**IOT**

**1a) Give the Defination of IOT?**

The internet of things (IoT) is a computing concept that describes the idea of everyday physical objects being connected to the internet and being able to identify themselves to other devices. It has dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual “things” have identities, physical attributes and virtual network and use intelligent interfaces.

**1b) Define Embedded system.**

The embedded devices are the objects that build the unique computing system. These systems may or may not connect to the Internet. An embedded device system generally runs as a single application. However, these devices can connect through the internet connection, and able communicate through other network devices.

The embedded system can be of type microcontroller or type microprocessor. Both of these types contain an integrated circuit (IC). The essential component of the embedded system is a RISC family microcontroller like Motorola 68HC11, PIC 16F84, Atmel 8051 and many more. The most important factor that differentiates these microcontrollers with the microprocessor like 8085 is their internal read and writable memory.

**1c) What is Big Data Analytics?**

Big data analytics refers to the strategy of analyzing large volumes of data, or big data. This big data is gathered from a wide variety of sources, including social networks, videos, digital images, sensors, and sales transaction records. The large data is difficult to store, manage, process and analyze the data using traditional databases and data processing tools. There are several steps which involves in analyzing big data are data cleansing, data managing, data processing and visualization.

**1d) Define Smart Irrigation.**

* + Smart irrigation system can improve crop yields while saving water.
	+ Smart irrigation systems use IoT devices with soil moisture sensors to determined the amount of moisture on the soil and release the flow of the water through the irrigation pipes only when the moisture levels go below a predefined threshold.
	+ It also collect moisture level measurements on the server on in the cloud where the collected data can be analyzed to plan watering schedules.

1e) What is fleet tracking?

**Fleet tracking** is the use of a vehicle **tracking** device and software to monitor and manage a **fleet** of vehicles. Through **fleet tracking**, a **fleet** manager or business owner can collect data on location and vehicle health, as well as driving behavior and **fleet** productivity.

**1f) How to monitor Air Pollution?**

* IoT based air pollution monitoring system can monitor emission of harmful gases by factories and automobiles using gaseous and meteorological sensors.
* The collected data can be analyzed to make informed decisions on pollutions control approaches.

**1g) What is pcDuino?**

PCDuino = “PC + Arduino”

This board is similar to the Raspberry Pi, it’s becoming like the new choice of developers. It has all the features of the Raspberry Pi but it’s more powerful. Which ultimately makes it more expensive. It already comes with 2GB of flash memory. And you can add an SD card up to 32GB. Which is a really advantage over the Rpi.

**1h) What is an IOT Device.**

* + A "Thing" in Internet of Things (IoT) can be any object that has a unique identiﬁer and which can send/receive data (including user data) over a network (e.g., smart phone, smart TV, computer, refrigerator, car, etc. ).
	+ IoT devices are connected to the Internet and send information about themselves or about their surroundings (e.g. information sensed by the connected sensors) over a network (to other devices or servers/storage) or allow actuation upon the physical entities/environment around them remotely.

**1i) Define internet of Everything?**

The **Internet of Everything** (IoE) is a concept that extends the **Internet** of Things (IoT) emphasis on machine-to-machine (M2M) communications to describe a more complex system that also encompasses people and processes.

**1j) What is the use Big Data?**

**Big data** has been **used** in the industry to provide customer insights for transparent and simpler products, by analyzing and predicting customer behavior through **data** derived from social media, GPS-enabled devices, and CCTV footage. The **big data** also allows for better customer retention from insurance companies.

* Familiarize yourself with and understand industry-specific challenges.
* Understand or know the data characteristics of each industry.
* Understand where spending is occurring.
* Match market needs with your own capabilities and solutions.

**2a) What are IOT Communication models? Explain**

* **Request-Response**



 In Request–Response communication model client sends request to the server and the server responds to the request. When the server receives the request it decides how to respond, fetches the data, retrieves resources, and prepares the response and sends to the client. R-R is a communication model where request and response pair is independent of each others.

* **Publish-Subscribe:**

This model involves publishers, brokers and consumers. Publishers are the sources of data. Publishers send the data to the topic which are managed by the broker. Publishers are not aware of the consumers. Consumers subscribe to the topics which are managed by the broker. When broker receives the data from the Publisher, it sends to all the consumers.



* **Push-Pull:**

In this model the producers push the data in queues and the consumers pull the data from the queues. Producers do not need to be aware of the consumers. Queues help in decoupling the messaging between the producer and consumers. Queues also act as buffer which helps in situation when there is mismatch between the rate at which the producers push the data and consumers pull the data. 

* **Exclusive Pair:**

Exclusive pair is a bi-directional, fully duplex communication model that uses a persistent connection between the client and server, once the connection is established it remains open until the client sends a request to closer the connection. Client and server can send the message to each other after connection setup. In this model server is aware of all the open connection.



**2b) Explain about IOT Communication API’s in details**

The application program (or programming) interface, or API, is arguably what really ties together the connected “things” of the “internet of things.” IoT APIs are the points of interaction between an IoT device and the internet and/or other elements within the network.

As API management company Axway puts it, “APIs are tightly linked with IoT because they allow you to securely expose connected devices to customers, go-to-market channels and other applications in your IT infrastructure.”

.

Generally we used Two APIs For IoT Communication. These IoT Communication APIs are:

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

**REST-based Communication APIs**

Representational state transfer (REST) is a set of architectural principles by which you can design Web services the Web APIs that focus on systems’s resources and how resource states are addressed and transferred. REST APIs that follow the request response communication model, the rest architectural constraint apply to the components, connector and data elements,  within a distributed hypermedia system.  The rest architectural constraint are as follows:

**Client-server –**The principle behind the client-server constraint is the separation of concerns. for example clients should not be concerned with the storage of data which is concern of the serve. Similarly the server should not be concerned about the user interface, which is concern of the clien.  Separation allows client and server to be independently developed and updated.

**Stateless** – Each request from client to server must contain all the information necessary to understand the request, and cannot take advantage of any stored context on the server. The session state is kept entirely on the client.

**Cache-able** – Cache constraints requires that the data within a response to a request be implicitly or explicitly leveled as cache-able or non cache-able. If a response is cache-able, then a client cache is given the right to reuse that repsonse data for later, equivalent requests. caching can partially or completely eliminate some instructions and improve efficiency and scalability.

**Layered system** – layered system constraints, constrains the behavior of components such that each component cannot see beyond the immediate layer with they are interacting. For example, the client cannot tell whether it is connected directly to the end server or two an intermediaryalong the way. System scalability can be improved by allowing intermediaries to respond to requests instead of the end server, without  the client having to do anything different.

**Uniform interface**– uniform interface constraints requires that the method of communication between client and server must be uniform. Resources are identified in the requests (by URIsin web based systems) and are themselves is separate from the representations of the resources data returned to the client. When a client holds a representation of resources it has all the information required to update or delete the resource you (provided the client has required permissions). Each message includes enough information to describe how to process the message.

**Code on demand** – Servers can provide executable code or scripts for clients to execute in their context. this constraint is the only one that is optional.

A RESTful web service is a ” Web API ” implemented using HTTP and REST principles. REST is most popular IoT Communication APIs.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Uniform Resource Identifier (URI)** | **GET** | **PUT** | **PATCH** | **POST** | **DELETE** |
| **Collection, such as https://api.example.com/resources/** | *List* the URIs and perhaps other details of the collection’s members. | *Replace* the entire collection with another collection. | Not generally used | *Create* a new entry in the collection. The new entry’s URI is assigned automatically and is usually returned by the operation. | *Delete* the entire collection. |
| **Element, such as https://api.example.com/resources/item5** | *Retrieve* a representation of the addressed member of the collection, expressed in an appropriate Internet media type. | *Replace* the addressed member of the collection, or if it does not exist, create it. | *Update* the addressed member of the collection. | Not generally used. Treat the addressed member as a collection in its own right and create a new entry within it. | *Delete* the addressed member of the collection. |
| HTTP methods |

**WebSocket based communication API**

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



2c) Discuss in details about IOT Level 6

**IoT level-6:**

* A level-6 IoT system has multiple independent end nodes that perform sensing and/or actuation and send data to the cloud. Data is stored in the cloud and the application is cloud-based. The analytics component analyzes the data and stores the results in the cloud database.
* The results are visualized with the cloud-based application. The centralized controller is aware of the status of all the end nodes and sends control commands to the nodes.



**2d) Describe home automation in domain specific IOT.**

**IoT applications for smart homes automation:**

* Smart Lighting
* Smart Appliances
* Intrusion Detection
* Smoke / Gas Detectors

**Home Automation Smart Lighting**

* Smart lighting achieves energy savings by sensing the human movements and their environments and controlling the lights accordingly.
* Key enabling technologies for smart lighting include :

 - Solid state lighting (such as LED lights)

- IP-enabled lights

* Wireless-enabled and Internet connected lights can be controlled remotely from IoT applications such as a mobile or web application.
* Paper:

- Energy-aware wireless sensor network with ambient intelligence for smart LED lighting system control [IECON, 2011]-> presented controllable LED lighting system that is embedded with ambient intelligence gathered from a distributed smart WSN to optimize and control the lighting system to be more efficient and user-oriented.

**Home Automation Smart Appliances**

* Smart appliances make the management easier and provide status information of appliances to the users remotely. E.g: smart washer/dryer that can be controlled remotely and notify when the washing/drying cycle is complete.
* **Open Remote** is an open source automation platform for smart home and building that can control various appliances using mobile and web applications.
* It comprises of three components:

- a Controller-> manages scheduling and runtime integration between devices.

- a Designer -> allows to create both configuration for the controller and user interface designs.

- Control Panel -> allows to interact with devices and control them.

* Paper: - An IoT-based Appliance Control System for Smart Home [ICICIP, 2013] implemented an IoT based appliance control system for smart homes that uses a smart-central controller to set up a wireless sensor and actuator network and control modules for appliances

**Home Automation Intrusion Detection**

* Home intrusion detection systems use security cameras and sensors to detect intrusions and raise alerts.
* The form of the alerts can be in form: - SMS - Email - Image grab or a short video clip as an email attachment
* Papers :

 - Could controlled intrusion detection and burglary prevention stratagems in home automation systems [BCFIC, 2012] ->present a controlled intrusion detection system that uses location-aware services, where the geo-location of each node of a home automation system is independently detected and stored in the cloud

 - An Intelligent Intrusion Detection System Based on UPnP Technology for Smart Living [ISDA, 2008] -> implement an intrusion detection system that uses image processing to recognize the intrusion and extract the intrusion subject and generate Universal-Plug-and-Play (UPnP-based) instant messaging for alerts.

**Home Automation Smoke / Gas Detectors**

* Smoke detectors are installed in homes and buildings to detect smoke that is typically an early sign of fire.
* It uses optical detection, ionization or air sampling techniques to detect smoke
* The form of the alert can be in form :

 Signals that send to a fire alarm system

* Gas detector can detect the presence of harmful gases such as carbon monoxide (CO), liquid petroleum gas (LPG), etc.
* Paper :

- Development of Multipurpose Gas Leakage and Fire Detector with Alarm System [TIIEC, 2013]-> designed a system that can detects gas leakage and smoke and gives visual level indication.

**2e) Explain about wearable electronics.**

**Health & Lifestyle Wearable Electronics**

* + Wearable electronics such as wearable gadgets (smart watch, smart glasses, wristbands, etc) provide various functions and features to assist us in our daily activities and making us lead healthy lifestyles.
	+ Using the smart watch, the users can search the internet, play audio/video files, make calls, play games, etc.
	+ Smart glasses allows users to tae photos and record videos, get map directions, check flight status or search internet using voice commands
	+ Smart shoes can monitor the walking or running speeds and jumps with the help of embedded sensors and be paired with smart-phone to visualize the data.
	+ Smart wristbands can tract the daily exercise and calories burnt.

**2f) Give difference between IOT and M2M.**

* + Communication Protocols

• M2M and IoT can differ in how the communication between the machines or devices happens.

• M2M uses either proprietary or non-IP based communication protocols for communication within the M2M area networks.

* + Machines in M2M vs Things in IoT

• The "Things" in IoT refers to physical objects that have unique identiﬁers and can sense and communicate with their external environment (and user applications) or their internal physical states.

• M2M systems, in contrast to IoT, typically have homogeneous machine types within an M2M area network.

* + Hardware vs Software Emphasis

 • While the emphasis of M2M is more on hardware with embedded modules, the emphasis of IoT is more on software.

* + Data Collection & Analysis

• M2M data is collected in point solutions and often in on-premises storage infrastructure.

• In contrast to M2M, the data in IoT is collected in the cloud (can be public, private or hybrid cloud).

* + Applications

• M2M data is collected in point solutions and can be accessed by on-premises applications such as diagnosis applications, service management applications, and on- premisis enterprise applications.

• IoT data is collected in the cloud and can be accessed by cloud applications such as analytics applications, enterprise applications, remote diagnosis and management applications, etc.

M2M and IoT both are used when electronic devices are connected and share data with each other. There are some differences between IoT and M2M based on technologies, system architectures and types of applications

|  |  |
| --- | --- |
| M2M | IoT |
| It is Machine to Machine communication and completely hardware based. | It’s Machine to Machine, Machine to sensors, or Humans to Machines. And software based. |
| M2M is a point to point communication and uses non –IP protocols. | Its uses IP networks and protocols as the communication is multipoint. |
| These devices don’t rely on internet. | Devices required internet connections. |
| Data can be stored locally | Data can be stored locally and also in cloud |
| Limited integration option devices must have corresponding communication standards | Unlimited integration option, but requires a solutions that can manage all the communication |

**2G) Describe case study on IOT System for Weather Monitoring.**

* It collects data from a number of sensor attached such as temperature, humidity, pressure, etc and send the data to cloud-based applications and store back-ends.
* The data collected in the cloud can then be analyzed and visualized by cloud-based applications.
* Weather alert can be sent to the subscribed users from such applications.
* AirPi is a weather and air quality monitoring kit capable of recording and uploading information about temperature, humidity, air pressure, light levels, UV levels, carbon monoxide, nitrogen dioxide and smoke level to the Internet.
* Paper:

- PeWeMoS – Pervasive Weather Monitoring System [ICPCA, 2008]-> Presented a pervasive weather monitoring system that is integrated with buses to measure weather variables like humidity, temperature, and air quality during the bus path

2h) Explain about interfacing an LED and switch with raspbeery pi.

**Raspberry Pi Interfaces**

* + **Serial**

 • The serial interface on Raspberry Pi has receive (Rx) and transmit (Tx) pins for communication with serial peripherals.

* + **SPI**

 • Serial Peripheral Interface (SPI) is a synchronous serial data protocol used for communicating with one or more peripheral devices.

* + **I2C**

 • The I2C interface pins on Raspberry Pi allow you to connect hardware modules. I2C interface allows synchronous data transfer with just two pins - SDA (data line) and SCL (clock line).

Interfacing LED and switch with Raspberry Pi

from time import sleep

import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BCM)

#Switch Pin

GPIO.setup(25, GPIO.IN)

#LED Pin

GPIO.setup(18, GPIO.OUT)

state=false

def toggleLED(pin):

 state = not state

 GPIO.output(pin, state)

while True:

 try:

 if (GPIO.input(25) == True):

 toggleLED(pin)

 sleep(.01)

 except KeyboardInterrupt:

 exit()

2i) Describe functional view specification of IOT Design methodology

**Functional View Specification**

* + Define the functions of IoT System grouped into various functional groups.

These functional groups provide functionalities for interacting with the concepts defined in Domain model specification.

* + The seventh step in the IoT design methodology is to define the Functional View. The Functional View (FV) defines the functions of the IoT systems grouped into various Functional Groups (FGs). Each Functional Group either provides functionalities for interacting with instances of concepts defined in the Domain Model or provides information related to these concepts.



**2J) Discuss about visualization in IOT.**

Internet of things (IoT) is about capturing and making sense of data from the physical world. Soon, every physical device will be a source of data and this data can potentially become a source of business value. We have all heard the predictions about the explosion of IoT data and the IoT data deluge. All of this IoT data is useless unless it is collected, understood and used to make smart decisions.

Real-time visualization helps improve response and event management by aggregating numerous IoT data streams from various systems, sensors, vehicles and video, providing an integrated operational view across large physical environments – like a sports stadium. Live Earth is just such a platform.

For IoT data streams to be useful and productive, visualization systems must execute the following:

* Develop situational awareness by presenting data streams in context within a physical environment making data actionable
* Create location-based intelligence by presenting multiple concurrent data streams
* Support real-time decision making by combining multiple data sources into a single multi-layered visual display
* Enable multiple data correlations to be viewed and analyzed in real time
* Combine new IoT data with existing data to discover new business patterns and trends
* Monitor IoT devices and infrastructure to ensure stable and proper operation, ensuring IoT data flows continuously without interruption

**2k) What is mesh networking? explain in details.**

A wireless mesh network is an infrastructure of nodes (a mesh topology) that are wirelessly connected to each other. These nodes piggyback off each other to extend a radio signal (like a Wi-Fi or cellular connection) to route, relay, and proxy traffic to/from clients. Each node spreads the radio signal a little further than the last, minimizing the possibility of dead zones.

Different component in mesh network.

1. **Gateway** — Border routers are the devices that have additional connectivities beyond mesh that allow them to pass messages between networks. You can think of these devices as providing a “backhaul” to the internet for the local mesh network.
2. **Repeater** — Routers are devices that forward messages between end devices (endpoints) in a mesh network. They are not typically designed to sleep because they are a part of the mesh networks’ infrastructure.
3. **Endpoint** — End devices are mesh-only devices that do not route messages for other devices in the mesh network. Because they have no networking responsibilities, they can enter sleep mode and are good candidates for battery-powered nodes and sensors.

It should be noted that each mesh networking solution works differently. So for this article, will focus on how Particle Mesh technology works. Particle Mesh is a wireless mesh network designed to connect the spaces in between existing Wi-Fi and cellular deployments with local networks that are low-cost, secure, and reliable.

Traditional IoT devices that use Wi-Fi and cellular connectivity depend on the cloud to relay messages between devices. This works great when you’re making a standalone product — but sometimes you need more than that. Particle Mesh development kits aren’t just connected to the Internet, they’re gateways to the Internet and create a local wireless mesh that other devices can join. These devices work together to ensure that messages get where they’re going, and power products that aren’t possible or economically feasible with Wi-Fi and cellular connectivity. Particle Mesh gives every IoT device a local network to understand and connect with the world around it, ensuring products have the information they need.

**2L Explain about Bluetooth low energy.**

**Bluetooth Low Energy**
Bluetooth Low Energy is the intelligent, power-friendly version of Bluetooth wireless technology. It is already playing a significant role in transforming smart gadgets to smarter gadgets by making them compact, affordable, and less complex.

Bluetooth Low Energy, also marketed as Bluetooth Smart, started as part of the Bluetooth 4.0 Core Specification. These design goals are evident through the core specification, which attempts to make BLE a genuine low-power standard, designed to actually be implemented by semiconductor manufacturers and used in real-world applications tight on energy with minimal budget. It is already a widely adopted technology that can realistically stake claim to run for an extended period of time off a single coin cell. The challenges classic Bluetooth faced were fast battery draining and frequent loss of connection, requiring frequent pairing and re-pairing. Being able to successfully address these is one of the reasons for BLE’s rapid growth.

**Physical Layer**

Bluetooth Low Energy uses 80 MHz of spectrum in the 2.4 GHz ISM band to provide low-power, low-rate wireless communications over a range of 10 to 1,000 metres (depending on environment/configuration). The Bluetooth Low Energy physical layer provides four transmission modes with bit rates ranging between 125 Kbps and 2 Mbps, although packet length limitations and protocol overheads mean that maximum application throughput will be lower than PHY bit rate by 20-60% depending on protocol features available/enabled.

**Communication Topologies**

Bluetooth Low Energy has network topologies that are geared toward ad hoc communication scenarios involving relatively-infrequent transfer of small quantities of data. The technology can also support connection-oriented use cases such as those involving virtual serial ports or human interface devices.

The Bluetooth specification defines four roles that a Bluetooth Low Energy device may take: *broadcaster*, *observer*, *peripheral* and *central*.

* Broadcaster devices are transmit-only, and periodically broadcast advertising packets that may be detected by devices acting in the observer role.
* Observer devices are the counterpart of broadcasters - they are receive-only and listen for advertisements from broadcaster devices.
* Peripheral devices initially act like broadcasters, but transmit *connectable* advertising packets and accept connections from central devices.
* Central devices initiate connections to peripherals by listening for connectable advertising packets and then exchanging packets with the peripheral device.

**Upper Layers**

Bluetooth Low Energy introduces a protocol and a profile that work hand-in-hand to provide the basis for all Bluetooth Low Energy applications. These are the *Attribute Protocol* and the *Generic Attribute Profile,* respectively, and together they allow applications to expose state variables to a peer, and query or manipulate state variables of a peer.

The Attribute Protocol (ATT) defines server and client roles, and the basic operations that can be performed on state in the attribute database of the server. For example:

* A client may discover the attributes available on the server.
* A client may read the value of an attribute from the server.
* A client may write a value to an attribute on the server (with variants requiring and not requiring ATT-layer acknowledgement from the server).
* A server may send the value of an attribute to the client (with variants requiring and not requiring ATT-layer acknowledgement from the client).

**Features**

1. The lowest power consumption
2. Cost efficient and compatible
3. Robustness, security, and reliability
4. Wireless co-existence
5. Connection range
6. Ease of use and integration

**3) What are charters tics of IOT? What are IOT Protocol explain in details.**

**Definition:** - The internet of things (IoT) is a computing concept that describes the idea of everyday physical objects being connected to the internet and being able to identify themselves to other devices. It has dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual “things” have identities, physical attributes and virtual network and use intelligent interfaces.

* **Dynamic and Self-Adapting:**- IoT devices and system may have the capability to change dynamically depending upon the system and operating conditions or sensed environment.

For example, the surveillance cameras can change their modes based on day or night.

* **Self-configuring:**- IoT devices have self-configuring capability which allows large number of devices to work together to work provide certain functionality they can change their networking and update the software automatically.
* **Interoperable Communication Protocol:** - IoT devices can communicate with number of interoperable (communicate with other devices without special effort) communication protocols.
* **Unique ID:** - IoT devices have a unique identity differentiated with unique IP address.**Integrated into**
* **Information Network:** - IoT devices are integrated into the information network that allows them to communicate and exchange data with other devices and system.

**Types of IoT Protocols**IoT protocols and standards can be broadly classified into two separate categories.

 

**1.   IoT Network Protocols**

IoT network protocols are used to connect devices over the network. These are the set of communication protocols typically used over the Internet. Using IoT network protocols, end-to-end data communication within the scope of the network is allowed. Following are the various IoT Network protocols:

* + **HTTP (HyperText Transfer Protocol)**HyperText Transfer Protocol is the best example of IoT network protocol. This protocol has formed the foundation of data communication over the web. It is the most common protocol that is used for IoT devices when there is a lot of data to be published. However, the HTTP protocol is not preferred because of its cost, battery-life, energy saving, and more constraints.

Additive manufacturing/3D printing is one of the use cases of the HTTP protocol. It enables computers to connect 3D printers in the network and print three-dimensional objects and pre-determined process prototypes.

* + **LoRaWan (Long Range Wide Area Network)**It is a long-range low power protocol that provides signal detection below the noise level. LoRaWan connects battery operated things wirelessly to the Internet in either private or global networks. This communication protocol is mainly used by smart cities, where there are millions of devices that function with less power and memory.

Smart street lighting is the practical use case of LoRaWan IoT protocol. The street lights can be connected to a LoRa gateway using this protocol. The gateway, in turn, connects to the cloud application that controls the intensity of light bulbs automatically based on the ambient lighting, which helps in reducing the power consumption during day-times.

* + **Bluetooth**Bluetooth is one of the most widely used protocols for short-range communication. It is a standard IoT protocol for wireless data transmission. This communication protocol is secure and perfect for short-range, low-power, low-cost, and wireless transmission between electronic devices. BLE (Bluetooth Low Energy) is a low-energy version of Bluetooth protocol that reduces the power consumption and plays an important role in connecting IoT devices.

Bluetooth protocol is mostly used in smart wearables, smartphones, and other mobile devices, where small fragments of data can be exchanged without high power and memory. Offering ease of usage, Bluetooth tops the list of [IoT device](https://www.kelltontech.com/kellton-tech-blog/iot-device-security-7-ways-secure-your-iot-devices) connectivity protocols.

* + **ZigBee**
	ZigBee is an IoT protocol that allows smart objects to work together. It is commonly used in home automation. More famous for industrial settings, ZigBee is used with apps that support low-rate data transfer between short distances. Street lighting and electric meters in urban areas, which provides low power consumption, use the ZigBee communication protocol.  It is also used with security systems and in smart homes.

**2.   IoT Data Protocols**

IoT data protocols are used to connect low power IoT devices. These protocols provide point-to-point communication with the hardware at the user side without any Internet connection. Connectivity in IoT data protocols is through a wired or a cellular network. Some of the IoT data protocols are:

* + **Message Queue Telemetry Transport (MQTT)**One of the most preferred protocols for IoT devices, MQTT collects data from various electronic devices and supports remote device monitoring. It is a subscribe/publish protocol that runs over Transmission Control Protocol (TCP), which means it supports event-driven message exchange through wireless networks.

MQTT is mainly used in devices which are economical and requires less power and memory. For instance, fire detectors, car sensors, smart watches, and apps for text-based messaging.

* + **Constrained Application Protocol (CoAP)**CoAP is an internet-utility protocol for restricted gadgets. Using this protocol, the client can send a request to the server and the server can send back the response to the client in HTTP. For light-weight implementation, it makes use of UDP (User Datagram Protocol) and reduces space usage. The protocol uses binary data format EXL (Efficient XML Interchanges).

CoAP protocol is used mainly in automation, mobiles, and microcontrollers. The protocol sends a request to the application endpoints such as appliances at homes and sends back the response of services and resources in the application.

* + **Advanced Message Queuing Protocol (AMQP)**AMQP is a software layer protocol for message-oriented middleware environment that provides routing and queuing. It is used for reliable point-to-point connection and supports the seamless and secure exchange of data between the connected devices and the cloud. AMQP consists of three separate components namely Exchange, Message Queue, and Binding. All these three components ensure a secure and successful exchange and storage of messages. It also helps in establishing the relationship of one message with the other.

AMQP protocol is mainly used in the banking industry. Whenever a message is sent by a server, the protocol tracks the message until each message is delivered to the intended users/destinations without failure.

* + **Machine-to-Machine (M2M) Communication Protocol**It is an open industry protocol built to provide remote application management of IoT devices. M2M communication protocols are cost-effective and use public networks. It creates an environment where two machines communicate and exchange data. This protocol supports the self-monitoring of machines and allows the systems to adapt according to the changing environment.

M2M communication protocols are used for smart homes, automated vehicle authentication, vending machines, and ATM machines.

* + **Extensible Messaging and Presence Protocol (XMPP)**
	The XMPP is uniquely designed. It uses a push mechanism to exchange messages in real-time. XMPP is flexible and can integrate with the changes seamlessly. Developed using open XML (Extensible Markup Language), XMPP works as a presence indicator showing the availability status of the servers or devices transmitting or receiving messages.

Other than the instant messaging apps such as Google Talk and WhatsApp, XMPP is also used in online gaming, news websites, and Voice over Internet Protocol (VoIP).

**4) explain detail about network function virtualization.**

**Network Function Virtualization (NFV)**

* Network functions virtualization (NFV) is the concept of replacing dedicated network appliances such as routers and firewalls with software running on general-purpose CPUs or virtual machines, operating on standard servers.
* NFV provides the infrastructure on which SDN can run. NFV and SDN are mutually beneficial to each other but not dependent.
* NFV replaces proprietary hardware network elements (NEs) with software running on standard servers, SDN deals with the replacement of standardized networking protocols with centralized control
* The key elements of NFV architecture are:
1. **NFV infrastructure (NFVI):**

NFV infrastructure layer consists of different layers such as hardware resources like computing resources (RAM, servers), storage resources (hard-disc), and network resources (routers, switch, and firewalls). Second layer is Virtualization layer which separates hardware and replaces it with software and third layer is virtualized resources such as virtual compute, network and storage

1. **Virtualized network function (VNF):**

VNF is a software implementation is a network function which is capable of running over the NFV infrastructure (NFVI). Example: - vFirewall, vRouters

1. **NFV management and orchestration:**

It has three parts

* 1. Virtualized infrastructure manager: - it controls and manages network functions with NFVI resources and monitors virtualization layer.
	2. VNF manager: - it manages the life cycle of VNF such as initialize, update, query, scale, terminate etc.
	3. Orchestrator: - it manages the life cycle of network services which includes policy management, performance measurement and monitoring.



**Need For IoT Systems Management**

* Automating Configuration
* Monitoring Operational & Statistical Data
* Improved Reliability
* System Wide Configurations
* Multiple System Configurations
* Retrieving & Reusing

5) Discuss about motivation for using python in details.

**What is Python?**

Python language is similar to Perl object-oriented programming language. With the precise syntax and readability, python programming language became more popular. Python is easy to learn and portable, it is an open source language and supports multiple operating systems including Unix, Mac, Das, and various versions of Microsoft Windows.

Python is efficient, fast and supports programming paradigms that include object-oriented, impressively functional, procedural, etc. With the scripting language, you can develop desktop applications and web-based applications, and it also translated into binary language like Java.

Python plays a significant role in developing internet of things. Now python is coming into the field, with the following features of python most of developers prefer python programming language.

Features:

* **Easy to learn**: Learning and implementation of python is relatively simple and easy when compared to other native languages like C++ and java.
* **Easy to debug**: Python scripting language is one of the better language to debug than C++ and C. In this source code is executed line by line.
* **Speed:** Python code is relatively speed because of it contains C-language features in it.
* **Library support**: Python supports large standard libraries. Installation of the libraries are easy, and it save the time.
* **Easy to code**: With the clear syntax developers get a idea on code identification instead of {};
* **Embeddable**: Python allows integration with other languages i.e. It is possible to put our code in other programming language like C++ etc.
* **Extensible**: Python is extensible language. One can replace few lines of C++ code with Python to reduce the amount of execution time.
* **Interpreted**: In python no need to compile. The source code is converted into intermediate form of bytecode. In C++/Java you should compile first than run the code.
* **Portable**: Python code is portable there is no need to change the code for different machines. You can run one code in many machines
* **Free and open source**: Python is a Free/Libre open source software(FLOSS). It source code is freely available to public you can download it, change it and distribute it.
* **Community supports:**Python has already got its huge response in the market with the above mentioned features thus provides many users grouped into community to support the advancements further.

Some years ago python was only used for web applications; no one thought it would apply in IoT development. But now developer’s uses python programming language for developing the IOT devices. With it’s efficiently programming and easily syntax’s most of looking towards python.

Developers have to create IoT devices to make life easy. The small IoT devices have low computational power and memory, so developers choose python scripting language. Nowadays most popular microcontrollers also use python language like Micropython board and software package and other.

**6) Explain about data intensive IOT for continues and reorganization and application**

**Data-intensive** computing is a class of parallel computing applications which use a data parallel approach to process large volumes of data typically terabytes or petabytes in size and typically referred to as big data. Computing applications which devote most of their execution time to computational requirements are deemed compute-intensive, whereas computing applications which require large volumes of data and devote most of their processing time to I/O and manipulation of data are deemed data-intensive.

Data-intensive is used to describe applications that are I/O bound or with a need to process large volumes of data. Such applications devote most of their processing time to I/O and movement and manipulation of data. Parallel processing of data-intensive applications typically involves partitioning or subdividing the data into multiple segments which can be processed independently using the same executable application program in parallel on an appropriate computing platform, then reassembling the results to produce the completed output data. The greater the aggregate distribution of the data, the more benefit there is in parallel processing of the data. Data-intensive processing requirements normally scale linearly according to the size of the data and are very amenable to straightforward parallelization. The fundamental challenges for data-intensive computing are managing and processing exponentially growing data volumes, significantly reducing associated data analysis cycles to support practical, timely applications, and developing new algorithms which can scale to search and process massive amounts of data. Researchers coined the term BORPS for "billions of records per second" to measure record processing speed in a way analogous to how the term MIPS applies to describe computers' processing speed

**Data-parallelism**

Computer system architectures which can support data parallel applications were promoted in the early 2000s for large-scale data processing requirements of data-intensive computing.[[12]](https://en.wikipedia.org/wiki/Data-intensive_computing#cite_note-12) Data-parallelism applied computation independently to each data item of a set of data, which allows the degree of parallelism to be scaled with the volume of data. The most important reason for developing data-parallel applications is the potential for scalable performance, and may result in several orders of magnitude performance improvement. The key issues with developing applications using data-parallelism are the choice of the algorithm, the strategy for data decomposition, load balancing on processing nodes, message passing communications between nodes, and the overall accuracy of the results

**Characteristics**

Several common characteristics of data-intensive computing systems distinguish them from other forms of computing:

1. The principle of collection of the data and programs or algorithms is used to perform the computation. To achieve high performance in data-intensive computing, it is important to minimize the movement of data.[[19]](https://en.wikipedia.org/wiki/Data-intensive_computing#cite_note-19) This characteristic allows processing algorithms to execute on the nodes where the data resides reducing system overhead and increasing performance.[[20]](https://en.wikipedia.org/wiki/Data-intensive_computing#cite_note-20) Newer technologies such as [InfiniBand](https://en.wikipedia.org/wiki/InfiniBand%22%20%5Co%20%22InfiniBand) allow data to be stored in a separate repository and provide performance comparable to collocated data.
2. The programming model utilized. Data-intensive computing systems utilize a machine-independent approach in which applications are expressed in terms of high-level operations on data, and the runtime system transparently controls the scheduling, execution, load balancing, communications, and movement of programs and data across the distributed computing cluster.[[21]](https://en.wikipedia.org/wiki/Data-intensive_computing#cite_note-21) The programming abstraction and language tools allow the processing to be expressed in terms of data flows and transformations incorporating new dataflow [programming languages](https://en.wikipedia.org/wiki/Programming_languages) and shared libraries of common data manipulation algorithms such as sorting.
3. A focus on reliability and availability. Large-scale systems with hundreds or thousands of processing nodes are inherently more susceptible to hardware failures, communications errors, and software bugs. Data-intensive computing systems are designed to be fault resilient. This typically includes redundant copies of all data files on disk, storage of intermediate processing results on disk, automatic detection of node or processing failures, and selective re-computation of results.
4. The inherent scalability of the underlying hardware and [software architecture](https://en.wikipedia.org/wiki/Software_architecture). Data-intensive computing systems can typically be scaled in a linear fashion to accommodate virtually any amount of data, or to meet time-critical performance requirements by simply adding additional processing nodes. The number of nodes and processing tasks assigned for a specific application can be variable or fixed depending on the hardware, software, communications, and [distributed file system](https://en.wikipedia.org/wiki/Distributed_file_system) architecture.

**System architectures**

A variety of system architectures have been implemented for data-intensive computing and large-scale data analysis applications including parallel and distributed relational database management systems which have been available to run on shared nothing clusters of processing nodes for more than two decades.However most data growth is with data in unstructured form and new processing paradigms with more flexible data models were needed. Several solutions have emerged including the Map Reduce architecture pioneered by Google and now available in an open-source implementation called Hadoop used by Yahoo, Facebook, and others.

**MapReduce**

he MapReduce architecture allows programmers to use a functional programming style to create a map function that processes a key-value pair associated with the input data to generate a set of intermediate key-value pairs, and a reduce function that merges all intermediate values associated with the same intermediate key. Since the system automatically takes care of details like partitioning the input data, scheduling and executing tasks across a processing cluster, and managing the communications between nodes, programmers with no experience in parallel programming can easily use a large distributed processing environment.

The programming model for MapReduce architecture is a simple abstraction where the computation takes a set of input key-value pairs associated with the input data and produces a set of output key-value pairs. In the Map phase, the input data is partitioned into input splits and assigned to Map tasks associated with processing nodes in the cluster. The Map task typically executes on the same node containing its assigned partition of data in the cluster. These Map tasks perform user-specified computations on each input key-value pair from the partition of input data assigned to the task, and generates a set of intermediate results for each key. The shuffle and sort phase then takes the intermediate data generated by each Map task, sorts this data with intermediate data from other nodes, divides this data into regions to be processed by the reduce tasks, and distributes this data as needed to nodes where the Reduce tasks will execute. The Reduce tasks perform additional user-specified operations on the intermediate data possibly merging values associated with a key to a smaller set of values to produce the output data. For more complex data processing procedures, multiple MapReduce calls may be linked together in sequence

**Hadoop**

Apache Hadoop is an open source software project sponsored by The Apache Software Foundation which implements the MapReduce architecture. Hadoop now encompasses multiple subprojects in addition to the base core, MapReduce, and HDFS distributed filesystem. These additional subprojects provide enhanced application processing capabilities to the base Hadoop implementation and currently include Avro, Pig, HBase, ZooKeeper, Hive, and Chukwa. The Hadoop MapReduce architecture is functionally similar to the Google implementation except that the base programming language for Hadoop is Java instead of C++. The implementation is intended to execute on clusters of commodity processors.

Hadoop implements a distributed data processing scheduling and execution environment and framework for MapReduce jobs. Hadoop includes a distributed file system called HDFS which is analogous to GFS in the Google MapReduce implementation. The Hadoop execution environment supports additional distributed data processing capabilities which are designed to run using the Hadoop MapReduce architecture. These include HBase, a distributed column-oriented database which provides random access read/write capabilities; Hive which is a data warehouse system built on top of Hadoop that provides SQL-like query capabilities for data summarization, ad hoc queries, and analysis of large datasets; and Pig – a high-level data-flow programming language and execution framework for data-intensive computing.

**HPCC**

HPCC (High-Performance Computing Cluster) was developed and implemented by LexisNexis Risk Solutions. The development of this computing platform began in 1999 and applications were in production by late 2000. The HPCC approach also utilizes commodity clusters of hardware running the Linux operating system. Custom system software and middleware components were developed and layered on the base Linux operating system to provide the execution environment and distributed filesystem support required for data-intensive computing. LexisNexis also implemented a new high-level language for data-intensive computing.