Introduction to IOT and its Applications

Unit Number: 01

Unit Name: Introduction to IoT

Kaushal Mittal

Aim

- To equip the students with basics of IOT.

Instructional Objectives

- Introduction of Unit
- Definition and IoT Chrematistics
- Physical Design of IoT- IoT Protocols
- Logical Design of IoT - IoT Functional Blocks
- IoT Communication Models
- IoT Communication APIs
- IoT Enable Technologies – Wireless Sensor Networks, Cloud Computing, Big Data Analytics, Communication Protocol, Embedded System
- Conclusion of the Unit

IOT

- The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction.
- The definition of the Internet of things has evolved due to the convergence of multiple technologies, real-time analytics, machine learning, commodity sensors, and embedded systems.
- The Internet of Things is making the fabric of the world around us more smarter and more responsive, merging the digital and physical universes.
- The IoT is a giant network of connected things and people – all of which collect and share data about the way they are used and about the environment around them.

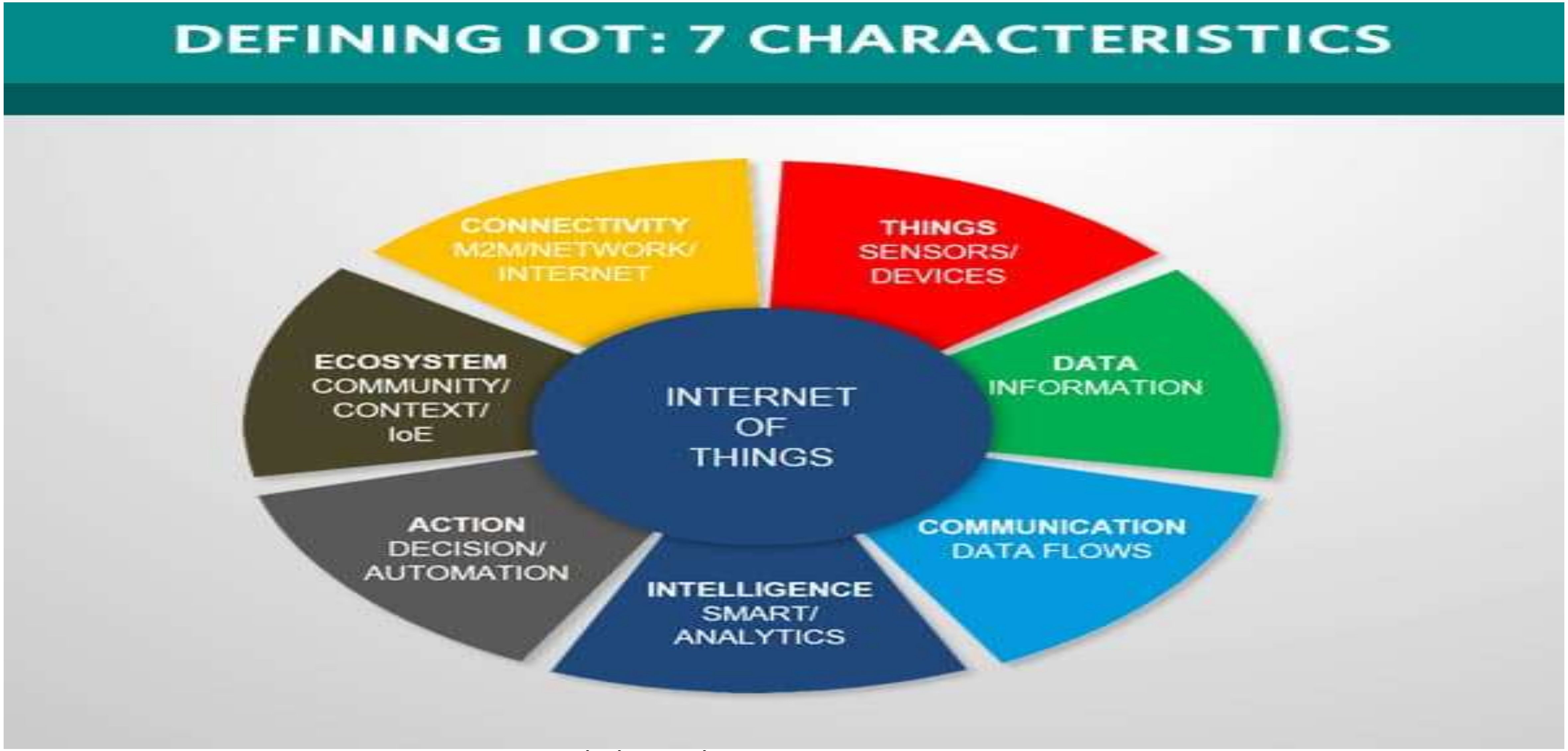
What Technologies Have Made IoT Possible?

- **Access to low-cost, low-power sensor technology.** Affordable and reliable sensors are making IoT technology possible for more manufacturers.
- **Connectivity.** A host of network protocols for the internet has made it easy to connect sensors to the cloud and to other "things" for efficient data transfer.
- **Cloud computing platforms.** The increase in the availability of cloud platforms enables both businesses and consumers to access the infrastructure they need to scale up without actually having to manage it all.
- **Machine learning and analytics.** With advances in machine learning and analytics, along with access to varied and vast amounts of data stored in the cloud, businesses can gather insights faster and more easily. The emergence of these allied technologies continues to push the boundaries of IoT and the data produced by IoT also feeds these technologies.
- **Conversational artificial intelligence (AI).** Advances in neural networks have brought natural-language processing (NLP) to IoT devices (such as digital personal assistants Alexa, Cortana, and Siri) and made them appealing, affordable, and viable for home use.

Top IoT Applications

- **Create new efficiencies in manufacturing through machine monitoring and product-quality monitoring.** Machines can be continuously monitored and analyzed to make sure they are performing within required tolerances. Products can also be monitored in real time to identify and address quality defects.
- **Improve the tracking and "ring-fencing" of physical assets.** Tracking enables businesses to quickly determine asset location. Ring-fencing allows them to make sure that high-value assets are protected from theft and removal.
- **Use wearables to monitor human health analytics and environmental conditions.** IoT wearables enable people to better understand their own health and allow physicians to remotely monitor patients. This technology also enables companies to track the health and safety of their employees, which is especially useful for workers employed in hazardous conditions.
- **Drive efficiencies and new possibilities in existing processes.** One example of this is the use of IoT to increase efficiency and safety in fleet management. Companies can use IoT fleet monitoring to direct trucks, in real time, to improve efficiency.
- **Enable business process changes.** An example of this is the use of IoT devices to monitor the health of remote machines and trigger service calls for preventive maintenance. The ability to remotely monitor machines is also enabling new product-as-a-service business models, where customers no longer need to buy a product but instead pay for its usage.

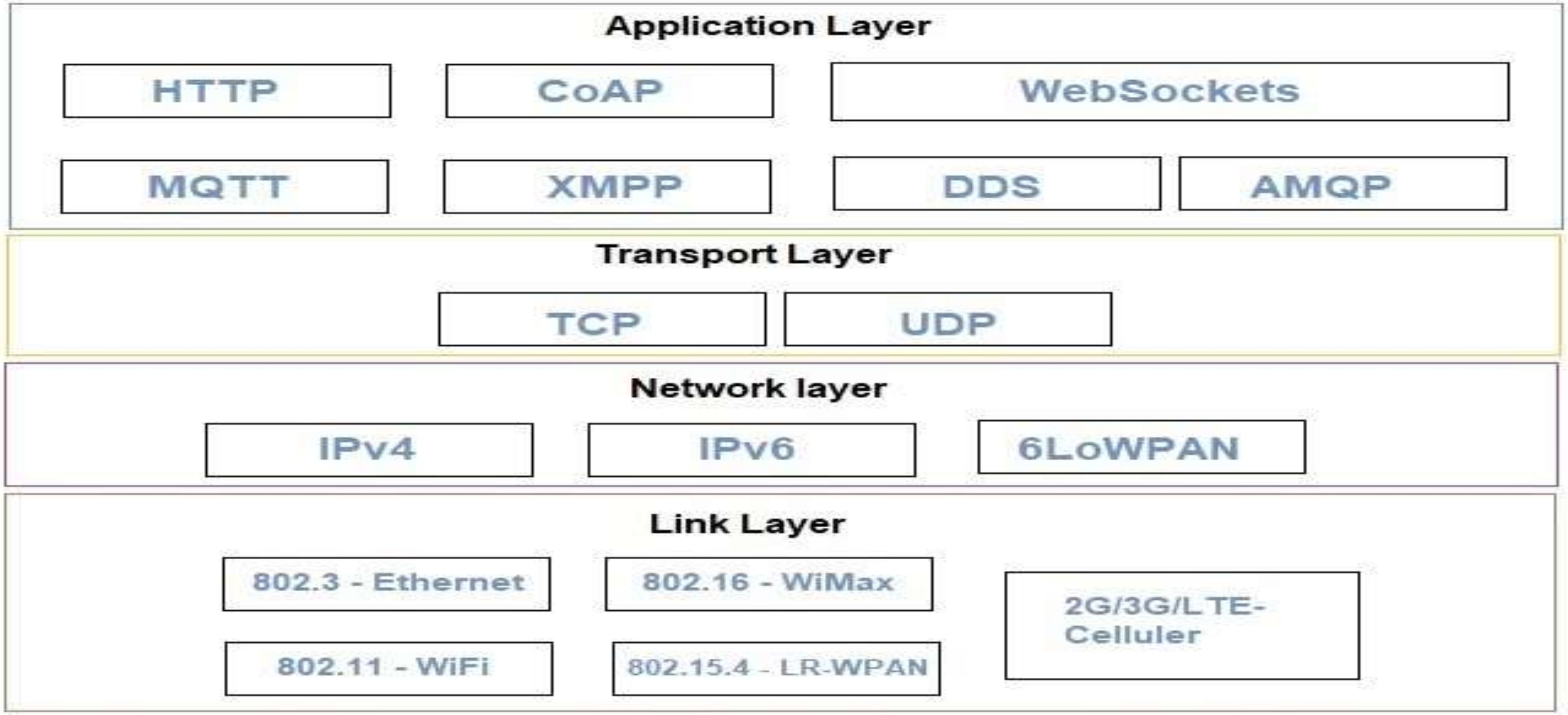
Key Characteristics


DEFINING IOT: 7 CHARACTERISTICS
CONNECTIVITY M2M/NETWORK/ INTERNET
THINGS SENSORS/ DEVICES
DATA INFORMATION
COMMUNICATION DATA FLOWS
INTELLIGENCE SMART/ ANALYTICS
ACTION DECISION/ AUTOMATION
ECOSYSTEM COMMUNITY/ CONTEXT/ IoE
INTERNET OF THINGS

Key Characteristics

- **Connectivity-** This doesn’t need too much further explanation. With everything going on in IoT devices and hardware, with sensors and other electronics and connected hardware and control systems there needs to be a connection between various levels.
- **Thing-** Anything that can be tagged or connected as such as it’s designed to be connected. From sensors and household appliances to tagged livestock. Devices can contain sensors or sensing materials can be attached to devices and items.
- **Data-** Data is the glue of the Internet of Things, the first step towards action and intelligence.
- **Communication-** Devices get connected so they can communicate data and this data can be analyzed. Communication can occur over short distances or over a long range to very long range. Examples: Wi-Fi, LPWA network technologies such as LoRa or NB-IoT.
- **Intelligence-** The aspect of intelligence as in the sensing capabilities in IoT devices and the intelligence gathered from big data analytics (also artificial intelligence).
- **Action-** The consequence of intelligence. This can be manual action, action based upon debates regarding phenomena (for instance in smart factory decisions) and automation, often the most important piece.
- **Ecosystem-** The place of the Internet of Things from a perspective of other technologies, communities, goals and the picture in which the Internet of Things fits. The Internet of Everything dimension, the platform dimension and the need for solid partnerships.

Physical Design of IOT



IOT Protocols

- Infrastructure (ex: 6LowPAN, IPv4/IPv6, RPL)
- Identification (ex: EPC, uCode, IPv6, URIs)
- Comms / Transport (ex: Wifi, Bluetooth, LPWAN)
- Discovery (ex: Physical Web, mDNS, DNS-SD)
- Data Protocols (ex: MQTT, CoAP, AMQP, Websocket, Node)
- Device Management (ex: TR-069, OMA-DM)
- Semantic (ex: JSON-LD, Web Thing Model)
- Multi-layer Frameworks (ex: Alljoyn, IoTivity, Weave, Homekit)

Infrastructure

- **IPv4/IPv6 -** IPv4 and IPv6 are the two major versions of the Internet Protocol (which is what IP stands for in their acronyms). The Internet Protocol is a method for the delivery of packets of data between hosts on a network (most notably, the Internet) and for uniquely identifying hosts on a network using what is known as an IP address. IPv4 and v6 packets both include headers, which are metadata related to the data packet, such as sender IP address and recipient IP address as well as a payload, which is the actual data to be transmitted.
- **6LoWPAN -** 6LoWPAN stands for IPV6 over Low Power Wireless Personal Area Networks. It's a standard aimed at making it possible for IoT devices, which often operate with limited battery supply and processing power, to communicate using IPV6 packets. Using header compression and other power-saving technologies, 6LoWPAN allows devices to communicate over IEEE 802.15.4 networks (which are designed for low power communication) using the same packet structure as the internet at large.
- **RPL -** RPL is a routing protocol designed for low power and lossy networks, meaning networks comprised of low power devices that may experience packet loss. RPL is characterized by its optimization for many-to-one communication and for its ability of nodes to efficiently share network topology information (information about the shape and composition of the network) with each other.

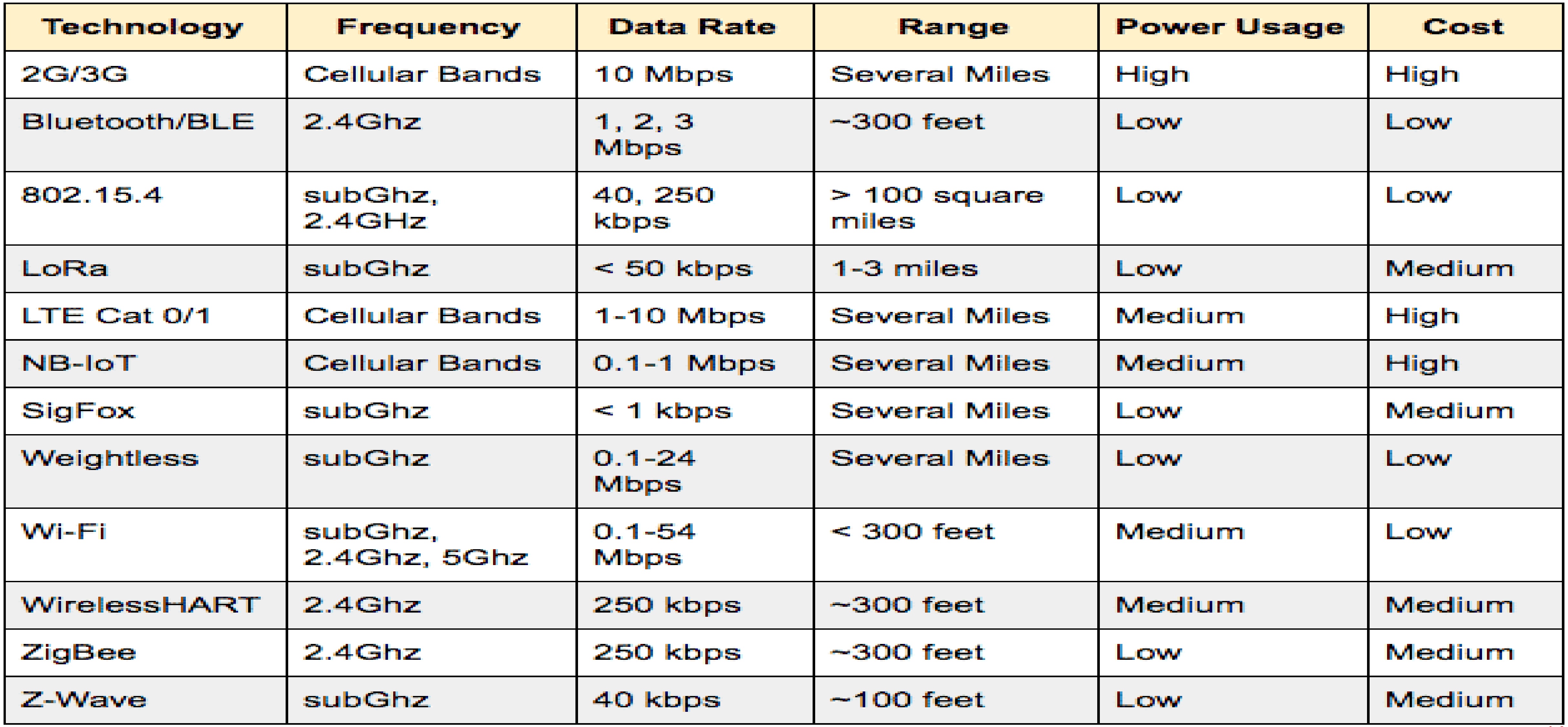
Infrastructure

- **NanoIP**
- "NanoIP, which stands for the nano Internet Protocol, is a concept that was created to bring Internet-like networking services to embedded and sensor devices, without the overhead of TCP/IP. NanoIP was designed with minimal overheads, wireless networking, and local addressing in mind."
- **Content-Centric Networking (CCN) -** Technical Overview
- "Next-gen network architecture to solve challenges in content distribution scalability, mobility, and security.
- CCN directly routes and delivers named pieces of content at the packet level of the network, enabling automatic and application-neutral caching in memory wherever it's located in the network. The result? Efficient and effective delivery of content wherever and whenever it is needed. Since the architecture enables these caching effects as an automatic side effect of packet delivery, memory can be used without building expensive application-level caching services."
- **Time Synchronized Mesh Protocol (TSMP)**
- A communications protocol for self-organizing networks of wireless devices called motes. TSMP devices stay synchronized to each other and communicate in timeslots, similar to other TDM (time-division multiplexing) systems.

Identification

- **EPC -** EPC stands for Electronic Product Code. It’s a standard defined by GS1, a not-for-profit concerned with standardizing business communications. EPC outlines a system for encoding and decoding universally unique identifiers for any physical object in the world based on product category information (such as a UPC) and serial numbers that identify unique instances of a type of object.
- **uCode -** ucode is a unique identification system that works in a similar manner to the ICANN domain name registration and DNS management systems. Unique codes are managed by a set of redundant ID servers that can issue ucode IDs and also resolve them, returning device information, much in the same way a DNS server resolves a unique URL to an IP address.
- **MAC Address -** All devices with a network connection have a MAC Address assigned to each of their network interfaces. MAC Addresses are unique and assigned to device manufacturers in batches. It’s then up to the device manufacturer to ensure that they assign the numbers from their batch to devices such that each device only has one unique number MAC Address.

Communication / Transport layer



Technology	Frequency	Data Rate	Range	Power Usage	Cost
2G/3G	Cellular Bands	10 Mbps	Several Miles	High	High
Bluetooth/BLE	2.4Ghz	1, 2, 3 Mbps	~300 feet	Low	Low
802.15.4	subGhz, 2.4GHz	40, 250 kbps	> 100 square miles	Low	Low
LoRa	subGhz	< 50 kbps	1-3 miles	Low	Medium
LTE Cat 0/1	Cellular Bands	1-10 Mbps	Several Miles	Medium	High
NB-IoT	Cellular Bands	0.1-1 Mbps	Several Miles	Medium	High
SigFox	subGhz	< 1 kbps	Several Miles	Low	Medium
Weightless	subGhz	0.1-24 Mbps	Several Miles	Low	Low
Wi-Fi	subGhz, 2.4Ghz, 5Ghz	0.1-54 Mbps	< 300 feet	Medium	Low
WirelessHART	2.4Ghz	250 kbps	~300 feet	Medium	Medium
ZigBee	2.4Ghz	250 kbps	~300 feet	Low	Medium
Z-Wave	subGhz	40 kbps	~100 feet	Low	Medium

Comms/Transport

- **Wi-Fi -** The same Wi-Fi technology that connects most of our computers and smart devices to the internet can be used to connect IoT devices as well. Because Wi-Fi has a relatively high power requirement compared to other network technologies, it's not often the first choice for power-constrained devices, but its ubiquity makes it a viable option for certain solutions.
- **Bluetooth -** Bluetooth has both the advantages of the ubiquity of adoption and (at least in the latest versions) a very low power profile. However, Bluetooth has a relatively short transmission range, which is not ideal for large geographic scale applications.
- **LPWAN -** LPWAN (short for low-power wide-area network) is a technology that is both power efficient and can allow data to be transmitted over long distances. The primary downside of LPWAN is its very low transmission rate, which makes it challenging for applications where large amounts of data need to be transferred to or from IoT devices. LPWAN is also not a universally adopted networking technology. Thus, if integration with existing networks is important, LPWAN may not be the best choice.
- **ZigBee -** The ZigBee protocol uses the 802.15.4 standard and operates in the 2.4 GHz frequency range with 250 kbps. The maximum number of nodes in the network is 1024 with a range up to 200 meter. ZigBee can use 128 bit AES encryption.
- **NFC-** Based on the standard ISO/IEC 18092:2004, using inductive coupled devices at a center frequency of13.56 MHz. The data rate is up to 424 kbps and the rangeis with a few meters short compared to the wireless sensornetworks.

Discovery

- **mDNS (multicast Domain Name System) -** Resolves host names to IP addresses within small networks that do not include a local name server.
- **Physical Web -** The Physical Web enables you to see a list of URLs being broadcast by objects in the environment around you with a Bluetooth Low Energy (BLE) beacon.
- **HyperCat -** An open, lightweight JSON-based hypermedia catalogue format for exposing collections of URIs.
- **UPnP (Universal Plug and Play) -** Now managed by the Open Connectivity Foundation is a set of networking protocols that permits networked devices to seamlessly discover each other's presence on the network and establish functional network services for data sharing, communications, and entertainment.
- **MQTT (Message Queuing Telemetry Transport)**
- "The MQTT protocol enables a publish/subscribe messaging model in an extremely lightweight way. It is useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium."
- **MQTT-SN (MQTT For Sensor Networks) -** An open and lightweight publish/subscribe protocol designed specifically for machine-to-machine and mobile applications
- **Mosquitto:** An Open Source MQTT v3.1 Broker
- IBM MessageSight

Data Protocols

- **MQTT -** MQTT is a publish/subscribe messaging protocol used to pass data between devices in an environment with low network bandwidth. There are two types of devices on an MQTT network: publishers and message brokers. A message broker manages subscription topics and the devices that are described to those types of messages. It then publishes updates to subscribers when new information is published to a topic.
- **CoAP -** CoAP is a web transfer protocol built on top of UDP, which is an alternative transport protocol to TCP. TCP is also what the web is built on. CoAP is designed to allow REST programming interfaces between IoT devices and servers. REST is a popular programming paradigm that allows a server to expose resources to clients that they can view, update, create and delete using a well documented and time tested format.
- **WebSocket -** WebSocket is a protocol that allows bi-directional communication between clients (devices and servers) over a single connection, allowing faster communication than alternatives like HTTP. WebSocket can be used, for instance, to channel data messages from devices or gateways to a server for processing, but can also allow the server to send messages back to devices or gateways (e.g. to request action on the part of the device or to send a software or firmware update).

Device Management

- **TR-069 -** TR-069 is a protocol designed to allow devices to be automatically configured for use on a network and for those devices to receive future software, firmware and configuration updates. The protocol is HTTP based and a series of defined message types allow devices to securely connect to an ACS or Auto Configuration Server and to receive configuration data. This allows embedded devices like set-top boxes to connect to a network successfully and to update themselves when appropriate.
- **OMA-DM -** Similar to TR-069, OMA-DM is a standard for configuring and updating devices on a network. It was created by the Open Mobile Alliance and as such has been used extensively for provisioning, configuration, and maintenance of mobile phones and other connected consumer devices.

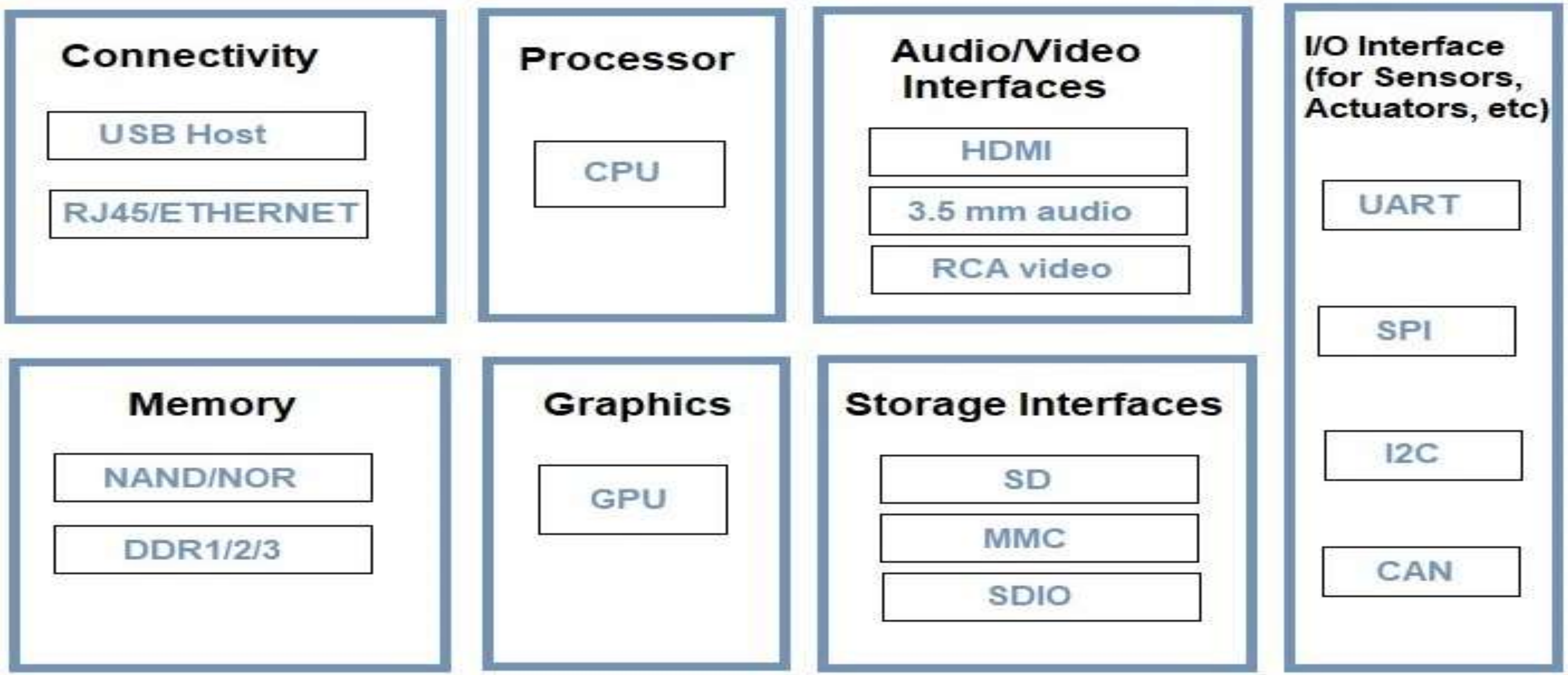
Multi-layer Frameworks

- **Alljoyn -** An open source software framework that makes it easy for devices and apps to discover and communicate with each other.
- **IoTivity** is an open source project hosted by the Linux Foundation, and sponsored by the OIC.
- **IEEE P2413 -** Standard for an Architectural Framework for the Internet of Things (IoT)
- **Thread** - Built on open standards and IPv6 technology with 6LoWPAN as its foundation.
- **IPSO Application Framework (PDF) -** "This design defines sets of REST interfaces that may be used by a smart object to represent its available resources, interact with other smart objects and backend services. This framework is designed to be complementary to existing Web profiles including SEP2 and oBIX."
- **OMA LightweightM2M v1.0 -** "The motivation of LightweightM2M is to develop a fast deployable client-server specification to provide machine to machine service.
- LightweightM2M is principly a device management protocol, but it should be designed to be able to extend to meet the requirements of applications. LightweightM2M is not restricted to device management, it should be able transfer service / application data."
- **Weave -** A communications platform for IoT devices that enables device setup, phone-to-device-to-cloud communication, and user interaction from mobile devices and the web.
- **Telehash - JSON+UDP+DHT=Freedom**
- A secure wire protocol powering a decentralized overlay network for apps and devices

Semantic

- **IOTDB -** "JSON / Linked Data standards for describing the Internet of Things"
- **SensorML -** "SensorML provides standard models and an XML encoding for describing sensors and measurement processes."
- **Semantic Sensor Net Ontology - W3C**
- "This ontology describes sensors and observations, and related concepts. It does not describe domain concepts, time, locations, etc. these are intended to be included from other ontologies via OWL imports."
- **Wolfram Language -** Connected Devices -"A symbolic representation of each device. Then there are a standard set of Wolfram Language functions like DeviceRead, DeviceExecute, DeviceReadBuffer and DeviceReadTimeSeries that perform operations related to the device."
- **RAML (RESTful API Modeling Language) -** Makes it easy to manage the whole API lifecycle from design to sharing. It's concise - you only write what you need to define - and reusable.
- **SENML (Media Types for Sensor Markup Language) -** A simple sensor, such as a temperature sensor, could use this media type in protocols such as HTTP or CoAP to transport the measurements of the sensor or to be configured.
- **LsDL (Lemonbeat smart Device Language) -** XML-based device language for service oriented devices

Logical Design of IOT



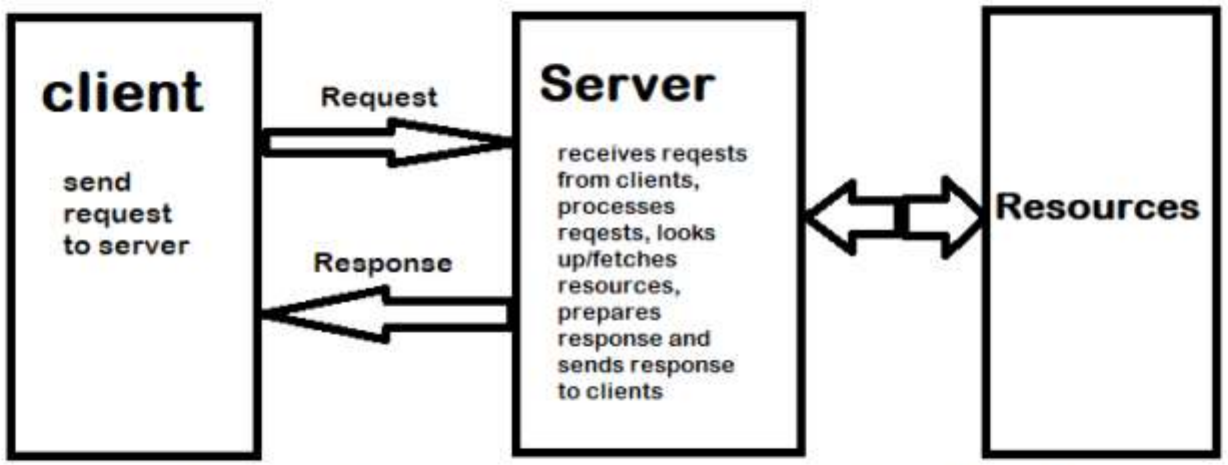
Generic Block Diagram of IoT Devices

Functional blocks

- **Device:** An IoT system comprises of devices that provide sensing, actuation, monitoring and control functions.
- **Communication:** Handles the communication for the IoT system.
- **Services:** services for device monitoring, device control service, data publishing services and services for device discovery.
- **Management:** this blocks provides various functions to govern the IoT system.
- **Security:** this block secures the IoT system and by providing functions such as authentication , authorization, message and content integrity, and data security.
- **Application:** This is an interface that the users can use to control and monitor various aspects of the IoT system. Application also allow users to view the system status and view or analyze the processed data.

IoT Communication Models

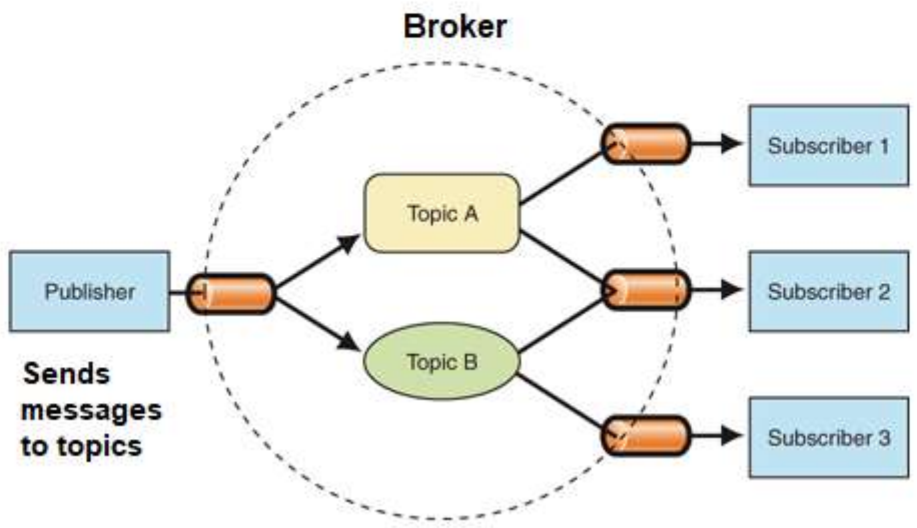
- **Request-Response Model -** Request-response model is communication model in which the client sends requests to the server and the server responds to the requests. When the server receives a request, it decides how to respond, fetches the data, retrieves resource representation, prepares the response, and then sends the response to the client. Request-response is a stateless communication model and each request-response pair is independent of others.
- HTTP works as a request-response protocol between a client and server. A web browser may be the client, and an application on a computer that hosts a web site may be the server.


client
send request to server
Request
Response
Server
receives reqests from clients, processes reqests, looks up/fetches resources, prepares response and sends response to clients
Resources

Request-Response Communication Model

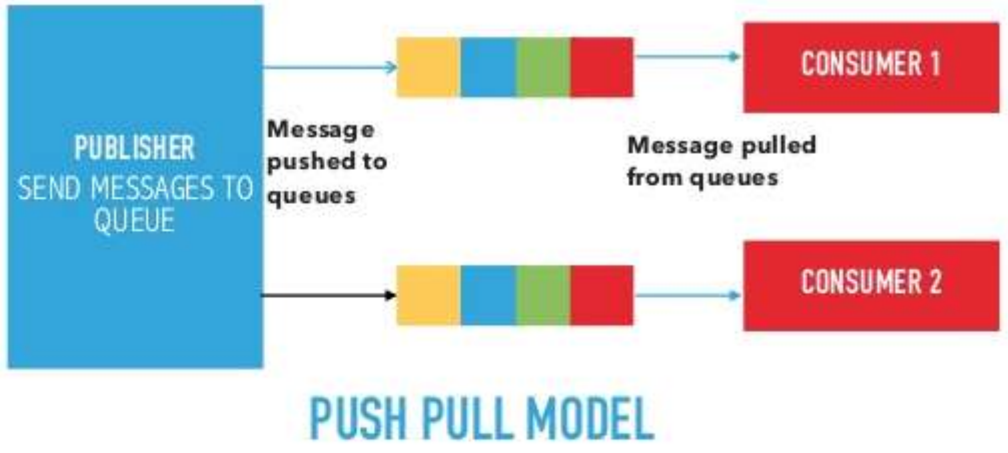
IoT Communication Models

- **Publish-Subscribe Model -** Publish-Subscribe is a communication model that involves publishers, brokers and consumers. Publishers are the source of data. Publishers send the data to the topics which are managed by the broker. Publishers are not aware of the consumers. Consumers subscribe to the topics which are managed by the broker. When the broker receive data for a topic from the publisher, it sends the data to all the subscribed consumers.



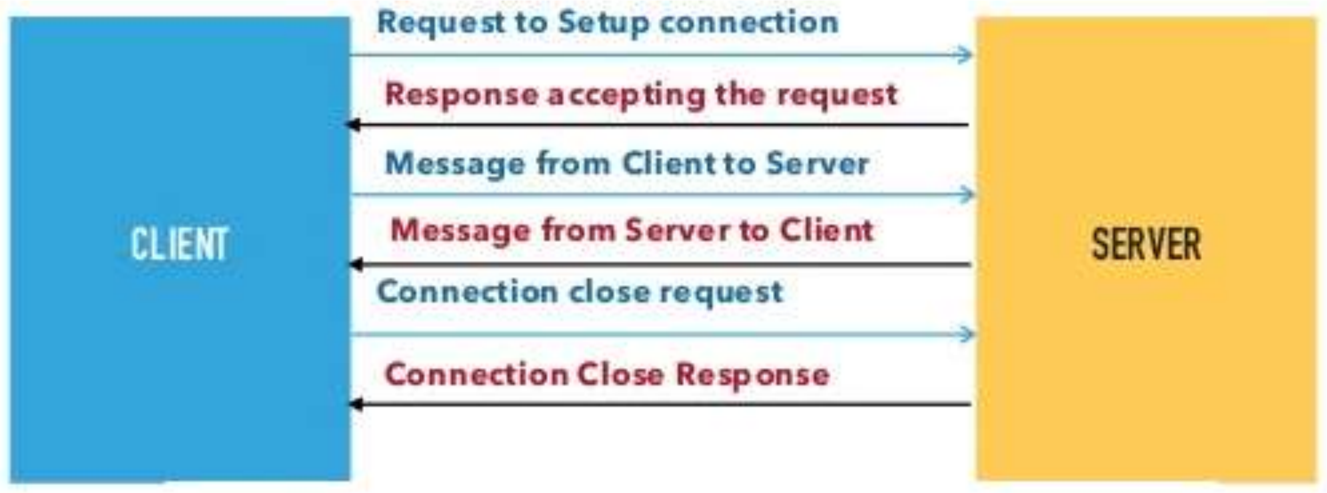
IoT Communication Models

- **Push-Pull Model -** Push-Pull is a communication model in which the data producers push the data to queues and the consumers Pull the data from the Queues. Producers do not need to be aware of the consumers. Queues help in decoupling the messaging between the Producers and Consumers. Queues also act as a buffer which helps in situations when there is a mismatch between the rate at which the producers push data and the rate rate at which the consumer pull data.



IoT Communication Models

- **Exclusive Pair Model -** Exclusive Pair is a bidirectional, fully duplex communication model that uses a persistent connection between the client and server. Connection is setup it remains open until the client sends a request to close the connection. Client and server can send messages to each other after connection setup. Exclusive pair is stateful communication model and the server is aware of all the open connections.


CLIENT
Request to Setup connection
Response accepting the request
Message from Client to Server
Message from Server to Client
Connection close request
Connection Close Response
SERVER

EXCLUSIVE PAIR COMMUNICATION MODEL

IoT Communication APIs

- REST-based Communication APIs
- WebSocket-based Communication APIs

REST-based Communication APIs

- **Client-server –** The principle behind the client-server constraint is the separation of concerns. for example clients should not be concerned with the storage of data which is concern of the serve. Similarly the server should not be concerned about the user interface, which is concern of the clien. Separation allows client and server to be independently developed and updated.
- **Stateless –** Each request from client to server must contain all the information necessary to understand the request, and cannot take advantage of any stored context on the server. The session state is kept entirely on the client.
- **Cache-able –** Cache constraints requires that the data within a response to a request be implicitly or explicitly leveled as cache-able or non cache-able. If a response is cache-able, then a client cache is given the right to reuse that repsonse data for later, equivalent requests. caching can partially or completely eliminate some instructions and improve efficiency and scalability.
- **Layered system –** layered system constraints, constrains the behavior of components such that each component cannot see beyond the immediate layer with they are interacting. For example, the client cannot tell whether it is connected directly to the end server or two an intermediaryalong the way. System scalability can be improved by allowing intermediaries to respond to requests instead of the end server, without the client having to do anything different.
- **Uniform interface –** uniform interface constraints requires that the method of communication between client and server must be uniform. Resources are identified in the requests (by URIsin web based systems) and are themselves is separate from the representations of the resources data returned to the client. When a client holds a representation of resources it has all the information required to update or delete the resource you (provided the client has required permissions). Each message includes enough information to describe how to process the message.
- **Code on demand –** Servers can provide executable code or scripts for clients to execute in their context. this constraint is the only one that is optional.

WebSocket-based Communication APIs

- Websocket APIs allow bi-directional, full duplex communication between clients and servers. Websocket APIs follow the exclusive pair communication model.
- Unlike request-response model such as REST, the WebSocket APIs allow full duplex communication and do not require new coonection to be setup for each message to be sent.
- Websocket communication begins with a connection setup request sent by the client to the server. The request (called websocket handshake) is sent over HTTP and the server interprets it is an upgrade request. If the server supports websocket protocol, the server responds to the websocket handshake response.
- After the connection setup client and server can send data/mesages to each other in full duplex mode. Websocket API reduce the network traffic and letency as there is no overhead for connection setup and termination requests for each message.
- Websocket suitable for IoT applications that have low latency or high throughput requirements. So Web socket is most suitable IoT Communication APIs for IoT System.

IoT Enable Technologies

- **Wireless Sensor Networks -** Central to the functionality and utility of the IoT are sensors embedded in smart objects. Such sensors are capable of detecting events or changes in a specific quantity (e.g., pressure), communicating the event or change data to the cloud (directly or via a gateway) and, in some circumstances, receiving data back from the cloud (e.g., a control command) or communicating with other smart objects.
- Since 2012, sensors have generally shrunk in physical size and thus have caused the IoT market to mature rapidly.
- “Technological improvements created microscopic scale sensors, leading to the use of technologies like Microelectro mechanical systems (MEMS). This meant that sensors were now small enough to be embedded into unique places like clothing or other [smart objects].”

IoT Enable Technologies

- **CLOUD COMPUTING -** Cloud computing refers to being able to access computing resources via the Internet rather than traditional systems where computing hardware is physically located on the premises of the user and any software applications are installed on such local hardware.
- A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management e ort or service provider interaction.
- Cloud computing – and its three service models of Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) – are important to the IOT because it allows any user with a browser and an Internet connection to transform smart object data into actionable intelligence.
- That is, cloud computing provides "the virtual infrastructure for utility computing integrating applications, monitoring devices, storage devices, analytics tools, visualization platforms, and client delivery… [to] enable businesses and users to access [IoT-enabled] applications on demand anytime, anyplace and anywhere."

IoT Enable Technologies

- **BIG DATA -** As more things (or "smart objects") are connected to the IoT, more data is collected from them in order to perform analytics to determine trends and associations that lead to insights.
- Big data refers to these large data sets that need to be collected, stored, queried, analyzed and generally managed in order to deliver on the promise of the IoT — insight!
- Further compounding the technical challenges of big data is the fact that IoT systems must deal with not only the data collected from smart objects, but also ancillary data that is needed to properly perform such analytics (e.g., public and private data sets related to weather, GIS, financial, seismic, map, GPS, crime, etc.).
- Thus, as more smart objects come online, at least three metrics ("the three V's") are typically used by IoT operators to describe the big data they handle: volume (i.e., the amount of data they collect from their IOT sensors measured in gigabytes, terabytes and petabytes); velocity (i.e., the speed at which data is collected from the sensors); and variety.

IoT Enable Technologies

- **DIGITAL TWIN -** Another consequence of the growing and evolving IOT is the concept of a "digital twin," introduced in 2003 by John Vickers, manager of NASA's National Center for Advanced Manufacturing.
- The concept refers to a digital copy of a physical asset (i.e., a smart object within the IOT), that lives and evolves in a virtual environment over the physical asset's lifetime.
- That is, as the sensors within the object collect real-time data, a set of models forming the digital twin is updated with all of the same information. Thus, an inspection of the digital twin would reveal the same information as a physical inspection of the smart object itself – albeit remotely.
- The digital twin of the smart object can then be studied to not only optimize operations of the smart object through reduced maintenance costs and downtime, but to improve the next generation of its design.

IoT Enable Technologies

- **COMMUNICATIONS -** With respect to sending and receiving data, wired and wireless communication technologies have also improved such that nearly every type of electronic equipment can provide data connectivity.
- This has allowed the ever-shrinking sensors embedded in smart objects to send and receive data over the cloud for collection, storage and eventual analysis.
- The protocols for allowing IoT sensors to relay data include wireless technologies such as RFID, NFC, Wi-Fi, Bluetooth, Bluetooth Low Energy (BLE), XBee, ZigBee, Z-Wave, Wireless M-Bus, SIGFOX and NuelNET, as well as satellite connections and mobile networks using GSM, GPRS, 3G, LTE, or WiMAX.
- Wired protocols, useable by stationary smart objects, include Ethernet, HomePlug, HomePNA, HomeGrid/G.hn and LonWorks, as well as conventional telephone lines.

IoT Enable Technologies

- **ANALYTICS SOFTWARE -** Within the IoT ecosystem, Application Service Providers (ASPs) – which may or may not di er from the companies who sell and service the smart objects – provide software to companies that can transform "raw" machine (big) data collected from smart objects into actionable intelligence (or insight).
- Such software performs data mining and employs mathematical models and statistical techniques to provide insight to users.
- That is, events, trends and patterns are extracted from big data sets in order to present the software's end-users with insightin the form of portfolio analysis, predictions, risk analysis, automations and corrective, maintenance and optimization recommendations.
- In many cases, the ASPs may provide general analytical software or software targeting specific industries or types of smart objects.

IoT Enable Technologies

- **EDGE DEVICES -** Not shown in our simplistic IoT ecosystem of ABOVE Figureis exactly how the smart objects embedded with sensors connect via the Internet to the various service provider systems.
- The answer is via “edge devices” – any device such as a router, routing switch, integrated access device (IAD), multiplexer, or metropolitan area network (MAN) and wide area network (WAN) access device which provides an entry point from the global, public Internet into an ASP’s or other enterprise’s private network.
- In Industry 4.0, these edge devices are becoming smarter at processing data before such data even reaches an enterprise network’s backbone (i.e., its core devices and cloud data centers).
- For example, edge devices may translate between di erent network protocols, and provide first-hop security, initial quality of service (QoS) and access/ distribution policy functionality.

IoT Enable Technologies

- **Embedded System -** A microcontroller-based, software-driven, reliable, real- time control system, designed to perform a specific task..
- It can be thought of as a computer hardware system having software embedded in it. An embedded system can be either an independent system or a part of a large system.

Industrial Robots
GPS
GPS Receivers
Digital Cameras
DVD Players
Embedded Systems
Wireless Routers
MP3 Players
Set top Boxes
Gaming Consoles
Photocopiers
Microwave Ovens

IoT Enable Technologies

- **Key Components of Embedded System**
  - Microprocessor or micro controller
  - Memory (RAM, ROM ect.)
  - Storage (Flash Memory)
  - Networking units(Ethernet, Wifi adaptors)
  - I/O units (Keyboard, display ect)

- Some Embedded systems have
  - DSP(Digital Signal Processor)
  - Graphics Processor
  - App Specific Processor

- Embedded systems run embedded OS
- Ex: RTOS(Real Time OS)(like symbian, Vxworks , Windows embedded compact ect.)

Summary

- Introduction of Unit
- Definition and IoT Chrematistics
- Physical Design of IoT- IoT Protocols
- Logical Design of IoT - IoT Functional Blocks
- IoT Communication Models
- IoT Communication APIs
- IoT Enable Technologies – Wireless Sensor Networks, Cloud Computing, Big Data Analytics, Communication Protocol, Embedded System
- Conclusion of the Unit