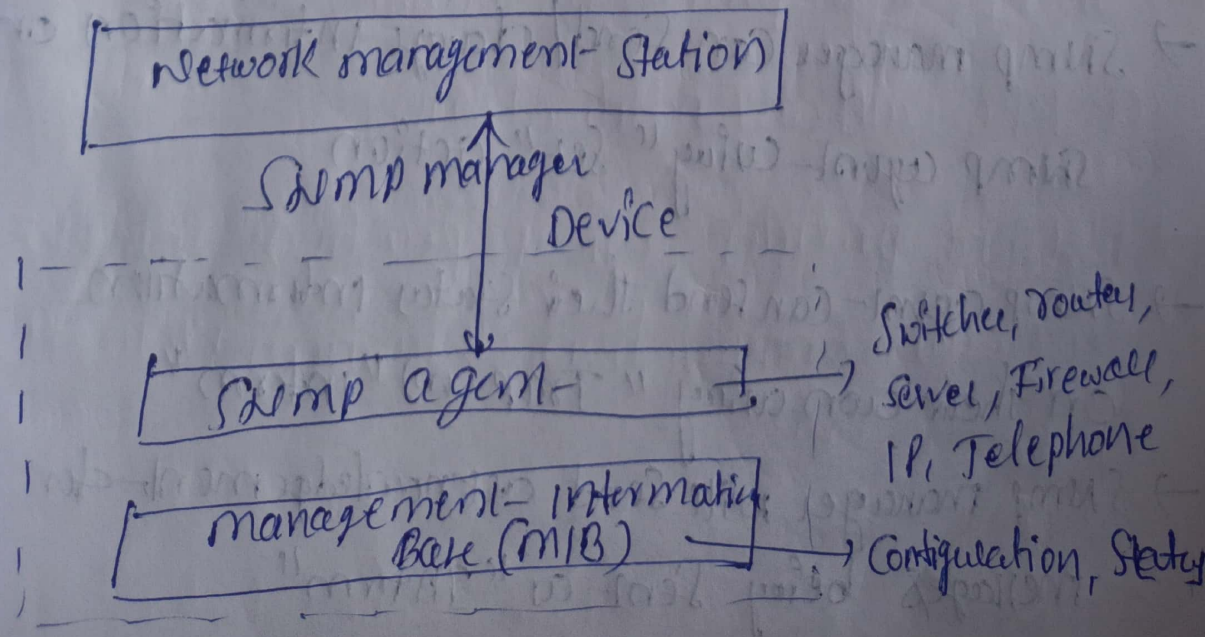
Simple Network management protocol (SNMP)

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

SNMP Component include

1) Network management station (NMS)
2) managed Device
3) management Information Base (MIB)
4) SNMP agent



Scanned with OKEN Scanner

→ SNMP manager a Network management application running on a PC (or) Server

→ MIB (management-Information Base) Contains Variables describing device Configuration & States
Variables are defined as
Name, interface Status, Uptime Temperature

→ SNMP manager can Send Query messages to the device being managed/monitored

→ SNMP manager retrieve information from SNMP agent-using "get" action
↑ Device Name
Interface Status
Routing Information

→ SNMP manager can Send/Change Information on SNMP agent-using "Set" action

→ SNMP agent-Can send their Status information on their own using "Trap (or) Inform"

→ SNMP manager will Send acknowledgement-for messages being Sent-as "Inform"

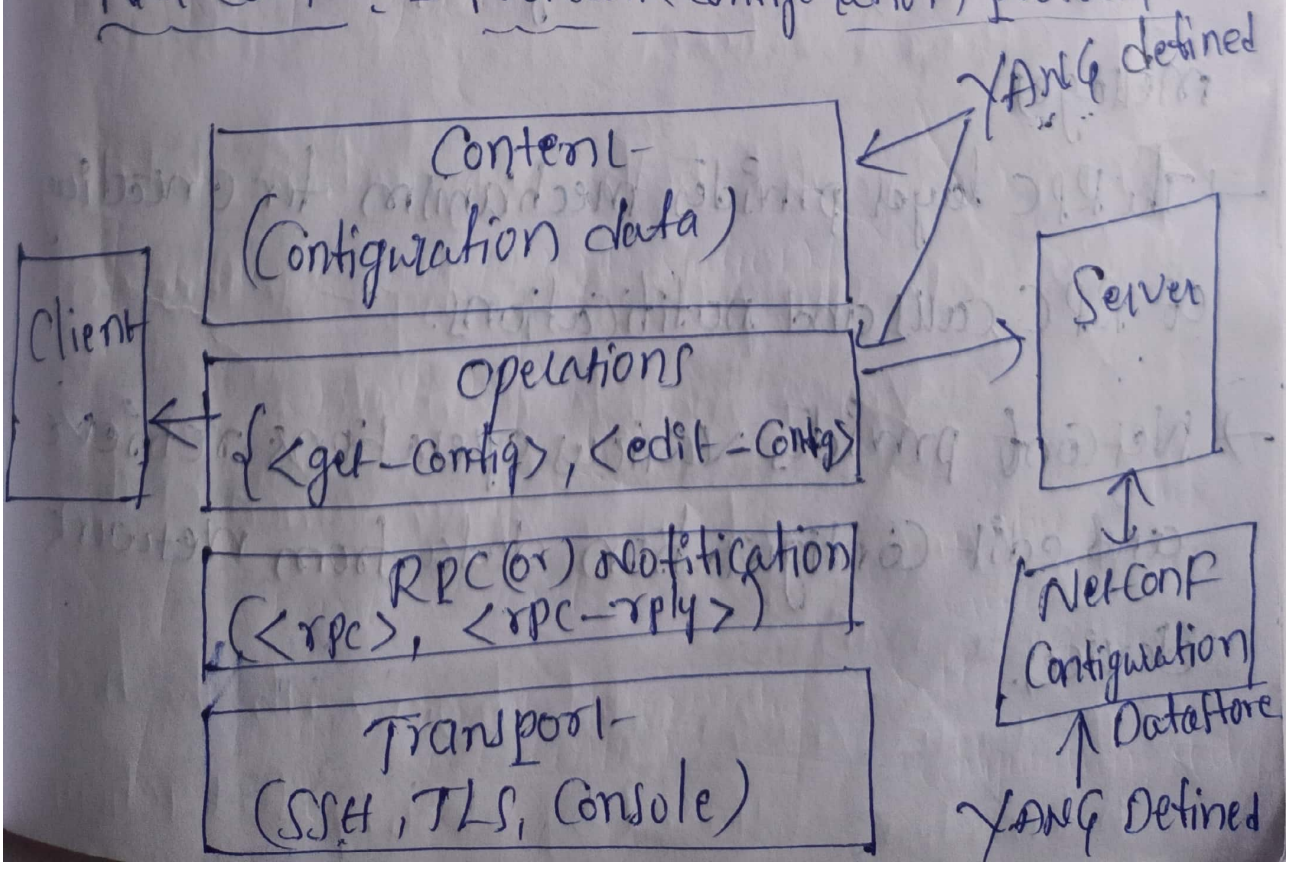
SNMP Version

SNMP V_1: This is initial implementation of SNMP Protocol

SNMPV2: it improved the performance in terms of security of SNMP V1 using community string password

SNMPV3: it provided Confidentiality, Integrity & authentication to SNMP
encription
no change

NetConf: Network Configuration protocol



Network Configuration protocol (NETCONF)

is a Sellion-based network management protocol. NetConf allows Retrieving State (or) operational data & statistical data Configuration data and manipulating Configuration data on network devices

→ NETCONF works on SSH Transport protocol.

Transport layer provideds end-to-end Connectivity and ensure reliable delivery of messages.

→ NETCONF uses XML-encoded Remote procedure Calls (RPC's) for framing request and response messages.

→ The RPC layer provides mechanism for encoding of RPC calls and notifications.

→ Netconf provides various operations retrieve and edit Configuration data from network devices.

→ The Content layer Consists of Configuration and State data which is XML encoded.

The schem of the Configuration and State data is defined in a data modeling language called YANG

→ NETCONF provides a clear operation of the Configuration and State data

→ The Configuration data besides within a NETCONF Configuration data store on the Server.

YANG (Yet-Another Next Generation)

⇒ YANG is a data modeling language used to model Configuration and State data manipulated by the NETCONF protocol.

⇒ YANG module Contain the definions of the Configuration data. State data, RPC calls that can be issued and the format of the notifications.

⇒ YANG module defines the data exchanged blw the NETCONF client & Server

⇒ A module consists 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 construct-

⇒ A YANG module can import definitions from other module

⇒ Constraints can be defined on the data nodes e.g. allowed value

⇒ YANG can model both configuration data & State data using the 'Config' Statement-

YANG module Example

⇒ The YANG module is a YANG version of the toaster MIB

⇒ The toaster YANG module begin with the header information followed by

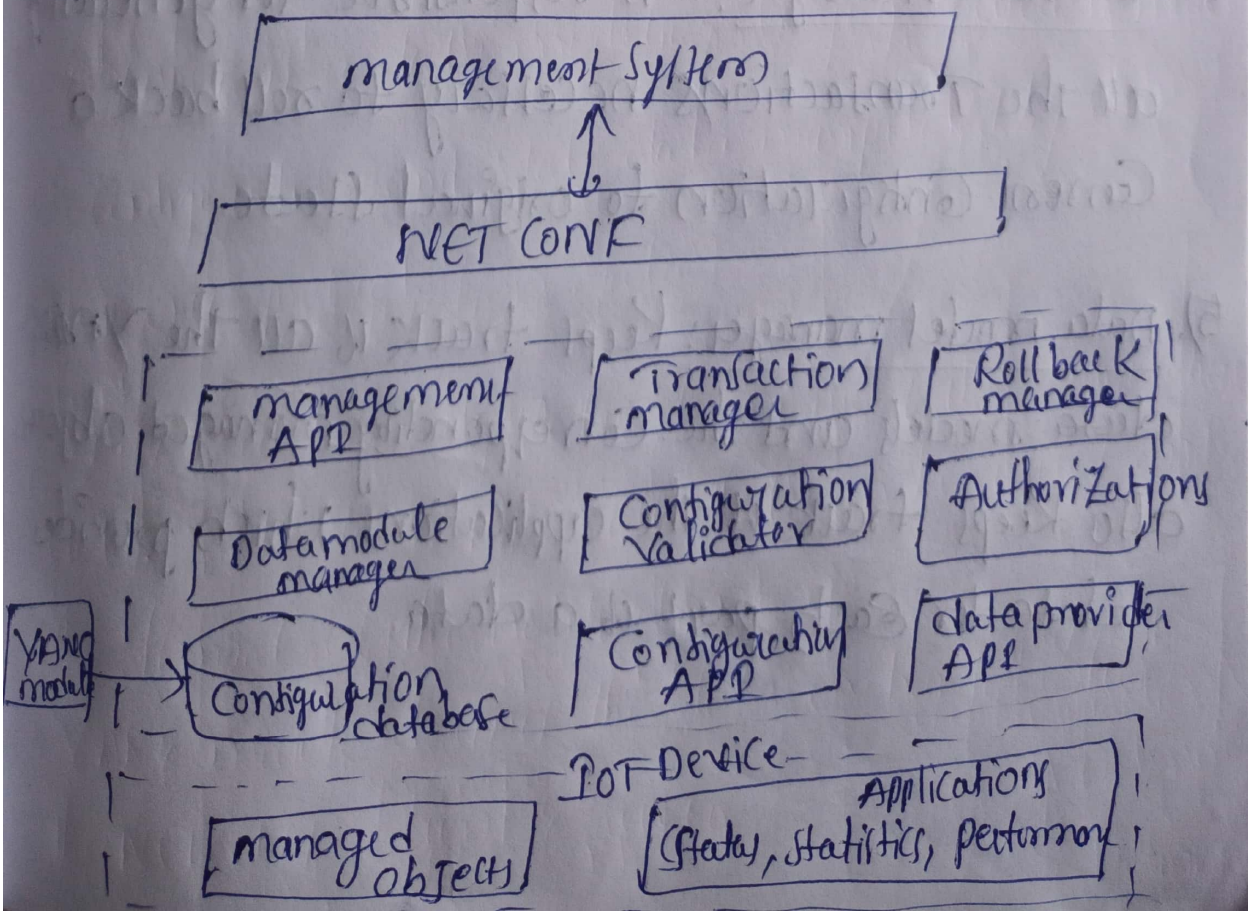
which defines various bread type

⇒ The leaf nodes ('toaster manufacturer', 'toaster model number' and 'toaster status') 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



⇒ 1) Management System - The operator uses a management system to send NETCONF message to configure the IoT Device and receive State information and notifications from the device as NETCONF messages.

2) management APP :- allows management application to start NETCONF sessions.

3). Transaction manager :- executes all the NETCONF transactions and ensure that ACID properties hold true for the Transactions

4). Rollback manager :- is responsible for generating all the Transactions necessary to roll back a Current Configuration to original state.

5). Data model manager : Keeps track of all the YANG data models and the Corresponding managed objects also keeps track of the applications which provides data for each part of a data.

6. Configuration Validator: checks if the resulting configuration after applying a Transaction would be a valid configuration.

7. Configuration database: Contains both configuration and operational data.

8. Configuration API: using the configuration API the application on the IoT device can be read configuration data from the configuration data store and write operational data to the operational data store.

9. Data provider API: - Application on the IoT device can register for call backs for various events using the data provider API. Through the data provider API, the applications can report statistics and operational data.

Scanned with OKEN Scanner