Experience of Designing a Network Management System

Yasuhiro TERAMOTO
Kyoto University

Y. ATARASHI
AlaxalA Networks

R. ATARASHI
IIJ Innovation Institute

Yasuo OKABE
Kyoto University
Motivation and Goals

• This presentation describes our experiences of designing a network management system
  – Mainly using the NETCONF protocol for configuring
• Feedback of the experience to the WG
  – Issues of the NETCONF protocol
Background

• We used the NETCONF protocol to manage configurations of multiple networking equipment by a central server
• The server compiles policy into the configuration using resource database and set/edit the configuration via NETCONF
• The server aggregate events (now using SNMP, in future NETCONF)
Topics of NETCONF experiences

• The main topic of this presentation
  – Transport layer
    • SSH
    • SOAP/HTTP
    • Error handling
  – Capability exchange
    • Error handling
NETCONF transport protocol

• We implemented SSH and SOAP(experimental) transport (using Java and Scala)
• The SSH protocol is complicated and hard to ensure performance
  – One SSH session per one device (if keeping session)
  – Negotiation takes time
  – The SSH protocol has no way to notify transport error

<table>
<thead>
<tr>
<th>Protocol</th>
<th>Our Implementation (using libraries)</th>
<th>Transport</th>
<th>Transport Error Notification</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSH</td>
<td>1000 lines</td>
<td>Framing</td>
<td>No mechanism</td>
</tr>
<tr>
<td>SOAP/HTTP</td>
<td>&lt;100 lines</td>
<td>HTTP messaging</td>
<td>400 Bad Request Response</td>
</tr>
</tbody>
</table>
Capabilities Exchange

- The peer terminates the session without notification on receiving invalid hello
- Difficult to determine the reason of disconnection
  - No error notification
  - Client may send <rpc> before disconnection
Conclusion

• The SSH protocol is too complicated for mandatory transport protocol
  – The core protocol itself should be as simple as possible

• Notification errors on capability exchange
Appendix: Data Model

• Current approach of data model is device oriented
  – Models become complex to fit one model into various kinds of devices
• We are expecting result models to be in a reasonable compromise
Appendix: Notification Mechanism

• Notification capability [RFC5277]
  – Large number of states and conditions
  – Mandatory support of :interleave makes it simple
  – More simple with start up notification

current notification mechanism

interleave as mandatory

notification on startup