Internet of Things:

In 2018, the number of installed IoT devices was estimated to be about 1.2 billion; by 2030, the number is expected to reach 125 billion.

Google's definition:

The Internet of Things (IoT) is a sprawling set of technologies and use cases that has no clear, single definition. One workable view frames IoT as the use of network-connected devices, embedded in the physical environment, to improve some existing process or to enable a new scenario not previously possible.

A "Thing" in the "Internet of Things" is a processing unit that is capable of connecting to the internet and exchanging data with the cloud. Devices are often called "smart devices" or "connected devices." They communicate two types of data: telemetry and state.

1. Telemetry:

   Data collected by the device is called telemetry. This is the eyes-and-ears data that IoT devices provide to applications. Telemetry is read-only data about the environment, usually collected through sensors.

2. State:

   State information describes the current status of the device, not of the environment. This information can be read/write. It is updated, but usually not frequently.

Actuators
An actuator is a basic motor that can be used to move or control a mechanism or system, based on a specific set of instructions.

Typically, in the Industrial IoT, there are three types of actuators:

• Electrical - Powered by a motor that converts electrical energy into mechanical operations

• Hydraulic - Uses fluid pressure to perform mechanical movement

• Pneumatic - Uses compressed air to enable mechanical operations

In other areas, an actuator can be responsible for transforming an electrical signal into physical output. This physical output could provide information to a user via LEDs or modify another device or environment. For example, the heater in the thermostat feedback loop is an actuator because it changes the status of the controlled environment (temperature) in response to an electrical signal.

Regardless of how the actuator causes the movement to be performed, the basic function of an actuator is to receive a signal from the controller, and based on that signal, perform a set action.

Controllers
Controllers are responsible for collecting data from sensors and providing network or Internet connectivity. Controllers may have the ability to make immediate decisions, or they may send data to a more powerful computer for analysis. This more powerful computer might be in the same LAN as the controller or might only be accessible through an Internet connection.

Imagine smart traffic lights that contain sensors and actuators. The sensors detect and report traffic activity to the controller. The controller is able to process this data locally and determine optimal traffic patterns. Using this information the controller will send signals to the actuators in the traffic lights to adjust traffic flows. This is an example of machine-to-machine (M2M) communication. In this scenario, the sensors, actuators, and the controller all exist within the fog. That means that the information is not sent beyond the local network of end devices.

IP-enabled controllers
The IP-enabled controller forwards information across an IP network, and allows individuals to access the controller remotely. In addition to forwarding basic information in an M2M configuration, some controllers are able to perform more complex operations. Some controllers can consolidate information from multiple sensors or perform basic analysis of data received.

A controller may also act as a gateway to the local network. As an example, if the IoT system is designed to capture the temperature changes within every apartment of the building, the controller may pass the data up to be stored or analyzed on servers in the local or edge network. Where the data is processed will impact the speed in which change can take place in the system. Data can be stored and processed on devices that are near the edge of the network or even closer to the sensors. This type of processing is called fog computing. Fog nodes that create areas for processing are part of this system.

Closed Loop Controllers
Where there is a feedback to the system

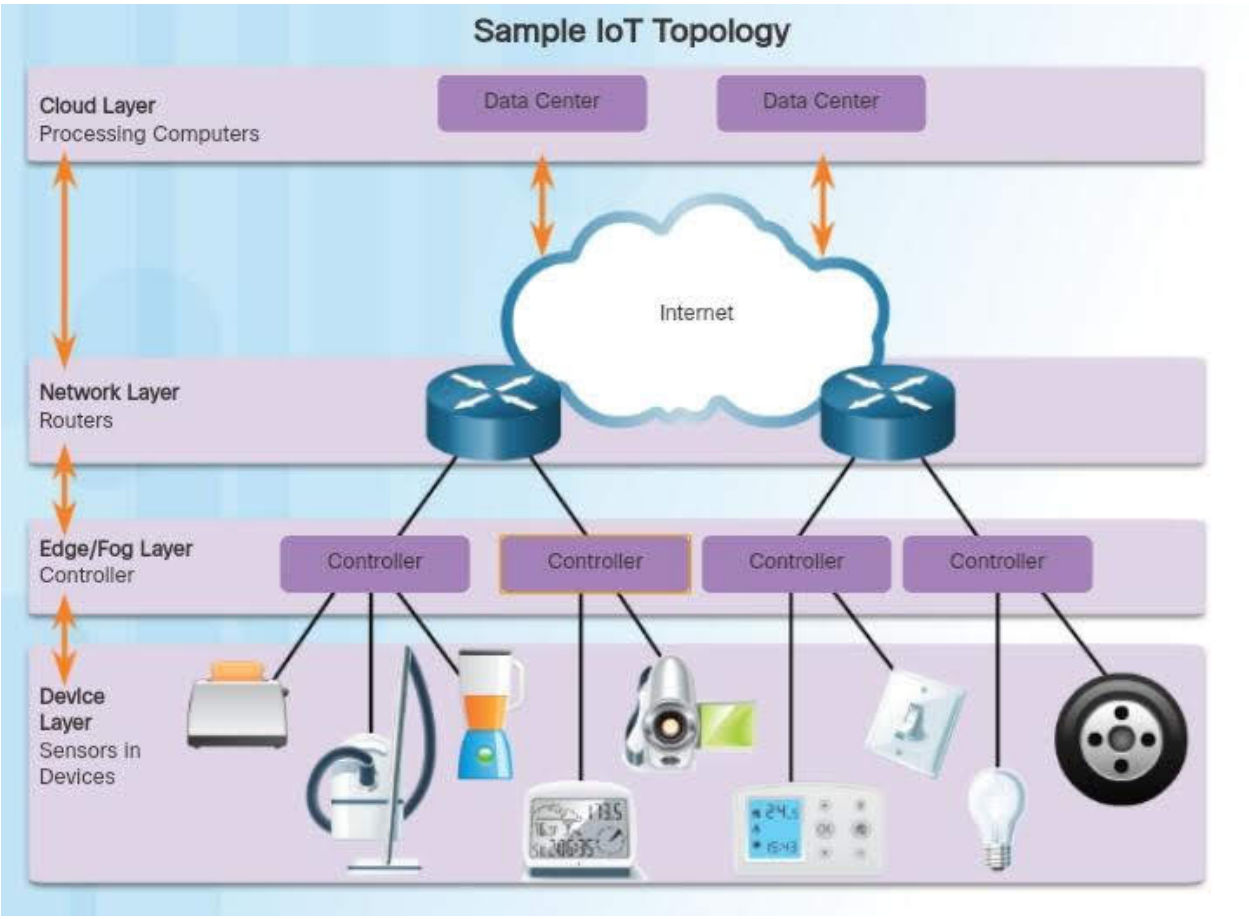
There are many types of closed-loop controllers. A proportional-integral-derivative (PID) controller is an efficient way to implement feedback control. PID controllers are used in many types of industrial applications because they are simple to use. PID controllers are easy to understand, implement, test, and troubleshoot. The Arduino and Raspberry Pi devices can be used to implement PID controllers.

The acronym PID stands for proportional, integral, and derivative.

Proportional controllers (P) – These controllers look specifically at the difference between the measured output and the desired output. The amount of change sent to the plant by the controller is proportional to the size of the error from the last iteration. For example, in an HVAC system the controller would activate 50% of the chillers if the sensor detected a one degree deviation from the set temperature. At 2 degrees deviation, 100% capacity would be activated. Proportional controls will usually overshoot the set temperature, as shown in the P graph because they are only looking at the deviation from the set point at any given time.

Proportional Integral controllers (PI) – These proportional and integral controllers use historical data to measure how long the system has deviated from the set temperature. The longer the system has deviated from the set temperature, the larger the response from the controller. In an HVAC system, the controller would account for historical data and time when adjusting the system. Although integral controllers will also overshoot the set temperature, the variation will decrease overtime.

Derivative controllers (PID) – These proportional, integral, and derivative controllers include data about how quickly the system is approaching the desired output. In an HVAC system, the derivative function of a PID controller looks at the rate of change in temperature. This allows the controller to quickly adjust the output as the system approaches the desired output, as shown in the PID graph.
Note that proportional control can be implemented with pneumatic, analog, or electronic controllers.



Many new organizations such as the Industrial Internet Consortium, OpenFog Consortium, and the Open Connectivity Foundation, are working with members from industry, government, and academia to support and encourage the creation and adoption of new IoT systems. To assist in this goal, they are helping to develop standard architectures and frameworks that will allow devices to connect and communicate reliably and safely in the IoT.

The Industrial Internet Consortium
(IIC) has used workgroups with representatives from industries such as energy, manufacturing, transportation, and healthcare, to create the Industrial Internet Reference Architecture. This architecture is a standards-based architectural template and methodology for the industrial IoT. The IIC is also collaborating with companies such as Cisco, Bosch Rexroth, Intel, and National Instruments to develop the world's first Time Sensitive Networking (TSN) testbed. The goal of the testbed is to display the value of new Ethernet standards required to support seamless interoperability among the emerging smart devices required for new IoT systems. The testbed will provide feedback to relevant standards organizations on areas for consideration and improvement.

The OpenFog Consortium is creating an open reference architecture for fog computing. They also build operational models and testbeds, define and advance technology, educate the market, and promote business development through an OpenFog ecosystem.

The Open Connectivity Foundation (OCF) is working towards the creation of solutions that map to a single, open IoT interoperability specification. To that end, the OFC has sponsored the IoTivity Project, an open source software framework. This framework enables seamless connectivity for devices such as appliances, phones, computers, and industrial equipment. These devices will be able to communicate with one another regardless of manufacturer, operating system, or chipset.

IoT World Forum Reference Model(7 levels)

Level		Description
7	Collaboration & Processes (Involving people and business processes)	Transcends multiple applications to include the communication and collaboration required between people and business processes.
6	Application (Reporting, analytics, control)	Information interpretation based on the nature of the device data and business needs.
5	Data Abstraction (Aggregation and access)	Focused on rendering the data and its storage in ways to enable application development.
4	Data Accumulation (Storage)	Data in motion is converted to data at rest. The data is also transformed so that it can be consumed by upper levels.
3	Edge (Fog) Computing (Data element analysis and transformation)	Converts data into information that is suitable for storage and higher level processing.
2	Connectivity (Communication and processing units)	Responsible for reliable and timely data transmission between devices and the network, across networks, and between the network and data processing in Level 3.
1	Physical Devices & Controllers (The “Things” of IoT)	Includes a wide range of endpoint devices that send and receive information.

Simplified IoT Architecture

Developing IoT systems to interconnect smart objects is a complex task. Many smart objects designed by different vendors have constraints concerning proprietary software that makes interoperability a challenge. Also, devices such as sensors, actuators, and controllers, have constraints in bandwidth, power, size, and installed location. These constraints amplify the issues surrounding security and privacy.

Several architectures exist to help facilitate the design and creation of IoT systems. The OSI model, TCP/IP model, and the IoT World Forum Reference model have been presented as examples.

Emerging opinions are now opting for a simpler approach based on the type or level of connections between the smart objects. Each level of expanded connectivity has a different set of design issues and requirements for security and privacy to consider.

Device-to-Device

IoT solutions often support one smart object connecting directly to another via a wireless protocol such as Bluetooth or Zigbee. An example of this level is a sensor that is located in a vineyard and detects dry soil. It sends a signal to an actuator that triggers the watering system.

Device-to-Cloud

In a device-to-network-to-cloud communication model, the IoT device connects through a local network directly to an Internet cloud service using traditional wired Ethernet or Wi-Fi connections. This model establishes a connection between the device, the IP network, and the cloud to allow the exchange of data and control messages.

Device-to-Gateway-to-Cloud

Many smart devices, such as fitness trackers, are not IP-enabled and do not have the native ability to connect directly to the fog or the cloud. For these devices, there is application software operating on a local gateway device which acts as an intermediary between the device and the cloud service. The gateway may also provide security and data or protocol translation. For devices, like fitness trackers, the gateway is often an application running on a smartphone.

Device-to-Gateway-to-Cloud-to-Application

Another connection option supports smart device data collection and transfer through a gateway to a local IP network. The data then flows to the fog or the cloud and is then available for users to export and analyze. The data is often analyzed in combination with data from other sources or other cloud services.

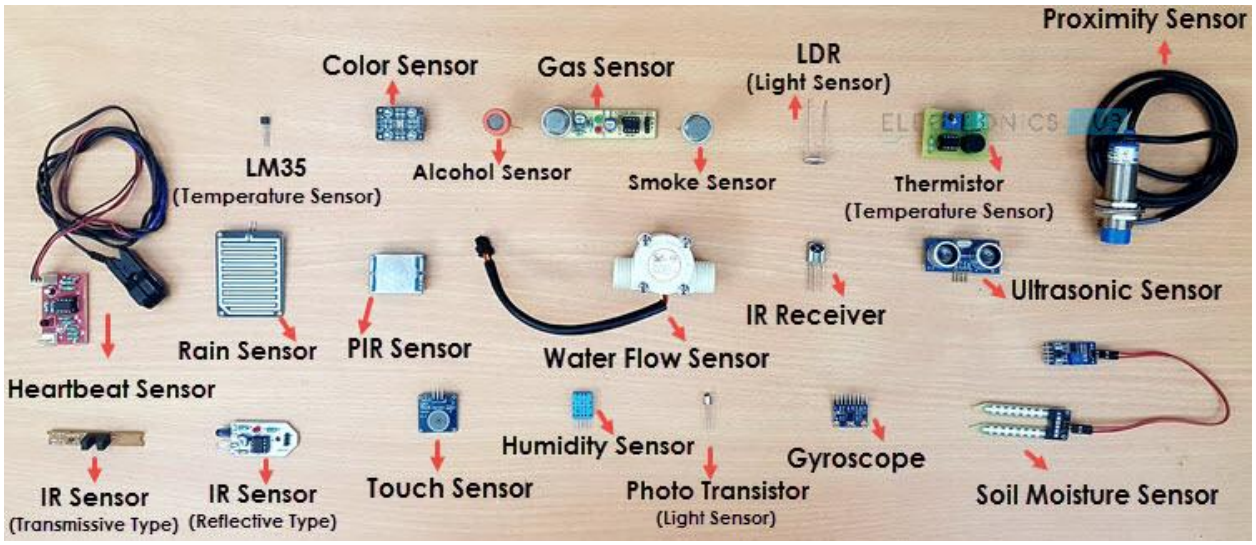
Knowledge of the four basic levels of connections will ensure that consideration is given to device interoperability and open standards. These are key considerations when developing an internetworked IoT system.

What is a Sensor?

There are numerous definitions as to what a sensor is but I would like to define a Sensor as an input device which provides an output (signal) with respect to a specific physical quantity (input).

The term “input device” in the definition of a Sensor means that it is part of a bigger system which provides input to a main control system (like a Processor or a Microcontroller).

Another unique definition of a Sensor is as follows: It is a device that converts signals from one energy domain to electrical domain. The definition of the Sensor can be understood if we take an example in to consideration.



The simplest example of a sensor is an LDR or a Light Dependent Resistor. It is a device, whose resistance varies according to intensity of light it is subjected to. When the light falling on an LDR is more, its resistance becomes very less and when the light is less, well, the resistance of the LDR becomes very high.

We can connect this LDR in a voltage divider (along with other resistor) and check the voltage drop across the LDR. This voltage can be calibrated to the amount of light falling on the LDR. Hence, a Light Sensor.

Now that we have seen what a sensor is, we will proceed further with the classification of Sensors.

https://www.electronicshub.org/different-types-sensors/

Classification of Sensors

There are several classifications of sensors made by different authors and experts. Some are very simple and some are very complex. The following classification of sensors may already be used by an expert in the subject but this is a very simple classification of sensors.

In the first classification of the sensors, they are divided in to Active and Passive. Active Sensors are those which require an external excitation signal or a power signal.

Passive Sensors, on the other hand, do not require any external power signal and directly generates output response.

The other type of classification is based on the means of detection used in the sensor. Some of the means of detection are Electric, Biological, Chemical, Radioactive etc.

The next classification is based on conversion phenomenon i.e. the input and the output. Some of the common conversion phenomena are Photoelectric, Thermoelectric, Electrochemical, Electromagnetic, Thermooptic, etc.

The final classification of the sensors are Analog and Digital Sensors. Analog Sensors produce an analog output i.e. a continuous output signal with respect to the quantity being measured.

Digital Sensors, in contrast to Analog Sensors, work with discrete or digital data. The data in digital sensors, which is used for conversion and transmission, is digital in nature.

Different Types of Sensors

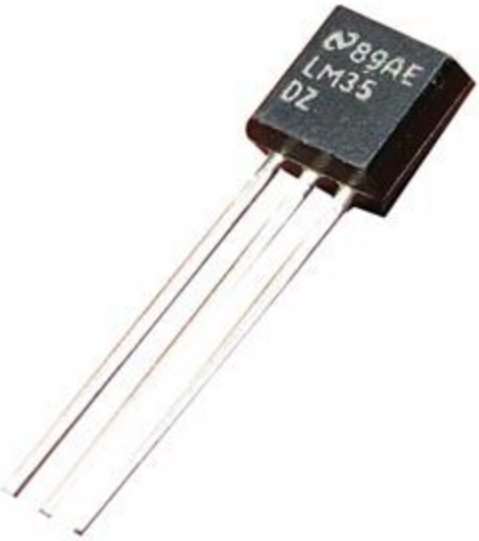
The following is a list of different types of sensors that are commonly used in various applications. All these sensors are used for measuring one of the physical properties like Temperature, Resistance, Capacitance, Conduction, Heat Transfer etc.

- Temperature Sensor
- Proximity Sensor
- Accelerometer
- IR Sensor (Infrared Sensor)
- Pressure Sensor
- Light Sensor
- Ultrasonic Sensor
- Smoke, Gas and Alcohol Sensor
- Touch Sensor
- Color Sensor
- Humidity Sensor
- Tilt Sensor
- Flow and Level Sensor

We will see about few of the above mentioned sensors in brief. More information about the sensors will be added subsequently. A list of projects using the above sensors is given at the end of the page.

Temperature Sensor

One of the most common and most popular sensor is the Temperature Sensor. A Temperature Sensor, as the name suggests, senses the temperature i.e. it measures the changes in the temperature.



LM35 - Temperature Sensor IC

10KΩ NTC Thermistor

In a Temperature Sensor, the changes in the Temperature correspond to change in its physical property like resistance or voltage.

There are different types of Temperature Sensors like Temperature Sensor ICs (like LM35), Thermistors, Thermocouples, RTD (Resistive Temperature Devices), etc.

Temperature Sensors are used everywhere like computers, mobile phones, automobiles, air conditioning systems, industries etc.

A simple project using LM35 (Celsius Scale Temperature Sensor) is implemented in this project: **TEMPERATURE CONTROLLED SYSTEM**.

Proximity Sensors

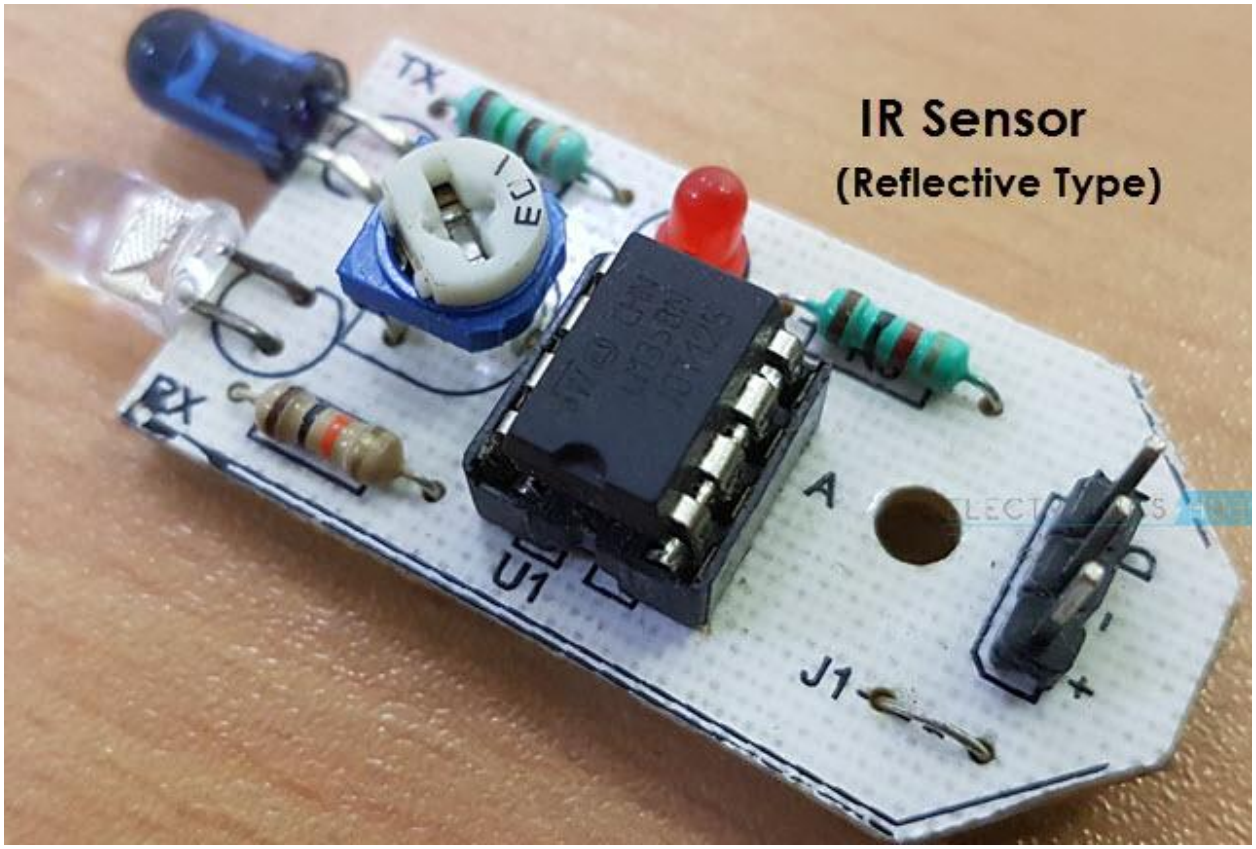
A Proximity Sensor is a non-contact type sensor that detects the presence of an object. Proximity Sensors can be implemented using different techniques like Optical (like Infrared or Laser), Ultrasonic, Hall Effect, Capacitive, etc.

Some of the applications of Proximity Sensors are Mobile Phones, Cars (Parking Sensors), industries (object alignment), Ground Proximity in Aircrafts, etc.

Proximity Sensor in Reverse Parking is implemented in this Project: **REVERSE PARKING SENSOR CIRCUIT**.

Infrared Sensor (IR Sensor)

IR Sensors or Infrared Sensor are light based sensor that are used in various applications like Proximity and Object Detection. IR Sensors are used as proximity sensors in almost all mobile phones.



There are two types of Infrared or IR Sensors: Transmissive Type and Reflective Type. In Transmissive Type IR Sensor, the IR Transmitter (usually an IR LED) and the IR Detector (usually a Photo Diode) are positioned facing each other so that when an object passes between them, the sensor detects the object.

The other type of IR Sensor is a Reflective Type IR Sensor. In this, the transmitter and the detector are positioned adjacent to each other facing the object. When an object comes in front of the sensor, the sensor detects the object.

Different applications where IR Sensor is implemented are Mobile Phones, Robots, Industrial assembly, automobiles etc.

A small project, where IR Sensors are used to turn on street lights: **STREET LIGHTS USING IR SENSORS**.

Ultrasonic Sensor

An Ultrasonic Sensor is a non-contact type device that can be used to measure distance as well as velocity of an object. An Ultrasonic Sensor works based on the properties of the sound waves with frequency greater than that of the human audible range.

Using the time of flight of the sound wave, an Ultrasonic Sensor can measure the distance of the object (similar to SONAR). The Doppler Shift property of the sound wave is used to measure the velocity of an object.

Arduino based Range Finder is a simple project using Ultrasonic Sensor: **PORTABLE ULTRASONIC RANGE METER**.

The following is a small list of projects based on few of the above mentioned Sensors.

Light Sensor – LIGHT DETECTOR USING LDR

Smoke Sensor – SMOKE DETECTOR ALARM CIRCUIT

Alcohol Sensor – HOW TO MAKE ALCOHOL BREATHALYZER CIRCUIT?

Touch Sensor – TOUCH DIMMER SWITCH CIRCUIT USING ARDUINO

Color Sensor – ARDUINO BASED COLOR DETECTOR

Humidity Sensor – DHT11 HUMIDITY SENSOR ON ARDUINO

Tilt Sensor – HOW TO MAKE A TILT SENSOR WITH ARDUINO?

M2M	IoT
Simple device-to-device communication usually within an embedded software at client site	Grand-scale projects and want-it-all approach
Isolated systems of devices using same standards	Integrates devices, data and applications across varying standards
Limited scalability options	Inherently more scalable
Wired or cellular network used for connectivity	Usually devices require active Internet connection
Extensive background of historical applications	State-of-the-art approach with roots in M2M

Mobile Networks

https://www.youtube.com/watch?v=1JZG9x_VOwA
https://www.youtube.com/watch?v=GEx_d0SjvS0&t=40s
https://www.youtube.com/watch?v=PAqEjQjxgdY
https://www.youtube.com/watch?v=TQVI5-G3u2U

12 Aug, 2020

IoT Architecture: the Pathway from Physical Signals to Business Decisions

Share:  **Comment:**

CONTENTS

Reading time: 8 minutes

IoT solutions have become a regular part of our lives. From the smartwatch on your wrist to industrial enterprises, connected devices are everywhere. Having *things* work for us is no longer sci-fi fantasy.

You tap the screen of your smartphone or say a word, and get immediate results. A door automatically opens, a coffee machine starts grinding beans to make a

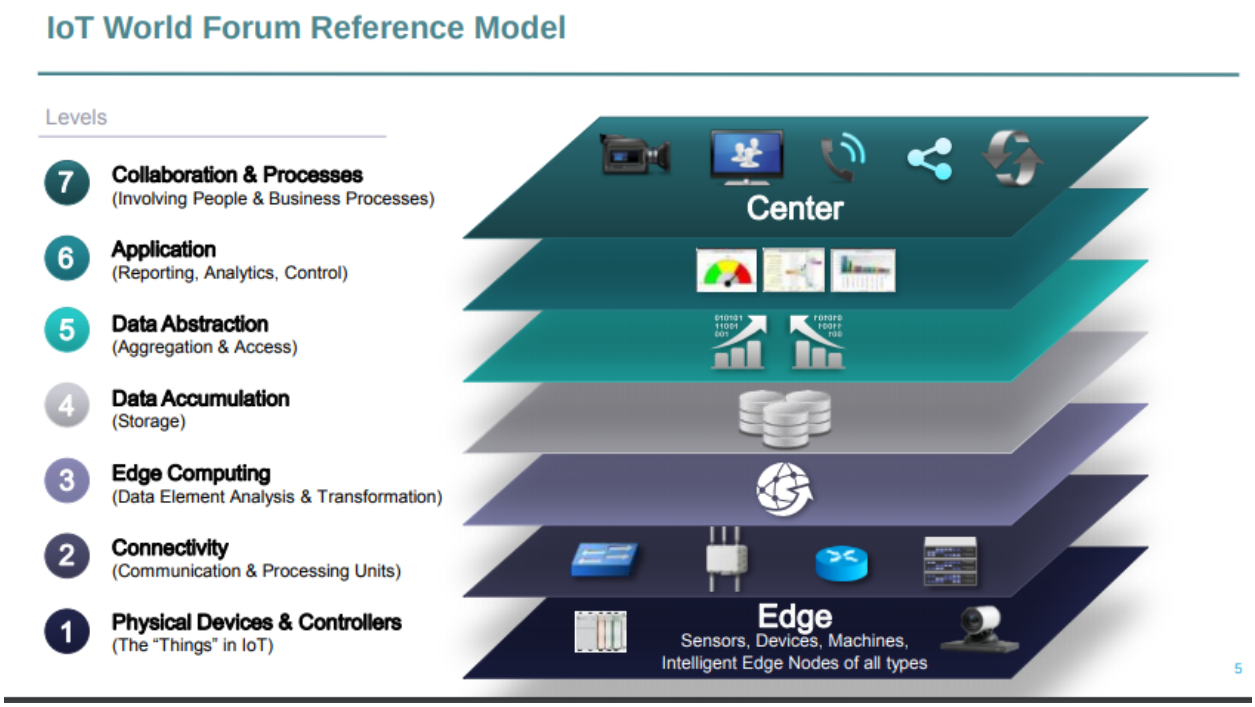
perfect cup of espresso while you receive analytical reports based on fresh data from sensors miles away.

But between your command and tasks fulfilled, there lies a large and mostly invisible infrastructure, that involves multiple elements and interactions. This article describes IoT — the Internet of Things — through its architecture, layer to layer. Let's peek behind the curtain to see how everyday magic works.

Major IoT building blocks and layers

Before we go any further, it's worth pointing out that there is no single, agreed-upon IoT architecture. It varies in complexity and number of architectural layers depending on a particular business task.

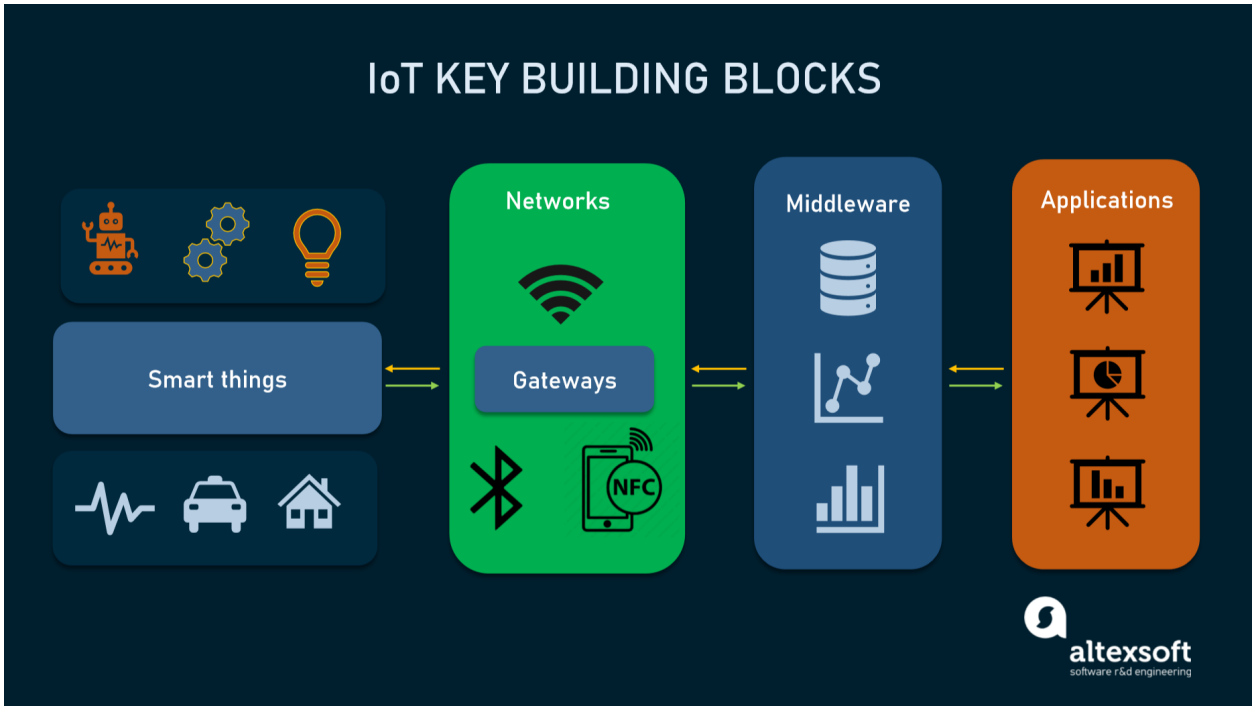
For example, the Reference Model introduced in 2014 by Cisco, IBM, and Intel at the 2014 IoT World Forum has as many as seven layers. According to an official press release by Cisco forum host, the architecture aims to *"help educate CIOs, IT departments, and developers on deployment of IoT projects, and accelerate the adoption of IoT."*



The standardized architectural model proposed by IoT industry leaders. Source: Internet of Things World Forum

But no matter the use case and number of layers, the key building blocks of any IoT structure are always the same, namely:

- **smart things;**
- **networks and gateways enabling low-power devices (which is often the case in IoT) to enter the big Internet;**
- **the middleware or IoT platforms providing data storage spaces and advanced computing engines along with analytical capabilities; and**
- **applications, allowing end users to benefit from IoT and manipulate the physical world.**



The skeleton of an IoT system.

These elements make up the backbone of any IoT system upon which effective, multi-layered architecture can be developed. Most commonly, these layers are:

- **the perception layer hosting smart things;**
- **the connectivity or transport layer transferring data from the physical layer to the cloud and vice versa via networks and gateways;**
- **the processing layer employing IoT platforms to accumulate and manage all data streams; and**
- **the application layer delivering solutions like analytics, reporting, and device control to end users.**

Besides the most essential components, the article also describes three additional layers:

- **the edge or fog computing layer performing data preprocessing close to the edge, where IoT things collect new information. Typically, edgy computing occurs on gateways;**
- **the business layer where businesses make decisions based on the data; and**
- **the security layer encompassing all other layers.**

Often viewed as optional, these extra components none the less make an IoT project neatly fit modern business needs.

Perception layer: converting analog signals into digital data and vice versa

The initial stage of any IoT system embraces a wide range of “things” or endpoint devices that act as a bridge between the real and digital worlds. They vary in form and size, from tiny silicon chips to large vehicles. By their functions, IoT things can be divided into the following large groups.

Sensors such as probes, gauges, meters, and others. They collect physical parameters like temperature or humidity, turn them into electrical signals, and

send them to the IoT system. IoT sensors are typically small and consume little power.

Actuators, translating electrical signals from the IoT system into physical actions. Actuators are used in motor controllers, lasers, robotic arms.

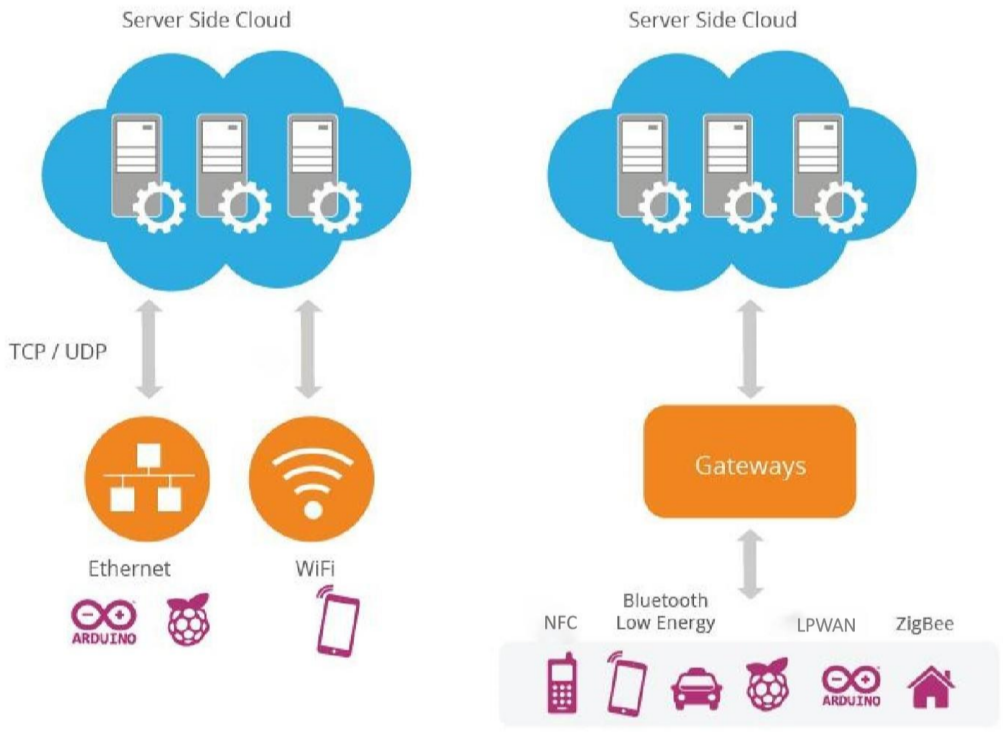
Machines and devices connected to sensors and actuators or having them as integral parts.

It's important to note that the architecture puts no restriction on the scope of its components or their location. The edge-side layer can include just a few "things" physically placed in one room or myriads of sensors and devices distributed across the world.

Connectivity layer: enabling data transmission

The second level is in charge of all communications across devices, networks, and cloud services that make up the IoT infrastructure. The connectivity between the physical layer and the cloud is achieved in two ways:

- **directly, using TCP or UDP/IP stack;**
- **via gateways — hardware or software modules performing translation between different protocols as well as encryption and decryption of IoT data.**



Two key models of connectivity between physical and cloud levels in IoT. Source: WSO2

The communications between devices and cloud services or gateways involve different networking technologies.

Ethernet connects stationary or fixed IoT devices like security and video cameras, permanently installed industrial equipment, and gaming consoles.

WiFi, the most popular technology of wireless networking, is a great fit for data-intensive IoT solutions that are easy to recharge and operate within a small area. A good example of use is smart home devices connected to the electrical grid.

NFC (Near Field Communication) enables simple and safe data sharing between two devices over a distance of 4 inches (10 cm) or less.

Bluetooth is widely used by wearables for short-range communications. To meet the needs of low-power IoT devices, the Bluetooth Low-Energy (BLE) standard was designed. It transfers only small portions of data and doesn't work for large files.

LPWAN (Low-power Wide-area Network) was created specifically for IoT devices. It provides long-range wireless connectivity on low power consumption with a battery life of 10+ years. Sending data periodically in small portions, the technology meets the requirements of smart cities, smart buildings, and smart agriculture (field monitoring).

ZigBee is a low-power wireless network for carrying small data packages over short distances. The outstanding thing about ZigBee is that it can handle up to 65,000 nodes. Created specifically for home automation, it also works for low-power devices in industrial, scientific, and medical sites.

Cellular networks offer reliable data transfer and nearly global coverage. There are two cellular standards developed specifically for IoT things. LTE-M (Long Term Evolution for Machines) enables devices to communicate directly with the cloud and exchange high volumes of data. NB-IoT or Narrowband IoT uses low-frequency channels to send small data packages.

NETWORKING TECHNOLOGIES USED in IoT

Network	Connectivity	Pros and Cons	Popular use cases
Ethernet	Wired, short-range	☺ High speed ☺ Security ☹ Range limited to wire length ☹ Limited mobility	Stationary IoT: video cameras, game consoles, fixed equipment
WiFi	Wireless, short-range	☺ High speed ☺ Great compatibility ☹ Limited range ☹ High power consumption	Smart home, devices that can be easily recharged
NFC	Wireless, ultra-short-range	☺ Reliability ☺ Low power consumption ☹ Limited range ☹ Lack of availability	Payment systems, smart home
Bluetooth Low-Energy	Wireless, short-range	☺ High speed ☺ Low power consumption ☹ Limited range ☹ Low bandwidth	Small home devices, wearables, beacons
LPWAN	Wireless, long-range	☺ Long range ☺ Low power consumption ☹ Low bandwidth ☹ High latency	Smart home, smart city, smart agriculture (field monitoring)
ZigBee	Wireless, short-range	☺ Low power consumption ☺ Scalability ☹ Limited range ☹ Compliance issues	Home automation, healthcare and industrial sites
Cellular networks	Wireless, long-range	☺ Nearly global coverage ☺ High speed ☺ Reliability ☹ High cost ☹ High power consumption	Drones sending video and images

altexsoft
software r&d engineering

Major networking technologies used in the IoT projects.

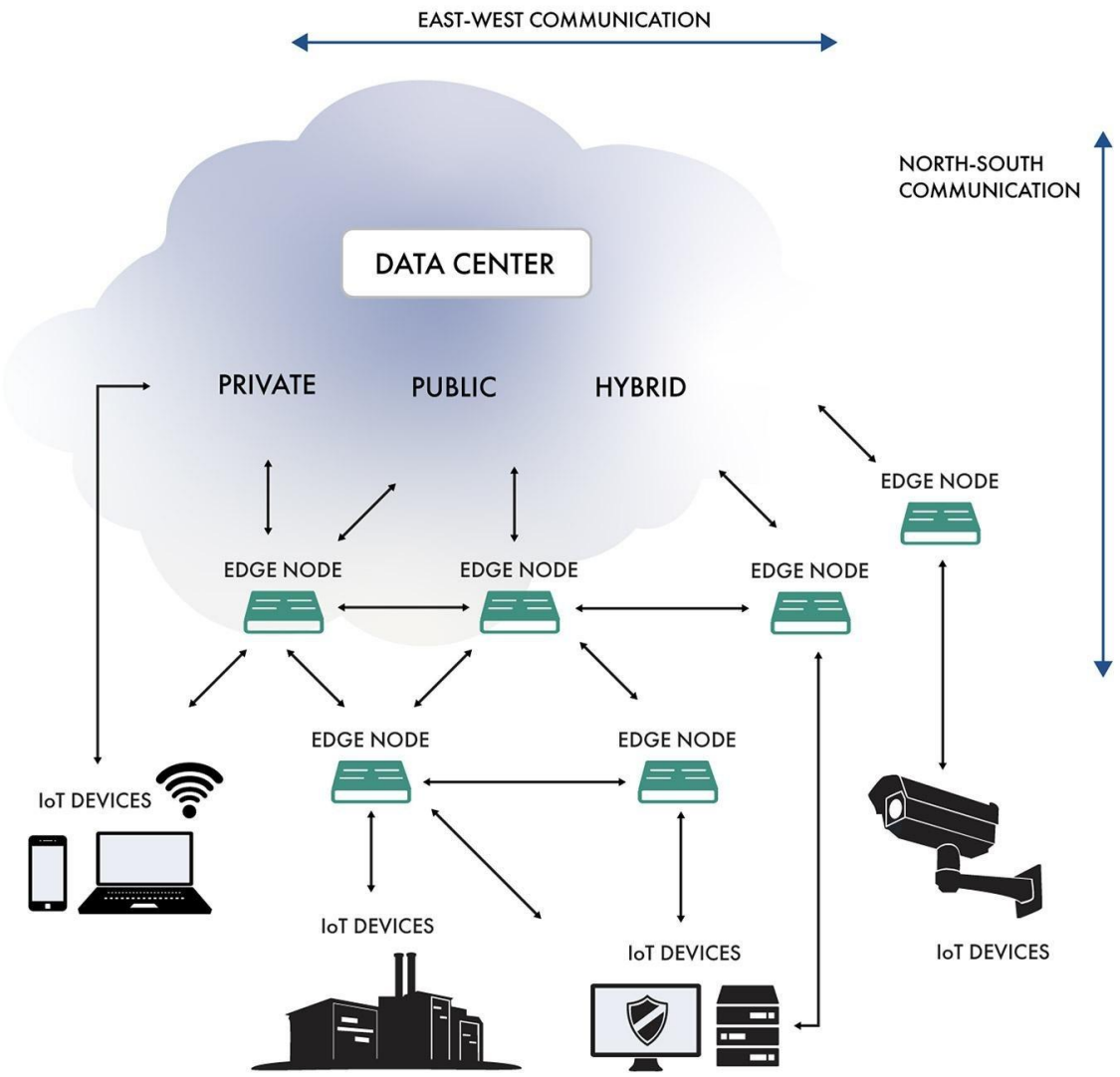
Once parts of the IoT solution are networked, they still need messaging protocols to share data across devices and with the cloud. The most popular protocols used in the IoT ecosystems are:

- **DDS (the Data Distribution Service) which directly connects IoT things to each other and to applications addressing the requirements of real-time systems;**
- **AMQP (the Advanced Message Queuing Protocol) aiming at peer-to-peer data exchange between servers;**
- **CoAP (the Constrained Application Protocol), a software protocol designed for *constrained devices* — end nodes limited in memory and power (for example, wireless sensors). It feels much like HTTP but uses fewer resources;**
- **MQTT (the Message Queue Telemetry Transport), a lightweight messaging protocol built on top of TCP/IP stack for centralized data collection from low-powered devices.**

Edge or fog computing layer: reducing system latency

This level is essential for enabling IoT systems to meet the speed, security, and scale requirements of the 5th generation mobile network or 5G. The new wireless standard promises faster speeds, lower latency, and the ability to handle many more connected devices, than the current 4G standard.

The idea behind edge or fog computing is to process and store information as early and as close to its sources as possible. This approach allows for analyzing and transforming high volumes of real-time data locally, at the edge of the networks. Thus, you save the time and other resources that otherwise would be needed to send all data to cloud services. The result is reduced system latency that leads to real-time responses and enhanced performance.



The scheme of communications between IoT devices, edge nodes, and cloud data centers. Source: DesignNews

Edge computing occurs on gateways, local servers, or other edge nodes scattered across the network. At this level, data can be:

- **evaluated to determine if it needs further processing at higher levels,**
- **formatted for further processing,**
- **decoded,**
- **filtered, and**
- **redirected to an additional destination**

To sum up, the first three layers see data in motion, as it is constantly moving and altering. Only on hitting the next level, is data finally at rest and available for use by consumer applications.

Processing layer: making raw data useful

The processing layer accumulates, stores, and processes data that comes from the previous layer. All these tasks are commonly handled via IoT platforms and include two major stages.

Data accumulation stage

The real-time data is captured via an API and put at rest to meet the requirements of non-real-time applications. The data accumulation component stage works as

a transit hub between event-based data generation and query-based data consumption.

Among other things, the stage defines whether data is relevant to the business requirements and where it should be placed. It saves data to a wide range of storage solutions, from data lakes capable of holding unstructured data like images and video streams to event stores and telemetry databases. The total goal is to sort out a large amount of diverse data and store it in the most efficient way.

Data abstraction stage

Here, data preparation is finalized so that consumer applications can use it to generate insights. The entire process involves the following steps:

- **combining data from different sources, both IoT and non-IoT, including ERM, ERP, and CRM systems;**
- **reconciling multiple data formats; and**
- **aggregating data in one place or making it accessible regardless of location through data virtualization.**

Similarly, data collected at the application layer is reformatted here for sending to the physical level so that devices can "understand" it.

Together, the data accumulation and abstraction stages veil details of the hardware, enhancing the interoperability of smart devices. What's more, they let software developers focus on solving particular business tasks — rather than on delving into the specifications of devices from different vendors.

Application layer: addressing business requirements

At this layer, information is analyzed by software to give answers to key business questions. There are hundreds of IoT applications that vary in complexity and function, using different technology stacks and operating systems. Some examples are:

- **device monitoring and control software,**
- **mobile apps for simple interactions,**
- **business intelligence services, and**
- **analytic solutions using machine learning.**

Currently, applications can be built right on top of IoT platforms that offer software development infrastructure with ready-to-use instruments for data mining, advanced analytics, and data visualization. Otherwise, IoT applications use APIs to integrate with middleware.

Business layer: implementing data-driven solutions

The information generated at the previous layers brings value if only it results in problem-solving solution and achieving business goals. New data must initiate

collaboration between stakeholders who in turn introduce new processes to enhance productivity.

The decision-making usually involves more than one person working with more than one software solution. For this reason, the business layer is defined as a separate stage, higher than a single application layer.

Security layer: preventing data breaches

It goes without saying that there should be a security layer covering all the above-mentioned layers. IoT security is a broad topic worthy of a separate article. Here we'll only point out the basic features of the safe architecture across different levels.

Device security. Modern manufacturers of IoT devices typically integrate security features both in the hardware and firmware installed on it. This includes

- **embedded TPM (Trusted Platform Module) chips with cryptographic keys for authentication and protection of endpoint devices;**
- **a secure boot process that prevents unauthorized code from running on a powered-up device;**
- **updating security patches on a regular basis; and**
- **physical protection like metal shields to block physical access to the device.**

The technologies we'll be comparing are:

- *Bluetooth*® Technology
- Wi-Fi
- IEEE 802.15.4-Based Technologies (Thread, Zigbee).
- Z-Wave
- Cellular Low-Power Wide Area Network Technologies (NB-IoT, LTE-M)
- Non-Cellular Low-Power Wide Area Network Technologies (LoRaWAN, Sigfox)

The attributes we'll be comparing for each of the technologies are:

- Range
- Throughput
- Power consumption
- Cost
- Topology

Before we cover how these different technologies compare in terms of the attributes, let's briefly introduce each technology. Going into detail for each of these technologies is out of scope for this article.

Bluetooth Technology

Definition

Bluetooth® technology is a low-power wireless solution that operates in the 2.4 GHz ISM band. It has expanded over the years and now provides tremendous flexibility in range, bandwidth, and communications topologies to address different IoT applications.

Technical Details

There are two different Bluetooth radio options: Bluetooth Classic and Bluetooth Low Energy (LE). Bluetooth Classic (or BR/EDR) is the original Bluetooth radio that's still widely used in streaming applications, especially audio streaming. Bluetooth LE, on the other hand, has traditionally focused on low-bandwidth applications that involve infrequent data transmission between devices. Bluetooth LE is known for its very low power consumption and its proliferation in smartphones, tablets, and PCs.

Bluetooth LE provides the option to operate in point-to-point, star, mesh, and broadcast topologies. In a mesh topology, nodes connect directly to each other without the need to communicate with others through a central hub. This allows nodes to relay data and information to other nodes out of reach from the original source node, extending the reach of the network in a large area.

Primary Use Cases

Bluetooth LE is most popular in health and fitness devices, smart lighting systems, real-time location systems, and indoor navigation applications.

FEATURED TOOL

The Bluetooth Range Estimator

Calculate the expected range between two Bluetooth devices.

TRY IT NOW

Wi-Fi

Definition

Wi-Fi is the trademark name for any Wireless Local Area Network (WLAN) that follows the IEEE 802.11 standard. It most commonly operates in the 2.4 and 5 GHz ISM Bands, but newer versions also target other frequency bands.

Technical Details

There are many variants of Wi-Fi, and the Wi-Fi Alliance has recently adopted a version numbering system: Wi-Fi 1 (802.11b), Wi-Fi 2 (802.11a), Wi-Fi 3 (802.11g), Wi-Fi 4 (802.11n), Wi-Fi 5 (802.11ac), and Wi-Fi 6 (802.11ax). More recently released versions were introduced to target different kinds of applications including longer range, higher throughput, and better coverage.

The most popular topology utilized in Wi-Fi is the star topology, in which nodes can only communicate with each other through a central hub.

Primary Use Cases

Wi-Fi is popular in applications involved with transferring large files and higher-bandwidth data transfer applications, such as video streaming.

In the case of IoT applications, Wi-Fi is most commonly used for devices that need a direct connection to the internet. However, it is not usually associated with low power consumption, and its use is limited in applications and devices that require running on small batteries for long periods of time.

IEEE 802.15.4-Based Technologies (Thread, Zigbee)

Definition

IEEE 802.15.4 technologies refer to the access layer for low-rate wireless personal area networks (LR-WPANs).

Technical Details

Thread and Zigbee are two different technologies built on top of this specification. They are characterized by their low power consumption and low data rates. IEEE 802.15.4 is used primarily for small amounts of data over a low range while maintaining low power consumption.

While a star topology is supported, the most popular topology utilized is the mesh topology.

Primary Use Cases

These technologies are most commonly used in wireless control and monitoring applications in the smart home space.

Z-Wave

Definition

Z-Wave started out as a protocol for controlling lighting systems and evolved into a home automation protocol managed by the Z-Wave Alliance.

Technical Details

It is a proprietary technology that operates in the 908/915 MHz band in the U.S. and 868 MHz in Europe. This is designed to avoid interference with the 2.4 GHz ISM band and extend coverage.

The primary topology utilized is the mesh topology.

Primary Use Cases

This technology's main use is in smart home applications.

FEATURED DOWNLOAD

Bluetooth Mesh Models – A Technical Overview

INSTANT DOWNLOAD

Cellular Low-Power Wide Area Network Technologies (NB-IoT, LTE-M)

Definition

LTE-M (LTE Cat-M1, or Long-Term Evolution for Machines) and NB-IoT (NarrowBand IoT) are both technology standards developed by 3GPP (3rd Generation Partnership Project) as cellular-based technology solutions for IoT applications.

These two technologies are poised as an integral part of the long-term 5G IoT strategy by allowing them to co-exist with other 5G technologies. 5G is the overarching term used to describe the fifth generation of cellular technology. It promises high speeds of 2 Gbps (and even up to 100 Gbps in the future). 5G technologies also promise reduced latency and wider coverage (in terms of the number of devices concurrently connected to a network).

Technical Details

LTE-M and NB-IoT differ in several attributes that make each of them suitable for different types of applications.

NB-IoT is ideal for simple applications with low power consumption and low bandwidth requirements, while LTE-M has a higher data rate and is best suited for real-time and mission-critical applications. The main differences between the two are speed (higher for LTE-M) and latency (lower for LTE-M).

NB-IoT and LTE-M operate primarily by design in a star topology.

Primary Use Cases

The main use cases of NB-IoT include smart agriculture, smart city, and smart meter applications. The main use cases of LTE-M, on the other hand, include logistics, healthcare devices as a backhaul communications channel, and in automotive applications.

Attribute	Bluetooth® Low Energy Technology	Wi-Fi	Z-Wave	IEEE 802.15.4 (Zigbee, Thread)	LTE-M	NB-IoT	Sigfox	LoRaWAN
Range	10 m – 1.5 km	15 m – 100 m	30 m - 50 m	30 m – 100 m	1 km – 10 km	1 km – 10 km	3 km – 50 km	2 km – 20 km
Throughput	125 kbps – 2 Mbps	54 Mbps – 1.3 Gbps	10 kbps – 100 kbps	20 kbps – 250 kbps	Up to 1 Mbps	Up to 200 kbps	Up to 100 bps	10 kbps – 50 kbps
Power Consumption	Low	Medium	Low	Low	Medium	Low	Low	Low
Ongoing Cost	One-time	One-time	One-time	One-time	Recurring	Recurring	Recurring	One-time
Module Cost	Under $5	Under $10	Under $10	$8-$15	$8-$20	$8-$20	Under $5	$8-$15
Topology	P2P, Star, Mesh, Broadcast	Star, Mesh	Mesh	Mesh	Star	Star	Star	Star
Shipments in 2019 (millions)	~3,500	~3,200	~120	~420	~7	~16	~10	~45

Non-Cellular Low-Power Wide Area Network Technologies (LoRaWAN, Sigfox)

Definition

LoRaWAN is an open wireless networking protocol maintained by the LoRa Alliance. LoRaWAN is built on top of LoRa, a proprietary modulation format developed by a company called Semtech.

Technical Details

LoRa only defines the lower-level layers of the network stack, and LoRaWAN defines the upper layers of the stack. LoRaWAN is simply one of several protocols built on top of LoRa.

LoRaWAN is categorized as a low-power wide-area network (LPWAN) technology. It enables long-range communication between devices while maintaining low-power consumption.

Sigfox is also an LPWAN technology, however, it's a proprietary technology that's offered by the French company Sigfox, which acts as the sole network operator for the technology.

Primary Use Cases

LoRaWAN is popular in smart city applications, such as smart utility meters, smart parking meters, and supply chain and logistics applications like asset tracking. Sigfox is also popular in smart city applications but more so in the EU region than in the U.S.

Comparison Table

In the following table, we compare the different wireless technologies mentioned previously.

Attribute	Bluetooth® Low Energy Technology	Wi-Fi	Z-Wave	IEEE 802.15.4 (Zigbee, Thread)	LTE-M	NB-IoT	Sigfox	LoRaWAN
Range	10 m – 1.5 km	15 m – 100 m	30 m - 50 m	30 m – 100 m	1 km – 10 km	1 km – 10 km	3 km – 50 km	2 km – 20 km
Throughput	125 kbps – 2 Mbps	54 Mbps – 1.3 Gbps	10 kbps – 100 kbps	20 kbps – 250 kbps	Up to 1 Mbps	Up to 200 kbps	Up to 100 bps	10 kbps – 50 kbps
Power Consumption	Low	Medium	Low	Low	Medium	Low	Low	Low
Ongoing Cost	One-time	One-time	One-time	One-time	Recurring	Recurring	Recurring	One-time
Module Cost	Under $5	Under $10	Under $10	$8-$15	$8-$20	$8-$20	Under $5	$8-$15
Topology	P2P, Star, Mesh, Broadcast	Star, Mesh	Mesh	Mesh	Star	Star	Star	Star
Shipments in 2019 (millions)	~3,500	~3,200	~120	~420	~7	~16	~10	~45

In the subsequent articles within the series, we will cover a wide variety of industrial applications, list the most important attributes for each application, and assess the most suitable technologies for that application.

Connection security. Whether data is being sent over devices, networks, or applications, it should be encrypted. Otherwise, sensitive information can be read by anybody who intercepts information in transit. IoT-centric messaging protocols like MQTT, AMQP, and DDS may use standard Transport Layer Security (TSL) cryptographic protocol to ensure end-to-end data protection.

Cloud security. Data at rest stored in the cloud must be encrypted as well to mitigate risks of exposing sensitive information to intruders. Cloud security also involves authentication and authorization mechanisms to limit access to the IoT

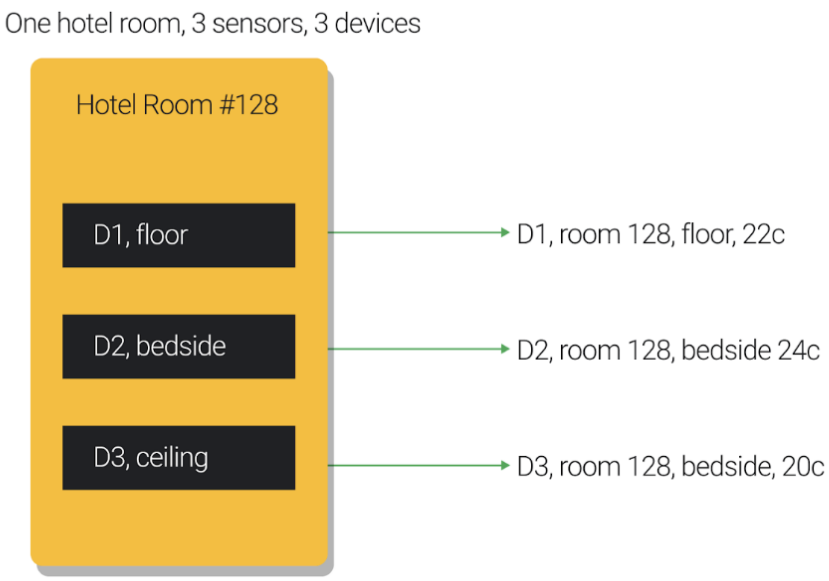
applications. Another important security method is device identity management to verify the device's credibility before allowing it to connect to the cloud.

The good news is that IoT solutions from large providers like Microsoft, AWS, or Cisco come with pre-built protection measures including end-to-end data encryption, device authentication, and access control. However, it always pays to ensure that security is tight at all levels, from the tiniest devices to complex analytical systems.

In IoT, the definition of a device can change depending on the needs of the project. You want to think about levels of abstraction as you design your project. There may be times when you want to consider each device as a separate entity, and other times when you want to consider a group of sensors as a single reporting device.

The specific requirements of your application will help you understand whether something that generates information should be treated as a device, and therefore deserves its own ID, or is simply a channel or state detail of another device.

For example, consider a project for monitoring the temperature in hotel rooms. Each room has three sensors: one near the floor, one near the bed, and one near the ceiling.



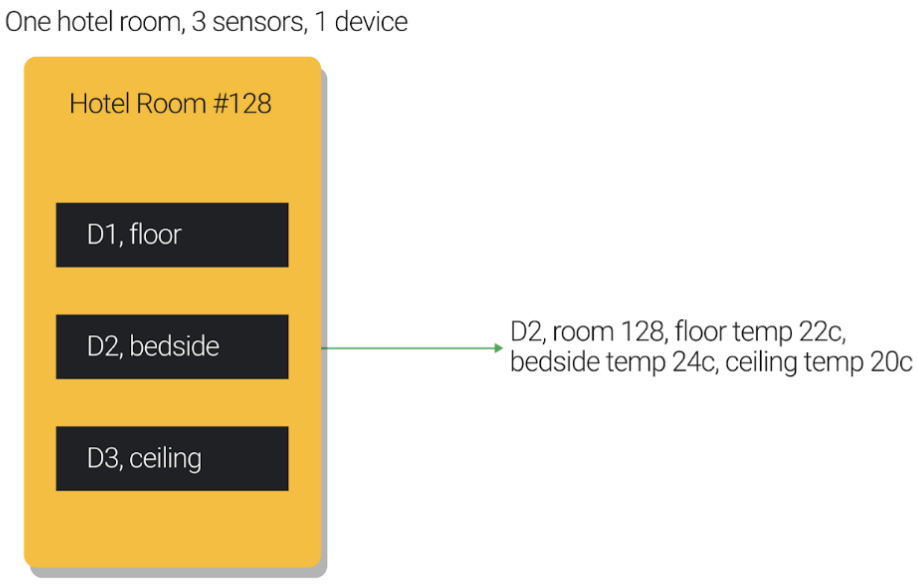
In this design, data is sent to the cloud as three different devices, each sending temperature information to the cloud.

{deviceID: "dh28dslkja", "location": "floor", "room": 128, "temp": 22 }

{deviceID: "8d3kiuhs8a", "location": "ceiling", "room": 128, "temp": 24 }

{deviceID: "kd8s8hh3o", "location": "bedside", "room": 128, "temp": 23 }

Or, you can model the room as a single device, send the data for the entire room.



In this design the data is sent to the cloud as a single device with three sensors. Note that an additional field has been added to the data, 'average temperature.

{deviceID: "dh28dslkja", "room": 128, "temp_floor": 22, "temp_ceiling": 24, "temp_bedside": 23, "average_temp": 23 }

The design you choose will depend upon how you intend to use the information, now and potentially in the future.

What can be done with IIoT?

Today, companies are using IoT devices to perform a wide variety of tasks. As companies find new use cases for IoT, the industry will continue to grow. Some of the more wide spread uses of IIoT are listed below.

Accelerate business agility

When you combine a global network with an intelligent IoT platform, you can unlock valuable business insights. You can accelerate business agility by connecting globally dispersed devices, at the edge or in the cloud, with comprehensive cloud services. For example:

Real-time asset tracking: Real-time asset tracking: Embed devices in valuable assets and track them in real time, perform complex analytics and machine learning on the data collected, and assess the status of your business to deliver actionable insights.

Machine learning on the edge

You can also run IoT solutions with machine learning capabilities both locally on the device (using Tensorflow and a TPU board) and in the cloud. For example:

Predictive maintenance: Embed sensors in equipment and automatically predict when equipment needs maintenance; optimize equipment performance in real time; predict downtime; detect anomalies; and track device status, state, and location.

Machine learning is beyond the scope of this course; it will be discussed in a later IoT course.

Improve operational efficiency

When your device is connected to the cloud, you can manage global assets and perform firmware updates. Discover how efficiently your devices operate, manage global assets, and carry out firmware updates on Cloud IoT. The platform supports a wide variety of embedded operating systems and provides out-of-the-box support for devices from leading manufacturers like Intel and Microchip. Plus, you can trigger automatic changes based on real-time events using Cloud Functions workflows.

For example:

Logistics and supply chain management: Embedding cloud connected sensors and devices in company transport vehicles can improve the management of the fleet, inventory tracking, and cargo integrity monitoring.

Localization intelligence

IoT devices allow you to visualize where assets are located in real time, where they've traveled, and how often they've moved. Whether your IoT assets are indoors, in remote areas, or distributed across hundreds of cities, you can track them with precision.

Smart Cities and buildings: Embed cloud-connected sensors and devices in buildings and infrastructure. Build a comprehensive solution that spans across billions of sensors and

edge devices and bring a new level of intelligence and automation to entire homes, buildings, or cities.