

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: April 12, 2013

J. Medved
Cisco Systems, Inc.
H. Gredler
Juniper Networks
S. Previdi
Cisco Systems, Inc.
S. Amante
Level 3 Communications, Inc.
October 9, 2012

Topology API Requirements
draft-medved-irs-topology-requirements-00

Abstract

This document describes requirements for collecting topology data from network elements and other sources, creating a coherent view of the network topology from the collected data, and presenting the topology view to various applications that need to: a) understand the topology; and, b) interact/change the topology of the physical or logical network. for reflecting changes to the topology back in network elements..

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 12, 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 3
- 1.1. Requirements Language 3
- 2. Operational Model 3
- 3. Topology Manager Requirements 5
- 3.1. General Requirements 5
- 3.2. Data Model Requirements 6
- 3.2.1. Layer 2 Data Model Requirements 8
- 3.2.2. IP/MPLS Layer Data Model Requirements 9
- 3.3. Northbound (Client) API Requirements 10
- 3.4. Southbound (Network & Device) API Requirements 11
- 4. Network Element Requirements 12
- 5. Acknowledgements 13
- 6. IANA Considerations 13
- 7. Security Considerations 13
- 8. Normative References 13
- Appendix A. Additional Stuff 13
- Authors' Addresses 13

1. Introduction

[I-D.amante-irs-topology-use-cases] demonstrates the need for a network function that would provide a common, standard-based topology view to applications. This function is called 'Topology Manager' in this document. which specifies requirements for the Topology Manager. Topology Manager requirements fall into one of the following categories:

General: describes general requirements for the Topology Manager

Data Model: describes requirements for the network topology data model & view

Northbound (Client) API: describes requirements for Topology Manager's communication with clients

Southbound (Network & Device) API: describes requirements for Topology Manager's communication with Network Elements and other topology data sources

Network Elements: describes requirements for the Network Element's data model, capabilities, etc. required to support collection of network topology data

The rest of this document is organized as follows. First, the Topology Manager's operational model is described in [Section 2](#). Topology Manager requirements are presented in [Section 3](#). Finally, Network Element requirements are presented in [Section 4](#).

1.1. Requirements Language

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

2. Operational Model

As defined in [[I-D.amante-irs-topology-use-cases](#)], the Topology Manager performs the following tasks:

- o Collection of topology data from multiple sources: Network Elements, routing protocols, inventory collection, statistics collection, etc, as discussed in [[I-D.amante-irs-topology-use-cases](#)]. Note that some of the data sources, such as statistics or inventory, will be used not only by the Topology Manager, but by other applications as well. Note

also that topology data sources may reside in multiple IGP areas or multiple ASes, and/or in multiple network layers.

- o Creation of global topology view / common data model from the collected data; the topology view can span multiple network layers as well as multiple routing and switching domains. The global view includes all network elements and resources existing in the infrastructure, whether they are currently actively used or not. An example consists of reconstructing the global view of the network including router or switch ports that are available but not in use.
- o Presentation of the network view / model to clients (applications). The Topology Manager can create multiple application-specific views from its common global topology database.

The operational model is shown in Figure 1.

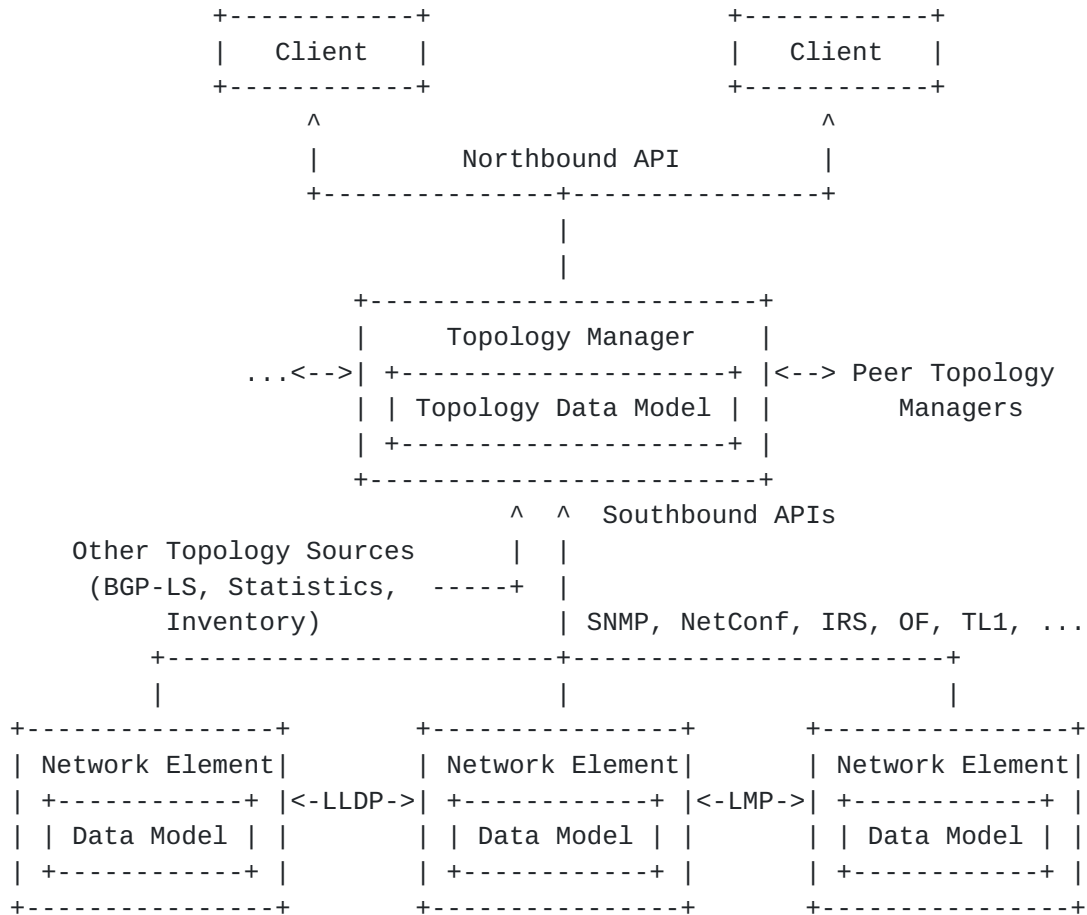


Figure 1: Topology Manager operational model

Topology information from network elements is relayed into the Topology Manager Function via its Southbound API, as shown in Figure 1. Sources of topology information may be Network Elements at different layers of the network, such as appliances, routers, L2 switches, optical transponders, optical switches. or monitoring, provisioning and network analytics tools, such as Statistics Collection or an Inventory subsystem.

The Topology Manager Function reconstructs the network/global view (that can be on a per client or per application base) and distributes it to its clients through northbound APIs. Examples of possible northbound APIs are ReST, XMPP or Websockets. The Topology Manager Function can be instantiated in a standalone server, be a part of a comprehensive orchestration / data collection / presentation framework (see [[I-D.amante-irs-topology-use-cases](#)]), or embedded in a routing element. A client can be an application or a function in an upper layer framework, such as a policy function.

Depending on the data it collects, a Topology Manager may not have visibility into the entire network. In order to create a global topology, the Topology Manager may get complementary partial topologie views from other Topology Managers via a Peer Topology Manager API.

3. Topology Manager Requirements

3.1. General Requirements

The general requirements are as follows:

- TMF.GEN-1: IRS MUST define a common vocabulary that describes attributes associated with topological components of a network. This is more commonly referred to as a "normalized" topology.

- TMF.GEN-2: IRS MUST define a data model that describes the topological components represented by the Topology Manager service. This data model will be written using a common vocabulary, that allows one to assemble the components of a network topology so that one can, for example, describe a physical link and all of its associated physical layer attributes such as: media type, bandwidth, MTU, etc.

- TMF.GEN-3: The Topology Manager has a Northbound (Client) API and multiple Southbound APIs for collecting topology data from networks. Southbound APIs can be, for example, SNMP, NETCONF, CLI, IRS, OpenFlow, or routing protocols (IGPs/BGP).
- TMF.GEN-4: The Topology Manager has an East-West (Peer Manager) API to exchange topology information with peer Topology Managers. Topologies exchanged with peer Topology Managers can be real or abstract. Note that the East-West API can be the same as the Northbound API.
- TMF.GEN-5: The Topology Manager MUST be able to collect and process topology data from hundreds of thousands of sources (network elements, etc.). Being able to collect data from large number of sources is especially important in very large optical and transport network.
- TMF.GEN-6: The Topology Manager SHOULD be able to provide multiple application-specific views to different clients.
- TMF.GEN-7: The Topology Manager MUST allow for bi-directional flow of topology data between clients and the network: clients MUST be able to get network topology information and to inject policies and/or state related to network topology.
- TMF.GEN-8: The transport protocol on all Topology Manager APIs MUST support incremental updates between Topology Managers or between a Topology Manager and a client.

3.2. Data Model Requirements

The requirements for the topology data model are as follows:

- TMF.DM-1: The topology data model MUST be able to describe topology and characteristics of different network layers:
- * Optical DWDM (for future study)
 - * Optical OTN (for future study)
 - * L2 (Aggregated links, L2 topologies)
 - * IP/MPLS,

- * VPNs

- * Services, such as cloud services, or CDNs

- TMF.DM-2: The topology data model MUST support multiple administrative domains.
- TMF.DM-3: The Topology Manager MUST provide data normalization, i.e. various types of topology information from different network elements at different network layers and in different admin domains is processed into a single, common format for clients' consumption.
- TMF.DM-4: The topology data model MUST be able to convey enough information so that a client can correlate topologies in different layers and from multiple administrative domains.
- TMF.DM-5: The topology data model MUST support multi-layer grouping of elements as a means of coalescing different nodes and links into "network layers". For example, links with IPv4 addresses might represent Layer 3 of the network topology while links with Ethernet MAC addresses might represent Layer 2. Furthermore, virtual private networks can be represented using this grouping mechanism.
- TMF.DM-6: Association between components of different layers is needed as a means of indicating relationships (i.e.: dependency, stacking, etc...) between components where layers are associated. For example, a layer 2 port might have several IPv4 or IPv6 interfaces that utilize it. These would be represented as associations to and from these components.
- TMF.DM-7: Both active and inactive topologies MUST be represented in the topology database. Inactive topology is not exhaustively seen in IGP routing protocols. It must be retrieved by querying inventory in network elements, for example new line cards inserted in a chassis, new chassis, ports in the down state, etc.
- TMF.DM-8: The topology data model MUST be hierarchical and MUST support summarization of sub-topologies. Topology summarization and creation of abstract topologies can be provided by the Topology Manager or the Orchestration Function

- TMF.DM-9: The topology data model MUST be able to describe abstract topologies. Abstract topologies can contain real and abstract nodes and real and abstract links. An abstract topology MAY be used by a provider to describe characteristics of a transit network (bandwidth, delay, protection, etc.)
- TMF.DM-10: The topology data model MUST support dynamic data, such as link and node utilizations (perhaps as optional attributes).
- TMF.DM-11: The topology data model MUST allow a user (operator) to determine the path between two nodes. The path should be traceable at all network layers that participate in the delivery of packets between two nodes. For example, for IP traffic exchanged between 2 routers, the user should be able to identify all L2 switches and optical switches that carry the traffic between the routers.
- TMF.DM-12: The topology data model MUST support multiple BGP Autonomous Systems and multiple IGP areas. Support for multiple administrative domains is for further study.
- TMF.DM-13: The topology data model MUST be human-friendly, i.e. not SNMP MIBs, but something much more analogous to YANG models.
- TMF.DM-14: The data model SHOULD support topology abstraction, allowing clients that consume topology information in a constrained manner. For example, a client wishing to view only interfaces and nodes present in a sub-graph of the Layer 3 topology should be able to specify an interest in this subset of information rather than having to read out and parse through the entire set of links and nodes.

3.2.1. Layer 2 Data Model Requirements

The requirements for the L2 topology data model are as follows:

- TMF.DM.L2-1: Inventory, (link) status and counter information MUST be retrieved from L2 Network Elements via NETCONF and SNMP.
- TMF.DM.L2-2: L2 Network Elements MUST be able to provide data obtained from L2 topology discovery protocols.

3.2.2. IP/MPLS Layer Data Model Requirements

The requirements for the IP/MPLS topology data model are as follows:

- TMF.DM.IP-1: The data model of the IP/MPLS layer MUST support both link topology and prefixes.
- TMF.DM.IP-2: Link topology may be retrieved from an IGP, BGP-LS, or by getting node neighbor information directly from network elements via a management protocol such as SNMP, NETCONF or CLI.
- TMF.DM.IP-3: Links in the IP/MPLS data model can include, among others, one or more of the following link attributes:
- * Local and Remote anchor node IDs (Router ID, AS#, Area ID, MT Topology ID)
 - * Metrics
 - * Admin Group
 - * Max link bandwidth
 - * Unreserved / utilized bandwidth
 - * Link Protection type
 - * MPLS Protocol mask
 - * IS-IS DIS
 - * OSPF DR/BDR
 - * Link prefix
 - * Link characteristics (BW, delay, error rate)
 - * Link description
 - * Link-specific timers, (Hello & Holddown)
- TMF.DM.IP-4: Nodes in the IP/MPLS data model MAY include one or more of the following node attributes:
- * Node Type: simple node / aggregate node

- * Intra area router
- * Inter area router (ISIS L1L2 or OSPF ABR)
- * AS boundary router
- * MPLS-VPN Edge router (PE)
- * Tunnel head-end/tail-end
- * Node ID:: Node ID (Router ID, AS#, Area ID, MT Topology ID)
- * Flags: Overload, Attached, External, ABR
- * Node capabilities as defined in [RFC5073](#)
- * BGP: aggregate IP prefix + mask (with associated tie-down route information to inject the aggregate route into BGP)
- * BGP policy associated with a given iBGP/eBGP neighbor; policy matching criteria can be, among others, remote-AS, local-AS, Local-AS tweaks to manipulate AS_PATH recv'd or transmitted, timers (Hello, HoldTime), AFI/SAFI, route-map/policy-statements applied/active (including associated prefix-lists, AS_PATH regexp filters, etc. and resulting manipulation of BGP path attributes, e.g.: changing LOCAL_PREF, MED, BGP community, etc.)
- * BGP statistics collection: number of IP paths/ prefixes learned per neighbor

3.3. Northbound (Client) API Requirements

The requirements for the Topology Manager's northbound (client) interface are as follows:

- TMF.NB-1: The transport protocol between the Topology Manager and its clients SHOULD have efficient encoding so that large topologies can be transferred with reasonable performance.

- TMF.NB-2: The transport protocol MUST support flow-control in case a client cannot digest updates fast enough from its server.
- TMF.NB-3: The transport protocol MUST support push of topology updates from the Topology Manager to clients.
- TMF.NB-4: The protocol MUST implement a mechanism through which a client can efficiently determine the latest information especially when it receives multiple copies of the same topology from multiple Topology Managers. An example is the IGP Sequence Number that is used on IGP routing updates.
- TMF.NB-5: The Northbound API MUST support publishing of events to a dynamic list of clients/applications. This is helpful for the Northbound API to signal events as they occur in the network, e.g.: insertion or removal of linecards, etc.
- TMF.NB-6: The Northbound API SHOULD support subscription to events from a dynamic list of clients/applications. This will allow the Topology System to react dynamically when, for example, new requests for services are asked to be created.
- TMF.NB-7: The Northbound API SHOULD allow clients to subscribe to a rich assortment of attributes and/or data models so that they are automatically notified of changes within the network, (as a result of a node failure, card insertion, etc.), as well as changes made by other upper-layer applications to the network, (for example, addition/subtraction of physical links to the network during network augmentation or optimization, etc.)
- TMF.NB-8: The Northbound API MUST provide a means for non-routers to communicate with the model. Until now, clients that could gather network topology and inventory information were generally limited to those that could speak routing protocols.

3.4. Southbound (Network & Device) API Requirements

The requirements for the Topology Manager's southbound (network element) interface are as follows:

- TMF.SB-1: The transport protocol between the Topology Manager and the topology data sources (Network Elements, etc.) SHOULD have efficient encoding so that large data models can be transferred with reasonable performance.
- TMF.SB-2: The transport protocol MUST support incremental updates from a Network Element to the Topology Manager.
- TMF.SB-3: The transport protocol MUST support push of topology updates from a Network Element to the Topology Manager.

4. Network Element Requirements

The requirements for the topology data model are as follows:

- NE-1: Each network element SHOULD contain an inventory database, which SHOULD be a definitive source of information with respect to the physical HW and logical, locally significant identifiers, e.g.: VLANs, deployed in the system. The Topology Manager collects inventory data from network elements and creates an authoritative view of physical hardware deployed in the network.
- NE-2: The inventory DB SHOULD be augmented with the physical properties associated with the ports/interfaces that are directly connected to the device, (e.g.: BW, media type, etc.).
- NE-3: The inventory DB MAY be augmented through the IRS framework pushing information into the network element to, for example, associate information the installer may have knowledge of, but the network element may not be able to learn on its own, e.g.: it's physical location (address), rack/bay information, etc.
- NE-4: Relevant network elements should automatically and dynamically acquire information associated with their immediately adjacent neighbors using protocols such as LLDP, LMP, etc. A network element should further augment it's physical port inventory DB with this information so that the system can report on whom it's directly connected.

5. Acknowledgements

The authors would like to thank Alia Atlas, Chris Metz, Dave Ward, and Tom Nadeau for their comments and input.

6. IANA Considerations

This memo includes no request to IANA.

7. Security Considerations

tbd.

8. Normative References

[I-D.amante-irs-topology-use-cases]

Amante, S., Medved, J., and T. Nadeau, "Topology API Use Cases", [draft-amante-irs-topology-use-cases-00](#) (work in progress), October 2012.

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

Appendix A. Additional Stuff

This becomes an Appendix.

Authors' Addresses

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

Email: jmedved@cisco.com

Hannes Gredler
Juniper Networks
USA

Email: hannes@juniper.net

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

Email: sprevidi@cisco.com

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

Email: shane@level3.net

