

Netconf
Internet-Draft
Intended status: Standards Track
Expires: November 20, 2007

B. Lengyel
Ericsson Hungary
M. Bjorklund
Tail-f Systems
May 19, 2007

Partial Lock RPC for Netconf
draft-lengyel-ngo-partial-lock-00

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of 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/ietf/1id-abstracts.txt>.

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

This Internet-Draft will expire on November 20, 2007.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

The NETCONF protocol describes the lock and unlock operations that (un)lock an entire configuration datastore. In some situations it is needed to lock only a part of a configuration datastore. This draft proposes a capability based extension to the NETCONF protocol to allow this.

Table of Contents

1.	Introduction	3
1.1.	Definition of Terms	3
2.	Partial Locking Capability	3
2.1.	Overview	3
2.2.	Dependencies	3
2.3.	Capability Identifier	4
2.4.	New Operations	4
2.4.1.	<partial-lock>	4
2.4.2.	<partial-unlock>	6
2.5.	Modifications to Existing Operations	7
2.6.	Interactions with Other Capabilities	7
2.6.1.	Writable-Running Capability	7
2.6.2.	Candidate Configuration Capability	7
2.6.3.	Distinct Startup Capability	7
3.	Security Considerations	7
4.	IANA Considerations	7
5.	Change Log	7
5.1.	TODO	8
6.	XML Schema for Partial Locking	8
7.	Acknowledgements	9
8.	Normative References	9
	Authors' Addresses	9
	Intellectual Property and Copyright Statements	10

1. Introduction

The NETCONF protocol describes the lock and unlock operations that (un)lock an entire configuration datastore. Often it is needed to allow multiple management sessions to modify the configuration of a managed device in parallel. In these cases it is needed to be able to lock only a part of a configuration datastore. This draft proposes an extension to the NETCONF protocol to allow this.

The mechanism for partial locking will be based on the existing XPath filtering mechanisms.

Partial locking will be introduced as a capability into NETCONF.

1.1. Definition of Terms

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](#).

2. Partial Locking Capability

2.1. Overview

The partial locking capability indicates that the device supports the locking of it's configuration with a scope smaller than a complete configuration datastore.

The system will ensure that locked configuration resources will not be modified by other NETCONF or non-NETCONF management operations such as SNMP and CLI.

The duration of the partial-lock is defined as beginning when the partial-lock is granted and lasting until either the corresponding partial-unlock operation succeeds or the NETCONF session terminates.

A NETCONF session MAY have multiple parts of one or more datastores locked using partial lock operations. The <partial-lock> operation returns a lock-id to identify each successfully acquired lock.

2.2. Dependencies

If XPath Capability is also supported, the filter expression might include full XPath 1.0 expressions. Otherwise only restricted XPath is available.

2.3. Capability Identifier

urn:ietf:params:netconf:capability:partial-lock:1.0

2.4. New Operations

2.4.1. <partial-lock>

The partial-lock operation allows the client to lock a portion of a data store. The portion to lock is specified by using an XPath filter parameter in the partial-lock operation. The XPath expression MUST always result in a non-empty nodeset. If any node is locked all its child nodes i.e. the subtree under that node will also be locked.

The XPath filter expression is evaluated only once at locking. If later the configuration data is altered in a way so that the original XPath filter expression would evaluate to a different set of nodes, this does not affect the scope of the partial lock. There is still a way the scope of the lock can change: if we add new nodes under a locked subtree the scope of that partial lock will include the newly created nodes. Similarly if we delete locked nodes, they will not be part of the locked section any more.

If a node is locked by a session only that same session will be able to create new nodes under the locked node, as the lock is always valid for a subtree under any locked node.

If a top level node of a locked subtree is deleted, any other session can recreate it, as it is not covered by the lock any more.

A partial lock MUST fail if:

- o Any part of the scope to be locked is already locked by another management session/protocol including other NETCONF sessions using the [[RFC4741](#)] (global) lock operation or partial-lock or any other non-NETCONF management method.
- o The locking user does not have at least some basic access rights, e.g. read rights, to the whole of the datastore section to be locked. The exact handling of access rights is outside the scope of this document, but it is assumed that there is an access control system that MAY deny or allow the partial lock operation.

Parameters:

target: Name of the configuration datastore part of which will be locked. URIs are not accepted.

filter: The filter element contains a 'select' attribute, which contains an XPath expression. The XPath expression is evaluated in a context where the context node is the root node, and the set of namespace declarations are those in scope on the filter element.

The filter MUST return a non-empty node set.

If the device does not support the XPath Capability beside partial locking, the XPath expression MUST be limited to an absolute path expression, which contains only elements and/or attributes defined as keys to distinguish multiple instances.

Example: Lock the interface with ifIndex 2

```
<nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
  xmlns="urn:ietf:params:netconf:capability:partial-lock:1.0"
  xmlns:if="http://example.com/ns/interface">
  nc:message-id="135">
  <partial-lock>
    <nc:running/>
    <filter select="/if:interface[if:ifIndex='2']"/>
  </partial-lock>
</nc:rpc>

<nc:rpc-reply xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
  xmlns="urn:ietf:params:netconf:capability:partial-lock:1.0"
  nc:message-id="135">
  <nc:data>
    <lock-id>127</lock-id>
  </nc:data>
</nc:rpc-reply>
```

Positive Response:

If the device was able to satisfy the request, an <rpc-reply> is sent that includes a <lock-id> element in the <data> element with the lock identifier.

Negative Response:

If a lock is already held on any node within the subtrees to be locked, the <error-tag> element will be 'lock-denied' and the <error-

info> element will include the <session-id> of the lock owner. If the lock is held by a non-NETCONF entity, a <session-id> of 0 (zero) is included.

If the filter returns an empty node set, the <error-tag> will be 'operation-failed', and the <error-app-tag> will be 'no-matches'.

If the filter returns anything but a node set, the <error-tag> will be 'invalid-value'.

If the XPath capability is not supported and the XPath expression does not conform to the specified limitations, the <error-tag> will be 'invalid-value'.

If access control denies the partial lock, the <error-tag> will be 'access-denied'.

[2.4.2.](#) <partial-unlock>

The operation unlocks a part of a datastore that was previously locked using <partial-lock> during the same session.

Parameters:

lock-id: Lock identifier to unlock; from a reply to a previous <partial-lock> operation.

Example: Unlock

```
<nc:rpc xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
  xmlns="urn:ietf:params:netconf:capability:partial-lock:1.0"
  nc:message-id="136">
  <partial-unlock>
    <lock-id>127</lock-id>
  </partial-unlock>
</nc:rpc>
```

Positive Response:

If the device was able to satisfy the request, an <rpc-reply> is sent that contains an <ok> element. A positive response SHOULD be given even if the whole of the locked part of the datastore is already deleted.

Negative Response:

If the lock-id parameter does not identify a lock which is owned by the session, an 'invalid-value' error is returned.

2.5. Modifications to Existing Operations

None.

2.6. Interactions with Other Capabilities

2.6.1. Writable-Running Capability

Parts of the running datastore can only be locked if the Writable-Running Capability is supported by the device.

2.6.2. Candidate Configuration Capability

Parts of the candidate datastore can only be locked if the Candidate Configuration Capability is supported by the device. The partial locking of the candidate datastore does not depend on whether the datastore was modified or not.

2.6.3. Distinct Startup Capability

Parts of the startup datastore can only be locked if the Distinct Startup Capability is supported by the device.

3. Security Considerations

The same considerations as for the base NETCONF Protocol [[RFC4741](#)] are valid. It is assumed that the <partial-lock> and <partial-unlock> RPCs are only allowed for an authenticated user after passing some access control mechanism.

4. IANA Considerations

The capability's URI should be registered by IANA.

5. Change Log

[draft-00](#) Initial version

5.1. TODO

Do we get an error if part of the to-be-locked section does not exist? Propose no. The non-existing parts will be silently ignored. This is not an existence test.

6. XML Schema for Partial Locking

The following XML Schema defines the <partial-lock> and <partial-unlock> operations:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns="urn:ietf:params:netconf:capability:partial-lock:1.0"
  xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0"
  targetNamespace="urn:ietf:params:netconf:capability:partial-lock:1.0"
  elementFormDefault="qualified" attributeFormDefault="unqualified">

  <xs:annotation>
    <xs:documentation>
      Schema defining the partial-lock and unlock operations.
    </xs:documentation>
  </xs:annotation>

  <xs:import namespace="urn:ietf:params:xml:ns:netconf:base:1.0"
    schemaLocation="urn:ietf:params:xml:ns:netconf:base:1.0"/>

  <xs:complexType name="filterType">
    <xs:complexContent>
      <xs:extension base="xs:anyType">
        <xs:attribute name="select" use="required"/>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="partialLockType">
    <xs:complexContent>
      <xs:extension base="nc:rpcOperationType">
        <xs:sequence>
          <xs:element ref="nc:config-name"/>
          <xs:element name="filter" type="filterType"/>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>

  <xs:complexType name="partialUnLockType">
```



```
<xs:complexContent>
  <xs:extension base="nc:rpcOperationType">
    <xs:sequence>
      <xs:element name="lock-id" type="xs:unsignedInt"/>
    </xs:sequence>
  </xs:extension>
</xs:complexContent>
</xs:complexType>

<!-- <partial-lock> operation -->
<xs:element name="partial-lock" type="partialLockType"
  substitutionGroup="nc:rpcOperation"/>

<!-- <partial-unlock> operation -->
<xs:element name="partial-unlock" type="partialUnLockType"
  substitutionGroup="nc:rpcOperation"/>

<!-- reply to <partial-lock> -->
<xs:element name="lock-id" type="xs:unsignedInt"/>

</xs:schema>
```

7. Acknowledgements

Thanks to Andy Bierman for providing important input to this document.

8. Normative References

[RFC4741] "NETCONF Configuration Protocol", [RFC 4741](#), December 2006.

Authors' Addresses

Balazs Lengyel
Ericsson Hungary

Email: balazs.lengyel@ericsson.com

Martin Bjorklund
Tail-f Systems

Email: mbj@tail-f.com

Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

