Connectivity of the Internet of Things

Communication Protocols for Internet of Things

Connectivity, it is one of the main things to keep in mind while developing any Internet-of-Things (IoT) project.

The first few questions that pop up in my head when I embark on any new IoT project are:

- How do I want it to be connected?
- Do I have any power or range constraints?
- What would be my data rate?
- What network infrastructures are currently available?

I am sure a lot of you would have the same questions when starting any IoT project. Sometimes you have an idea about what communication protocol you want to use, but it doesn't hurt to do your research and make sure that would be the most suitable for your application.

Fortunately, there are a bunch of network infrastructures and communication protocols available. Unfortunately, there are so many that they might render you confused.

In this tutorial, we will discuss all (well, most) of the popular **communication protocols** and how to pick the most suitable one for your project. We will also go into detail about the pros and cons of each:

1. WiFi
2. Thread
3. ZigBee
4. Bluetooth
5. RFID and NFC

This is not the full list of connection types, but these will help you get started with most any IoT project.

Network Topology

It is important to understand the various network protocols we will mention.

Network topology is the way in which various elements are arranged in any network. It defines physically or logically the structure of the network. I like to view it as a pictorial representation of the network elements and how data moves through them.

Again, there are a bunch of possible network topologies, but we will keep our scope limited to the ones you would see most often when dealing with IoT-based communication protocols:

- Point to Point (P2P)
- Star
- Mesh
- Hybrid

Point to Point

P2P is the simplest topology and has a permanent link between two endpoints. The simplest example of P2P would be the connection in the paper cup-and-string telephones that you might have had fun with as kids, where two nodes (or endpoints) have a dedicated channel for communication. Using switching technologies, P2P can be set up dynamically. Switched P2P topology was the basis of the early telephony.

Star

In the star network configuration, every node (endpoint) connects to a single central device. The nodes cannot directly communicate with each other; they communicate through the central device. The central device acts as a server, whereas the nodes act as the clients. This is one of the most common configurations and one of the easiest to set up. It is simple to add and remove devices without disrupting the network. The biggest challenge with this kind of network is it has a single point of failure (i.e., the central computer); if the central computer fails, the network fails.

Mesh

Mesh is the type of network where each node is connected to every other node. A mesh network provides a high amount of redundancy when it comes to network links. Even if one link fails, the nodes can communicate using another link. This is not the most commonly used network topology for obvious reasons of increased costs to establish the redundant links and the complicated nature of the network.

Hybrid

Hybrid networks, as the name suggests, are combinations of two or more basic network topologies. It could be a star-mesh network or a star-ring network. Hybrid networks prove to be more flexible and reliable, as they come with the best of both worlds. But at the same time, they have increased complexity, which makes them expensive to set up and difficult to manage. However, hybrids have their benefits when we require a network with the capabilities of more than one network topology.

Infrastructure and Ad-hoc Networks

Since we are going to talk about wireless networks today, we need to discuss the two basic modes (also referred to as topologies) in which wireless networks operate.

Infrastructure mode is when the wireless network requires a physical structure to support it. This essentially means there should be a medium handling the network functions, creating an infrastructure around which the network sustains.

It performs these typical **functions**:

- Providing access to other networks
- Forwarding
- Medium access control

In infrastructure-based wireless networks, the communication takes place between the wireless nodes (i.e., endpoints in the network such as your computer, your phone, etc.) and the access points (i.e., the router) only.

There can be more than one access point on the same network handling different wireless nodes.

A typical example of an infrastructure network would be cellular phone networks. They have to have a set infrastructure (i.e., network towers) to function.

When to use an **infrastructure network**:

- If you can easily add more access points to boost the range
- If you want to set up a more permanent network
- If you will need to bridge to other types of networks (e.g., you can connect to a wired network if required)

The one major downfall with infrastructure networks is that they are costly and time consuming to set up once. So, if you need your device to operate in remote areas where the infrastructure is weak or nonexistent, you cannot rely on infrastructure networks.

Ad-hoc wireless networks, on the other hand, do not require a set infrastructure to work. In ad-hoc networks, each node can communicate with other nodes, so no access point that provides access control is required.

Whereas the routing in infrastructure networks is taken care of by the access point, in ad-hoc networks the nodes in the network take care of **routing**.

Routing is to find the best possible path between the source and destination nodes to transfer data.

All the individual nodes in an ad-hoc network maintain a routing table, which contains the information about the other nodes. As the nature of the ad-hoc network is dynamic, this results in ever-changing router tables. One important thing to note is that an ad-hoc network is asymmetric by nature, meaning the path of data upload and download between two nodes in the network may be different.

A typical example of an ad-hoc network is connecting two or more laptops (or other supported devices) to each other directly without any central access point, either wirelessly or using a cable.

When to use an **ad-hoc network**:

- If you want to quickly set up a peer-to-peer (P2P) network between two devices
- When creating a quick temporary network
- If there is no network infrastructure set up in the area (ad-hoc is the only network mode that can be used in areas like this)

As the routing is handled by each node in the network, this uses more resources; as the number of devices connected in an ad-hoc network increases, the network interference increases, which may lead to slower networks.

WiFi

WiFi is one of the most common communication protocols. You probably couldn't imagine your life without it. From the comfort of our homes to classrooms, cafés and airports, we see WiFi everywhere.

WiFi essentially uses an infrastructure network, which additionally supports ad-hoc networking in infrastructure mode.

Infrastructure mode of wireless communication provides a bridge to other networks, medium access control and forwarding. The network handling functions are placed into the access point (router), and the clients can remain simple (in context of network).

Also, WiFi is a star-based network. The communication goes from wireless nodes (devices) to a wireless access point (router or network controller).

The current standard being used is 802.11ac, which was released in 2013, although the version 802.11n, which was released in 2009, is still prevalent. The 802.11ac offers speeds up to 800Mbit/s whereas 802.11n offered up to 150Mbits/s.

You might have also seen devices that have even older standards of 802.11a/b/g, which are now called legacy devices. However, since WiFi has downward compatibility, old devices continue to work with devices that have new standards.

The range of your device's WiFi depends on a few factors:

- Which WiFi standard the device is running. The latest standards obviously offer more range than the older versions.
- Physical obstructions like walls also play a critical role in determining the range. Therefore, in open spaces the range of the WiFi network would be more than in enclosed spaces with walls and other interfering objects.

To address the weakness of WiFi over other low-powered technologies, an activity was started to standardize low-power WiFi (IEEE 802.11ah). The development of this protocol is being worked on, but its worldwide adaptation is doubtful. One of the main reasons is this is not backward compatible to the existing 802.11bgn networks.

Advantages of WiFi:

- WiFi has a decent range coverage and can penetrate walls and other obstacles in the way.
- Adding and removing devices in a WiFi network is a piece of cake.

Disadvantages of WiFi:

- Obviously, the lack of wires comes at a cost of lower bandwidth. The radio waves of the network might interfere with other equipment.
- Most importantly, security of WiFi is weaker than its wired counterparts

Now thinking about your project, WiFi is ideal when we want to set up a quick connection between our device (should be WiFi compatible) and the internet. WiFi is designed around the goal of keeping its power consumption limited, so you can run your project on a dedicated battery as well. WiFi should be used when you do not care much about how and when exactly your device should connect and communicate with your server and all you are looking for is hassle-free connection to the internet.

Thread

Thread is an open standard for reliable, cost-effective, low-power, wireless D2D (device to device) communication. It was designed specifically for connected home applications.

It came into existence in 2014, when the Thread Group was formed. It now has big organizations like Google, Samsung, Qualcomm, and ARM to design and develop the Thread protocol.

Nest uses Thread network for its Thermostat and Nest Cam products. Thread is designed while keeping in mind the home automation space – to have devices set up and connect easily, have low power consumption for longer battery life and be secure!

It is based on the standard 802.15.4 (6loWPAN) architecture. The best part about Nest is that it is an open protocol that has Internet Protocol version 6 (IPv6) built in.

Devices in Thread

The Thread protocol defines the three main types of devices in the network:

- Border routers
- Routers and router-eligible end devices
- Sleepy end devices

Border Router

Thread has a system with a special type of router called the border router, which provides connectivity from the 802.15.4 network to adjacent networks on different physical layers (e.g., WiFi, Ethernet). If one border router fails, another router in the network can assume the role of a border router, ensuring the robustness of the Thread protocol.

Routers

Routers, as the name suggests, provide routing services to the network devices. They are used in commissioning new devices to the network. Generally routers are always active, but they can downgrade to become REEDs (Router-Eligible End Devices). These devices are not used in routing or data transfer in the Thread network but as a redundant end point that can be called to assume the role of a router when needed.

Sleepy End Devices

These are the end points of a Thread network, also called host devices. Host devices are the individual IP-addressed functional equipment like thermostats, security cameras, heaters, etc. These devices can

also be referred to as a sleepy child or sleepy node. The router paired directly to the sleepy device is called the parent. The sleepy devices (end points) spend most of their time in the sleep mode and only wake up to transmit data. They only communicate through the parent device (e.g., the Nest thermostat is a sleepy end device).

A typical send cycle for a device might be:

- Wake from sleep mode.
- Perform any required startup and radio initialization.
- Go into receive mode and check if clear to transmit.
- Go into transmit mode.
- Transmit data.
- Get acknowledgment as applicable.
- Sleep.

In contrast to WiFi, which uses infrastructure mode, Thread uses the ad-hoc mode of networking.

Advantages:

- IP based, so easier to connect to other IP-based networks. Since it's based on 802.15.4, existing devices like ZigBee and 6loWPAN can easily migrate to Thread.
- No single point of failure by architecture, as it is capable of adjusting to network conditions. It supports full mesh-based network topology.
- Low-power operation, as it offers sleeping devices
- Secure

Note: Although there is no single point of failure by architecture in Thread networks, a single point of failure may exist due to poor network design.

Disadvantages:

- Not a very DIY-friendly protocol because of its complexity. Aims at the high-volume home-automation market.
- Still a very new network protocol that needs time to establish itself

As Silicon Labs and NXP are part of the Thread Alliance, they are pushing development boards that support Thread protocol.

ZigBee

ZigBee, like Thread is an alliance of over 200 companies which have collaborated to develop a robust and simple network for home and industrial automation.

ZigBee Design Goals

ZigBee was created for the sole purpose of serving the home and industrial automation, so the design goals are set in accordance to that:

- Low-power
- Security
- Co-existence with other radio networks which use the ISM band.
- Standardization
- Low-cost

Similar to Thread, ZigBee is built on top of the IEEE 802.15.4 standard.

ZigBee specifications fills in the gap left by the 802.15.4 to create a true mesh network but also supports other topologies like Star and Tree.

Devices in a ZigBee Network

There are three types of devices as defined by the ZigBee protocol:

- ZigBee Coordinator
- ZigBee Router
- ZigBee End Device (Node)

ZigBee Coordinator

The coordinator is the brains of the ZigBee network, it commissions devices to the network, stores the security keys and also bridges to other networks. There is only one ZigBee Coordinator in any network.

ZigBee Router

ZigBee Networks may have several routers to serve as intermediate routers or to transmit data within the network.

ZigBee End Device

The end device can only talk to the parent node (router or coordinator). It cannot talk to other end devices directly. Similar to Thread, these devices are designed by keeping in mind that they spend most of their lifetimes in sleep mode and only wake up to transmit data to the parent.

ZigBee assumes the channels from 11-26 in the 2.4GHz radio band used by most radios. The ZigBee channels have been specifically spaced as to co-exist with Wi-Fi channels without interference, if the channel assignments of both the networks are properly.

Note: The best spectral usage for both is achieved by setting the Wi-Fi channels to 1, 6, 11 and ZigBee channels to 15, 20, 25.

ZigBee vs Thread

ZigBee standard faced a big challenge when Thread came into existence in 2014. The ZigBee Alliance has then pushed to introduce their newest protocol called the **ZigBee 3.0**. They have tried to address the areas, where thread and other networks were better alternatives in the home automation space.

ZigBee 3.0 came with additional features such as ZigBee RF4CE and ZigBee Green Power.

ZigBee RF4CE - was developed to replace the IR remote with radio based remotes. This is aimed at making an universal remote controller that could be used to control your remote and as well as lights, lamps etc. Also this does away with the point and shoot limitation of IR.

ZigBee Green Power - was developed as an ultra-low power standard to support energy harvesting devices. It ensures very low power consumption by managing the network as such that these devices can be off for the most time.

One major feature that put ZigBee behind thread was IP compatibility. ZigBee 3.0 on the other hand is fully IP compatible. So you can now connect your ZigBee devices to the internet via a router.

Advantages: Pretty much the same as thread

Disadvantages: Short range, Low data speeds

Xbee's by Digi International are radio communication modules which support the ZigBee Protocol. They can also be loaded with firmwares to support ZigBee Pro and DigiMesh.

Bluetooth

Here we will discuss the evergreen Bluetooth Classic radio, as well as the new and coming Bluetooth Low Energy (BLE), which is specifically designed around low-powered devices used in IoT.

Like other radio technologies. Bluetooth uses the 2.4GHz spectrum in the ISM band. It has a range from 10m up to 100m (at higher transmit powers, and that means higher power consumption!). Bluetooth is again an ad-hoc type of network and provides point to point (P2P) connections.

Bluetooth Classic supports up to one master and seven slaves in one piconet. It also follows the star network topology, which means that other peripherals cannot talk to each other.

A few key things to note are: A master in one piconet cannot be a master in another but a master in one piconet can be a slave in another. Bluetooth Classic can be used to transmit audio, data but not video.

We will now focus our discussion on the BLE protocol and how Bluetooth has evolved because of this technology.

Three Flavors of Bluetooth

Bluetooth Low Energy is a complete divergence from Bluetooth Classic radio. It was designed with a new protocol stack, new profile architecture and to specifically be able to run on low-power sources such as a coin cell battery.

We need to understand that this radio technology has not taken over or replaced the existing Bluetooth Classic radio. This has led to a species of different flavors of Bluetooth, which correlate with each other.

Bluetooth technology can be classified into three types of devices:

- **Bluetooth Classic** – The traditional Bluetooth having a higher throughput, mostly used for wireless audio and file transmission. The ‘classic’ radio has support for Bluetooth Smart.
- **Bluetooth Smart** – Bluetooth Low Energy has been branded as Bluetooth Smart and transmits just state information. It was designed specifically for applications with low-duty cycles (i.e., the radio is effectively on for a short period of time). Bluetooth Smart devices cannot communicate with Bluetooth Classic devices.
- **Bluetooth SmartReady** – These devices are essentially the “hub” devices such as computers, smartphones, etc. They support both the “classic” and “smart” devices, just as our smartphones can connect to a Bluetooth speaker to transmit audio and also communicate to a fitness tracker.

Classic vs. Low Energy

BLE uses the same 2.4GHz ISM band as other wireless protocols. In contrast to Bluetooth Classic's 79, 1MHz wide channels, Bluetooth Low Energy just has 40 Channels, which are 2MHz wide.

BLE also uses a 1Mbps GFSK modulation, which gives it a higher range than Bluetooth Classic.

BLE uses an adaptive frequency-hopping algorithm to hop amongst the available channels, where only a subset of available frequencies are used and it can quickly recover from loss of packets due to a bad channel. This technique ensures lower energy consumed in the radio. Bluetooth Classic uses a pseudo random hope sequence, changing the transmission frequency 1,600 times a second.

One main thing to note is that BLE allows up to 128 devices to be connected to a single master, in contrast with just seven in classic.

BLE Stack

The BLE stack was specifically designed with low-powered applications in mind.

The core of BLE rests in the GAP and GATT profiles. We will limit our discussion to just these today.

Think of Generic Access Profile (GAP) and Generic Attribute Profile (GATT) as analogous to the basic way we communicate and network with the people around us. When you meet someone, you may announce yourself and offer basic information about yourself. If you want to connect with that person, you then exchange other personal information, such as a phone number or email address, to be able to communicate. You no longer need to identify or announce yourself when you meet the person again and can start sharing other information. GAP is the initial meeting phase, when you introduce yourself, and GATT is the phase when you establish a connection with the person and start communication.

Generic Access Profile (GAP)

The GAP defines the mechanisms a BLE device can use to communicate with the outside world.

Advertising

In this phase the device can be in either of the two phases:

- **Broadcasting Phase**: Where the device broadcasts public advertising data packets such as device name, signal strength, manufacturing details, etc.
- **Observing Phase**: Where the device listens to the advertising packets. There is still no connection between the devices. There can be more than one device observing the same advertiser.

In the process of establishing connection, the devices also assume roles, namely:

- **Peripheral**: The broadcasting device assumes the role of the peripheral, forming a pseudo connection to the device and responding to connection request of the central device to provide it with more information before connecting.
- **Central**: The observer, when initiating a connection to the advertising device, assumes the role of central device. It can also be considered the master and can connect to more than one peripheral at a time.

Once the connection is established between the peripheral and central devices, the advertising packet is no longer sent. Now the GATT profile has to be utilized to communicate in both directions.

Generic Attribute (GATT) Profile

GATT profile defines the way two BLE devices communicate with each other using attributes such as services and characteristics, which are defined in the attribute protocol.

Similar to GAP, there are certain defined roles that the communicating devices assume:

- **Client**: Usually, the central device assumes the role of the client. It typically sends a request to the GATT server. It can read and/or write attributes in the server.
- **Server**: Usually, the peripheral assumes the role of the server. It is called the server as it stores these attributes. The server responds to client request and sends the required attributes to it.

Peripheral or central can both act as a server or client, depending on the data flow.

When the connection is being established the central and peripheral decide upon a “connection interval,” which is the time between different connection events.

RFID and NFC

RFID

Radio-Frequency Identification (RFID) is a communication method used for tracking and identifying objects wirelessly.

As complicated as it may seem, it is one of the simplest communication methods, which is what makes it ubiquitous and yet invisible. It is used not only in tracking consumer products worldwide, but also in tracking vehicles for toll collection. Hospitals use it to track their patients, and farmers to track their cattle. RFID technology has become a part of our lives, and we might not even be aware of how it affects us.

Simply put, we can think of RFID tags as a close replacement to Universal Product Code or the bar code. But they are more efficient.

The RFID tags have read and write capabilities over traditional bar codes. They can be updated, changed and locked.

RFID technology includes tags and readers.

Tags

The tags are the end points in an RFID system. They store identity information along with other information as required by the purpose of the tag. There are two types of tags:

- **Active Tags:** These tags have an on-board power source of some sort, usually a battery, which means they can transmit stronger signals and therefore have more range. This type of tag can periodically transmit a signal irrespective of a reader.
- **Passive Tags:** These tags do not have any internal power source and get activated in the vicinity of a reader. Your metro or bus pass is generally a passive tag, which gets activated when you touch it to the reader. These tags harvest the radio energy transmitted by the reader.

Passive RFID tags primarily operate at three frequency ranges:

- Low Frequency (LF) 125 to 134 kHz
- High Frequency (HF) 13.56 MHz
- Ultra-High Frequency (UHF) 856 MHz to 960 MHz

Readers

Readers have a similar construction to an RFID tag. They have an antenna to receive and transmit signals to/from the tags. They might be either battery powered or plugged in to a wall outlet, as a reader requires strong RF signals to activate the tag (for passive) when it comes in the vicinity of the reader. The reader is connected to a reader controller, which manages the information read by the reader. The reader may also write or update a tag depending on the application. For example, the reader in subway stations is at the entry point. When the rider places a card (tag) on the reader, it reads the available money in the card and grants the user entry. At the passenger's exit, it calculates the fare and updates the amount on the card.

RFID tags can have three main components:

- An Integrated Circuit (IC) for storing identity information, processing it and modulating/demodulating the RF signals

- An antenna to receive and send the radio signals
- A power source (battery) if an active tag

The reader, along with the tags, form an RFID system.

NFC

Near-Field Communication (NFC) is an RF-based communication protocol. It is a subset of the RFID protocol, which is why it is similar to RFID but has significant differences. NFC has also become a very popular technology, and 9 out of 10 smartphones today are shipped with NFC capabilities. This has enabled contact-less payments such as Apple Pay and Google Wallet.

NFC smartphones pass along information from one smartphone to the other by tapping the two devices together, which turns sharing data such as contact info or photographs into a simple task. Applications like Android Beam facilitate this, whereas in Apple phones NFC can only be used for Apple Pay as of now.

NFC is being used to open car doors in NFC-enabled car keys with car manufacturers such as BMW. You might have also come across advertisements and marketing content that requires you to tap or wave your NFC-enabled smartphone to download an application or get more information about a product.

NFC is designed for very short-range communication (a few centimeters). It is one of the most power-efficient communication protocols. NFC also operates in the 13.56 MHz band, which is used by high-frequency RFID as well. Therefore, NFC can also read HF RFID tags.

NFC has two types of devices (one thing to note is that, in NFC, the device can act as a tag or a reader):

- **Initiator:** The device that initiates the communication is labeled as the initiator. It actively generates an RF field that can power the passive target.
- **Target:** This is the device that receives the information from the initiator. The target can either be passive (in the case of simple NFC tags) or active for peer to peer communication, such as in smartphones.

RFID vs. NFC

Due to the inherent similarity between these two communication protocols, we often find ourselves asking the differences in the technologies.

Since both NFC and RFID are omnipresent technologies, used with a large set of objects and devices around us, they both are useful protocols when it comes to IoT. These protocols provide a bridge to

various objects around us by not just communicating within the tag and the reader but connecting all of them to the internet.

Going Further

We have witnessed how the internet revolutionized how we communicate and collaborate. The new era of internet is not just about people. It is about the world around us. It is about intelligent connected devices.

The protocols we just reviewed were not all designed with Internet of Things in mind, but still they have proved to adapt quite well to IoT applications.

With the growth of IoT, it is inevitable that newer and tailor-made protocols are in the works specifically to cater to IoT networks. These connected smart devices have requirements different from today's human-centric internet.

There are three different types of connections in these protocols:

- **Device to Device** (D2D) – Where devices must communicate with each other
- **Device to Server** (D2S) – Where device data is collected and sent to a server
- **Server to Server** (S2S) – Where the server data is shared amongst other servers for analysis or to send back to the device.

We will talk about four such protocols, touching on all the different types of connections as mentioned above:

- **MQTT** (Message Queue Telemetry Transport) – a protocol for collecting device data and transmitting to the servers
- **XMPP** (Extensible Messaging and Presence Protocol) – a protocol for connecting devices to people who are connected to servers
- **DDS** (Data Distribution Service) – a protocol for fast communication between intelligent devices
- **AMQP** (Advanced Message Queuing Protocol) – a protocol designed for effective communications between different servers

MQTT

MQTT.org offers this definition of MQTT: a machine-to-machine (M2M) IoT connectivity protocol. It was designed as an extremely lightweight publish/subscribe messaging transport. It is useful for connections with remote locations where a small code footprint is required and/or network bandwidth is at a premium. For example, it has been used in sensors communicating to a broker via satellite link, over

occasional dial-up connections with health care providers, and in a range of home-automation and small-device scenarios. It is also ideal for mobile applications because of its small size, low power usage, minimized data packets and efficient distribution of information to one or many receivers.

MQTT's goal is to collect data from many devices and transport the data to the IT infrastructure. It can be one of the best solutions where sensor data from thousands of sensors has to be transported to a single location for analysis.

XMPP

XMPP.org describes it as: the Extensible Messaging and Presence Protocol, a set of open technologies for instant messaging, presence, multiparty chat, voice and video calls, collaboration, lightweight middleware, content syndication, and generalized routing of XML data.

Think of XMPP as a means for the connected devices to discover each other and start chatting – that is, to start exchanging information as we do!

DDS

Different from MQTT and XMPP, which are device-to-server protocols, DDS uses devices that directly utilize the device data. It is a protocol to distribute the data of one device to another. DDS can effectively deliver millions of messages per second to many simultaneous receivers.

DDS is peer-to-peer communication. Elimination of servers and message brokers simplifies communications, minimizes latency and reduces complexity. It is reliable for IoT applications that require a reliable and high-performance architecture.

AMQP

AMQP is a server-to-server type of protocol. It sends transactional messages between servers. The main feature of AMQP is reliability, and it is capable of sending thousands of queued transactions without losing any data.

AMQP.org defines the protocol as: an open standard for passing business messages between applications or organizations. It connects systems, feeds business processes with the information they need and reliably transmits onward the instructions that achieve their goals.

It was initially developed for the banking industry as a middleware for tracking and delivering highly important messages. In the IoT paradigm, AQMP is most suitable for server-based analysis functions, where data from different servers need to be communicated for effective analysis.

In conclusion, there is no set best protocol for IoT, which is still in a phase where we cannot standardize it and may never be able to do so. Its diverse nature of applications and millions of different types of connected devices make it unique.