Unit III

IoT and M2M

&

IoT System Management with NETCONF – YANG

3.1 M2M

3.2 Difference between IoT and M2M

3.3 Software Defined Networking for IoT

3.4 Network Function Virtualization for IoT

3.5 Need for IoT Systems Management

3.6 Simple Network Management Protocol (SNMP)

3.7 Network Operator Requirements

3.8 NETCONF

3.9 YANG

3.10 IoT Systems Management with NETCONF – YANG.

IoT and M2M

&

IoT System Management with NETCONF – YANG

INTRODUCTION

Machine-to-machine (M2M) is a technology that uses a device attached to a machine to capture an event which is relayed through a mobile phone or fixed line network to an application that translates the event into meaningful information. The Internet of Things (IoT) is the next generation of the Internet based on the Internet Protocol (IP).

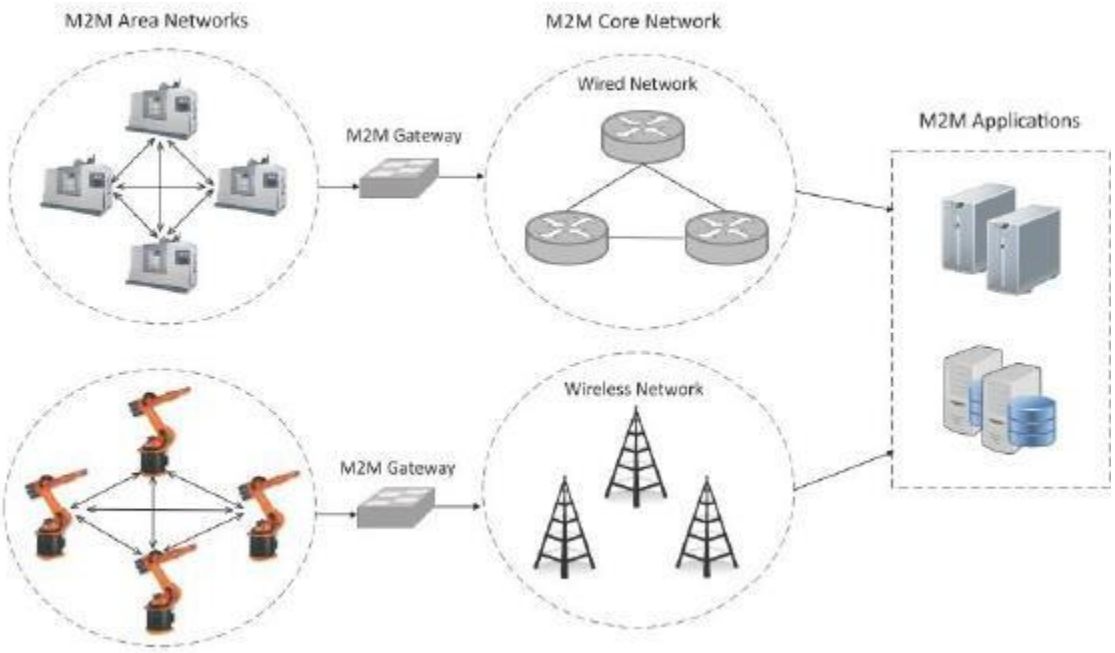
- Term which is often synonymous with IoT is Machine-to-Machine (M2M).
- IoT and M2M are often used interchangeably.

3.1 M2M

Machine-to-Machine (M2M) refers to networking of machines (or devices) for the purpose of remote monitoring and control and data exchange.

M2M System Architecture :

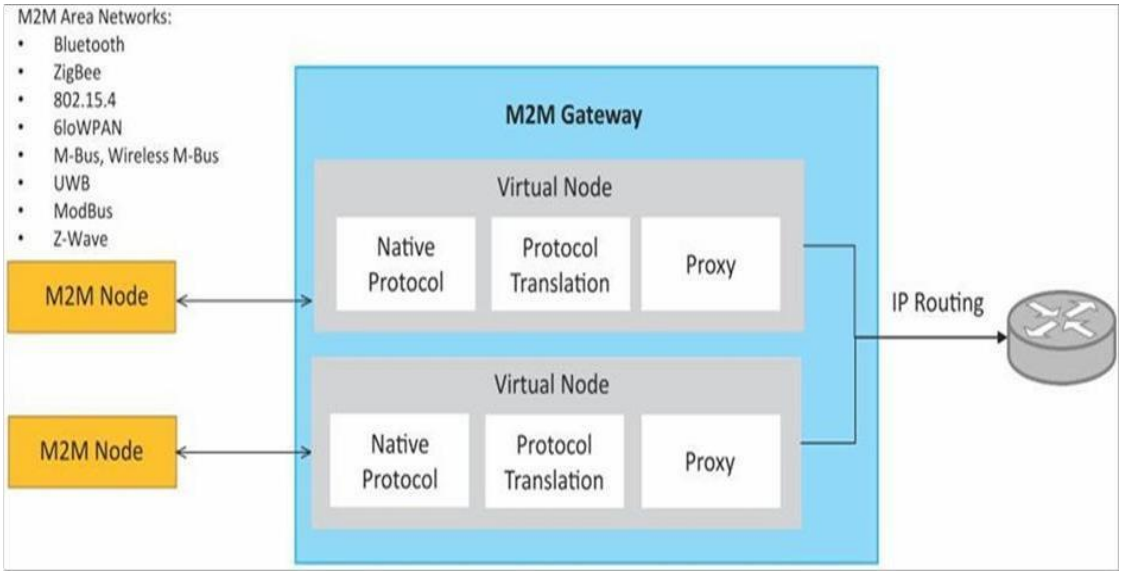
The following diagram shows the end-to-end architecture of M2M systems comprises of M2M area networks, communication networks and application domain.


M2M Area Networks
M2M Core Network
Wired Network
M2M Gateway
M2M Applications
Wireless Network
M2M Gateway

- An M2M area network comprises of machines (or M2M nodes) which have embedded hardware modules for sensing, actuation and communication.
- Various communication protocols can be used for M2M LAN such as ZigBee, Bluetooth, M-bus, Wireless M-Bus, Powerline communication(PLC), 6LoWPAN, IEEE 802.15.4 etc.., These protocols provide connectivity between M2M nodes within an M2M area network.
- The communication network provides connectivity to remote M2M area networks.The communication network can use either wired or wireless network . While the M2M are networks use either proprietary or non-IP based communication protocols, the communication network uses IP-based network. Since non-IP based protocols are used within M2M area network, the M2M nodes within one network cannot communicate with nodes in an external network.
- To enable the communication between remote M2M are network, M2M gateways are used.

Block diagram of an M2M gateway:

The below diagram shows a block diagram of an M2M gateway.



The communication between M2M nodes and the M2M gateway is based on the communication protocols which are naive to the M2M are network. M2M gateway performs protocol translations to enable IP-connectivity for M2M are networks. M2M gateway acts as a proxy performing translations from/to native protocols to/from Internet Protocol (IP). With an M2M gateway, each mode in an M2M area network appears as a virtualized node for external M2M area networks.

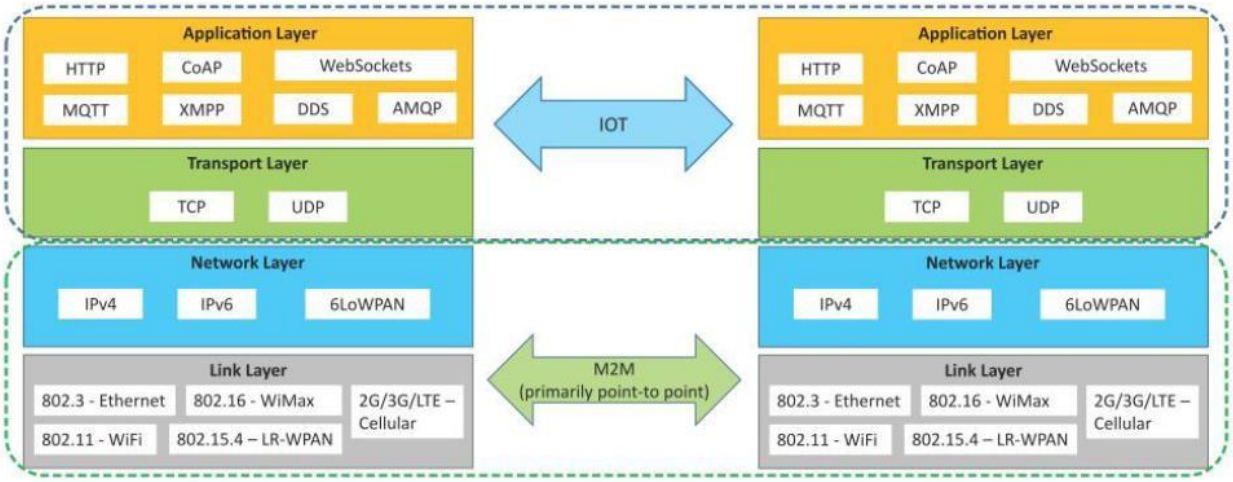
M2M data is gathered into point solutions such as Enterprise applications, service management applications, Remote monitoring applications. M2M has various applications domain such as Smart metering, Home automation, Industrial automation, smart Grids etc.,. M2M solution designs (such as data collection and

storage architecture and application apps) are specific to the M2M application domain.

3.2 Difference between IoT and M2M

i)Communication Protocols:

- M2M and IoT can differ in how the communication between the machines or devices happens.
- Commonly uses M2M protocols include ZigBee, Bluetooth, ModBus, M-Bus, WirelessM-Bustec.,
- In IoT uses HTTP, CoAP, WebSocket, MQTT,XMPP,DDS,AMQPetc.,
- M2M uses either proprietary or non-IP based communication
- The focus of communication in M2M is usually on the protocols below the network layer.
- The focus of communication in IoT is usually on the protocols above the network layer.
-



ii) Machines in M2M Vs Things inIoT:

The "Things" in IoT refers to physical objects have unique identifier and can sense and communicate with their external environment and user applications or their inter physical state. The unique identifier for the things in IoT are the IP addresses MAC addresses. Things have software components for accessing and processing and storing sensor information, or controlling actuators connectors.

• Machines in M2M will be homogenous whereas Things in IoT will be heterogeneous.(eg. Home automation, Fire alarms, Door Alarms, Lighting control devices etc.,.

• M2M systems, in contrast to IoT, typically have homogeneous machine types within an M2M area network.

iii) Hardware Vs Software Emphasis:

The emphasis of M2M is more on hardware with embedded modules

The emphasis of IoT is more on software.

IOT spends specialized software for sensor data collection, Data analysis and interfacing with the cloud through IP based communications

iv) Data Collection &Analysis

•M2M data is collected in point solutions and often in on-premises storage infrastructure.

•The data in IoT is collected in the cloud (can be public, private or hybrid cloud).

The analytics components analyses the data and stores the result in the cloud database. The centralized IoT data and analysis result are visualized with the cloud based applications. The centralized controller is aware of the status of all the end nodes and sends control commands to the nodes. Observer nodes can process information. And use it for various applications, however observer nodes do not perform any control functions.

v) Applications

M2M data is collected in point solutions and can be accessed by on-premises applications such as diagnosis applications, service management applications, and on premises enterprise applications.

• IoT data is collected in the cloud and can be accessed by cloud applications such as analytics applications, enterprise applications, remote diagnosis and management applications, etc.

The scale of collected data in IoT is massive , cloud based real Time and batch data analysis framework are used for data analysis

3.3 SOFTWARE DEFINED NETWORKING FOR IOT

- Software Defined Networking(SDN) is a networking architecture that separates the control plane from the data plane and centralizes the network controller.
- Software-based SDN controllers maintain a unified view of the network and make configuration, management and provisioning simpler.
- The underlying infrastructure in SDN uses simple packet forwarding hardware as opposed to specialized hardware in conventional networks.
- Control plane is the part of the network that carries the payload data traffic.

Limitations of Conventional Network:

i) Complex Network Devices:

Conventional Network are getting increasingly complex with more and more protocols being implemented. To improve link speeds and reliability. Interoperability is limited due to the lack of standard and open interfaces . Network devices use proprietary hardware and software and have slow product life cycles limiting innovation. The conventional network were well suited for static traffic patterns and had large number of protocols designed for specific applications

ii) Management Overhead:

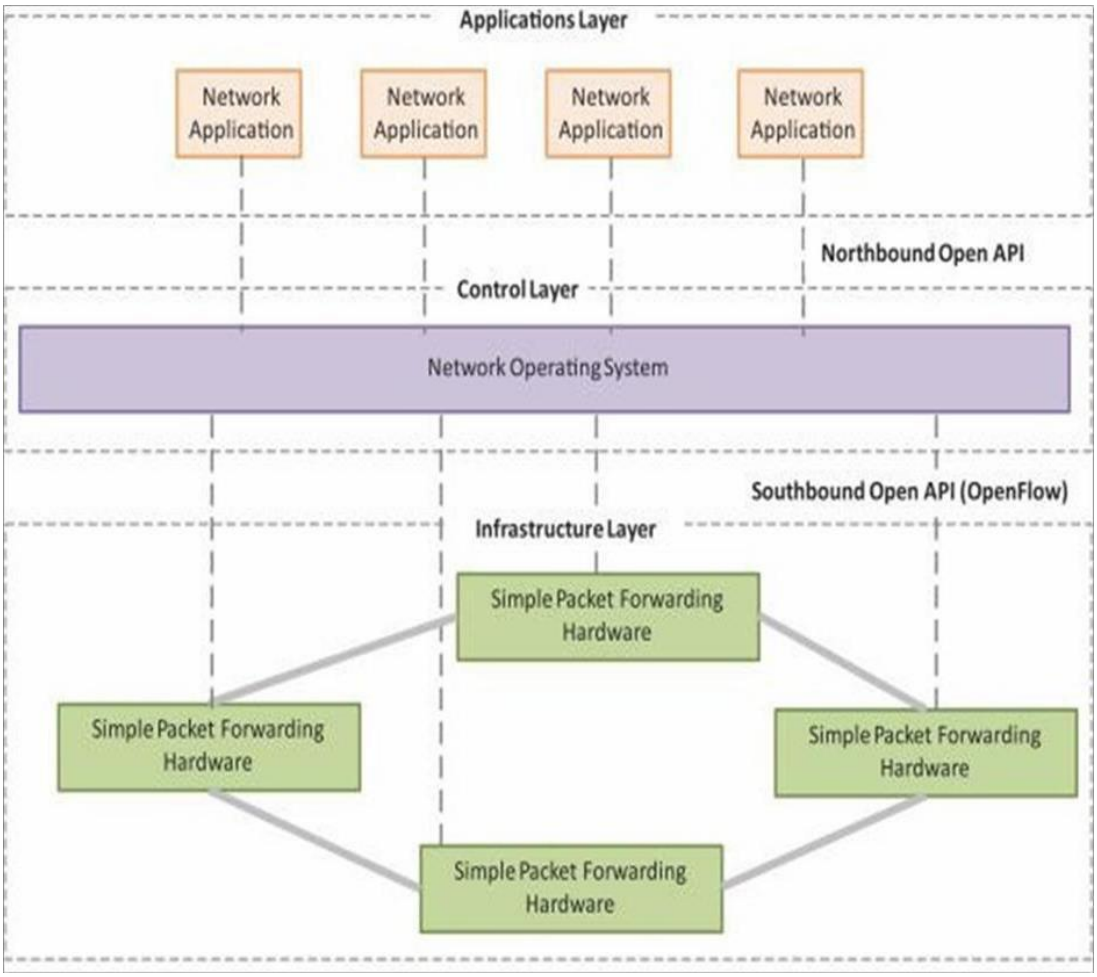
Conventional network involves significant management overhead. Network manager find it increasingly difficult to manage multiple network devices and interfaces from multiple vendors. Upgradation of network requires configuration changes in multiple devices (switches, routers, firewalls etc.,)

iii) Limited Scalability:

The virtualization technology used in cloud computing environment has increase the number of virtual host requiring network access. IoT applications hosted in the cloud are distributed across multiple virtual machines that require exchange of traffic. The analytics components of IoT applications run distributed algorithms on a large number of virtual machines and require huge amount of data exchange between virtual machines

SDN Architecture

Figure shows the SDN Architecture and SDN Layers in which the control and data planes are decoupled and the network controller is centralized



Key elements of SDN:

- **Centralized Network Controller**

  With decoupled control and data planes and centralized network controller, the network administrators can rapidly configure the network. SDN applications can be deployed through programmable open APIs. This speeds up innovation as the network administrator no longer need to wait for the device vendors to embed new features in their proprietary hardware

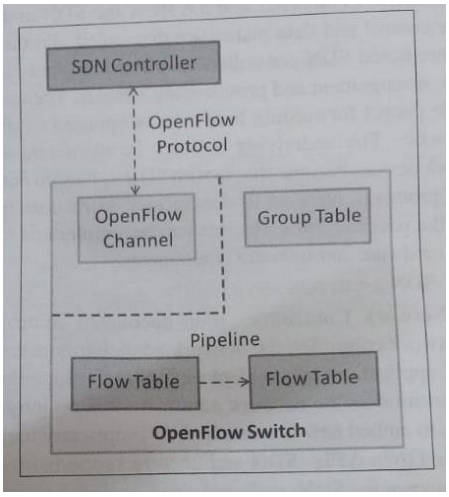
- **Programmable OpenAPIs**

  SDN architecture supports programmable open APIs for interface between the SDN application and control layers (Northbound interface). With these open APIs various network services can be implemented, such as routing, quality of service (QOS) access control etc.,

- **Standard Communication Interface(OpenFlow)**

  SDN architecture uses a standard communication interface between the control and infrastructure layers (Southbound interface). OpenFlow, which is defined by the Open Networking Foundation (ONF) is the broadly accepted SDN protocol for the Southbound interface. With

openflow, the forwarding plane of the network devices can be directly access and manipulated. Openflow uses the concept of flows to identify network traffic based on predefined match rules.



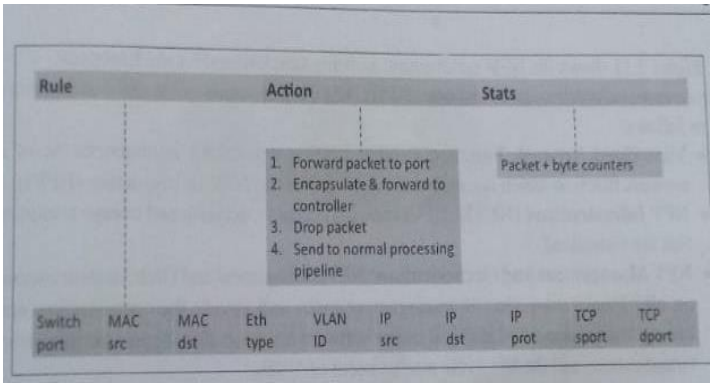


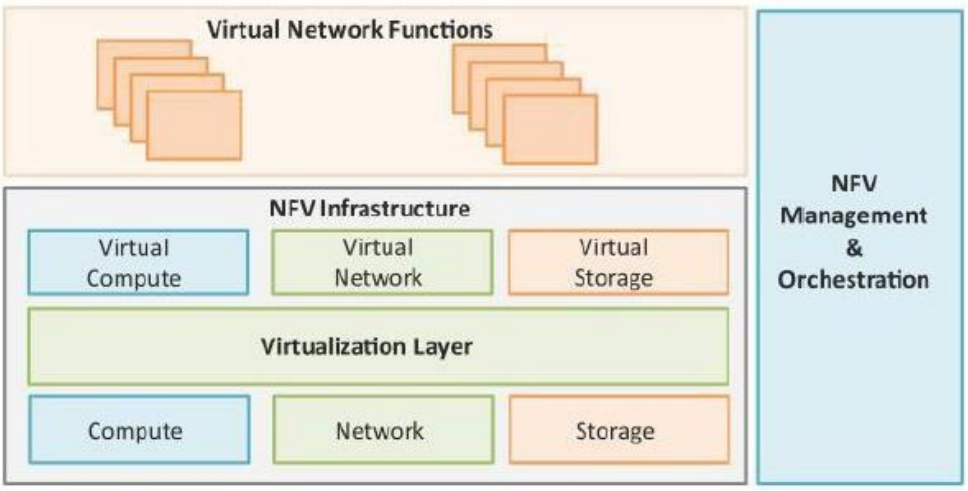
Figure 1 **Figure 2**

Figure 1 shows the components of an Openflow switch comprising of one or more flow table and group table

Figure 2 shows the example of Openflow table

3.4 NETWORK FUNCTION VIRTUALIZATION FOR IOT

- Network Function Virtualization (NFV) is a technology that leverages virtualization to consolidate the heterogeneous network.
- devices onto industry standard high volume servers, switches and storage.
- NFV is complementary to SDN as NFV can provide the infrastructure on which SDN can run
- NFV and SDN are mutually beneficial to each other but not dependent.
- Network functions can be virtualized without SDN similarly SDN can run without NFV



Key elements of NFV

• Virtualized Network Function (VNF):

VNF is a software implementation of a network function which is capable of running over the NFV Infrastructure (NFVI).

• NFV Infrastructure (NFVI):

NFVI includes compute, network and storage resources that are virtualized.

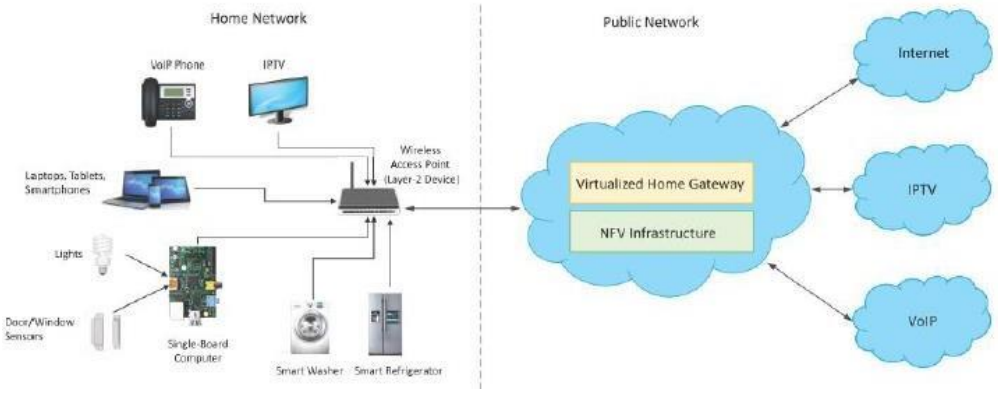
• NFV Management and Orchestration:

NFV Management and Orchestration focuses on all virtualization-specific management tasks and covers the orchestration and life-cycle management of physical and/or software resources that support the infrastructure virtualization, and the life-cycle management of VNFs.

NFV Use Case

- ❑ NFV can be used to virtualize the Home Gateway.
- ❑ The NFV infrastructure in the cloud hosts a virtualized Home Gateway.
- ❑ The virtualized gateway provides private IP addresses to the devices in the home.
- ❑ The virtualized gateway also connects to network services such as VoIP and IPTV.

❑



3.5 NEED FOR IOT SYSTEMS MANAGEMENT

Automating Configuration

Monitoring Operational & Statistical Data

Improved Reliability

System Wide Configurations

Multiple System Configurations

Retrieving & Reusing Configurations

4.1 Need for IoT Systems Management

Internet of Things (IoT) systems can have complex software, hardware and deployment designs including sensors, actuators, software and network resources, data collection and analysis services and user interfaces. IoT systems can have distributed deployments comprising of a number of IoT devices which collect data from sensors or perform actuation. Managing multiple devices within a single system requires advanced management capabilities. The need for managing IoT systems is described as follows:

- **Automating Configuration**: IoT system management capabilities can help in automating the system configurations. System management interfaces provide predictable and easy to use management capability and the ability to automate system configuration. Automation becomes even more important when a system consists of multiple devices or nodes. In such cases automating the system configuration ensures that all devices have the same configuration and variations or errors due to manual configurations are avoided.
- **Monitoring Operational & Statistical Data**: Operational data is the data which is related to the system's operating parameters and is collected by the system at runtime. Statistical data is the data which describes the system performance (e.g. CPU and memory usage). Management systems can help in monitoring operational and statistical data of a system. This data can be used for fault diagnosis or prognosis.
- **Improved Reliability**: A management system that allows validating the system configurations before they are put into effect can help in improving the system reliability.
- **System Wide Configuration**: For IoT systems that consist of multiple devices or nodes, ensuring system-wide configuration can be critical for the correct functioning of the system. Management approaches in which each device is configured separately (either through a manual or automated process) can result in system faults or undesirable outcomes. This happens when some devices are running on an old configuration while others start running on new configuration. To avoid this, system wide configuration is required where all devices are configured in a single atomic transaction. This ensures that the configuration changes are either applied to all devices or to none. In the event of a failure in applying the configuration to one or more devices, the configuration changes are rolled back. This 'all or nothing' approach ensures that the system works as expected.
- **Multiple System Configurations**: For some systems it may be desirable to have multiple valid configurations which are applied at different times or in certain conditions.
- **Retrieving & Reusing Configurations**: Management systems which have the capability of retrieving configurations from devices can help in reusing the configurations for

3.6 SIMPLE NETWORK MANAGEMENT PROTOCOL (SNMP)

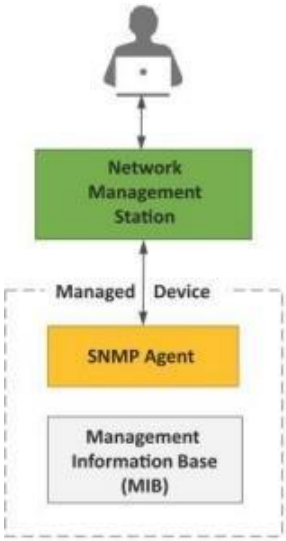
SNMP is a well-known and widely used network management protocol that allows monitoring and configuring network devices such as routers, switches, servers, printers, etc. • SNMP component include

• Network Management Station (NMS)

• Managed Device

• Management Information Base (MIB)

• SNMP Agent that runs on the device



3.6.1 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.

3.7 NETWORK OPERATOR REQUIREMENTS

Ease of use

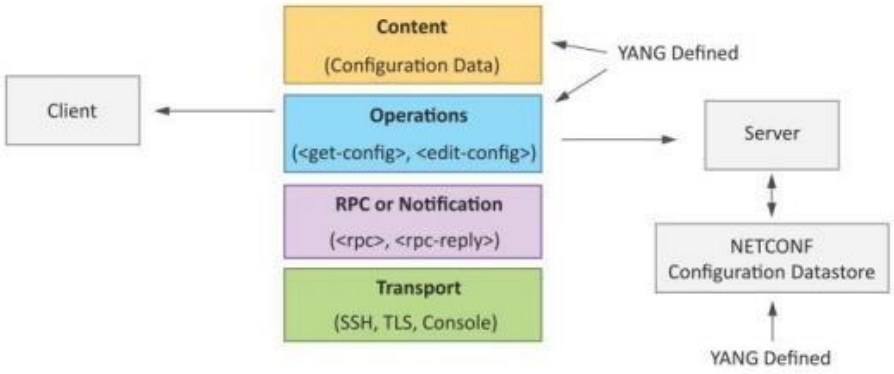
- Distinction between configuration and state data
- Fetch configuration and state data separately
- Configuration of the network as a whole
- Configuration transactions across devices
- Configuration deltas
- Dump and restore configurations
- Configuration validation
- Configuration database schemas
- Comparing configurations
- Role-based access control
- Consistency of access control lists:
- Multiple configuration sets
- Support for both data-oriented

- **Distinction between configuration and state data**: Configuration data is the set of writable data that is required to transform the system from its initial state to its current state. State data is the data which is not configurable. State data includes operational data which is collected by the system at runtime and statistical data which describes the system performance. For an effective management solution, it is important to make a clear distinction between configuration and state data.
- **Fetch configuration and state data separately**: In addition to making a clear distinction between configuration and state data, it should be possible to fetch the configuration and state data separately from the managed device. This is useful when the configuration and state data from different devices needs to be compared.
- **Configuration of the network as a whole**: It should be possible for operators to configure the network as a whole rather than individual devices. This is important for systems which have multiple devices and configuring them within one network wide transaction is required to ensure the correct operation of the system.
- **Configuration transactions across devices**: Configuration transactions across multiple devices should be supported.
- **Configuration deltas**: It should be possible to generate the operations necessary for going from one configuration state to another. The devices should support configuration deltas with minimal state changes.
- **Dump and restore configurations**: It should be possible to dump configurations from devices and restore configurations to devices.
- **Configuration validation**: It should be possible to validate configurations.
- **Configuration database schemas**: There is a need for standardized configuration database schemas or data models across operators.
- **Comparing configurations**: Devices should not arbitrarily reorder data, so that it is possible to use text processing tools such as *diff* to compare configurations.
- **Role-based access control**: Devices should support role-based access control model, so that a user is given the minimum access necessary to perform a required task.
- **Consistency of access control lists**: It should be possible to do consistency checks of access control lists across devices.
- **Multiple configuration sets**: There should be support for multiple configurations sets on devices. This way a distinction can be provided between candidate and active configurations.
- **Support for both data-oriented and task-oriented access control**: While SNMP access control is data-oriented, CLI access control is usually task oriented. There should be support for both types of access control.

3.8 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 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.

3.9 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.

YANG Module Example

• 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.

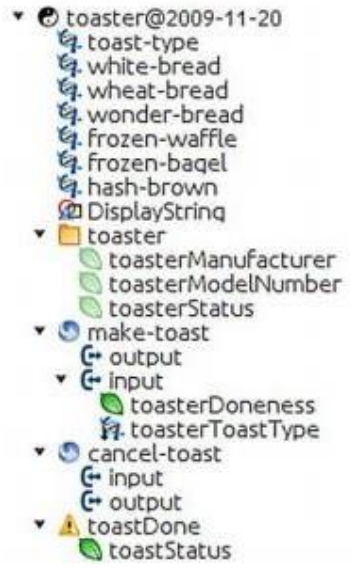
This YANG module is a YANG version of the toaster MIB

• 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’).



3.10 IoT Systems Management with NETCONF – YANG.

• Management System

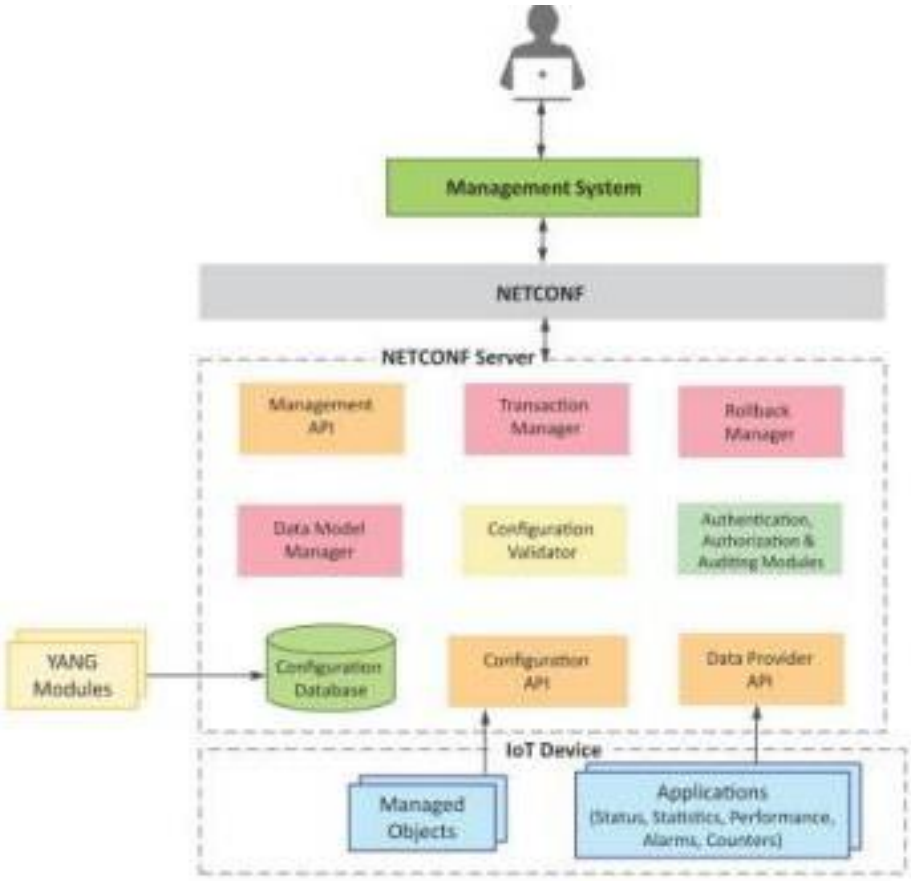
• Management API

• Transaction Manager

• Rollback Manager

• Data Model Manager

- Configuration Validator
- Configuration Database
- Configuration API
- Data Provider API



Part A-Multiple Choice Questions

[Separately discussed]

Part B- 8 Marks

1. Differentiate IoT and M2M in detail.
2. Make a note on Simple Network Management Protocol (SNMP)
3. Explain the role of NETCONF in IoT
4. Explain the role of YANG in IoT

Part C- 16 Marks

1. Explain how Software Defined Networking can be used for various levels of IoT ?
2. Describe how NFV can be used for virtualizing IoT devices.
3. Brief overview of the Network Operator Requirements.