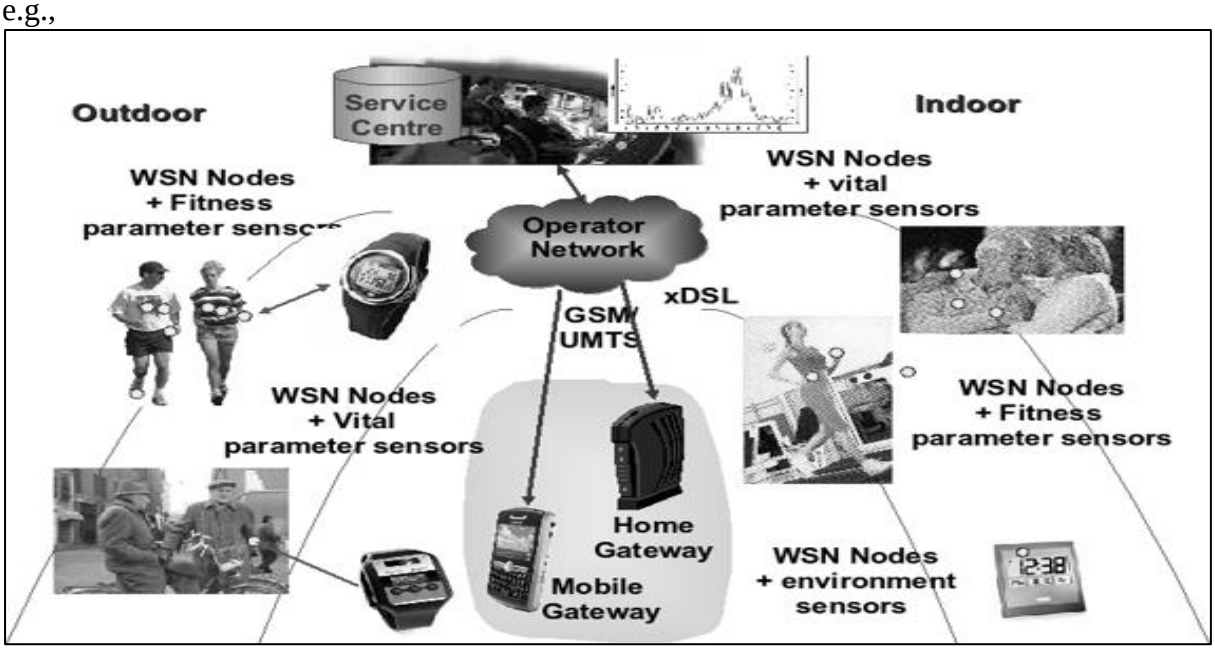
Internet of Everything (IoE)

Wireless Sensor Networks

Def:

Wireless Sensor Networks (WSNs) are the networks that consists of sensors which are distributed in an ad-hoc manner. These sensors work with each other to sense some physical phenomenon and then the information gathered is processed to get relevant results. WSNs consist of protocols and algorithms with self-organizing capabilities.

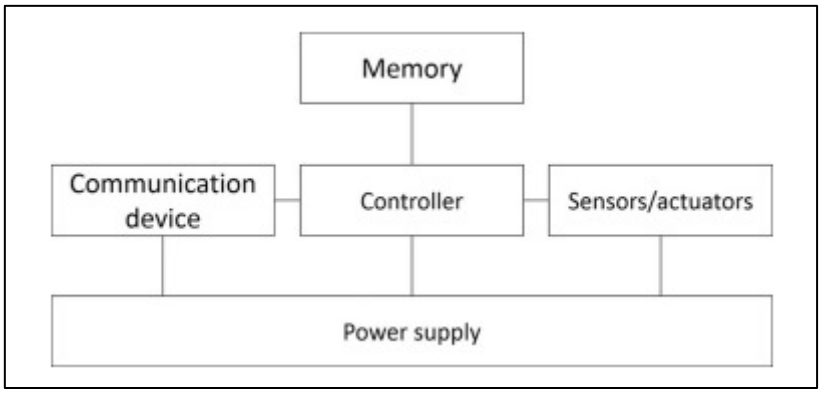
e.g.,



Sensor Node:

Wireless sensor nodes are battery-powered devices, since it is generally difficult or impossible to run a mains supply to their deployment site. Power to the wireless sensor nodes is usually provided through primary batteries. Although the utilization of WSNs has increased continuously over the years, battery technology has not improved at the same rate. Therefore, batteries are seen as the limitation of wireless sensor nodes.

Components of sensor node:



Memory:
Some sensors need memory to store programs and data. In general, different types of memory are used: RAM, ROM, or EEPROM. Due to power restrictions in wireless sensor networks, it is important to consider the time required in read and write operations, due to the amount of energy needed to perform them.

Communication Device:
The communication device is used to exchange data between individual nodes. In some cases, cable communication is the method of choice and is often applied in many network-like environments. Wireless communication can be carried out by means of radio frequencies, optical communication, or ultrasound. Other media such as magnetic inductance is only used in very specific cases.

Controller:
The controller is a driver to process all relevant data, capable of executing arbitrary code. A microcontroller is a small integrated circuit, usually composed of a central processing unit, memory, parallel input and output interfaces, a clock generator, one or more analog-to-digital converters and serial communication interfaces. They are recommended for stand-alone applications, since they offer programming flexibility.

Sensors/actuators:
Different types of sensors can be used in a WSN. These are classified into three categories:
Passive, omnidirectional sensors: These sensors can measure a physical magnitude; they have only one active prove.
Passive, narrow-beam sensors: These sensors are passive and have a well-defined idea of the direction of the measurement. A typical example is a camera, which can "take action" in a given direction and allows you to rotate if necessary.
Active sensors: This last group of sensors actively probes the environment; for example, a sonar or radar sensor or some types of seismic sensors, which generate shock waves using small explosions.
As with sensors, a variety of actuators can be used, but for a WSN one must bear in mind that, in general, the sensor node can only open/close a switch.

Power supply:
Power to feed the sensor nodes can be supplied from batteries, solar cells, etc. The means of feeding each sensor is directly proportional to the life of the network, so be careful when selecting a protocol that optimizes its use to the maximum. It is important to consider the storage and supply of energy that is required, in addition to the replenishment of energy consumed.

Networking Node:
Wireless sensor networks mainly use broadcast communication while ad hoc networks use point-to-point communication. Unlike ad hoc networks wireless sensor networks are limited by sensors limited power, energy and computational capability. Sensor nodes may not have global ID because of the large amount of overhead and large number of sensors.

Motes:

Motes mainly consist of three parts:

- Mote basically consists of a low cost and power computer.

- The computer monitors one or more sensors. Sensors may be for temperature, light, sound, position, acceleration, vibration, stress, weight, pressure, humidity, etc.
- The computer connects to the outside world with a radio link.

Examples of Motes:
Mica 2 Motes: These motes sold by Crossbow were originally developed at the University of California Berkeley. The MICA2 motes are based on the ATmega128L AVR microprocessor. The motes run using TinyOS as the operating system. Mica2 mote is one of the most popular and commercially available sensors which are marketed by CrossBow technologies.

Telosb Motes: Telosb motes have USB programming capability. An IEEE 802.15.4 compliant, high data rate radio with integrated antenna, a low-power MCU. There are also equipped with extended memory and an optional sensor suite.

Design Challenges of WSN:
- **Heterogeneity**: The devices deployed maybe of various types and need to collaborate with each other.
- **Distributed Processing:** The algorithms need to be centralized as the processing is carried out on different nodes.
- **Low Bandwidth Communication**: The data should be transferred efficiently between sensors.
- **Large Scale Coordination**: The sensors need to coordinate with each other to produce required results.
- **Utilization of Sensors**: The sensors should be utilized in a way that produce the maximum performance and use less energy.
- **Real Time Computation:** The computation should be done quickly as new data is always being generated.

Operational Challenges of WSN:
- Energy Efficiency
- Limited storage and computation
- Low bandwidth and high error rates
- Errors are common
  - Wireless communication
  - Noisy measurements
  - Node failure are expected
- Scalability to a large number of sensor nodes
- Survivability in harsh environments
- Experiments are time- and space-intensive

Applications of WSN:
1. Monitoring of objects
   - Structural Monitoring Eco-Physiology Condition-based Maintenance
   - Medical Diagnostics Urban terrain mapping
2. Monitoring of Area
   - Environmental and Habitat Monitoring Precision Agriculture
   - Indoor Climate Control Military Surveillance
   - Treaty Verification Intelligent Alarms
3. Monitoring of both area and objects
   - Wildlife Habitats Disaster Management Emergency Response

- Ubiquitous Computing Asset Tracking
- Health Care Manufacturing Process Flows

Standards of WSN:

- The IEEE 802.15.4 low-rate wireless personal area network (LR-WPAN) standard research a low data rate solution with multi-year battery life and very low complexity. It intended to operate in an unlicensed, international frequency band. The eighteenth draft of this standard was accepted in MAY 2003.
- This standard defines the physical and MAC layer specifications for sensor and other WPAN networks. Low power consumption is an important feature targeted by the standard. This requires reduced transmission rate, power efficient modulation techniques, and strict power management techniques such as sleep modes.
- Other standard, SensIT project by DARPA which focuses on large distributed military system.

TinyOS: TinyOS is an open source operating system used for wireless devices. This operating system (OS) is small in size and consumes low memory. The application programs that run on TinyOS are also small in size as compared to normal OS. Another feature of TinyOS is that it is made for some specific device. Normal OS is multithreaded and consumes high voltage of computer. But TinyOS consumes low battery. Normal OS that we use in our computer supports all available devices and has the large source code. But TinyOS has the small source code and it is written in nesC language. nesC is a computer language derived from C language. nesC has a separate compiler.

TinyOS does not support multithreaded applications because it has low computation power. Main components of TinyOS are tasks, events, commands and data. Data is getting from the outside environment. For example, in the smoke detection device, when smoke is detected then sensor of the device creates event and information about that event is data. Then the command is made which splits water or any chemical which stops smoke or fire.

Applications of TinyOS:

Many applications of TinyOS are there that we see in our daily life. Some of the examples of devices that use TinyOS are as follows-

- Smoke detection device
- Used in military activities
- Temperature control like AC in the room or in car
- Security system in the bank
- Resource monitoring
- Environmental monitoring
- Industrial measurement
- Supports Bluetooth devices
- Used in a microwave oven
- Used in agriculture for detecting fault crop

Networking and the Internet:

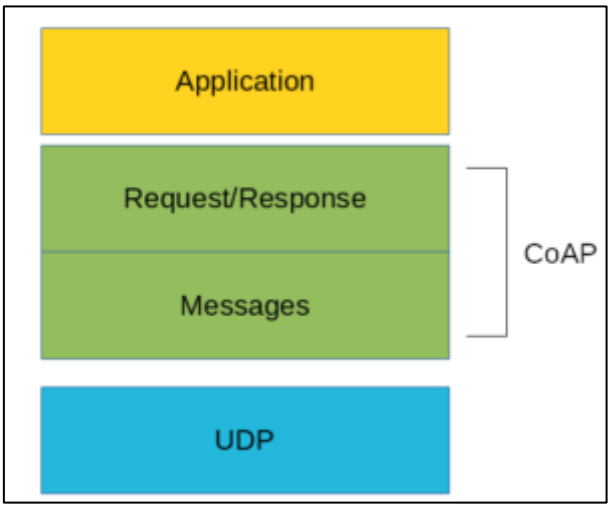
IP Address: An Internet Protocol address (IP address) is a numerical label assigned to each device connected to a computer network that uses the Internet Protocol for communication. An IP address serves two main functions: host or network interface identification and location addressing. Internet Protocol version 4 (IPv4) defines an IP address as a 32-bit number. However, because of the growth of the Internet and the depletion of available IPv4 addresses, a new version of IP (IPv6), using 128 bits for the IP address, was standardized in

1998. IPv6 deployment has been ongoing since the mid-2000s. IP addresses are written and displayed in human-readable notations, such as 172.16.254.1 in IPv4, and 2001:db8:0:1234:-0:567:8:1 in IPv6. The size of the routing prefix of the address is designated in CIDR notation by suffixing the address with the number of significant bits, e.g., 192.168.1.15/24, which is equivalent to the historically used subnet mask 255.255.255.0.

Protocols:
MQTT
Refer Assignment No.3.

CoAP:
CoAP is an IoT protocol. CoAP stands for Constrained Application Protocol, and it is defined in RFC 7252. CoAP is a simple protocol with low overhead specifically designed for constrained devices (such as microcontrollers) and constrained networks. This protocol is used in M2M data exchange and is very similar to HTTP.
The main features of CoAP protocols are:

- Web protocol used in M2M with constrained requirements
- Asynchronous message exchange
- Low overhead and very simple to parse
- URI and content-type support
- Proxy and caching capabilities



As you can see there are two different layers that make CoAP protocol: Messages and Request/Response. The Messages layer deals with UDP and with asynchronous messages. The Request/Response layer manages request/response interaction based on request/response messages.

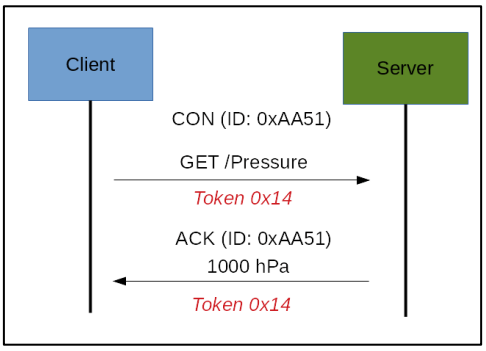
CoAP Messages Model:
This is the lowest layer of CoAP. This layer deals with UDP exchanging messages between endpoints. Each CoAP message has a unique ID; this is useful to detect message duplicates. A CoAP message is built by these parts:

- A binary header
- A compact option
- Payload

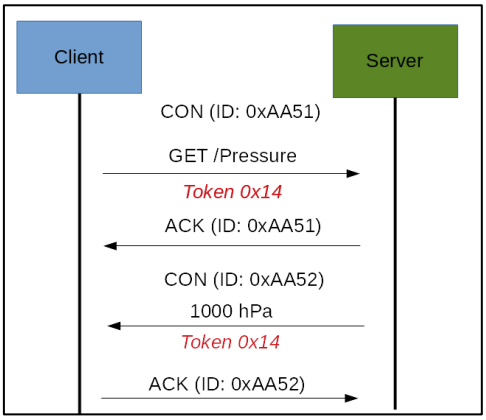
CoAP protocol uses two kinds of messages:
Confirmable message
Non-Confirmable message

A confirmable message is a reliable message. When exchanging messages between two endpoints, these messages can be reliable. In CoAP, a reliable message is obtained using a Confirmable message (CON). Using this kind of message, the client can be sure that the message will arrive at the server. A Confirmable message is sent again and again until the other party sends an acknowledge message (ACK). The ACK message contains the same ID of the confirmable message (CON).

CoAP Request/Response Model:
The CoAP Request/Response is the second layer in the CoAP abstraction layer. The request is sent using a Confirmable (CON) or Non-Confirmable (NON) message. There are several scenarios depending on if the server can answer immediately to the client request or the answer if not available.
If the server can answer immediately to the client request, then if the request is carried using a Confirmable message (CON), the server sends back to the client an Acknowledge message containing the response or the error code.

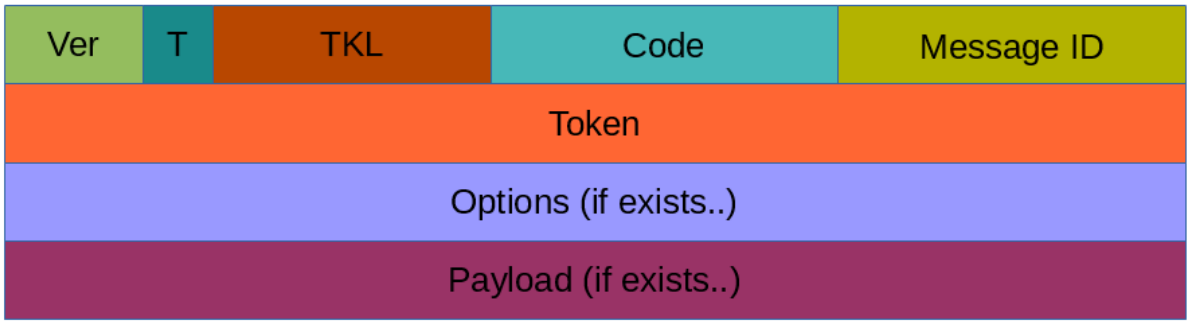


If the server can't answer to the request coming from the client immediately, then it sends an Acknowledge message with an empty response. As soon as the response is available, then the server sends a new Confirmable message to the client containing the response. At this point, the client sends back an Acknowledge message.



If the request coming from the client is carried using a NON-confirmable message, then the server answer using a NON-confirmable message.

CoAP Message Format:
The constrained application protocol is the meat for constrained environments, and for this reason, it uses compact messages. To avoid fragmentation, a message occupies the data section of a UDP datagram. A message is made by several parts:



Where:
Ver: It is a 2 bit unsigned integer indicating the version
T: It is a 2 bit unsigned integer indicating the message type: 0 confirmable, 1 non-confirmable
TKL: Token Length is the token 4 bit length
Code: It is the code response (8 bit length)
Message ID: It is the message ID expressed with 16 bit
And so on.

CoAP Security Aspects
One important aspect when dealing with IoT protocols is the security aspects. As stated before, CoAP uses UDP to transport information. CoAP relies on UDP security aspects to protect the information. As HTTP uses TLS over TCP, CoAP uses Datagram TLS over UDP. DTLS supports RSA, AES, and so on. Anyway, we should consider that in some constrained devices some of DTLS cipher suits may not be available. It is important to notice that some cipher suites introduce some complexity and constrained devices may not have resources enough to manage it.

REST
Refer Assignment No.3.