NETCONF Interoperability Lab

Vaibhav Bajpai & Jürgen Schönwälder
Computer Science, Jacobs University Bremen
(v.bajpai | j.schoenvaelder)@jacobs-university.de

Need for interoperability testing
Within the Internet Engineering Task Force (IETF), interoperability reports are needed in order to advance specifications on the standards-track because interoperability helps:

- to uncover errors and ambiguities in standards.
- to identify human programming errors.
- to resolve different interpretations of the standard.
- to evaluate different choice of options allowed by the standard.

Background
The Network Configuration (NETCONF) protocol [1] and the associated YANG data modeling language [2] are foundations of a new network management framework evolving in the IETF. Here is a list of recent events related to NETCONF interoperability testing:

- Early work on NETCONF interoperability testing published in 2009 [3].

Research Contribution
An online NETCONF interoperability testing lab that is openly accessible 24 hours a day:

- to improve NETCONF interoperability.
- to foster community-based NETCONF education.

System Architecture

<table>
<thead>
<tr>
<th>Control services (dom0)</th>
<th>Web frontend (domU)</th>
<th>ConfD server (domU)</th>
<th>YumaPro server (domU)</th>
<th>libnetconf server (domU)</th>
<th>OpenYuma server (domU)</th>
</tr>
</thead>
</table>

- Separate virtual machines running Debian are used for each NETCONF server.
- A separate virtual machine is deployed to provide the web frontend.
- The NETCONF servers are dual-stacked and reachable both over IPv4 and IPv6.

NETCONF clients and servers
The following NETCONF client and server implementations have been considered for integration into the NETCONF interoperability lab:

- ConfD: http://www.tail-f.com/confd
- YumaPro: http://www.yumaworks.com/yumapro
- OpenYuma: http://github.com/OpenClovis/OpenYuma
- libnetconf: https://code.google.com/p/libnetconf

Feature Comparison
The supported NETCONF capabilities and protocol extensions are enlisted below (✓ means the capability is available but disabled by default):

<table>
<thead>
<tr>
<th>Capability</th>
<th>ConfD</th>
<th>YumaPro</th>
<th>OpenYuma</th>
<th>libnc</th>
</tr>
</thead>
<tbody>
<tr>
<td>base:1.0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>wriable-reading:1.0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>commit:1.0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>rollback-on-error:1.0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>startup:1.0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>with-default:1.0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>confed-commit:1.0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>confed-commit:1.1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>candidate:1.0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>candidate:1.1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>validate:1.0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>notification:1.0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>iteration:1.0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>with-default:1.0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

The supported standardized YANG data models are enlisted below:

<table>
<thead>
<tr>
<th>Data Models</th>
<th>ConfD</th>
<th>YumaPro</th>
<th>OpenYuma</th>
<th>libnc</th>
</tr>
</thead>
<tbody>
<tr>
<td>self-netconf-types:1.0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>self-yang-types:1.0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>self-netconf-monitoring:1.0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>self-netconf-notifications:1.0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>self-netconf-semantics:1.0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>self-netconf-with-default:1.0</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

Available Services
The NETCONF interoperability lab provides the following services:

- Access to a set of NETCONF server implementations including supporting material.
- An online catalogue of NETCONF server implementations.
- An online catalogue of NETCONF client implementations.
- A collection of tests written in Python that can be used for testing of NETCONF servers.
- Possibility to execute tests against remote NETCONF servers.

Xen hypervisor
http://www.interop-lab.net

References


Acknowledgements
We would like to thank Martin Björklund (Tail-f) and Andy Bierman (YumaWorks) for providing us with licensed copies of their commercial NETCONF implementations. This work was partly funded by Flamingo, a Network of Excellence project (ICT-318488) supported by the European Commission under its Seventh Framework Programme.