Chapter 4

IoT System Management with NETCONF-YANG
Outline

- Need for IoT Systems Management
- SNMP
- Network Operator Requirements
- NETCONF
- YANG
- IoT Systems Management with NETCONF-YANG

Need for IoT Systems Management

- Automating Configuration
- Monitoring Operational & Statistical Data
- Improved Reliability
- System Wide Configurations
- Multiple System Configurations
- Retrieving & Reusing Configurations
Simple Network Management Protocol (SNMP)

• SNMP is a well-known and widely used network management protocol that allows monitoring and configuring network devices such as routers, switches, servers, printers, etc.

• SNMP component include
  • Network Management Station (NMS)
  • Managed Device
  • Management Information Base (MIB)
  • SNMP Agent that runs on the device
Limitations of SNMP

• SNMP is stateless in nature and each SNMP request contains all the information to process the request. The application needs to be intelligent to manage the device.

• SNMP is a connectionless protocol which uses UDP as the transport protocol, making it unreliable as there was no support for acknowledgement of requests.

• MIBs often lack writable objects without which device configuration is not possible using SNMP.

• It is difficult to differentiate between configuration and state data in MIBs.

• Retrieving the current configuration from a device can be difficult with SNMP.

• Earlier versions of SNMP did not have strong security features.
Network Operator Requirements

- Ease of use
- Distinction between configuration and state data
- Fetch configuration and state data separately
- Configuration of the network as a whole
- Configuration transactions across devices
- Configuration deltas
- Dump and restore configurations

- Configuration validation
- Configuration database schemas
- Comparing configurations
- Role-based access control
- Consistency of access control lists:
  - Multiple configuration sets
  - Support for both data-oriented and task-oriented access control
NETCONF

- Network Configuration Protocol (NETCONF) is a session-based network management protocol. NETCONF allows retrieving state or configuration data and manipulating configuration data on network devices.
NETCONF

- NETCONF works on SSH transport protocol.
- Transport layer provides end-to-end connectivity and ensure reliable delivery of messages.
- NETCONF uses XML-encoded Remote Procedure Calls (RPCs) for framing request and response messages.
- The RPC layer provides mechanism for encoding of RPC calls and notifications.
- NETCONF provides various operations to retrieve and edit configuration data from network devices.
- The Content Layer consists of configuration and state data which is XML-encoded.
- The schema of the configuration and state data is defined in a data modeling language called YANG.
- NETCONF provides a clear separation of the configuration and state data.
- The configuration data resides within a NETCONF configuration datastore on the server.
YANG

• YANG is a data modeling language used to model configuration and state data manipulated by the NETCONF protocol.
• YANG modules contain the definitions of the configuration data, state data, RPC calls that can be issued and the format of the notifications.
• YANG modules defines the data exchanged between the NETCONF client and server.
• A module comprises of a number of 'leaf' nodes which are organized into a hierarchical tree structure.
• The 'leaf' nodes are specified using the 'leaf' or 'leaf-list' constructs.
• Leaf nodes are organized using 'container' or 'list' constructs.
• A YANG module can import definitions from other modules.
• Constraints can be defined on the data nodes, e.g. allowed values.
• YANG can model both configuration data and state data using the 'config' statement.
This YANG module is a YANG version of the toaster MIB.

The toaster YANG module begins with the header information followed by identity declarations which define various bread types.

The leaf nodes (‘toasterManufacturer’, ‘toasterModelNumber’ and ‘oasterStatus’) are defined in the ‘toaster’ container.

Each leaf node definition has a type and optionally a description and default value.

The module has two RPC definitions (‘make-toast’ and ‘cancel-toast’).
IoT Systems Management with NETCONF-YANG

- Management System
- Management API
- Transaction Manager
- Rollback Manager
- Data Model Manager
- Configuration Validator
- Configuration Database
- Configuration API
- Data Provider API