NETCONF-Based Network Management System for NDN

Rajender Kumar, Alex Afanasyev
Florida International University, Miami

NDNComm 2018
September 20, 2018

Contents

Motivation

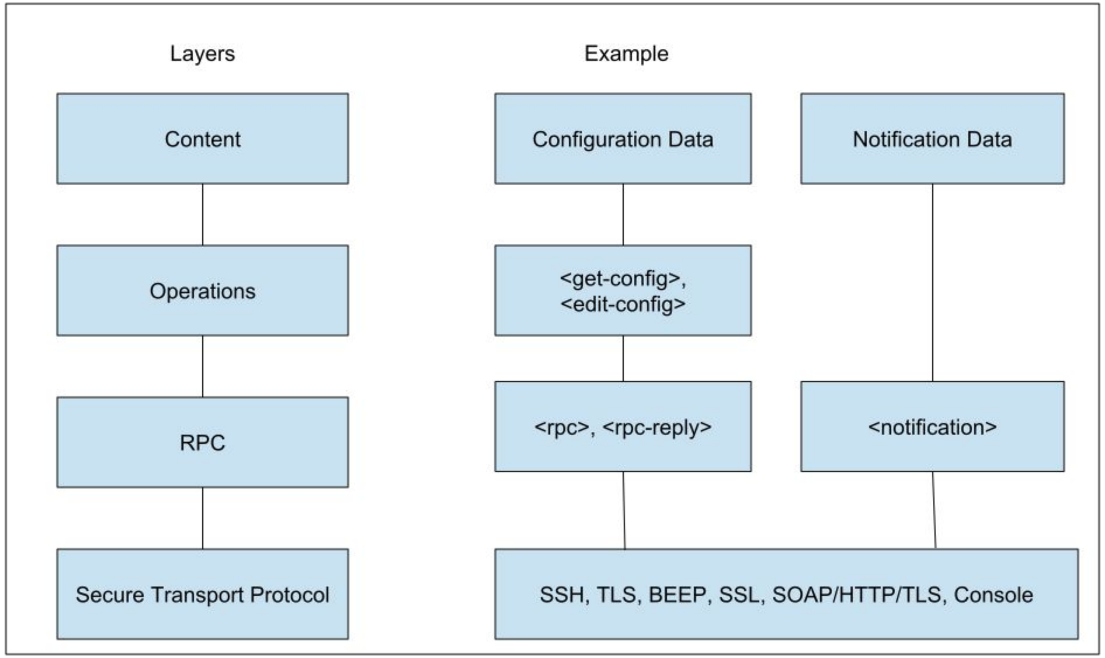
- Network Management protocols have traditionally been used for
  - Managing networks
  - Backup and restore configurations
  - Error checking to ensure consistent configuration
  - Reports generation and analysis
- Existing Network Management protocols include:
  - SNMP
  - Command Line Interface (CLI)
  - Ansible
  - Chef
  - NETCONF

NETCONF Overview

NETCONF is a network management protocol specifically designed to support network configuration management

- Distinction between configuration data and state data
- Network wide configurations instead of single devices
- Multiple configuration datastores (running, startup, . . .)
- Support for configuration change transactions
- Configuration testing and validation support
- Selective data retrieval with filtering
- Streaming and playback of event notifications
- Extensible remote procedure call mechanism

NETCONF Layering Model

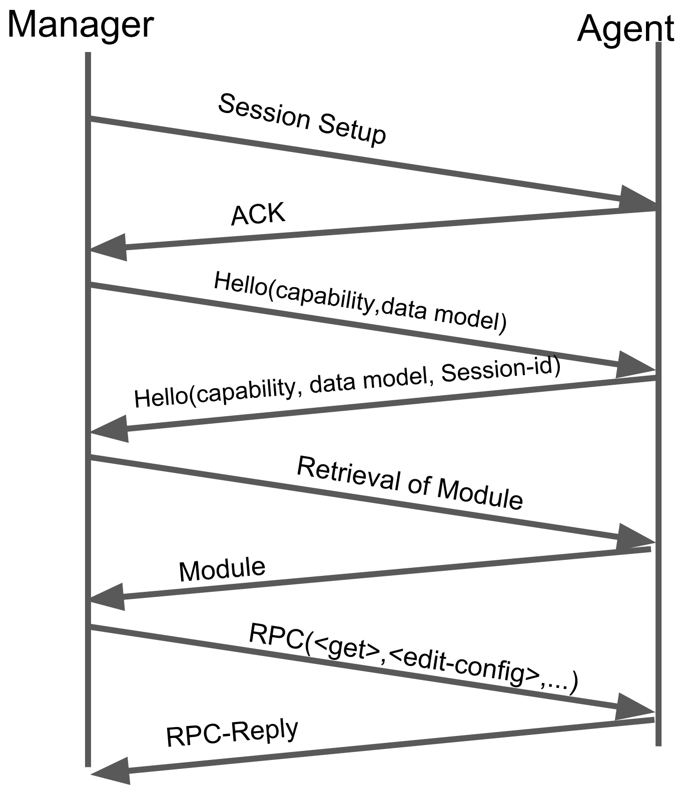


Multiplicity of Configurations

NETCONF designed to support multiple complete sets of configuration information that is required to get a device from its initial default state into a desired operational state.

- **<running>**
- **<startup>**
- **<candidate>**

Overview of NETCONF Protocol Actions

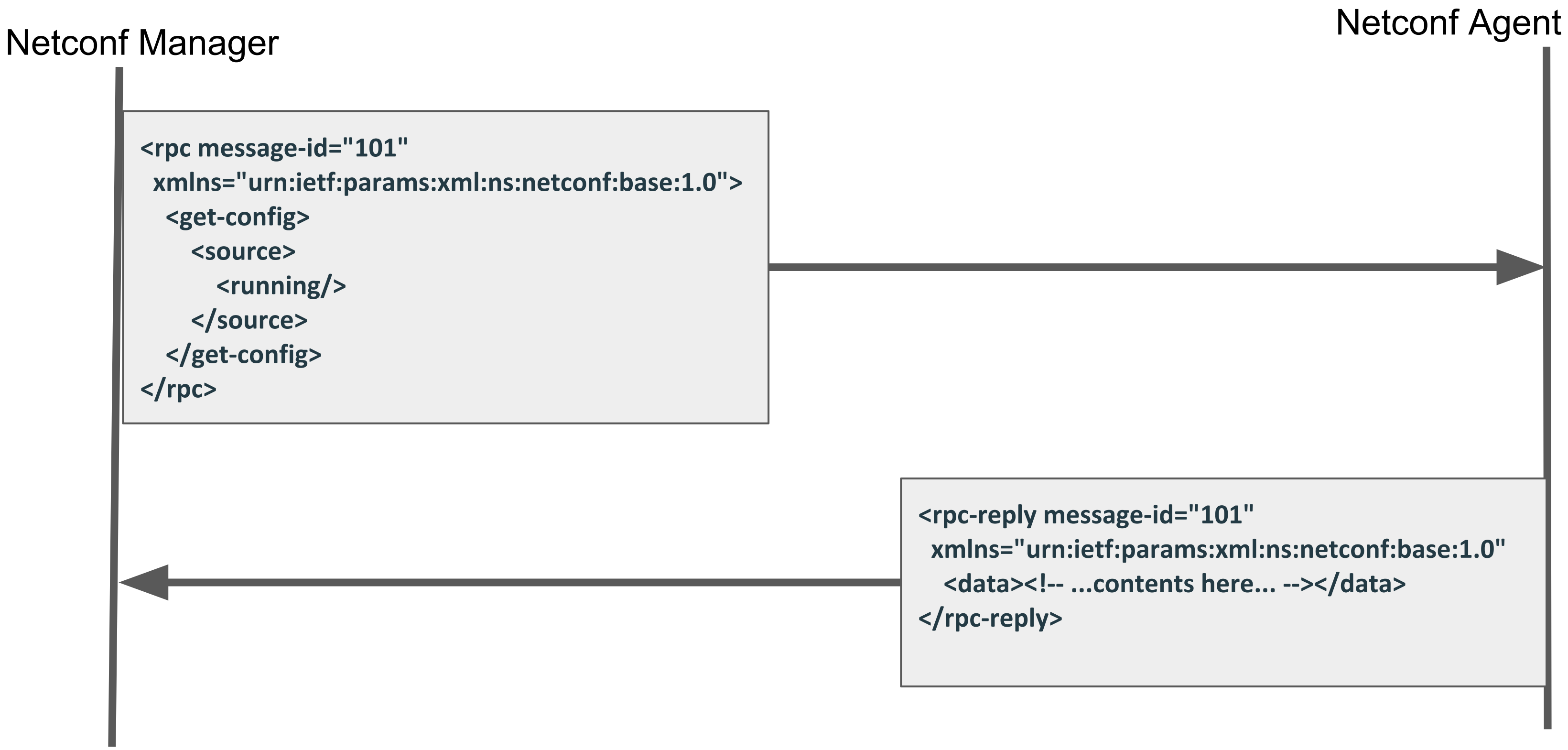


Manager
Agent
Session Setup
ACK
Hello(capability,data model)
Hello(capability, data model, Session-id)
Retrieval of Module
Module
RPC(<get>,<edit-config>,...)
RPC-Reply

Some Data model

- urn:onf:params:xml:ns:yang:core-model
- urn:onf:params:xml:ns:yang:ltp-path
- urn:onf:params:xml:ns:yang:g.874.1-model

Remote Procedure Call



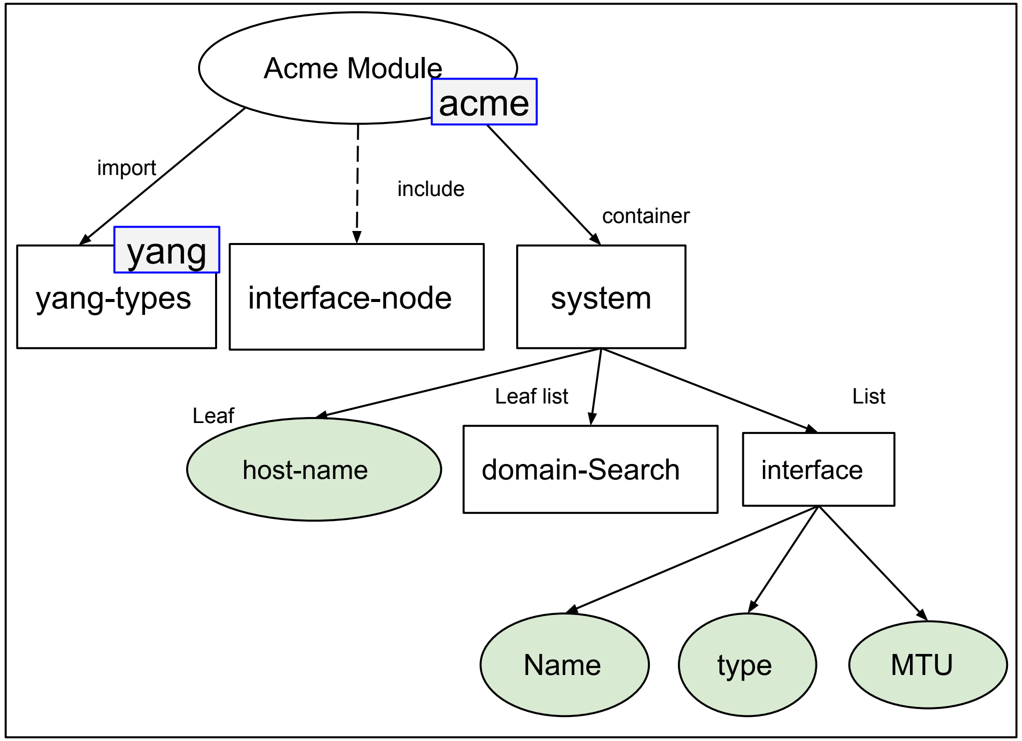
```
<rpc message-id="101"
 xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
  </get-config>
</rpc>
```

```
<rpc-reply message-id="101"
 xmlns="urn:ietf:params:xml:ns:netconf:base:1.0"
  <data><!-- ...contents here... --></data>
</rpc-reply>
```

YANG Overview

- Extensible data modeling language designed specifically for network management
- Ability to model configuration data, state data, operations, and notifications
- Easy to create YANG model for any device or implementation
- Provide hierarchical data models
- YIN is XML representation of YANG

YANG Modules and Submodules



```
module acme-system {
   namespace "http://acme.example.com/system";
   prefix "acme";
   import "yang-types" {
     prefix "yang";
   }
   include "interface-node";
   ...
   revision 2007-11-05 { description "Initial revision.";  }
   container system {
     leaf host-name {
       type string;
       description "Hostname for this system";
     }
     leaf-list domain-search {
       type string;
       description "List of domain names to search";
     }
     list interface {
       key "name";
       description "List of interfaces in the system";
       leaf name {
         type string;
       }
       leaf type {
         type string;
       }
       leaf mtu {
         type int32;
       }
     }
   }
}
```

NDNCONF Protocol

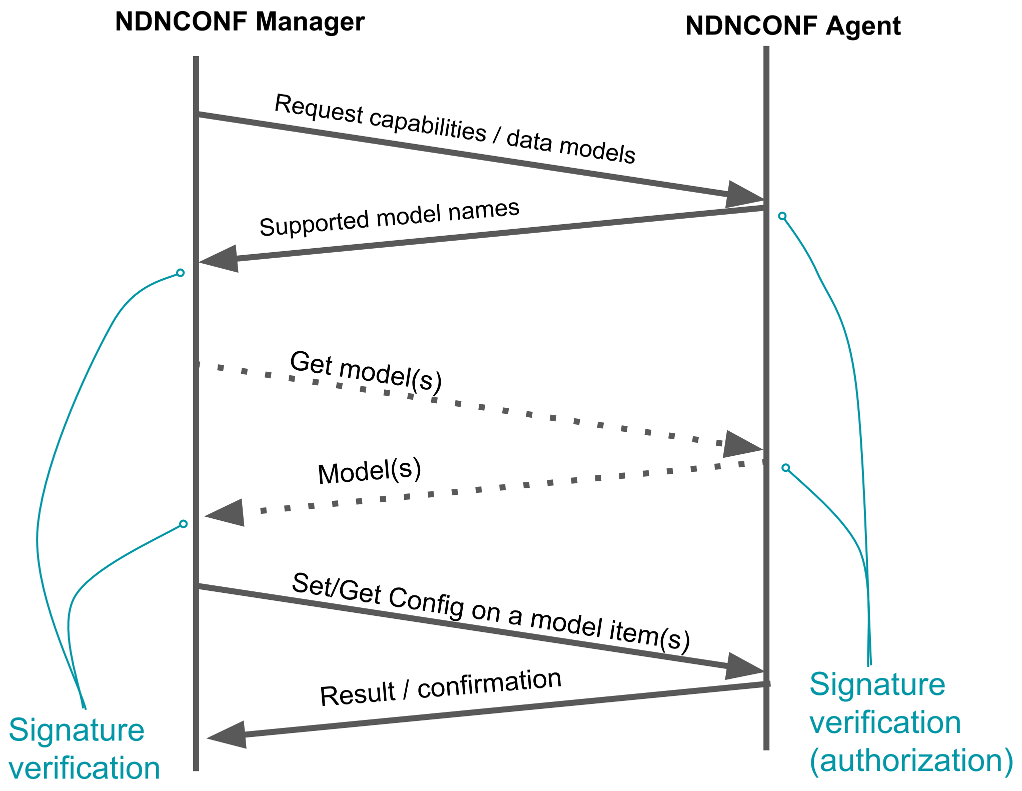
YANG effectively defines naming hierarchy

- NDNCONF use it to issue commands and request data

NDN can directly operate on YANG hierarchy

- Interests to retrieve “readable”
- Command interests to write
- Command interests and responses leverage NDN security
- Leverage NDN trust schema
  - per-command
  - per (sub-)namespace granularity

Overview of NDNCONF Protocol Actions



- NDNCONF manager request capabilities, and the supported data models of agent.
- NDNCONF manager perform configuration operation by sending interest packets.
- Signature verification is performed on each end.

NDN Core Models

Core System Data Model

- [existing] System Identification
- [existing] System Time Mgmt

Core NDN Data model

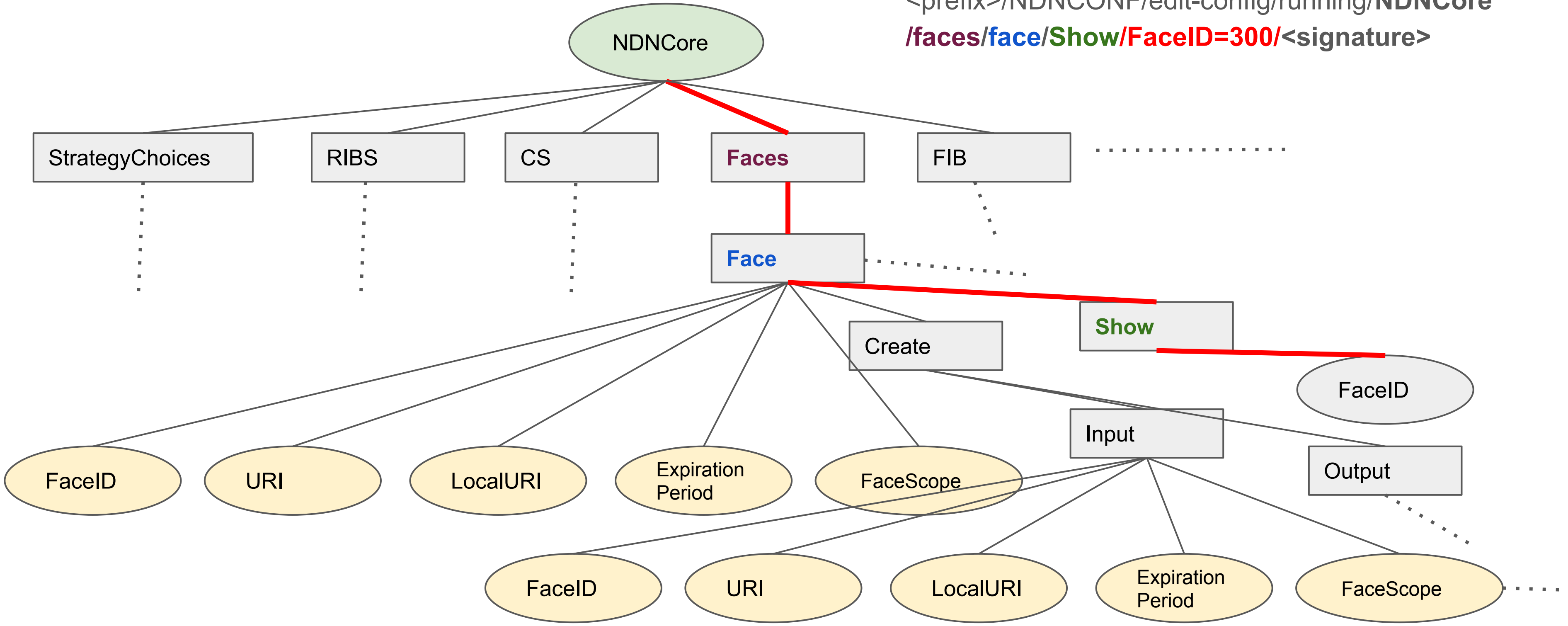
- Faces, RIB, FIB, CS, Measurements
  - Effectively a generalization of our current NDN management protocol

Some Data model

- <prefix>/ndnconf/NDNCore/rib
- <prefix>/ndnconf/NDNCore/ContentStore
- <prefix>/ndconf/NDNCore/FIB
- <prefix>/ndconf/NDNCore/face

Core NDN Data Model: Construction of Interest Packet

<prefix>/NDNCONF/edit-config/running/**NDNCore**
/faces/face/Show/FaceID=300/<signature>



Interest Packet Format for NDNCONF

NDNCONF Interest Packet:
<prefix>/NDNCONF/<operation>/<datastore>/<module>/container>/<subcontainer>/…/Parameters:<leaf>=<value>/<params-sha256>

- **<prefix>**
  - Name of the server/device
- **<operation>**
  - get-config, set-config , edit-config, get, set, etc.
- **<datastore>**
  - Running, Candidate, Start-up
- **<Module>**
  - Name of yang module. For example, NDNCore.yang
- **<container>**
  - Face, RIB, ContentStore, FIB, etc.
- **<subcontainer>**
  - Command like create(face), Show(face list), and etc.
- **Parameters: <leaf>=<value>**
  - Parameter names and corresponding values for the command
- **Signature**
  - Signature of the requester

Example: Interest Packet to Create a Face

Example: Create a face with the specified remote FaceUri, local FaceUri, and persistency.

- **NDNCONF Interest Packet:** <prefix>/ndnconf/edit-config/running/faces/face/create/remoteUri=ether//:[08:00:27:01:01:01] &localUri=dev://eth2&persistency=permanent/<signature>

Command Interests And Responses Leverage NDN Security

- Command interests and responses leverage NDN security
    - Each command interest and data packet are directly secured, independent of session security
- Relation between command/data names and keys can manage control granularity

NDNCONF Control Granularity

<prefix>/ndnconf/edit-config/KEY/23

Authorized to configure any parameters of the network device identified by <prefix>

<prefix>/ndnconf/edit-config/running/KEY/11

Only running configuration of the network device identified by <prefix>

<prefix>/ndnconf/edit-config/running/faces/KEY/54

Only running configuration of the faces of the network device identified by <prefix>

Summary

- Formulated an initial Yang Data model for NDN Management Module
    - https://github.com/rkuma013/NDNCONF-Yang/blob/master/NDNCore.yang
- Designed the format of Interest and Data Packets for NDNCONF
    - https://github.com/rkuma013/NDNCONF-Yang

Future Work

- Explore the security aspect of NDNCONF in more details
- Finalize design of NDNCONF
- Implement and test
- Prepare formal documentation of NDNCONF protocol

Thank you