements for GMPLS-Based Multi-Region and Multi-Layer Networks (MRN/MLN)", RFC 5212, July 2008.

[RFC5623] Oki, E., Takeda, T., Le Roux, JL., and A. Farrel, "Framework for PCE-Based Inter-Layer MPLS and GMPLS Traffic Engineering", RFC 5623, September 2009.

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

Authors' Addresses

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

Email: jmedved@cisco.com

Stefano Previdi
Cisco Systems, Inc.
170, West Tasman Drive
San Jose, CA 95134
USA

Email: sprevidi@cisco.com

Victor Lopez
Telefonica I+D
c/ Don Ramon de la Cruz 84
Madrid 28006
Spain

Email: vlopez@tid.es

Shane Amante