Using Netconf for Configuring Monitoring Probes

Gerhard Münz*, Albert Antony*, Falko Dressler†*, and Georg Carle*

* Computer Networks and Internet, Wilhelm Schickard Institute for Computer Science, University of Tübingen, Germany
† Autonomic Networking, Department of Computer Science 7, University of Erlangen-Nuremberg, Germany

Abstract— Netconf is a new protocol for configuration and management of network devices, based on a flexible XML-encoded message format. Netconf aims to overcome the shortcomings of SNMP and CLIs that are predominantly used for configuration tasks. We demonstrate that Netconf is highly suitable for the configuration of IPFIX/PSAMP monitoring probes, as required in order to dynamically and remotely adapt to the varying needs of applications that receive and process monitoring data. In this regard, we present an XML-based data model covering all common configurable parameters for flow metering and aggregation, packet sampling, and data export. Finally, we describe how we implemented the Netconf-based configuration approach based on Web Services and SOAP.

Index Terms— network configuration, network monitoring, flow accounting, packet sampling

I. INTRODUCTION

Cisco Netflow [1], IPFIX (IP Flow Information eXport) [2], and PSAMP (Packet SAMPling) [3] define mechanisms and protocols for monitoring network traffic and exporting flow and packet information. Different versions of the Netflow technology have already been successfully introduced into the market. Similar success can be predicted for the upcoming IPFIX standard as it represents the successor of Netflow Version 9. The exported monitoring data can be used for various purposes, e.g. accounting, quality of service (QoS) measurements, and detection of suspicious activities, such as attacks, propagating worms etc.

This paper deals with the configuration of monitoring probes. Depending on the capabilities of the device, the configuration comprises parameters for flow metering and aggregation, packet sampling, and/or the export of monitoring data. It is common practice to set the monitoring parameters using a device-specific command line interface (CLI) or a configuration file. This process, however, is cumbersome and complicated, especially if used in heterogeneous networks consisting of different device models or if frequent reconfigurations of the monitoring functions are performed.

As an amendment, we developed an interface for configuring monitoring probes based on the Netconf protocol [4]. Therefore, we specified a device-independent configuration data model in XML (Extensible Markup Language) covering the common configurable parameters of a monitoring probe. We implemented Netconf using SOAP (Simple Object Access Protocol) as transport protocol and extended the Netconf server with the functionality to configure the open-source IPFIX/PSAMP monitoring probe VERMONT (VERsatile MONitoring Toolkit) [5].
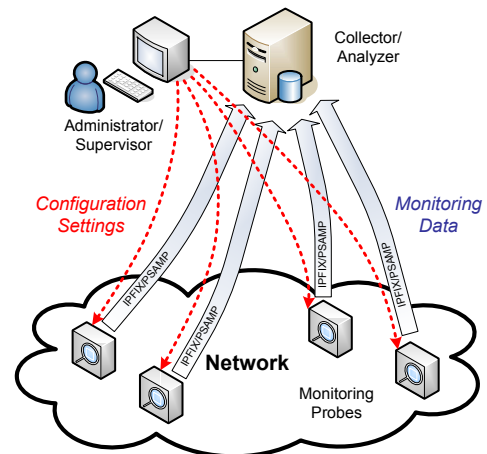

Fig. 1.   Network Monitoring and Analysis

This paper is structured as follows. Section II introduces network monitoring based on flow accounting and packet sampling. Section III outlines the configuration issue and presents our alternative approach of using Netconf and an XML-based data model. Details about the implementation are given in Section IV. Section V sketches the deployment of the monitoring probe configuration in a specific scenario. Related work is presented in Section VI, before we draw some final conclusions in Section VII.

II. NETWORK MONITORING

Network monitoring has become a major research issue in the networking community. One reason is that the available bandwidth grows significantly faster than the processing speed of the monitoring probes. Solutions have been developed that allow reducing the processing requirements for network monitoring and analysis. The primary idea behind these concepts is to split the monitoring and the subsequent analysis into two separate tasks. As shown in Figure 1, monitoring probes observe the network traffic, gather statistics and other kinds of monitoring data, and export them to an analyzer for further processing. Ideally, the exported monitoring data would be well adapted to the requirements and processing capabilities of the analyzer.

Common network monitoring techniques are *flow accounting* and *packet sampling*. Flow accounting stores information and statistics about observed packet flows. According to the definition of the IPFIX working group at the IETF (Internet Engineering Task Force), a flow is defined as a unidirectional

stream of IP packets that are observed at an observation point in the network and that share a set of common properties called the *flow key* [6]. The common way to define a flow key is the IP-five-tuple (protocol type, source IP address, destination IP address, source port, destination port). The exported flow records include the number of octets and the number of packets observed per flow within a specific time interval. However, this may still result in an unmanageably high number of records under certain circumstances, e.g. during distributed denial-of-service (DDoS) attacks with spoofed source addresses. Also, many applications do not require detailed flow-level information but only information about flow aggregates, where the quality and level of flow aggregation is very application-specific. Therefore, flow aggregation mechanisms [7] can be deployed that allow adapting the amount and detailedness of exported flow information to the current needs and available resources of the analyzer.

In contrast to flow accounting, packet sampling, as specified by the IETF PSAMP working group [3], allows exporting specified header fields and parts of the payload of selected packets. The selection of packets is based on filters and samplers. While filters are used for deterministic packet selection based on header field values, samplers probabilistically select packets applying a specific sampling algorithm [8]. Again, the amount and detailedness of exported packet samples can be adapted to the needs of the analyzer by configuring the involved filters, samplers, and exporters accordingly.

III. MONITORING PROBE CONFIGURATION

A. The Configuration Issue

The network monitoring techniques described in Section II are being used by a growing number of applications such as accounting, QoS measurements, and attack detection. Many of these require or at least benefit from the possibility of dynamically adapting the configuration of monitoring probes to changing traffic conditions and the varying needs of the analyzer. Especially the configurable parameters of flow aggregation and packet sampling are subject to frequent changes.

Despite its importance, the configuration of monitoring probes has been out of scope of the IPFIX working group so far. The PSAMP working group is standardizing a MIB (Management Information Base) module [9] covering sampling and filtering parameters. Cisco also specified two MIB modules for Netflow: CISCO-NETFLOW-MIB and CISCO-NDE-MIB. However, only CISCO-NDE-MIB can be used for configuration purposes, and the configuration is limited to the addresses and port numbers of the receiving collectors. In short, it can be said that currently no mechanism exists that would allow configuring monitoring probes in a consistent way.

B. Netconf: An Appropriate Configuration Protocol

We developed a solution for remote configuration of monitoring probes based on the Netconf protocol [4]. With respect to network device configuration, Netconf is an interesting alternative to SNMP (Simple Network Management Protocol).

The main difference from SNMP is that Netconf messages and configuration data are encoded in XML, which has some advantages as compared to binary encoding schemes (as used by SNMP):

- XML is human-readable, which facilitates debugging of erroneous implementations.
- Many standard libraries and tools for XML processing are available.
- Configuration data can be structured in a flexible way.
- Message format and data models can be easily extended.

In order to use Netconf, an XML-based data model for the configuration parameters has to be defined using a description language such as XML Schema or DTD (Document Type Definition).

Netconf defines some useful optional capabilities such as supporting up to three different configurations per device (*startup*, *running*, and *candidate*), validating new configuration settings before committing them, performing a rollback to the previous configuration in case of an error etc. Furthermore, the multi-manager problem[1], that arises if SNMP is used for configuration, has been solved in Netconf by providing a lock mechanism that grants exclusive write access to a single Netconf client.

The Netconf working group specified three different possibilities to implement Netconf based on SSH (Secure SHell) [10], BEEP (Blocks Extensible Exchange Protocol) [11], and SOAP [12]. We decided to implement Netconf over SOAP because SOAP is widely used for Web Services applications. In addition, a large number of tools exist that facilitate the implementation of SOAP-based client-server applications.

C. An XML Data Model for IPFIX and PSAMP

In order to define a configuration data model for IPFIX/PSAMP monitoring probes, we identified sets of configurable parameters for the sampling, metering, aggregation, exporting, and collection processes. The results are summarized in Table I. In contrast to [6], we assigned the definition of templates to the sampling and metering processes and not to the exporting process. This is because an exporting process may transmit data from different metering and/or sampling processes using different templates. In case of aggregation, the template is implicitly defined by the aggregation rules [7]. The active and inactive flow timeout of the metering process define the period of time after which the record of an active or inactive flow is exported. The export timeout of the exporting process defines the maximum time the exporting process waits until sending an IPFIX packet (if data is available). The template refresh intervals and template timeout are related to the usage of UDP as transport protocol, where templates have to be sent periodically. Depending on the capabilities of the device, there may be additional parameters not mentioned in the table.

---

[1]SNMP does not provide any mechanism that resolves conflicts in case multiple NMSs (Network Management Stations) try to access and change MIB entries simultaneously. This is called the *multi-manager problem*.

TABLE I

IPFIX AND PSAMP CONFIGURABLE PARAMETERS

Process	Parameters
Packet Capturing	- list of capturing interfaces
Packet Sampling	- filtering and sampling parameters,
	- template definition
Flow Metering	- active and inactive flow timeout,
	- template definition
Flow Aggregation	- set of aggregation rules (see [7])
Export	- list of recipients (IP address, port number, transport protocol),
	- export timeout,
	- template refresh intervals
Collection	- listening interface, port, transport protocol,
	- template timeout

Based on the parameters listed in Table I, we specified a device-independent data model in XML Schema that can be found in [13] and [14]. Figure 2 shows a sample configuration for a packet sampler, which defines the capturing interfaces, a filter and a sampler, followed by the template and exporter properties.

IV. IMPLEMENTATION

We implemented Netconf over SOAP with the help of the gSOAP Web Services Toolkit (version 2.7.2) [15], [16] which provides an open-source SOAP implementation in C/C++. gSOAP generates a very compact code that already includes an XML parser and an HTTP stack, and does not depend on any third party libraries. Furthermore, gSOAP is said to be fast and interoperable with other SOAP implementations. We added authentication and encryption capabilities using OpenSSL [17], which is supported by gSOAP.

gSOAP provides a code-generator that generates skeleton codes for the SOAP client and server, based on a given WSDL (Web Services Description Language) file. However, we encountered many unexpected problems when applying it to the WSDL and XML specifications of the Netconf protocol included in [12]. These problems were mainly related to faults in gSOAP, but also partly provoked by the convoluted XML Schema definition of the Netconf messages in [4] making abundant use of abstract types and inheritance. We got around these problems by rewriting the Schema in a simplified way without altering the resulting message format, such that gSOAP could handle it correctly.

Based on the gSOAP-generated skeleton code, we implemented full-fledged Netconf services including the optional capabilities *candidate configuration*, *rollback on error*, *validate*, and *distinct startup* configuration. For the time being, we have not implement support of filters, URLs, and the *confirmed-commit* operation as we currently do not need them.

Finally, we implemented functions that convert the device-independent configuration settings from the XML data model into configuration files of the open-source IPFIX/PSAMP monitoring probe VERMONT. This is shown in Figure 3.

```xml
<monitorConfig>
  <sampler Id="1" operation="create">
    <interface Id="1">eth0</interface>
    <interface Id="2">eth1</interface>
    <packetProcessor Id="1">
      <ipFilter>
        <dstAddress>10.0.2.66</dstAddress>
      </ipFilter>
    </packetProcessor>
    <packetProcessor Id="2">
      <randOutOfN>
        <population>5</population>
        <size>3</size>
      </randOutOfN>
    </packetProcessor>
    <template>
      <templateId>1025</templateId>
      <field>
        <name>sourceIPv4Address</name>
      </field>
      <field>
        <name>sourceTransportPort</name>
      </field>
      <field>
        <name>destinationIPv4Address</name>
      </field>
      <field>
        <name>destinationTransportPort</name>
      </field>
    </template>
    <exporter>
      <sourceId>4712</sourceId>
      <exportTimeout>500</exportTimeout>
      <exportTo Id="1">
        <address>10.0.2.5</address>
        <port>1200</port>
        <protocol>udp</protocol>
      </exportTo>
    </exporter>
  </sampler>
</monitorConfig>
```
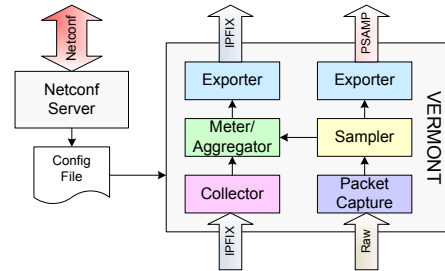
Fig. 2. Sampler Configuration

Fig. 3. Configurable Monitoring Probe VERMONT

VERMONT captures raw packets, performs flow accounting, flow aggregation and packet sampling, and exports the resulting monitoring data using the IPFIX/PSAMP protocol. VERMONT can also operate as a concentrator that receives and aggregates IPFIX data exported by other monitoring probes.

The Netconf server runs as a separate process that receives remote procedure calls (RPCs) from one or more Netconf clients. VERMONT runs as a child process of the Netconf server, which makes recovery possible if VERMONT terminates because of an error. If a reconfiguration fails, a rollback is performed and the previous working configuration is restored. Furthermore, a Netconf error message is returned

to the Netconf client. A major disadvantage of the current implementation is that every reconfiguration requires a stop and restart of VERMONT, i.e. for a short period of time monitoring is disabled completely. This problem can be solved by enhancing VERMONT with capabilities for dynamic reconfiguration at runtime.

V. DEPLOYMENT

We deploy the presented Netconf-based configuration within the European project DIADEM Firewall [18]. In this context, adaptive monitoring probes are used to deliver IPFIX and PSAMP data for anomaly and attack detection purposes. The reconfiguration of monitoring probes is necessary to adapt exported monitoring data to the varying needs of the detection system. For example, flow aggregates and some randomly sampled packets might be analyzed as long as no anomalous or suspicious behavior is detected. If there are hints that an attack is underway, the monitoring configuration is changed in order to get more detailed information about the traffic directed to the potential victim(s).

VI. RELATED WORK

In [19], Choi et al. present an XML-based configuration management system (XCMS) that uses a slightly modified version of the Netconf protocol for configuring an IP sharing device. Like us, they chose SOAP as transport protocol for Netconf and made use of gSOAP for their implementation.

In [20], Schönwälder et al. give an excellent overview on the evolution of network management and identify a general trend towards XML-based solutions, especially for configuration tasks. The authors of [21] show how Web Services can be appropriately deployed for network management.

Several methods and tools have been developed that translate MIB modules into XML Schema definitions, or MIB data into XML data. This aims at facilitating the use of new XML technologies in combination with legacy devices supporting only SNMP (see [22] and the references therein). Another work evaluated the performance of management based on Web Services compared to SNMP [23].

VII. CONCLUSION

In this paper, we presented Netconf as an appropriate protocol for the remote configuration of monitoring probes. We introduced a device-independent configuration data model in XML covering all common configurable parameters for flow metering and aggregation, packet sampling, and data export and collection. We described how we implemented the Netconf protocol with the help of the gSOAP Web Services Toolkit. Moreover, we showed how the Netconf server was extended to control the configuration of the IPFIX/PSAMP monitoring probe VERMONT.

In summary, it can be said that Netconf is a promising alternative to SNMP with respect to the configuration of monitoring probes. Necessarily, the usage of Netconf requires the standardization of XML-based configuration data models in order to guarantee interoperability between different Netconf implementations. The configuration of monitoring probes will probably be an agenda item in the IPFIX standardization process. Hence, this paper may provide valuable input to the upcoming discussion.

REFERENCES

[1] B. Claise, "Cisco Systems NetFlow Services Export Version 9," RFC 3954 (Informational), Oct. 2004.

[2] B. Claise, "IPFIX Protocol Specifications," Internet Draft, Work in progress, draft-ietf-ipfix-protocol-19, Sept. 2005.

[3] N. Duffield, "A Framework for Packet Selection and Reporting," Internet-Draft, Work in progress, draft-ietf-psamp-framework-10, Jan. 2005.

[4] R. Enns, "NETCONF Configuration Protocol," Internet Draft, Work in progress, draft-ietf-netconf-prot-10, Dec. 2005.

[5] F. Dressler and G. Carle, "History - high speed network monitoring and analysis," in 24th IEEE Conference on Computer Communications (IEEE INFOCOM 2005), Mar. 2005.

[6] J. Quittek, T. Zseby, B. Claise, and S. Zander, "Requirements for IP Flow Information Export (IPFIX)," RFC 3917 (Informational), Oct. 2004.

[7] F. Dressler, C. Sommer, and G. Münz, "IPFIX Aggregation," Internet Draft, Work in progress, draft-dressler-ipfix-aggregation-02, Dec. 2005.

[8] T. Zseby, M. Molina, N. Duffield, S. Niccolini, and F. Raspall, "Sampling and Filtering Techniques for IP Packet Selection," Internet-Draft, Work in progress, draft-ietf-psamp-sample-tech-07, July 2005.

[9] T. Dietz and B. Claise, "Definitions of Managed Objects for Packet Sampling," Internet-Draft, Work in progress, draft-ietf-psamp-mib-05, Oct. 2005.

[10] M. Wasserman and T. Goddard, "Using the NETCONF Configuration Protocol over Secure Shell (SSH)," Internet Draft, Work in progress, draft-ietf-netconf-ssh-05, Oct. 2005.

[11] E. Lear and K. Crozier, "Using the NETCONF Protocol over Blocks Extensible Exchange Protocol (BEEP)," Internet Draft, Work in progress, draft-ietf-netconf-beep-08, Jan. 2006.

[12] T. Goddard, "Using the Network Configuration Protocol (NETCONF) over the Simple Object Access Protocol (SOAP)," Internet Draft, Work in progress, draft-ietf-netconf-soap-07, Dec. 2005.

[13] D. Gabrijeleie, Y. Carlinet, G. Münz, F. Dressler, R. Wehage, S. Yusuf, P. Sagmeister, and G. Dittmann, "Revised Interfaces Specification, DIADEM Firewall Deliverable D6," Jan. 2006.

[14] G. Münz, O. Paul, and F. Dressler, "Initial Violation Detection Prototype, DIADEM Firewall Deliverable D9," July 2005.

[15] R. v. Engelen, gSOAP Web Services Toolkit Homepage, http://www.cs.fsu.edu/~engelen/soap.html.

[16] R. v. Engelen and K. A. Gallivany, "The gSOAP Toolkit for Web Services and Peer-To-Peer Computing Networks," in IEEE Cluster Computing and the GRID 2002, May 2002, pp. 128–135.

[17] R. S. Engelschall, OpenSSL Project Homepage, http://www.openssl.org/.

[18] DIADEM Firewall Homepage, http://www.diadem-firewall.org.

[19] M.-J. Choi, H.-M. Choi, J. W. Hong, and H.-T. Ju, "XML-Based Configuration Management for IP Network Devices," IEEE Commun. Mag., vol. 42, no. 7, pp. 84–91, 2004.

[20] J. Schönwälder, A. Pras, and J.-P. Martin-Flatin, "On the Future of Internet Management Technologies," IEEE Commun. Mag., vol. 41, no. 10, pp. 90–97, 2003.

[21] J. v. Sloten, A. Pras, and M. v. Sinderen, "On the Standardisation of Web Service Management Operations," in 10th Open European Summer School and IFIP WG6.3 Workshop (EUNICE 2004), Tampere, Finland, 2004, pp. 143–150.

[22] M.-J. Choi, J. W. Hong, and H.-T. Ju, "XML-Based Network Management for IP Networks," ETRI Journal, vol. 25, no. 6, pp. 445–463, 2003.

[23] A. Pras, T. Drevers, R. v. d. Meent, and D. Quartel, "Comparing the Performance of SNMP and Web Services-Based Management," IEEE eTNSM (Transactions on Network and Service Management), vol. 1, no. 2, 2004.