

Network Working Group  
Internet Draft  
Intended status: Informational  
Expires: January 3, 2015

B. Liu, Ed.  
G. Yan  
Huawei Technologies  
July 2, 2014

Multi-Instances Configuration Issue in NETCONF  
draft-liu-netconf-multi-instances-00.txt

## Abstract

This document puts together the NETCONF issues of configuring multiple instances including configuring multiple network element instances and multiple service instances. The main problem is how to configure the multiple instances in one NETCONF channel.

## 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 January 3, 2015.

## Copyright Notice

Copyright (c) 2011 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
2. Requirements Language and Terminology .....	3
3. Multiple Instance Management Requirements and Gaps .....	4
3.1. Multiple Instance Introduction .....	4
3.1.1. Multiple Network Element Instances (MNEI).....	4
3.1.2. Multiple Service Instances (MSI) .....	4
3.2. Management Requirements .....	5
3.2.1. MNEI Management Requirements .....	5
3.2.2. MSI Management Requirements .....	5
3.3. Gap Analysis .....	6
4. Solutions Discussion.....	7
4.1. NETCONF Context Extension .....	7
4.2. Common Service Container .....	8
4.3. Allowing Reusing One module in another .....	8
4.4. Common MSI Data Model .....	8
5. Security Considerations .....	8
6. IANA Considerations .....	8
7. References .....	9
7.1. Normative References .....	9
7.2. Informative References .....	9
8. Acknowledgments .....	9
Authors' Addresses .....	10

## 1. Introduction

In modern networks, according to the wide spread of network multiplex technologies such as VPN, a key network element (e.g. a PE router) or a basic service (e.g. routing) is usually separated as multiple instances to support network multiplex more efficiently.

NETCONF is been adopted by more and more network management systems to be the southbound interface of configuring the devices. When using NETCONF to configure the above mentioned key network element or service, sometimes it is actually to configure each instance rather than configure the whole network element. However, current NETCONF protocol and YANG models seem to not specifically consider the multiple instance configuration issues.

This document discusses the multiple instances configuring issue among the various scenarios. Multiple instances are separated as multiple network element instances and multiple service instances to be discussed respectively.

## 2. Requirements Language and Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119] when they appear in ALL CAPS. When these words are not in ALL CAPS (such as "should" or "Should"), they have their usual English meanings, and are not to be interpreted as [RFC2119] key words.

Terminology:

NEI: network element instance

SI: service instance

MNEI: multiple network element instance

MSI: multiple service instance

### 3. Multiple Instance Management Requirements and Gaps

#### 3.1. Multiple Instance Introduction

##### 3.1.1. Multiple Network Element Instances (MNEI)

###### o Logic System LS

A network element can be divided into multiple logical systems that each system can deal with a task independent from others. For example, a physical router can be divided into multiple logical systems (an LS on router could be also called LR: Logical Router) and each LS can run independent routing/MPLS protocols such as OSPF, IS-IS, BGP, RIP, LDP, and RSVP-TE .etc.

###### o Virtual System (VS)

A network element can also be divided into multiple virtual systems. The VSes also can do different tasks respectively. Different with LS, VS is basically virtualized in software level. If LS and VS are co-existing in one platform, one VS normally belongs to a specific LS and one LS could have multiple VSes.

For example, a physical router can also be divided into multiple virtual systems (then it could be called VR: virtual router). A VR can also run independent routing/MPLS protocols as OSPF, IS-IS, BGP, RIP, LDP, and RSVP-TE .etc. So in the perspective of processing a service, VS could be seen as equivalent to LS.

##### 3.1.2. Multiple Service Instances (MSI)

Following are several representative examples of multiple service instances.

###### o VRF Virtual Routing and Forwarding

VRF allows a router to have multiple independent routing tables so that they could deal with different routing tasks.

VRF could be seen as a container SI, which means it might contain other SIs such as following IGP/MTR SI.

###### o IGP Multi-Processes

OSPF and IS-IS can also have multiple instances in one device in the form of multiple processes of the same protocol stack. Each process

has its own independent routing area, path calculation and data stores like routing tables .etc.

- o MTR Multi-Topology Routing

From the perspective of services (e.g. voice, video, data traffic .etc), one physical network could be seen as different higher-layer topologies. MTR technology allows the forwarding capabilities to be separated for each topology. Specifically, MTR supports a separate multicast topology and multiple unicast topologies.

Normally, an MTR instance belongs to a specific VRF, and one VRF could contain multiple MTR instances.

### 3.2. Management Requirements

#### 3.2.1. MNEI Management Requirements

- o Req-1: Independently Managing Each MNEI

Management of multiple network element instances normally has the following two modes.

- Independent Management

- In independent management mode, the NMS considers each instance as a standalone network element. All the features of them, except board-level hardware features, could be operated independently from each other by the user.

- Central Management

- In central management mode, normally the NMS does not need to independently manage each instance in network element management level. However, there are two exceptions:

- 1) When the instance is at initialing or failure stage, the NMS still needs to distinguish and configure each instance.

- 2) Although NMS doesn't need to independently manage each NEI, it still needs to independently manage different services. So when the services are bear on different NEIs, the NMS still needs to distinguish the NEIs in terms of distinguishing services.

In summary, both in independent management mode and central management mode, the NMS needs to independently manage each NEI.

#### 3.2.2. MSI Management Requirements

- o Req-2: Independently Managing Each MSI

Most of the time, the NMS also needs to distinguish different service instances, because they do different tasks which the NMS needs to configure respectively. So in NMS perspective, the requirement of independently managing each instance is similar between MSI and MNEI.

o Req-3: Crossing Configuration between Different SIs

Besides independently managing each SI for a standalone service, MSI has more complicity than MNEI that one SI might logically belong to another container SI. For example, an IGP instance might belong to a VRF instance. This requirement could have two choices as the following.

- Req-3a: Configure one SI under another container SI

As mentioned above, since SI-A might logically belong to SI-B, the straightforward way for the NMS to configure SI-A is to configure it under SI-B. For the above example, the NMS can configure an IGP instance under the hosting VRF configuration and does not need to particularly configure it in IGP.

- Req-3b: Configuring one SI binding to another container SI

Instead of Req-3a, Req-3b requires the NMS to directly configure the targeted SI and bind it to another container SI. For example, the NMS configures the IGP instance and clearly specify the IGP instance is for a specific VRF instance.

### 3.3. Gap Analysis

o Gap-1: NETCONF cannot configure multiple agents within one session

For R1 (independently manage each MNEI), in independent management mode, each instance would be assigned an IP address that the NMS could initial one NETCONF session for each thus there is no problem. However, when the instances are at initialing or failure stage, the NMS just couldn't connect to them directly to set up NETCONF sessions. The NMS will have to set up a NETCONF session with the main agent and configure different instances within this NETCONF session.

In central management mode, since the instances belong to one main network element, they often need to share the same IP address for communication to the NMS. According to R1, when the NMS needs to independently configure the instances, it also needs to configure them within the NETCONF session which is set between the NMS and the main device.

It is obvious that current NETCONF doesn't support configuring multiple agents within one NETCONF session, because NETCONF session is set up on a transport-layer (SSH, TLS .etc) channel, thus one NETCONF session can only correspond to one IP address.

o Gap-2: Lack of Multiple Instances Extension Capability in Data Models

For Req-2 (independently manage each MSI), since MSIs also need to share the same IP address for communication to the NMS, the NMS also needs to configure multiple services within one NETCONF session, which is similar with Gap-1. However, the bigger issue is that if a service model was not constructed as multiple instances at first, it is hard to be extended to support multiple instances. If the service models would support multiple instances internally, they will have to be re-constructed, which is a heavy burden for both standard revision and implementation. And the model re-construction needs to be one by one, thus it is not scalable.

o Gap-3a: Cannot Reuse one Module in another

For Req-3a (Configure one SI under another container SI), if the targeted service module could be directly reused when defining the container module, it would be very convenient to support the Req-3a operation. However, current YANG model doesn't have the capability.

o Gap-3b: Modules are not aware of MSI of the container module

For Req-3b (Configuring one SI binding to another container SI), it requires the targeted service module to be aware of the MSI of the container module so that the NMS could bind it to a specific container SI through a key.

#### 4. Solutions Discussion

##### 4.1. NETCONF Context Extension

As defined in [RFC3411], an SNMP context is a collection of management information accessible by an SNMP entity. An item of management information may exist in more than one context. An SNMP entity potentially has access to many contexts.

In NETCONF, similar context extension could be done that the multiple instances could be distinguished in one NETCONF session. To extend context, several key problems need to be solved as the following:

- To add a Context field in current NETCONF <rpc-request> message.
- There needs a kind of ID to identify each instance.
- The NMS needs to collect the list of the instance IDs so that it can specify which instance to be configured in the context field.

The context extension is mostly suitable for Gap-1 (NETCONF cannot configure multiple objects within one session). Especially in the MNEI cases, the context is just like simply pointing to another network element and nothing is relevant to data modules. The context extension can also be used for MSIs; however, the service module itself has to support multiple instances as the prediction.

#### 4.2. Common Service Container

Make a service container which indexes all the services in the network element into a single dedicated list. Thus, if some a service model needs to be extended to support multi-instance, it only needs the service container to reconstruct without reconstructing the data models.

This approach mainly fits the Gap-2. Gap-3b could also consider this approach.

#### 4.3. Allowing Reusing One module in another

As described in Req-3a, YANG could be extended to supporting module-level reuse rather than current group-level reuse.

#### 4.4. Common MSI Data Model

This issue was discussed in IETF NETCONF and NETMOD mailing list and the result of the discussion was to write a module defining the various types of service instances (e.g. VRF, IGP Multi, MTR .etc) and specifying the semantics of this kind of multi-instance. Then, if a data model needs to be aware of multiple service instance, multi-service-id (e.g. VRF-id) leafs will be added to the existing module via augments.

This approach basically fit the Gap-3b.

#### 5. Security Considerations

TBD.

#### 6. IANA Considerations

None.



## 7. References

### 7.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.
- [RFC6241] Enns, R., Bjorklund, M., Schoenwaelder, J., and A. Bierman, "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

### 7.2. Informative References

- [RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.

## 8. Acknowledgments

Many valuable comments were received from Guangying Zheng and Xiaofeng Ji to improve the draft.

This document was prepared using 2-Word-v2.0.template.dot.

Authors' Addresses

Bing Liu  
Q14-4-A Building  
Huawei Technologies Co., Ltd  
Zhong-Guan-Cun Environment Protection Park, No.156 Beiqing Rd.  
Hai-Dian District, Beijing  
P.R. China

Email: leo.liubing@huawei.com

Gang Yan  
Q15-2-A Building  
Huawei Technologies Co., Ltd  
Zhong-Guan-Cun Environment Protection Park, No.156 Beiqing Rd.  
Hai-Dian District, Beijing  
P.R. China

Email: yangang@huawei.com

