Experience of Designing a Network Management System

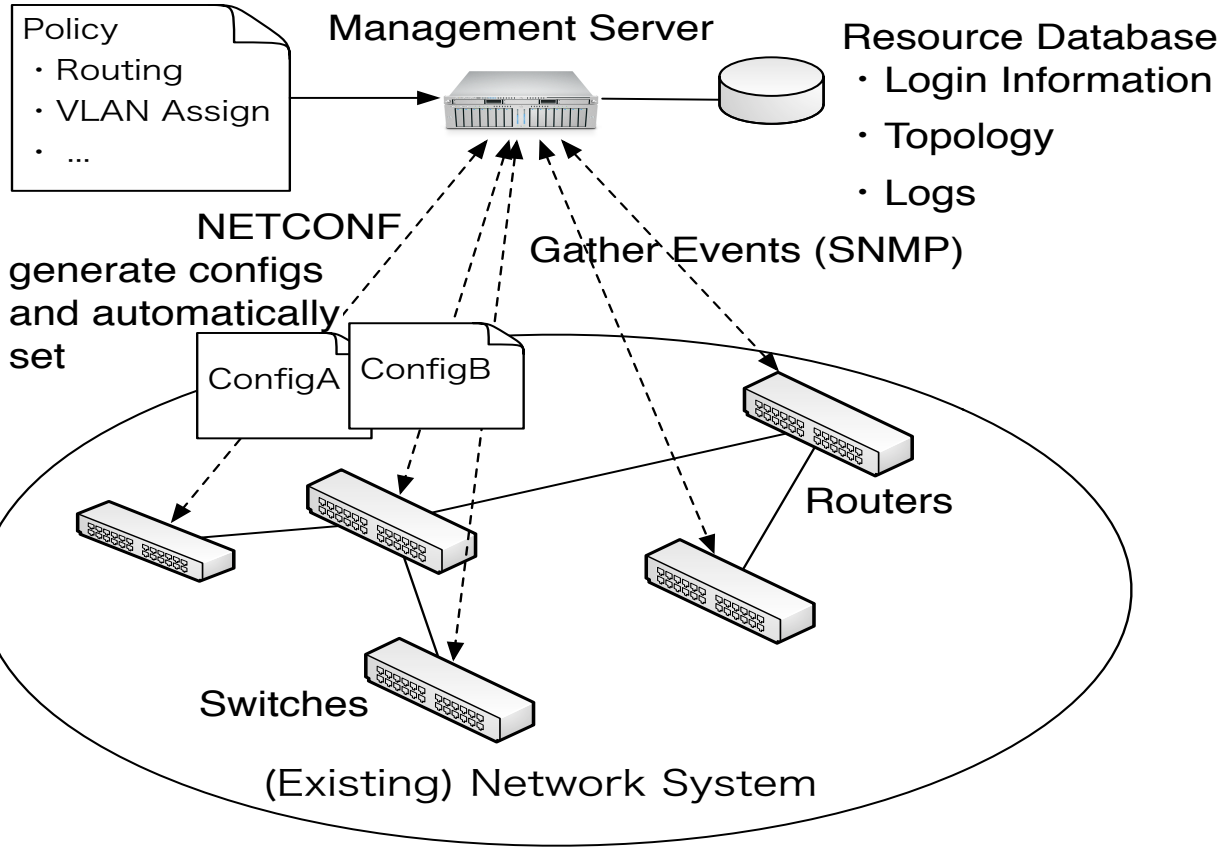
Yasuhiro TERAMOTO
Kyoto University
Y. ATARASHI
AlaxalA Networks
R. ATARASHI
IIJ Innovation Institute
Yasuo OKABE
Kyoto University

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

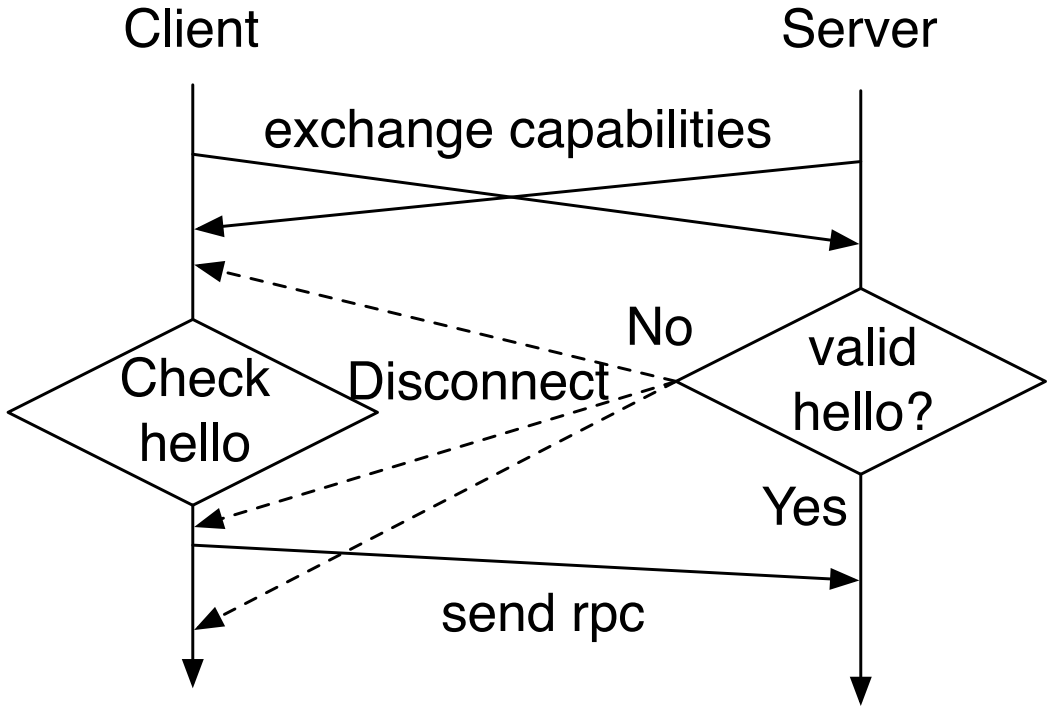
NETCONF transport protocol

- 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

Protocol	Our Implementation (using libraries)	Transport	Transport Error Notification
SSH	1000 lines	Framing	**No mechanism**
SOAP/HTTP	<100 lines	HTTP messaging	400 Bad Request Response

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

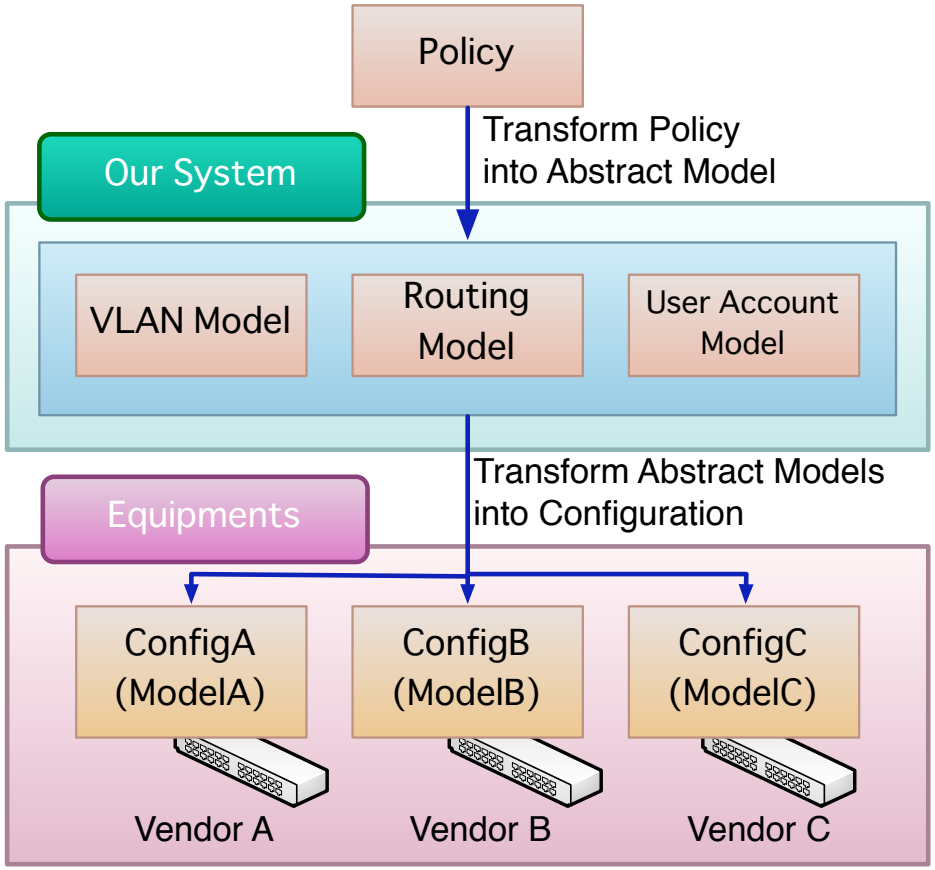


Conclusion

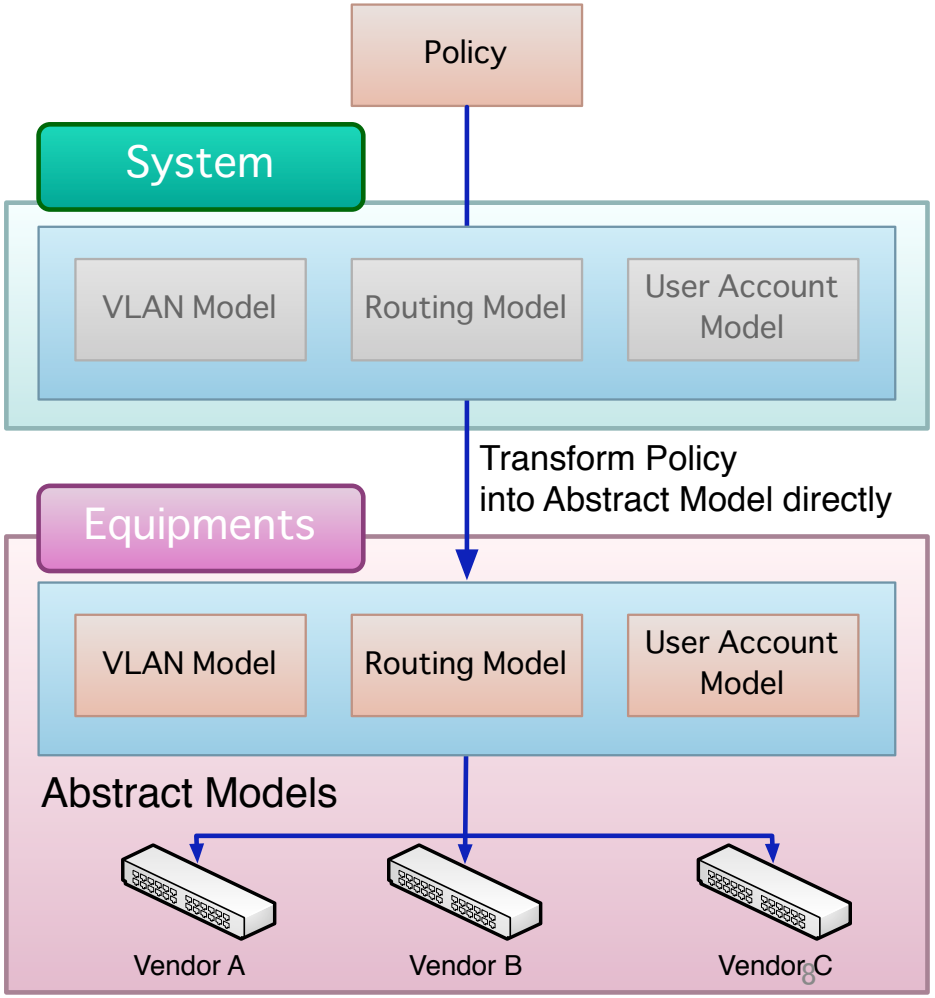
- The SSH protocol is too complicated for mandatory transport protocol
  - The core protocol itself should be as simple as possible
- Notification errors on capability exchange

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



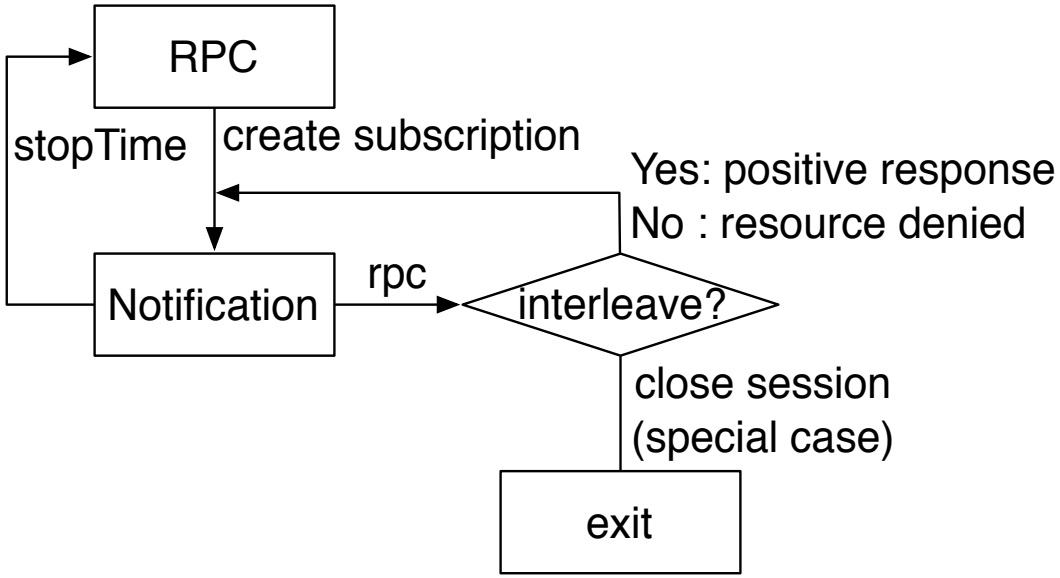
Our current implementation of data modeling



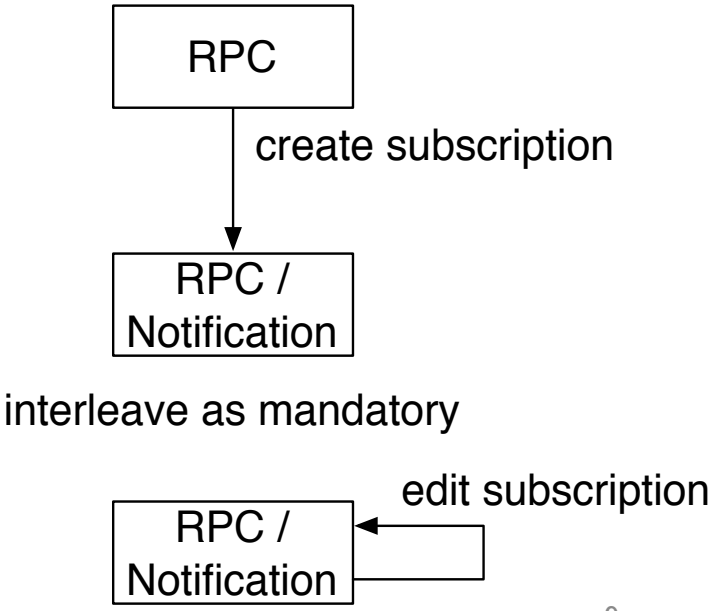
Our expecting data modeling

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



notification on startup