```
Network Working Group                                         T. Mizrahi
Internet Draft                                                  Y. Moses
Intended status: Experimental   Technion, Israel Institute of Technology
Expires: January 2014                                       July 8, 2013
```

Time Capability in NETCONF
draft-mm-netconf-time-capability-00.txt

Abstract

This document defines a capability-based extension to the Network Configuration Protocol (NETCONF) that allows time-triggered configuration and management operations. This extension allows NETCONF clients to invoke configuration updates according to scheduled times, and allows NETCONF servers to attach timestamps to the data they send to NETCONF clients.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction

The Network Configuration Protocol (NETCONF) defined in [RFC6241] provides mechanisms to install, manipulate, and delete the configuration of network devices. NETCONF allows clients to configure and monitor NETCONF servers using remote procedure calls (RPC).

NETCONF, as defined in [RFC6241], is asynchronous; when a client invokes an RPC, it has no control over the time at which the RPC is executed, nor does it have any feedback from the server about the execution time.

Time-based configuration ([HotSDN], [TimeTR]) can be a useful tool that enables an entire class of coordinated and scheduled configuration procedures. Time-triggered configuration allows coordinated network updates in multiple devices; a client can invoke a coordinated configuration change by sending RPCs to multiple servers with the same scheduled execution time. A client can also invoke a time-based sequence of updates by sending n RPCs with n different update times, T1, T2, ..., Tn, determining the order in which the RPCs are executed.

This memo defines the time capability in NETCONF. This extension allows clients to determine the scheduled execution time of RPCs they send. It also allows a server that receives an RPC to report its actual execution time to the client.

2. Conventions used in this document

2.1. Keywords

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 [RFC2199].

2.2. Abbreviations

NETCONF	Network Configuration Protocol
RPC	Remote Procedure Call
TAI	International Atomic Time

2.3. Terminology

o Capability [RFC6142]: A functionality that supplements the base NETCONF specification.

o Client [RFC6142]: Invokes protocol operations on a server. In addition, a client can subscribe to receive notifications from a server.

- Execution time: The execution time of an RPC is defined as the time at which a server completes the execution of an RPC.

- Scheduled time: The scheduled time of an RPC is the time at which the RPC should be completed. The scheduled time is determined by the client, and enforced by the server.

- Server [RFC6142]: Executes protocol operations invoked by a client. In addition, a server can send notifications to a client.

3. Using Time in NETCONF

3.1. The Time Capability in a Nutshell

The :time capability provides two main functions:

- Scheduling:
  When a client sends an RPC to a server, the RPC message MAY include a scheduled time, Ts (see Figure 1). The server then executes the RPC at the scheduled time Ts, and once completed the server can respond with an RPC reply message.

- Reporting:
  When a client sends an RPC to a server, the RPC message MAY include a get-time element (see Figure 2), requesting the server to return the execution time of the RPC. In this case, after the server performs the RPC it responds with an RPC reply that includes the execution time, Te.

```
                      RPC _________
                    executed       \
                                   \/
                                   Ts
            server  ---------------+--------------       ----> time
                              /\      \
                          rpc /        \ rpc-reply
                         (Ts)/          \
                            /           \/
            client  -----------------------------
```

Figure 1 Scheduled RPC

```
                   RPC _________
                 executed       \
                                \/
                                Te
            server  ------------+----------------        ----> time
                              /\   \
                       rpc    /     \ rpc-reply
                   (get-time)/       \ (Te)
                            /        \/
            client  -----------------------------
```

Figure 2 Reporting the Execution Time of an RPC

The two scenarios discussed above imply that a third scenario can also be supported (Figure 3), where the client invokes an RPC that includes a scheduled time, Ts, as well as the get-time element. This allows the client to receive feedback about the actual execution time, Te. Ideally, Ts=Te. However, the server may execute the RPC at a slightly different time than Ts, for example if the server is tied up with other tasks at Ts.

```
                      RPC _________
                    executed       \
                                   \/
                                Ts Te
            server  -------------+-+-------------        ----> time
                            /\        \
                   rpc      /          \ rpc-reply
            (Ts + get-time)/            \ (Te)
                          /             \/
```

```
         client  -----------------------------
```

Figure 3 Scheduling and Reporting

3.2. Synchronization Aspects

The time capability defined in this document requires clients and servers to maintain clocks. It is assumed that clocks are synchronized by a method that is outside the scope of this document.

This document does not define any requirements pertaining to the degree of accuracy of performing scheduled RPCs. Note that two factors affect how accurately the server can perform a scheduled RPC; one factor is the accuracy of the clock synchronization method used

to synchronize the clients and servers, and the second factor is the server's ability to execute real-time configuration changes, which greatly depends on how it is implemented. Typical networking devices are implemented by a combination of hardware and software. While the execution time of a hardware module can typically be predicted with a high level of accuracy, the execution time of a software module may be variable and hard to predict. A configuration update would typically require the server's software to be involved, thus affecting how accurately the RPC can be scheduled.

Since servers do not perform configuration changes instantaneously, the processing time of an RPC should not be overlooked. The scheduled time and execution time always refer to the completion time of the RPC.

3.3. Time Format

The scheduled time and execution time fields in RPC messages use a common time format field.

The time format defined in this document is similar to the one defined in [IEEE1588].

Time is represented as follows:

```
  grouping time-parameters {
    description
    "Contains the parameters of the time element.";
```

```
     leaf seconds {
       description
       "The seconds portion of the time element.";
       type uint64;
     }
     leaf nanoseconds {
       description
       "The nanoseconds portion of the time element.";
       type uint32;
     }
   }
```

The time-parameters grouping consists of two sub-fields; a seconds field, representing the integer portion of time in seconds, and a nanoseconds field, representing the fractional portion of time in nanoseconds.

Time is measured according to the International Atomic Time (TAI) timescale. The epoch is defined as 1 January 1970 00:00:00 TAI.

4. Time Capability

The structure of this section is as defined in Appendix D of [RFC6241].

4.1. Overview

A server that supports the time capability can perform time-triggered operations as defined in this document.

A server implementing the :time capability:

o MUST support the ability to receive <rpc> messages that include a time element, and perform a time-triggered operation accordingly.

o MUST support the ability to include a time element in the <rpc-reply> messages that it transmits.

4.2. Dependencies

None.

4.3. Capability Identifier

The :time capability is identified by the following capability string (to be assigned by IANA - see Section 7.):

```
urn:ietf:params:netconf:capability:time:1.0
```

4.4. New Operations

None.

4.5. Modifications to Existing Operations

Three new elements are added to all existing operations:

- o <scheduled-time>
  This element is added to the input of each operation, indicating the time at which the server is scheduled to complete the operation. Every <rpc> message MAY include the <scheduled-time> element. A server that supports the :time capability and receives an <rpc> message with a <scheduled-time> element MUST perform the operation at the scheduled time.

- o <get-time>
  This element is added to the input of each operation. An <rpc> message MAY include a <get-time> element, indicating that the server MUST include an <execution-time> in its corresponding <rpc-reply>.

- o <execution-time>
  This element is added to the output of each operation, indicating the time at which the server completed the operation. An <rpc-reply> MAY include the <execution-time> element. A server that supports the :time capability and receives an operation with the <get-time> element MUST include the execution time in its response.

4.6. Interactions with Other Capabilities

Confirmed Commit Capability

The confirmed commit capability is defined in Section 8.4 of [RFC6241]. According to [RFC6241], a confirmed <commit> operation

MUST be reverted if a confirming commit is not issued within the timeout period (which by default is 600 seconds).

When the time capability is supported, and a confirmed <commit> operation is used with the <scheduled-time> element, the confirmation timeout MUST be counted from the scheduled time, i.e., the client begins the timeout measurement starting at the scheduled time.

5. Examples

5.1. <scheduled-time> Example

The following example extends the example presented in Section 7.2 of [RFC6241] by adding the time capability. In this example, the <scheduled-time> element is used to specify the scheduled execution time of the configuration update (as shown in Figure 1).

```
<rpc message-id="101"
    xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <edit-config>
    <target>
      <running/>
    </target>
    <execution-time
     xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-time">
```

```
      <seconds>1234567890</seconds>
      <nanoseconds>500000000</nanoseconds>
    </execution-time>
    <config>
      <top xmlns="http://example.com/schema/1.2/config">
        <interface>
          <name>Ethernet0/0</name>
          <mtu>1500</mtu>
        </interface>
      </top>
    </config>
  </edit-config>
</rpc>

<rpc-reply message-id="101"
     xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <ok/>
</rpc-reply>
```

5.2. <get-time> Example

The following example is similar to the one presented in Section 5.1. , except that in this example the client includes a <get-time> element in its RPC, and the server consequently responds with an <execution-time> element (as shown in Figure 2).

```
   <rpc message-id="101"
       xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
     <edit-config>
       <target>
         <running/>
       </target>
       <get-time
        xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-time">
       </get-time>
       <config>
         <top xmlns="http://example.com/schema/1.2/config">
           <interface>
             <name>Ethernet0/0</name>
             <mtu>1500</mtu>
           </interface>
```

```
         </top>
       </config>
     </edit-config>
   </rpc>

   <rpc-reply message-id="101"
        xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
     <ok/>
     <execution-time>
       <seconds>1234567890</seconds>
       <nanoseconds>500000000</nanoseconds>
     </execution-time>
   </rpc-reply>
```

6. Security Considerations

The security considerations of the NETCONF protocol in general are discussed in [RFC6241].

The usage of the time capability defined in this document can assist an attacker in gathering information about the system, such as the exact time of future configuration changes. Moreover, the time elements can potentially allow an attacker to learn information about the system's performance. Furthermore, an attacker that sends malicious RPC messages can use the time capability to amplify her attack; for example, by sending multiple RPC messages with the same scheduled time. It is important to note that the security measures described in [RFC6241] can prevent these vulnerabilities.

The time capability relies on an underlying time synchronization protocol. Thus, an attack against the time protocol can potentially compromise NETCONF when using the time capability. A detailed discussion about the threats against time protocols and how to mitigate them is presented in [TimeSec].

7. IANA Considerations

This document proposes to register the following capability identifier URN in the 'Network Configuration Protocol (NETCONF) Capability URNs' registry:

```
urn:ietf:params:netconf:capability:time:1.0
```

This document proposes to register the following XML namespace URN in the 'IETF XML registry', following the format defined in [RFC3688]:

```
URI: urn:ietf:params:xml:ns:yang:ietf-netconf-time
```

8. Acknowledgments

This work was supported in part by Israel Science Foundation grant ISF 1520/11.

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

9. References

9.1. Normative References

[RFC2199] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., Bierman, A., Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, June 2011.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

9.2. Informative References

[HotSDN] Mizrahi, T., Moses, Y., "Time-based Updates in Software Defined Networks", the second workshop on hot topics in software defined networks (HotSDN), to appear, 2013.

[TimeTR] Mizrahi, T., Moses, Y., "Time-based Updates in OpenFlow: A Proposed Extension to the OpenFlow Protocol", Technion - Israel Institute of Technology, technical report, CCIT Report #835, EE Pub No. 1792, 2013. http://tx.technion.ac.il/~dew/OFTimeTR.pdf

[IEEE1588] IEEE TC 9 Instrumentation and Measurement Society, "1588 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Version 2", IEEE Standard, 2008.

[TimeSec] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", draft-ietf-tictoc-security-requirements (work in progress), April 2013.

Authors' Addresses

Tal Mizrahi
7/43 Gotl Levin st.
Haifa, 3292207, Israel

Email: dew@tx.technion.ac.il

Yoram Moses
Department of Electrical Engineering
Technion - Israel Institute of Technology
Technion City, Haifa, 32000, Israel

Email: moses@ee.technion.ac.il