

ALTO WG  
Internet-Draft  
Intended status: Informational  
Expires: September 10, 2015

R. Huang  
Huawei  
H. Guo  
CAICT  
Y. Yang  
Yale University  
March 9, 2015

Network Topology Service (NTS) Framework  
draft-huang-alto-topocomp-framework-00

Abstract

This document introduces a network topology service (NTS) framework to collect network topologies from the physical heterogeneous network, NTS analyses and stores the topology information, and provides the path computing and topology information inquiring ability to applications (including network applications like OSS, and third-party applications).

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/lid-abstracts.html>

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>

Copyright and License Notice

Copyright (c) 2015 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 . . . . .	2
2. Terminology . . . . .	3
3. Network Topology Service (NTS) Framework . . . . .	3
4. Topology Managing of NTS Framework . . . . .	6
5. Topology Inquiring of NTS Framework . . . . .	6
6. Path Computing of NTS Framework . . . . .	7
7. Relationship with Other Existing IETF work . . . . .	8
7.1. I2RS . . . . .	8
7.2. PCE . . . . .	8
8. Security Considerations . . . . .	8
9. IANA Considerations . . . . .	8
10. Acknowledgments . . . . .	8
11. References . . . . .	8
11.1. Normative References . . . . .	8
11.2. Informative References . . . . .	8
Authors' Addresses . . . . .	9

## 1. Introduction

Network topology is a prerequisite for performing many critical network management tasks, including resource managements, path computation, event correlation, fault monitoring and analysis. Current networks are continually being refined and upgraded as needs change and technology evolves. Many technologies have developed protocol-specific ways to obtain network topologies for their own usages. For example, a router supporting OSPF maintains an identical area-topology database to determine the shortest path to any neighboring router; BGP maintains a consistent view of network topology to optimize routing and scale the network. However, when network topologies or route paths are required by applications,

applications usually wish to be shielded from protocol-specific information, even if network state information is collected in protocol-specific ways. It is obvious that none of these methods offer a general-purpose tool that can efficiently manage the network topology for a heterogeneous network with multiple technologies including BGP/OSPF/ISIS, and even SDN Open Flow, etc.

This document introduces a network topology service (NTS) framework to collect network topologies from the physical heterogeneous network, NTS analyses and stores the topology information, and provides flexible path computing and topology information inquiring ability to applications (including network applications like OSS, and third-party applications).

## 2. Terminology

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].

This document uses the following terms:

NTS: Network Topology Service

BGP: Border Gateway Protocol

OSPF: Open Shortest Path First

IS-IS: Intermediate System to Intermediate System

SDN: Software Defined Network

OSS: Operational Support Systems

## 3. Network Topology Service (NTS) Framework

This section describes the NTS framework as shown in Figure 1:



Topology Agent: A logical entity located in network devices like switches, routers, etc. It is responsible for reporting the topology information produced in some protocol-specific ways and network changes, event, or states to its topology server. One topology agent can only be controlled by one topology server to avoid global network topology duplicating.

Topology Server: A server that collects topology information from physical network for a subset of devices, analyses, abstracts, and stores the subset topology information in a protocol independent way. Usually, a large network is too hard for a single topology server to handle. Thus, multiple topology servers are considered in this framework. Each of them is only responsible for a part of the global network. Topology server has the ability to calculate the special topology or the most optimized path based on specific algorithms from applications. Different algorithms may lead to different results.

Aggregator: A server that maintains a sketch topology information among all topology servers. It does not perceive any detailed subset topology information as individual topology servers. This entity is only responsible for generating and maintaining relationship among different topology servers, and calculating the final topology or optimized path based on the results calculated from some or all of the topology servers.

Application: It represents network applications like OSS, and third-party applications that require to use network topology service.

The interfaces needed in this framework:

Interface between topology agent and topology server: An interface that can be used by TA to report different protocol-specific topology information, e.g., BGP/OSPF/IS-IS, or SDN OpenFlow, to TS. Besides, TA can use it to notify TS the changes, states, and events.

Interface between topology server and aggregator: Communication between topology servers and aggregator. It includes topology servers reporting their ingress and egress information to the aggregator, aggregator conveying applications' local topology algorithms information to topology servers, and topology servers returning their calculation results based on the local topology algorithms from applications to the aggregator.

Topology Service Interface: This interface is used by applications to communicate with the aggregator on path computation requesting

and topology information obtaining. Applications use the interface to insert their own algorithms and requirements for the network topology service system to do some application specific calculations. Two kinds of algorithms are considered here: One for a topology server to calculate the special topology or the most optimized path based on its subset topology information (local topology algorithm); The other for the aggregator to calculate the global and the special topology or the most optimized path based on the results from all topology servers and the sketch topology information generated by the aggregator (global topology algorithm).

#### 4. Topology Managing of NTS Framework

In NTS Framework, there are two level topologies: One is the generic network topology that is managed by TS; Other is managed by Aggregator.

TA discovers network topology and states/events, then reports the protocol-specific network topology to TS. TS analyses the network topology information, and constructs a generic network topology. TS stores the generic network topology in a protocol independent way. TA also can generate an abstract topology basing the generic network topology by partition or aggregation to avoid giving the detail network information to Application. TS also reports the ingress and egress information of its managing subnet network to the Aggregator.

Aggregator generates a sketch topology information reflecting the relationship among all the TSes. In a sketch topology, each subset network managed by a TS is condensed into a node. Aggregation may be give the sketch topology to Application to assist the path computing

#### 5. Topology Inquiring of NTS Framework

Any Application can inquiry a special network topology, for example Network Map or Cost Map in ALTO service, from the NTS framework. The detailed steps of topology inquiring is listed as following:

- \* Application inputs local topology algorithm and global topology algorithm the aggregator to request a special topology.
- \* Aggregator instructs all (or some relevant) TSes to calculate their own special topology in their subset topologies based on the local topology algorithm of the application.
- \* TS independently calculates the special topology in its subset

topology, and reports its result to the aggregator.

\* Aggregator calculates the final global topology based on the results of all the TSes, the sketch topology information, and the global topology algorithm, then Aggregator returns the final topology to Application.

## 6. Path Computing of NTS Framework

In SDN network, applications without knowledge of physical network can be benefit from the NTS framework to obtain the most suitable and efficient network path based on which they can then do some programming.

The detailed steps of path computing is listed as following:

\* Application inputs local topology algorithm, global topology algorithm, source information and destination information to the aggregator to request an optimized path.

\* (Optional) Aggregator gives the sketch topology to Application.

\* (Optional) With the sketch topology, Application decides whether or not to filter some irrelevant-like TSes that maybe not appear in the end to end network path. If so, Application inputs the filtering algorithm to Aggregator.

\* (Optional) Aggregator uses the filtering algorithm to filter some irrelevant-like TSes.

\* Aggregator instructs all (or some relevant) TSes to calculate their own optimized path in their subset topologies based on the local topology algorithm of the application.

\* TS independently calculates the optimized path in its subset topology, and reports its result to the aggregator.

\* Aggregator calculates the final global optimized path based on the results of all the TSes, the sketch topology information, and the global topology algorithm, then Aggregator returns the final global optimized path to Application.

When the number of TSes increasing, the performance of this framework maybe reduced as all of the TSes need to do calculation. This can be solved by applications inputting another algorithm or requirement to allow the aggregator filtering some irrelevant-like TSes before

sending any instructions to TSes. Thus, only those TSes which are responsible for the network topology between the end-to-end network path are required to do the calculation. However, this function is optional here since some applications may need to all of the TSes to take part in the calculation.

## 7. Relationship with Other Existing IETF work

### 7.1. I2RS

I2RS is discussing a generic topology data model. However, current I2RS charter says it is not responsible to develop protocols, encoding languages, or data models. The topology work in I2RS can be considered to use in the interface between TA and TS. However, I2RS will not discuss a detailed topology service. The protocols and data models produced in I2RS can be considered in this work.

### 7.2. PCE

PCE is discussing to specify the required protocols so as to compute of paths for MPLS and GMPLS Point to Point and Point to Multi-point Traffic Engineered LSPs. PCE does not consider to manage the network topology from multiple protocols. NTS framework can provides more flexible path computing algorithm and topology information inquiring.

## 8. Security Considerations

TBD.

## 9. IANA Considerations

This document does not require any IANA creations or modifications.

## 10. Acknowledgments

TBD.

## 11. References

### 11.1. Normative References

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

### 11.2. Informative References

Authors' Addresses

Rachel Huang  
Huawei  
101 Software Avenue, Yuhua District  
Nanjing 210012  
China

EMail: rachel.huang@huawei.com

Huaming Guo  
China Academy of Information and Communications Technology  
36 A Nanlishi Road, Xicheng District  
Beijing 10037  
China

EMail: guohuaming@caict.ac

Y. Richard Yang  
Yale University  
51 Prospect St  
New Haven, CT 06511  
USA

EMail: yry@cs.yale.edu