INTERNET OF THINGS

Module - 1

* **What is IoT?** Imagine a world where just about anything you can think of is online and communicating to other things and people in order to enable new services that enhance our lives. From self-driving drones delivering your grocery order to sensors in your clothing monitoring your health, the world you know is set to undergo a major technological shift forward. This shift is known collectively as the Internet of Things (IoT).

⇒ The basic premise & goal of IoT is to "Connect the unconnected". This means that objects that are not currently joined to a Computer N/w, namely the Internet, will be connected so that they can communicate and interact with people & other objects.

IoT is a technology transition in which devices will allow us to sense & control the physical world by making objects smarter & connecting them through an intelligent n/w.

⇒ When objects & machines can be sensed & controlled remotely across a n/w, a tighter integration between the physical world & computers is enabled. This allows for improvements in the areas of efficiency, accuracy, automation & the enablement of advanced applications.

⟹ Instead of viewing IoT as a single technology domain, it is good to view it as an umbrella of various Concepts, Protocols & technologies, all of which are at times somewhat dependent on a particular industry. While the wide array of IoT elements is designed to create numerous benefits in the areas of productivity & automation, at the same time it introduces new challenges, such as scaling the vast no. of devices & amounts of data that need to be processed.

* Genesis of IoT

The age of IoT is often said to have started between the years 2008 & 2009.

→ The person credited with the creation of the term "Internet of Things" is Kevin Ashton. While working for procter & Gamble in 1999, Kevin used this phrase to explain a new idea related to linking the Company's Supply chain to the Internet.

→ Kevin has subsequently explained that IoT now involves the addition of Senses to Computers. He was quoted as saying: "In the twentieth century, Computers were brains without Senses – they only knew what we told them." Computers dependent on humans to input data & knowledge through typing, bar Codes & So on. IoT is changing this paradigm: in the twenty-first Century, Computers are Sensing things for themselves.

→ It is widely accepted that IoT is a major technology shift, but what is its scale & importance? where does it fit in the evolution of the Internet?

→ As shown in below figure, the evolution of the Internet can be categorized into four phases.

[Evolutionary phases of the Internet]

These phases are further defined as below:

Internet phase	Definition
Connectivity (Digitize Access)	This phase Connected people to email, web Services, & Search so that information is easily accessed.
Networked Economy (Digitize Business)	This phase is enabled e-commerce & Supply chain enhancements along with Collaborative engagement to drive increased efficiency in business processes.
Immersive Experiences (Digitize interactions)	This phase extended the Internet experience to encompass widespread video & social media while always being Connected through mobility. more & more applications are moved into the cloud.

Internet of Things (Digitize the world)

This phase is adding Connectivity to objects & machines in the world around us to enable new services & experiences. It is Connecting the unconnected.

* IoT and Digitization

IoT & Digitization are terms that are often used interchangeably. In most contexts, this duality is fine but there are key differences to be aware of.

⇒ At a high level, IoT focuses on Connecting "things", such as objects & machines, to a Computer N/w, such as Internet. IoT is a well understood ~~thing~~ term used across the industry as a whole. on the other hand, digitization can mean different things to different people but generally encompasses the Connection of "things" with the data they generate and the business insights that result.

For example in a shopping mall where Wi-Fi location tracking has been deployed, the "things" are the Wi-Fi devices. Wi-Fi location tracking is simply the capab-ility of knowing where a Consumer in a retail environme-nt through his or her Smart phone's Connection to the retailer's Wi-Fi n/w. while the value of Connecting Wi-Fi devices or "things" to the Internet is obvious & appreciated by shopers, tracking real-time location of Wi-Fi clients provide a specific business benefit to the mall and shop owners. In this case, it helps the business understand

where shoppers tend to congregate and how much time they spend in different parts of a mall or store. Analysis of this data can lead to significant changes to the locations of product displays & advertising, where to place certain types of shops, how much rent to charge and even where to station security guards.

⟹ Digitization, as defined in its simplest form, is the conversion of information into a digital format. For ex, the whole photography industry has been digitized. Pretty much everyone has digital cameras these days either standalone devices or built into their mobile phones.

Other examples of digitization include the video rental industry and transportations. In the past, people went to a store to rent or purchase videotapes or DVDs of movies. With digitization, just about everyone is streaming video content or purchasing movies as downloaded files.

The transportation industry is currently undergoing digitization in the area of taxi services. Businesses such as Uber & Lyft use digital technologies to allow people to get a ride using a mobile phone app. This app identifies the car, the driver and the fare. The rider then pays the fare by using the app.

* IoT Impact

Projections on the potential impact of IoT are impressive. About 14 billion, or just 0.06%, of "things" are connected to the Internet today. Cisco Systems predicts that

by 2020, this number will reach 50 billion. A UK government report speculates that this number could be even higher, in the range of 100 billion objects connected. Cisco further estimates that these new connections will lead to $19 trillion in profits & cost savings.

* Convergence of IT and OT

IT – Information Technology
OT – Operational Technology

⇒ IT supports connections to the Internet along with related data & technology systems & is focused on the secure flow of data across an organization.

OT controls & monitors devices & processes on physical operational systems. These systems includes assembly lines, utility distribution networks, production facilities, roadway systems & many more. Typically IT did not get involved with the production & logistics of OT environments.

⇒ Specifically, the IT organizations is responsible for the information systems of a busines, such as email, file & point services, databases & so on. In comparison, OT is responsible for the devices & processes acting on ~~individual~~ industrial equipment such as factory machines, meters, actuators, electrical distribution automation devices, SCADA (Supervisory control & data acquisition systems) & so on.

The below table highlights some of the differences between IT & OT nlws & their various challenges.

Criterion	Industrial OT Nlw	Enterprise IT Nlw
Operational focus	Keep the business operating 24x7	Manage the computers, data & employee Comms System in a Secure way.
priorities	1. Availability 2. Integrity 3. Security	1. Security 2. Integrity 3. Availability
Types of data	Monitoring, Control & Supervisory data	Voice, video, transactional & bulk data
Security	Controlled physical access to devices	Devices & users authenticated to the Nlw.
Implication of failure	OT nlw disruption directly impacts busines	Can be business impacting, depending on industry but workarounds may be possible
Nlw upgrades (SW or HW)	only during operational maintainance windows	often requires an outage window when workers are not onsite; impact can be mitigated.
Security vulnerability	Low: OT nlws are isolated & often use proprietary protocols	High: Continual patching of hosts is required, & the nlw is connected to Internet and requires vigilant protection.

[Comparing OT & IT]

* IoT challenges - The following table highlights few of the most significant challenges & problems that IoT is currently facing.

Challenges	Description
Scale	While the scale of IT n/w can be large, the scale of OT can be several orders of magnitude larger. For eg, one large electrical utility in Asia recently began deploying IPv6-based smart meters on its electrical grid. While this utility company has tens of thousands of employees, the no. of meters in the service area is tens of millions. This means the scale of the n/w the utility is managing has increased by more than 1000 fold!
Security	With more "things" becoming connected with other "things" & people, security is an increasingly complex issue for IoT. Your threat surface is now greatly expanded & if a device gets hacked, its connectivity is a major concern.
Privacy	As sensors become more prolific (Possible) in our daily lives, much of the data they gather will be specific to individuals & their activities. This data can range from health info to shopping patterns & transactions at a retail establishment. For a business, this data has monetary (Money) value.
Big Data & Data analytics	IoT & its large no. of sensors is going to trigger a deluge (Flood) of data that must be handled

This data will provide critical information & insights if it can be processed in an efficient manner. The challenge, however is evaluating massive amounts of data arriving from different sources in various forms & doing so in a timely manner.

Interoperability — As with any other nascent (emerging) technology, various protocols & architectures are jockeying for market share & standardization within IoT. Some of these protocols & architectures are based on proprietary (privately owned) elements & others are open. Recent IoT standards are helping minimize this problem, but there are often various protocols & implementations available for IoT devices.

* IoT Network Architecture and Design

Drivers Behind New Network Architectures

Imagine an experienced architect who has built residential houses for his whole career. He is an expert in this field & knows exactly what it takes to not only make a house architecturally attractive but also to be functional & livable & meet the construction codes mandated by local govt. One day this architect is asked to take on a new project: Construct a massive stadium that will be a showpiece for the city & which will support a variety of sporting teams, concerts, & community events & which has a seating capacity of 60000+.

→ While the architect has extensive experience in designing homes, those skills will clearly not be enough to meet the demands of this new project.

⟹ The difference between IT & IoT n/ws is much like the difference between residential arch. & Stadium arch.

→ The key difference between IT & IoT is the data. While IT systems are mostly concerned with reliable & continuous support of busines applications such as email, web, databases, CRM systems & so on, IoT is all about the data generated by sensors & how that data is used. The essence of IoT architectures thus involve how the data is transported, collected, analyzed & ultimately acted upon.

following table covers some of the differences between IT & IoT n/ws, with a focus on the IoT requirements that are driving new n/w architectures & considers what adjustments are needed.

Challenge	Description	IoT architectural change required
Scale	The massive scale of IoT End-points (Sensors) is far beyond that of typical IT n/ws.	The IPv4 address space has reached exhaustion & is unable to meet IoTs scalability requirements. Scale can be met only by using IPv6. IT n/ws continue to use IPv4 through features like N/w Address Translation (NAT).

Security	IoT devices especially those on wireless sensor nws are often physically exposed to the world.	Security is required at the every level of the IoT nws. Every IoT endpoint node on the nws must be part of the overall security strategy & must support device level authentication & link encryption. It must also be easy to deploy with some type of zero-touch deployment model.
Devices & Nws Constrained by power, CPU, memory, & link speed	Due to massive scale & longer distances, the nws are often constrained, lossy, & capable of supporting only minimal data rates	New last-mile wireless technologies are needed to support constrained IoT devices over long distances. The nws is also constrained, meaning modifications need to be made to traditional nw-layer transport mechanisms.
The massive volume of data generated	The sensors generate a massive amount of data on a daily basis, causing nw bottlenecks & slow analytics in the cloud.	Data analytics capabilities need to be distributed throughout the IoT nw, from the edge to the cloud. In traditional IT nws, analytics & applications typically run only in the cloud.

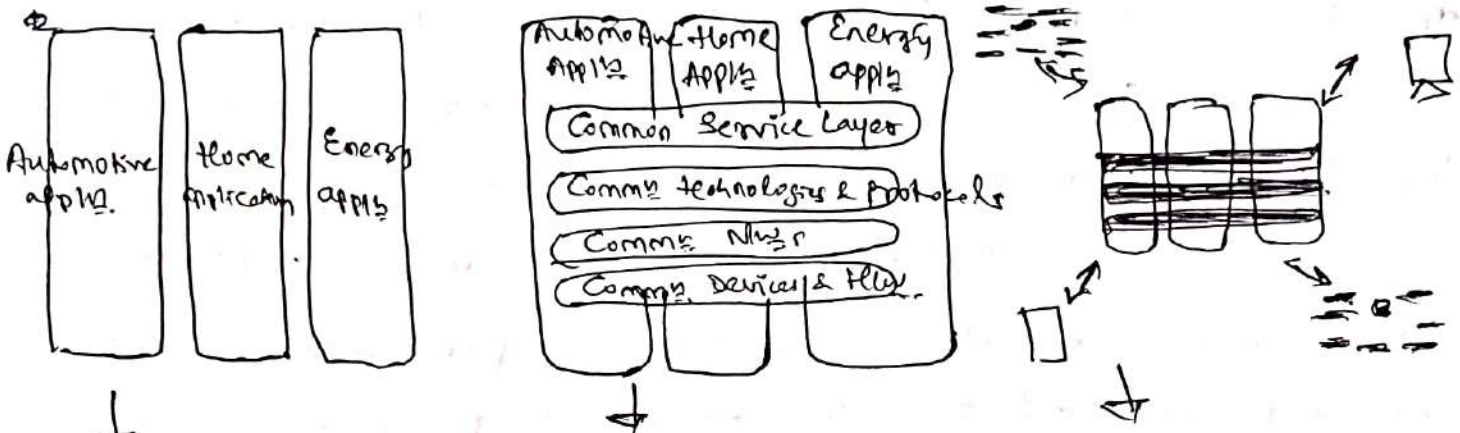
Support for legacy devices	An IoT n/w often comprises a collection of modern, IP-capable endpoints as well as legacy, non-IP devices that rely on serial or proprietary protocols.	Digital transformation is a long process that may take many years, & IoT n/ws need to support protocol translation and/or tunneling mechanisms to support legacy protocols over standards-based protocols, such as Ethernet & IP.
The need for data to be analyzed in real time	whereas traditional IT n/ws perform scheduled batch processing of data, IoT data needs to be analyzed & responded in real-time.	Analytics s/w needs to be positioned closer to the edge & should support real-time streaming analytics. Traditional IT analytics s/w (such as relational databases or even Hadoop) are better suited to batch-level analytics that occur after the fact.

* Comparing IoT Architectures

In an effort to standardize the rapidly growing field of machine-to-machine (M2M) communications, the European telecommunications Standards Institute (ETSI) created the M2M Technical Committee in 2008. The goal of this committee was to create a common architecture that would help accelerate the adoption of M2M applications & devices. Over the time, the scope has expanded to include the Internet of things.

The oneM2M arch. divides IoT functions into three major domains: the appln layer, the services layer & the n/w layer.

While this arch. may seem simple & somewhat generic at first glance, it is very rich & promotes interoperability through IT-friendly APIs & supports a wide range of IoT technologies.



Appln Layer: The oneM2M arch. gives major attention to connectivity between devices & their applications. This domain includes the appln-layer protocols & attempts to standardize northbound API definitions for interaction with Business Intelligence (BI) systems. Applns tend to be industry-specific & have their own sets of data models, & thus they are shown as vertical entities.

Services Layer: This layer is shown as horizontal framework across the vertical industry applications. At this layer, horizontal modules include the physical nw that the IoT appln runs on, the underlying management protocols, & the HW. Examples includes backhaul comms via cellular, MPLS nws, VPNs & so on. Riding on top is the common services layer. This conceptual layer adds APIs & middleware supporting third-party services & applications.

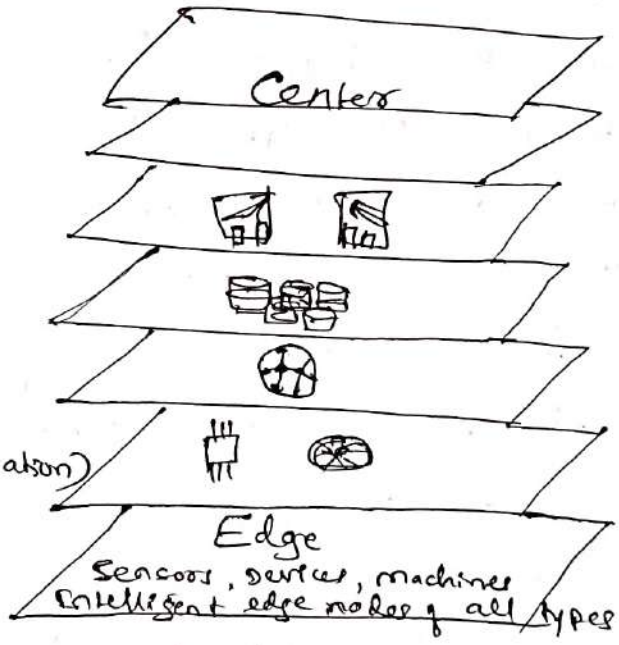
Network Layer: This is the comms domain for the IoT devices & end points. It includes the devices themselves & the comms n/w that links them.

The IoT World Forum (IoTWF) Standardized Architecture

In 2014 the IoTWF architectural committee (led by Cisco, IBM, Rockwell Automation, & others) published a seven-layer IoT architectural reference model. While various IoT reference models exists, the one put forth by the IoT world forum offers a clean, simplified perspective on IoT & includes edge computing, data storage, & access. It provides a succinct way of visualizing IoT from a technical perspective. Each of the seven layers is broken down into specific funs & & the security encompasses the entire model.

Levels

(7) Collaboration & Processes (Involving people & Business processes)
(6) Application (Reporting, Analytics, control)
(5) Data Abstraction (Aggregation & Access)
(4) Data Accumulation (Storage)
(3) Edge computing (Data Element Analysis & Transformation)
(2) Connectivity (Communication & processing units)
(1) Physical devices & Controllers (The "Things" in IoT)



[IoT Reference model published by the IoT World Forum]

⇒ As shown in above figure, IoT Reference model defines a set of levels with control flowing from the center (this cloud be either a cloud service or a dedicated data center), to the edge, which includes sensors, devices

machines & other types of intelligent end nodes. In general, data travels up the stack, originating from the edge & goes northbound to the Center. Using this reference model, we are able to achieve the following:

↳ Decompose the IoT problem into smaller parts
↳ Identify different technologies at each layer & how they relate to one another
↳ Define a system in which different parts can be provided by different vendors
↳ Have a process of defining interfaces that lead to interoperability
↳ Define a tiered security model that is enforced at the transition points between levels.

Layer 1: Physical Devices & Controllers Layer

This layer is home to the "things" in the Internet of Things, including the various endpoint devices & sensors that send & receive information. The size of these "things" can range from almost microscopic sensors to giant machines in a factory. Their primary job is generating data & being capable of being queried and/or controlled over a n/w.

Layer 2: Connectivity Layer

The most important function of this IoT layer is the reliable & timely transfer of data, more specifically, this includes transmissions between layer 1 devices & the n/w & between the n/w & information processing that occurs at layer 3 (the edge computing layer)

Connectivity - (Communication & processing units) Sensors, gateways, routers & n/w.

Layer 2 functions :

↳ Communications between layer 1 devices
↳ Reliable delivery of information across the n/w
↳ Switching & routing
↳ Translation betn protocols
↳ N/W level Security

Layer 3 : Edge Computing layer

Edge computing is often referred to as the "fog" layer. At this layer, the emphasis is on data reduction & converting n/w data flows into info. that is ready for storage & processing by higher layers. one of the basic principle of this reference model is that information processing is initiated as early & as close to the edge of the n/w as possible.

Layer 3 functions :

↳ Evaluate & Reformat Data for processing at higher levels.
↳ Filter data to reduce traffic higher level processing
↳ Assess data for alerting, Notification, or other actions.

Layer 4 : Data accumulation layer

Captures data & stores it so it is usable by applications when necessary. Converts event-based data to query based processing.

Layer 5 : Data abstraction layer

Reconciles multiple data formats & ensures (restore) (consistent) Coasistent Semantics from various sources. Confirms the dataset is Complete & Consolidates data into one place or multiple data stores using virtualization.

Layer 6 : Applications Layer

Interprets data using Software applications. Applications may monitor, Control & provide reports based on the analysis of data

Layer 7 : Collaboration & processes layer

Consumes & shares the appln information, Collaborating on & Communicating IoT info often requires multiple steps & it is what makes IoT useful. This layer can change business processes & delivers the benefits of IoT.

---

backhaul (backbone) — set of links that connect core n/ws with smaller subnetworks

MPLS - multiprotocol Label Switching - technique where data is directed from one node to next node based on labels rather than n/w addresses (protocol-agnostic)

Benefits - Speed & shape traffic.

VPN - creates a secure tunnel b/w user's computer & VPN server, which hides their online activity, location, masks your IP address, etc.

Edge Computing - is a distributed computing paradigm that brings computation & data storage closure to the sources of data

→ Edge computing brings data processing closer to the devices & sensors that generate it.

Fog computing extends the capabilities of an edge computing to a larger n/w of devices & sensors.

Ex Edge Computing - Smart phone connected to cloud

Fog Computing — IIOT environment in manufacturing plant.

Automotive applns — Data logging, RADAR tracking, mechanical testing, Vibration analysis, Ignition monitoring, component testing.

Asset tracking — Scanning barcodes, ~~using~~ GPS, RFID

Fleet management — Fleet - group of ships sailing together engaged in some activity under some ownership.

Energy applications — Heating & Cooling, driving cars, Lighting house, manufacturing products etc.

Event-based vs Query based — Queries retrieve data, need to be real-time & hence need some creativity to scale

Events describe a fact that has happened in the past, dont need to be handled real-time & hence scale well.

Data Abstraction — is the process of hiding unwanted or irrelevant details from the end-user.

Ex) Sort function without knowing algorithm behind
) Print in usage without knowing the code behind

---

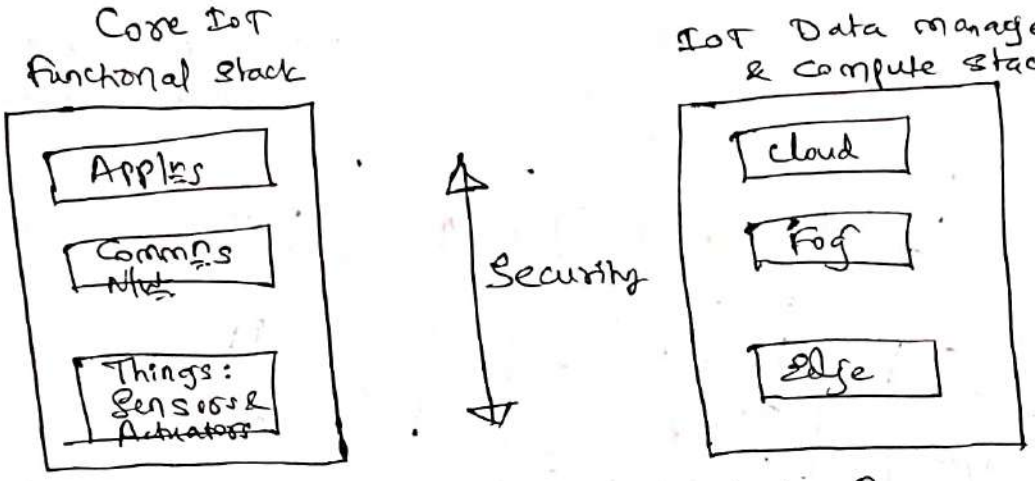
* IT & OT responsibilities in the IoT reference model

7		Query Based	Data at rest	Non real-time
6	IT			
5				
4	OT	Event Based	Data in motion	real time
3				
2				
1				

* The top of the stack is in the IT area & includes things like the servers, databases & applications.
* At the Bottom, in the OT layers the devices generate real-time data at their own rate
* Data has to be buffered or stored at certain points within the IoT stack.
* The real-time "data in motion" close to the edge has to be organized & stored so that it becomes "data at rest" for the applications in the IT tiers. The IT & OT organizations need to work together for overall data management.

* A Simplified IoT Architecture

The below figure shows Simplified IoT architecture.



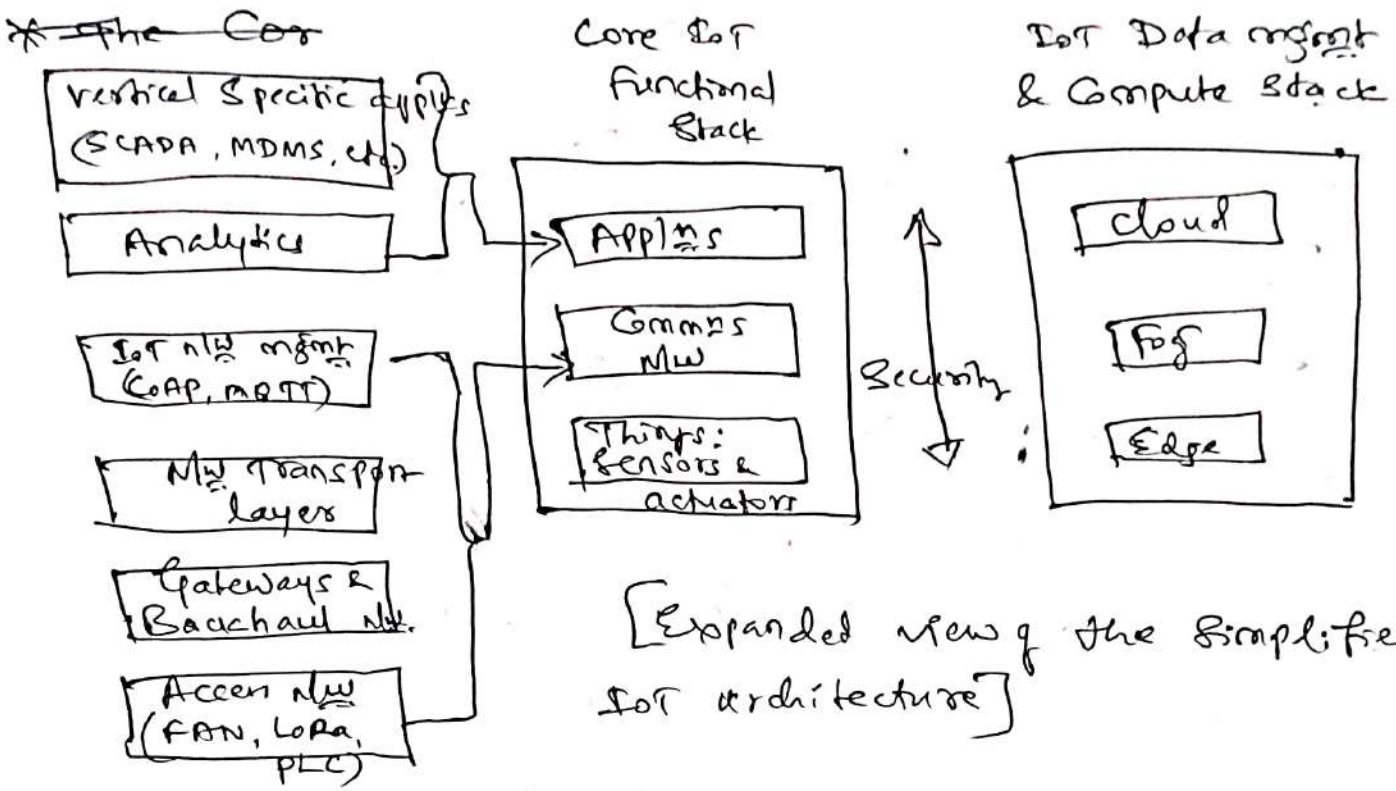
(Simplified IoT Architecture)

→ Nearly every published IoT model includes core layers similar to those shown on the left side of above figure including "things", a communications Nw & applications. However unlike other models, the framework presented here separates the Core IoT & data management into parallel & aligned stacks, allowing you to carefully examine the fns of both the nw & the applications at each stage of a complex IoT system. This seperation gives you better visibility into the functions of each layer.

→ Following figure shows the expanded view of the Simplified IoT Architecture. As shown, the Core IoT functional stack can be expanded into sublayers containing greater detail & specific nw functions. For example, the communication layer is broken down into four seperate sublayers: the access nw, gateways & backhaul, IP transport & operations & management sublayers.

The application layer of IoT nws is quite different from the appln layer of a typical enterprise nw. Instead

of simply using business applications, IoT often involves a strong big data analytics component.



[Expanded view of the simplified IoT architecture]

SCADA (Supervisory Control & Data Acquisition) – is a system of S/W & H/W elements that allows industrial organizations to: Control industrial processes, monitor, gather & process real-time data

MDM (Mobile Device mgmt) – to optimize the functionality & security of mobile devices.

Analytics – use of data analysis tools & techniques to extract value from massive data volumes generated by connected IoT devices.

CoAP – Constrained App protocol, used for resource-constrained, low-power devices in lossy nlws, especially optimized for deployments with a high no. of end devices within the nlw.

MQTT – (Message Queuing Telemetry Transport) is a messaging protocol for restricted low-bandwidth nlws & extremely high latency IoT devices.

Gateway – is a physical device that connects sensors, IoT modules & Smart devices to the cloud.

Backhaul – n/w or links connect to the Core (backbone) n/w (ex cell phone/smart phone connects to the Internet by receiving data from a cell tower or BS.

FAN – Fixed Access N/w

PLC – programmable logic Controller

LoRa – long range data links (Ref modulation technology)

* The Core IoT functional Stack

From an architectural standpoint, several Components have to work together for an IoT n/w. to be Operational:

↳ "Things" Layer: At this layer, the physical devices need to fit the Constraints of the environment in which they are deployed while still being able to provide the info needed.

↳ Communications N/w layer: when smart objects are not self-contained, they need to communicate with an external system. In many cases, this Comm. uses a wireless technology. this layer has four Sublayers:

↳ Access n/w Sublayer – This is typically made up of Wireless technologies. The sensors connected to the access n/w may also be wired.

↳ Gateways & backhaul n/w Sublayer – A common Comm. System organizes multiple smart objects in a given area around a common gateway. The gateway communicate directly with the smart objects. The role of the gateway is to forward the collected info through a

longer-range medium (called the backhaul) to a headend central station where the information is processed

↳ Nw transport Sublayer - for comm to be successful, nw & transport layer protocols such as IP & VPP must be implemented to support the variety of devices to connect and media to use.

↳ IoT nw mgmt Sublayer - Additional protocols must be in place to allow the headend applns to exchange data with the sensors. Examples include CoAP & MQTT.

↳ Application & analytics layer - At the upper layer, an application needs to process the collected data, not only to control the smart objects when necessary, but to make intelligent decision based on info collected & in turn instruct the "things" or other systems to adapt to the analyzed conditions & change their behaviors & parameters.

Layer 1 : Things : Sensors & Actuator layer

There are myriad ways to classify smart objects. one architectural classification could be :

↳ Battery powered or power connected - This classification is based on whether the object carries its own energy supply or receives continuous power from an external power source. Battery powered things can be moved more easily than line-powered objects. However batteries limit the lifetime and amount of energy that the object is allowed to consume

↳ mobile or static - This classification is based on whether the "thing" should move or always stay at the same location. The range of mobility (from few inches to miles away) often drives the possible power source.

↳ Low or high reporting frequency - This classification is based on how often the object should report monitored parameters. A mult sensor may report values once a month. A motion sensor may report acceleration several hundred times per second. Higher frequencies drive higher energy consumption, which may create constraints on the possible power source & the tx°. range.

↳ Simple or rich data - This classification is based on the quantity of data exchanged at each report cycle. A humidity sensor in a field may report a simple daily index value (on a binary scale from 0 to 255), while an engine sensor may report hundreds of parameters, from temperature to pressure, gas, velocity, compression speed, carbon index & many others. Richer data typically drives higher power consumption.

↳ Report range - This classification is based on the distance at which the gateway is located. For ex, for your fitness band to communicate with your phone, it needs to be located a few meters away at most. The assumption is that your phone needs to be at visual distance for you to consult the reported data on the phone screen.

↳ Object density per cell - This classification is based on the no. of smart objects over a given area, connected to the same gateway.

Layer2 : Communications NW Layer.

Access NW Sublayer - one key parameter determining the choice of access technology is the range between the Smart Object & the Information Collector. Common groups are as follows:

↳ PAN (Personal Area NW) - Scale of few meters. This is the Personal Space around a person. A common wireless technology for this scale is Bluetooth.

↳ HAN (Home Area NW) - Scale of a few tens of meters. At this scale, Common Wireless technologies for IoT include ZigBee & Bluetooth Low Energy (BLE).

↳ NAN (Neighborhood area NW) - Scale of a few hundreds of meters. The term NAN is often used to refer to a group of house units from which data collected.

↳ LAN (local area NW) Scale of upto 100m.

Gateways & Backhaul Sublayer Data Collected from a Smart object may need to be forwarded to a central Station where data is processed. As this Station is often in a different location from the Smart object, data Directly received from the Sensor through an access technology needs to be forwarded to another medium (the backhaul) & transported to the central station. The gateway is in charge of this inter-medium Comm.

NW Transport Sublayer - The transport layer protocols built over IP (UDP and TCP) can easily be leveraged to decide whether the NW should Control the data packet delivery (with TCP) or whether the Control task should be left to the application (UDP). UDP is much lighter & faster than TCP.

IoT Nw mgmt Sublayer – IP, TCP and UDP bring connectivity to IoT nws. Upper-layer protocols need to take care of data transmission between the Smart objects & other systems. multiple protocols have been leveraged or created to solve IoT Data Communication problems. Some alws rely on push model (i.e Sensor reports at a regular interval or based on a local trigger), whereas others rely on a pull model (i.e an appln queries the Sensor over the nw) & multiple hybrid approaches are also possible.

Eg – CoAP & MQTT protocols

Layer 3 : Applications & Analytics Layer once connected to a nw, your Smart objects exchange info with other systems.

Analytics versus Control Applications

Analytics application – This type of application collects data from multiple Smart objects, processes the collected data, & displays info resulting from the data that was processed.

Control application – This type of application controls the behavior of the Smart object or the behavior of an object related to the Smart object. for eg a pressure Sensor may be connected to a pump. A control application increases the pump speed when the connected Sensor detects a drop in pressure.

Data versus Nw Analytics

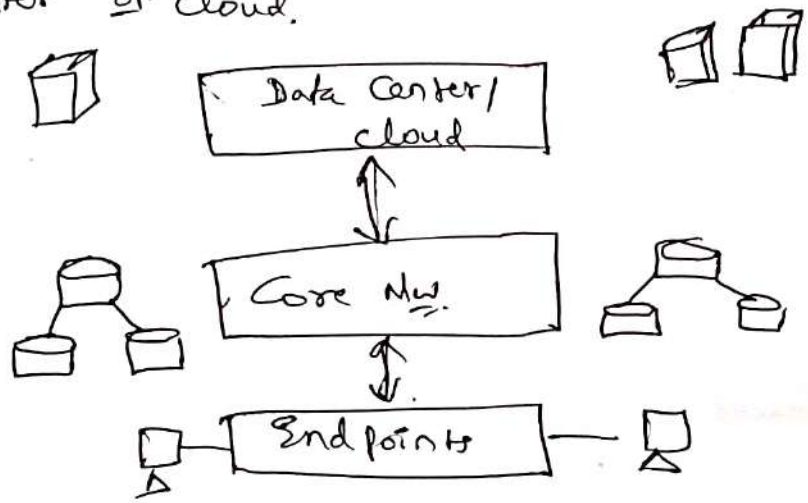
Analytics is a general term that describes processing info to make sense of collected data.

Data Analytics – This type of analytics processes the data collected by Smart objects & combines it to provide an intelligent view related to the IoT System.

Network Analytics - most IoT systems are built around smart objects connected to the nlw. A loss or degradation in connectivity is likely to affect the efficiency of the system. Such a loss can have dramatic effects.

* IoT Data management & Compute Stack

Data management in traditional IT systems is very simple. The endpoints (laptops, printers, IP phones & so on) communicate over an IP core nlw to servers in the data center or cloud.



[The traditional IT cloud computing model]

⇒ IoT systems function differently. Several data-related problems need to be addressed:

↳ B.W in last-mile IoT nlws is very limited.

↳ Latency can be very high.

↳ Nlw backhaul from the gateway can be unreliable and often depends on 3G/LTE or even satellite links.

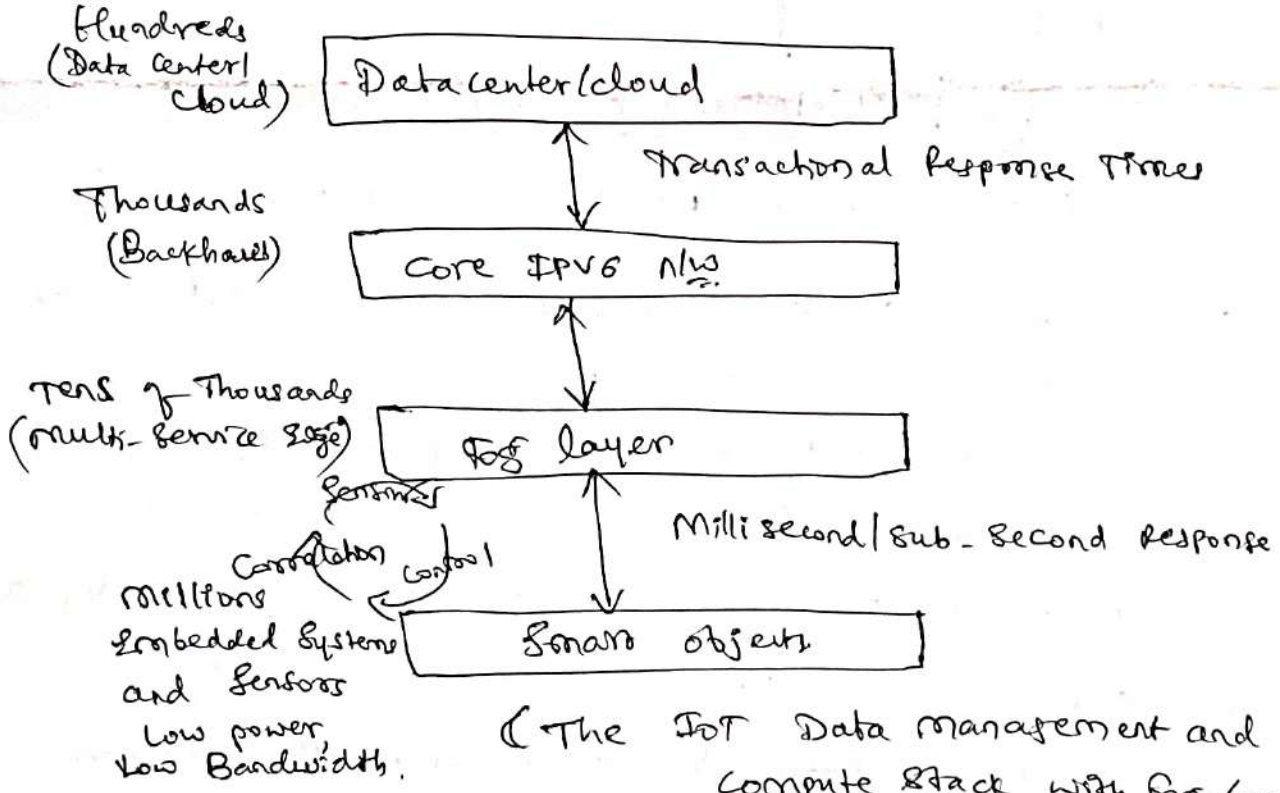
↳ The volume of data transmitted over the backhaul can be high.

↳ Big data is getting bigger.

Fog Computing - The solution to the challenges mentioned on the previous section is to distribute data management

throughout the IoT System, as close to the edge of the IP n/w. as possible. The best-known embodiment of edge services in IoT is fog Computing. Any device with Computing, storage and n/w Connectivity can be a fog node. Examples include industrial controllers, Switches, routers, embedded Servers, IoT gateways. Analyzing IoT data close to where it is Collected minimizes latency, offloads gigabytes of n/w traffic from the core n/w & keeps sensitive data inside the local n/w.

⇒ An advantage of this structure is that the fog node allows intelligence gathering (such as analytics) and Control from the closest possible point, & in doing so, it allows better performance over Constrained n/ws.



(The IoT Data management and Compute Stack with fog Computing)

⇒ fog Services are typically accomplished very close to the edge device, sitting as close to the IoT endpoints as possible. One significant advantage of this is that the fog node has contextual awareness of the sensors it is managing because of its geographic proximity to those sensors.

Module-2

* Smart objects: The "Things" in IoT

Sensors - A sensor does exactly as its name indicates: It senses. more specifically, a sensor measures some physical quantity & converts that measurement reading into a digital representation. That digital representation is typically passed to another device for transformation into useful data that can be consumed by intelligent devices or humans.

⟹ There are a no. of ways to group and cluster sensors into different categories, including the following:

↳ Active or passive - Sensors can be categorized based on whether they produce an energy o/p & typically require an external power supply (active) or whether they simply receive energy & typically require no external power supply (passive).

↳ Invasive & non-Invasive - based on whether sensor is part of the environment it is measuring (invasive) or external to it (non-invasive).

↳ Contact or no-contact - based on whether they require physical contact with what they are measuring (contact) or not (no-contact).

↳ Absolute or relative - based on whether they measure on an absolute scale (absolute) or based on a difference with a fixed or variable reference value (relative).

↳ Area of application - based on the specific industry or vertical where they are being used.

↳ How sensors measure - based on the physical mechanism used to measure sensory i/p (for ex thermoelectric, electromechanical, piezo-resistive, optic, electric, ~~fluid~~ fluid mechanic, photoelastic.

↳ What sensors measure - based on their applications or what physical variables they measure.

Note this is by no means an exhaustive list, & there are many other classification & taxonomic schemes for sensors, including those based on material, cost, design & other factors.

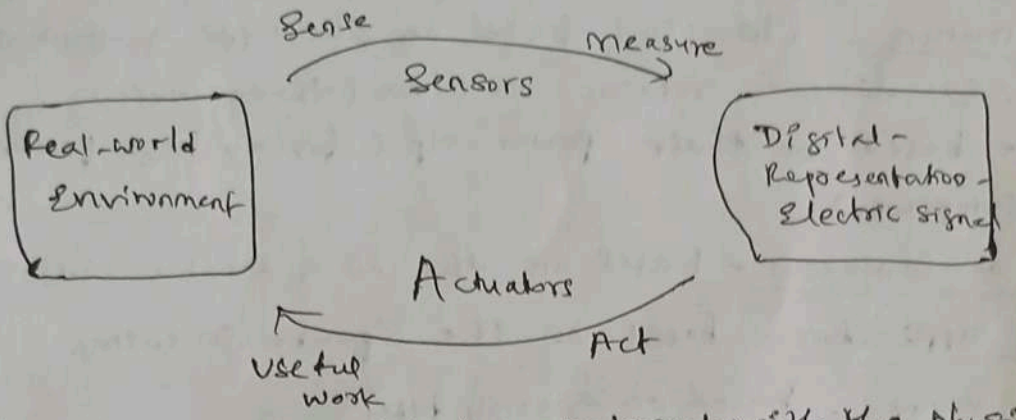
Sensor Type	Description	Examples
Position	A position sensor measures the position of an object; the position measurement can be either in absolute terms (absolute position sensor) or in relative terms (displacement sensor). Position sensors can be linear, angular or multi-axis.	Potentiometer, inclinometer, proximity sensor
Occupancy & motion	Occupancy sensors detect the presence of people & animals in a surveillance area, while motion sensors detect movement of people & objects. The occupancy sensors generate a signal even when a person is stationary, whereas motion sensors do not.	Electric Eye, radar
Velocity & acceleration	Velocity (speed of motion) sensors may be linear or angular, indicating how fast an object moves along a straight line or how fast it rotates. Acceleration sensors measures changes in velocity.	Accelerometer, gyroscope.
Force	Force sensors detects whether a physical force is applied & whether the magnitude of force is beyond threshold.	Force gauge, viscometer, tactile sensor (touch sensors)

Pressure	Pressure sensors are related to force sensors, measuring force applied by liquids or gases. Pressure is measured in terms of force per unit area.	Barometer, Bourdon gauge, Piezometer
Flow	Flow sensors detect the rate of fluid flow. They measure the volume (mass flow) or rate (flow velocity) of fluid that has passed through a system in a given period of time.	Anemometer, mass flow sensors, Water meter
Acoustic	Acoustic sensors measure sound levels & convert that information into digital or analog data signals	microphone, geophone, hydrophone
Humidity	Detect humidity (amount of water vapour) in the air or a mass. Humidity levels can be measured in various ways: absolute humidity, relative humidity, mass ratio & so on.	Hygrometer, humistor, Soil moisture sensor
Light	Light sensors detect the presence of light (visible or invisible)	Infrared sensor, photo detector, Flame detector
Radiation	Detect radiation in the environment. Radiation can be sensed by scintillating or ionization detection.	Geiger-Muller counter, Scintillator, neutron detector

Temperature	measure the amount of heat or cold that is present in the system. They can be broadly of two types: Contact and non-Contact. Contact temperature sensors need to be in physical Contact with the object being sensed. Non-contact sensors do not need physical Contact, as they measure temperature through convection & radiation.	Thermometer, Calorimeter, temperature gauge
Chemical	measure the concentration of chemicals in a system. When subjected to a mix of chemicals, chemical sensors are typically selective for a target type of chemical (for ex a CO_2 sensor senses only Carbon dioxide)	Breathalyzer, olfactometer, smoke detector
Biosensors	Biosensors detect various biological elements, such as organisms, tissues, cells, enzymes, antibodies & nucleic acid	Blood glucose, biosensor, pulse oximetry, electrocardiograph.

[Sensor Types]

* Actuators - are natural complement to sensors. following figure demonstrates the symmetry & complementary nature of these two types of devices



[How Sensors & Actuators interact with the physical world]

→ Sensors are designed to sense & measure practically any measurable variable in the physical world. They convert their measurements (typically analog) into electric signals or digital representation that can be consumed by an intelligent agent (a device or a human).

Actuators on the other hand, receive some type of control signal (commonly an electric signal or digital command) that triggers a physical effect, usually some type of motion, force, & so on.

→ Humans use their five senses to sense & measure their environment. The sensory organs converts this sensory info into electrical impulses that the nervous system sends to the brain for processing. Likewise for sensors sense & measure the physical world & signal their measurements as electric signals sent to some type of microprocessor or microcontroller for additional processing. The human brain signals motor fn & movement & the nervous system carries that info to the appropriate part of the muscular system. Correspondingly a processor can send an electric signal to an actuator that translates signal into some type of ~~processing~~ movement. This interaction between sensors, actuators & processors & the similar functionality in biological systems is the basis for various technical fields, including robotics & biometrics.

Much like sensors, actuators also vary greatly in function, size, design & so on. Some common ways that they can be classified including the following:

↳ Type of motion - classified based on the type of motion they produce (ex linear, rotary, one/two (three-axis)

↳ power - based on their power o/p (for ex high power, low power, micropower)

↳ Binary & Continuous - based on the no. of stable-state o/ps.

↳ Area of application - based on the specific industry.

↳ Type of energy - based on energy types.

Type	Examples
* mechanical Actuators	Lever, screw jack, hand crank
* Electrical actuators	Thyristor, bipolar transistor, diode
* Electromechanical actuators	Ac motor, Dc motor, Step motor
* Electromagnetic actuator	Electromagnet, linear Solenoid
* Hydraulic & Pneumatic actuator	Hydraullic Cylinder, pneumatic Cylinder, piston, pressure Control valves, air motors
* Smart materials actuators	shape memory alloy (SMA), ion exchange fluid, magnetoresistive material, bimetallic strip, piezoelectric bemorph
* micro & nano actuators	electrostatic motor, microvalve, comb drive

→ whereas sensors provide the info, actuators provide the action. The most interesting use cases for IoT are those where sensors & actuators work together in an intelligent, strategic & complementary fashion.

Smart Objects

→ Smart objects are, quite simply the building blocks of IoT. They are what transform everyday objects into a new generation of intelligent objects that are able to learn from & interact with their environment in a meaningful way.

Smart objects : A Definition The term Smart object, despite some semantic differences, is often used interchangeably with terms such as Smart Sensor, Smart device, IoT device, thing, Smart thing, intelligent device, intelligent thing, ubiquitous thing & intelligent product.

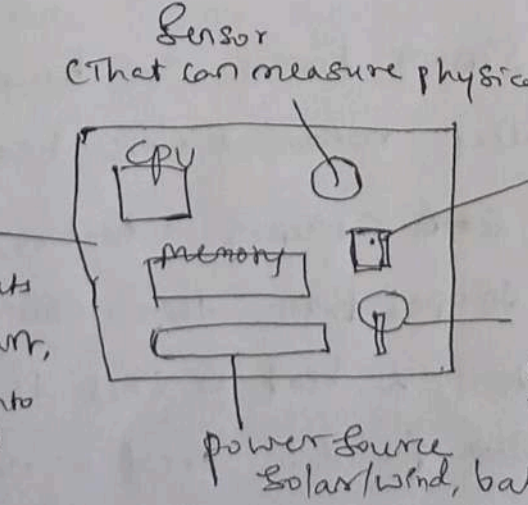
⇒ A Smart object, is a device that has, at a minimum, the following four defining characteristics.

↳ **Processing unit** – A smart object has some type of processing unit for acquiring data, processing & analyzing sensing info. received by the sensor(s), co-ordinating control signals to any actuators, & controlling a variety of functions on the Smart Object, including the Comm & power systems.

↳ **Sensor(s) and/or actuator** – A Smart object is capable of interacting with the physical world through sensors and actuators.

↳ **Communication device** – The Comm unit is responsible for connecting a Smart object with other Smart objects & the outside world (via the n/w). Comm devices for Smart objects can be wired or wireless.

↳ **power source** – Smart objects have components that need to be powered.



[Characteristics of Smart objects]

Trends in Smart objects - Following are the trends impacting IoT:

↳ Size is decreasing - There is clear trend of ever decreasing size.

↳ Power Consumption is decreasing - The diff. hlw components of a Smart object continually consume less power.

↳ Processing power is increasing - Processors are continually getting more powerful & smaller.

↳ Comm. Capabilities are improving - Wireless speeds are continually increasing but they are also increasing in range.

↳ Comm. is being increasingly standardized - There is a strong push in the industry to develop open standards for IoT comm. protocols.

Sensor Nlws

→ A Sensor/Actuator nlw (SANET), as the name suggests, is a nlw of sensors that sense & measure their environment and/or actuators that act on their environment. The sensors & actuators in a SANET are capable of communicating and cooperating in a productive manner.

→ SANETs offer highly co-ordinated sensing & actuation capabilities. Smart homes are a type of SANET that display this coordination between distributed sensors & actuators.

For ex Smart homes can have temperature sensors that are strategically networked with heating, ventilation, & air conditioning ~~(HVAC)~~ (HVAC) actuators. When a sensor detects a specified temperature, this can trigger an actuator to take action & heat or cool the home as needed. SANETs are typically found in "real world" means that they need an extreme level of deployment flexibility.

The following are some advantages & disadvantages that a wireless-based solution offers:

* Advantages
  - ↳ Greater deployment flexibility
  - ↳ Simpler scaling to a large no. of nodes
  - ↳ Lower implementation costs
  - ↳ Easier long-term maintainance
  - ↳ Effortless introduction of new sensor/Actuator nodes
  - ↳ Better equipped to handle dynamic/rapid topology changes

* Disadvantages
  - ↳ Potentially less secure (ex hijacked APs)
  - ↳ Typically lower tx^{n} speeds
  - ↳ Greater level of impact/influence by environment

Wireless Sensor Nw^{s} (WSNs)

- are made up of wirelessly connected smart objects, which are sometimes referred to as motes. The fact that there is no infrastructure to consider with WSN's is surely a powerful advantage for flexible deployments, but there are variety of design constraints to consider with these wirelessly connected smart objects.

→ The following are some of the most significant limitations of the smart objects in WSNs:
  - ↳ Limited processing power (minimize energy use)
  - ↳ Limited memory (few tens of KBs)
  - ↳ Lossy comm^{ns}
  - ↳ Limited transmission speeds
  - ↳ Limited power

wirelessly connected smart objects generally have one of the following two communication patterns:

↳ Event-driven - Type of sensory info is triggered only when a smart objects detects a particular event or predetermined threshold.

↳ periodic - type of sensory info occurs only at periodic intervals.

* Connecting Smart objects - The following section covers technologies for connecting smart objects.

↳ IEEE 802.15.4 - an older but foundational wireless protocol for connecting smart objects.

↳ IEEE 802.15.4g (2047 bytes) & IEEE 802.15.4e (127 bytes) - improvements to 802.15.4 that are targeted to utilities & smart cities deployments.

↳ IEEE 1901.2a - which is a technology for connecting smart objects over power lines.

↳ IEEE 802.11ah - a technology built on the well known 802.11 wi-fi standards that is specifically for smart objects.

↳ LoRaWAN - a scalable technology designed for longer distances with low power requirements in the unlicensed spectrum.

↳ NB-IoT and other LTE variations - which are often the choice of mobile service providers looking to connect smart objects over distances in the licensed spectrum.

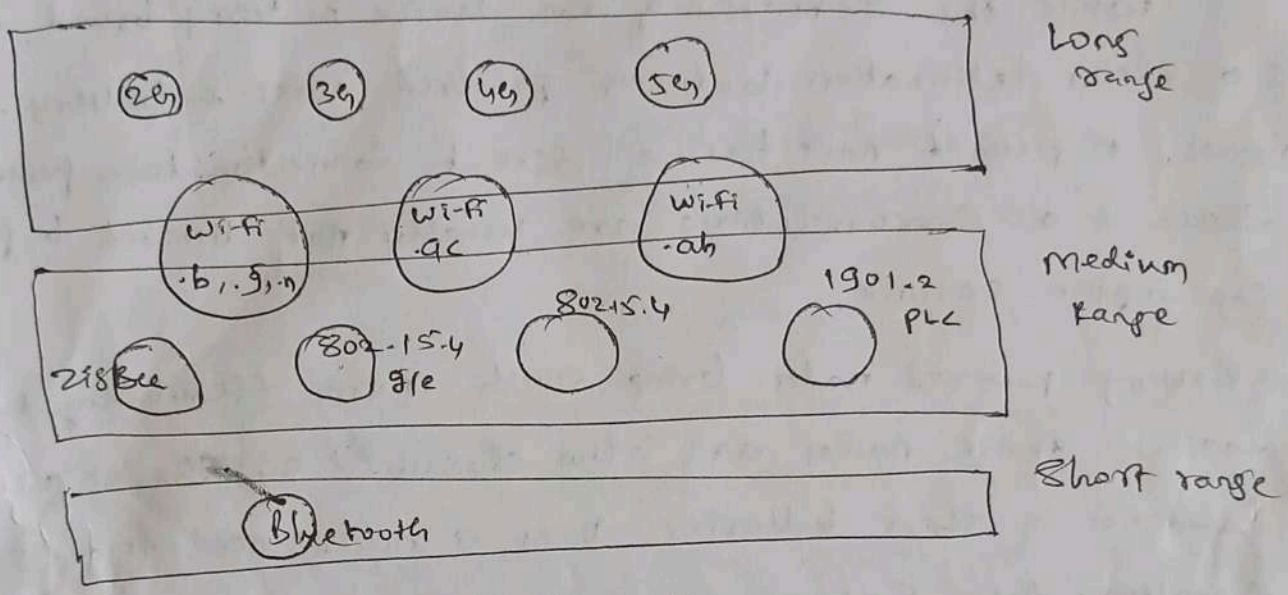
1901.2a - addresses, grid automation, electric vehicle to charging station etc.

Licensed Spectrum - used for tightly controlled activities

Unlicensed Spectrum - free & open to the public

* Communications criteria

Range - How far does the signal need to be propogated? i.e what will be the area of coverage for a selected wireless technology? Should indoor versus outdoor deployments be differentiated? very often, these are the first questions asked when discussing wired & wireless access technologies.



(Wireless access technologies)

Short range - The classical wired example is a serial cable. wireless short range technologies are often considered as an alternative to a serial cable, supporting tens of meters of maximum distance between two devices. Examples of short-range wireless technologies are IEEE 802.15.1 Bluetooth and IEEE 802.15.7 visible light Communications (VLC).

Medium range - In the range of tens to hundred of meters, many specifications & implementations are available. The maximum distance is generally less than 1 mile between 2 devices, although RF technologies do not have the real maximum distances defined, as long as the radio signal is detected & received in the scope of the applicable specification. Examples include IEEE 802.11 Wi-Fi, IEEE 802.15.4 & 802.15.4g WPAN. wired technologies such as IEEE 802.3 Ethernet & IEEE 1901.2

narrow band power line comm$.

Long range - Distances greater than 1 mile between two devices require long-range technologies, wireless examples are Cellular (2G, 3G, 4G) & some applications of outdoor IEEE 802.11 Wi-Fi & Low-power wide-Area (LPWA) technologies.

Power Consumption

While the definition of IoT device is very broad, there is a clear delineation between powered nodes & battery-powered nodes. A powered node has a direct connection to a power source & a communications are usually not limited by power consumption criteria.

→ Battery powered nodes bring much more flexibility to IoT devices. These nodes are often classified by the required lifetimes of their batteries. Does a node need 10 to 15 years of battery life, such as on water or gas meters? Or is a 5-7 year battery life sufficient for a device such as smart parking sensors? For devices under regular maintainance, a battery life of 2 to 3 years is an option.

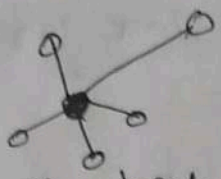
→ IoT wireless access technologies must address the needs of low power consumption & connectivity for battery-powered nodes

Topology - Among the access technologies available for connecting IoT devices, three main topology schemes are dominant: star, mesh & peer-to-peer. For long-range & short-range technologies, a star topology is prevalent, as seen with cellular, LPWA & Bluetooth nlws

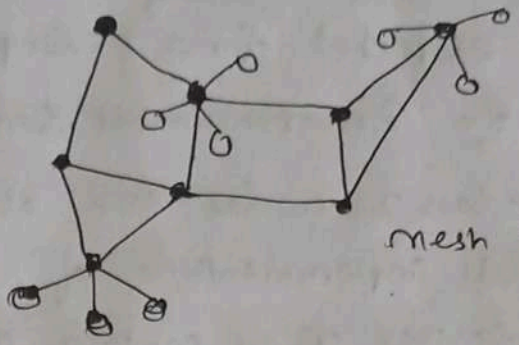
→ Star topologies utilize a single central base station or controller to allow communications with end-points.

→ peer-to-peer technologies allow any device to communicate with any other device as long as they are in the range of

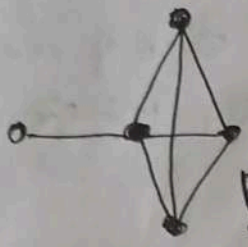
each other. peer-to-peer topologies enable more complex formations, such as mesh networking topology.



star topology



mesh topology



peer to peer topology

● - Full function device
○ - Reduced function device

(Star, peer to peer, & mesh topologies)

→ For ex, indoor Wi-Fi deployments are mostly a set of nodes forming a star topology around their APs. meanwhile outdoor Wi-Fi may consists of mesh topology for the backbone of APs with nodes connecting to the APs in a star topology.

Constrained devices - According to RFC 7228, constrained nodes can be broken down into the classes defined as below:

class	Definition
class 0	This class of nodes is severely constrained, with less than 10KB of memory & less than 100KB of flash processing & storage capabilities. These nodes are typically battery-powered. They do not have resources required to directly implement an IP stack & associated security mechanisms. ~~The~~ of class 0 node is a push button that sends (byte of info when changing its status. This class is particularly well suited to leveraging new unlicensed LPWA wireless technology.

class 1 while greater than class 0, the processing & code space characteristics (approx 10KB RAM & 100KB flash) of class 1 are still lower than expected for a complete IP stack implementation.
Eg: Environmental Sensors

class 2 class 2 nodes are characterized by running full implementations of an IP stack on embedded devices. They contain more than 50KB of memory & 250KB flash, so they can be fully integrated to IP n/ws.
Eg: Smart ~~meter~~ power meter.

Flash memory (ROM) - low cost, high density, non-volatile computer storage chip that can be electrically erased & reprogrammed

Frequency bands - Radio spectrum is regulated by countries and/or organizations, such as International Telecomm Union (ITU) and the Federal Communications Commission (FCC). These groups define the regulations & tx requirements for various frequency bands. For ex, portions of the spectrum are allocated to types of telecommunications such as radio, television, military & so on.

⇒ The ITU has also defined unlicensed spectrum for the industrial, scientific & medical (ISM) portions of the radio bands.

For IoT access, these are the most well known ISM bands:

↳ 2.4 GHz band as used by IEEE 802.11b/g/n Wi-Fi
↳ IEEE 802.15.1 Bluetooth
↳ IEEE 802.15.4 WPAN

Constrained node nws - Layer 1 & Layer 2 protocols that can be used for Constrained-node nws must be evaluated in the context of the following characteristics for use-case applicability:

- ↳ data rate & throughput
- ↳ latency & determinism
- ↳ overhead & payload

* IoT Access Technologies Following topics are addressed for each IoT access technology:

- ↳ Standardization & alliances - The standard bodies that maintain the protocols for a technology
- ↳ Physical layer - The wired or wireless methods and relevant frequencies
- ↳ MAC layer - Considerations at the Media Access Layer (MAC) layer, which bridges the physical layer with data link control
- ↳ Topology - The topologies supported by the technology
- ↳ Security - Security aspects of the technology
- ↳ Competetive technologies - other technologies that are similar & may be suitable alternatives to the given technology

* IEEE 802.15.4 - is a wireless access technology for low-cost & low data-rate devices that are powered or run on batteries. IEEE 802.15.4 is commonly found in the following types of deployments:

- ↳ Home & building automation
- ↳ Automotive nws
- ↳ Industrial wireless sensor nws
- ↳ Interactive toys & remote controls

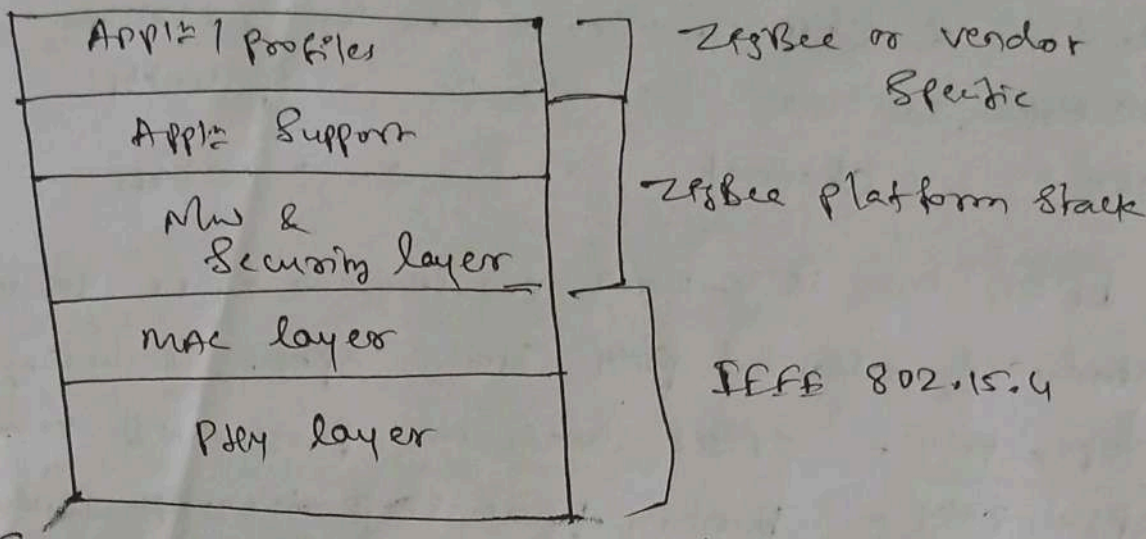
Standardization And Alliances – Some of the most well-known protocol stacks based on 802.15.4 are listed below:

Protocol	Description
ZigBee	ZigBee defines upper layer components as well as appln profiles. Common profiles includes building automation, home automation & healthcare. ZigBee also defines device object functions such as device role, device discovery, NW form & security
6LoWPAN	is an IPv6 adaptation layer defined by the IETF 6LoWPAN working group that describes how to transport IPv6 packets over IEEE 802.15.4 layers.
ZigBee IP	adopts 6LoWPAN adaptation layer, IPv6 NW layer & RPL routing protocol.
ISA100.11a	is developed by the International Society of Automation (ISA) as "Wireless Systems for Industrial Automation: process Controlled and Released applications."
WirelessHART	is a protocol stack that offers a time-Synchronized, self-organizing & self-healing mesh architecture over 2.4 GHz frequency band.
Thread	Thread is a protocol stack for secure and reliable mesh NW to connect & control products in the home

(Protocol stacks utilizing IEEE 802.15.4)

RPL – Routing protocol for Low power & lossy NWs

ZigBee The traditional ZigBee stack is illustrated as below:

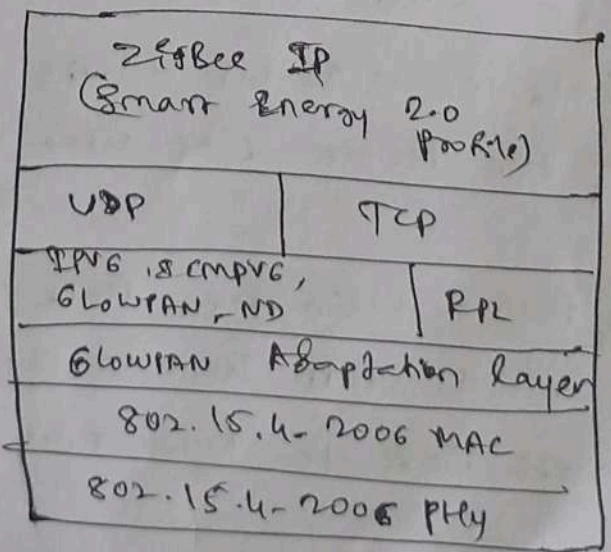


(High-level ZigBee protocol stack)

⇒ The ZigBee nlw & Security layer provides mechanisms for Nw Startup, Configuration, routing & Security Comms. This includes Calculating routing paths in what is often a changing topology, discovering neighbors & managing routing tables.

ZigBee utilizes the 802.15.4 for Security at the MAC layer, using AES with 128-bit-key & also provides Security at the nlw & appln layers.

ZigBee IP



ND-Neighbor Discovery

(ZigBee IP protocol stack)

Physical layer - The original physical layer two options were as follows:

↳ 2.4 GHz, 16 channels with a data rate of 250 Kbps

↳ 915 MHz, 10 " " " " of 40 Kbps

↳ 868 " , 1 channel " " " 20 Kbps

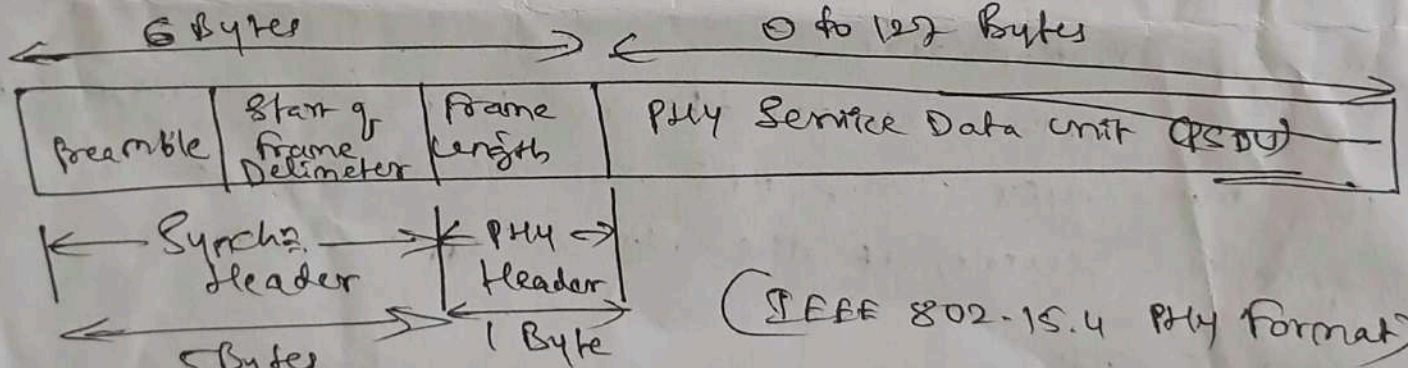
⇒ IEEE 802.15.4-2006, 802.15.4-2011, IEEE 802.15.4-2015 introduced additional PHY Comm options, including:

OQPSK PHY - Offset quadrature phase-shift keying

BPSK PHY - Binary phase shift keying modulation. (Specifies two unique phase shifts as its data encoding scheme)

ASK PHY - Amplitude shift keying

⇒ Below figure shows the frame for the 802.15.4 physical layer.



(IEEE 802.15.4 PHY Format)

→ **Preamble** field is a 32-bit 4 byte pattern that identifies the start of the frame (is used to synchronize the data rate.

→ The **start of frame delimeter** field informs the receiver that frame contents start immediately after this byte.

→ **Frame length** indicate how much total data to expect in PSDU.

→ The PSDU is the data field or payload. The max size of PSDU is 127 bytes.

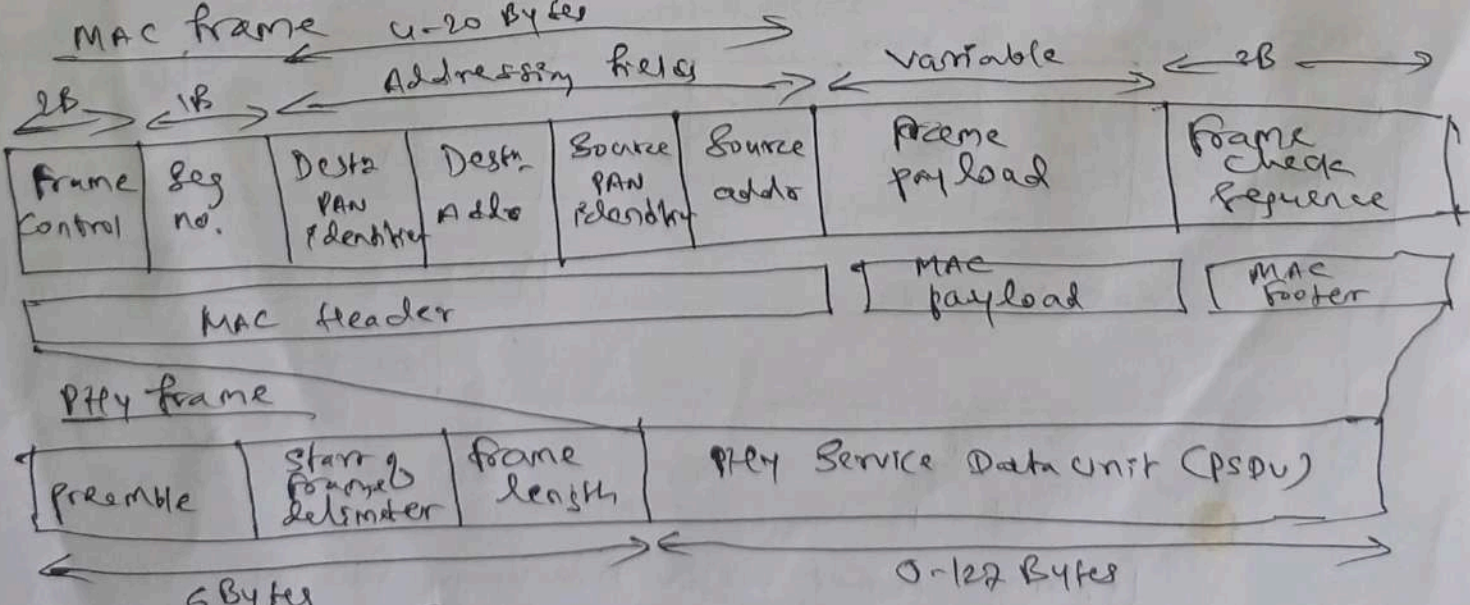
MAC layer - manages access to the PHY channel by defining how devices in the same area will share the frequencies allocated. At this layer, the scheduling and routing of data frames are also co-ordinated. The 802.15.4 MAC layer performs the following tasks:

- ↳ NW beaconing for devices acting as co-ordinators
- ↳ PAN association & disassociation by a device.
- ↳ Device Security
- ↳ Reliable link comms betn two peer MAC entities.

The MAC layer achieves these tasks by using various predefined frame types. In fact, four types of MAC frames are specified in 802.15.4:

- ↳ Data frames Handles all transfers of data
- ↳ Beacon frame: used in txn of beacons from a PAN Coordinator
- ↳ Acknowledgment frame - Confirms the successful reception of a frame
- ↳ MAC Command frame - responsible for control comms between devices

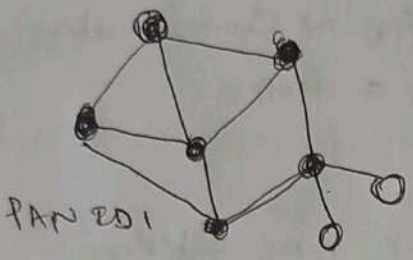
Each of these four MAC frame types follows the frame format shown in figure



(IEEE 802.15.4. MAC format)

Licensed

Topology – IEEE 802.15.4 - based nlws can be built as star, peer-to-peer, or mesh topologies. Mesh nlws tie together many nodes. This allows nodes that would be out of range if trying to communicate directly to leverage intermediate nodes to transfer communications.



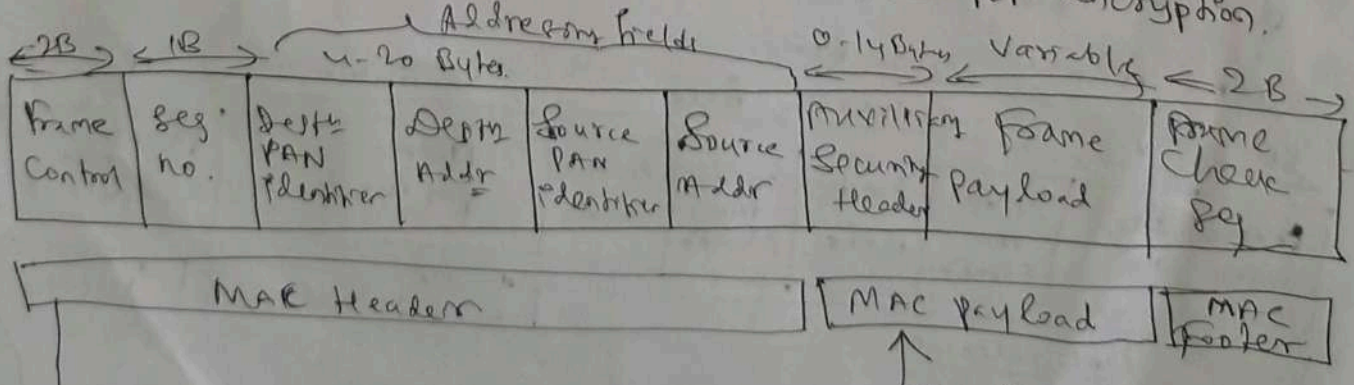
[mesh nlw topology]

→ Every 802.15.4 PAN should be set up with a unique ID. All the nodes in the same 802.15.4 nlws should use the same PAN ID.

Security –

→ IEEE 802.15.4 specification uses AES with its 128-bit key length as the base encryption algorithm for securing its data. AES is a block cipher, which means it operates on fixed-size blocks of data.

→ In addition to encrypting data, AES in 802.15.4 also validates the data that is sent. This is accomplished by a message integrity code (MIC) which is calculated for the entire frame using the same AES key i.e used for encryption.



[Frame format with Auxiliary Security Header field for 802.15.4-2006 & later versions]

Competetive technologies - A Competetive radio technology that is different in its PHY & MAC layers is DASH7. DASH7 was originally based on ISO18000-7 standard and positioned for industrial Comms, whereas IEEE 802.15.4 is more generic. Commonly employed in active radio frequency identification (RFID) implementations, DASH7 was used by US military forces for many years for logistic purposes.

Active RFID utilizes radio waves generated by a battery powered-tag on an object to enable continuous tracking.

→ The current DASH7 tech. offers low-power consumption, a compact protocol stack, range upto 1 mile & AES encryption. Frequencies of 433MHz, 868MHz & 915MHz have been defined enabling data rates upto 166.667 kbps & a max payload of 256 bytes.

* IEEE 802.15.4g & 802.15.4e -

→ The IEEE 802.15.4e amendment of 802.15.4-2011 expands the MAC layer feature set to remedy the disadvantages associated with 802.15.4, including MAC reliability, unbounded latency & multipath fading.

→ The focus of IEEE 802.15.4g is the Smart grid, or more specifically Smart utility Comms. 802.15.4g seeks to optimize large outdoor wireless mesh nws for field area nws. New PHY defns are introduced as well as some MAC modifications needed to support their implementation. This technology applies to IoT use cases such as the following:

↳ Distribution Automation & industrial supervisory

control & data acquisition (SCADA) environments for remote monitoring & control

↳ public lighting
↳ Environmental wireless sensors in smart cities
↳ Electrical vehicle charging stations
↳ Smart parking meters
↳ Micro grids
↳ Renewable energy

Standardization & Alliances -

Commercial Name/ Trademark	Industry org^n	Standard body
Wi-Fi	Wi-Fi alliance	IEEE 802.11 Wireless LAN
WiMAX	WiMAX forum	" 802.16 " MAN
Wi-SUN	Wi-SUN Alliance	" 802.15.4g " SUN

[Industry Alliances for some Common IEEE Standards]

Physical layer

→ maximum PSDU or payload size of 127 bytes was increased for the SUN PHY to 2047 bytes.
→ fragmentation is no longer necessary at layer 2 when IPv6 packets are tunneled over IEEE 802.15.4g MAC frames. Also error correction/protection was improved by evolving the CRC from 16 to 32 bit.
→ Data must be modulated onto the frequency using atleast one of the following PHY mechanisms to be IEEE 802.15.4g Compliant:
↳ Multi-Rate & Multi-Regional Frequency Shift Keying

↳ multirate & multi-regional orthogonal frequency division multiplexing

↳ multi-rate & multi-regional offset quadrature phase shift keying

MAC layer - The following are some of the main enhancements to the MAC layer proposed by IEEE 802.15.4e-2012:

Time-slotted channel Hopping (TSCH): channel hopping also known as frequency hopping utilizes different channels for tx/rx at different times. TSCH divides time into fixed time periods or "time slots", which offer guaranteed b.w & predictable latency.

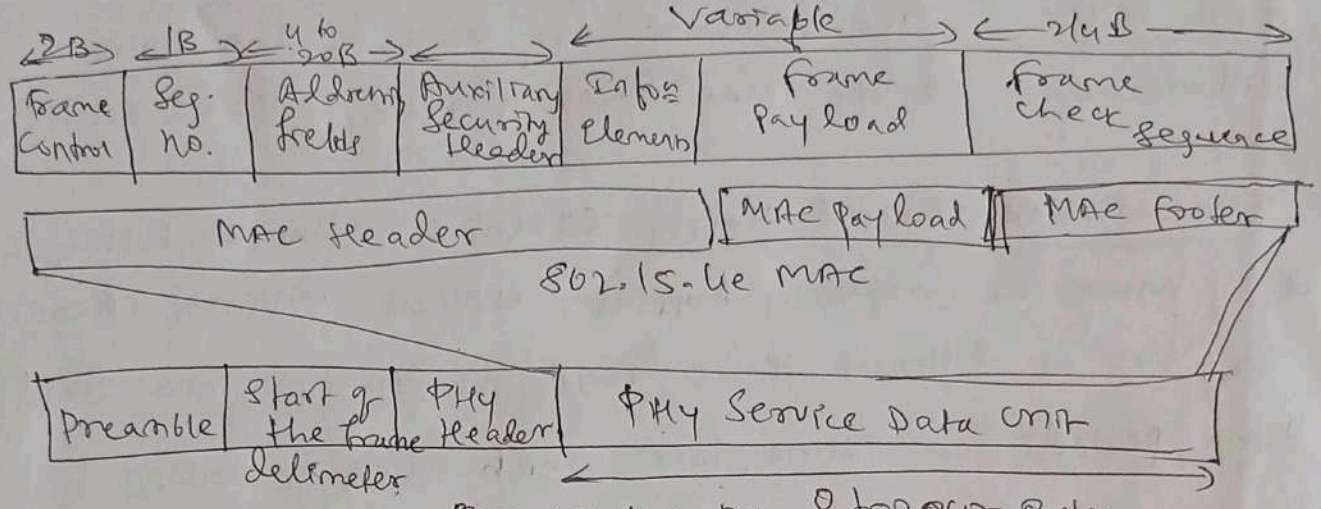
Info elements (IEs): - allow for the exchange of info at the MAC layer in an extensible manner. Specified in a tag, length, value (TLV) format, the IE field allows frames to carry additional metadata to support MAC layer services.

Enhanced beacons (EBs) - extend the flexibility of IEEE 802.15.4 beacons to allow the construction of app-specific beacon content. This is accomplished by including relevant IEs in EB frames.

Enhanced beacon requests (EBRs) - The IEs in EBRs allow the sender to selectively specify the request of info. Beacon responses are then limited to what was requested in the EBR.

Enhanced acknowledgement - The enhanced acknowledgment frame allows for the integration of a frame counter for the frame being acknowledged. This feature helps protect against certain attacks that occur when Acknowledgement frames are spoofed.

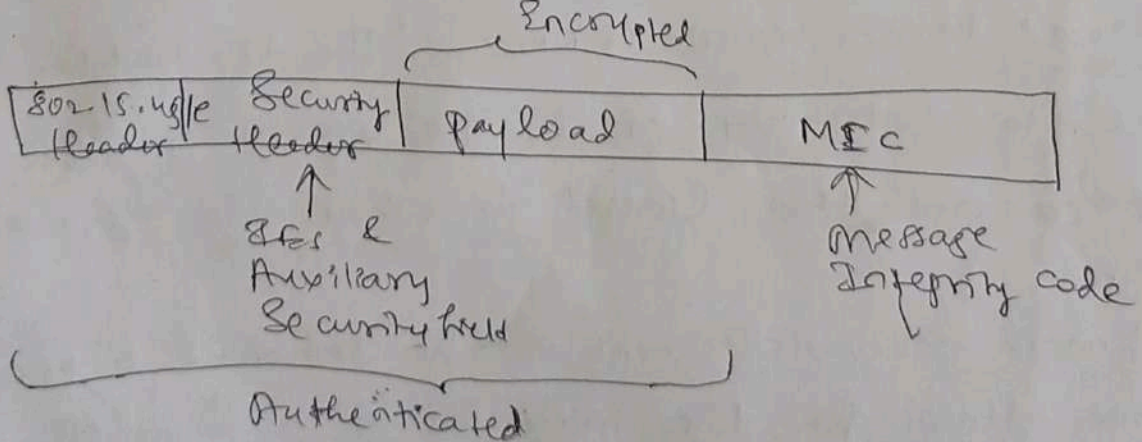
⇒ The 802.15.4g-2012 PHY is similar to the 802.15.4 PHY. The main difference between the two is the payload size, with 802.15.4g supporting upto 2047 bytes & 802.15.4 supporting only 127 bytes.



The 802.15.4e MAC is similar to the 802.15.4 MAC. The main changes shown in the IEEE 802.15.4e header in figure are the presence of the auxiliary Security Header & IEs.

Topology - Deployments of IEEE 802.15.4g are mostly based on a mesh topology. This is because a mesh topology is typically the best choice for use cases in the industrial & smart cities areas

Security - uses AES with 128 bit key.



Competetive Technologies - DASH7.

Module-3

IP as the IoT Network Layer

The Business Case for IP - Data flowing from or to "things" is consumed, controlled & monitored by data center servers either in the cloud or locations ~~is the~~ that may be distributed or centralized. Dedicated applications are then run over virtualized or traditional operating systems on the edge platforms. These lightweight apps. communicate with the data center servers. Therefore, the system solutions combining various physical & data link layers call for an architectural approach with a common layer(s) independent from the lower (connectivity) and/or upper (application) layers. This is how & why the IP suite started playing key architectural role. IP was not preferred in the IT markets but also for the OT environment.

The key advantages of Internet protocol

↳ **open and standards-based** - The IoT creates a new paradigm (example or pattern) in which devices, applications & users can leverage (lift) a large set of devices & functionalities while guaranteeing interchangeability & interoperability, security & management. This calls for implementation, validation and deployment of open, standard-based solutions. The IETF is an open standards body that focuses on the development of the Internet protocol suite & related Internet technologies & protocols.

Versatile - A large spectrum of access technologies is available to offer connectivity of "things" in the last mile. Additional protocols are used to transport the data through backhaul

links and in the data center. Even if physical and data link layers such as Ethernet, Wi-Fi and cellular are widely adopted, the history of data communications demonstrates that no given wired or wireless technology fits all deployment criteria.

The layered IP architecture is well equipped to cope up with any type of physical & data link layers. This makes IP ideal as a long-term investment because various protocols at these layers can be used in a deployment now & over time, without requiring changes to the whole solution arch. & data flow.

Ubiquitous - All recent OS releases from general purpose computers & servers to lightweight embedded systems (TinyOS) have an integrated dual (IPV4 & IPV6) IP stack that gets enhanced over time. IoT application protocols in many industrial or solutions have been updated in recent years to run over IP. In fact, IP is the most pervasive protocol when you look at what is supported across the various IoT solutions & industry verticals.

Scalable - Millions of private & public IP infrastructure nodes have been operational for years. Of course, adding huge no. of "things" to private & public infrastructures may require optimizations & design rules specific to the new devices, IP has proven before that scalability is one of its strengths.

Manageable & highly secure - One of the benefits that comes from 30 years of operational IP NWs is the well-understood NW management & security protocols, mechanisms & toolsets that are widely available. Well-known

n/w & Security management tools are easily leveraged with an IP n/w layer.

Stable & resilient - IP has a large & well established knowledge base & more importantly, it has been used for years in critical infrastructures such as financial & defense n/ws. In addition, IP has deployed for critical services such as voice & video, which have already transitioned from closed environments to open IP standards.

Consumer's market adoption - The main consumer devices range from Smart phones to tablets & pcs. The common protocol that links IoT in the Consumer Space to these devices is IP.

The innovation factor - IP is the underlying protocol for applications ranging from file transfer & e-mail to the WWW, e-Commerce, Social networking, mobility & more. Even the recent Computing evolution from pc to mobile & mainframe to cloud services are perfect demonstrations of these innovative ground enabled by IP. Innovations in IoT can also leverage an IP underpinning.

Adoption or adaptation of the IP

⇒ Adaptation means appl^{n} layered Gateways (ALGs) must be implemented to ensure the translation between non-IP and IP layers.

Adoption involves replacing all non-IP layers with their IP layer counterparts, Simplifying the deployment model and operations.

Eg In the industrial & manufacturing Sector, there has been a move toward IP adoption. Solutions & Product lifecycles in

this space are spread over 10+ years, and many protocols have been developed for serial communications. While IP & Ethernet support were not specified in the initial versions, more recent specifications for these serial comms protocols integrate Ethernet & IPv4.

→ SCADA applications are typical examples of vertical market deployments that operate both the IP adaptation model & the adoption model. Implementations that make use of IP adaptation have SCADA devices attached through serial interfaces to a gateway tunneling or translating the traffic.

With the IP adoption model, SCADA devices are attached via Ethernet to switches & routers forwarding their IPv4 traffic.

⇒ Another example is ZigBee solutions that rely on a non-IP stack between devices and a ZigBee gateway that forwards traffic to an appln. server. A ZigBee gateway often acts as a translator between the ZigBee & IP protocol stacks.

⇒ We should consider the following factors when trying to determine which model is best suited for last-mile connectivity:

↳ Bidirectional versus unidirectional data flow - While bidirectional communications are generally expected, some last-mile technologies offer optimization for unidirectional comms. For ex different classes of IoT devices may only infrequently need to report a few bytes of data to an appln. These sort of devices, particularly ones that communicate

through LPWA technologies, include fire alarms sending alerts or daily test reports, electrical switches being pushed on or off, and water or gas meters sending weekly indexes.

If there is only one-way communication to upload data to an appln, then it is not possible to download new fw or firmware to the devices. This makes integration new features and bug & security fixes more difficult.

↳ overhead for last-mile communication paths – IP adoption implies a layered arch. with a per-packet overhead that varies depending on the IP version. IPv4 has 20 bytes of header at a minimum and IPv6 has 40 bytes at the IP n/w layer. For the IP transport layer UDP has 8 bytes of header overhead, while TCP has minimum of 20 bytes. If the data to be forwarded by a device is infrequent and only a few bytes, you can potentially have more header overhead than a device data. Consequently you need to decide whether IP adoption model is necessary and if it is, how it can be optimized.

↳ Data flow model – In many IoT solutions a device's data flow is limited to one or two applications. In this case the adaptation model can work because the translation of traffic needs to occur only between the end device & one or two appln servers. Depending on the n/w topology and the data flow needed, both IP adaptation & adoption models have roles to play in last-mile connectivity.

↳ N/w diversity – one of the drawbacks of the adaptation model is a general dependency on simplicity and

MAC layers. For 6p, ZigBee devices must only be deployed in ZigBee n/w islands. Therefore deployment must control which applications have to run on the gateway connecting these islands and the rest of the world.

* The need for optimization optimizations are needed at various layers of IP stack to handle restrictions that are present in IoT n/ws.

Constrained nodes - Different classes of devices co-exists. Depending on the functions in a n/w a "things" and may or may not offer similar characteristics compared to general PC or server in an IT environment.

→ Another limit is that this n/w protocol stack may be required to communicate through an unreliable path. This causes the problems such as limited or unpredictable throughput & low convergence when a topology change occurs.

→ Power Consumption is a key characteristics of constrained nodes. Power consumption is much often of a concern on nodes that do not require batteries or an energy source.

(low power, always-on nodes)

→ IoT constrained nodes can be classified as follows:

↳ Devices that are very constrained in resource, may communicate infrequently to transmit a few bytes & may have limited security & management capabilities. This drives the need for the IP adaptation model, where nodes communicate through gateways & proxies.

↳ Devices with enough power & capacities to implement a stripped-down IP stack or non IP-stacks. You may implement either an optimized stack and directly communicate with app's servers (adoption model)

→ Devices that are similar to generic PCs in terms of computing & power resources but have constrained networking capacities, such as b.w: These nodes usually implement a full IP stack (adoption model) but n/w design & application behaviors must cope with the b.w constraints.

Constrained Networks - are often referred to as low-power and lossy n/ws. A constrained n/w can have high latency & a high potential for packet loss.

⇒ Constrained n/ws have unique characteristics & requirements. In contrast, with typical IP n/ws, where highly stable & fast links are available. Constrained n/ws are limited by low-power, low b.w links (wireless & wired). They operate between a few kbps & few hundred kbps & may utilize a star, mesh or combined n/w topologies, ensuring proper operations.

IP versions Techniques such as tunneling & translation need to be employed in IoT solutions to ensure interoperability between IPv4 & IPv6.

→ A variety of factors dictate whether IPv4, IPv6 or both can be used in an IoT solution. The following are some of the main factors applicable to IPv4 and/ IPv6 support in an IoT solution:

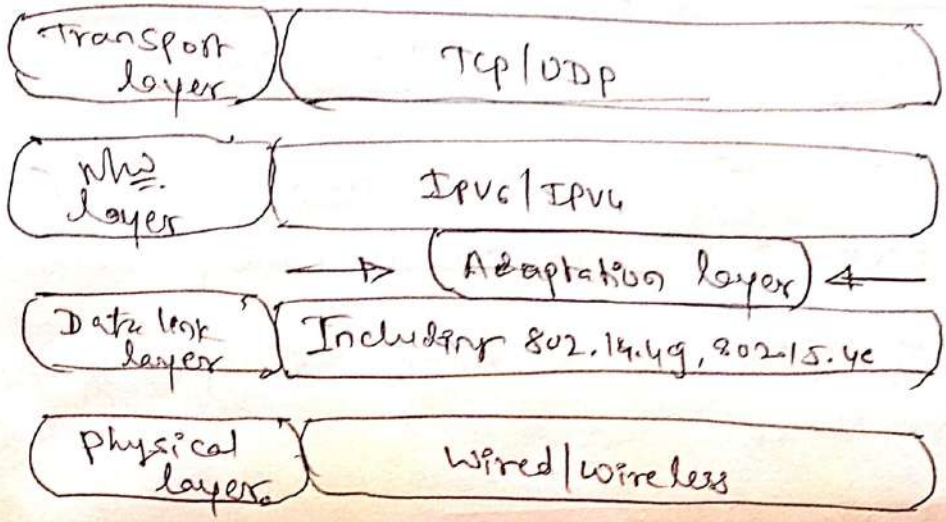
↳ Application protocol: IoT devices implementing Ethernet or Wi-Fi interfaces can communicate over both IPv4 & IPv6, but the appl^{n} protocol may dictate the choice of the IP version.

↳ Cellular provider & Technology - IOT devices with * cellular modems are dependent on the generation of the cellular technology as well as the data services offered by the provider. For the first three generations of data services - GPRS, Edge and 3G - IPV4 is the base protocol version. on 4G/LTE n/ws, data services can use IPV4 or IPV6 as a base protocol, depending on the provider.

↳ Serial Communications - many legacy devices in certain industries, such as manufacturing & utilities, communicate through serial lines. Data is transferred using either proprietary or standard-based protocols such as DNP3, modBus or IEC 60870-5-101. Encapsulation of serial protocols over IP leverages mechanisms such as raw socket TCP or UDP. while raw socket sessions can run over both IPV4 and IPV6, current implementations are mostly available for IPV6 only.

↳ IPV6 Adaptation Layer - IPV6-only adaptation layers for some physical & data link layers for recently standardized IOT protocols support only IPV6. While the most common physical & data-link layers (Ethernet, Wi-Fi & so on) stipulate adaptation layers for both versions, newer technologies such as IEEE 802.15.4, IEEE 1901.2 & ITU G.9903. only have an IPV6 adaptation layer specified. This means that any device implementing a technology that requires an IPV6 adaptation layer must communicate over an IPV6-only subnetwork.

* Optimizing IP for IoT While the Internet protocol is key for a successful IoT, Constrained nodes & Constrained nlws. Mandate optimization at various layers & on multiple protocols of the IP architecture. Below figure highlight the TCP/IP layers where optimization is applied.



(Optimizing IP for IoT using an Adaptation layer)

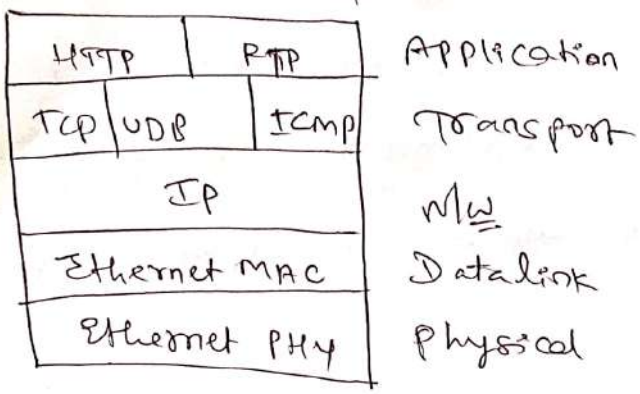
From 6LoWPAN to 6Lo - The model for packaging IP into lower layer protocols is often referred to as an adaptation layer.

An RFC is a publication from the IETF that officially documents Internet standards, specifications, protocols, procedures & events. For ex. RFC 864 describes how an IPV4 packet gets encapsulated over an ethernet frame, & RFC 2464 describes how the same fun. is performed for an IPV6 packet.

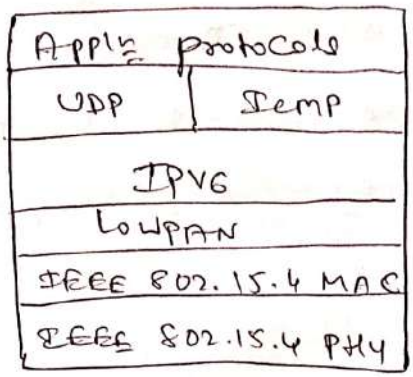
Below figure shows an example of an IoT protocol stack using the 6LoWPAN adaptation layer beside the well-known IP protocol stack for reference. RFC 4944 defines frame headers for the capabilities of header compression, fragmentation,

& mesh addressing.

IP protocol stack

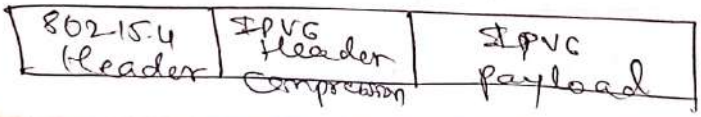


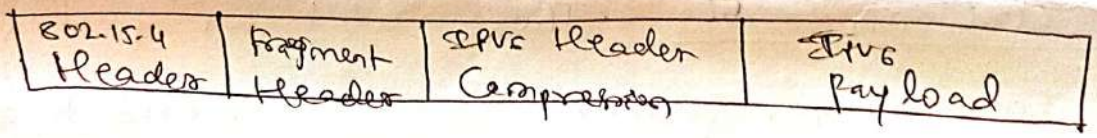
IOT protocol stack with 6LoWPAN Adaptation layer

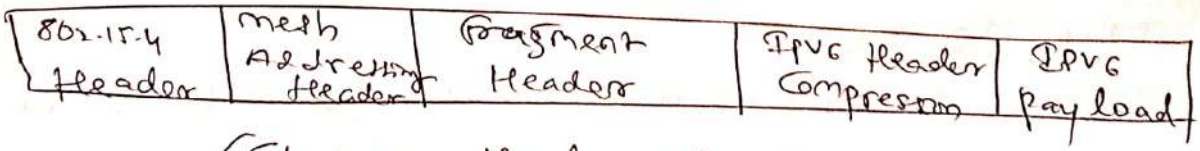


[Comparison of an IoT protocol stack utilizing 6LoWPAN & an IP protocol stack]

Below figure shows some examples of typical 6LoWPAN header stacks.

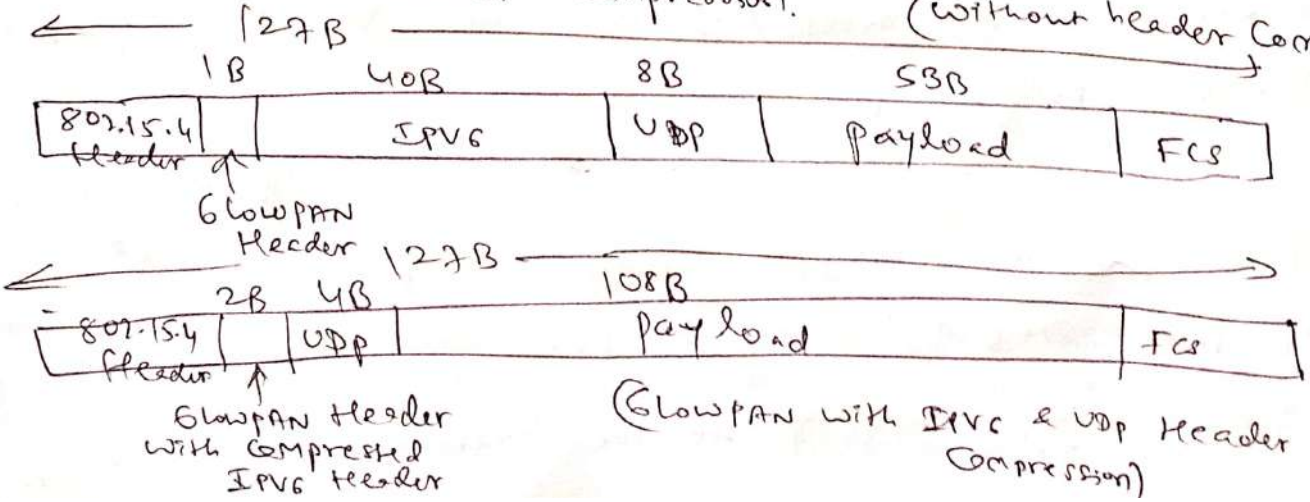






(6LoWPAN Header stacks)

Header Compression Below figure highlight an example that shows the amount of reduction that is possible with 6LoWPAN header compression.

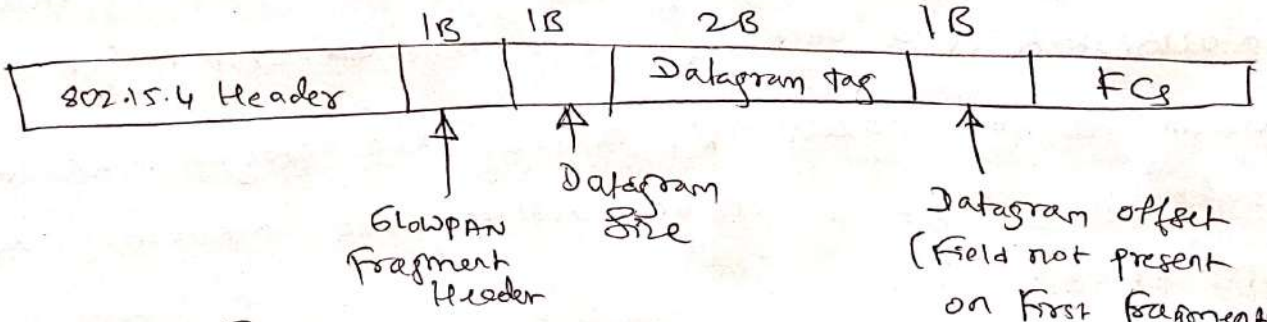


At the top of the figure, a 6LoWPAN frame without any header compression enabled: The full 40-Byte IPv6 header & 8-byte UDP header are visible. The 6LoWPAN header is only a single byte in this case. Notice that uncompressed IPV6 & UDP headers leave only 53 bytes of data payload out of the 127-byte maximum frame size in the case of IEEE 802.15.4.

The bottom figure shows a frame where header Compression has been enabled. The 6LoWPAN header increases to 2 bytes to accommodate the compressed IPv6 header, & UDP has been reduced to half to 4 bytes from 8. The header Compression has allowed the payload to more than double from 53 bytes to 108 bytes.

Fragmentation - The MTU for an IPV6 n/w must be atleast 1280 bytes. The term MTU defines the size of the largest protocol data unit that can be passed.

The fragment header utilized by 6LoWPAN is composed of three primary fields: Datagram Size, Datagram Tag & Datagram offset.



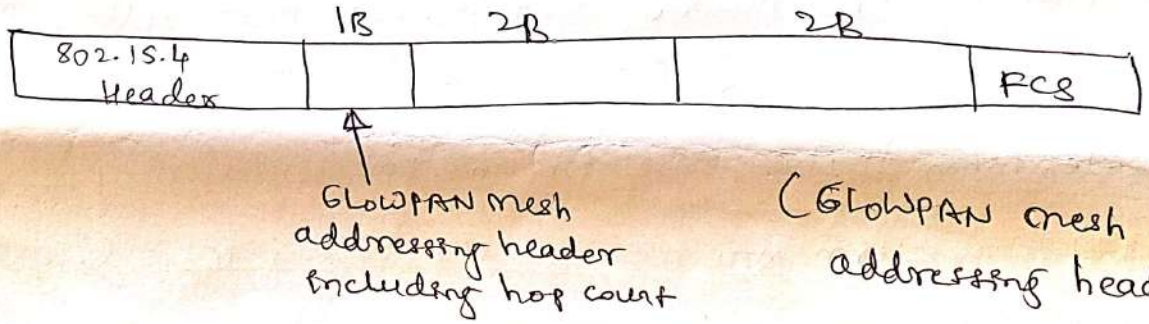
[6LoWPAN fragmentation Header]

⟹ The 1-byte datagram size field specifies the total size of the unfragmented payload. Datagram tag identifies the

set of fragments for a payload. Finally, the datagram offset field delineates how far into a payload a particular fragment occurs.

Mesh Addressing - The purpose of the 6LoWPAN mesh addressing function is to forward packets over multiple hops. Three fields are defined for this header: Hop limit, Source address, & Destn address.

The hop-limit provides an upper limit on how many times the frame can be forwarded. Each hop decrements this value by 1 as it is forwarded. once the value hits 0, it is dropped and no longer forwarded.



6TISCH -

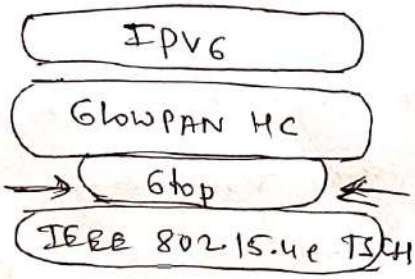
- IEEE 802.15.4e, Time Slotted channel Hopping (TSCH) is an add-on to the media Access Control (MAC) portion of the IEEE 802.15.4 standard. Here devices follow TDMA Schedule. An allocation of a unit of b.w or time Slot is Scheduled between neighbor nodes. This allows the programming of predictable trans & enable deterministic, industrial-type applications.

→ To standardize IPv6 over TSCH mode of IEEE 802.15.4e the IETF formed the 6TISCH Working group.

An important element specified by the 6TISCH working group is 6top, a Sublayer that glues together the MAC and

6LoWPAN adaptation layer. This sublayer provides Commands to the upper n/w layers, such as RPL. In return, these Commands enable functionalities including n/w layer routing decisions, Configuration, & Control procedures for 6TiSCH Schedule management.

The below figure shows where 6top resides in relation to IEEE 802.15.4e, 6LoWPAN HC & IPV6.



(Location of 6TiSCH's 6top Sublayer)

The 6TiSCH arch. defines four schedule management mechanisms:

↳ Static Scheduling - All the nodes in the Constrained n/w share a fixed schedule. Cells are shared & nodes contend for slot access in a slotted aloha manner. The drawback with static scheduling is that nodes may expect a packet at any cell in the schedule. Therefore, energy is wasted idly listening across all cells.

↳ Neighbor-to-Neighbor Scheduling - A Schedule is established that correlates with the observed no. of tx's between nodes. Cells in this Schedule can be added or deleted as traffic requirements & b.w needs change.

↳ Remote monitoring & Schedule management - Time slots and other resource allocation are handled by a management entity that can be multiple hops away. This mechanism provides quite a bit of flexibility & control in allocating cells for communication between nodes.

↳ Hop-by-hop Scheduling - A node reserves a path to a destination node multiple hops away by requesting the allocation of cells in a schedule at each intermediate node hop in the path.

⇒ Schedules in 6TiSCH are broken down into cells. A cell is simply a single element in the TSCH schedule that can be allocated for unidirectional or bidirectional communications between specific nodes.

In addition to Schedule management functions, the 6TiSCH arch. also defines three diff. forwarding models: Track forwarding (TF), fragment forwarding (FF) & IPv6 forwarding (6F).

Track forwarding - A "track" in this model is a unidirectional path between a source & destination. This track is constructed by pairing bundles of receive cells in a schedule with a bundle of receive cells set to transmit. So, a frame received within a particular cell or cell bundle is switched to another cell or cell bundle.

Fragment forwarding - This model takes advantage of 6LoWPAN fragmentation to build layer 2 forwarding table. With FF, a mechanism is defined where the first fragment is routed based on the IPv6 header present. The 6LoWPAN sublayer learns the next-hop selection of this first fragment, which is then applied to all subsequent fragments of that packet. otherwise, IPv6 packets undergo hop-by-hop reassembly. This increases latency & can be power- and CPU-intensive for a constrained node.

IPV6 forwarding – This model forwards traffic based on the IPV6 routing table. Flows of packets should be prioritized by traditional QoS & RED operations.

RPL

→ The IETF chartered the ROLL (Routing over Low-power & Lossy NWs) working group to evaluate all layer 3 IP routing protocols & determine the needs & requirements for developing a routing solution for IP smart objects. The new distance-vector routing protocol was named the IPV6 routing protocol for Low Power & Lossy nlws (RPL). The RPL specification was published as RFC 6550.

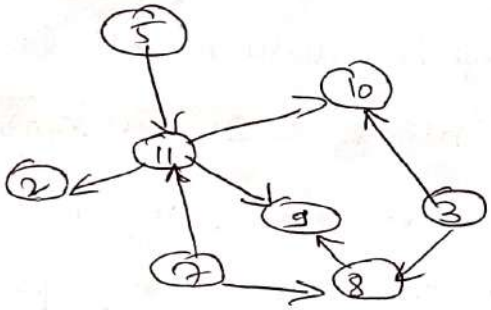
⇒ In an RPL nlw, each node acts as a router & becomes part of the mesh nlw. Routing is performed at IP layer. Each node examines every received IPV6 packet & determines the next-hop destination based on the info contained in the IPV6 header. No info from MAC layer header is needed to perform next-hop determination.

⇒ To cope with the constraints of computing & memory that are common characteristics of constrained nodes, the protocol defines two modes:

↳ Storing mode - All nodes contain the full routing table of the RPL domain. Every node knows how to directly reach every other node.

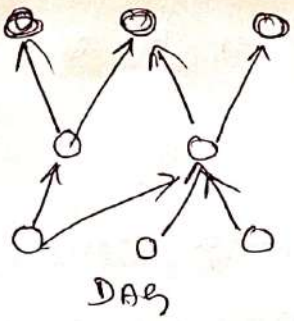
↳ Non-storing mode — only the border router(s) of the RPL domain contain(s) the full routing table. All other nodes in the domain maintain their list of parents and use this as a list of default routes toward the border router.

⇒ RPL is based on the concept of Directed Acyclic graph(D, A
A DAG is a directed graph where no cycles exist. This means that from any vertex or point in the graph, you cannot follow an edge or line back to this same point.

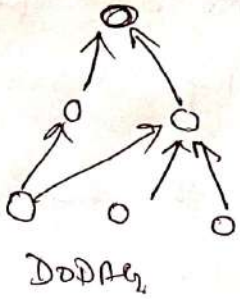


(Example of a DAG)

→ A basic RPL process involves building a destn-oriented directed acyclic graph (DODAG). A DODAG is a DAG rooted to one destn. In RPL, this destn occurs at a border router known as the DODAG root. The below figure compares DAG and DODAG. You can see that a DAG has multiple roots whereas DODAG has just one.



DAG Roots



(DAG & DODAG Comparison)

→ In DODAG, each node maintains upto three parents that provide a path to the root. Typically one of these parents is the preferred parent, which means it is the preferred next hop for upward routes towards the root.

Objective function (OF) - defines how metrics are used to select routes & establish a node's rank. (RFC 6552 & 6719)

Rank - The rank is rough approximation of how "close"

a node is to the root & helps avoid routing loops & the Count to infinity problem.

RPL Headers – RFC 6553 defines headers.

Metrics – Some of the RPL routing metrics & Constraints defined in RFC 6551 include the following:

↳ Expected Txn Count (ETX) – Assigns a discrete value to the no. of txns a node expects to make to deliver a packet.

↳ Hop Count – tracks the no. of nodes traversed in a path.

↳ Latency – varies depending on power conservation.

↳ Link quality level – measures the reliability of a link by taking into account packet error rates caused by factors, such as signal attenuation & interference.

Node State & Attribute – Identifies nodes that function as traffic aggregators & node that are being impacted by high workloads.

Nodes Energy – Avoids nodes with low power.

Throughput – provides an amount of throughput for a node link

Link Color – Setting values to make a link more or less desirable.

* Application Protocols for IoT

The Transport Layer – With the TCP/IP protocol, two main protocols are specified for the transport layer:

↳ Txn Control protocol (TCP) – this connection-oriented protocol requires a session to get established between the source and dest. before exchanging data. You can view it as an

equivalent to a traditional telephone Conversation, in which two phones must be Connected & Comm^{n} link established before the parties can talk.

↳ User Datagram protocol - With this Connectionless protocol, data can be quickly sent between source & destination - but with no guarantee of delivery. This is analogous to the traditional mail delivery system, in which a letter is mailed to a destination Confirmation of the reception of this letter does not happen until another letter is sent in response.

⟹ With the predominance of human interactions over the Internet, TCP is the main protocol used at the transport layer. This is largely due to its inherent characteristics, such as its ability to transport large volumes of data into smaller sets of packets. In addition, it ensures reassembly in a Correct sequence, flow Control & window adjustements, & retx^{n} of lost packets.

In Contract, UDP is most often used in the Context of n/w services, such as DNS, NTP, SNMP & DHCP, or for real-time data traffic, including voice & video over IP. In these cases, performance & latency are more important than packet retransmissions because - re-sending a lost voice or video packet does not add value.

→ when Considering the choice of a transport layer by a given IoT appln layer protocol, it is recommended to evaluate the impact of this choice on both the lower & upper layers of the stack.

While the use of TCP may not strain generic Compute platforms & high data-rate n/ws, it can be challenging & is often

overkill on constrained IoT devices & n/ws. This is particularly true when an IoT device needs to send only a few bytes of data per trxn. When using TCP, each packet needs to add a minimum of 20 bytes of TCP overhead, while UDP adds only 8 bytes. TCP also requires the establishment & potential maintainance of an open logical channel.

→ IoT nodes may also be limited by intrinsic characteristics of the data link layers. For ex, low-power & lossy n/ws may not cope well with supporting large no. of TCP sessions.

This may explain why a new IoT application protocol such as Constrained Application protocol (CoAP) almost always uses UDP & why implementation of industrial appln layer protocols now call for the optimization & the adoption of the UDP transport layer if run over LLNs. For ex, the Device Lang. message specification/ Companion Specification for energy. metering (DLMS/COSEM) appln layer protocol, a popular protocol for reading smart meters in the utilities space, is the defacto standard in the Europe. Adjustments or optimizations to this protocol should be made depending on the IoT transport protocols that are present in the lower layers. For ex, if you compare the transport of DLMS/COSEM over a cellular n/w versus an LLN deployment, you should consider the following:

↳ Select TCP for cellular n/ws because these n/ws are typically more robust & can handle the overhead. For LLNs, where both the devices & n/w itself are usually constrained, UDP is a better choice and often mandatory.

↳ DLMS/COSEM can reduce the overhead associated with session establishment by offering "long association" over

LLNs. Long association means that sessions stay up once put in place because the Comms overhead necessary to keep a session established is much less than is involved in opening & closing many seperate sessions over the same time period. When transferring large amounts of DLMS/COSEM data, cellular links are preferred to optimize each open associations. Smaller amounts of data can be handled efficiently over LLNs.

* IoT Appl. Transport methods The following Categories of IoT application Protocols & their transport methods are explored in the following sections:

Appl. layer protocol not present: In this case, the data payload is directly transported on top of the lower layers. No application layer protocol is used.

SCADA - is one of the most common in industrial protocols, but it was developed long before the days of IP & it has been adapted for IP nlws.

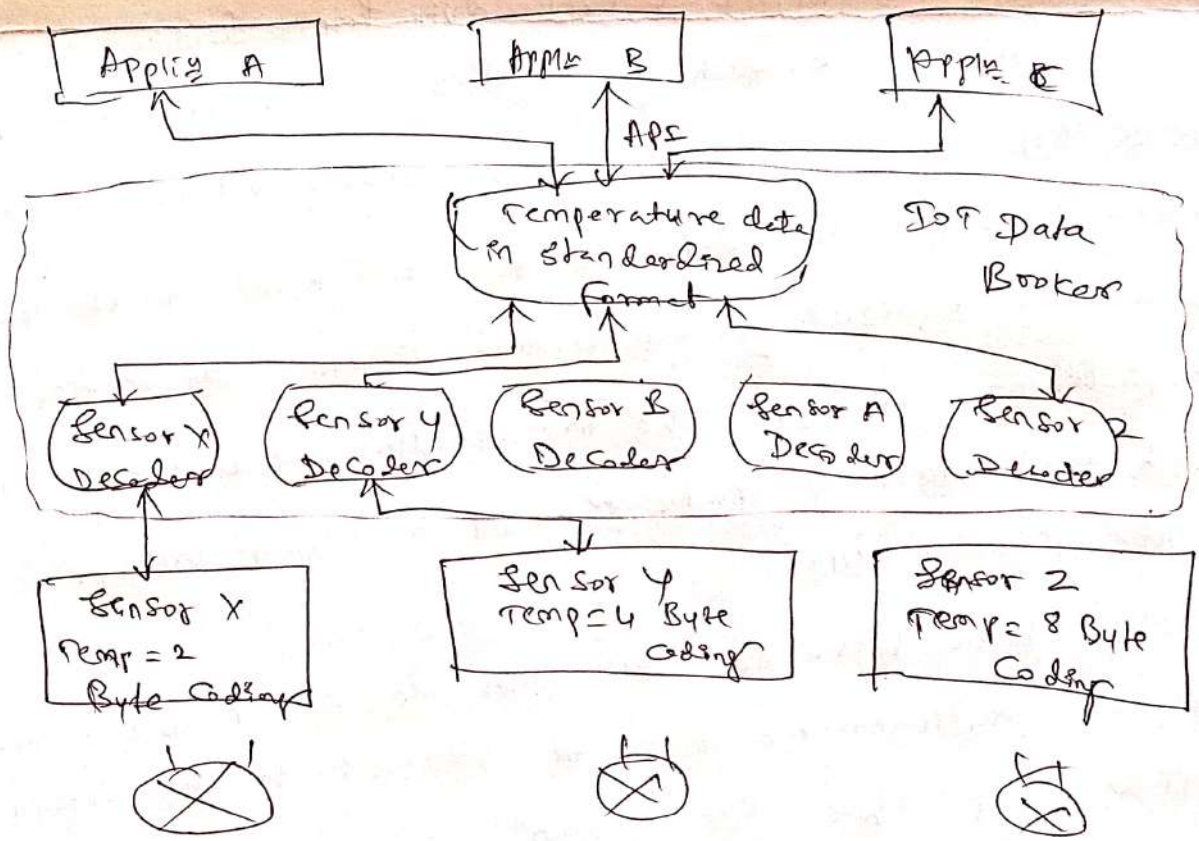
Generic web-based protocols - Generic protocols such as Ethernet, Wi-Fi, & 4G/LTE are found on many Consumer-and enterprise-class IoT devices that Communicate over non-Constrained nlws.

IoT application layer protocols - are devised to run on constrained nodes with a small Compute footprint & are well adapted to the nlw b.w Constraints on Cellular or Satellite links or Constrained 6LoWPAN nlws.

Application layer protocol not present - while many Constrained devices, such as Sensors & actuators have adopted deployments that have no application layer, this transportation mode

has not been standardized. This lack of standardization makes it difficult for generic implementations of this transport method to be successful from an interoperability perspective.

Imagine example to different kind of temperature sensors from different manufacturers. These sensors will report temp data in varying formats. A temp value will always be present in the data transmitted by each sensor, but decoding this data will be vendor specific. The solution to this problem is to use an IoT data broker as detailed in below figure. An IoT data broker is a piece of middleware that standardizes sensor o/p into a common format that can then be retrieved by authorized applications.



(IoT Data Broker)

In above figure sensors X, Y & Z are all temp. sensors but their o/p is encoded differently. The IoT data

broker understands the different formats in which the temp. is encoded & is therefore able to decode that data into a common, standardized format. Apps A, B & C can access this temperature data without having to deal with decoding multiple temperature data formats.

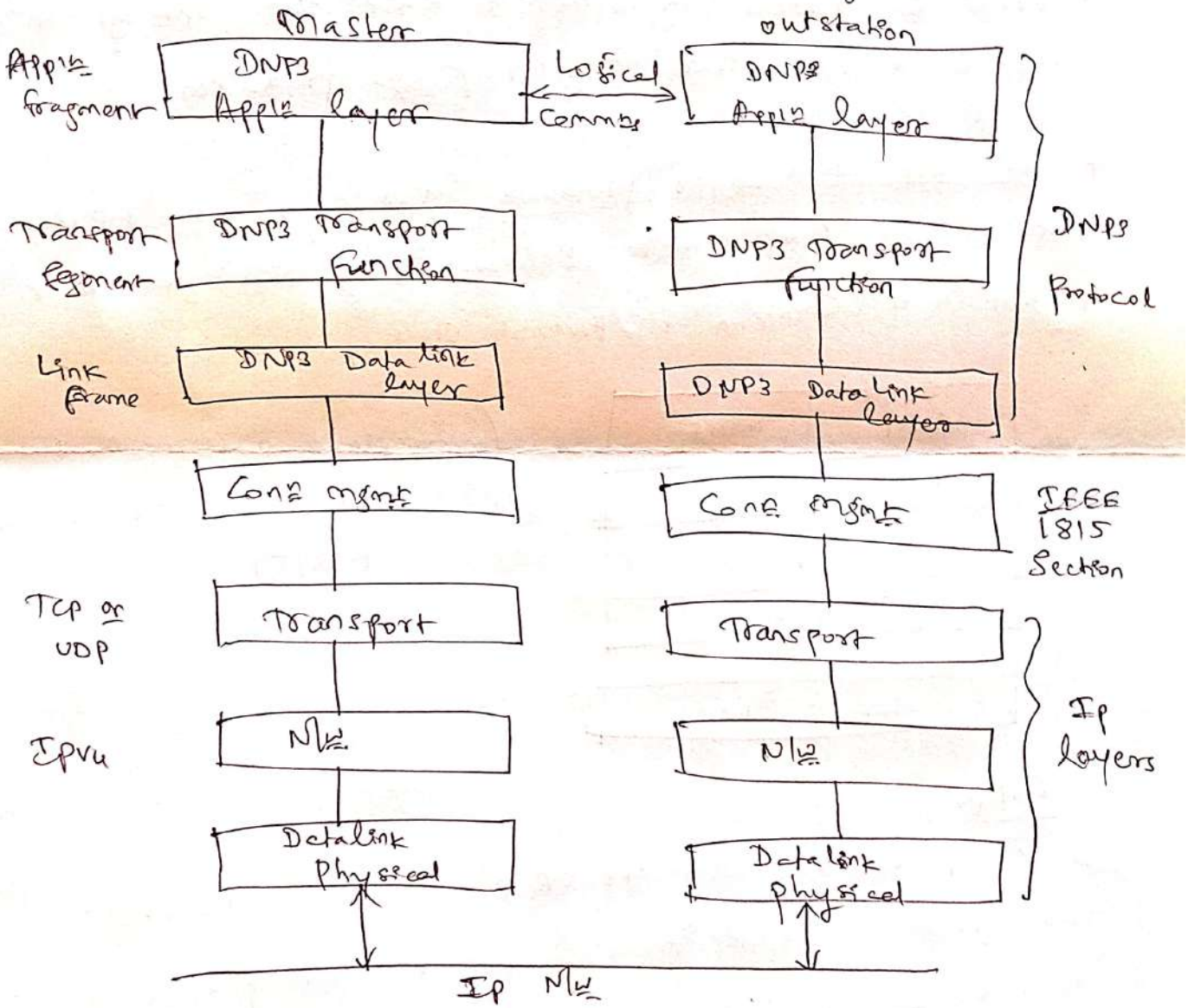
SCADA

Like many of the other SCADA protocols, DNP3 is based on a master/slave relationship. The term master in this case refers to what is typically a powerful computer located in the control center of a utility, & a slave is a remote device with computing resources found in a location such as a sub-station. DNP3 refers to slaves specifically as Outstations.

⇒ outstation monitor & collect data from devices that indicate their state, such as whether a circuit breaker is on or off & take measurements, including voltage, current, temp & so on. This data is then transmitted to the master when it is requested, or events and alarms can be sent in an asynchronous manner. The master also issues control commands, such as to start a motor or reset a circuit breaker, & logs the incoming data.

⇒ The IEEE 1815-2012 specification describes how the DNP3 protocol implementation must be adapted to run either over TCP or UDP. This specification defines Comm mgmt between the DNP3 protocol & the IP layers, as shown in below figure. Comm management links the DNP3 layers with the IP layers in addition to the configuration parameters & methods

necessary for implementing the network connection. The IP layers appear transparent to the DNP3 layers as each piece of the protocol stack on one station logically communicates with the respective part in the other. This means that the DNP3 endpoints or devices are not aware of the underlying IP transport that is occurring.



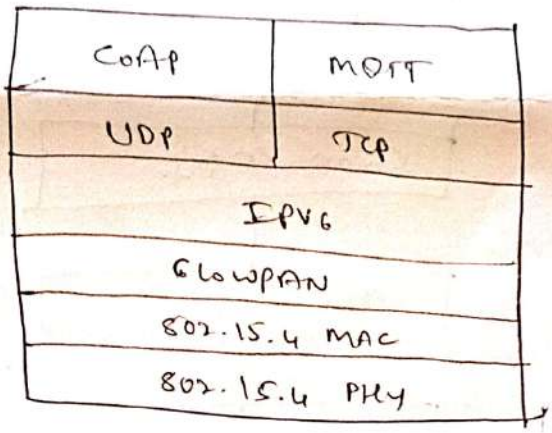
(Protocol stack for transporting Serial DNP3 SCADA over IP)

In above figure, the master side initiates connections by performing a TCP active open. The outstation listens for a connection requesting by performing a TCP Passive open. Dual end point is defined as a process that can both

listen for connection requests & perform an active open on the channel if required.

Generic web-Based Protocols over the years, web-based protocols have become common in consumer & enterprise apps & services. Therefore, it makes sense to try to leverage these protocols when developing IoT applications, services & devices in order to ease the integration of data & devices from prototyping to production.

* IoT Application Layer protocols Two of the most popular protocols are CoAP & MQTT. Below figure highlights their position in a common IoT protocol stack.



(Example of a high-level IoT protocol stack for CoAP & MQTT)

CoAP

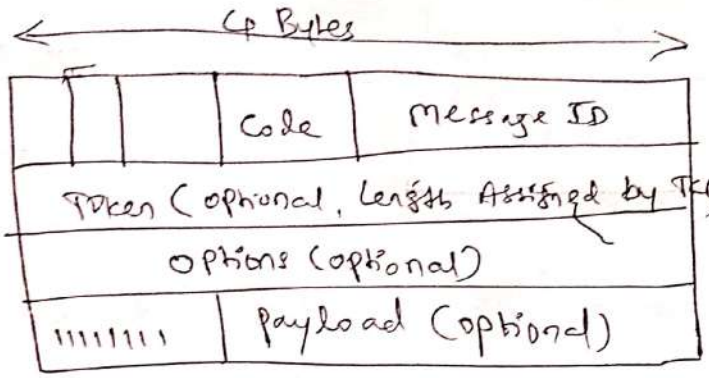
→ CoAP resulted from the IETF Constrained RESTful Environments working group's effort to develop a generic framework for resource-oriented applications targeting constrained nodes & nws.

→ The CoAP framework defines simple & flexible ways to manipulate sensors & actuators for data or device management.

→ The CoAP messaging model is primarily designed ~~for~~ to

facilitate the exchanges of messages over UDP between end-points.

→ The below figure details the CoAP message format, which delivers low overhead while decreasing parsing complexity.



(CoAP message format)

CoAP message field	Description
Ver (Version)	Identifies the CoAP version.
T (Type)	Defines one of the following four message types: Confirmable (CON), Non-Confirmable (NON), Acknowledgement (ACK) & Reset (RST)
TKL (Token length)	Specifies the size (0-8 bytes) of the token field.
Code	Indicates the request method for a request message & a response code for a response message
Message ID	Detects msg duplication & used to match ACK & RST msg types to CON & NON msg types.
Token	With a length specified by TKL, correlates requests & responses.
Options	Specifies option number, length & option value.
Payload	Carries the CoAP appln data.

[CoAP message fields]

* MQTT message format

← 1 Byte →

Message Type	DUP	QoS	Retain
Remaining Length			
Variable Header (optional)			
payload (optional)			

(MQTT message format)

→ message type field – indicates message types. (Refer Table 6.2)

→ DUP (Duplication flag) – This flag ~~is set~~ when set, allows the client to notate that the packet has been sent previously but an acknowledgment was not received.

→ QoS – field allows the selection of diff. QoS levels.

→ Retain flag – notifies the server to hold onto the msg data.

→ Remaining length – Specifies the no. of bytes in the MQTT Packet following this field.

—— o ——