

NETCONF Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 8, 2019

W. Zheng
Q. Wu
Huawei
November 4, 2018

Inline Action Capability for NETCONF
draft-zheng-netconf-inline-action-capability-02

Abstract

NETCONF provides mechanism to install configuration of network devices. In many cases, it is required that the same configuration repeats on many interfaces. In the absence of protocol semantics for performing operations with group-specific scope, this results in either a significant amount of signaling traffic and configuration template applying on a periodic basis or large packet size between a given network management system and a network devices. This document defines optimizations to the NETCONF protocol operations for performing operations with group-specific scope with the use of a group identifier.

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 <https://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 May 8, 2019.

Copyright Notice

Copyright (c) 2018 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 (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

Internet-Draft

Inline Action Capability

November 2018

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
1.1.	Terminology	2
2.	Inline-Action Capability	3
2.1.	Description	3
2.2.	Dependencies	3
2.3.	Capability Identifier	3
2.4.	New Operations	3
2.5.	Modifications to Existing Operations	3
3.	Security Considerations	9
4.	IANA Considerations	9
4.1.	NETCONF Capability URN	9
5.	Normative References	10
	Authors' Addresses	10

[1.](#) Introduction

NETCONF provides mechanism to install configuration of network devices. In many cases, it is required that the same configuration repeats on many interfaces, e.g., configure multiple VLAN ranges on the same Trunk interface. In the absence of protocol semantics for performing operations with group-specific scope, using configuration template to replicate multiple copies on the same interface result in either a significant amount of signaling traffic(e.g, multiple data retrieval for vlan tag configuration on the interface) on a periodic basis or large packet size (e.g,edit-config operation related to protocol message) between a given network management system and a network devices.

This document defines optimizations to the NETCONF protocol operation(i.e., inline action operation) for performing bulk operations with group-specific scope with the use of a group identifier and allows NETCONF protocol operation work together with inline action operation that apply to different conceptual node in the underlying data model in one transaction.

[1.1.](#) 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 BCP

14 [[RFC2119](#)] [[RFC8174](#)] when, and only when, they appear in all capitals, as shown here.

[2.](#) Inline-Action Capability

[2.1.](#) Description

The :inline-action capability indicates that the device supports Inline-action operation within protocol operation on specific datastore. In other words, the device supports <record-merge> operation which is included in existing protocol operations.

[2.2.](#) Dependencies

None.

[2.3.](#) Capability Identifier

The :inline-action capability is identified by the following capability string:

urn:ietf:params:netconf:capability:inline-action:1.1

[2.4.](#) New Operations

None.

[2.5.](#) Modifications to Existing Operations

The :inline-action:1.1 capability modifies the protocol operation to accept <record-merge> attribute value within operation attribute.

As described in [[RFC6241](#)], "operation" attribute is defined in a element within <config> subtree and identify the point in the configuration to perform the operation and MAY appear on multiple elements throughout the <config> subtree. In this document, two new

"operation" attribute values are added as follows:

record-split: The range constraint of the configuration data identified by the element containing this attribute is split at the corresponding level in the configuration datastore identified by the <target> parameter.

record-merge: The range constraint of the configuration data identified by the element containing this attribute is merged at the corresponding level in the configuration datastore identified by the <target> parameter.

In addition, the record-merge and record-split operation attributes and other "operation" attributes MUST apply to the different conceptual nodes in the underlying data model.

As described in [[RFC6241](#)], the config subtree is expressed as a hierarchy of configuration data as defined by one of the device's data models. The contents MUST follow the constraints of that data model, as defined by its capability definition. If inline action capability is supported, the config subtree may contain a schema node with the name "input" and a schema node with the name "output" connected to a specific container or list data node containing record-merge element in a datastore.

Example:

```
container interfaces {
  list trunk-interface {
    key "name";
    config true;
    leaf name {
      type string;
    }
  }
  container vlan-id-ranges{
    list vlan-id-range{
      key "group-id";
      leaf group-id {
        type string;
        description
          "Specified VLAN group ID.";
      }
    }
  }
}
```


capability, edit-config with configuration template will be used to merge them as one record with extra computation in the client app.

```
<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.1">
  <edit-config>
    <target>
      <running/>
    </target>
    <default-operation>none</default-operation>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.1">
      <top xmlns="http://example.com/schema/1.2/config">
        <interfaces>
          <interface>
            <name>Ethernet0/0</name>
            <vlan-id-ranges>
              <vlan-id-range xc:operation="delete">
                <group-id>0</group-id>
```

```

        <lower-vlan-id>1</lower-vlan-id>
        <upper-vlan-id>3</upper-vlan-id>
    </vlan-id-range>
    <vlan-id-range xc:operation="delete">
        <group-id>2</group-id>
        <lower-vlan-id>5</lower-vlan-id>
        <upper-vlan-id>6</upper-vlan-id>
    </vlan-id-range>
    <vlan-id-range xc:operation="delete">
        <group-id>3</group-id>
        <lower-vlan-id>7</lower-vlan-id>
        <upper-vlan-id>8</upper-vlan-id>
    </vlan-id-range>
    <vlan-id-range xc:operation="delete">
        <group-id>4</group-id>
        <lower-vlan-id>9</lower-vlan-id>
        <upper-vlan-id>10</upper-vlan-id>
    </vlan-id-range>
    <vlan-id-range xc:operation="create">
        <group-id>0</group-id>
        <lower-vlan-id>1</lower-vlan-id>
        <upper-vlan-id>10</upper-vlan-id>
    </vlan-id-range>
</vlan-id-ranges>
</interface>
</interfaces>
</top>
</config>
</edit-config>
</rpc>

```

If inline-action capability supported, bulk operation in one protocol message will be used to merge multiple records with different vlan tag range will into one record:

```

<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.1">
  <edit-config>
    <target>

```

```

    <running/>
  </target>
  <default-operation>none</default-operation>
  <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.1">
    <top xmlns="http://example.com/schema/1.2/config">
      <interfaces>
        <interface xc:operation="record-merge">
          <name>Ethernet0/0</name>
          <vlan-id-ranges>
            <vlan-id-range>
              <action xmlns="http://example.com/schema/1.2/config">
                <range-merge>
                  <input>
                    <lower-vlan-id>1</lower-vlan-id>
                    <upper-vlan-id>10</upper-vlan-id>
                  </input>
                </range-merge>
              </action>
            </vlan-id-range>
          </vlan-id-ranges>
        </interface>
      </interfaces>
    </top>
  </config>
</edit-config>
</rpc>

```

Suppose we have a trunk interface with vlan tag range [1,10], we delete one vlan tag from this trunk interface, without inline action capability, edit-config with configuration template will be used to split valn tag range into multiple records with extra computation in the client app.


```

xmlns="urn:ietf:params:xml:ns:netconf:base:1.1">
  <edit-config>
    <target>
      <running/>
    </target>
    <default-operation>none</default-operation>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.1">
      <top xmlns="http://example.com/schema/1.2/config">
        <interfaces>
          <interface>
            <name>Ethernet0/0</name>
            <vlan-id-ranges>
              <vlan-id-range xc:operation="delete">
                <group-id>0</group-id>
                <lower-vlan-id>4</lower-vlan-id>
                <upper-vlan-id>4</upper-vlan-id>
              </vlan-id-range>
              <vlan-id-range xc:operation="create">
                <group-id>0</group-id>
                <lower-vlan-id>1</lower-vlan-id>
                <upper-vlan-id>3</upper-vlan-id>
              </vlan-id-range>
              <vlan-id-range xc:operation="create">
                <group-id>1</group-id>
                <lower-vlan-id>4</lower-vlan-id>
                <upper-vlan-id>10</upper-vlan-id>
              </vlan-id-range>
            </vlan-id-ranges>
          </interface>
        </interfaces>
      </top>
    </config>
  </edit-config>
</rpc>

```

If inline-action capability supported, bulk operation in one protocol message will be used to split one record with vlan tag range into two records:

```
<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.1">
  <edit-config>
    <target>
      <running/>
    </target>
    <default-operation>none</default-operation>
    <config xmlns:xc="urn:ietf:params:xml:ns:netconf:base:1.1">
      <top xmlns="http://example.com/schema/1.2/config">
        <interfaces>
          <interface xc:operation="record-merge">
            <name>Ethernet0/0</name>
            <vlan-id-ranges>
              <vlan-id-range>
                <lower-vlan-id>1</lower-vlan-id>
                <upper-vlan-id>10</upper-vlan-id>
                <action xmlns="http://example.com/schema/1.2/config">
                  <range-split>
                    <input>
                      <lower-vlan-id>4</lower-vlan-id>
                      <upper-vlan-id>4</upper-vlan-id>
                    </input>
                  </range-split>
                </action>
              </vlan-id-range>
            </vlan-id-ranges>
          </interface>
        </interfaces>
      </top>
    </config>
  </edit-config>
</rpc>
```

[3.](#) Security Considerations

This document does not introduce any security vulnerability besides on defined in [\[RFC6241\]](#).

[4.](#) IANA Considerations

[4.1.](#) NETCONF Capability URN

IANA has created and now maintains a registry "Network Configuration Protocol (NETCONF) Capability URNs" that allocates NETCONF capability identifiers. Additions to the registry require IETF Standards Action.

IANA has added the following capabilities to the registry:

Internet-Draft

Inline Action Capability

November 2018

Index

Capability Identifier

:inline-action:1.1

urn:ietf:params:netconf:capability:inline-action:1.1

5. Normative References

[I-D.ietf-netconf-nmda-netconf]

Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "NETCONF Extensions to Support the Network Management Datastore Architecture", [draft-ietf-netconf-nmda-netconf-08](#) (work in progress), October 2018.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", [RFC 6020](#), DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.

[RFC6021] Schoenwaelder, J., Ed., "Common YANG Data Types", [RFC 6021](#), DOI 10.17487/RFC6021, October 2010, <<https://www.rfc-editor.org/info/rfc6021>>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", [RFC 6241](#), DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", [RFC 6242](#), DOI 10.17487/RFC6242, June 2011, <<https://www.rfc-editor.org/info/rfc6242>>.

[RFC6470] Bierman, A., "Network Configuration Protocol (NETCONF) Base Notifications", [RFC 6470](#), DOI 10.17487/RFC6470,

February 2012, <<https://www.rfc-editor.org/info/rfc6470>>.

Authors' Addresses

Zheng & Wu

Expires May 8, 2019

[Page 10]

Internet-Draft

Inline Action Capability

November 2018

Walker Zheng
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: zhengguangying@huawei.com

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

Email: bill.wu@huawei.com

