ANNAMACHARYA
INSTITUTE OF TECHNOLOGY AND SCIENCES
(AUTONOMOUS)

Approved by AICTE, New Delhi & Permanent Affiliation to JNTUA, Anantapur.

Three B. Tech Programmes (CSE , ECE & CE) are accredited by NBA, New Delhi,Accredited by NAAC with 'A' Grade , Bangalore.

A-grade awarded by AP Knowledge Mission. Recognized under sections 2(f) & 12(B) of UGC Act 1956.

Venkatapuram Village, Renigunta Mandal, Tirupati, Andhra Pradesh-517520.

Department of Computer Science and Engineering

Academic Year 2023-24

IV. B.Tech I Semster

Enabling Technologies for Data Science & Analytics : IoT

(Common to CSE, AIDS)

(20A0E3601)

Prepared By

Mrs. N.Ashalatha., M.Tech.
Assistant Professor
Department of CSE, AITS

UNIT-I

1.Introduction and Concepts

1 .Introduction to Internet ofThings

Introduction

Definition and characteristic of IOT

2.Physical DesignofIOT

Things in IOT

IOT Protocols

3.Logical DesignOf IOT

IOT Functional Blocks

IOT Communication Models

IOT Communication API's

4. **IOT Enabling Technologies**

Wireless Sensor Network

Cloud Computing

Big Data Analytics

Communication Protocols

Embedded Systems

5. **Domain specific IOTs**

Introduction

Home Automation

Cities

Environment

Retail

Agriculture

Industry

Health &life style

Introduction to Internet of Things

Internet of Things (IOT) comprises things that have unique identities and areconnected to the internet. While many existing devices, such as networked computersor 4G-enabled mobile phones, already have some form of unique identities and arealso connected to the Internet, that are focus on IOT is in the configuration,
controlandnetworkingviatheInternetofdevicesor“things”thataretraditionallynotassociatedwiththeInternet.Theseincludedevicessuchasthermostats,utilitymeters , a Bluetooth connected headset , irrigation pumps and sensors, or control circuits foranelectriccar’sengine.
Internet of Things is a new revolution in the capabilities of the endpoints that areconnected to the Internet, and is being driven by the advancement in capabilities insensor networks, mobile devices, wireless communications and networking and cloudtechnologies.

Data:Rawandunprocesseddataobtained from IOTdevice/Systems
Information: Information is inferred from data by filtering, processing,categorizing,condensing and contextualizingdata.
Knowledge:knowledgeisinferredfrominformationbyorganizing&structuringinformationa nd is put into action toachievespecific objectives.

Fig1.1:Inferring informationandknowledgeinIOT

- The scope of IOT is limited to just connecting things (devices, appliances, machines)totheinternet.
- IOT allows these things to communicate and exchange data (control & information,thatcouldincludedataassociatedwiththeusers)whileexecutingmeaningfulapplicationstowards a common user ormachinegoal.
- Data itself does not have a meaning until it is contextualized processed into usefulinformation
- Application on IOT networks extract and create information from lower level data byfiltering,processing, categorizing,condensingandcontextualizingthedata.
- This information obtained is then organized and structured to infer knowledge aboutthesystemand/oritsusers,itsenvironment,anditsoperationsandprogresstowardsitsobjectives, allowingasmarterperformance.

ApplicationsofIOT

Home

- Smart lighting

- Smart Appliance
- Intrusions Detection

- Smoke/Gas Detector Cities
  - Smart Parking
  - Smart Roads
  - Structural Health Monitoring
  - Emergency Response

Environment

- Wealth Monitoring
- Air pollution Monitoring
- Noise pollution Monitoring
- Forest fire Detection

Energy

- Smart Grids
- Renewable Energy Systems
- Prognostics

Retail

- Inventory Management
- Smart Payments
- Smart Vending Machines

Logistics

- Route Generation & Scheduling
- Fleet Tracking
- Shipment Monitoring
- Remote Vehicle Diagnostics

Agriculture

- Smart Irrigation
- Green House Control

Industry

- Machine Diagnosis &Prognosis

Indoor Air Quality Monitoring

Health &Lifestyle

- Health &Fitness Monitoring
- Wearable Electronics

IOT has several applications such as smart lighting that adapt the lighting to suit the ambient conditions, Smart appliances that can be remotely monitored and controlled, intrusion detection systems, smarts make detectors ,etc.

For cities, IoT has applications such as smart parking systems that provide status updates on available slots, Smart lighting that helps in saving energy, smart roads
thatprovideinformationondrivingconditionsandstructuralhealthmonitoringsystem
.
For environment, IoT has applications such as weather monitoring, air and noise pollution ,forest fire detection and river flood detection systems.

For energy systems, IoT has applications such a sinclud in smart grids ,grid integration of renewable energy sources.
Forretaildomain, IoT has applicationssuchasinventorymanagement,smartpaymentsandsmartvendingmachines.
Foragriculture domain,IoT has applications such as smart irrigation systemsthathelpinsavingwater whileenhancingproductivityandgreenhousecontrolsystems.
Industrial applications of IoT include machine diagnosis and prognosis system thathelp in predicting faults and determining the cause of faults and indoor air qualitysystems.
For health and life style, IoT has applications such as health and fitness monitoringsystemsand wearable electronics.

Definition&characteristicsofIoT

TheInternetofThings(IoT)hasdefinedas

Definition:

A dynamic global network infrastructure with self-configuring capabilities based onstandard and interoperable communication protocols where physical and virtual "things" haveidentities, physical attributes, and virtual personalities and the intelligent interfaces, and areseamlessly integrated into the information network, often communicate data associated withusersandtheirenvironment's

- **Dynamic and Self-Adapting:** IoT devices and systems may have the capability todynamicallyadaptwiththechangingcontextsandtakeactionsbasedontheiroperatingconditions,usercontextorsensedenvironment.Forexample,thesurveillance cameras can adapt their modes based on whether it is day or night. In thisexample, the surveillance system is adapting itself based on the context and changing(e.g.,dynamic)conditions.
- **Self-Configuring:** IoT devices may have self-configuring capability, allowing a largenumber of devices to work together to provide certain functionality (such as weathermonitoring). These devices have theability configure themselves (in associate withthe IoT Infrastructure), setup the networking, and fetch latest software upgrades withminimal manual or user intervention.
- **Interoperable Communication Protocols:** IoT devices may support a number ofinteroperable communication protocols and can communicate with other devices andalsowith theinfrastructure.
- **Unique Identity:** Each IoT device has a unique identity and unique identifier (such asanIPaddress(or)aURI).IoTsystemsmayhaveintelligentinterfacewhichadapt

based on the context, allow communicating with users and environmental contexts.IoTdeviceinterfacesallowuserstoquery thedevices,monitortheirstatus,andcontrol them remotely in association with the control, configuration and managementinfrastructure.

- **Integrated into Information Network:** IoT devices are usually integrated into theinformation network that allows them to communicate and exchange data with otherdevices and systems. IoT devices can be dynamically discovered in the network, byother devices and/or the network, and have capability to describe themselves to otherdevices or user applications. For example, weather monitoring node can describe itsmonitoring capabilities to another connected node so that they can communicate andexchange data. Integration into the information network helps to make IoT systems"smarter" due to the collection intelligence of the individual devices in collaborationwiththeinfrastructure.

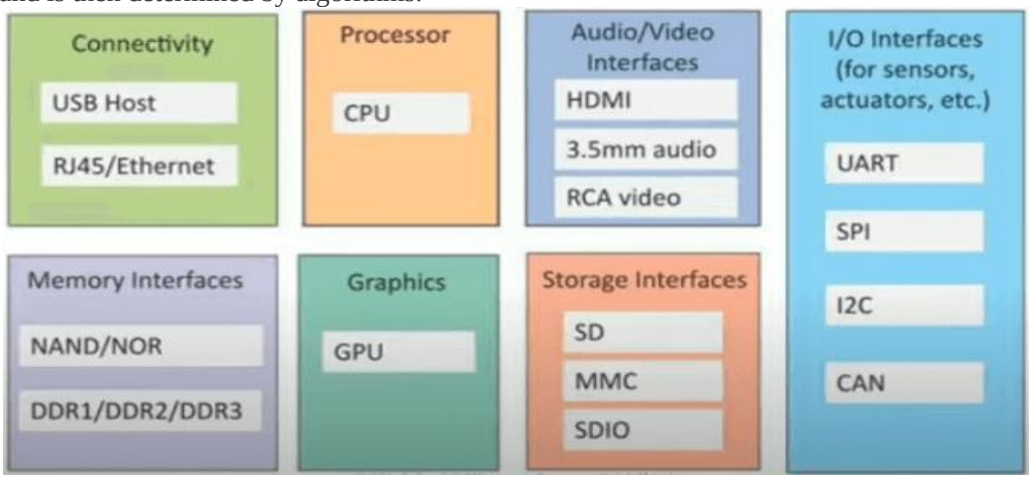
Physical Design of Internet of Things (IOT)

The **physical design** of an **IoT system is referred to as the Things/Devices and protocols that are used to build an IoT system. All these things/Devices are called Node Devices and every device has a unique identity that performs remote sensing, actuating,** and monitoring work. and the protocols that are used to establish communication between the Node devices and servers over the internet.

Physical Design of IoT

Things/Devices

Things/Devices are used to build a connection, process data, provide interfaces, provide storage, and provide graphics interfaces in an IoT system. All these generate data in a form that can be analyzed by an analytical system and program to perform operations and used to improve the system.
for example temperature sensor that is used to analyze the temperature generates the data from a location and is then determined by algorithms.



devices in IoT(Internet of things)

Connectivity

Devices like USB hosts and ETHERNET are used for connectivity between the devices and the server.

Processor

A processor like a CPU and other units are used to process the data. these data are further used to improve the decision quality of an IoT system.

Audio/Video Interfaces

An interface like HDMI and RCA devices is used to record audio and videos in a system.

Input/Output interface

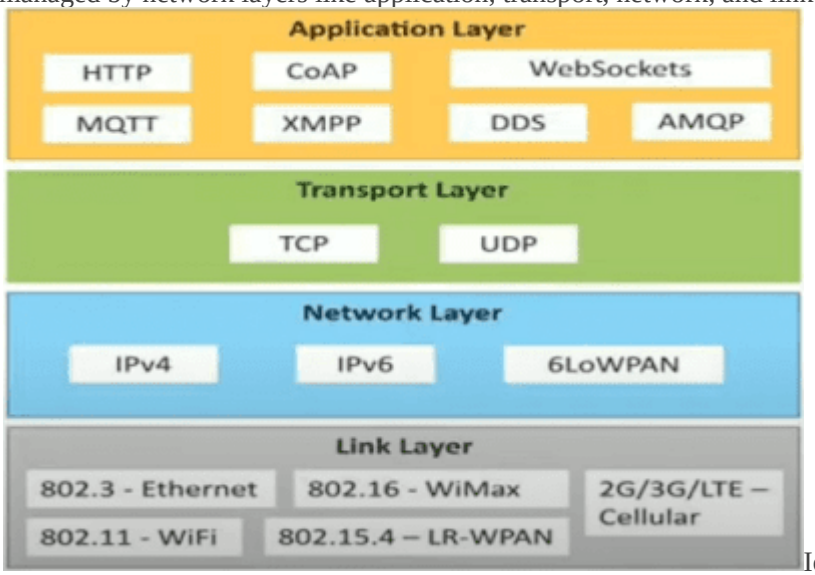
To give input and output signals to sensors, and actuators we use things like UART, SPI, CAN, etc.

Storage Interfaces

Things like SD, MMC, and SDIO are used to store the data generated from an IoT device.
Other things like DDR and GPU are used to control the activity of an IoT system.

IoT Protocols

These protocols are used to establish communication between a node device and a server over the internet. it helps to send commands to an IoT device and receive data from an IoT device over the internet. we use different types of protocols that are present on both the server and client side and these protocols are managed by network layers like application, transport, network, and link layer.



IoT(Internet of Things) protocols

Application Layer protocol

In this layer, protocols define how the data can be sent over the network with the lower layer protocols using the application interface. these protocols include HTTP, WebSocket, XMPP, MQTT, DDS, and AMQP protocols.

HTTP

Hypertext transfer protocol is a protocol that presents an application layer for transmitting media documents. it is used to communicate between web browsers and servers. it makes a request to a server and then waits till it receives a response and in between the request server does not keep any data between the two requests.

WebSocket

This protocol enables two-way communication between a client and a host that can be run on an untrusted code in a controlled environment. This protocol is commonly used by web browsers.

MQTT

It is a machine-to-machine connectivity protocol that was designed as a publish/subscribe messaging transport. and it is used for remote locations where a small code footprint is required.

Transport Layer

This layer is used to control the flow of data segments and handle error control. also, these layer protocols provide end-to-end message transfer capability independent of the underlying network.

TCP
The transmission control protocol is a protocol that defines how to establish and maintain a network that can exchange data in a proper manner using the internet protocol.

UDP
a user datagram protocol is part of an internet protocol called the connectionless protocol. this protocol is not required to establish the connection to transfer data.

Network Layer
This layer is used to send datagrams from the source network to the destination network. we use IPv4 and IPv6 protocols as host identification that transfers data in packets.

IPv4
This is a protocol address that is a unique and numerical label assigned to each device connected to the network. an IP address performs two main functions host and location addressing. IPv4 is an IP address that is 32-bit long.

IPv6
It is a successor of IPv4 that uses 128 bits for an IP address. it is developed by the IETF task force to deal with long-anticipated problems.

Link Layer
Link-layer protocols are used to send data over the network's physical layer. it also determines how the packets are coded and signaled by the devices.

Ethernet
It is a set of technologies and protocols that are used primarily in LANs. it defines the physical layer and the medium access control for wired ethernet networks.

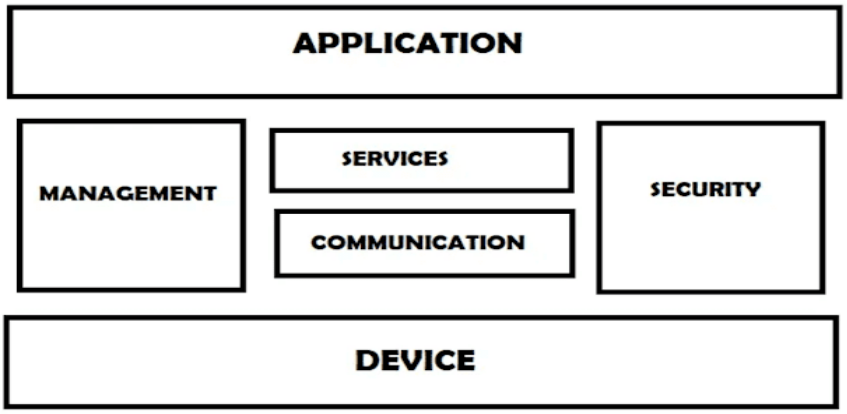
WiFi
It is a set of LAN protocols and specifies the set of media access control and physical layer protocols for implementing wireless local area networks.

Logical Design of IoT

Logical Design of the Internet of Things(IoT)

1. IoT Functional Blocks
2. IoT Communication Models
3. IoT Communication APIs

IoT Functional blocks
An IoT system consists of a number of functional blocks like Devices, services, communication, security, and application that provide the capability for sensing, actuation, identification, communication, and management.



IoT functional blocks

These functional blocks consist of devices that provide monitoring control functions, handle communication between host and server, manage the transfer of data, secure the system using authentication and other functions, and interface to control and monitor various terms.

Application

It is an interface that provides a control system that use by users to view the status and analyze of system.

Management

This functional block provides various functions that are used to manage an IoT system.

Services

This functional block provides some services like monitoring and controlling a device and publishing and deleting the data and restoring the system.

Communication

This block handles the communication between the client and the cloud-based server and sends/receives the data using protocols.

Security

This block is used to secure an IoT system using some functions like authorization, data security, authentication, 2-step verification, etc.

Device

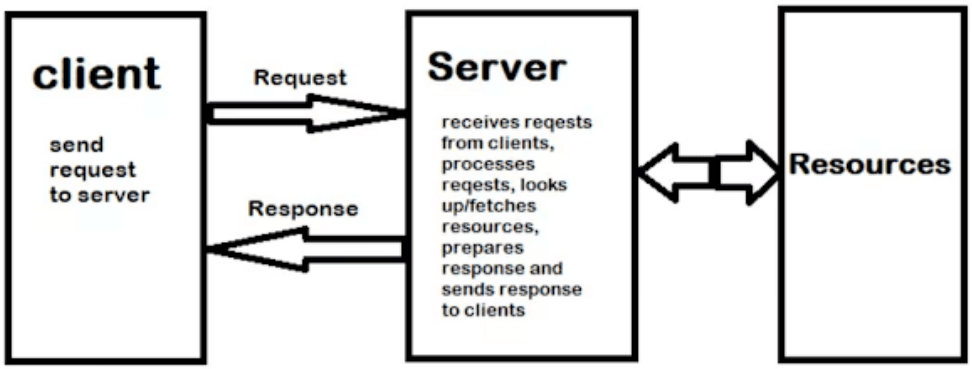
These devices are used to provide sensing and monitoring control functions that collect data from the outer environment.

IoT Communication Models

There are several different types of models available in an IoT system that is used to communicate between the system and server like the request-response model, publish-subscribe model, push-pull model, exclusive pair model, etc.

Request-Response Communication Model

This model is a communication model in which a client sends the request for data to the server and the server responds according to the request. when a server receives a request it fetches the data, retrieves the resources and prepares the response, and then sends the data back to the client.



Request-Response Communication Model

Request response communication model

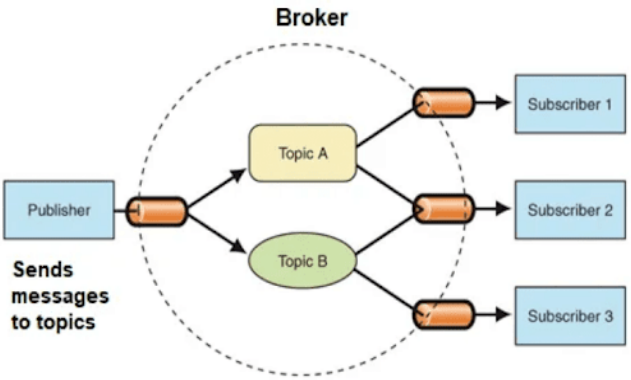
In simple terms, we can say that in the request-response model, the server sends the response equivalent to the request of the client. in this model, HTTP works as a request-response protocol between a client and server.

Example

When we search a query on a browser then the browser submits an HTTP request to the server and then the server returns a response to the browser(client).

Publish-Subscribe Communication Model

In this communication model, we have a broker between the publisher and the consumer. here publishers are the source of data but they are not aware of consumers. they send the data managed by the brokers and when a consumer subscribes to a topic that is managed by the broker and when the broker receives data from the publisher it sends the data to all the subscribed consumers.



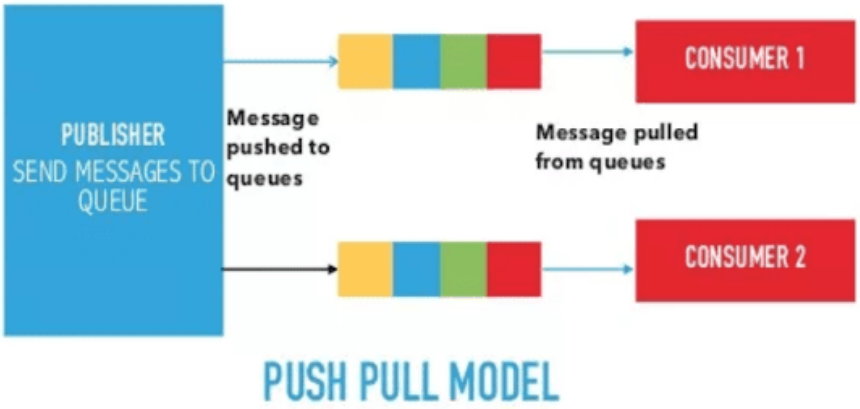
Published-subscribe communication model

Example

On the website many times we subscribed to their newsletters using our email address. these email addresses are managed by some third-party services and when a new article is published on the website it is directly sent to the broker and then the broker sends these new data or posts to all the subscribers.

Push-Pull Communication Model

It is a communication model in which the data push by the producers in a queue and the consumers pull the data from the queues. here also producers are not aware of the consumers.



Push Pull Model

Example

When we visit a website we saw a number of posts that are published in a queue and according to our requirements, we click on a post and start reading it.

Exclusive Pair Communication Model

It is a bidirectional fully duplex communication model that uses a persistent connection between the client and server. here first set up a connection between the client and the server and remain open until the client sends a close connection request to the server.

Exclusive Pair communication model

IoT communication APIs

These APIs like REST and WebSocket are used to communicate between the server and system in IoT.

REST-based communication APIs

Representational state transfer(REST) API uses a set of architectural principles that are used to design web services. these APIs focus on the systems' resources that how resource states are transferred using the request-response communication model. This API uses some architectural constraints.

Client-server

Here the client is not aware of the storage of data because it is concerned about the server and similarly the server should not be concerned about the user interface because it is a concern of the client. and this

separation is needed for independent development and updating of the server and client. no matter how the client is using the response of the server and no matter how the server is using the request of the client.

Stateless

It means each request from the client to the server must contain all the necessary information to understand the server. because if the server can't understand the request of the client then it can't fetch the requested data in a proper manner.

Cacheable

In response, if the cache constraints are given then a client can reuse that response in a later request. it improves the efficiency and scalability of the system without loading extra data.
A RESTful web API is implemented using HTTP and REST principles.

WebSocket-based communication API

This type of API allows bi-directional full-duplex communication between server and client using the exclusive pair communication model. This API uses full-duplex communication so it does not require a new connection setup every time when it requests new data. WebSocket API begins with a connection setup between the server and client and if the WebSocket is supported by the server then it responds back to the client with a successful response after the setup of a connection server and the client can send data to each other in full-duplex mode.
this type of API reduces the traffic and latency of data and makes sure that each time when we request new data it cannot terminate the request.

Internet of Things (IoT) Enabling Technologies

IoT(internet of things) enabling technologies are

1. Wireless Sensor Network
2. Cloud Computing
3. Big Data Analytics
4. Communications Protocols
5. Embedded System

1. Wireless Sensor Network(WSN) :

A **WSN** comprises distributed devices with sensors which are used to monitor the environmental and physical conditions. A **wireless sensor network** consists of end nodes, routers and coordinators. End nodes have several sensors attached to them where the data is passed to a coordinator with the help of routers. The coordinator also acts as the gateway that connects WSN to the internet.
Example –

- Weather monitoring system
- Indoor air quality monitoring system
- Soil moisture monitoring system
- Surveillance system
- Health monitoring system

2. Cloud Computing :

It provides us the means by which we can access applications as utilities over the internet. Cloud means something which is present in remote locations.
With Cloud computing, users can access any resources from anywhere like databases, webservers, storage, any device, and any software over the internet.

Characteristics –

1. Broad network access
2. On demand self-services
3. Rapid scalability
4. Measured service
5. Pay-per-use

Provides different services, such as –

- **IaaS (**Infrastructure as a service**)**
  Infrastructure as a service provides online services such as physical machines, virtual machines, servers, networking, storage and data center space on a pay per use basis. Major IaaS providers are

Google Compute Engine, Amazon Web Services and Microsoft Azure etc.
Ex : Web Hosting, Virtual Machine etc.

- **PaaS (**Platform as a service**)**
  Provides a cloud-based environment with a very thing required to support the complete life cycle of building and delivering West web based (cloud) applications – without the cost and complexity of buying and managing underlying hardware, software provisioning and hosting. Computing platforms such as hardware, operating systems and libraries etc. Basically, it provides a platform to develop applications.
  Ex : App Cloud, Google app engine
- **SaaS (**Software as a service**)**
  It is a way of delivering applications over the internet as a service. Instead of installing and maintaining software, you simply access it via the internet, freeing yourself from complex software and hardware management.
  SaaS Applications are sometimes called web-based software on demand software or hosted software.
  SaaS applications run on a SaaS provider's service and they manage security availability and performance.
  Ex : Google Docs, Gmail, office etc.

3. Big Data Analytics :

It refers to the method of studying massive volumes of data or big data. Collection of data whose volume, velocity or variety is simply too massive and tough to store, control, process and examine the data using traditional databases.
Big data is gathered from a variety of sources including social network videos, digital images, sensors and sales transaction records.
Several steps involved in analyzing big data –

1. Data cleaning
2. Munging
3. Processing
4. Visualization

Examples –

- Bank transactions
- Data generated by IoT systems for location and tracking of vehicles
- E-commerce and in Big-Basket
- Health and fitness data generated by IoT system such as a fitness bands

4. Communications Protocols :

They are the backbone of IoT systems and enable network connectivity and linking to applications. Communication protocols allow devices to exchange data over the network. Multiple protocols often describe different aspects of a single communication. A group of protocols designed to work together is known as a protocol suite; when implemented in software they are a protocol stack.
They are used in

1. Data encoding
2. Addressing schemes

5. Embedded Systems :

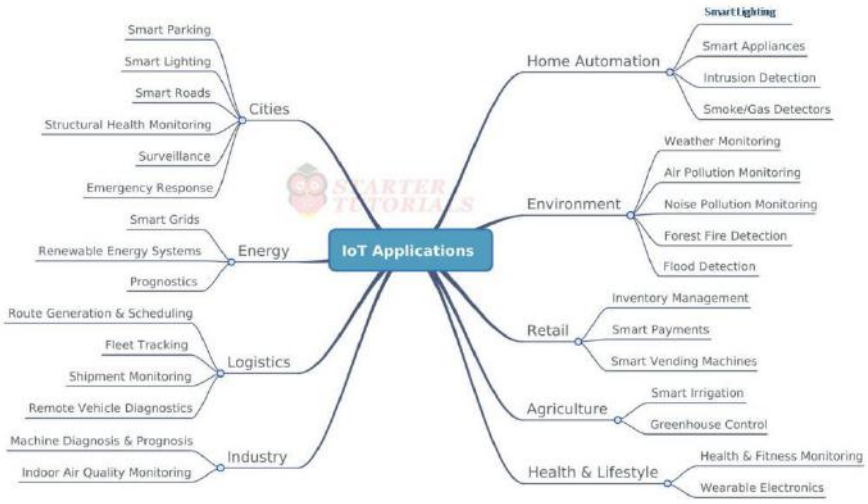
It is a combination of hardware and software used to perform special tasks.
It includes microcontroller and microprocessor memory, networking units (Ethernet Wi-Fi adapters), input output units (display keyword etc.) and storage devices (flash memory).
It collects the data and sends it to the internet.
Embedded systems used in
Examples –

1. Digital camera
2. DVD player, music player
3. Industrial robots
4. Wireless Routers etc.

Domain Specific IoT Applications

IoT applications span a wide range of domains like:

- Home Automation
- Smart Cities
- Environment
- Energy systems
- Retail
- Logistics
- Industry
- Agriculture
- Health

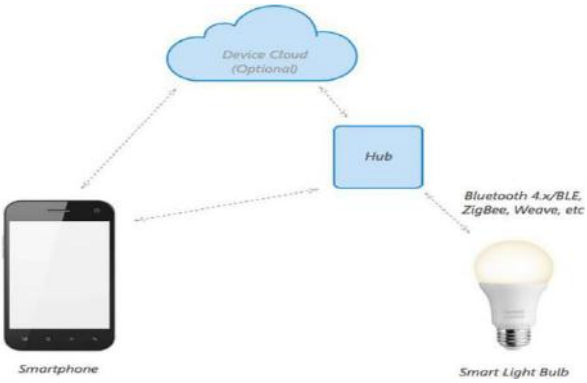


Watch this video to learn about iot applications in home automation, smart cities and environment domains:

Home Automation

Smart Lighting

Smart lighting for home helps in saving the energy by adapting the lighting to the ambient conditions. Energy can be saved by sensing human movements and their environment. Wireless and Internet connected lights can be operated remotely using mobile or web application.



Smart Appliances

Smart appliances makes the management easier and also provide status information to the users remotely. For example, a smart refrigerator can keep track of items and notify the user when a item is low on stock. Examples of smart appliances are TVs, refrigerators, music systems, washing machines, etc.

Intrusion Detection

Home intrusion detection systems use cameras and sensors to detect intrusions and for raising alerts. Alerts can be sound, SMS or email sent to the user. An advanced system can even send an image or a short video clip related to the intrusion event.

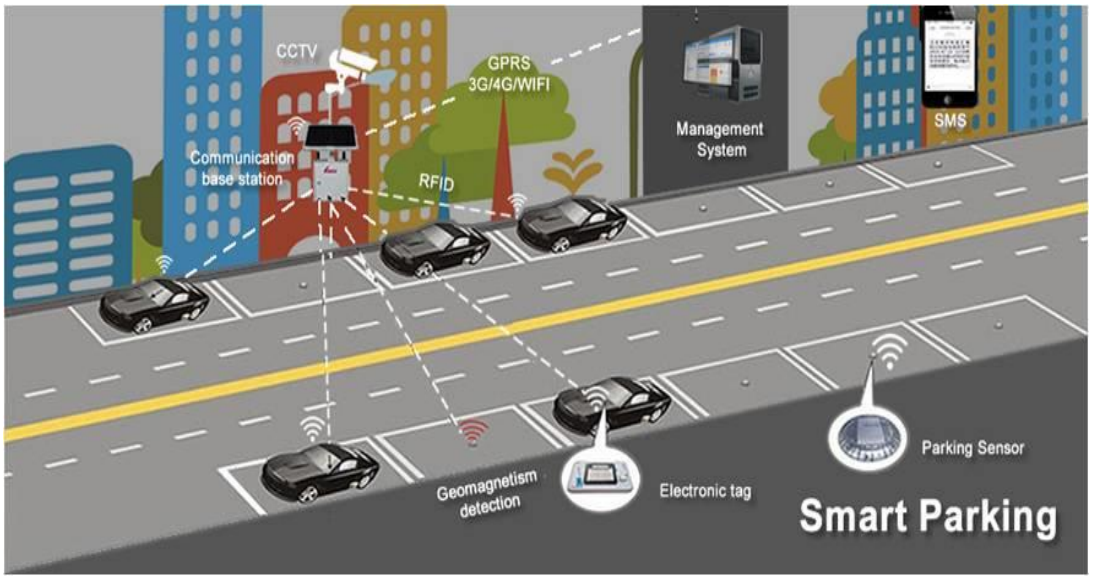
Smoke/Gas Detection

Smoke detectors installed at home can detect smoke and alert the users. Smoke detectors use optical detection, ionization, or air sampling techniques to detect smoke. Gas detectors can detect harmful gases like CO or LPG. These detectors can send alerts in the form of email, SMS, or voice.

Smart Cities

Smart Parking

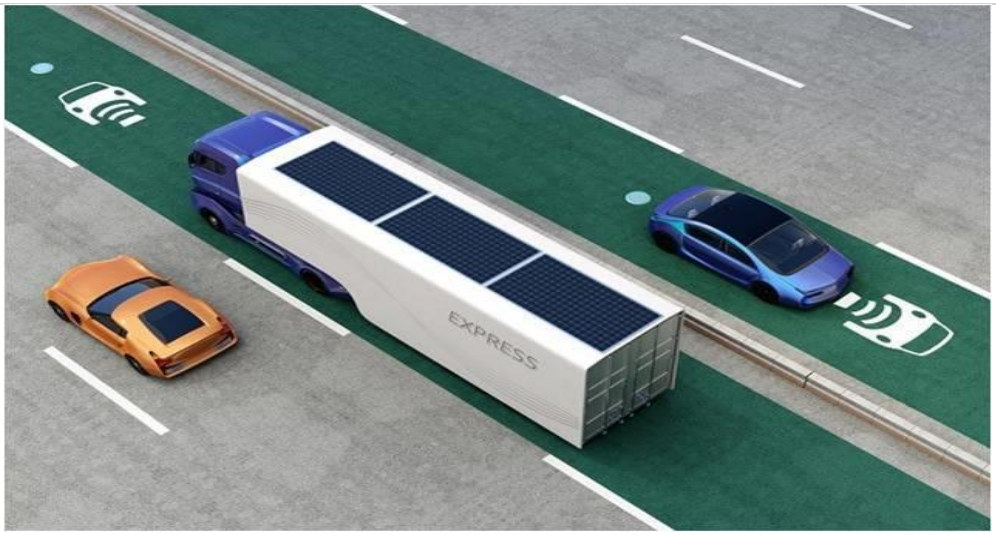
Smart parking makes the search for parking space easier and convenient for drivers. In smart parking, sensors are used for each parking slot, to detect whether the slot is occupied or not. This information is aggregated by local controllers and sent over the Internet to the database. Drivers can use an application to know about empty parking slots.



Smart Lighting

Smart lighting systems for roads, parks, and buildings can help in saving energy. Smart lighting allows lighting to be dynamically controlled and also adaptive to the ambient conditions. Smart lights connected to the Internet can be controlled remotely to configure lighting intensity and lighting schedule.

Smart Roads



Smart roads equipped with sensors can alert the users about poor driving conditions, traffic congestion, and accidents. Information sensed from the roads can be sent via Internet to applications or social media. This helps in reducing traffic jams.

Structural Health Monitoring

A network of sensors are used to monitor the vibration levels in the structures. Data from the sensors is analyzed to assess the health of the structures. By analyzing the data it is possible

to detect cracks, locate damages to the structures and also calculate the remaining life of the structure.

Surveillance

Surveillance of infrastructure, public transport and events in cities is required to ensure safety and security. City wide surveillance requires a large network of connected cameras. The video feeds from the cameras can be aggregated in cloud-based storage. Video analytics applications can be used to search for specific patterns in the collected feeds.

Emergency Response

IoT systems can be used to monitor buildings, gas and water pipelines, public transport and power substations. These systems provides alerts and helps in mitigating disasters. Along with cloud-based applications IoT systems helps to provide near real-time detection of adverse events.

Environment

Weather Monitoring

IoT-based weather monitoring systems use different sensors to gather data. That data is sent to the cloud-based storage. The collected can be analyzed and visualized with applications. Weather alerts can be subscribed by users from such applications.

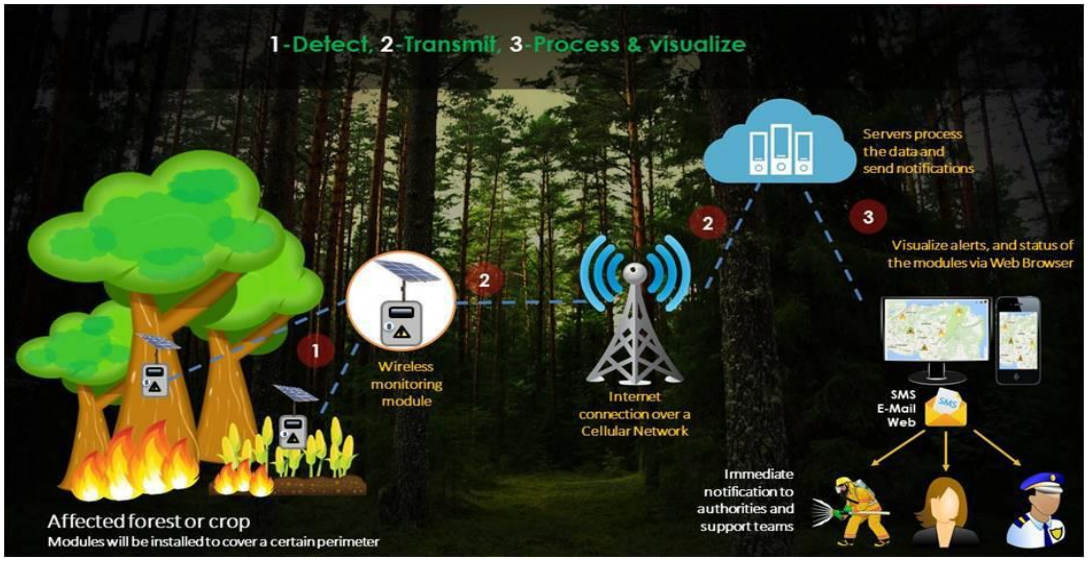
AirPollution Monitoring

IoT-based air pollution monitoring systems can monitor harmful gas emissions by factories and vehicles using gaseous and meteorological sensors. The collected data can be analyzed to take decisions on pollution control approaches.

Noise Pollution Monitoring

IoT-based noise pollution monitoring systems use a number of noise pollution monitoring systems that are deployed at different places in the city. The data on noise levels from the stations is collected on servers or in the cloud. The collected data can be analyzed to generate noise maps.

Forest Fire Detection



IoT-based forest fire detection systems use number of nodes deployed at various locations in the forest. Each monitoring node collects data about ambient conditions. This data will be collected and analyzed for the presence of fire and corresponding people will be alerted.

River Floods Detection

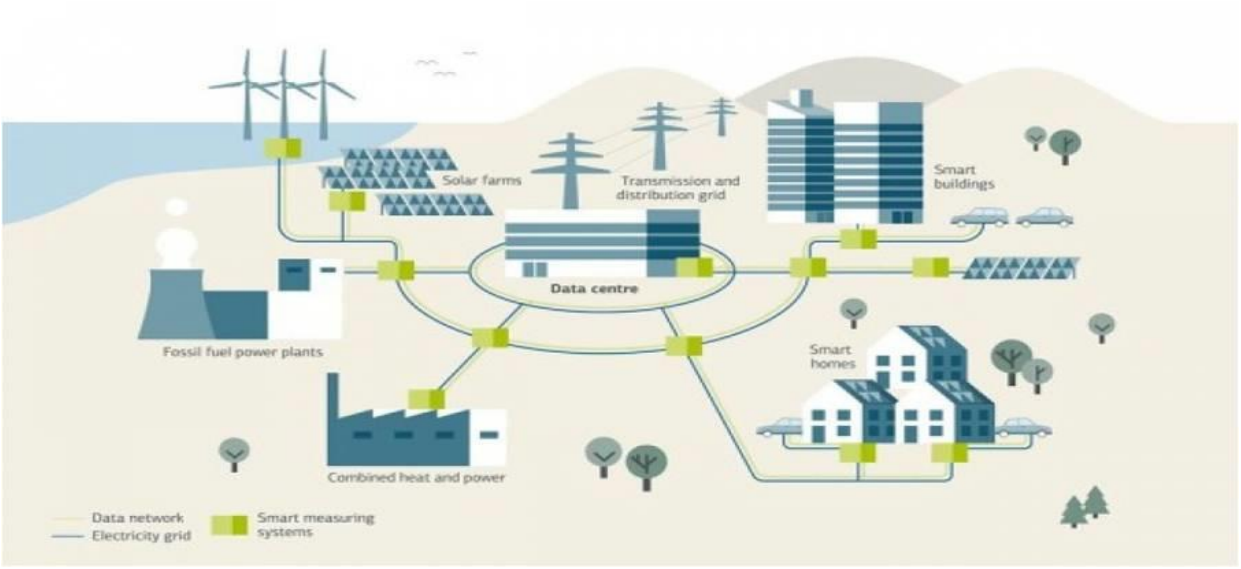
IoT-based flood monitor systems use number of sensor nodes to monitor the water level. Data from the sensors is aggregated on the server or in the cloud. Monitoring applications raise alerts in case of rapid increase in water level or when rapid flow rate is detected.

Watch this video to learn about IoT applications in energy, retail, logistics, agriculture, industry, health and lifestyle domains:

Energy

Smart Grids

Smart grid is a data communications network integrated with the electrical grid. Smart grid technology provides predictive information and recommendations to utilities, their suppliers and consumers, and how best to manage power. Smart meters can capture real-time power consumption and allows to manage power distribution remotely.



Renewable Energy Systems

Renewable energy sources (like solar and wind) produce variable output. Variable output produces local voltage swings that can impact power quality. IoT-based systems integrated with the transformers measures how much power is fed into the grid.

Prognostics

Energy systems have a large number of critical components whose health is essential for working correctly. IoT-based monitoring systems allows for the data to be gathered about these critical components. Analysis of massive amounts of data gathered by sensors can provide predictions for the impending failures.

Retail

Inventory Management

The inventory in a store or warehouse can be managed by using IoT. The products or items in the store can be attached with RFID tags. By using the RFID tags, the RFID reader or software can automatically show the number of items in the store or warehouse. If a product goes out of stock a notification can be sent to the store owner automatically.

Smart Payments

Now-a-days new types of payments are coming into picture like QR codes, NFC, contact less technologies etc. These technologies enables smart payments.

Smart Vending Machines

A smart vending machine contains several items. A consumer can insert money and get the item they want as shown in the image below. Several sensors can be attached to these vending machines such that whenever an item quantity is less, the owner of that machine will be automatically notified so that the owner can be arrangements to get that item beforehand.

Also, the vending machines can maintain the history of the consumers. So, when a consumer visits the vending machine next day, it can suggest the same item that the consumer purchased before.

Logistics

Route Generation and Scheduling

While delivering packages to various locations, different sensors can be fixed in those routes and they can be monitored remotely through an application. By looking at the data sent by the sensors, the delivery company can automatically know which routes are less congested and schedule the delivery of packages in such routes.

Fleet Tracking

A delivery company will have several delivery personnel working with them. Different people will use different vehicles for delivering the packages. Sensors can be fixed to those vehicles and their location can be tracked to know how long will it take to deliver the package.

Shipment Monitoring

The packages can be fixed with RFID tags or other form of remote tracking sensors to send data periodically to a server via Internet. The delivery company can use that data to track where the package is and update the user about the remaining time that will be needed to deliver the package.

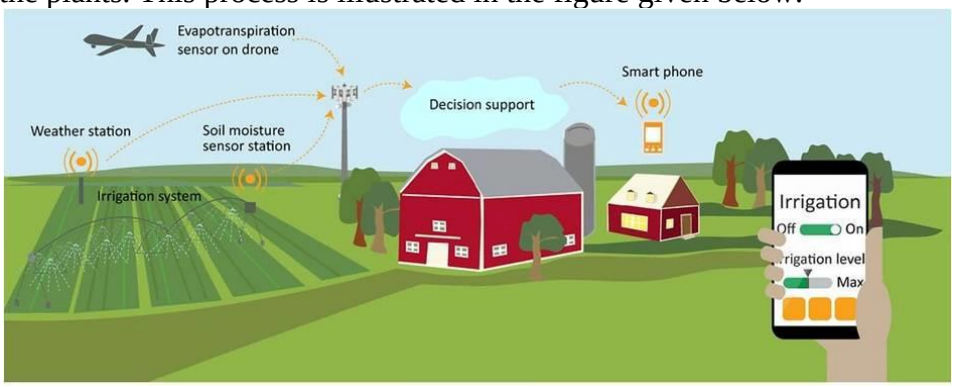
Remote Vehicle Diagnostics

A vehicle rental company can fix sensors into the vehicles before giving them for rent to the customers. The company can check the data sent by the sensors to know the current location of the vehicle and easily track them.

Agriculture

Smart Irrigation

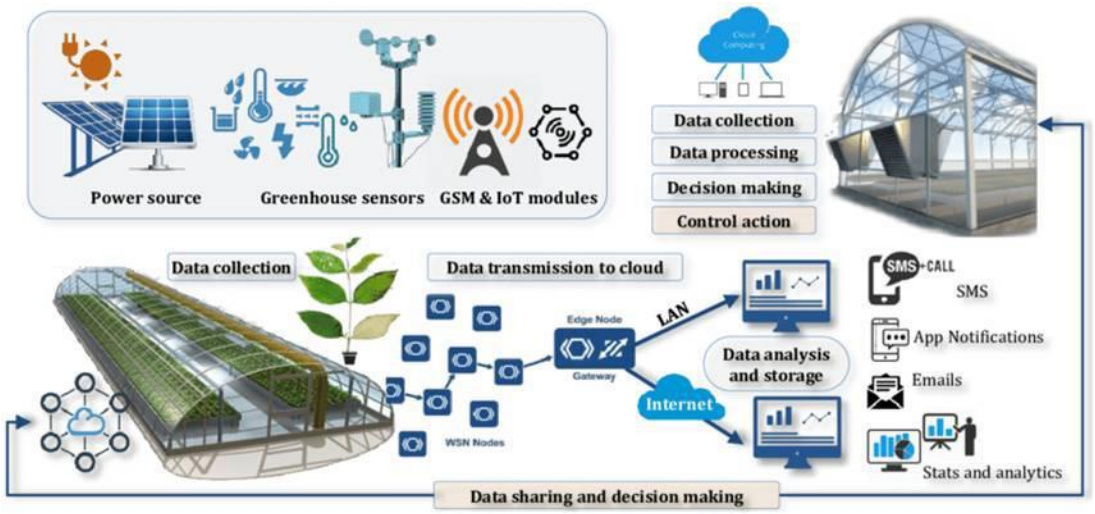
Irrigation refers to the watering of plants. By using different sensors like temperature sensor, humidity sensor, soil moisture sensor, etc., data can be collected about the soil and the environment and let the framer know when to turn on the water sprinklers to provide water to the plants. This process is illustrated in the figure given below.



Green House Control

A green house is an artificial field that can be grown inside buildings or on the roof tops. It is a controlled environment in which several types of sensors are fixed to gather data about the soil, environment and other parameters.

The data from the green house is aggregated at a local gateway and sent to the server via Internet. The data at the server is analyzed and appropriate alerts are sent to the owner of the green house. This process is illustrated in the figure below.



Industry

Machine Diagnosis & Prognosis

The machines used in the industry can be fixed with sensors. The data from the sensors can be used to diagnose the machines. We can know if the machine is working up to the expected performance or not. The data analysis will also let the owner of the machine know when the life of machine will be over.

Indoor Air Quality Monitoring

The quality of air for the working personnel inside the industry is also important. Often times leakage of dangerous gases leads to the death of industry personnel. Sensors can be fixed at different location to monitor the working environment for any leakage of hazardous gases and notify the appropriate personnel to deal with it.

Health & Lifestyle

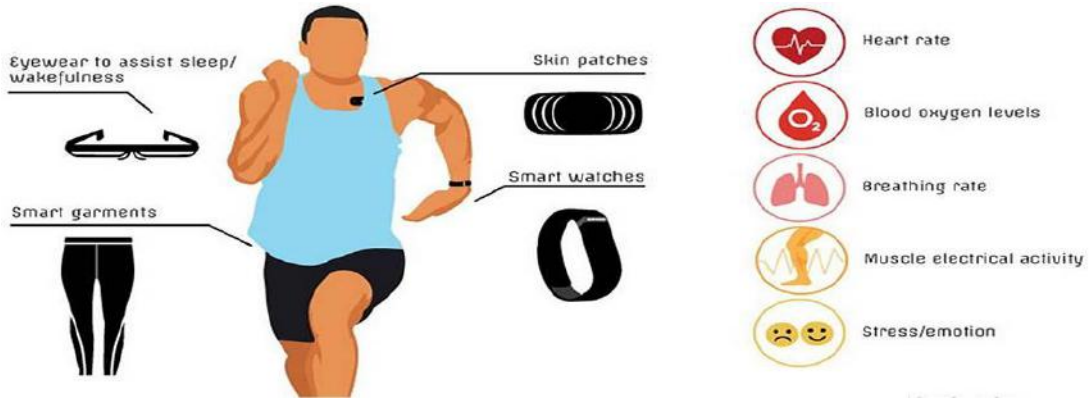
Health and Fitness Monitoring

With the advent of IoT remote healthcare has become an viable option for attending to patients. There is no need for patient to visit hospital for every minor health problem.

The doctor can attend to such patients from a remote location. Different sensors can be fixed on near the patient to monitor the health vitals of that patient. The data sent by the sensors is monitored by the doctor and appropriate decisions are made.

Wearable Electronics

Now-a-days there are different types of wearables available in the market to monitor health and lifestyles. Some examples of such wearables are smart watches, smart glasses, smart patches, smart garments, etc., as shown in the below figure.



UNIT-II

1. Introduction M2M

2. DifferencebetweenIoTandM2M

3. SDVand NFVfor IoT

3.1.Softwaredefinednetworking

3.2NetworkFunctionVisualization

4. IoT System Management with NETCONF-YANG

4.1. Need for IoT Systems Management

4.2. Simple Network Management protocol(SNMP)

4.3. Limitations of SNMP

4.4.Network Operator Requirement

4.5.IoT Systems Management with NETCONF-YANG

NETOPEER

2.1.Introduction TO M2M

- Machine-to-Machine (M2M) refers to networking of machines (or devices) for thepurposeof remotemonitoringandcontrol anddata exchange.
- AnM2Mareanetworkcomprisesofmachines(orM2Mnodes)whichhaveem beddedhardwaremodules forsensing,actuation and communication.
- Various communication protocols can be used for M2M local area networks such asZigBee, Bluetooh, ModBus, M-Bus, Wirless M-Bus, Power Line Communication(PLC),6LoWPAN,IEEE802.15.4, etc.
- Thecommunication networkprovides connectivityto remoteM2M areanetworks.
- Thecommunicationnetworkcanuseeither wiredorwirelessnetworks(IPbased).
- WhiletheM2Mareanetworksuseeitherproprietaryornon-IPbasedcommunicationprotocols,thecommunication networkusesIP-based networks.

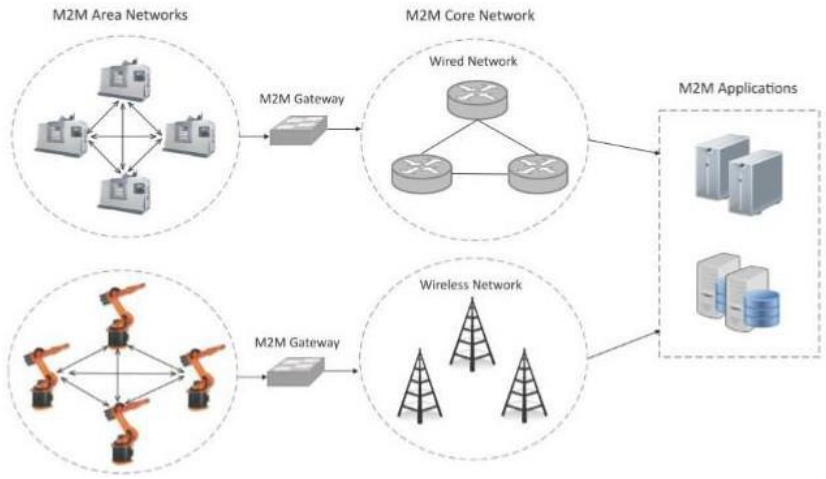


Fig:M2MSystemArchitecture

M2Mgateway

- Sincenon-IPbasedprotocolsareusedwithinM2Mareanetworks,theM2Mnodeswithin onenetworkcannotcommunicate with nodesin an externalnetwork.
- ToenablethecommunicationbetweenremoteM2Mareanetworks,M2Mgat ewaysareused.

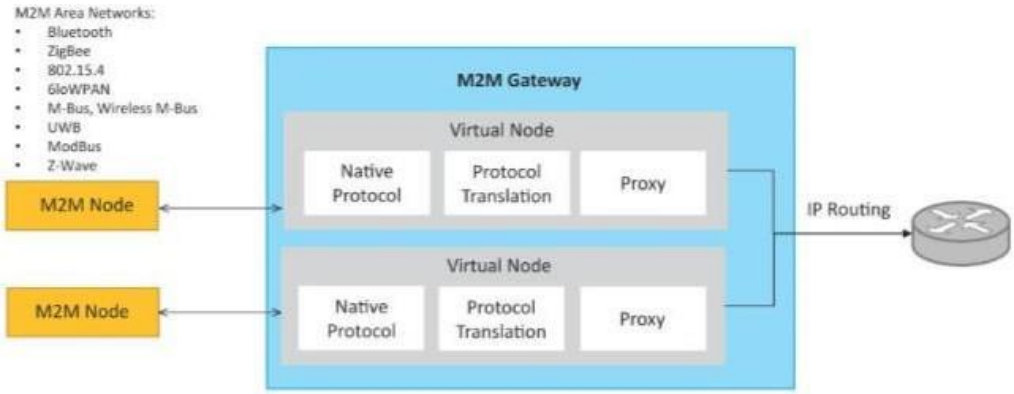


Fig: BlockDiagramof anM2Mgateway

DifferencebetweenIoTandM2M

CommunicationProtocols

- M2M and IoT can differ in how the communication between the machines or deviceshappens.
- M2Museseitherproprietaryornon-IPbasedcommunicationprotocolsforcommunicationwithin theM2M areanetworks.

Machinesin M2MvsThingsin IoT

- The "Things" in IoT refers to physical objects that have unique identifiers and cansense and communicate with their external environment (and user applications) ortheirinternal physical states.
- M2M systems, in contrast to IoT, typically have homogeneous machine types withinanM2M areanetwork.

HardwarevsSoftwareEmphasis

- WhiletheemphasisofM2Mismoreonhardwarewithembeddedmodules,the emphasisofIoT ismore on software.

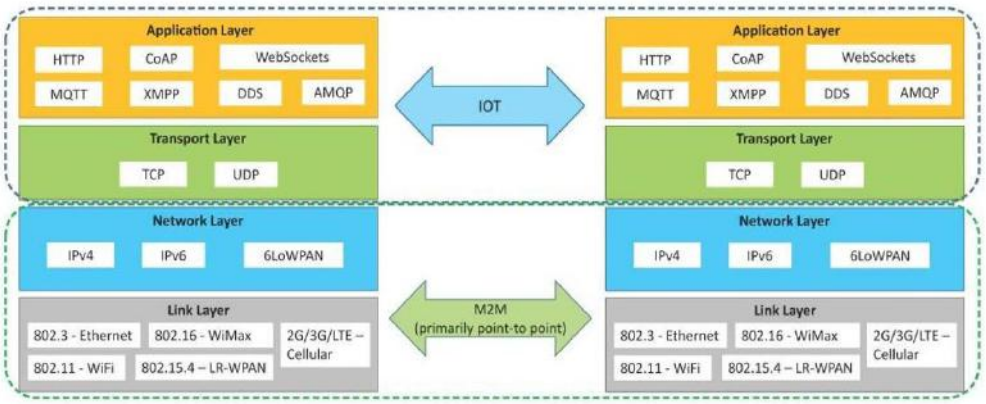
DataCollection & Analysis

- M2Mdataiscollectedinpointsolutionsandofteninon-premisesstorageinfrastructure.
- In contrast to M2M, the data in IoT is collected in the cloud (can be public, private orhybridcloud).

Applications

M2Mdataiscollectedinpointsolutionsandcanbeaccessedbyon-premisesapplications such as diagnosis applications, service management applications, and onpremisisenterpriseapplications.

- IoT data is collected in the cloud and can be accessed by cloud applications such asanalyticsapplications,enterpriseapplications,remotediagnosisandmanagementapplications,etc.



Software defined Networking(SDN)

SDN stands for Software Defined Network which is a networking architecture approach. It enables the control and management of the network using software applications. Through Software Defined Network (SDN) networking behavior of the entire network and its devices are programmed in a centrally controlled manner through software applications using open APIs.

To understand software-defined networks, we need to understand the various planes involved in networking.

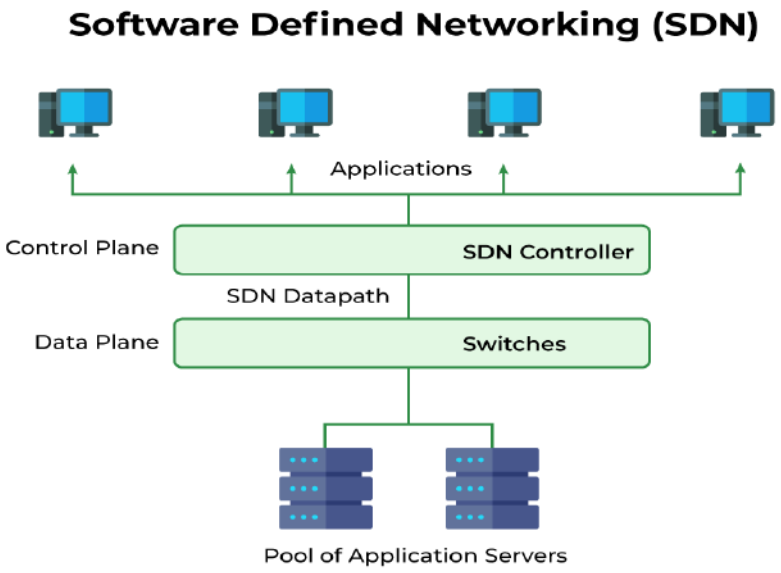
1. Data Plane
2. Control Plane

Data plane: All the activities involving as well as resulting from data packets sent by the end-user belong to this plane. This includes:

- Forwarding of packets.
- Segmentation and reassembly of data.
- Replication of packets for multicasting.

Control plane: All activities necessary to perform data plane activities but do not involve end-user data packets belong to this plane. In other words, this is the brain of the network. The activities of the control plane include:

- Making routing tables.
- Setting packet handling policies.



Software Defined Networking

Why SDN is Important?

- **Better Network Connectivity:** SDN provides very better network connectivity for sales, services, and internal communications. SDN also helps in faster data sharing.
- **Better Deployment of Applications:** Deployment of new applications, services, and many business models can be speed up using Software Defined Networking.
- **Better Security:** Software-defined network provides better visibility throughout the network. Operators can create separate zones for devices that require different levels of security. SDN networks give more freedom to operators.
- **Better Control with High Speed:** Software-defined networking provides better speed than other networking types by applying an open standard software-based controller.

In short, it can be said that- SDN acts as a "Bigger Umbrella or a HUB" where the rest of other networking technologies come and sit under that umbrella and get merged with another platform to bring out the best of the best outcome by decreasing the traffic rate and by increasing the efficiency of data flow.

Where is SDN Used?

- Enterprises use SDN, the most widely used method for application deployment, to deploy applications faster while lowering overall deployment and operating costs. SDN allows IT administrators to manage and provision network services from a single location.
- Cloud networking software-defined uses white-box systems. Cloud providers often use generic hardware so that the Cloud data center can be changed and the cost of CAPEX and OPEX saved.

Components of Software Defining Networking (SDN)

The three main components that make the SDN are:

1. **SDN Applications:** SDN Applications relay requests or networks through SDN Controller using API.
2. **SDN controller:** SDN Controller collects network information from hardware and sends this information to applications.
3. **SDN networking devices:** SDN Network devices help in forwarding and data processing tasks.

SDN Architecture

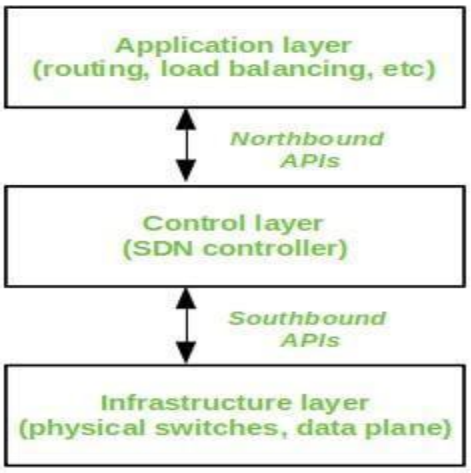
In a traditional network, each switch has its own data plane as well as the control plane. The control plane of various switches exchange topology information and hence construct a forwarding table that decides where an incoming data packet has to be forwarded via the data plane. Software-defined networking (SDN) is an approach via which we take the control plane away from the switch and assign it to a centralized unit called the SDN controller. Hence, a network administrator can shape traffic via a centralized console without having to touch the individual switches. The data plane still resides in the switch and when a packet enters a switch, its forwarding activity is decided based on the entries of flow tables, which are pre-assigned by the controller. A flow table consists of match fields (like input port

number and packet header) and instructions. The packet is first matched against the match fields of the flow table entries. Then the instructions of the corresponding flow entry are executed. The instructions can be forwarding the packet via one or multiple ports, dropping the packet, or adding headers to the packet. If a packet doesn't find a corresponding match in the flow table, the switch queries the controller which sends a new flow entry to the switch. The switch forwards or drops the packet based on this flow entry.

A typical SDN architecture consists of three layers.

- **Application layer:** It contains the typical network applications like intrusion detection, firewall, and load balancing
- **Control layer:** It consists of the SDN controller which acts as the brain of the network. It also allows hardware abstraction to the applications written on top of it.
- **Infrastructure layer:** This consists of physical switches which form the data plane and carries out the actual movement of data packets.

The layers communicate via a set of interfaces called the north-bound APIs(between the application and control layer) and southbound APIs(between the control and infrastructure layer).



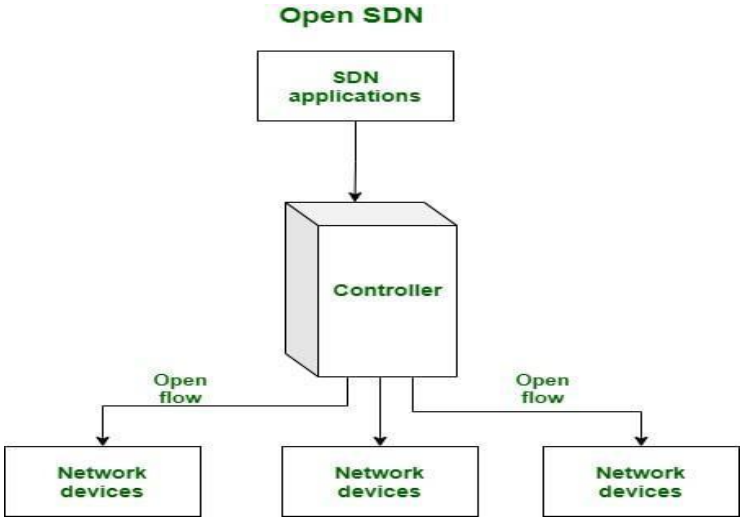
SDN Architecture

Different Models of SDN

There are several models, which are used in SDN:

1. Open SDN
2. SDN via APIs
3. SDN via Hypervisor-based Overlay Network
4. Hybrid SDN

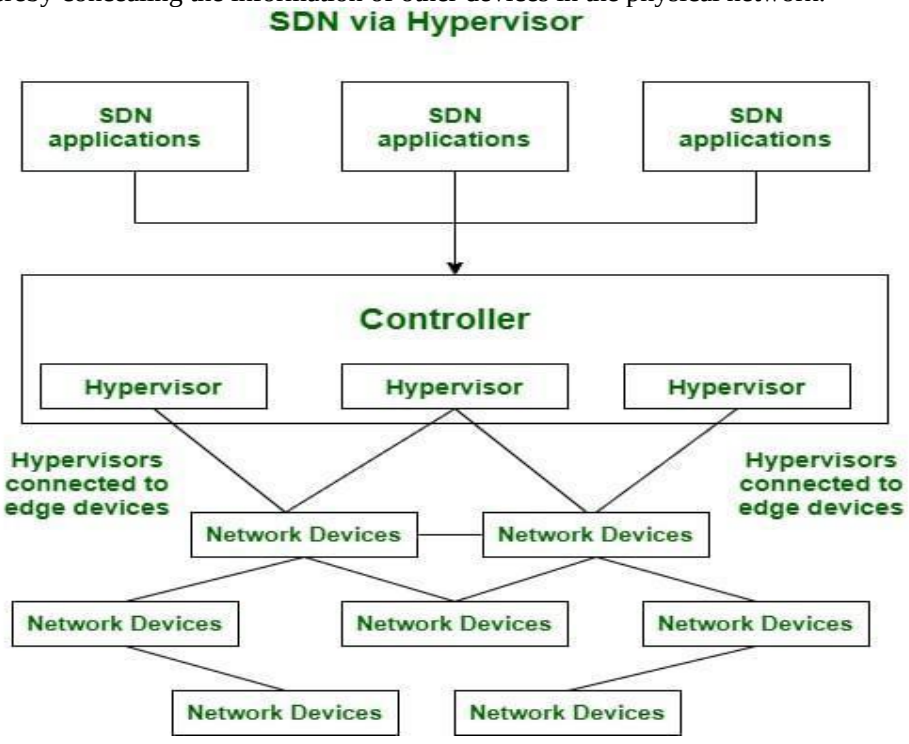
1. Open SDN: Open SDN is implemented using the OpenFlow switch. It is a straightforward implementation of SDN. In Open SDN, the controller communicates with the switches using south-bound API with the help of OpenFlow protocol.



Open SDN

2. SDN via APIs: In SDN via API, the functions in remote devices like switches are invoked using conventional methods like SNMP or CLI or through newer methods like Rest API. Here, the devices are provided with control points enabling the controller to manipulate the remote devices using APIs.

3. SDN via Hypervisor-based Overlay Network: In SDN via the hypervisor, the configuration of physical devices is unchanged. Instead, Hypervisor based overlay networks are created over the physical network. Only the devices at the edge of the physical network are connected to the virtualized networks, thereby concealing the information of other devices in the physical network.

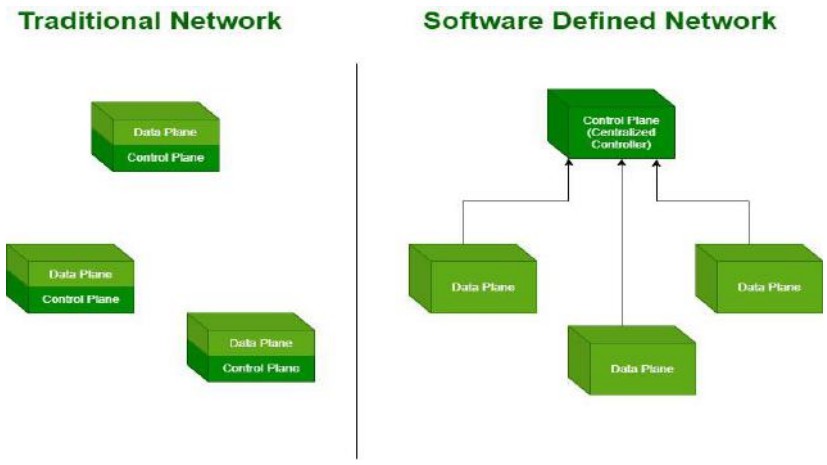


SDN via Hypervisor-based Overlay Network

4. Hybrid SDN: Hybrid Networking is a combination of Traditional Networking with software-defined networking in one network to support different types of functions on a network.

Difference between SDN and Traditional Networking

Software Defined Networking	Traditional Networking
Software Defined Network is a virtual networking approach.	A traditional network is the old conventional networking approach.
Software Defined Network is centralized control.	Traditional Network is distributed control.
This network is programmable.	This network is nonprogrammable.
Software Defined Network is the open interface.	A traditional network is a closed interface.
In Software Defined Network data plane and control, the plane is decoupled by software.	In a traditional network data plane and control plane are mounted on the same plane.



Difference between SDN and Traditional Networking

Advantages of SDN

- The network is programmable and hence can easily be modified via the controller rather than individual switches.
- Switch hardware becomes cheaper since each switch only needs a data plane.

- Hardware is abstracted, hence applications can be written on top of the controller independent of the switch vendor.
- Provides better security since the controller can monitor traffic and deploy security policies. For example, if the controller detects suspicious activity in network traffic, it can reroute or drop the packets.

Disadvantages of SDN

- The central dependency of the network means a single point of failure, i.e. if the controller gets corrupted, the entire network will be affected.
- The use of SDN on large scale is not properly defined and explored.
- **Network Functions Virtualization**

The term "Network Functions Virtualization" (NFV) refers to the use of virtual machines in place of physical network appliances. There is a requirement for a hypervisor to operate networking software and procedures like load balancing and routing by virtual computers. A network functions virtualization standard was first proposed at the OpenFlow World Congress in 2012 by the European Telecommunications Standards Institute (ETSI), a group of service providers that includes AT&T, China Mobile, BT Group, Deutsche Telekom, and many more.

Need of NFV:

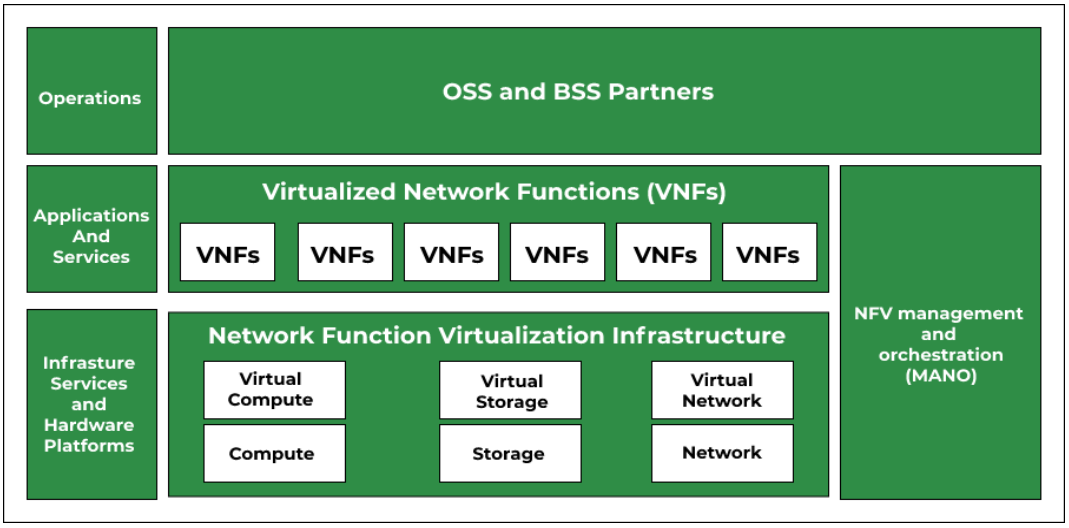
With the help of NFV, it becomes possible to separate communication services from specialized hardware like routers and firewalls. This eliminates the need for buying new hardware and network operations can offer new services on demand. With this, it is possible to deploy network components in a matter of hours as opposed to months as with conventional networking. Furthermore, the virtualized services can run on less expensive generic servers.

Advantages:

- Lower expenses as it follows Pay as you go which implies companies only pay for what they require.
- Less equipment as it works on virtual machines rather than actual machines which leads to fewer appliances, which lowers operating expenses as well.
- Scalability of network architecture is quite quick and simple using virtual functions in NFV. As a result, it does not call for the purchase of more hardware.

Working:

Usage of software by virtual machines enables to carry out the same networking tasks as conventional hardware. The software handles the task of load balancing, routing, and firewall security. Network engineers can automate the provisioning of the virtual network and program all of its various components using a hypervisor or software-defined networking controller.



Benefits of NFV:

- Many service providers believe that advantages outweigh the issues of NFV.

- Traditional hardware-based networks are time-consuming as these require network administrators to buy specialized hardware units, manually configure them, then join them to form a network. For this skilled or well-equipped worker is required.
- It costs less as it works under the management of a hypervisor, which is significantly less expensive than buying specialized hardware that serves the same purpose.
- Easy to configure and administer the network because of a virtualized network. As a result, network capabilities may be updated or added instantly.

Risks of NFV:

Security hazards do exist, though, and network functions virtualization security issues have shown to be a barrier to widespread adoption among telecom companies. The following are some dangers associated with implementing network function virtualization that service providers should take into account:

- **Physical security measures do not work:** Comparing virtualized network components to locked-down physical equipment in a data center enhances their susceptibility to new types of assaults.
- Malware is difficult to isolate and contain: Malware travels more easily among virtual components running on the same virtual computer than between hardware components that can be isolated or physically separated.
- **Network activity is less visible:** Because traditional traffic monitoring tools struggle to detect potentially malicious anomalies in network traffic going east-west between virtual machines, NFV necessitates more fine-grained security solutions.

NFV Architecture:

An individual proprietary hardware component, such as a router, switch, gateway, firewall, load balancer, or intrusion detection system, performs a specific networking function in a typical network architecture. A virtualized network substitutes software programs that operate on virtual machines for these pieces of hardware to carry out networking operations.

Three components make up an NFV architecture:

- **Centralized virtual network infrastructure:** The foundation of an NFV infrastructure can be either a platform for managing containers or a hypervisor that abstracts the resources for computation, storage, and networking.
- **Applications:** Software delivers many forms of network functionality by substituting for the hardware elements of a conventional network design (virtualized network functions).
- **Framework:** To manage the infrastructure and provide network functionality, a framework is required (commonly abbreviated as MANO, meaning Management, Automation, and Network Orchestration).oTSystemManagementwithNETCONF-YANGIoT Systems Management with NETCONF-YANG YANG is a data modelling language used to model configuration and state data manipulated by the NETCONF protocol. The generic approach of IoT device management with NETCONF-YANG. Roles of various components are:

1) Management System
2) Management API
3) Transaction Manager
4) Rollback Manager
5) Data Model Manager
6) Configuration Validator
7) Configuration Database
8) Configuration API
9) Data Provider API

IoTSystemManagementwithNETCONF-YANG

NeedforIoTSystems 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:

- AutomatingConfiguration
- MonitoringOperational &StatisticalData
- ImprovedReliability
- SystemWideConfigurations
- MultipleSystemConfigurations
- Retrieving&ReusingConfigurations

- **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 other devices of the same type. For example, for an IoT system which has multiple devices and requires same configuration for all devices, it is important to ensure that when a new device is added, the same configuration is applied. For such cases, the management system can retrieve the current configuration from a device and apply the same to the new devices.

4.2 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.

the components of the entities involved in managing a device with SNMP, including the Network Management Station (NMS), Managed Device, Management Information Base (MIB) and the SNMP Agent that runs on the device. NMS executes SNMP commands to monitor and configure the Managed Device. The Managed Device contains the MIB which has all the information of the device attributes to be managed. MIBs use the Structure of Management Information (SMI) notation for defining the structure of the management data. The structure of management data is defined in the form of variables which are identified by object identifiers (OIDs), which have a hierarchical structure. Management applications can either get or set the values of these variables. SNMP is an application layer protocol that uses User Datagram Protocol (UDP) as the transport protocol.

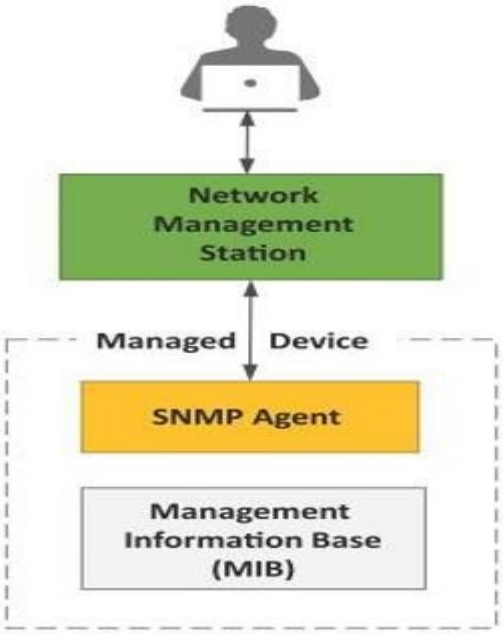


Fig:Managingadevicewith SN

4.2.1 Limitations of SNMP

While Simple Network Management Protocol (SNMP) has been the most popular protocol for network management, it has several limitations which may make it unsuitable for configuration management.

- SNMP was designed to provide a simple management interface between the management applications and the managed devices. 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. For a sequence of SNMP interactions, the application

needs to maintain state and also to be smart enough to roll back the device into a consistent state in case of errors or failures in configuration.

- 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. With the absence of writable objects, SNMP can be used only for device monitoring and status polling.
- It is difficult to differentiate between configuration and state data in MIBs.
- Retrieving the current configuration from a device can be difficult with SNMP. SNMP does not support easy retrieval and playback of configurations.
- Earlier versions of SNMP did not have strong security features making the management information vulnerable to network intruders. Though security features were added in the later versions of SNMP, it increased the complexity a lot.

4.3 Network Operator Requirements

To address the limitations of the existing network management protocols and plan the future work on network management, the Internet Architecture Board (IAB), which oversees the Internet Engineering Task Force (IETF) held a workshop on network management in 2002 that brought together network operators and protocol developers. Based on the inputs from operators, a list of operator requirements was prepared [122]. The following points provide a brief overview of the operator requirements.

- **Ease of use**: From the operators point of view, ease of use is the key requirement for any network management technology.
- **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.

4.4 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

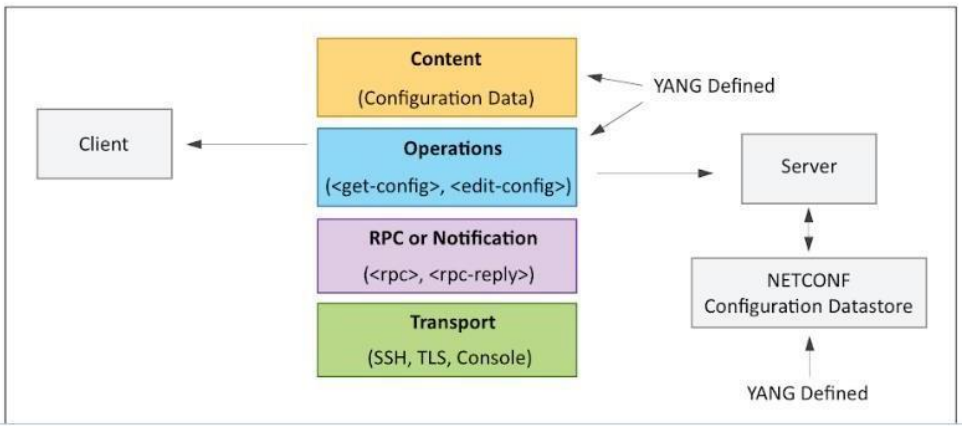


Fig:NETCONFprotocol layers

For network management architecture based on NETCONF, the terms client and management system and the terms server and device are often used interchangeably. NETCONF works on SSH transport protocol. In addition to Secure Shell Transport Layer Protocol (SSH), NETCONF implementations can support other transport mappings such as Blocks Extensible Exchange Protocol (BEEP). 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. For example, the NETCONF operation <get-config> retrieves the configuration data only, while the operation <get> retrieves the configuration and state data.

The configuration data resides within a NETCONF configuration datastore on the server. The NETCONF server resides on the network device. The management application plays the role of a NETCONF client. For managing a network device the client establishes a NETCONF session with the server. When a session is established the client and server exchange 'hello' messages which contain information on their capabilities. Client can then send multiple requests to the server for retrieving or editing the configuration data. NETCONF allows the management client to discover the capabilities of the server (on the device). NETCONF gives access to the native capabilities of the device.

NETCONF defines one or more configuration datastores. A configuration store contains all the configuration information to bring the device from its initial state to the operational state. By default a <running> configuration store is present. Additional configuration datastores such as <startup> and <candidate> can be defined in the capabilities.

NETCONF is a connection oriented protocol and NETCONF connection persists between protocol operations. For authentication, data integrity, and confidentiality, NETCONF depends on the transport protocol, e.g., SSH or TLS. NETCONF overcomes the limitations of SNMP and is suitable not only for monitoring state information, but also for configuration management.

4.5 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. YANG defines four types of nodes for data modeling

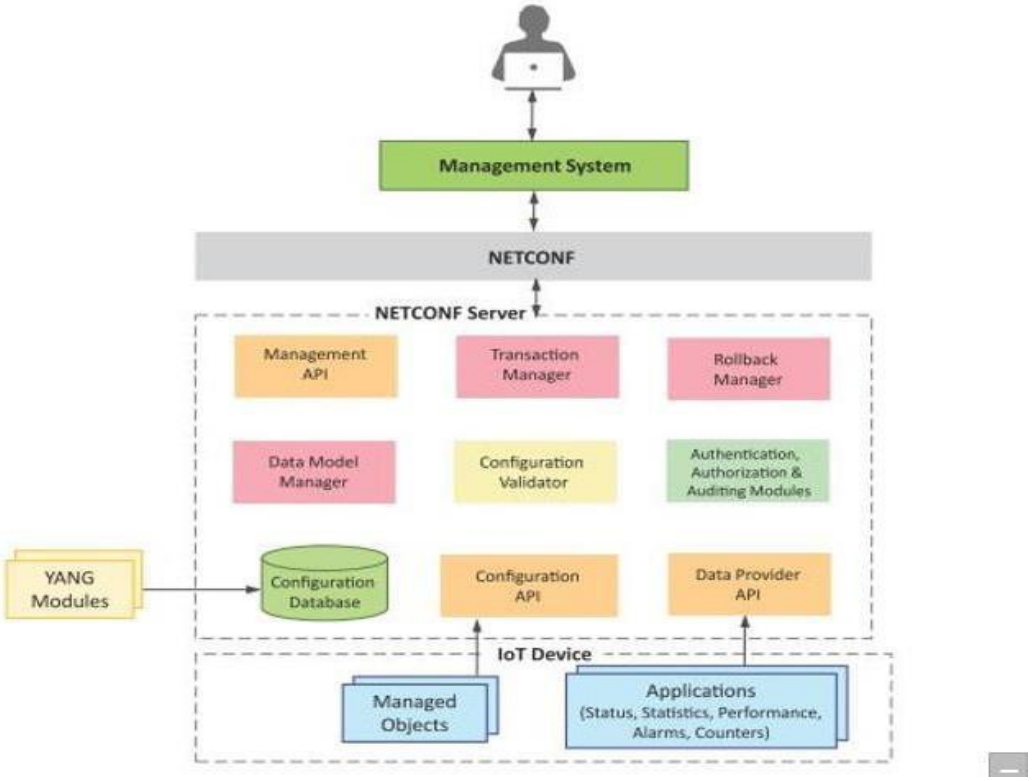
Let us now look at an example of a YANG module. a YANG module for a "network-enabled toaster". This YANG module is a YANG version of the toaster Management Information Base (MIB). We use the Toaster MIB since it has been widely used as an example in introductory tutorials on SNMP to explain how SNMP can be used for managing a network-connected toaster. A YANG module has several sections starting from header information, followed by imports and includes, type definitions, configuration and operational data declarations, and RPC and notification declarations. The toaster YANG module begins with the header information followed by identity declarations which define various bread types. The leaf nodes ('toasterManufacturer', 'toasterModelNumber' and 'toasterStatus') 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').

4.6 IoT Systems Management with NETCONF-YANG

In this section you will learn how to manage IoT systems with NECONF and YANG. Figure 4.5 shows the generic approach of IoT device management with NETCONF-YANG.

Let is look at the roles of the various components:

- **Management System**: The operator uses a Management System to send NETCONF messages to configure the IoT device and receives state information and notifications from the device as NETCONF messages.
- **Management API**: Management API allows management applications to start NETCONF sessions, read and write configuration data, read state data, retrieve configurations, and invoke RPCs, programmatically, in the same way as an operator can.
- **Transaction Manager**: Transaction Manager executes all the NETCONF transactions and ensures that the ACID (Atomicity, Consistency, Isolation, Durability) properties hold true for the transactions. Atomicity property ensures that a transaction is executed



either completely or not at all. Consistency property ensures that a transaction brings the device configuration from one valid state to another. Isolation property ensures that concurrent execution of transactions results in the same device configuration as if transactions were executed serially in order. Durability property ensures that a transaction once committed will persist.

- **Rollback Manager** : Rollback manager is responsible for generating all the transactions necessary to rollback a current configuration to its original state.
- **Data Model Manager**: The Data Model manager keeps track of all the YANG data models and the corresponding managed objects. The Data Model manager also keeps track of the applications which provide data for each part of a data model.
- **Configuration Validator**: Configuration validator checks if the resulting configuration after applying a transaction would be a valid configuration.
- **Configuration Database**: This database contains both the configuration and operational data.
- **Configuration API**: Using the configuration API the applications on the IoT device can read configuration data from the configuration datastore and write operational data to the operational datastore.
- **Data Provider API**: Applications on the IoT device can register for callbacks for various events using the Data Provider API. Through the Data Provider API, the applications can report statistics and operational data.

4.6.1 NETOPEER

While the previous section described a generic approach of IoT device management with NETCONF-YANG, this section describes a specific implementation based on the Netopeer tools Netopeer is set of open source NETCONF tools built on the Libnetconf library how to manage an IoT device using the Netopeer tools. The Netopeer tools include:

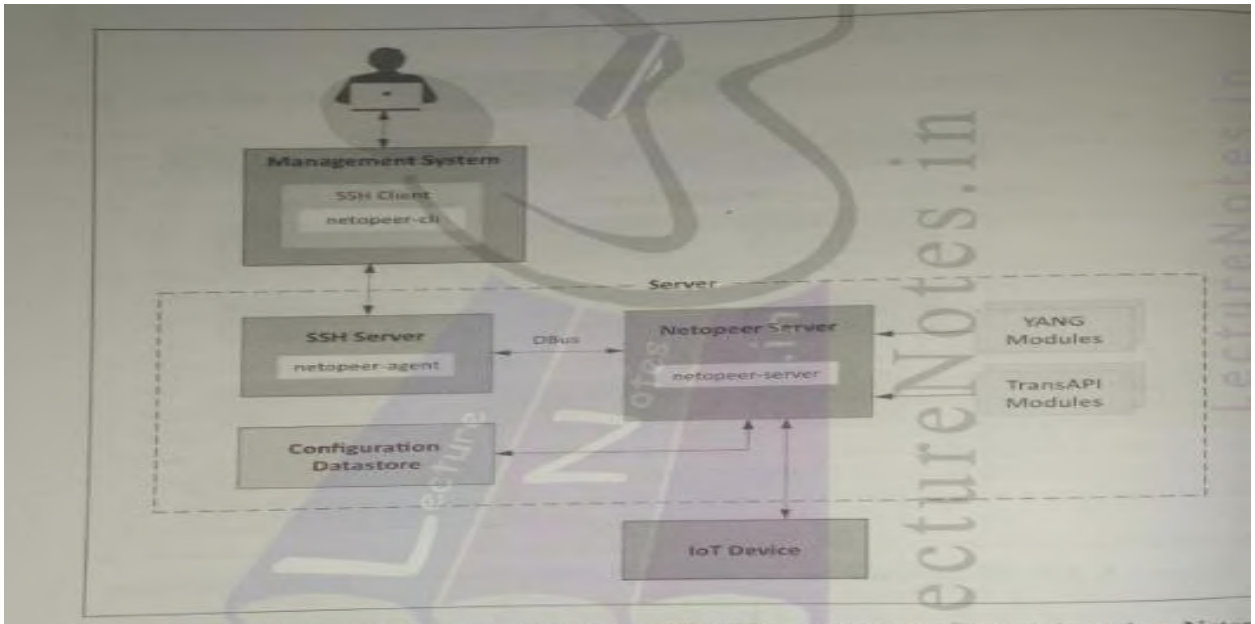


Figure 4.6: IoT device management with NETCONF - a specific approach based on Netop tools

- **Netopeer-server**: Netopeer-server is a NETCONF protocol server that runs on the managed device. Netopeer-server provides an environment for configuring the device using NETCONF RPC operations and also retrieving the state data from the device.
- **Netopeer-agent**: Netopeer-agent is the NETCONF protocol agent running as a SSH/TLS subsystem. Netopeer-agent accepts incoming NETCONF connection and passes the NETCONF RPC operations received from the NETCONF client to the Netopeer-server.
  - **Netopeer-cli**: Netopeer-cli is a NETCONF client that provides a command line interface for interacting with the Netopeer-server. The operator can use the Netopeer-cli from the management system to send NETCONF RPC operations for configuring the device and retrieving the state information.
  - **Netopeer-manager**: Netopeer-manager allows managing the YANG and Libnetconf Transaction API (TransAPI) modules on the Netopeer-server. With Netopeer-manager modules can be loaded or removed from the server.
  - **Netopeer-configurator**: Netopeer-configurator is a tool that can be used to configure the Netopeer-server.

Steps for IoT device Management with NETCONF-YANG

1. Create a YANG model of the system that defines the configuration and state data of the system.
2. Compile the YANG model with the 'lnctool' which comes with Libnetconf.

Libnetconf provides a framework called Transaction API (TransAPI) that provides a mechanism of reflecting the changes in the configuration file in the actual device. The 'lnctool' generates a TransAPI module (callbacks C file). Whenever a change is made in the configuration file using the NETCONF operations, the corresponding callback function is called. The callback functions contain the code for making the changes on the device.

3. Fill in the IoT device management code in the TransAPI module (callbacks C file). This file includes configuration callbacks, RPC callbacks and state data callbacks.
4. Build the callbacks C file to generate the library file (.so).
5. Load the YANG module (containing the data definitions) and the TransAPI module (.so binary) into the Netopeer server using the Netopeer manager tool.
6. The operator can now connect from the management system to the Netopeer server using the Netopeer CLI.
7. Operator can issue NETCONF commands from the Netopeer CLI. Commands can be issued to change the configuration data, get operational data or execute an RPC on the IoT device.

UNIT-III

Developing internet of Things

3.1.Introduction to Developing internet of Things

3.2.Iot Design Methodology

3.3. Case Study on IoT System for weather Monitoring

3.4.case studies illustrating IoT Design

3.4.1.introduction

3.4.2.Home Automation

3.4.3.Cities

3.4.4.Environment

3.4.5. Agriculture

3.4.5. productivity Applications

IoT Design Methodology

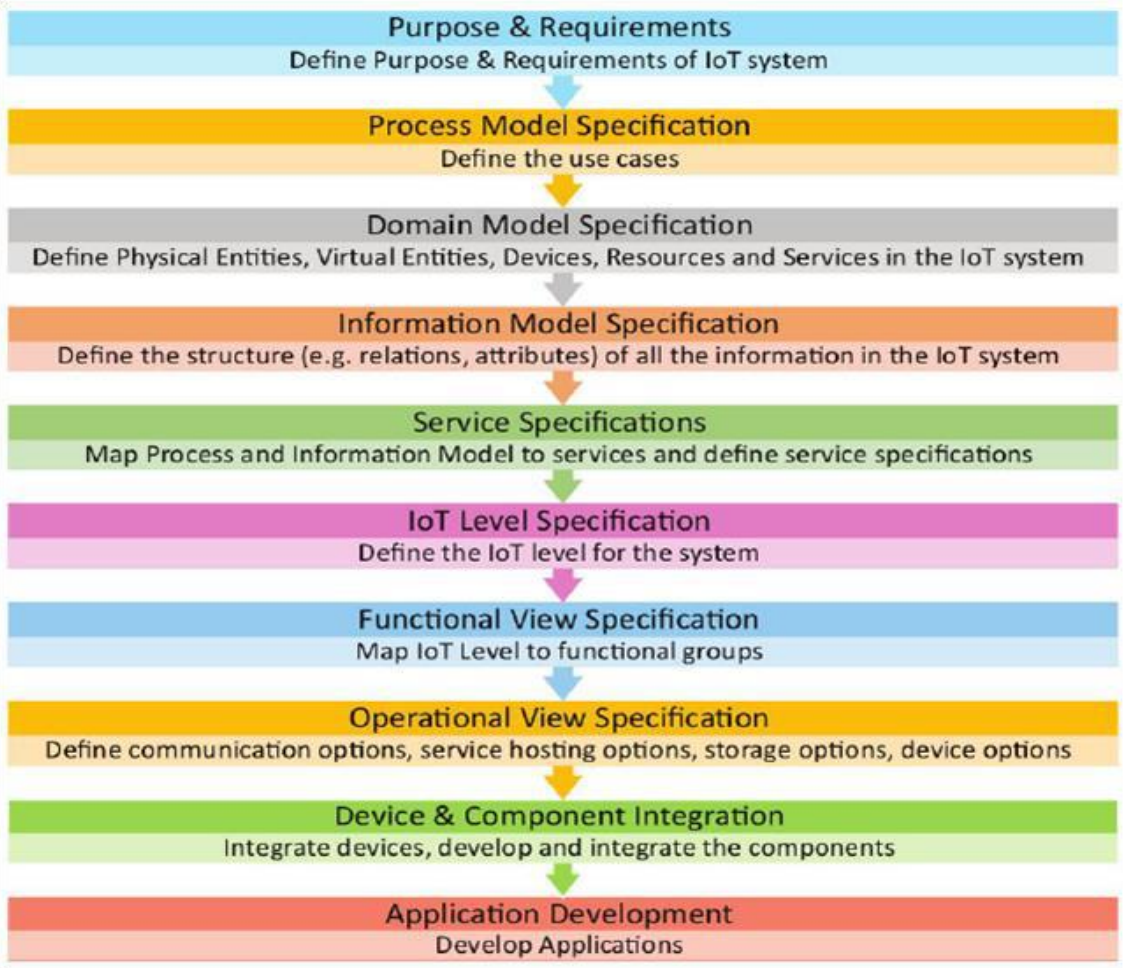
Introduction to IoT Design Methodology

Designing IoT systems can be a complex and challenging task as these systems involve interactions between various components. A wide range of choices are available for each component. IoT designers often tend to design the system keeping specific products in mind.

We will look at a generic design methodology which is independent of specific product, service or programming language. IoT systems designed with this methodology will have reduced design time, testing time, maintenance time, complexity and better interoperability.

Watch this video to learn about IoT design methodology:

The steps involved in the designing of an IoT system or application can be summarized as shown in the below figure:



Let's discuss all the ten steps in the IoT design methodology with the help of a case study: Home Automation System.

1. Purpose and Requirements Specification

First step is to define the purpose and requirements of the system. In this step, the system purpose, behavior and requirements are captured. Requirements can be:

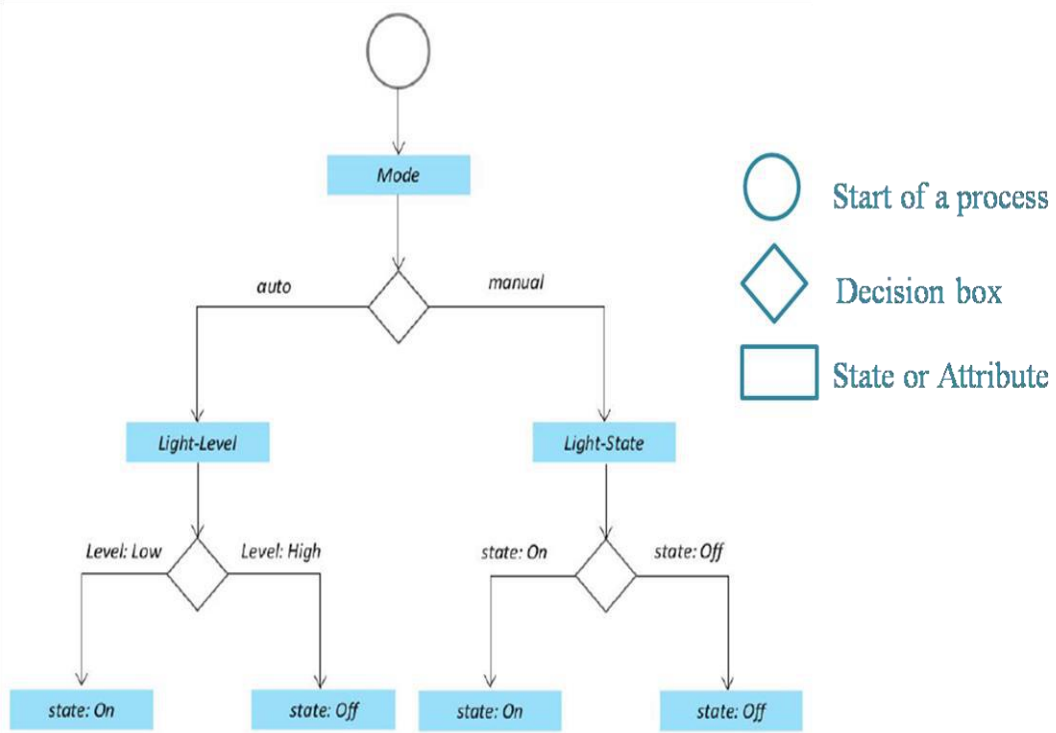
- Data collection requirements
- Data analysis requirements
- System management requirements
- Security requirements
- User interface requirements

For home automation system the purpose and requirements specification is as follows:

Purpose	A home automation system that allows controlling the lights remotely using a web application
Behavior	Home automation system should support two modes: auto and manual **Auto:** System measures the light level in the room and switches on the light when it is dark **Manual:** Allows remotely switching lights on and off
System Management	System should provide remote monitoring and control functions
Data Analysis	System should perform local analysis of the data
Application Deployment	Application should be deployed locally, but should be accessible remotely
Security	Should provide basic security like user authentication

2. Process Specification

The use cases of the IoT system are formally described based on or derived from the purpose and requirements specifications. The process specification for home automation system is as shown below.

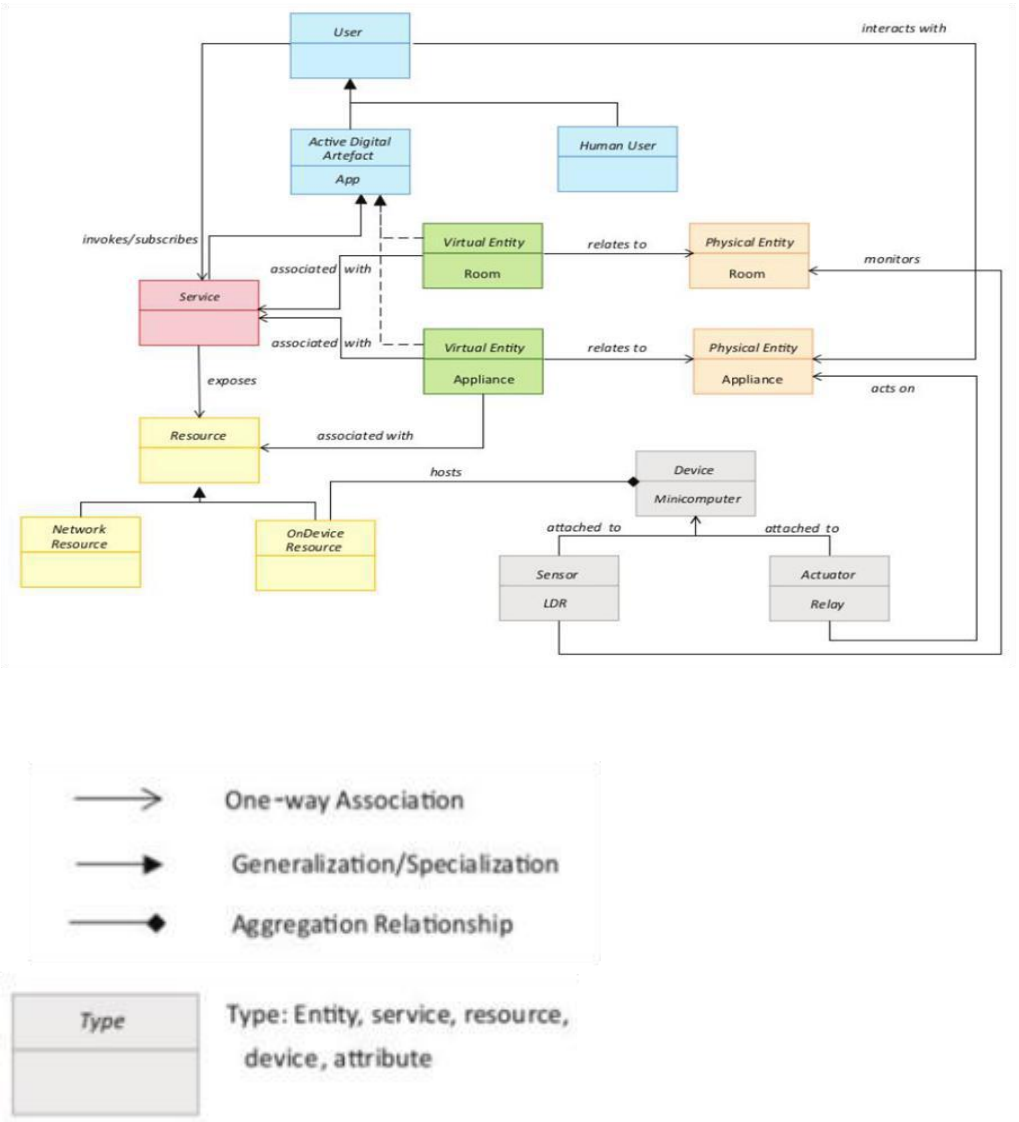


3. Domain Model Specification

The domain model describes the main concepts, entities and objects in the domain of the IoT system to be designed. Domain model defines the attributes of the objects and relationships between objects. The domain model is independent of any specific technology or platform.domain model, system designers can get an understanding of the IoT domain for which the system is to be designed. The entities, objects and concepts defined in the domain model of home automation system include the following:

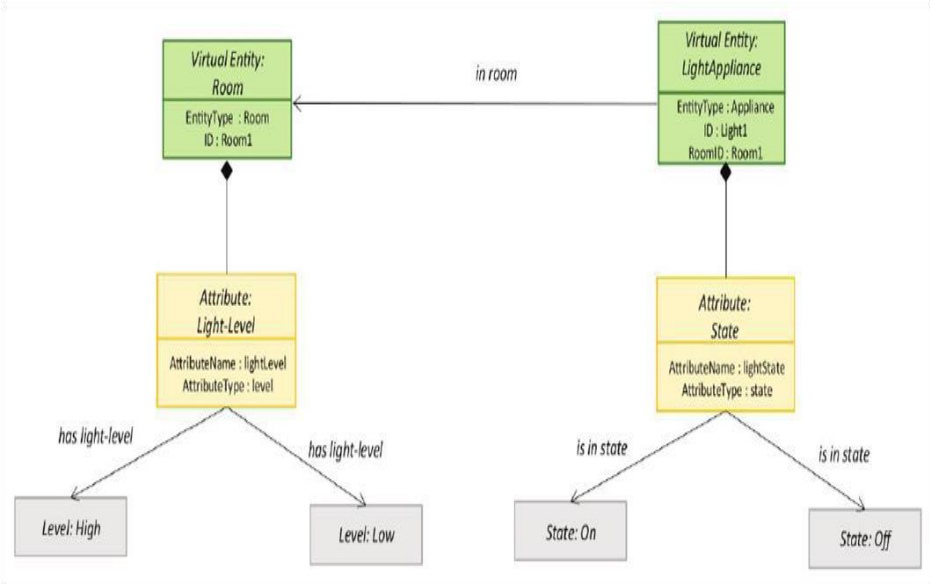
Physical Entity	• The physical identifiable objects in the environment • IoT system provides information about the physical entity (using sensors) or performs actuation upon the physical entity
Virtual Entity	• Virtual entity is a representation of the physical entity in the digital world • For every physical entity there is a virtual entity
Device	• Devices provide a medium for interaction between physical and virtual entities • Devices are used to gather information from or perform actuation on physical entities
Resource	• Resources are software components which can be either on-device or network-resources • On-device resources are hosted on the device and provide sensing or actuation (eg: operating system) • Network-resources include software components that are available on the network (eg: database)
Service	• Services provide an interface for interacting with the physical entity • Services access resources to perform operations on physical entities

The domain model specification diagram for home automation system is as shown in the below figure.



4. Information Model Specification

Information model defines the structure of all the information in the IoT system. Does not describe how the information is stored and represented. To define the information model, we first list the virtual entities. Later more details like attributes and relationships are added. The information model specification for home automation system is as shown below:



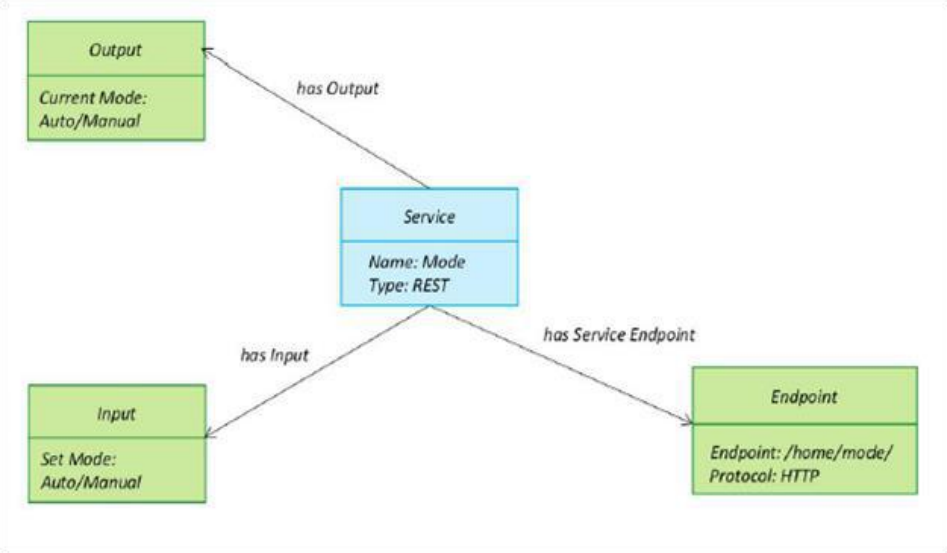
Watch the below video to learn about the rest of the steps in IoT design methodology:

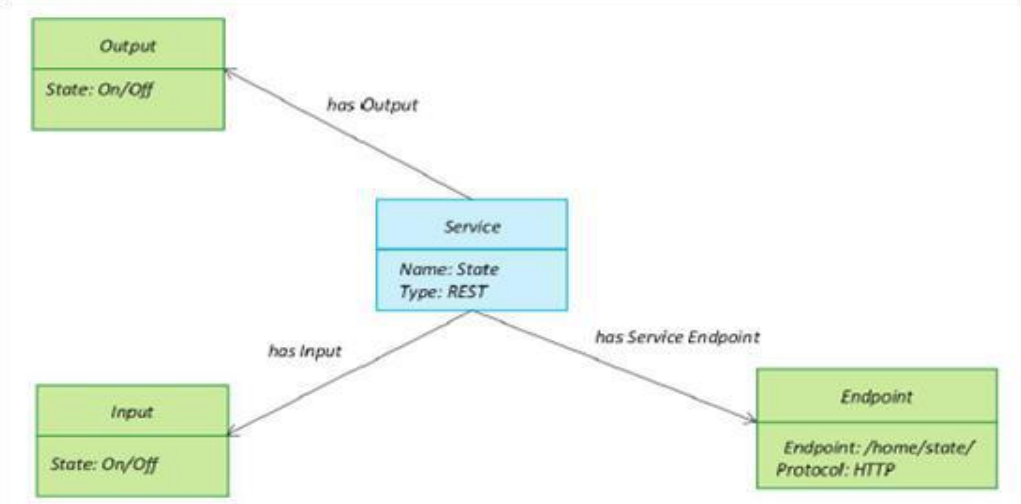
5. Service Specifications

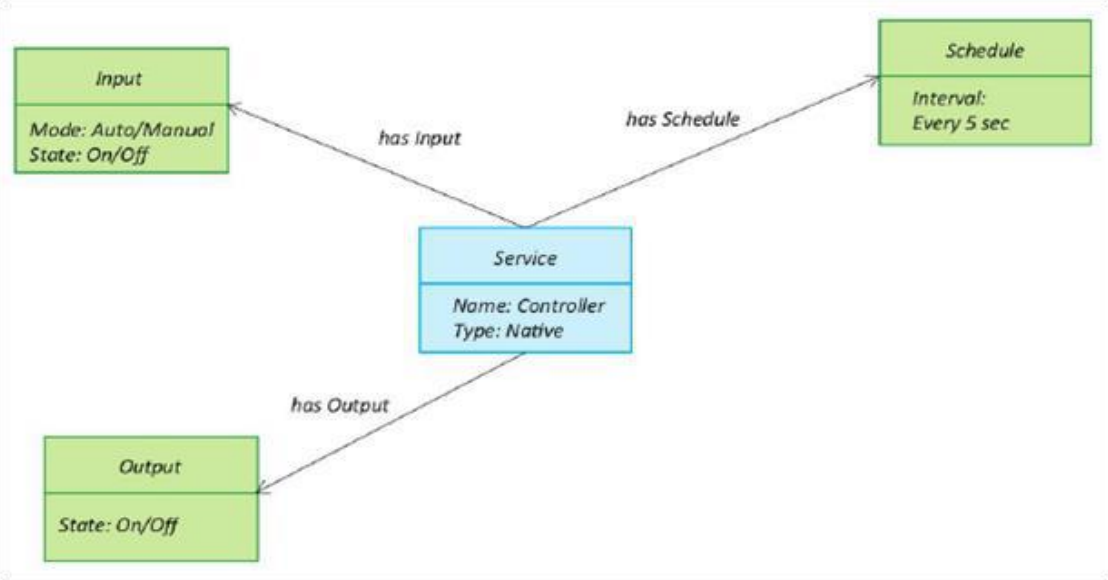
The service specification defines the following:

- Services in the system
- Service types
- Service inputs/output
- Service endpoints
- Service schedules
- Service preconditions
- Service effects

For each state and attribute in the process specification and information model, we define a service. Services either change the state of attributes or retrieve their current values. The service specification for each state in home automation systems are as shown below:

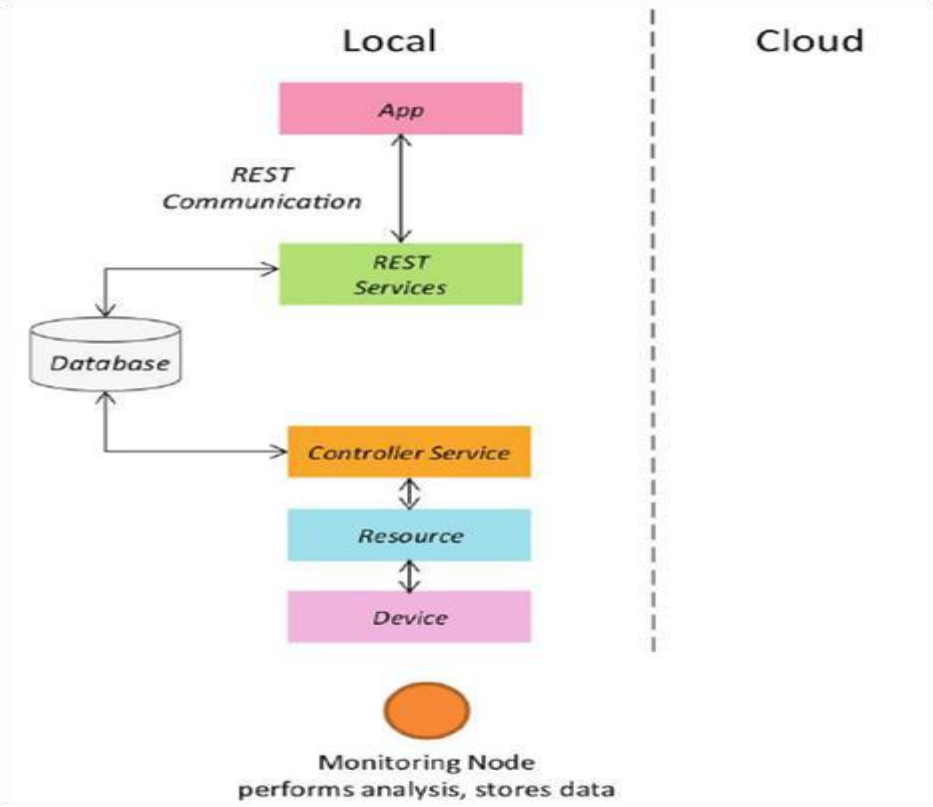

Output
Current Mode:
Auto/Manual
has Output
Service
Name: Mode
Type: REST
has Input
has Service Endpoint
Input
Set Mode:
Auto/Manual
Endpoint
Endpoint: /home/mode/
Protocol: HTTP


Output
State: On/Off
has Output
Service
Name: State
Type: REST
has Input
has Service Endpoint
Input
State: On/Off
Endpoint
Endpoint: /home/state/
Protocol: HTTP



6. IoT Level Specification

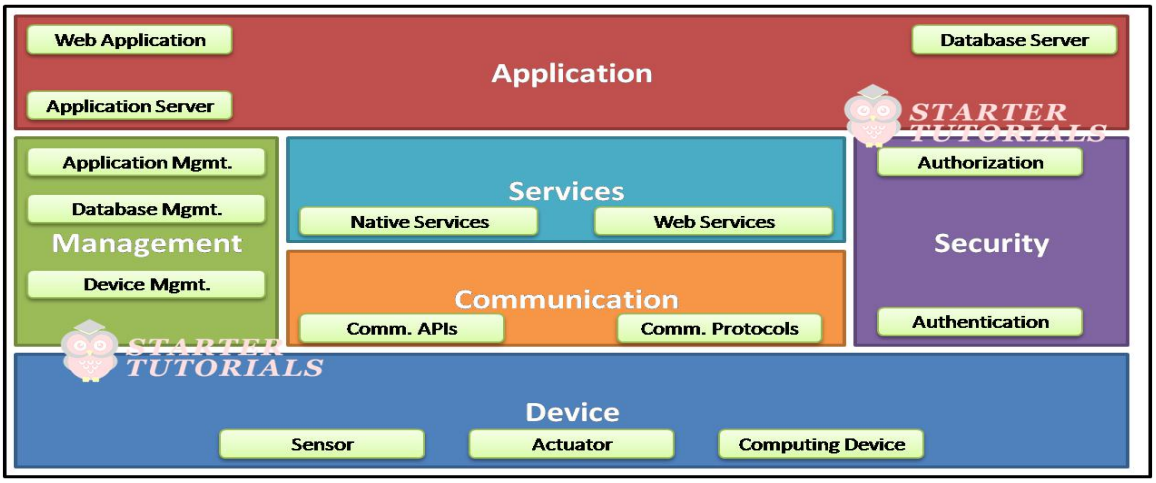
Based on the requirements we will choose the IoT application deployment level. The deployment level for home automation system is shown in the below figure.



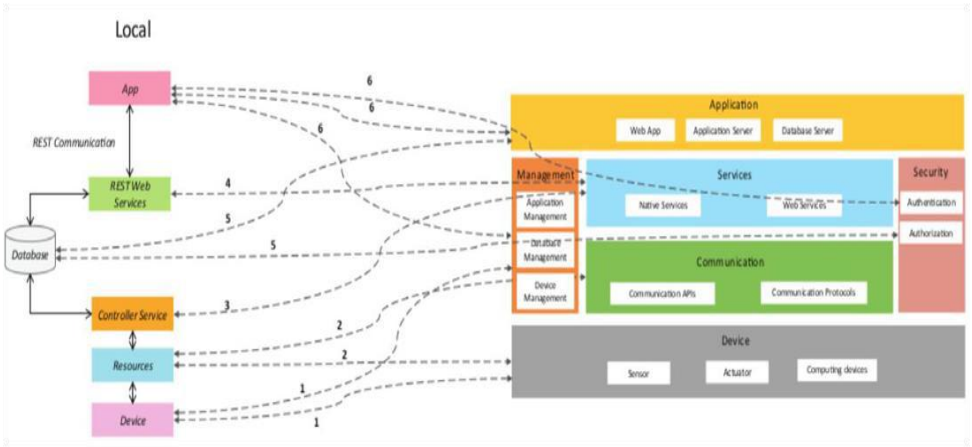
7. Functional View Specification

The functional view defines the functions of the IoT systems grouped into various functional groups. Each functional group provides functionalities for interacting with concepts in the domain model and information related to the concepts.

The functional groups in a functional view include: Device, Communication, Services, Management, Security, and Application. The functional view specification for home automation system is shown in the below figure:



The mapping between the IoT level and the functional groups is as shown in the below figure.

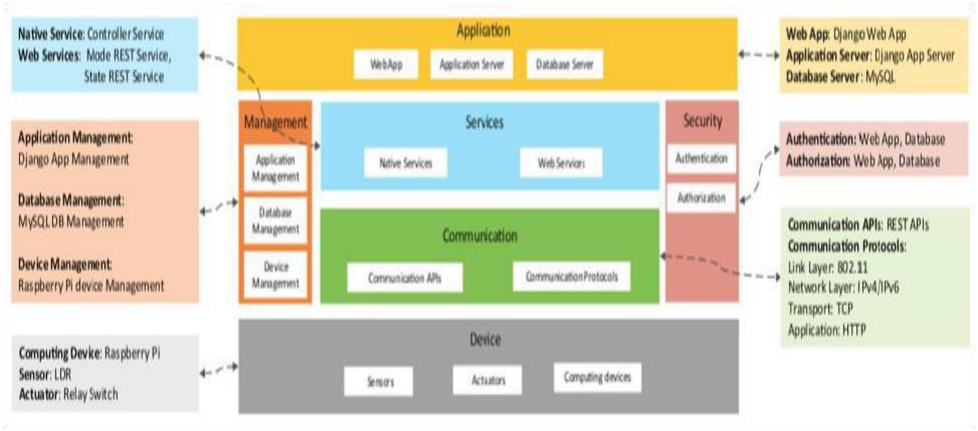


8. Operational View Specification

In this step, various options related to the IoT system deployment and operation are defined, such as:

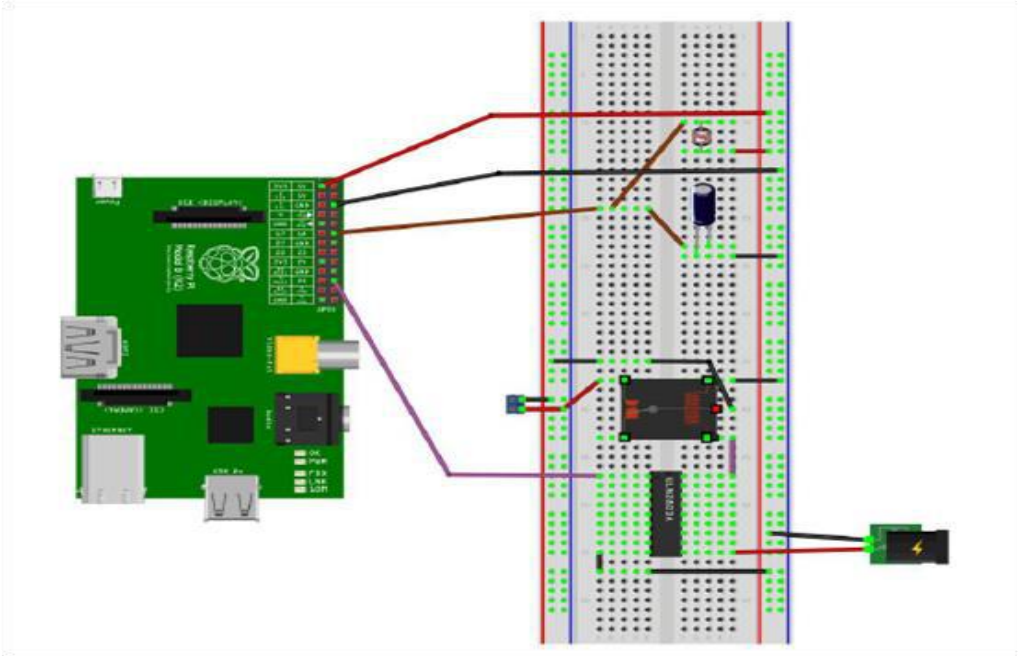
- Service hosting options
- Storage options
- Device options
- Application hosting options

The options chosen for home automation system are as shown in the below figure.



9. Device and Component Integration

In this step the devices like sensors, computing devices and other components are integrated together. The interconnection of different components in our home automation system are as shown in the figure given below.



10. Application Development

Using all the information from previous steps, we will develop the application (code) for the IoT system. The application interface for home automation system is shown below.

Case study on IoT System for Weather Monitoring

Because of the rapidly changing climate, the weather forecast is uncertain and inaccurate these days. As a result, the Weather Reporting System is primarily utilized to monitor the constantly changing climatic and weather conditions over-regulated areas like homes, industry, agriculture, and so on.

When objects like an environment furnished with sensor devices, microcontrollers, and different software applications become a self-monitoring and self-protecting environment, it is called a smart environment.

Similarly, here, the system uses sensors to monitor and adjust environmental parameters such as temperature, CO levels, and relative humidity. Then, it sends the data to a web page to plot the sensor data, shown as graphical statistics. The data updated from this system can be accessed on the internet from anywhere in the world. The embedded system enables the user to access the various criteria and store the data in the cloud.

Hence, the Internet of Things (IoT) is the core root of linking all the sensors to the internet and monitoring the weather in real-time.

Weather Monitoring System Using IoT Block Diagram

Event Detection-based and Spatial Process Estimation are the two kinds to which applications are classified.

This ecosystem consists of a microcontroller (e.g., Arduino UNO or ESP8266) which acts as the main processing unit for the entire system and where all the sensors (e.g., Humidity Sensors and Temperature Sensors) and devices are connected. When a proper connection is established with the server device, the data collected from various sensor devices implanted in specific areas of interest is immediately relayed to the webserver. Using any Wi-Fi module such as Node-MCU, this processed sensor data is then uploaded and stored on a website to serve as a database.

We will be able to monitor and control the system using the webserver page. It provides information on the variations in humidity, temperature, and CO levels in the exact region where the embedded monitoring device is installed. The data collected will be saved on the cloud. The cloud data can be used for parameter analysis and continuous monitoring. Temperature, humidity, and carbon monoxide levels in the air are recorded at regular intervals. All this information will be stored in the cloud, allowing us to monitor temperature, humidity, and CO levels at a given place at any time.

Areas Benefiting from Weather Forecasting System Using IoT

An accurate weather report is forecasted directly or indirectly to influence other sectors of the economy to raise the need for a system that facilitates higher accuracy of real-time monitoring and future weather prediction.

But what exactly are the different sectors that benefit from the IoT weather station? Well, let’s have a look at them!

Agriculture

With the current global trends for agriculture and the depletion of natural resources, the demand has increased. Preparation of soil, sowing, irrigation, and harvesting of crops are directly dependent on weather conditions. Thankfully, the IoT technology is fueling the transformation, helping farmers vulnerable to weather hazards to use the IoT intelligence for improving their crop fertility and cost. The integration of real-time data into supply chain plans is assisting in the transportation of perishable commodities across the country, resulting in increased productivity and efficiency.

Manufacturing

The environment where a manufacturing plant is located can have an unnoticed but significant impact on the final product. Small changes in temperature and humidity, for example, might impact the way industrial glues adhere, affecting product quality directly.

A machine, for example, is built to run at a temperature of 100°F. If it runs in much hotter or colder areas, it will have a completely different life cycle.

Businesses may have a more accurate perspective of the condition of their important business assets with an IoT weather reporting system, allowing them to optimize their maintenance and efforts in the event of asset breakdown.

Automotive

With the current trend of technology, cars are increasingly becoming digital computers on wheels. They are constantly gathering data about driving behaviour and component conditions. Its compilation with advanced analytics of real-time weather data gives a complete picture of how weather conditions impact driver behaviour and safety in different scenarios.

Future scope of IoT based weather monitoring system

The technology of IoT has expanded in all sectors, and with the future scope and advantages of IoT-based weather monitoring systems, numerous industries can leverage them.

- The IoT weather reporting system has an application for farmers where they can ensure higher productivity of crops and lower the risk of weather hazards via the IoT weather.
- The IoT-based weather station proves helpful for monitoring the weather in areas like places with volcanoes or rain forests. This is especially important with drastic changes in the weather conditions we are experiencing.
- The IoT weather monitoring system using IoT supporting controllers is fully automated and efficient. It does not require any manual labor or attention.
- You can plan and visit the places anytime you like with prior notification of the weather conditions. You can simply get the status of the weather condition and the air quality, etc.

Therefore, with the help of embedded devices and sensors, any environment can be converted to a smart environment for accumulating the data and analyzing the environment with real-time monitoring.

Hence, with such advances on the Internet of Things (IoT), organizations are focusing on understanding the impact of weather on their operations and finding cutting-edge analytics on how to control the impact of their business

IoT Home Automation

- Home automation is constructing automation for a domestic, mentioned as a sensible home or smart house. In the IoT home automation ecosystem, you can control your devices like light, fan, TV, etc.
- A domestic automation system can monitor and/or manage home attributes adore lighting, climate, enjoyment systems, and appliances. It is very helpful to control your home devices.
- It’s going to in addition incorporates domestic security such as access management and alarm systems. Once it coupled with the internet, domestic gadgets are a very important constituent of the Internet of Things.
- A domestic automation system usually connects controlled devices to a central hub or gateway.
- The program for control of the system makes use of both wall-mounted terminals, tablet or desktop computers, a smartphone application, or an online interface that may even be approachable off-site through the Internet.

- Smart Home automation refers to the use of technology to control and automate various functions in a home, such as lighting, heating, air conditioning, and security. In the context of IoT (Internet
  of Things) and M2M (Machine-to-Machine) communications, home automation systems can be controlled and monitored remotely through a network connection.
- One of the key benefits of IoT-enabled home automation is the ability to control and monitor a wide range of devices and systems from a single, centralized location, such as a smartphone or tablet. This can include everything from lighting and temperature control to security cameras and alarm systems.
- Another advantage of IoT-enabled home automation is the ability to remotely monitor and control devices, even when away from home. This can be useful for controlling energy consumption and ensuring the safety and security of the home.
- IoT-enabled home automation systems typically involve the use of smart devices, such as thermostats, light bulbs, and security cameras, that can be controlled and monitored through a centralized hub or app. These smart devices can communicate with each other and with the centralized hub using wireless protocols such as Zigbee, Z-Wave, and Bluetooth.
- In addition, IoT-enabled home automation systems can integrate with other smart home technologies, such as voice assistants like Alexa and Google Home, to provide additional functionality and convenience.
- Overall, IoT-enabled home automation can provide many benefits to homeowners, including increased convenience, energy efficiency, and security. However, it is important to ensure the security of these systems, as they may be vulnerable to hacking and other cyber threats.

Components :

Here, you will see the smart home components like smart lighting, smart appliances, intrusion detection, smoke/gas detector, etc. So, let's discuss it.

Component-1 :
Smart Lighting –

- Smart lighting for home helps in saving energy by adapting the life to the ambient condition and switching on/off or dimming the light when needed.
- Smart lighting solutions for homes achieve energy saving by sensing the human movements and their environments and controlling the lights accordingly.

Component-2 :
Smart Appliances –

- Smart appliances with the management are here and also provide status information to the users remotely.
- Smart washer/dryer can be controlled remotely and notify when the washing and drying are complete.
- Smart refrigerators can keep track of the item store and send updates to the users when an item is low on stock.

Component-3 :
Intrusion Detection –

- Home intrusion detection systems use security cameras and sensors to detect intrusion and raise alerts.
- Alert can we inform of an SMS or an email sent to the user.
- Advanced systems can even send detailed alerts such as an image shoot or short video clips.

Component-4 :
Smoke/gas detectors –

- Smoke detectors are installed in homes and buildings to detect smoke that is typically an early sign of Fire.
- It uses optical detection, ionization for Air sampling techniques to detect smoke.
- Gas detectors can detect the presence of harmful gases such as CO, LPG, etc.
- It can raise alerts in the human voice describing where the problem is.

IoT Case Studies in Smart Cities

The Internet of Things (IoT) is transforming the way cities operate and deliver services to their citizens. By connecting various devices and systems to the internet, cities can leverage data and insights to improve urban planning, transportation, energy, environment, safety, and more. Here, we share some of the most impressive IoT use cases and case studies in smart cities. See how IoT can help you develop smart parking meters that optimize parking availability and revenue, home automation systems that enhance comfort and security, and other innovative solutions that make your city smarter and more sustainable.

UNIT-IV

Advanced Topics

4.1.Introduction

4.2.Apache Hadoop

4.3.Using Hadoop Map Reduce for Batch Data Analysis

4.4.IEEE 802.15.4

4.4.1.The IEEE 802 Committee family of Protocols

4.4.2.The Physical layer

4.4.3.The Media Access Control Layer

4.4.4.Uses Of 802.15.4

What is Hadoop

Hadoop is an open source framework from Apache and is used to store process and analyze data which are very huge in volume. Hadoop is written in Java and is not OLAP (online analytical processing). It is used for batch/offline processing.It is being used by Facebook, Yahoo, Google, Twitter, LinkedIn and many more. Moreover it can be scaled up just by adding nodes in the cluster.

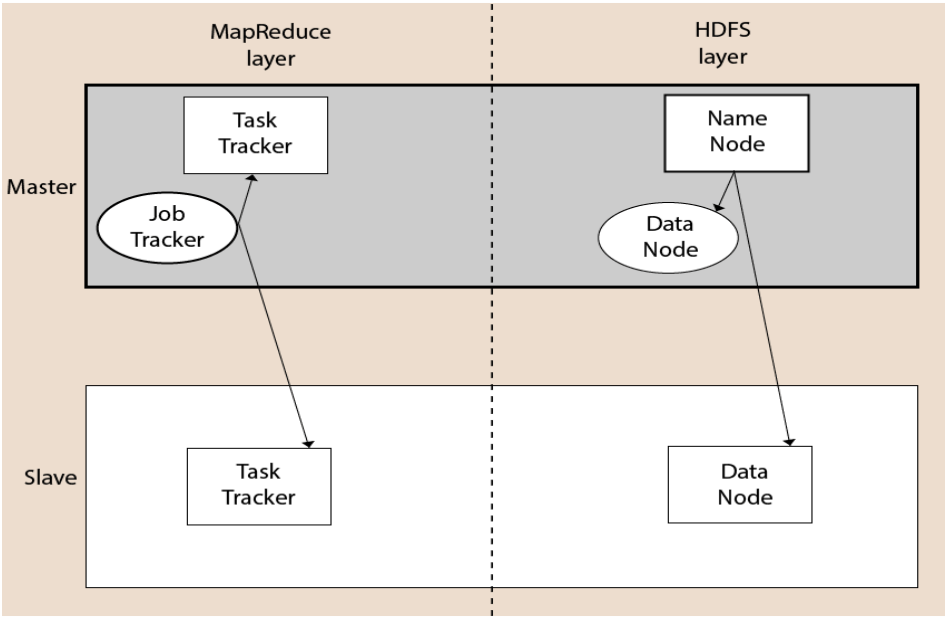
Modules of Hadoop

1. **HDFS:** Hadoop Distributed File System. Google published its paper GFS and on the basis of that HDFS was developed. It states that the files will be broken into blocks and stored in nodes over the distributed architecture.
2. **Yarn:** Yet another Resource Negotiator is used for job scheduling and manage the cluster.
3. **Map Reduce:** This is a framework which helps Java programs to do the parallel computation on data using key value pair. The Map task takes input data and converts it into a data set which can be computed in Key value pair. The output of Map task is consumed by reduce task and then the out of reducer gives the desired result.
4. **Hadoop Common:** These Java libraries are used to start Hadoop and are used by other Hadoop modules.

Hadoop Architecture

The Hadoop architecture is a package of the file system, MapReduce engine and the HDFS (Hadoop Distributed File System). The MapReduce engine can be MapReduce/MR1 or YARN/MR2.

A Hadoop cluster consists of a single master and multiple slave nodes. The master node includes Job Tracker, Task Tracker, NameNode, and DataNode whereas the slave node includes DataNode and TaskTracker.



Hadoop Distributed File System

The Hadoop Distributed File System (HDFS) is a distributed file system for Hadoop. It contains a master/slave architecture. This architecture consist of a single NameNode performs the role of master, and multiple DataNodes performs the role of a slave.

Both NameNode and DataNode are capable enough to run on commodity machines. The Java language is used to develop HDFS. So any machine that supports Java language can easily run the NameNode and DataNode software.

NameNode

- It is a single master server exist in the HDFS cluster.
- As it is a single node, it may become the reason of single point failure.
- It manages the file system namespace by executing an operation like the opening, renaming and closing the files.
- It simplifies the architecture of the system.

DataNode

- The HDFS cluster contains multiple DataNodes.
- Each DataNode contains multiple data blocks.
- These data blocks are used to store data.
- It is the responsibility of DataNode to read and write requests from the file system's clients.
- It performs block creation, deletion, and replication upon instruction from the NameNode.

Job Tracker

- The role of Job Tracker is to accept the MapReduce jobs from client and process the data by using NameNode.
- In response, NameNode provides metadata to Job Tracker.

Task Tracker

- It works as a slave node for Job Tracker.
- It receives task and code from Job Tracker and applies that code on the file. This process can also be called as a Mapper.

MapReduce Layer

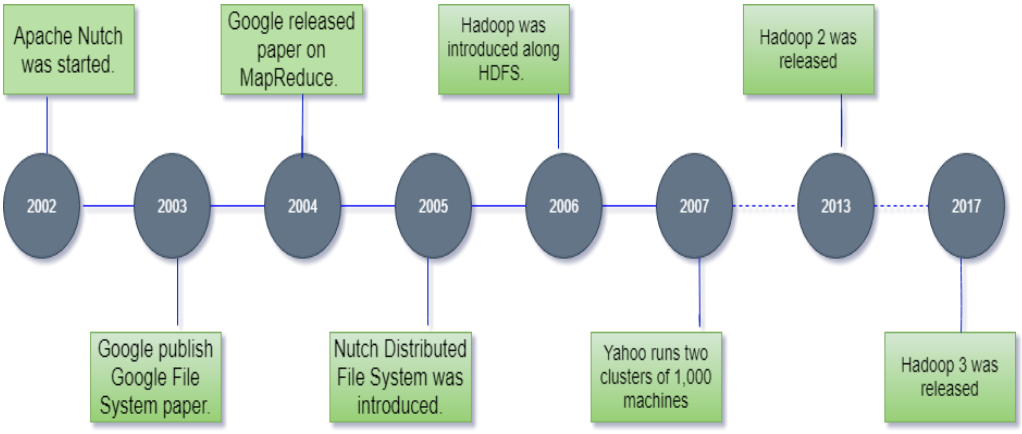
The MapReduce comes into existence when the client application submits the MapReduce job to Job Tracker. In response, the Job Tracker sends the request to the appropriate Task Trackers. Sometimes, the TaskTracker fails or time out. In such a case, that part of the job is rescheduled.

Advantages of Hadoop

- **Fast:** In HDFS the data distributed over the cluster and are mapped which helps in faster retrieval. Even the tools to process the data are often on the same servers, thus reducing the processing time. It is able to process terabytes of data in minutes and Peta bytes in hours.
- **Scalable:** Hadoop cluster can be extended by just adding nodes in the cluster.
- **Cost Effective:** Hadoop is open source and uses commodity hardware to store data so it really cost effective as compared to traditional relational database management system.
- **Resilient to failure:** HDFS has the property with which it can replicate data over the network, so if one node is down or some other network failure happens, then Hadoop takes the other copy of data and use it. Normally, data are replicated thrice but the replication factor is configurable.

History of Hadoop

The Hadoop was started by Doug Cutting and Mike Cafarella in 2002. Its origin was the Google File System paper, published by Google.



Let's focus on the history of Hadoop in the following steps: -

- In 2002, Doug Cutting and Mike Cafarella started to work on a project, **Apache Nutch.** It is an open source web crawler software project.

- While working on Apache Nutch, they were dealing with big data. To store that data they have to spend a lot of costs which becomes the consequence of that project. This problem becomes one of the important reason for the emergence of Hadoop.
- In 2003, Google introduced a file system known as GFS (Google file system). It is a proprietary distributed file system developed to provide efficient access to data.
- In 2004, Google released a white paper on Map Reduce. This technique simplifies the data processing on large clusters.
- In 2005, Doug Cutting and Mike Cafarella introduced a new file system known as NDFS (Nutch Distributed File System). This file system also includes Map reduce.
- In 2006, Doug Cutting quit Google and joined Yahoo. On the basis of the Nutch project, Dough Cutting introduces a new project Hadoop with a file system known as HDFS (Hadoop Distributed File System). Hadoop first version 0.1.0 released in this year.
- Doug Cutting gave named his project Hadoop after his son's toy elephant.
- In 2007, Yahoo runs two clusters of 1000 machines.
- In 2008, Hadoop became the fastest system to sort 1 terabyte of data on a 900 node cluster within 209 seconds.
- In 2013, Hadoop 2.2 was released.
- In 2017, Hadoop 3.0 was released.

Year	Event
2003	Google released the paper, Google File System (GFS).
2004	Google released a white paper on Map Reduce.
2006	• Hadoop introduced. • Hadoop 0.1.0 released. • Yahoo deploys 300 machines and within this year reaches 600 machines.
2007	• Yahoo runs 2 clusters of 1000 machines. • Hadoop includes HBase.
2008	• YARN JIRA opened • Hadoop becomes the fastest system to sort 1 terabyte of data on a 900 node cluster within 209 seconds. • Yahoo clusters loaded with 10 terabytes per day. • Cloudera was founded as a Hadoop distributor.
2009	• Yahoo runs 17 clusters of 24,000 machines. • Hadoop becomes capable enough to sort a petabyte. • MapReduce and HDFS become separate subproject.
2010	• Hadoop added the support for Kerberos. • Hadoop operates 4,000 nodes with 40 petabytes. • Apache Hive and Pig released.
2011	• Apache Zookeeper released. • Yahoo has 42,000 Hadoop nodes and hundreds of petabytes of storage.

2012 Apache Hadoop 1.0 version released.

2013 Apache Hadoop 2.2 version released.

2014 Apache Hadoop 2.6 version released.

2015 Apache Hadoop 2.7 version released.

2017 Apache Hadoop 3.0 version released.

2018 Apache Hadoop 3.1 version released.

4.4. The IEEE 802.15.4

- Committee Family of Protocols The Institute of Electrical and Electronics Engineers (IEEE) committee 802 defines physical and data link technologies.
- The IEEE decomposes the OSI link layer into two sub layers:

The media-access control (MAC) layer, sits immediately on top of the physical layer (PHY), and implements the methods used to access the network, typically the carrier sense multiple access with collision detection (CSMA/CD) used by Ethernet and the carrier-sense multiple access with collision avoidance (CSMA/CA) used by IEEE wireless protocols.
The logical link control layer (LLC), which formats the data frames sent over the communication channel through the MAC and PHY layers.

- IEEE 802.2 defines a frame format that is independent of the underlying MAC and PHY layers, and presents a uniform interface to the upper layers.

4.4.2.The Physical Layer

The design of 802.15.4 takes into account the spectrum allocation rules of the United States (FCC CFR 47), Canada (GL 36), Europe (ETSI EN 300 328-1, 328-2, 220-1) and Japan (ARIB STD T66). In the United States, the management and allocation of frequency bands is the responsibility of the Federal Communications Commission (FCC). The FCC has allocated frequencies for industrial scientific and medical (ISM) applications, which do

not require a license for all stations emitting less than 1 W. In addition, for low-power applications, the FCC has allocated the Unlicensed National Information Infrastructure (U-NII) band

MAC layer		BAND
802.11	WiFi	802.11, 802.11b, 802.11g, 802.11n : ISM 802.11a : U-NII
802.15.1	Bluetooth	ISM 2.4 GHz
802.15.4	ZigBee, SLowPAN	ISM 2.4 GHz worldwide ISM 902–928 MHz USA 868.3 MHz European countries 802.15.4a: 3.1–10.6 GHz
802.16	Wireless Metropolitan Access Networks Broadband Wireless Access (BWA) WiMax	802.16 : 10–66 GHz 802.16a: 2–11 GHz 802.16e: 2–11 GHz for fixed/2–6 GHz for mobile

Figure 1.1 IEEE-defined MAC layers.

Japan (ARIB STD T66). In the United States, the management and allocation of frequency bands is the responsibility of the Federal Communications Commission (FCC). The FCC has allocated frequencies for industrial scientific and medical (ISM) applications, which do not require a license for all stations emitting less than 1 W. In addition, for low-power applications, the FCC has allocated the Unlicensed National Information Infrastructure (U-NII) band.

FCC band	Maximum transmit power	Frequencies
Industrial Band	<1W	902 MHz–928 MHz
Scientific Band	<1W	2.4 GHz–2.48 GHz
Medical Band	<1W	5.725 GHz–5.85 GHz
U-NII	<40 mW	5.15 GHz–5.25 GHz
	<200 mW	5.25 GHz–5.35 GHz
	<800 mW	5.725 GHz–5.82 GHz

Figure 1.2 FCC ISM and U-NII bands.

IEEE 802.15.4 can use:

- The 2.4 GHz ISM band (S-band) worldwide, providing a data rate of 250 kbps (O-QPSK modulation) and 15 channels (numbered 11–26);
- The 902–928 MHz ISM band (I-band) in the US, providing a data rate of 40 kbps (BPSK modulation), 250 kbps (BPSK+O-QPSK or ASK modulation) or 250 kbps (ASK modulation) and ten channels (numbered 1–10)
- The 868–868.6 MHz frequency band in Europe, providing a data rate of 20 kbps

(BPSK modulation), 100 kbps (BPSK+O-QPSK modulation) or 250 kbps (PSSS: BPSK+ASK modulation), and a single channel (numbered 0 for BPSK or O-QPSK modulations, and 1 for ASK modulation).

Interferences with Other Technologies

Because the scientific band (2.4–2.48 GHz) is also unlicensed in most countries, this frequency band is used by many wireless networking standards, among which are WiFi (802.11, 802.11b, 802.11g, 802.11n), 802.15.4, and other devices such as cordless phonesand microwave oven

Frequency band	Modulation	Page number	Channel number and center frequency
2.4 GHz	0-QPSK	0	11 : 2405 MHz 12 : 2410 MHz 13 : 2415 MHz 14 : 2420 MHz 15 : 2425 MHz 16 : 2430 MHz 17 : 2435 MHz 18 : 2440 MHz 19 : 2445 MHz 20 : 2450 MHz 21 : 2455 MHz 22 : 2460 MHz 23 : 2465 MHz 24 : 2470 MHz 25 : 2475 MHz 26 : 2480 MHz
915 MHz	BPSK	0	1 : 906 MHz
	BPSK+ ASK	1	2 : 908 MHz
	BPSK+ 0–QPSK	2	3 : 910 MHz 4 : 912 MHz 5 : 914 MHz 6 : 916 MHz 7 : 918 MHz 8 : 920 MHz 9 : 922 MHz 10 : 924 MHz
868 MHz	BPSK	0	0 : 868.3 MHz
	BPSK+ ASK	1	1 : 868.3 MHz
	BPSK+ 0–QPSK	2	0 : 868.3 MHz

Figure 1.4 802.15.4 frequency bands, modulations and channels.

FHSS Wireless Standards

The 802.11 physical layer uses frequency hopping spread spectrum (FHSS) and

direct spread spectrum modulation. Bluetooth (802.15.1) uses FHSS in the ISM band. The FHSS technology divides the ISM band into 79 channels of 1 MHz. The FCC requires that a transmitter should not use any channel more than 400 ms at a time (dwell time), and should try to use at least 75 channels (but this may not always be possible if some channels are too noisy).

FHSS Channel	Frequency (GHz)		
2	2.401–2.402		
3	2.402–2.403		
4	2.403–2.404		
...			
80	2.479–2.480		

Figure 1.5 FHSS channels defined by the FCC in the S-Band.

DSSS Wireless Standards

802.11b and 802.11g use only direct spread spectrum (DSSS). 11 DSSS channels have been defined, each of 16 MHz bandwidth, with center frequencies of adjacent channels separated by 5 MHz. Only 3 channels do not overlap these channels should be used in order to minimize interference issues in adjacent deployments (3 channels are sufficient for a bidirectional deployment, however in tri dimensional deployments, for example, in a building, more channels would be required)

Choice of a 802.15.4 Communication Channel, Energy Detection, Link Quality Information

The 2.4 GHz frequency band is commonly used by the network and applications layers on top of 802.15.4, typically ZigBee and 6LoWPAN. The transmission power is adjustable from a minimum of 0.5 mW (specified in the 802.15.4 standard) to a maximum of 1 W (ISM band maximum). For obvious reasons, on links involving a battery-operated device, the transmission power should be minimized. A transmission power of 1 mW provides a theoretical outdoor range of about 300 m (100 m indoors).

DSSS channel	Frequency (GHz)
1	**2.404–(2.412)–2.420**
2	2.409–(2.417)–2.425
3	2.414–(2.422)–2.430
4	2.419–(2.427)–2.435
5	2.424–(2.432)–2.440
6	**2.429–(2.437)–2.445**
7	2.434–(2.442)–2.450
8	2.439–(2.447)–2.455
9	2.444–(2.452)–2.460
10	2.449–(2.457)–2.465
11	**2.456–(2.462)–2.470**

Figure 1.6 DSSS channels used by 802.11b.

Synchronous header (SHR)		Physical header (PHR)		Physical Service Data Unit
Preamble	SFD 111100101	Frame length (7 bits)	Ibit (reserved)	0 to 127 bytes

Figure 1.7 802.15.4 physical layer frame.

802.15.4 does not use frequency hopping (a technique that consumes much more energy), therefore the choice of the communication channel is important. Interference with FHSS technologies is only sporadic since the FHSS source never stays longer than 400 ms ona given frequency.

The 802.15.4 physical layer provides an energy detection (ED) feature that enables applications to request an assessment of each channel's energy level. Based on the results, a

802.15.4 network coordinator can make an optimal decision for the selection of a channel.

For each received packet, the 802.15.4 physical layer also provides link quality information (LQI) to the network and application layers (the calculation method for the LQIis proprietary and specific to each vendor).

Based on this indication and the number of retransmissions and lost packets, transmitters may decide to use a higher transmission power, and some applications for example, ZigBee Pro provide mechanisms to dynamically change the 802.15.4 channel in case the selected one becomes too jammed, however, such a channel switch should remain exceptional.

Sending a Data Frame

- 802.15.4 uses carrier-sense multiple access with collision avoidance (CSMA/CA): prior to sending a data frame, higher layers are first required to ask the physical layer to performs a clear channel assessment (CCA).
- The exact meaning of "channel clear" is configurable: it can correspond to an energy threshold on the channel regardless of the modulation (mode 1), or detection of

  802.15.4 modulation (mode 2) or a combination of both (energy above threshold and

  802.15.4 modulation: mode 3).
- After a random back-off period designed to avoid any synchronization of transmitters, the device checks that the channel is still free and transmits a data frame. Each frame is transmitted using a 30- to 40-bit preamble followed by a start

frame delimiter (SFD), and a minimal physical layer header composed only of a 7 bits frame length.

The Media-Access Control Layer

802.15.4 distinguishes the part of the MAC layer responsible for data transfer (the MAC common part sublayer or MCPS), and the part responsible for management of the MAC layeritself (the Mac layer management entity or MLME).

The MLME contains the configuration and state parameters for the MAC layer, such as the 64-bit IEEE address and 16-bit short address for the node, how many times to retry accessing the network in case of a collision (typically 4 times, maximum 5 times), how long to wait for an acknowledgment (typically 54 symbol duration units, maximum 120), or how many times to resend a packet that has not been acknowledged (0–7)

802.15.4 Reduced Function and Full Function Devices, Coordinators, and the PAN Coordinator

802.15.4 networks are composed of several device types:

- 802.15.4 networks are setup by a PAN coordinator node, sometimes simply called the coordinator. There is a single PAN coordinator for each network identified by its PAN ID. The PAN coordinator is responsible for scanning the network and selecting the optimal RF channel, and for selecting the 16 bits PAN ID (personal area network identifier) for the network. Other 802.15.4 nodes must send an association request for this PAN ID to the PAN coordinator in order to become part of the 802.14.4 network.
- **Full Function Devices (FFD),** also called coordinators: these devices are capable of relaying messages to other FFDs, including the PAN coordinator. The first coordinator to send a beacon frame becomes the PAN coordinator, then devices join the PAN coordinator as their parent, and among those devices the FFDs also begin to transmit a periodic beacon, or to respond to beacon requests. At this stage more devices may be able to join the network, using the PAN coordinator or any FFD as their parent. Reduced Function Devices (RFD) cannot route messages. Usually their receivers are switched off except during transmission. They can be attached to the network only as leaf nodes.
- **Reduced Function Devices (RFD)** cannot route messages. Usually their receivers

are switched off except during transmission. They can be attached to the network only as leaf nodes.

Two alternative topology models can be used within each network, each with its corresponding data-transfer method:

- **The star topology**: data transfers are possible only between the PAN coordinator and the devices.
- **The peer to peer topology**: data transfers can occur between any two devices. However, this is simple only in networks comprising only permanently listening devices. Peer to peer communication between devices that can enter sleep mode requires synchronization, which is not currently addressed by the 802.15.4 standard.

Each network, identified by its PAN ID, is called a cluster. A 802.15.4 network can be formed of multiple clusters (each having its own PAN ID) in a tree configuration: the root PAN coordinator instructs one of the FFD to become the coordinator of an adjacent PAN.

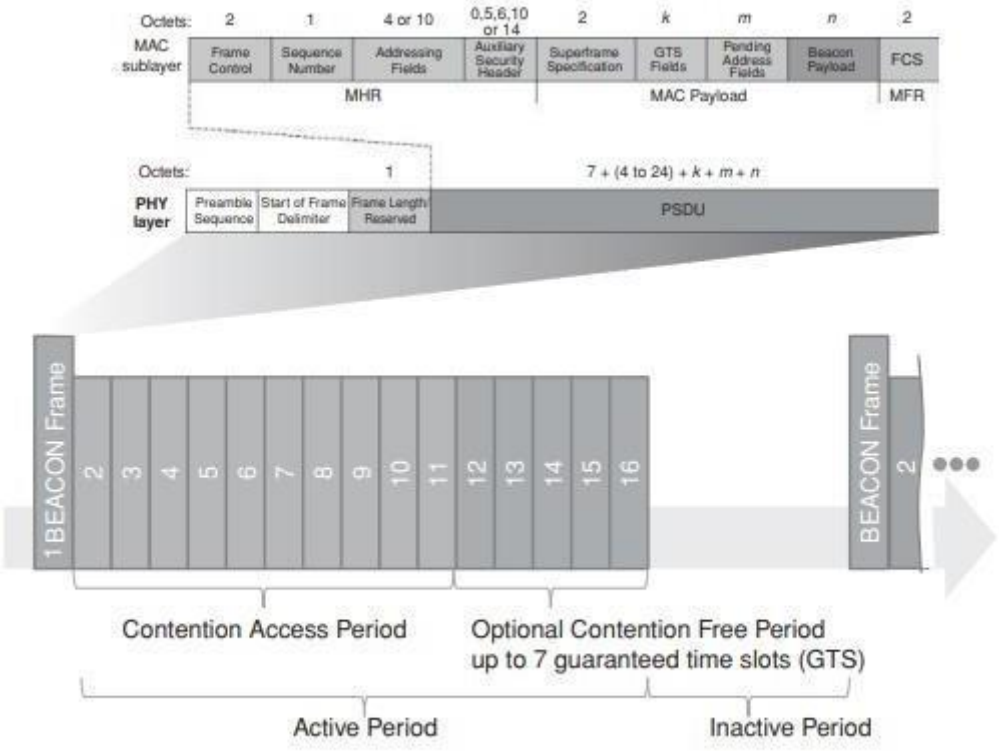


Figure 1.8 802.15.4 Superframe structure.

Each child PAN coordinator may also instruct a FFD to become a coordinator for another PAN, and so on.

The MAC layer specified by 802.15.4 defines two access control methods for the network:

- **The beacon-enabled access method** (or slotted CSMA/CA). When this mode is selected, the PAN coordinator periodically broadcasts a superframe, composed of a starting and ending beacon frame, 15 time slots, and an optional inactive periodduring which the coordinator may enter a low-power mode.

  The beacon frame starts by the general MAC layer frame control field, then includes the source PAN ID, a list of addresses for which the coordinator has pending

data, and provides super frame settings parameters. Devices willing to send data to a coordinator first listen to the super frame beacon, then synchronize to the super frame and transmit data either during the CAP using CSMA/CA, or during the CFP. Devices for which the coordinator has pending data should request it from the coordinator using a MAC data request command.

	Bytes	
Frame Control Field	2	000----------: Beacon frame 001----------: Data Frame 010----------: Ack Frame 011----------: Command frame ---1----------: Security enabled at MAC layer ----1----------: Frame pending -----1----------: Ack request ------1----------: PAN ID compression (source PAN ID omitted, same as destination) -------XXX------: reserved ----------XX----:Destination address mode *00 : PAN ID and destination not present (indirect addressing)* *01 : reserved* *10 : short 16-bit addresses* *11 : extended 64-bit addresses* ----------XX--:Frame version (00 : 2003, 01 : 2006) ----------XX:Source address mode
Sequence number	1	
Destination PAN ID	0 or 2	
Destination address	0 or 2 or 8	
Source PAN ID	0 or 2	
Source address	0 or 2 or 8	
Auxiliary security	variable	Contains security control, Frame counter, Key identifier fields
Payload	variable	
FCS	2	CRC 16 frame check sequence

Figure 1.9 802.15.4 MAC layer frame format.

- The **non beacon-enabled access method** (unslotted CSMA/CA). This is the mode used by ZigBee and 6LoWPAN. All nodes access the network using CSMA/CA. The coordinator provides a beacon only when requested by a node, and sets the beaconorder (BO) parameter to 15 to indicate use of the nonbeacon-enabled access method. Nodes (including the coordinator) request a beacon during the active scan procedure, when trying to identify whether networks are located in the vicinity, and what is their PAN ID.

01	Association request
02	Association response
03	Disassociation notification
04	Data request
05	PAN ID conflict notification
06	Orphan notification
07	Beacon request
08	Coordinator realignment
09	GTS request

Figure 802.15.4 command identifiers.

The devices have no means to know whether the coordinator has pending data for them, and the coordinator cannot simply send the data to devices that are not permanently listening and are not synchronized: therefore, devices should periodically(at an application defined rate), request data from the coordinator.

Association

A node joins the network by sending an association request to the coordinator's address. The association request specifies the PAN ID that the node wishes to join, and a set of capability flags encoded in one octet:

- **Alternate PAN:** 1 if the device has the capability to become a coordinator
- **Device type:** 1 for a full function device (FFD), that is, a device capable of becominga full function device (e.g., it can perform active network scans).
- **Power source**: 1 if using mains power, 0 when using batteries. Receiver on whiletransceiver is idle: set to 1 if the device is always listening.
- **Security capability:** 1 if the device supports sending and receiving secure MACframes.
- **Allocation address:** set to 1 if the device requests a short address from the

coordinator.

The coordinator assigns a 16-bit short address to the device (or 0xFFFE as a special code meaning that the device can use its 64-bit IEEE MAC address), or specifies the reason for failure (access denied or lack of capacity). Both the device and the coordinator can issue a disassociation request to end the association.

When a device loses its association with its parent (e.g., it has been moved out of range), it sends orphan notifications (a frame composed of a MAC header, followed by the orphan command code). If it accepts the reassociation, the coordinator should send a

realignment frame that contains the PAN ID, coordinator short address, and the device short address. This frame can also be used by the coordinator to indicate a change of PAN ID.

802.15.4 Addresses

EUI-64

- Each 802.15.4 node is required to have a unique 64-bit address, called the extended unique identifier (EUI-64).
- In order to ensure global uniqueness, device manufacturers should acquire a 24-bit prefix, the organizationally unique identifier (OUI), and for each device, concatenatea unique 40-bit extension identifier to form the complete EUI-64.
- In the OUI, one bit (M) is reserved to indicate the nature of the EUI-64 address (unicast or multicast), and another bit (L) is reserved to indicate whether the address was assigned locally, or is a universal address.

16-Bit Short Addresses

- Since longer addresses increase the packet size, therefore require more transmission time and more energy, devices can also request a 16-bit short address from the PAN controller.
- The special 16-bit address FFFF is used as the MAC broadcast address. The MAC layer of all devices will transmit packets addressed to FFFF to the upper layers.

1.3.3 802.15.4 Frame Format

The MAC layer has its own frame format,

The type of data contained in the payload field is determined from the first 3 bits of the framecontrol field:

- **Data frames** contain network layer data directly in the payload part of the MAC frame.
- **The Ack frame** format is specific: it contains only a sequence number and frame check sequence, and omits the address and data fields. At the physical layer, Ack frames are transmitted immediately, without waiting for the normal CSMA/CA clear

  channel assessment and random delays. This is possible because all other CSMA/CA transmissions begin after a minimal delay, leaving room for any potential Ack.
- The payload for command frames begins with a command identifier, followed by a command specific payload.

In its desire to reduce frame sizes to a minimum, 802.15.4 did not include an upper-layer protocol indicator field (such as Ethertype in Ethernet). This now causes problems, since bothZigBee and 6LoWPAN can be such upper layers.

Security

- 802.15.4 is designed to facilitate the use of symmetric key cryptography in order to provide data confidentiality, data authenticity and replay protection.
- It is possible to use a specific key for each pair of devices (link key), or a common key for a group of devices. However, the mechanisms used to synchronize and exchange keys are not defined in the standard, and left to the applications.
- The degree of frame protection can be adjusted on a frame per frame basis. In addition, secure frames can be routed by devices that do not support security.

CCM∗ Transformations

- 802.15.4 uses a set of security transformations known as CCM∗ (extension of CCM

defined in ANSI X9.63.2001), which takes as input a string "a" to be authenticated using a hash code and a string "m" to be encrypted, and delivers an output cipher text comprising both the encrypted form of "m" and the CBC message authentication code(CBC MAC) of "a".

- The transformations employed by CCM∗, which uses the AES block cipher algorithm E.

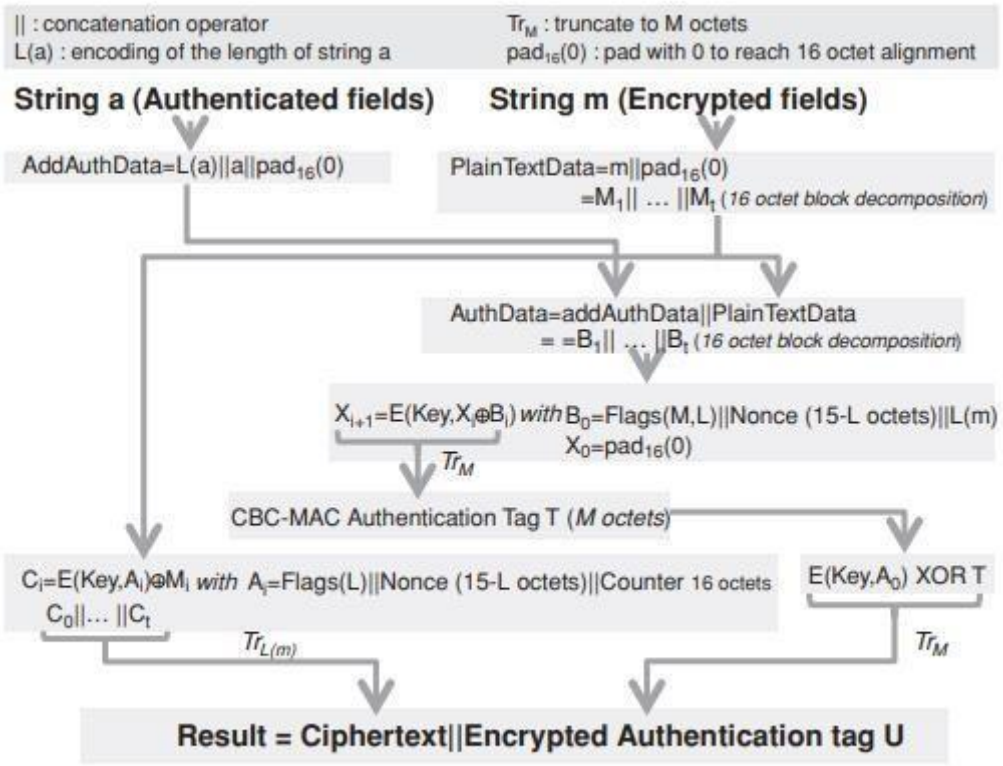


Figure 1.11 Overview of CCM* security transformations.

Security control field	Security attributes	Data confidentiality (data in "m" string)	Data authenticity (data in "a" string)
'000'	None	OFF	No
'001'	MIC-32	OFF	MHR, Auxiliary security header, Nonpayload fields, Unsecured payload fields
'010'	MIC-64	OFF	
'011'	MIC-128	OFF	
'100'	Encrypted fields	Unsecured payload fields	No
'101'	Encr. Fields+MIC-32		MHR, Auxiliary security header, Nonpayload fields
'110'	Encr. Fields +MIC-64		
'111'	Encr. Fields +MIC-128		

Figure Security control field codes.

In the case of 802.15.4, L = 2 octets, and the nonce is a 13-octet field composed of the 8-octet address of the device originating the frame, the 4-octet frame counter,

and the one-octet security-level code.

The Auxiliary Control Header

- The required security parameters are contained in the auxiliary control header, which is composed of a security control field (1 octet), a frame counter (4 octets) ensuring protection against replay attacks, and a key identifier field (0/1/5 or 9 octets).

- The first 3 bits of the security control field indicate the security mode for this data frame , the security mode determines the size of M in the CCM∗ algorithm (0, 4, 8 or 16 octets), and the data fields included in the "a" and "m" strings used for the computation of the final ciphertext (security attributes). The next 2 bits indicate the key identifier mode and the remaining bits are reserved.

Key identifier mode	Description	Key Identifier field length
'00'	Key determined implicitly from the originator and recipient of the frame	0
'01'	Key is determined from the 1-octet Key-index subfield of the Key identifier field, using the MAC layer default Key source	1
'10'	Key is determined explicitly from the 4-octet Key source subfield, and the 1-octet Key index subfield of the Key identifier field (part of the auxiliary security header)	5
'11'	Key is determined explicitly from the 8-octet Key source subfield, and the 1-octet Key index subfield of the Key identifier field (part of the auxiliary security header)	9

Figure Key identifier mode codes.

Key Selection

802.15.4 does not handle distribution of keys: the interface between the MAC layer and the key storage is a key lookup function, which provides a lookup string parameter that is used as an index to retrieve the appropriate key. The lookup material provided depends on thecontext.

- With implicit key identification (KeyIdMode = "00"), the lookup data is based on the

  802.15.4 addresses. The design implies that, in general, the sender indexes its keys

Addressing mode	Sender lookup data (based on *destination* addressing mode)	Receiver lookup data (based on *source* addressing mode)
Implicit	Source PAN short or extended address	Destination PAN short or extended address
Short	Destination PAN and destination node address	Source PAN and destination node address
Long	Destination node 802.15.4 8 octet extended address	Source node 802.15.4 8 octet extended address

according to destinations, and the receiver indexes its keys according to sources.

- With explicit key identification, the lookup data is composed of a key source identifier, and a key index. The design implies that the key storage is organized in several groups called key sources (one of which is the macDefaultKeySource). Each key source comprises several keys identified by an index.

The CCM standard specifies that a given key cannot be employed to encrypt more than 261 blocks, therefore the applications using 802.15.4 should not only assign keys, butalso change them periodically.

1.4 Uses of 802.15.4

802.15.4 provides all the MAC and PHY level mechanisms required by higher-level protocols to exchange packets securely, and form a network. It is, however, a very constrained protocol.

- It does not provide a fragmentation and reassembly mechanism. As the maximum packet size is 127 bytes (MAC layer frame),and the MAC headers and FCS will take between 6 and 19 octets, applications will need to be careful when sending unsecured packets larger than 108 bytes.

  Most applications will require security: the security headers add between 7 and 15 bytes of overhead, and the message authentication code between 0 and 16 octets. In the worst case, 77 bytes only are left to the application.
- Bandwidth is also very limited, and much less than the PHY level bitrate of 250 kbit/s. Packets cannot be sent continuously: the PHY layer needs to wait for Acks, and the CSMA/CA has many timers. After taking into account the PHY layer overheads (preamble, framing: about 5%) and MAC layer overheads (between 15 and 40%), applications have only access to a theoretical maximum of about 50 kbit/s, and only when no other devices compete for network access.

With these limitations in mind, 802.15.4 is clearly targeted at sensor and automation applications. Both ZigBee and 6LoWPAN introduce segmentation mechanisms that overcome the issue of small and hard to predict application payload

sizes at the MAC layer. An application like ZigBee takes the approach of optimizing the entire protocol stack, up to the application layer for use over such a constrained network. 6LoWPAN optimizes only the IPv6 layer and the routing protocols, expecting developers to make a reasonable use of bandwidth.

1.5 The Future of 802.15.4: 802.15.4e and 802.15.4g

there has been an increased focus on the use of 802.15.4 for mission critical applications, such as smart utility networks (SUN). As a result, several new requirements emerged: –

- The need for more modulation options, notably in the sub-GHz space, which is the preferred band for utilities who need long-range radios and good wireless building penetration.
- The need for additional MAC layer options enabling channel hopping, sampled listening and in general integrate recent technologies improving power consumption, resilience to interference, and reliability.

1.5.1 802.15.4e

Given typical sensor networks performance and memory buffers, it is generally considered that in a 1000-node network:

– Preamble sampling low-power receive technology allows one message per node every 100s;

– Synchronized receive technology allows one message per node every 33 s;

– Scheduled receive technology allows one message per node every 10 s.

Working group 15.4e was formed in 2008 to define a MAC amendment to 802.15.4:2006, which only supported the last mode, and on a stable carrier frequency. The focus of 802.15.4e was initially on the introduction of time-synchronized channel hopping, but in time the scope expanded to incorporate several new technologies in the 802.15.4 MAC layer. 802.15.4e also corrects issues with the 802.15.4:2006 ACK frame (no addressing information, no security, no payload) and defines a new ACK frame similar to a normal data frame except that it has an “ACK” type

Coordinated Sampled Listening (CSL)

- Sampled listening creates an illusion of “always on” for battery-powered nodes while keeping the idle consumption very low.

- This technology is commonly used by other technologies, for example, KNX-rf. The idea is that the receiver is switched on periodically but with a very low duty cycle
- e. On the transmission side, this requires senders to use preambles longer than the receiving periodicity of the target, in order to be certain that it will receive the preamble and keep the receiver on for the rest of the packet transmission.
- In 802.15.4e, CSL communication can be used between synchronized nodes (in which case the preamble is much shorter and simply compensates clock drifts), or between unsynchronized nodes in which case a long preamble is used (macCSLMaxPeriod)
- The latter case occurs mainly for the first communication between nodes and broadcast traffic: the 802.15.4e ACK contains information about the next scheduled

receive time of the target node, so the sender can synchronize with the receiver and avoid the long preamble for the next data packet

- 802.15.4e CSL uses a series of microframes ("chirp packets", a new frame type introduced in 15.4e) as preamble
- The microframes are composed of back-to-back 15.4 packets, and include a rendezvous time (RZtime) and optional channel for the actual data transmission: receivers need to decode only one chirp packet to decide whether the coming data frame is to their intention, and if so can decide to go back to sleep until RZtime and wake up again only to receive the data frame.
- CSL supports streaming traffic: a frame-pending bit in the 15.4e header instructs the receiver to continue listening for additional packets.

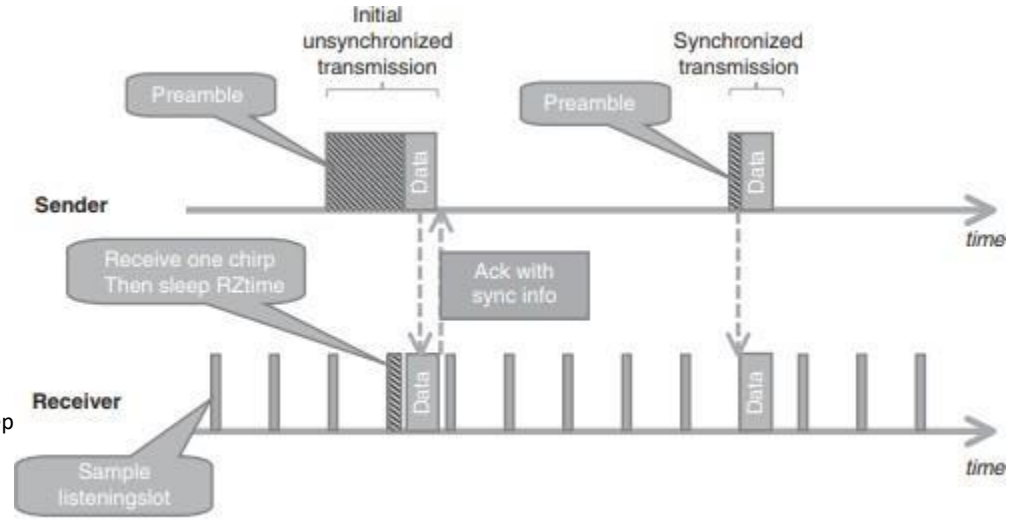


Figure 1.14 Overview of 802.15.4e CSL mode.

Receiver-Initiated Transmission (RIT)

The RIT strategy is a simple power-saving strategy that is employed by many existingwireless technologies:

- the application layer of the receiving node periodically polls a server in the network for pending data.
- When using the RIT mode, every macRitPeriod, the receiver broadcasts a datarequest frame and listens for a short amount of time (macRitDataWaitPeriod). The receiver can also be turned on for a brief period after sending data
- The downside of this approach is that the perceived receive latency is higher than in the CSL strategy, and multicast is not supported (must be emulated by multiunicast).

- The polling typically takes about 10 ms, so in order to achieve an idle duty cycle of 0.05% the macRITPeriod must be 20 s. RIT is adapted to sensor applications, which can tolerate long receive latency.

Time-Synchronized Channel Hopping (TSCH)

Channel hopping is a much-awaited feature of 802.15.4:

- It adds frequency diversity to other diversity methods (coding, modulation, retransmission, mesh routing), and will improve the resilience of 802.15.4 networks totransient spectrum pollution.
- In a multimode network, there are situations in which finding a common usable channel across all nodes is challenging. With channel hopping, each node to node linkmay use a specific set of frequencies.
- Channel hopping is supported in the new ACK frame, which contains synchronization information. In an uncoordinated peer to peer network, the channel hopping penalty is only for the initial transmission, as the sender will need to continue to send “chirp packets” on a given send frequency until it becomes aligned with the receiver frequency.
- After the first ACK has been received, the sender and the receiver are synchronized and the sender will select the sending frequency according to the channel schedule

of the receiver. If all joined nodes are in sync, then synchronizing to a single node is enough to be synchronized to the whole network.

- The time-synchronized channel hopping (TSCH) mode defined by 802.15.4e defines the operation model of a 802.15.4e network where all nodes are synchronized.
- The MAC layer of 802.15.4e nodes can be configured with several "slotframes", a collection of timeslots repeating in time characterized by the number of time slots in the cyclical pattern, the physical layer channel page supported, and a 27-bit channelMap indicating which frequency channels in the channel page are to be used for channel hopping.

The FFD nodes in a TSCH mode 802.15.4 network will periodically send advertisement frames that provide the following information: the PAN ID, the channel page supported by the physical layer, the channel map, the frequency-hopping sequence ID (predefined in the standard), the timeslot template ID3 (predefined in the standard), slot frame and link information, and the absolute slot number4 of the timeslot being used for transmission of this

advertisement frame. The advertisement frames are broadcast over all links configured to transmit this type of frame.

For PANs supporting beacons, synchronization is performed by receiving and decoding the beacon frames. For nonbeacon-enabled networks, the first nodes joining the network synchronize to the PAN coordinator using advertisement frame synchronization data, then additional nodes may synchronize to existing nodes in the network by processing advertisement frames. For networks using the time division multiple access mode, where precise synchronization of the whole network is essential, a new flag "clockSource" in the FFD state supports the selection of clock sources by 802.15.4e nodes without loops. A keep-alive mechanism is introduced to maintain synchronization.

1.5.2 802.15.4g

IEEE task group 802.15.4g focuses on the PHY requirements for smart utility networks (SUN).

802.15.4g defines 3 PHY modulation options:

- Multiregional frequency shift keying (MR-FSK): providing typically transmission capacity up to 50 kbps. "Multiregional" means that the standard maps a given channel page to a specific FSK modulation (2GFSK, 4GFSK . . .), frequency and bitrate. The current draft contains multiple variants for each region, implying that generic 802.15.4g radios will have to be extremely flexible.
- Multiregional orthogonal quadrature phase shift keying (O-QPSK): providing typically transmission capacity up to 200 kbps.
- Multiregional orthogonal frequency division multiplexing (OFDM): providing typically transmission capacity up to 500 kbps.

The number of frequency bands also increases to cover most regional markets:

- 2400–2483.5 MHz (Worldwide): all PHYs;
- 902–928 MHz (United States): all PHYs;
- 863–870 MHz (Europe): all PHYs;
- 950–956 MHz (Japan): all PHYs;
- 779–787 MHz (China): O-QPSK and OFDM;
- 1427–1518 MHz (United States, Canada): MR-FSK;
- 450–470 MHz, 896–901 MHz, 901–902 MHz, 928–960 MHz (United States): MR-FSK;
- 400–430 MHz (Japan);
- 470–510 MHz (China): all PHYs;
- 922 MHz (Korea): MR-OFDM.

802.15.4g is particularly interesting in Europe, where 802.15.4:2006 allowed a single channel (868.3 MHz). 802.15.4g now offers multiple channels:

- from 863.125 to 869.725 MHz in steps of 200 kHz (MR-FSK 200 kHz);
- from 863.225 to 869.625 in steps of 400 kHz (MR-FSK 400 kHz);
- from 868.3 to 869.225 MHz in steps of 400 kHz(O-QPSK);
- from 863.225 to 869.625 MHz in steps of 400 kHz (OFDM).

As the number of potential IEEE wireless standards and modulation options increases, the frequency scanning time would become prohibitively long if a coordinator was to scan all possible channels using all possible modulations. To solve this problem and improve coexistence across IEEE standards, 802.15.4g defines a new coex-beacon format, using a standard modulation method that must be supported by all coordinators (the common signaling mode or CSM defined in 802.15.4g).

UNIT- V

5. ZigBee

5.1.Development of the Standard

ZigBee Architecture

ZigBee and 802.15.4

ZigBee Protocol Layers

ZigBee Node Types

Association

5.2.Forming a Network

Joining a Parent Node in a Network Using 802.15.4 Association 5.3.3 Using NWK Rejoin

5.3.The ZigBee Network Layer

Short-Address Allocation

Network Layer Frame Format

Packet Forwarding

Routing Support Primitives

Routing Algorithms

5.4.The ZigBee APS Layer

Endpoints, Descriptors 106

The APS Frame

5.5.The ZigBee Device Object (ZDO) and the ZigBee Device Profile (ZDP)

ZDP Device and Service Discovery Services (Mandatory) 5.6.2 ZDP Network Management Services (Mandatory)

5.6.ZDP Binding Management Services (Optional)

Group Management

5.7.ZigBee Security

ZigBee and 802.15.4 Security

Key Types

The Trust Center

The ZDO Permissions Table

5.8.The ZigBee Cluster Library (ZCL)

Cluster

Attributes

Commands

ZCL Frame

ZigBee Application Profiles

The Home Automation (HA) Application Profile

ZigBee Smart Energy 1.0 (ZSE or AMI)

The ZigBee Gateway Specification for Network Devices

The ZGD

GRIP Binding

SOAP Binding

REST Binding

Example IPHA–ZGD Interaction Using the REST Binding

Development of the Standard

The 802.15.4 standard provides a physical and link layer technology optimized for low bitrate, low duty cycle applications. However, in practice sensor and control applications also need a mesh networking layer, and a standard syntax for application layer messages. In 2002, several companies decided to form the ZigBee alliance to build the missing standard layers that would be required to enable a multivendor mesh network on top of 802.15.4 radio links.

In 2008, the ZigBee alliance counted more than 200 members:

- **Promoter** members get early access to, contribute to and vote on the specifications of the alliance. They can veto decisions made by other participants in the alliance and get special marketing exposure in ZigBee events. New candidates for the promoter status must get co-opted by existing promoter members.
- **Participant** members have the same contribution and voting rights as promoters, but without veto rights.
- **Adopters** also get early access at the specifications, but can contribute only to the application profile working groups, and do not have voting rights.

The ZigBee alliance regularly organizes interop events, called ZigFests, and organize a developers conference twice a year. In order to ensure interoperability across vendors, the use of the ZigBee Compliant Platform (ZCP) certification and logo is reserved for products

passing the ZigBee test suite, which includes interoperability tests with the "Golden units"

(stacks from four reference implementations: Freescale, Texas Instruments, Ember, and Integration).

The deployment of many telecom standards either failed or was slowed down by multiple patent claims, many of which were not disclosed during the design phase of the standard. While the ZigBee alliance can do nothing against potential patent claims coming from nonmembers, it did verify that no technology included in the standard was subject of a known patent. In addition, every new member of the ZigBee alliance must sign a disclosure statement regarding patents that could potentially apply to ZigBee technology.

There are several versions of ZigBee. The current versions of ZigBee are ZigBee 2006/2007 (stack profile 0x01, ZigBee 2007 adds optional frequency agility and fragmentation), and ZigBee Pro (stack profile 0x02) that adds support for more nodes and more hops through source routing (it does not support tree routing), multicasting, symmetric links and a high security level. There was a ZigBee 2004 version, which is now deprecated.

ZigBee Architecture

5.2.1 ZigBee and 802.15.4

ZigBee sits on top of 802.15.4 physical (PHY) and medium-access control (MAC) layers, which provide the functionality of the OSI physical and link layers.

So far ZigBee uses only the 2003 version of 802.15.4. All existing ZigBee commercial devices use the 2.4 GHz S-Band as the 2003 version of 802.15.4 does not allow sufficient bandwidth on other frequencies. The 2006 version adds improved data-transfer rates for 868 MHz and 900 MHz but is not yet part of the ZigBee specification.

802.15.4 offers 16 channels on the 2.4 GHz, numbered 11 to 26. ZigBee uses only the nonbeacon-enabled mode of 802.15.4, therefore all nodes use CSMA/CA to access the network, and there is no option to reserve bandwidth or to access the network deterministically. ZigBee restricts PAN IDs to the 0x0000 – 0x3FFF range, a subset of the

802.15.4 PAN ID range (0x0000-0xFFFE). All unicast ZigBee commands request a hop by hop acknowledge (optional in 802.15.4), except for broadcast messages.

ZigBee Protocol Layers

The ZigBee network layer provides the functionality of the OSI network layer, adding the

missing mesh routing protocol to 802.15.4. It also encapsulates the network formation primitives of the 802.15.4 MAC layer (network forming and joining).

The rest of the ZigBee protocol layers do not follow the OSI model: The Application Support Sublayer (APS) layer has several functions:

• Multiplexing/demultiplexing: it forwards the network layer messages to the appropriate application objects, according to their endpoint ID (each application is allocated an endpoint ID).

• Binding: the APS layer maintains the local binding table, that is, records remote nodes and endpoints which have registered to receive messages from a local endpoint.

• 64-bit IEEE to 16-bit ZigBee network node address mapping.

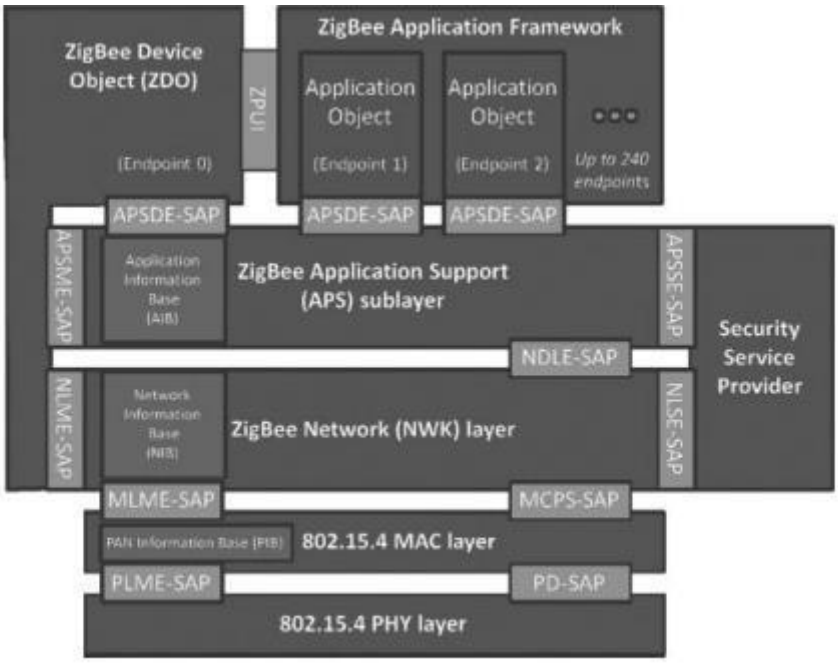


Figure 7.1 ZigBee architecture overview.

- Management of end to end acknowledgements. The application layer supports acknowledgements independently of the link layer acknowledgements of 802.14.4. The APS manages retries and duplicate filtering as required, simplifying application programming.
- Fragmentation. Also, as part of the application support sublayer management entity, or APSME:
- Group addressing: the APSME allows to configure the group membership tables of each endpoint ID, and forwards messages addressed to a group ID to the

application objects with relevant endpoint IDs

- .Security: management of keys.

– The ZigBee Device Object (ZDO) layer is a specific application running on endpoint 0, designed to manage the state of the ZigBee node. The ZDO application implements the interfaces defined by the ZigBee device profile (ZDP, application profile ID 0x0000). These

primitives encapsulate the 802.15.4 network formation primitives of the ZigBee network layer (node discovery, network joining), as well as additional primitives supporting thebinding.

– The ZigBee Cluster Library (ZCL) was a late addition to ZigBee, specified in a separate document. It consists in a library of interface specifications (cluster commands and attributes)that can be used in public and private application profiles.

considered as one of the key assets of ZigBee: while the ongoing evolution of ZigBee towards a 6LoWPAN-based networking layer is likely to replace the original networking layers of ZigBee, the ZCL is likely to remain the “lingua franca” of application developers. One important addition of the ZCL is the group cluster, which provides the network interface for group formation and management.

– The Application Framework layer provides the API environment of ZigBee application developers, and is specific of each ZigBee stack. Each application is assigned an Endpoint ID. The interfaces of ZigBee layers are called “service access points” (SAP), as in 802.15.4. One interface, the layer management entity ([layer name]-ME) is responsible for configuring internal data of the layer. Another interface, the data entity ([layer name]-DE), provides the data send/receive and other nonmanagement primitives.

ZigBee Node Types

The ZigBee node types listed below are not mutually exclusive. A given device could implement some application locally (e.g., a ZigBee power plug) acting as a ZigBee End Device, and also be a ZigBee router and even a ZigBee coordinator.

– ZigBee End-Device (ZED): this node type corresponds to the 802.15.4 reduced function device. It is a node with a low duty cycle (i.e. usually in a sleep state and not permanently listening), designed for battery operation. ZEDs must join a network through a router node, which is their parent.

– ZigBee router (ZR): this node type corresponds to the 802.15.4 full function device

(FFD). ZigBee routers are permanently listening devices that act as packet routers, once they have joined an existing ZigBee network.

– ZigBee Coordinator (ZC): this node type corresponds to a 802.15.4 full function device (FFD) having a capability to form a network and become a 802.15.4 PAN coordinator. ZigBee coordinators can form a network, or join an existing network (in which case they become simple ZigBee routers). In nonbeacon-enabled 802.15.4 networks, coordinators are

permanently listening devices that act as routers, and send beacons only when requested by a broadcast beacon request command.

The ZigBee coordinator also contains the trust center, which is responsible for admission of new nodes on the network and management of security keys

Association

Forming a Network

When forming a network, a ZigBee coordinator first performs an active scan (it sends beacon requests) on all channels defined in its configuration files. It then selects the channels with the fewer networks, and if there is a tie performs a passive scan to determine the quietest channel. It finally broadcasts a 802.15.4 beacon for the selected PAN ID on the selected radio channel, then remains silent (or repeats the beacon periodically, depending on the implementation). Depending on the configuration of the stack, the scan duration on each channel can range from 31 ms to several minutes, so the network-forming process can take significant time. If there are any ZigBee routers associated to the network, they will typically repeat the beacon with an offset in time relative to their parent's beacon (an extension of 802.15.4:2003).

The ZigBee specification allocates range 0x0000 to 0x3FFF for PAN IDs (a subset of the range defined by 802.15.4: 0x0000 to 0xFFFE). The PAN-ID should be unique for a given channel for networks not capable of dynamic channel change (ZigBee 2006), and unique on all channels if channel agility is enabled (ZigBee 2007, ZigBee PRO). A ZigBee coordinator beacon may also include an extended PAN ID (64 bit EPID), in addition of the 16-bit 802.15.4 PAN identifier, in order to facilitate vendor specific network selection for joining nodes. This EPID identifier is only used in the beacon frames and has no other

uses, while the 16-bit 802.15.4 PAN identifier is always used for joining and addressing purposes.

Joining a Parent Node in a Network Using 802.15.4 Association

ZigBee devices that are not yet associated either capture by chance the beacon, or try to locate a network by broadcasting a 802.15.4 beacon request on each of the 16 radio channels (active scan), unless the radio channel has been preconfigured or determined in the application profile. If a coordinator has formed a network on one of those channels, it responds to the beacon request by broadcasting a 802.15.4 beacon, which specifies the 16-bit PAN ID of the network, the address of the coordinator in short 16-bit format or extended 64-

bit format, and optionally an extended PAN ID (EPID). Any ZigBee router that has already joined the network will also respond with a beacon if they hear the beacon request.

The ZigBee payload of the 802.15.4 beacon also contains the ZigBee stack profile supported by the network, a flag indicating whether the responding node has remaining capacity for routers or end devices joining as new children, and the device depth of the sending device, that is, its level in the parent/child tree rooted at the coordinator.

Once it has discovered the PAN ID of the network, its radio channel, and the address of a router or coordinator within radio reach, the new ZigBee node sends a standard 802.15.4 association request command to the address of the specific parent node it wants to join as a child node (0x01 profile nodes must join the node with the smallest device depth). The association request message uses the extended 64-bit address of the joining node as the source address. Devices may wish to join a specific PAN ID, or may use a special PAN ID value 0xFFFF to signal that they are willing to join any PAN ID.

The parent node acknowledges the command, and then if it accepts the association responds with a 802.15.4 association response command sent to the extended address of the device. The association response specifies the 16 bit short address that the device

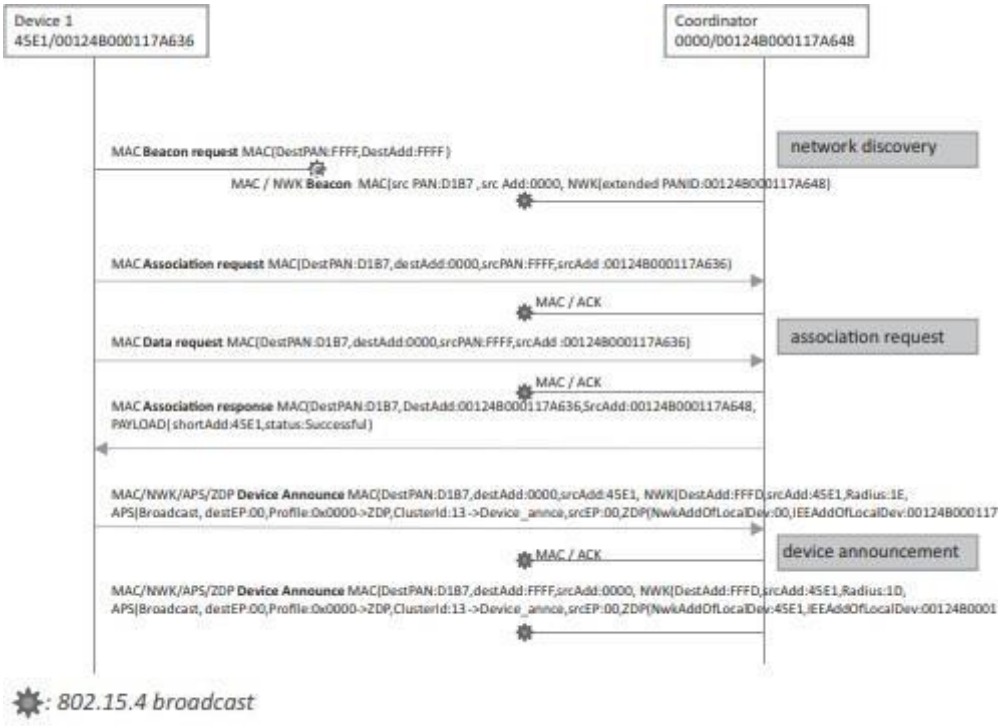


Figure 7.2 End device 1 joins the ZigBee network.

should use in the future (in order to save transmission time and therefore energy). The association response is acknowledged by the device. When the joining device is a sleeping

node (RxOnIdle=false), the association response is not sent immediately, but stored until the sleeping nodes polls it using a "data request".

Once associated, ZigBee devices usually send a data request command (now using the short 16-bit address assigned by the parent as source address) to its parent in order to receive any pending configuration data. After waiting for a response, battery-powered ZigBee end devices go to sleep until the next scheduled wake-up time or interrupt.

the joining device is within radio range of the coordinator. In a more general case, broadcast and unicast messages will be relayed by one or more ZigBee routers, and a new joining device may join the network from any location accessible through mesh networking

ZigBee joining, at the lowest level of security, only uses the "permit joining" flag of the beacon: nodes can join a network only when this flag is set in the beacon response. At the application level, most implementations allow administrators to "permit joining" for a limited amount of time, after which the network will not accept further joins. If the device is allowed to join and the nwkSecurityLevel parameter is set to 0x00, then the node becomes a new child of the parent node with relationship type 0x01 (child), otherwise set.

as type 0x05 (unauthenticated child). When security is enabled, interactions with the trust center (see Section 11.2.1) follow the unauthenticated joining process for key distribution. In order to facilitate commissioning, nodes may implement the commissioning ZDP cluster (see Section 7.6) to preconfigure security material and other parameters, and reset the node. Some nodes may be set up to join any network with permit-join enabled, or may be preconfigured to join the well-known commissioning network with extended PAN ID 0x00f0c27710000000. In theory, up to 31 100 nodes (9330 routers) can join a given network in stack profile 0x01, and over 64 000 nodes in stack profile 0x02.

Using NWK Rejoin

A device that loses connection to the network can attempt to rejoin using the ZigBee NWK layer rejoin command, which also triggers a beacon request. Since the NWK layer rejoin command use NWK layer security, the difference from a join based on 802.15.4 association is that no additional authentication step needs to be performed when security is enabled, and that nodes may rejoin any parent as long as it has available capacity, regardless of the status of the accept joining flag of the beacon. If it rejoins a different parent (e.g., because the original parent no longer responds), the node will be allocated a different short address, and must broadcast a device announce to the network in order to update bindingsthat may be configured in other nodes

After power cycles, most implementations do not immediately attempt an explicit rejoin in order to avoid network overloads, if they still have the address of their parent node and their own short address in nonvolatile memory. It is assumed that all nodes will restart in the same state as before the power cycle. An explicit rejoin is triggered only if the node fails to communicate with its parent. Such a procedure is often referred to as “silent rejoin”. It is also the default procedure, in ZigBee Pro/2007, when the coordinator triggers a channel change

The ZigBee Network Layer

The network layer is required for multihop routing of data packets in the mesh network, and is one of the key missing elements of 802.15.4. ZigBee uses the AODV public-domain mesh algorithm. The ZigBee network layer uses a specific data frame format, documented, which is inserted at the beginning of the 802.15.4 payload. 7.4.1 Short-Address Allocation ZigBee uses the 0x0000 – 0xFFF7 range for network node short addresses. The ZigBee coordinator uses short address 0x0000. The allocation of other networkaddresses, under control of the ZigBee Coordinator, depends on the routing technology in use:

ZigBee supports two address allocation modes:

– In stack profile 0x01, the network address depends on the position of the node in the tree. The distributed address assignment mechanism uses CSkip, a tree-based network address partition scheme designed to provide every potential parent with a subblock of network addresses. In addition to the default meshed routing, a tree-based routing can be used as a back-up (routers use the address allocation to decide whether to forward the packet to a parent or to a child).

– In stack profile 0x02 (ZigBee 2007, ZigBee Pro), a stochastic address assignment mechanism is used and ZigBee provides address-conflict detection and resolution mechanisms.

5.4.2 Network Layer Frame Format

The network layer PDU format , and is transported as 802.15.4 payload.

Table The ZigBee network layer frame format

Field name	Size (octets)	Field details
Frame Control	2	----------XX : Frame type (00 : network data) ----------0010-- : Protocol version (always 0x02 for ZigBee 2006/2007/Pro) --------XX------ : Route discovery (0x01:enable) -------X-------- : Multicast (0 : unicast) ------X--------- : Security (0 : disabled) -----X---------- : Source route (0 : not present) ----X----------- : Destination IEEE address (0 : not specified) ---X------------ : Source IEEE address (0 : not specified) 000------------- : Reserved
Dest. Address	2 or 8	0xffff broadcast to all nodes including sleeping devices 0xfffd broadcast to all awake devices (RxOnIdle = True) 0xfffc broadcast only to routers, not to sleeping devices
Source address	2 or 8	
Radius	1	Maximum number of hops allowed for this packet
Sequence number	1	Rolling counter
Payload	Variable	APS data, or network layer commands

7.4.3 Packet Forwarding

At the network layer, ZigBee packets can be:

– Unicast: the message is sent to the 16-bit address of the destination node

– Broadcast: if broadcast address 0xFFFF is used, the message is sent to all network nodes. If broadcast address 0xFFFD is used, the message is sent to all nonsleeping nodes. If broadcast address 0xFFFC is used, the message is broadcast to routers only (including the ZigBee coordinator). A radius parameter adjusts the number of hops that each broadcast

message may travel. The number of simultaneous broadcasts in a ZigBee network is limited by the size of the broadcast transaction table (BTT), which requires an entry for each broadcast in progress. The minimal size of the BTT is specified in ZigBee application profiles, forexample, 9 for HA.

– Multicast that is, sent to a 16-bit group ID.

ZigBee unicast packets are always acknowledged hop by hop (this is optional in 802.15.4). Broadcast packets are not acknowledged and are usually retransmitted several times by the ZigBee stacks of the originating node and by ZigBee routers on the path. Note that broadcast messages and messages sent to group IDs are not always broadcast at the link layer level: since sleeping nodes do not receive such messages, after wake up they send a data request to their parent node, and the queued messages are sent as unicast messages specifically to the sleeping node

Typically, a ZigBee node forwards a packet in about 10 ms. The propagation of data packets through the meshed network is limited by the initial value of Radius, a hop counter decremented at each routing node.

Delivery of packets to sleeping nodes uses the IEEE “Data request” packet. Parent nodes buffer received packets for their sleeping children. When it wakes up, the sleeping child sends a IEEE “Data request” packet to its parent. If it has data pending, the parent sets aspecific “more data” flag in the ACK response, then the pending data.

5.4.4 Routing Support Primitives

The network layer provides a number of command frames. The route request command enables a node to discover a route to the desired destination, and causes routers to update their routing tables. At the MAC level, the route request command is sent to the broadcast address (0xffff) and the current destination PAN ID. The response, if any, is a route reply command that causes routers on the path to update their routing tables.

Table ZigBee routing layer primitives

Command Frame Identifier	Command Name Reference
0x01	Route request
0x02	Route reply
0x03	Network Status
0x04	Leave
0x05	Route record
0x06	Rejoin request
0x07	Rejoin response
0x08	Link status
0x09	Network report
0x0a	Network update

Routing Algorithms

Broadcast, Groupcast, Multicast

At the 802.15.4 level, messages sent to multiple destinations are always broadcast. At the network layer, however, ZigBee offers more possibilities, depending on the destination address:

– 0xffff broadcast to all nodes including sleeping devices;

– 0xfffd broadcast to all awake devices (RxOnIdle = True);

– 0xfffc broadcast only to routers, not to end devices. In order to avoid 802.15.4 collisions, broadcast packet are relayed after a random delay of about 100 ms and therefore propagate

ten times more slowly than unicast messages. The radius parameter is decremented at each hop, so the broadcast propagation can be controlled with the initial radius value

ZigBee also uses a broadcast transaction table (BTT) in order to avoid any looping of broadcast messages: each broadcast packet is uniquely identified by its source address and network sequence number. When relaying a broadcast message, routers keep a copy of this unique identifier for 9 s (broadcast timeout), and will drop any looped packet. If the BTT is full, all broadcast messages are dropped. Routers that do not hear all neighbor routers retransmit a broadcast message may retransmit the broadcast message, implementing a form of implicit acknowledge mechanism.

Groupcast and multicast are implemented by the APS layer:

– APS messages sent to group addresses are filtered by the APS layer of the receiving node, so that only endpoints (and all of them) of member nodes will receive the message. However,all nodes receive the message (destination set to 0xffff at the network layer)

_ In ZigBee Pro, the radius is not decremented when the message is forwarded by a group member. This makes it possible to restrict the 802.15.4 broadcast propagation to group members only, allowing some slack (apsNonmemberRadius) in order to cope with disconnected groups. ZigBee Pro calls this “multicast”.

Neighbor Routing

This mode is not formally documented in ZigBee, but most vendors use it. If a router R already knows that the destination of a packet is a neighbor router or a child device

of R, it can send the packet directly to this node. ZigBee end devices, however, must always route outgoing packets to their parent.

Meshed Routing

This is the default routing model of ZigBee. It implements the advanced ad-hoc on-demand distance vectoring (AODV) algorithm.

The principle of AODV is illustrated

Node A needs to set up a route to node D. It broadcasts a route request to network address 0xfffc (routers only), which propagates through the network. Each ZigBee router that receives that message forwards it to its neighbors, adding their local estimation of quality of the link over which they received the route request to the path cost parameter of the route request. Note that the route we are discovering is in the A to D direction, while the path costs actually used are in the D to A direction. This is because the sender of the route request, which is broadcasting the message, cannot transmit different values of the link cost, therefore

the receiving node needs to update the path cost. ZigBee mesh routing assumes symmetrical link quality.