Internet of Things-A review of the Architecture of Networking and Communication Protocols

Jayanto Mitra

Student
Electronics Engineering Department
Vishwakarma Intitute of Technology, Pune-411037
Pune, India

***Abstract*— Internet of things aims to connect a variety of objects with one another so that the objects can perform certain action based on the inputs received. This paper aims to review a few research advancements that will be used extensively by developers in IoT projects. A logical understanding of the various communications and networking architecture involved in IoT is reviewed here.**

***Index Terms*—Internet of Things; Smart Things; IoT Communication protocols; Sensors; Wi-Fi, Zig-Bee, & Bluetooth, Automation.**

I. Introduction

Internet is nothing but the global inter-network of many different Computers and computing devices. Basically IOT expands the scope of internet.

IoT is going to interconnect different things that physical objects that we see around us, the different objects such as the lighting system in a room, the lights, the fans, the air conditioners and anything and everything including things such as the microwave oven, the refrigerator and so on so forth and not only in our homes, but also in our businesses such as internet working different machines, internetworking different equipments and so on. So, each and everything that we see around us that we use at our home in businesses, in workplaces, everything being internet worked. Advanced levels of services can be offered with the help of IoT based technology.

IOT will be connect devices, machines, tools to the internet by the means of wireless communication network technologies .Unification such as low powered embedded systems, cloud computing, big-data, machine learning and networking.

we are going to go through, they are going to cover the different challenges and how there are different tools that are available in order to, what are the different tools that are available in order to address these different challenges.

IOT enabling technologies that have been recognized:

- RFID
- NANOTECHNOLOGY
- SENSORS
- SMART NETWORK

Abbreviation: IoT – Internet of Things

II. Internet of Things Communication Protocols

For IoT, there has to be some sort of handshaking mechanism to be devised, concepts called Unique Building Blocks, such as the IoT LAN, IoT WAN, IoT Node, IoT Gateway, IoT Proxy and so on[1].

- IOT LAN is very similar to the traditional LAN -this is for Local Short Range communication may be building wide or campus wide and so on
- IoT WAN – is basically internetworking of two different LANs, connecting network segments
- IoT Node – connectivity of different nodes inside a LAN, sometimes directly connected to the internet through the WAN.
- IoT Gateway – It is connectivity of several LANs connected together through the WAN using the Gateways [1].
- IoT Proxy – Performs active application layer functions between IoT nodes and other entities.

III. IOT Components

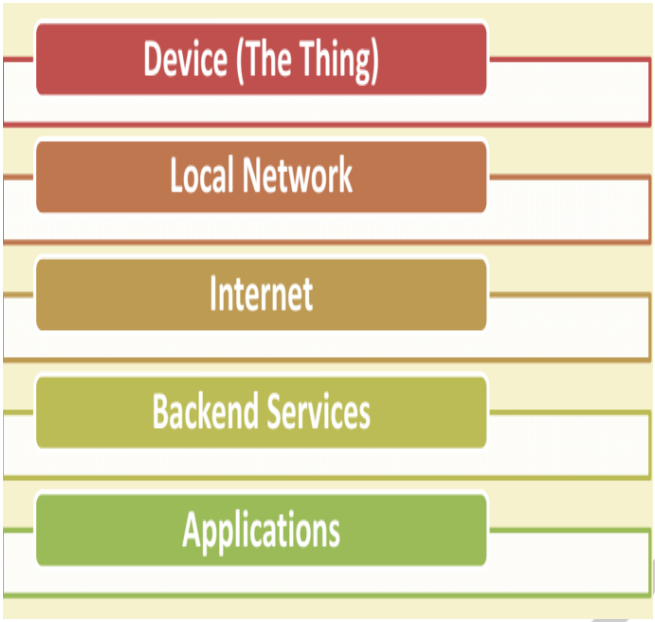

Figure 1.IOT Components

Functional components of IoT

i)Component for interaction and communication with other IoT devices.
ii)Component for processing and analysis of operations
(iii) Component for Internet interaction
(iv)Components for handling Web services of applications
(v)Component to integrate application services
(vi)User interface to access IoT.[2]

In terms of the functional components of IoT, one of the very important things is basically interaction. Interaction not only with the physical environment by this different sensors but also interaction and communication with the different devices, that means, the different nodes in the IoT network. Then comes the processing and analysis of the different functions and the subsequent operations that take place.

The third functional component is basically the interaction typically with the internet and because you know at present most of the times, the most of the IoT implementations are still using the internet. So, it is all you know internet powered IoT implementations. So, internet interaction is one of the very important components of building IoT. Then, we have the web services, web services machine to machine communication and so on.

Implementation Example

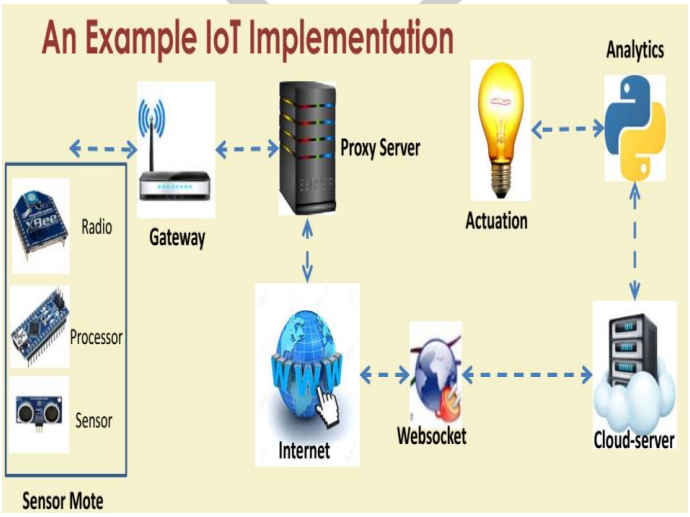

Fig. 2. IoT implementation

Application

The IoT implementation can be done to achieve different application needs. Figure [2] shows that we have different sensors, processors and radio. It refitted to each of these devices or the sensor nodes or the sensor motes or the IoT motes . These motes, they talk to one another, but these different sensor nodes, they are basically within the jurisdiction or the domain of the gateway. The gateway is basically tasked to assign different locally unique addresses to these different nodes, to these different IoT nodes and the gateway basically takes care of the local addressing within that particular local area network. So, from that point, all the data can flow through a proxy server if internet access is required. It will go through the internet, then a web socket and from the web socket, it goes through a cloud server. That means, this is where lot of analytics and backend processing takes place and based on that the actuation based on the analytics and the inference that are drawn from the sensed data actuation of different devices can take place[3]

IV. IOT Interdependencies: Another perspective of IOT

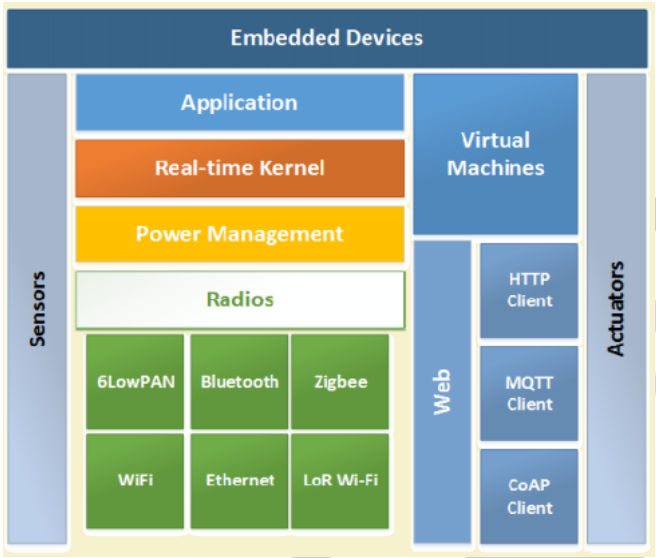

Figure 3.

So, if we look at IoT from another prospective or we have sensors, we have actuators and a bunch of figure. So, this is basically the entire span of these different embedded devices. So, the sensors basically sense the data and that data is serving the application requirements and then, we have an operating system and a power management unit which does things like duty cycling of the sensors, how much the sensors you know how much time we are going to be active or how much time they are going to be in the sleep state, how to power them because these are very small, very resource staved sensors. So, basically the power unit in these sensor nodes, these are very small in size. So, basically consequently what happens is these embedded devices; they themselves are very resource starved[4].

We have a very power management unit which basically takes care of power management as a whole. It takes care of the following major questions in the system.

i. How much power is required, for how long it is going to power,
ii. what are the ways to harvest energy if at all it can be harvested, and
iii. how much power consumption is going to take place at different points of time, can it be optimized different points of operation on and so on and so forth.

There after we have these different radios involving bluetooth, Zigbee, 6Low Pan, Wi-Fi, Ethernet and low range Wi-Fi.

These are the different radios that can help in communicating the data that is sensed onward to other nodes. These technologies have evolved over time and are a nice fit for the IoT communication. These basically different radio technologies can help for the communication purpose. Alongside we also have things like virtual machines which take care of the virtualization of the nodes, we have the web, and there are different things like http client, MQTT client, CoAP client. We have different application level protocols that are used for functioning of these different IoT devices and finally the actuator verticals. We have the sensors, we have different applications operating system, power management, radios, virtual machines web and then, we have these actuators all together which forms the embedded systems, the embedded devices.

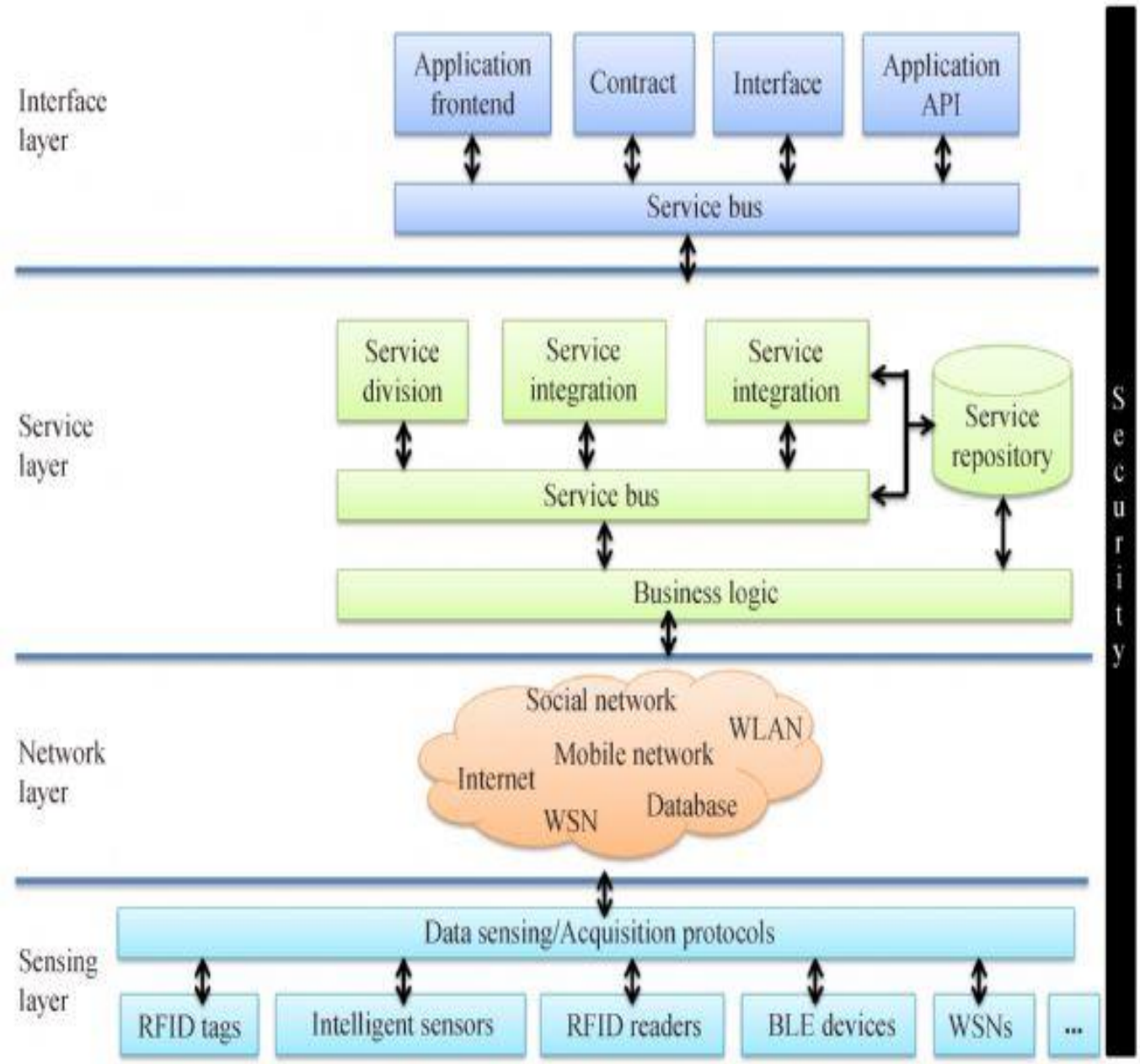

Figure 4. IoT service oriented architecture

In figure [4] we see that in IoT we have at different layers, like the sensing layer, the network layer, the service layer and the interface layer. sensing layer basically takes care of sensing through different RFID tags sensors and so on and, data are sensed are acquired and so on are sent to the next layer higher up which is the network layer. The network layer basically serves sensor networks, social networks, different other networks and data bases, internet and so on. That is the network layer. We have the service layer which deals mostly with the service delivery such as service division service integration, service repository, service logic, business logic and so on. So, all these different things have evolved with the offering of the services to support the different business functions. Then, we have the interface layer, we have the application frontend, we have a contract interface and application APIS. So, this becomes the interface layer and when we have the security issues which basically span all these different layer horizontals.

Lately, there is a move towards another system architecture, namely, fog computing[19- 21] where the sensors and network gateways do a part of the data processing and analytics. A fog architecture [22] presents a layered approach, which inserts monitoring, pre-processing, storage, and security layers between the physical and transport layers. The monitoring layer monitors power, resources, responses, and services. The pre-processing layer performs filtering, processing, and analytics of sensor data. The temporary storage layer provides storage functionalities such as data replication, distribution, and storage. Finally, the security layer performs encryption/decryption and ensures data integrity and privacy. Monitoring and pre-processing are done on edge of the network before sending data to the cloud.

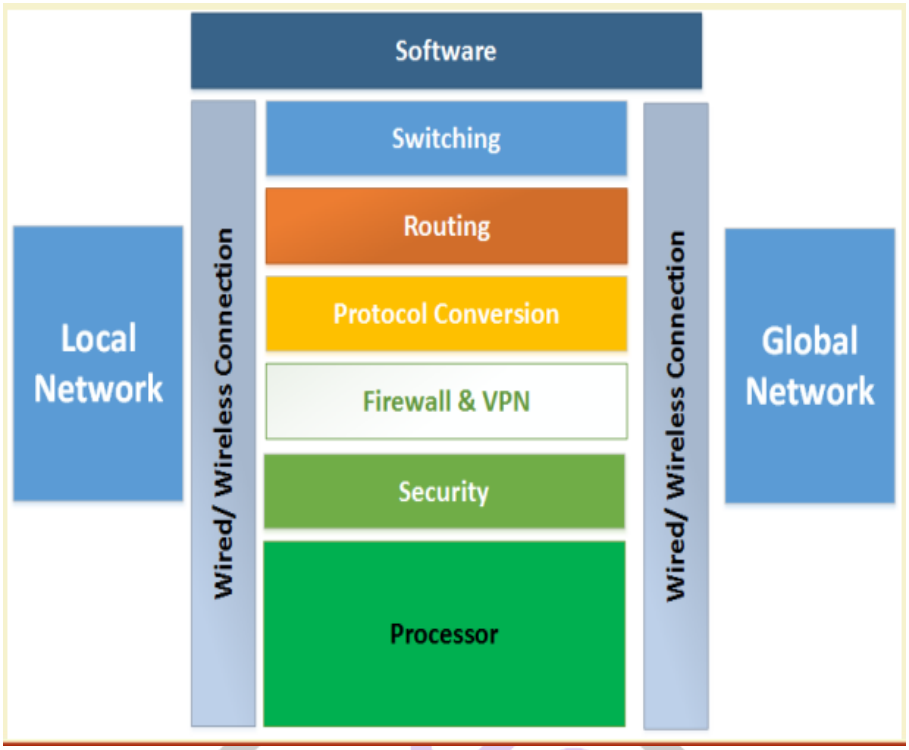

Figure 5

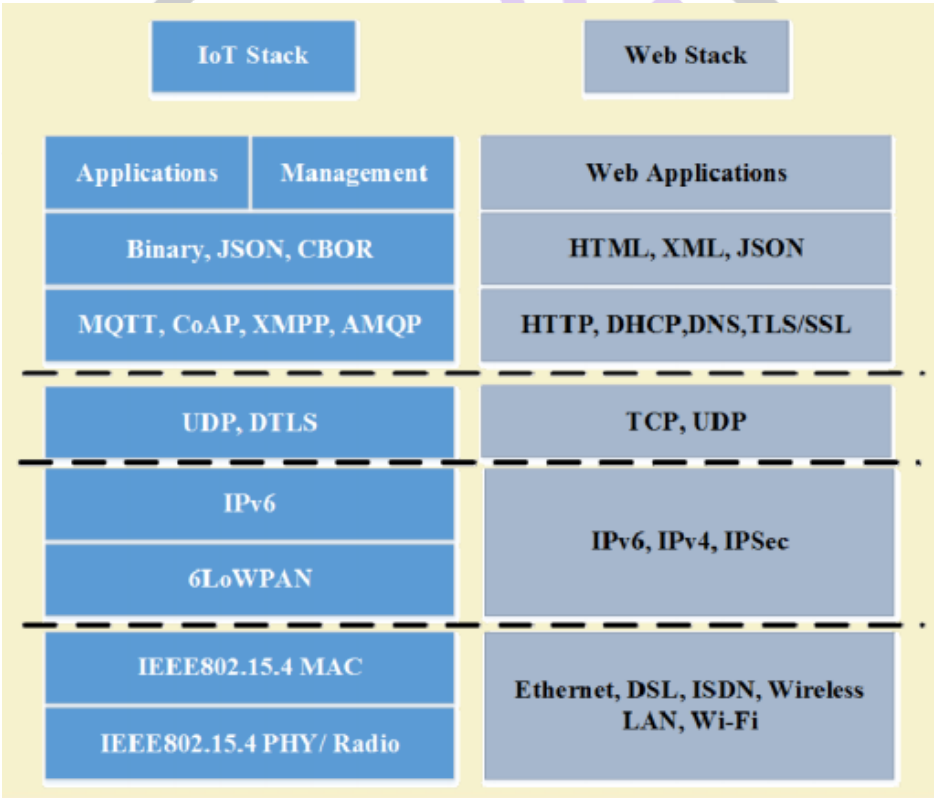

Figure 6

If we try to make a side by side comparison between the IoT stack and web stack, we will see that more or less the application layers remain the same for both IoT as well as the web. conceptually these application layers, these different layers, communication layers and application layers that communication layers remaining the same between IoT and web, but what is different is that we have a new set of protocols that are used over here. So, the new set of protocols and additionally in IoT, unlike in the case of web things such as different types of management, management of the network, management of the power, management of different other resources, these are all additionally taken care of in the IoT node, in the IoT stack which is not available in the case of the web and this is very much required because you know in the case of IoT, we are talking about heavily resource constraint nodes

and this heavily resource constraint nodes basically require management, network management in terms of energy, in terms of processing and in terms of data.

Figure 7

There are different key technologies that basically help IoT survive. We have the future internet knowledge aggregation obtained through data assignation, data collection processing and analysis. Then, we have different standards, we have sensor networks, we have communication, we have cloud computing, we have discovery services, nano electronics, embedded systems, software system integration and last, but not the least what is this over here on top is the security on privacy issues. security on privacy issues are per amount in IoT because there are heavy concerns we are dealing with resource constraint nodes with communication constraints, bandwidth constraints, processing constraints, energy constraints and. So, these nodes become very much valuable to different type of attacks, different types of security breaches and also because IoT systems are very much detailed intensive, there is lot of information that flows through the network as a consequence of which the privacy of the individuals of the organizations might be at stake. So, security and privacy and trust also which is not mentioned over here, these are very much important to power IoT technologies.

V. IoT communication Present Day

There are different types of challenges, securities, scalability, energy efficiency, bandwidth management, interfacing interoperability.

- Functionality Based IoT protocol organisation

Connectivity (6LoPAN, RPL).

- Communication/Transport(Wi-Fi, Bluetooth, LPWAN)
- Data Protocol(MQTT, CoAP, AMQP, Web Socket, Node)

1. MQTT

Message Queue Telemetry Transport. So, it is an ISO standard which is based on publish subscribe model. So, basically you know what happens is there is some kind of publishing of the data and then, the fetching of the data by the subscribers. So, this is how this publish subscribe model works and MQTT basically what it has done is, this publish subscribe model, it has been made lightweight through the use of this protocol, so that this lightweight protocol can be used in conjunction with the TCP IP protocol suit. This is what MQTT supports. MQTT going back to the history was introduced in 1999 by IBM and is standardized in 2013 by Oasis. It has standardized in the year 2013. So, this particular protocol does couple of things. One is offering connectivity between different embedded devices between the applications and then, middle ware of one device and network and communication on the other side of the device. So, we have connectivity between applications and middle ware for one side and the networks and communication on the other. This is what MQTT does[3].

MQTT there are three concepts that are involved. The first we are going to go through is the concept of a message broker. The concept of a message broker basically serves like a broker which takes control of publishing of the messages and subscription of the messages.

There is a concept of topic and this is what the client is subscribed and based on the updates. The data are sent to the clients by the message broker, this data are distributed by the message broker to the clients who have subscribed to the services. This is design for remote connections limited bandwidth environments and MQTT, basically the advantage is that it provides every small code foot print. The different components of MQTT are as follows. We have three principle components. The publishers which involve the different sensors, the subscribers and that means, those entities, those applications, those units that are interested in the data that is published by the sensors. It is the broker in between which helps the publishers and the subscribers connect to one another and also help in classifying the sensor data into different topics. MQTT there are a few different methods. One is connect, the second is disconnect, subscribe, unsubscribe and publish. So, basically the connect method helps to connect with the server, helps to connects this device with the server. Then, disconnect is the opposite. Whenever it is no longer required to be to remain connected, the disconnect method helps in disconnecting from the server from TCP IP service offerings and so on. And then comes the subscribe which is basically subscribing to the services and unsubscribe is the opposite that whenever it is no longer required to continue with getting the different data offerings, the data services and so on.

The unsubscribe method can be executed and then, we have the publish method which is basically publishing data for maybe you know publishing the data from these different sensors or these different devices to the broker for it to be fetched by the different application clients[3].

This is illustrated in the fig.8

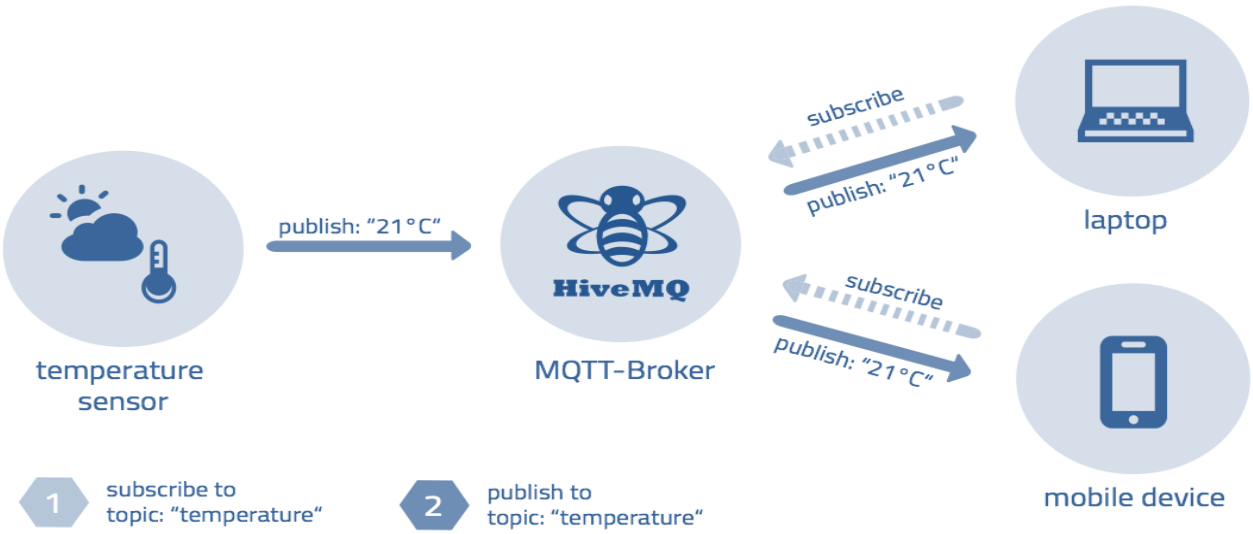

Figure 8

The protocol uses a **publish/subscribe** architecture (HTTP uses a request/response paradigm). Publish/subscribe is **event-driven** and enables messages to be pushed to clients. The central **communication point is the MQTT broker**, which is in charge of dispatching all messages between the senders and the rightful receivers. Each client that publishes a message to the broker, includes a **topic** into the message. The **topic is the routing information for the broker.** Each client that wants to receive messages subscribes to a certain topic and the broker delivers all messages with the matching topic to the client. Therefore the clients don't have to know each other. They only communicate over the topic. This architecture enables highly scalable solutions without dependencies between the data producers and the data consumers[4].

MQTT Topics

A topic is a **simple string** that can have more hierarchy levels, which are separated by a slash[4].

A sample topic for sending temperature data of the living room could be *house/living-room/temperature*. On one hand the client (e.g. mobile device) can subscribe to the exact topic or on the other hand, it can use a **wildcard**. Each client that wants to receive the messages subscribes to a certain topic and the broker delivers all the messages with in the matching topic to the client. So, essentially what is happening is the clients, they do not have to know each other and what is required is they only need to communicate with each other over the topic. So, they do not know about each other and this architecture basically appears to be a scalable architecture with a scalable solution. There is not much dependency between the producers and the consumers of the data and that is where MQTT is very popular[3].

The subscription to *house/+/temperature* would result in all messages sent to the previously mentioned topic *house/living-room/temperature*, as well as any topic with an arbitrary value in the place of living room, such as

house/kitchen/temperature.
The plus sign is a **single level wild card** and only allows arbitrary values for one hierarchy. If more than one level needs to be subscribed, such as, the entire sub-tree, there is also a **multilevel wildcard** (*#*). It allows us to subscribe to all underlying hierarchy. levels. For example *house/#* is subscribing to all topics beginning with *house*.

Applications

Face book Messenger uses MQTT for online chat.
Amazon Web Services use Amazon IoT with MQTT.
Microsoft Azure IoT Hub uses MQTT as its main protocol for telemetry messages.
The **EVRYTHNG IoT platform** uses MQTT as an M2M protocol for millions of connected products.
Adafruit launched a free MQTT cloud service for IoT experimenters called Adafruit IO.[17]

The secured version of MQTT which is called the secure MTTM, or SMQTT in short. So, this is known in both these ways and this actually to me is quite similar in notion to http and https, the secure http. So, we have http secure http, we have MQTT secure MQTT. So, secure MQTT is an extension of MQTT[5].So, basically it is an extension of the MQTT by using different security features such as encryption and so on. The advantage of such encryption is the broadcast encryption feature in which one message is encrypted and delivered to multiple other nodes which is quite common in IoT applications. In general, the algorithm consists of four main stages i.e. the setup stage, the encryption stage, the publish stage and the decryption stage. In the setup phase, the subscribers and publishers register themselves to the broker and get a master secret key according to their developer's choice of key generation algorithm. When the data is published, it is encrypted and published by the broker which sends it to the subscribers, which is finally decrypted at the subscriber end having the same master secret key. The key generation and encryption algorithms are not standardized. SMQTT is proposed only to enhance MQTT security features.

2. CoAP – Constrained Application Protocol.
CoAP is a **Web transfer protocol** for use with constrained nodes and networks[6].
It was basically **designed for Machine to Machine** (M2M) applications such as smart energy and building automation. It is based on **Request-Response model** between end-points.Client-Server interaction is **asynchronous over a datagram oriented transport protocol** such as UDP.

The Constrained Application Protocol (CoAP) is a session layer protocol designed by IETF Constrained RESTful Environment (CoRE) working group to provide lightweight RESTful (HTTP) interface.
Representational State Transfer (REST) is the standard interface between HTTP client and servers[5].
Lightweight applications such as those in IoT, could result in significant overhead and power consumption by REST.
CoAP is designed to enable low-power sensors to use RESTful services while meeting their power constraints[5].

Built over UDP, instead of TCP (which is commonly used with HTTP) and has a light mechanism to provide reliability[6]. CoAP architecture is divided into two main sub-layers:
1.Messaging
2.Request/response.
The messaging sub-layer is responsible for reliability and duplication of messages, while the request/response sub-layer is responsible for communication.
CoAP has four messaging modes:
1.Confirmable
2.Non-confirmable
3.Piggyback
4.Separate

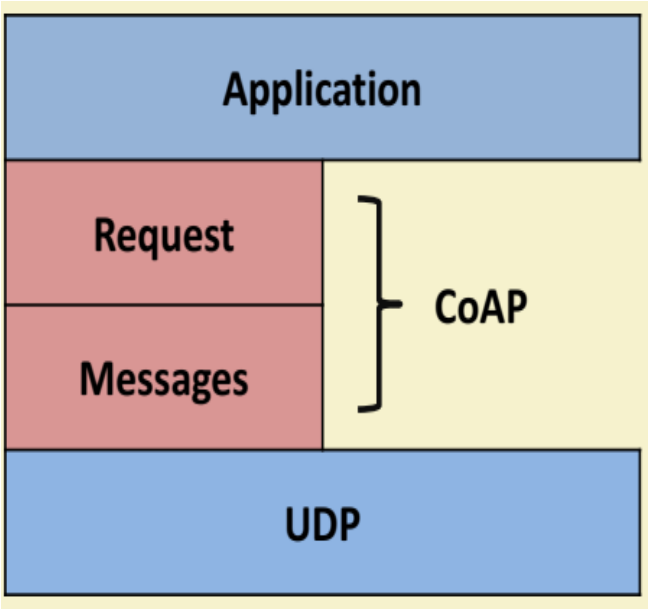

Figure.9

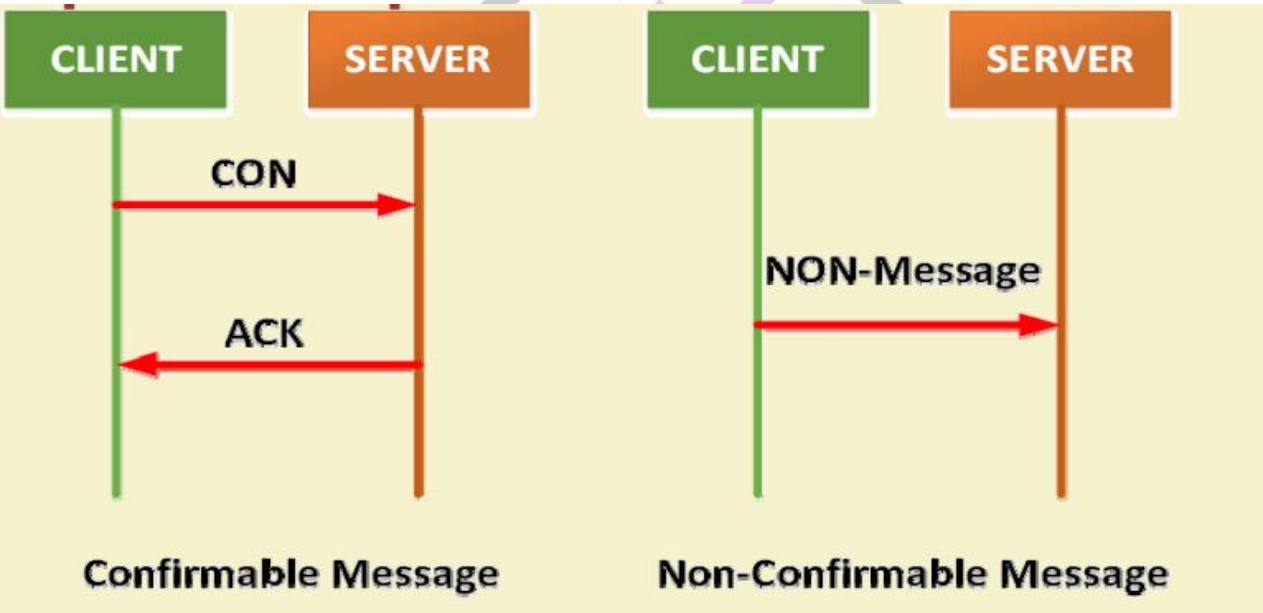

Figure 10

In the protocol stack in terms of the layered architecture, CoAP is a protocol of the session layer[5].

We can think of CoAP to be merged with the application layer, but if session layer is considered, it is a protocol of the session layer. So, session layer means that it lies between the transport layer and the application layer. So, at the transport layer, we have the UDP protocol and different applications being run in the application layer and CoAP basically sits in between. So, we have two sub-layers, one is the request response and the other one is the messages. Messages is mostly concerned about the reliability in sharing, reliability of the network, reliability in communication whereas, request response is more to do with the exact communication in terms of sending a request and getting a response back[5].

Confirmable and non-confirmable modes represent the reliable and unreliable transmissions, respectively, while the other modes are used for request/response[6].

Piggyback is used for client/server direct communication where the server sends its response directly after receiving the message, i.e., within the acknowledgment message[7].

On the other hand, the separate mode is used when the server response comes in a message separate from the acknowledgment, and may take some time to be sent by the server. Similar to HTTP, CoAP utilizes GET, PUT, PUSH, DELETE messages requests to retrieve, create, update, and delete, respectively[7].

CoAP Request-Response Model

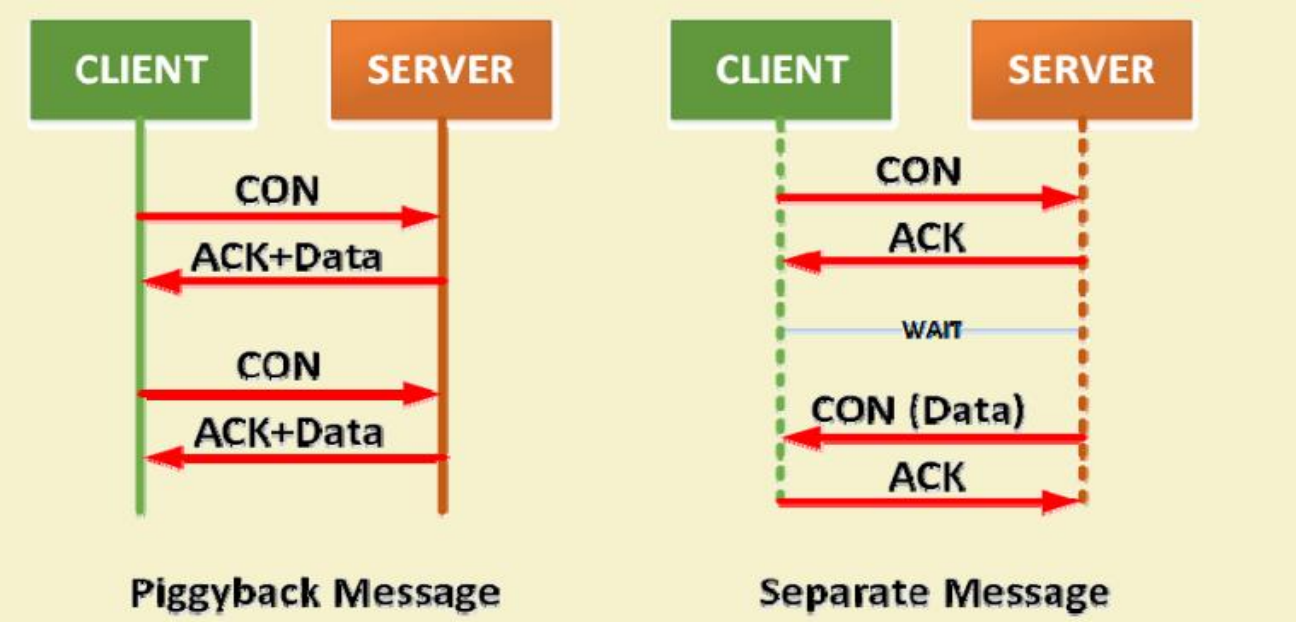

Figure 11

Features

Reduced overheads and parsing complexity.

URL and content-type support.

Support for the discovery of resources provided by known CoAP services. Simple subscription for a resource, and resulting push notifications. Simple caching based on maximum message age[5].

3. XMPP – Extensible Messaging and Presence Protocol.

XMPP is a communication protocol for message-oriented middleware based on XML (Extensible Mark-up Language). It provides a real-time exchange of structured data. It is an open standard protocol. XMPP uses a **client-server architecture[8]**. As the model is **decentralized**, no central server is required. XMPP provides for the **discovery of services** residing locally or across a network, and the **availability information** of these services. Well-suited for cloud computing where virtual machines, networks, and firewalls would otherwise present obstacles to alternative service discovery and presence-based solutions [8].Open means to support machine-to-machine or peer-to-peer communications across a diverse set of networks.

Applications[9]

1. Publish-subscribe systems
2. Signalling for VoIP
3. Video
4. File transfer
5. Gaming

VI. Communication Protocols

The following communication protocols have immediate importance to consumer and industrial IoTs:

1. IEEE 802.15.4
2. Zigbee
3. 6LoWPAN
4. Wireless HART
5. Z-Wave
6. ISA 100
7. Bluetooth
8. NFC
9. RFID

Features of IEEE 802.15.4

802.15.54 which is IEEE standard and this basically is used for forming Wireless Personal Area Network.
It is a well-known standard for low data-rate WPAN. Developed for low-data-rate monitoring and control applications and extended-life low-power-consumption uses. This standard uses only the first two layers (PHY, MAC) plus the logical link control (LLC) and service specific convergence sub-layer (SSCS) additions to communicate with all upper layers. Operates in the ISM band.[10]

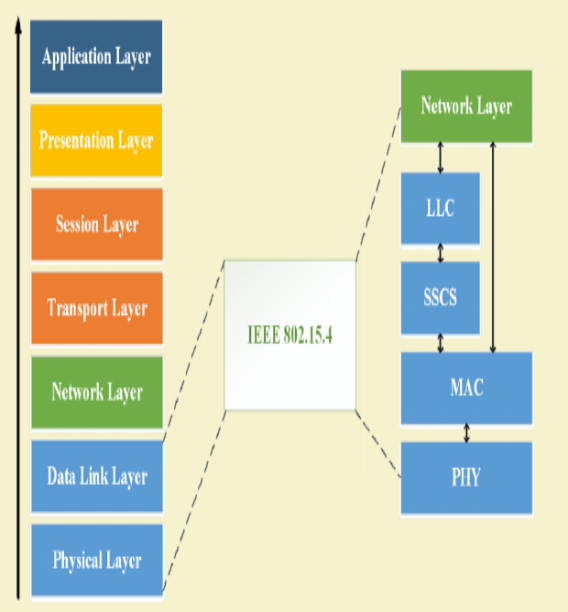

Figure 11

Uses direct sequence spread spectrum (DSSS) modulation. It is highly tolerant of noise and interference and offers link reliability improvement mechanisms. Low-speed versions use Binary Phase Shift Keying (BPSK). High data-rate versions use offset-quadrature phase-shift keying (O-QPSK).It uses carrier sense multiple access with collision avoidance (CSMA-CA)for channel access. Multiplexing allows multiple users or nodes interference-free access to the same channel at different times. Power consumption is minimized due to infrequently occurring very short packet transmissions with low duty cycle (<1%). The minimum power level defined is –3 dBm or 0.5 mW. Transmission, for most cases, is Line of Sight (LOS). Standard transmission range varies between 10m to 75m [13-15].

• **Full Function Device (FFD)** Figure [13]
• Can talk to all types of devices
• Supports full protocol
• **Reduced Function Device (RFD)**
• Can only talk to an FFD
• Lower power consumption
• Minimal CPU/RAM required

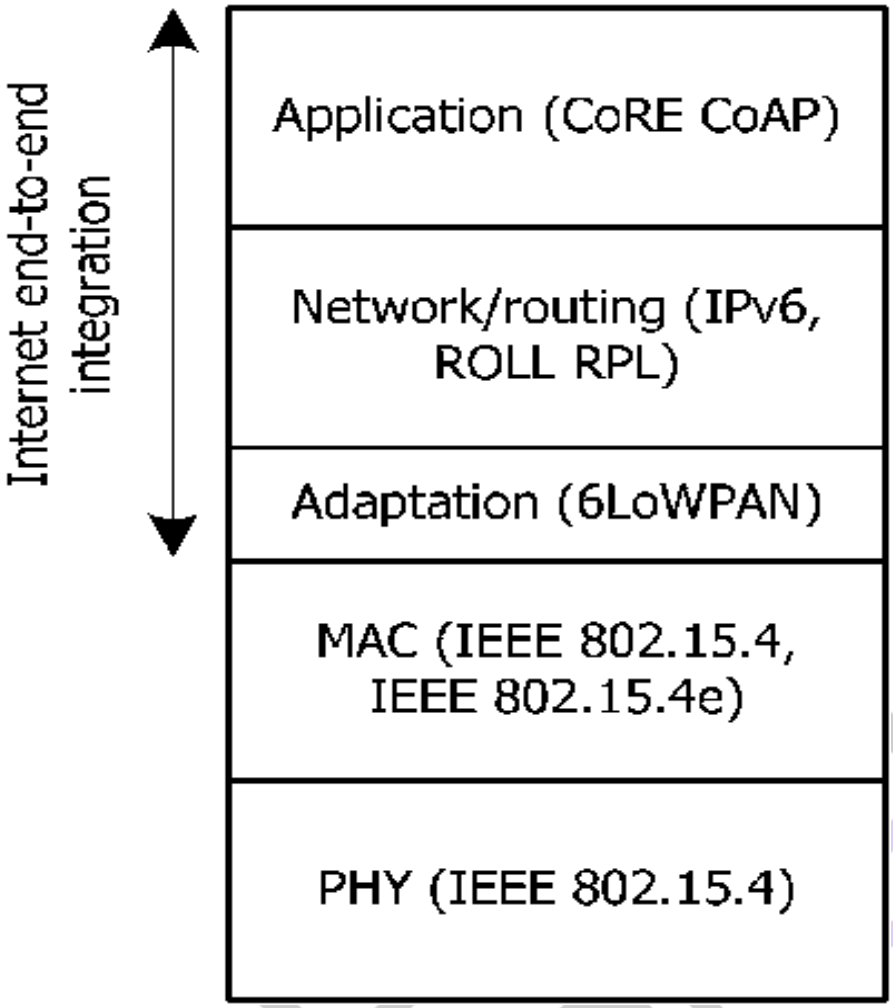

Figure 12

Best case transmission range achieved outdoors can be up to 1000m. Networking topologies defined are -- Star, and Mesh.[10]

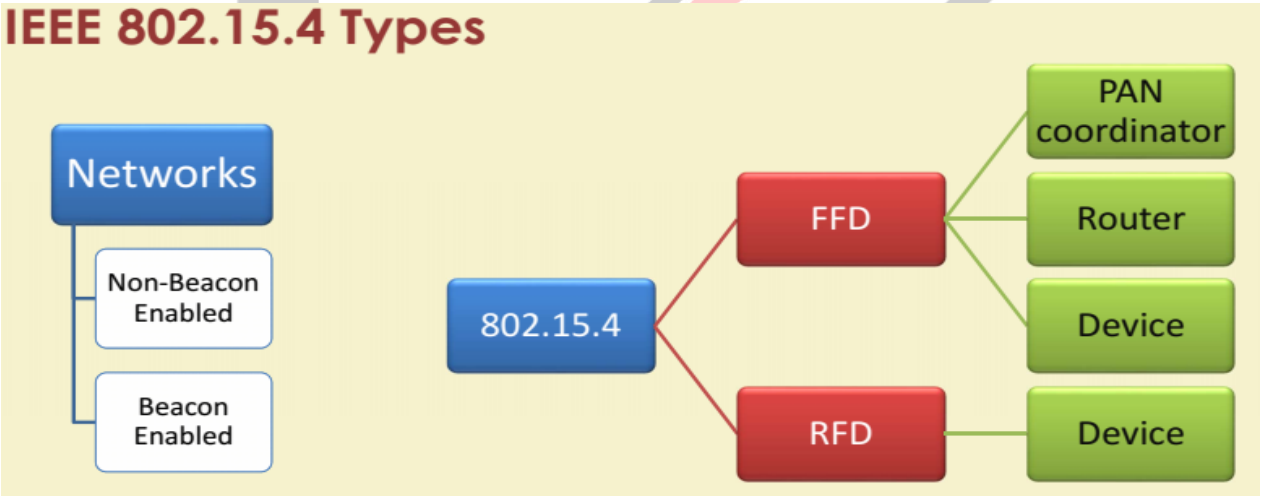

Figure 13

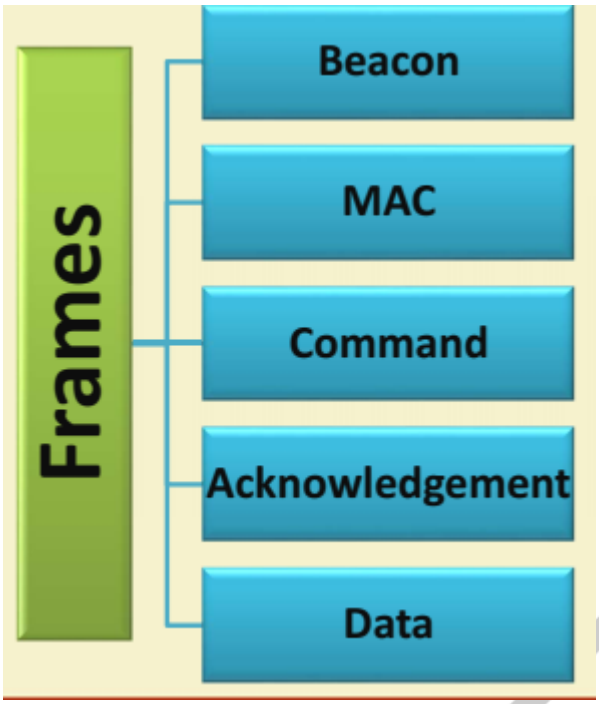

Figure 14

Beacon Enabled Networks
• Periodic transmission of beacon messages.
• Data-frames sent via Slotted CSMA/CA with a super frame structure managed by PAN coordinator.
• Beacons used for synchronization & association of other nodes with the coordinator.
• Scope of operation spans the whole network.

Non-Beacon Enabled Networks
Data-frames sent via un-slotted CSMA/CA (Contention Based).Beacons used only for link layer discovery. Requires both source and destination IDs. As 802.15.4 is primarily, a mesh protocol, all protocol addressing must adhere to mesh configurations. De-centralized communication amongst nodes.

ZigBee
ZigBee is the most widely deployed enhancement of IEEE 802.15.4. The ZigBee protocol is defined by **layer 3 and above**. It works with the 802.15.4 layers 1 and 2. The standard uses layers 3 and 4 to define additional communication enhancements. These enhancements include authentication with valid nodes, encryption for security, and a data routing and forwarding capability that enables mesh networking. The most popular use of ZigBee is wireless sensor networks using the mesh topology[13-15].

Standardisation
IoT standards is a requirement, because IoT support interactions among many heterogeneous sources of data and many heterogeneous devices through the use of interfaces and data models to ensure a high degree of interoperability among diverse systems. The standardization bodies are addressing the issue of interoperable protocol stacks and open standards for the IoT[16]. This includes as well expending the HTTP, TCP, IP stack to the IoT-specific protocol stack. This is quite challenging considering the different wireless protocols like ZigBee, RFID, Bluetooth, BACnet 802.15.4e, 6LoWPAN, RPL, CoAP, AMQP and MQTT. Some of these protocols use different transport layers. HTTP relies on the Transmission Control Protocol (TCP) [18].

Figure 15

Figure 16[11]

ZigBee Types

ZigBee Router (ZR):
Capable of running applications, as well as relaying information between nodes connected to it.

ZigBee End Device (ZED):
It contains just enough functionality to talk to the parent node, and it cannot relay data from other devices.
This allows the node to be asleep a significant amount of the time thereby enhancing battery life.
Memory requirements and cost of ZEDs are quite low, as compared to ZR or ZC[12].

ZigBee Network Layer

The network layer uses Ad Hoc On-Demand Distance Vector (AODV) routing.
To find the final destination, the AODV broadcasts a route request to all its immediate neighbours.
The neighbours relay the same information to their neighbours, eventually spreading the request throughout the network. Upon discovery of the destination, a low-cost path is calculated and informed to the requesting device via unicast messaging.

Applications

1. Building automation.
2. Remote control (RF4CE or RF for consumer electronics).
3. Smart energy for home energy monitoring.
4. Health care for medical and fitness monitoring.
5. Home automation for control of smart homes.
6. Light Link for control of LED lighting.
7. Telecom services [10].

VII. INTERNET OF THINGS AND RELATED FUTURE TECHNOLOGIES & DEVELOPMENT.

Many new technologies are related to IoT to prove the integration of wired as well as wireless control, communication and IT technologies together which are responsible for connecting several subsystems and things which operate under a unified platform controlled and managed smartly.

1. Cloud computing:

The two worlds of Cloud and IoT have seen a rapid and independent evolution. These worlds are very different from each other, but their characteristics are often complementary in general, in which IoT can benefit from the virtually unlimited capabilities and resources of the cloud to compensate its technological constraints for example storage, processing, and communication[19]. Cloud can offer an effective solution for IoT service management and composition as well as for implementing applications and services that exploit the things or the data produced by them. On the other hand, the cloud can benefit from IoT by extending its scope to deal with real-world things in a more distributed and dynamic manner, and for delivering new services in a large number of real-life scenarios. In many instances, Cloud can provide the common layer between the
things and the applications, protecting all the complexity and functionalities required to implement the latter[20]. This will affect future application development, where information collection, processing, and synchromesh will generate new challenges, particularly in multi-cloud conditions or in a fog cloud. Cloud aids for IoT application to allowing data collection and data processing, in expanding to rapid setup and integration of new things, while sustaining low costs for deployment and for complicated data processing[20]. Cloud is the most suitable and cost-effective solution to deal with data produced by IoT and, in this regard, it generates new opportunities for data gathering, integration, and sharing with third parties. Once into Cloud, data can be treated as homogenous through robust-defined APIs, can be guarded by applying top level security, and can be immediately accessed and visualized from any place.

2. Shared Computing:

Shared computing uses groups of networked computers for the identical computational goal; shared Computing has several common issues with concurrent and parallel computing, as all these three falls in the scientific computing field. Nowadays, a large volume of shared computing technologies coupled with hardware virtualization, service oriented architecture, and autonomic and utility computing have started to cloud computing. Internet of Things with shared computing represents a vision in which the Internet extends into the real world embracing everyday objects.
Physical things are no longer disconnected from the virtual environment, but can be remotely managed and can act as physical access points to Internet services.

3. Fog Computing

Fog computing is similar to the edge computing in the cloud. In opposition to the cloud, fog platforms have been defined as compact computational architectures at the network's edge[21]. Features of such platforms reportedly include low latency, location awareness and control of wireless access. While edge computing or edge analytics may especially refer to performing analytics at things that are on, or close to, the network's point, a fog computing structure would deliver analytics on anything from the network core to the edge[22]. IoT may considerably likely be presented by fog computing in which computing, storage, control and networking power may remain anywhere along the architecture, either in data stations, the cloud, edge devices such as gateways or routers, edge devices itself such as a computer, or in sensors[22].

VIII. Security and Privacy Technologies Security and Privacy Technologies

security and privacy are probably the most challenging issues in the Internet of Things. The connected smart thing in the IoT which is able to communicate with other smart objects, facing new security and privacy problems. Like confidentiality, authenticity, and integrity of data sensed and exchanged by things. Though providing improvements in social efficiency it creates an array of new problems concerning breach of privacy and that information security. The various threats in the security of IoT is as below[16]

1. Front-end sensors and Equipment

- Unauthorized access to data
- Threats to Internet
- Denial of service attack
- Attack and privacy analysis of M2M or contract information
- Attack to availability of M2M or contract information

2. Network

- Unauthorized access to data
- Unauthorized access to service
- Steal or change the communication information
- Viruses or malware attack

3. Network security
- Back-end of IS systems
- Safety management of code resources
- Replacement of operators.

IX.Conclusion

In this paper, we presented Internet of Things architecture and research agenda. We have reviewed communication and networking technologies which is one of the most significant research areas in the evolution and implementation of IoT We identified several open issues related to the research agenda that need to be addressed by research community. In future, research on the IoTs will remain a hot issue, many problems are waiting for researchers to deal with.

References

[1]Li Da Xu, Wu He, and Shancang Li, "Internet of Things in Industries: A Survey ", IEEE Transactions on Industrial Informatics, Vol. 10, No. 4, Nov. 2014.

[2] O Vermesan, P. Friess**, "**Internet of Things – Converging Technologies for Smart Environments and Integrated Ecosystems", River Publishers, Series in Communications, 2013

[3] http://www.apdaga.com

[4] **"**MQTT 101 – How to Get Started with the lightweight IoT Protocol", HiveMQ (Online)

[5] Z. Shelby, K. Hartke, C. Bormann, "The Constrained Application Protocol (CoAP)", Internet Engineering Task Force (IETF), Standards Track, 2014

[6] Constrained Application Protocol", Wikipedia (Online)

[7] V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. Alonso-Zarate, "A survey on application layer protocols for the internet of things," Transaction on IoT and Cloud Computing, vol. 3, no. 1, pp. 11-17, 2015

[8] "XMPP", Wikipedia (Online)

[9] "XMPP: Technology Overview", XMPP.org (Online)

[10] L.Fenzel, "What's The Difference Between IEEE 802.15.4 And ZigBee Wireless?", Electronic Design (Online), Mar. 2013

[11] T. Agarwal, "ZigBee Wireless Technology Architecture and Applications", Electronics Projects Focus (Online)

[12] "Wireless Sensor Networks Research Group". Sensor-networks.org. 2010-04-15.

[13]"Wireless Sensor Networks Research Group". Sensor-networks.org. 2009-02-05.

[14] Harald Sundmaeker, Patrick Guillemin, Peter Friess, Sylvie Woelfflé**,** "*Vision and Challenges for Realising the Internet of Things"*, Luxembourg: Publications Office of the European Union, 2010.

[15] D. Jiang, and C. ShiWei, "*A Study of Information Security for M2M of IoT*," 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), 2010, pp.576-579.

[16] J. Sathish Kumar , Dhiren R. Patel , "*A Survey on Internet of Things: Security and Privacy Issues"*, International Journal of Computer Applications (0975 – 8887) Volume 90 – No 11, March 2014.

[17] https://io.adafruit.com/

[18] Ovidiu Vermesan, Peter Friess,"*Internet of Things Position Paper on Standardization for IoT technologies"*, EUROPEAN RESEARCH CLUSTER ON THE INTERNET OF THINGS January,2015.

[19] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing: a platform for internet of things and analytics," in Big Data and Internet ofThings: A Road Map for Smart Environments, pp. 169– 186, Springer, Berlin, Germany, 2014.

[20] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in Proceedings of the 1st ACM MCC Workshop on Mobile Cloud Computing, pp. 13–16, 2012.

[21] I. Stojmenovic and S. Wen, "The fog computing paradigm: scenarios and security issues," in Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS '14), pp. 1–8, IEEE, Warsaw, Poland, September 2014.

[22] M. Aazam and E.-N. Huh, "Fog computing and smart gateway based communication for cloud of things," in Proceedings of the 2nd IEEE International Conference on Future Internet of Things and Cloud (FiCloud '14), pp. 464–470,Barcelona, Spain, August