Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 19, 2013

J. Schoenwaelder
Jacobs University Bremen

February 15, 2013

# Considerations on using NETCONF with LMAP Measurement Agents
draft-schoenw-lmap-netconf-00.txt

Abstract

This document discusses how the NETCONF protocol can be used to configure LMAP measurement agents.

Table of Contents

1. Introduction . . . . . . . . . . . . . . . . . . . . . . . . . 3
2. Properties of Measurement Agents . . . . . . . . . . . . . . . . 3
3. Properties of the NETCONF Protocol . . . . . . . . . . . . . . . 3
4. Discussion . . . . . . . . . . . . . . . . . . . . . . . . . . 4
4.1. Connection Initiation . . . . . . . . . . . . . . . . . . . . 4
4.2. Client and Server Role . . . . . . . . . . . . . . . . . . . 5
4.3. Identification of Configuration Versions . . . . . . . . . . 5
4.4. Pushing of Measurement Results . . . . . . . . . . . . . . . 5
4.5. NETCONF versus YANG-API . . . . . . . . . . . . . . . . . . . 6
5. Security Considerations . . . . . . . . . . . . . . . . . . . . 6
6. IANA Considerations . . . . . . . . . . . . . . . . . . . . . . 7
7. Acknowledgements . . . . . . . . . . . . . . . . . . . . . . . 7
8. Informative References . . . . . . . . . . . . . . . . . . . . 7
Author’s Address . . . . . . . . . . . . . . . . . . . . . . . . . 8

# 1. Introduction

This document discusses how the NETCONF protocol [RFC6241] can be used to configure Large-Scale Measurement of Broadband Performance (LMAP) measurement agents (MAs), sometimes also called measurement clients [I-D.schulzrinne-lmap-requirements].

MAs may be deployed as separate hardware devices or as functions embedded in consumer electronic devices and home routers or as pure software solutions that can be installed on off-the-shelf computing equipment. Measurement agents receive instructions from a controller when and how to conduct what measurements (the measurement schedule) and how and when to report measurement results to a data collector (the report schedule). Further information about the interaction between MAs and controllers and collectors can be found in [I-D.schulzrinne-lmap-requirements].

# 2. Properties of Measurement Agents

Measurement Agents (MAs) have a number of important properties:

1. MAs are often deployed behind Network Address Translators (NATs). This might even be true if MAs are part of a device on the demarcation line between a service provider and a home network due to the usage of Carried Grade NATs in the service provider network.

2. MAs may run on devices that are not always powered up and online.

3. A single controller may be responsible for a large number of MAs.

4. A large fraction of the MAs may be inactive (i.e., they do not perform any measurements) at any given point in time. Inactive MAs may need to be enabled on demand for example to troubleshoot specific problems (e.g., as part of customer helpdesk services) or to balance measurement traffic load.

# 3. Properties of the NETCONF Protocol

The Network Configuration Protocol (NETCONF) [RFC6241] provides mechanisms to install, manipulate, and delete the configuration of network devices. It uses an Extensible Markup Language (XML)-based data encoding for the configuration data as well as the protocol messages. The NETCONF protocol operations are realized as remote procedure calls (RPCs).

The NETCONF protocol runs by default over the Secure Shell protocol (SSH) [RFC6242] but it can also be used over Transport Layer Security (TLS) [RFC5539] with pairwise authentication using X.509 certificates.

NETCONF has been originally designed to be used on network devices such as backbone routers. A device supporting NETCONF has an embedded NETCONF server. Configuration management applications use embedded NETCONF clients to connect to NETCONF servers and then issue RPC calls to manipulate the configuration state of the devices.

## 4. Discussion

This section discusses certain technical challenges related to the usage of NETCONF to configure MAs.

### 4.1. Connection Initiation

Due to the nature of LMAP MAs (likely located behind NATs), it is crucial that MAs initiate connections to a controller. This is currently not supported in NETCONF. There were previous attempts to provide a so called call-home mechanism for NETCONF, see for example [I-D.kwatsen-reverse-ssh]. The current state of the art, however, is that there is no standardized call home mechanism for NETCONF over SSH.

For the NETCONF over TLS transport [RFC5539], which relies on mutual authentication using X.509 certificates, it seems easier to support call home. In fact, the NETCONF over TLS transport specification is currently being updated and hence there is an opportunity to get call home support into this update on relative short notice. The work essentially requires to detail aspects such as port numbers used to connect form a NETCONF server (acting as a TCP client) to a NETCONF client (acting as a TCP server). In addition, a YANG data model would be desirable that can be used to configure the call home policy (when to call home) and the addresses to connect to and perhaps the certificate to use.

Decoupling the NETCONF server / client role from the TCP server / client role turns out to be straight-forward since every NETCONF session starts with an exchange of <hello> messages. The <hello> message sent by the NETCONF server includes a session-id while the NETCONF client does not send a session-id. As such, both endpoints can easily identify and verify who is acting as NETCONF client and NETCONF server. [RFC6241] already handles the possible error cases (i.e., a NETCONF server connecting to a NETCONF server or a NETCONF client connecting to a NETCONF client).

## 4.2. Client and Server Role

Some large scale measurement deployments use proprietary protocols where the server role is on the controller. In a nutshell, the MA connects to the controller running a server and checks if there is a configuration update to load. If so, the MA fetches the necessary new configuration information and then applies it locally.

The NETCONF protocol, however, assumes that the NETCONF server role is taken by the device that is configured. This would be in the LMAP use case the MA while the NETCONF client would be running on the controller.

Even though this may seem like a major difference in the way the interaction works, it appears that NETCONF can provide the functionality needed. A MA initiating a transport connection and subsequently taking the NETCONF server role enables the controller (acting as a NETCONF client) after the <hello> exchange to take the initiative to determine whether any configuration changes need to be applied to the device. If so, standard <edit-config> operations can be used to modify the device’s configuration.

## 4.3. Identification of Configuration Versions

As mentioned above, the controller (running a NETCONF client) must determine whether a device’s configuration needs updates. While this could be achieved by retrieving the configuration using <get-config> and comparing the result with the expected configuration, this approach is not very efficient. It will be much more effective if the NETCONF server would indicate the version of the configuration it is currently using. The version can either be identified by a version number or a time-stamp of the last configuration change or simply an opaque tag that is handed out and interpreted only by the controller. While the configuration version might simply be modeled as a regular data object that the NETCONF client retrieves in the usual way, it might be useful to consider optimizations, e.g., carrying the configuration version as part of a new capability in the <hello> exchange.

## 4.4. Pushing of Measurement Results

NETCONF has not been designed as a data push protocol. While a NETCONF extension [RFC5277] provides support for event notifications, this mechanism requires in its simplest form that a NETCONF client first subscribes to an event stream and that the session used to carry event notification stays open. This is not scalable in the LMAP scenario.

One possible way to work around this limitation within the framework of the current NETCONF protocol is to make use of the event notification replay feature: A MA is locally collecting measurement results. After connecting to a collector (acting as a NETCONF client), the collector subscribes to an event stream with a request to replay the measurement results collected since the last time data has been fetched from the MA. An alternative, of course, would be to model test results as part of an LMAP data model and to use NETCONF <get> operations to retrieve the data.

That said, if close to soft real-time pushing of measurement results form the MA to the collector is required, then NETCONF likely is not the right choice.

### 4.5. NETCONF versus YANG-API

NETCONF provides a feature rich solution for network configuration management, including support for concurrent access to a NETCONF server by multiple NETCONF clients, different configuration datastores, explicit validation of configurations, and a confirmed-commit procedure to support configuration change transactions spanning multiple devices. A recent proposal called YANG-API [I-D.bierman-netconf-yang-api] aims at providing a simplified interface that follows RESTful principles and is compatible with a resource-oriented device abstraction.

While implementations of YANG-API are in progress, it seems too early to decide whether the benefits of RESTful YANG-API are significant enough to consider it as a possible alternative for LMAP. In particular, it might take a few years for YANG-API to become a stable specification.

## 5. Security Considerations

The NETCONF protocol [RFC6241] can run over several different transports. Since the protocol manipulates sensitive configuration information, NETCONF requires that all transports provide authentication, data integrity, confidentiality, and replay protection.

There are currently two transport for NETCONF on the standards track. The NETCONF over SSH transport [RFC6242] provides authentication and data encryption services. The NETCONF over SSH specification further requires that the identity of the SSH server must be verified and authenticated by the SSH client according to local policy before password-based authentication data or any configuration or state data is sent to or received from the SSH server. Similarily, the identity of the SSH client must also be verified and authenticated by the SSH server according to local policy to ensure that the incoming SSH client request is legitimate before any configuration or state data is sent to or received from the SSH client. Neither side should establish a NETCONF over SSH connection with an unknown, unexpected, or incorrect identity on the opposite side.

The NETCONF over TLS transport [RFC5539], currently being revised in [I-D.ietf-netconf-rfc5539bis], provides authentication and data encryption services. In particular, [RFC5539] assumes that both peers authenticate each other using X.509 certificates while [I-D.ietf-netconf-rfc5539bis] adds the possibility to use pre-shared keys.

The NETCONF access control model [RFC6536] provides an authorization model for NETCONF. It allows to configure access control rules that can be used to restrict NETCONF protocol access for particular users to a pre-configured subset of all available NETCONF protocol operations and content. The NETCONF access control model should be required for LMAP implementations that potentially allow access from multiple controllers.

## 6. IANA Considerations

TBD

## 7. Acknowledgements

TBD

## 8. Informative References

[I-D.bierman-netconf-yang-api]
Bierman, A. and M. Bjorklund, "YANG-API Protocol", draft-bierman-netconf-yang-api-01 (work in progress), November 2012.

[I-D.ietf-netconf-rfc5539bis]
Badra, M., Luchuk, A., and J. Schoenwaelder, "NETCONF Over Transport Layer Security (TLS)", draft-ietf-netconf-rfc5539bis-01 (work in progress), October 2012.

[I-D.kwatsen-reverse-ssh]
Watsen, K., "Reverse Secure Shell (Reverse SSH)",

draft-kwatsen-reverse-ssh-01 (work in progress), June 2011.

[I-D.schulzrinne-lmap-requirements]
Schulzrinne, H., Johnston, W., and J. Miller, "Large-Scale Measurement of Broadband Performance: Use Cases, Architecture and Protocol Requirements", draft-schulzrinne-lmap-requirements-00 (work in progress), September 2012.

[RFC5277] Chisholm, S. and H. Trevino, "NETCONF Event Notifications", RFC 5277, July 2008.

[RFC5539] Badra, M., "NETCONF over Transport Layer Security (TLS)", RFC 5539, May 2009.

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

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, June 2011.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, March 2012.

Author's Address

Juergen Schoenwaelder
Jacobs University Bremen

Email: j.schoenwaelder@jacobs-university.de