N – 24891

Register No. | | | | | | | | |

BE Degree Examination April 2024

Sixth Semester

Computer Science and Engineering

20CST62 – INTERNET OF THINGS AND CLOUD

(Regulations 2020)

Time: Three hours | Maximum: 100 marks

Answer all Questions

Part – A (10 × 2 = 20 marks)

1. List any two characteristics of IoT. [CO1,K1]
2. Imagine a chat application where users can send messages to each other in real-time. Identify a suitable communications model for this application. [CO1,K3]
3. For a smart home environment, identify a suitable technology to exchange the data. Justify your answer. [CO2,K3]
4. What is use of service management layer in the layer architecture of IoT? [CO2,K2]
5. Write a Python program that toggles a LED on and off alternatively. [CO3,K3]
6. Name the interfaces supported by Raspberry Pi. Differentiate between them. [CO3,K2]
7. What are the cloud service models? Give examples. [CO4,K1]
8. Distinguish between hybrid cloud and federated clouds. [CO4,K2]
9. Name the tools needed to create web based application for device communication among IoT devices. [CO5,K1]
10. What is meant by device shadows? [CO5,K2]

Part – B (5 × 16 = 80 marks)

11. a. For a remote environmental monitoring system, deploy sensors in remote locations to monitor environmental parameters such as temperature, humidity, air quality and soil moisture. Identify a suitable IoT deployment level and draw the suitable template and explain the various components in it. Justify the chosen IoT level. (16) [CO1,K4]

(OR)

b. Write requirement specification, process specification, domain model specification and information model specification for smart health care monitoring system. (16) [CO1,K4]

12. a. Draw the protocol architecture of an IoT ecosystem. Explain the role of each protocol with appropriate examples. (16) [CO2,K2]

(OR)

b. Identify the protocols available for IoT service discovery. With suitable sketches, illustrate the working of these protocols. (16) [CO2,K2]

13. a. Write a Python program that simulates traffic light controller. (16) [CO3,K3]

(OR)

b. Write a Python program that reads the current state of LED (on/off) and upload 0 of LED is off and 1 if LED is on to a cloud of your choice. (16) [CO3,K3]

14. a. i) Design a sensor embedded smart environment system. Explain how the system helps for real time data capture. (7) [CO4,K3]

ii) Classify the cloud federation approaches. List their advantages and disadvantages. (9) [CO4,K3]

(OR)

b. i) List the key architectural elements and capabilities of software defined storage. (9) [CO4,K1]

ii) Illustrate the role of edge computing in smarter traffic system. (7) [CO4,K3]

15. a. i) List IoT core services ad give the significance of each service. (6) [CO5,K1]

ii) How do we build cognitive IoT applications using rules engine? Explain with examples. (10) [CO5,K2]

(OR)

b. i) List the benefits of device shadows. (6) [CO5,K1]

ii) What are the protocols for communication with and between devices? Provide a brief note of them. (10) [CO5,K2]

Bloom's Taxonomy Level	Remembering (K1)	Understanding (K2)	Applying (K3)	Analysing (K4)	Evaluating (K5)	Creating (K6)
Percentage	20	33	29	18	–	–

Kongu Engineering College, Perundurai - 638 060

END SEMESTER

ANSWER KEY

(Regulation: 2020)

Course Code: 20CSTG2

Course Name: Internet of Things and cloud

Programme: BE Branch: CSE Semester: VI

PART-A (10 * 2 = 20 marks)

1. List any two characteristics of IoT (Any 2) (2*1=2)

   * Dynamic Self Adapting, self configuring
   * Interoperable Communication Protocols
   * Integrated into the information network

2. Imagine a chat application where users can send messages to each other in real time Identify a suitable communication model for this Application (Any 1) (1*2=2)

   * Request - Response
   * Publish - Subscribe
   * Exclusive Pair

3. For a smart home environment identify a suitable technology to exchange the data justify your answer

   * M2M [Mobile to Mobile] / D2D [Device to Device] [1]
   * Zwave
   * Home Security are merging with energy [1] management to provide remote alarm controls as well as remote heating, Ventilation and air conditioning controls for homes and businesses through mobile phones.

4. What is the use of service management layer in the layer architecture of IoT [2] middleware layer

It Provides flexibility to the IoT programmers to work on different types of heterogeneous Objects irrespective of their platforms

---

5. Write a python Program that toggles a led on and off alternatively [2]

```
import RPI.GPIO as GPIO
import time
LED-PIN =17
GPIO.setmode (GPIO.BCM)  (Board)
GPIO.setup (LED-PIN , GPIO.OUT)
try:
    while True:
        GPIO.output (LED-PIN , GPIO.HIGH)
        time.sleep (1000)
        GPIO.output (LED-PIN, GPIO.LOW)
        time.sleep (1000)
except KeyboardInterrupt:
    GPIO.cleanup()
```

6. Name the interfaces supported by raspberry Pi. Differentiate between them. [1]

Serial, GPIO, I2C, SPI, UART, USB, HDMI and Ethernet [1]

GPIO is versatile for general digital I/O tasks, while I2C and SPI are for communication with multiple low speed (I2C) or high speed (SPI) peripherals.

UART - Simple serial communication

USB - Connects peripherals like keyboards and storage

HDMI - transmits high quality audio and video

Ethernet - Provides wired network connectivity

---

7. What are the cloud service models give example. [2]

Iaas (Infrastructure as a service) EC2 | S3 | Lambda

Paas (Platform as a service], Google App Engine

Saas (software as a service), Microsoft Office

---

8 Distinguish between Hybrid and Federated cloud. [2*1=2]

Hybrid cloud	Federated Cloud
* Computing environment that combines on-premises infrastructure (private cloud) with public cloud services.	* Collection of interconnected cloud environment.

* Allowing data and applications to be shared between them	* Operated independently by different organizations or providers but linked together to enable resources Sharing and collaboration

9. Name the tools needed to create web based applications for device communication among IoT devices [2*1=2]

* IoT Protocols and Libraries: MQTT, websocket, etc.

* Cloud platforms and services: Aws IoT core, Google cloud IoT core, Microsoft Azure IoT hub. S2, EC2, Lambda

10. What is meant by device shadows. [2]

Virtual representations of Physical IoT devices in the cloud, enabling Synchronization of device state and Communication with cloud applications.

Part-B (5x16=80 Marks)

11 a. For a remote environmental monitoring system, deploy sensors in remote locations to monitor environmental parameters such as temperature, humidity, air quality and soil moisture. Identify a suitable IoT deployment level and draw the suitable template and explain the various components in it. Justify the chosen IoT Level. (16)

[CO1, K4]

IoT Level 4 Edge Analytics and Intelligence [4 marks]

Justification (For Level 5 or Level 6) (2 mark)

* A Level 4 IoT system has multiple nodes that perform local analysis
* Data is stored in the cloud and application is cloud based
* Level-4 contains local and cloud based observer nodes which can subscribe to and receive information collected in the cloud from IoT devices
* Level-4 IoT systems are suitable for solutions where multiple nodes are required
* Data involved is big and Analysis requirements are computationally intensive

Components [6 marks]

Device - Identification, Remote sensing, Actuating, Remote monitoring capabilities

Resource - Software components on IoT devices for accessing, Processing and storing sensor information or controlling actuators connected to the devices. Also include software component that enables network access for the devices.

Level Identification - 2
Diagram - 6
Discription - 4
Justification - 4

Controller Service - Sends data from the devices to the web services and receive commands from the application for controlling the devices

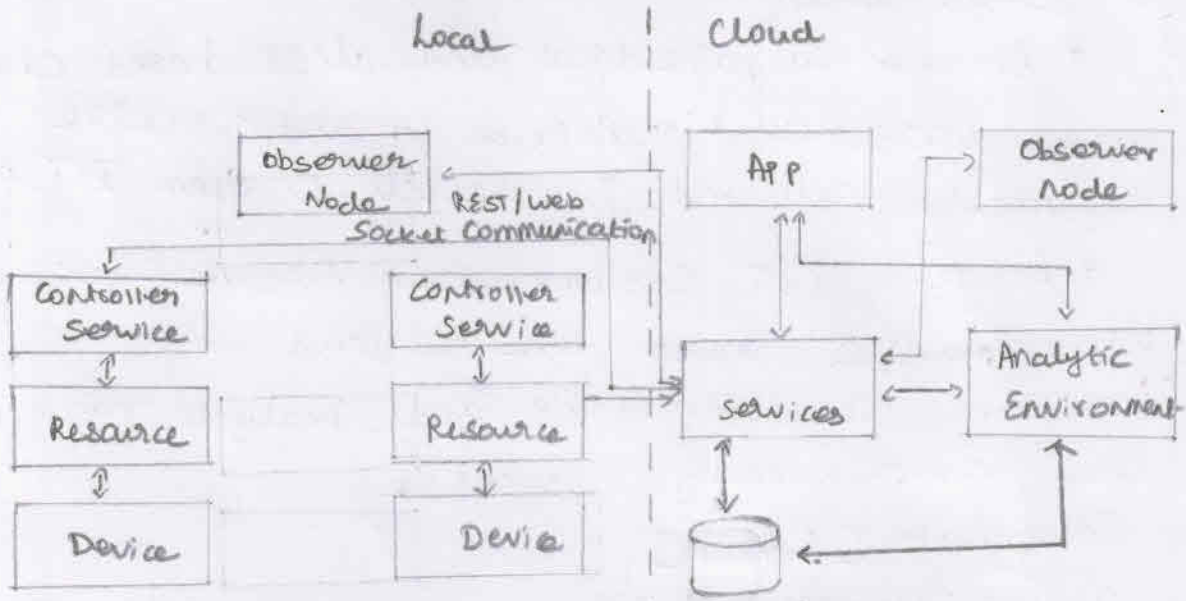
Database - Stores data generated by the devices

Web Service - Serves as a link between the IoT device, application, database and analysis component.

Analysis Component - Analyse the data and generate the result.

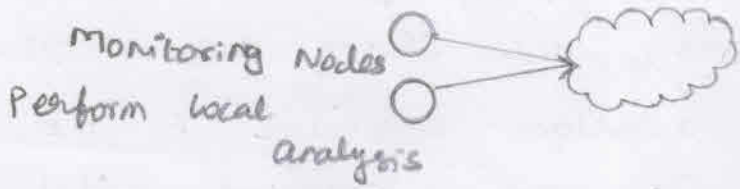
Application - Provides the interface that the user can use to control and monitor various aspects of IoT systems

IoT Level - 4/5/6 [6 marks]



Level 5 - Coordinator

Level 6 - Centralized coordinator



11 b. Write requirement Specification, Process specification domain model specification and Information model Specification for smart health care monitoring System. (16) [CO1, K4].

Purpose - 2 mark
steps/flow Diagram - 2 mark
Discription - 2 mark

Requirement Specification [4 Marks]

- Functional Requirements: (1m)
  - → Monitor Patient vital Signs continuously.
  - → Alert healthcare Providers in real time for emergencies.
  - → Log Patient data securely for further reference.
  - → Allow remote access to Patient data for healthcare Providers.
  - → Integrate with existing Electronic Health Record (EHR) Systems.
- Non-Functional Requirements: (1 m)
  - → Ensure Patient data Security.
  - → Scale System to handle more Patients and devices.
  - → Maintain System reliability for continuous monitoring
  - → Provide User-Friendly interface for health care Providers
  - → Support Compatibility with various medical devices.

Process Specification: [4 Marks]

steps/Flowchart diagram - 2 mark.
Discription - 2 mark

- Data Collection:
  - ↳ Sensors collect Vital Signs data Securely.
  - ↳ Data transmitted to central System.

- Data Processing
  - ↳ Central System analyzes data for abnormalities
  - ↳ Alerts generated for healthcare Providers.
- Alerting and Notification:
  - ↳ Healthcare Providers receive alert through sms, email or app
  - ↳ Access patient data and take action.
- Data Storage and Access:
  - ↳ Secure storage of patient data.
  - ↳ Authorized access for healthcare Providers.
- Integration with EHR:
  - ↳ Seamless integration with existing EHR systems.

Domain Model Specification (4 marks)

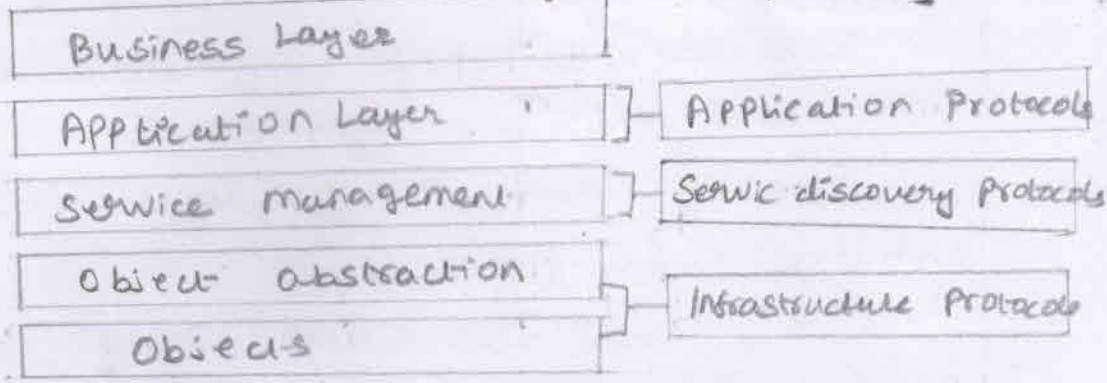
- Entities – Physical Entity, Virtual Entities
  - ↳ Patient, Healthcare Provider, Sensor, Alert Vital Signs & Electronic Health Record (EHR)
- Relationships
  - ↳ Patients have vital sign readings.
  - ↳ Healthcare Providers receive alerts for abnormal vital signs
  - ↳ Sensors collect vital signs data
  - ↳ Patient data stored in Electronic Health Record.

Information Model Specification (4 marks) – class diagram (2m)

- ↳ Patient Data: Name, Age, Gender, Medical History
- ↳ Vital Signs Data: Timestamp, Heart Rate, Blood Pressure, Temperature, Oxygen Saturation
- ↳ Alert Data: Severity, Description, Time Stamp
- ↳ Electronic Health Record (EHR): Patient ID Medical Records, Allergies, Medications

12 a. Draw the Protocol Architecture of an IoT ecosystem. Explain the role of each Protocol with appropriate examples. (16) [CO2, K2]

Diagram [4 marks]



Protocol Architecture of an IoT ecosystem

Role of Each Protocols

Routing Protocols (3 Marks)

Purpose: Determines the path data packets take from the source to the destination within the Network.

Protocols:

- RPL (Routing Protocol for Low-Power and Lossy Networks): IPV6 routing Protocol designed specifically for low power and lossy networks

Example: Wireless sensor networks in a smart grid.

- AODV (Ad hoc On-Demand Distance Vector): Routing Protocol for mobile Ad hoc Networks (MANETs)

Example: Vehicle to Vehicle Communication Systems

Link Layer: (3 marks)

Purpose:

Responsible for node-to-node data transfer and error detection/over a physical link.

Protocols:

IEEE 802.15.4: Defines the media access control (MAC) and physical layer for LR-WPANs.

Example: Smart meters in utility networks.

Network Layer (3 marks)

Purpose: Facilitates data packet transfer across multiple nodes and networks, ensuring the data reaches its intended destination

Protocols:

↳ IPV4/IPV6: Internet Protocol versions used for addressing and routing packets across networks.

Example: Assigning unique IP addresses to IoT devices for Internet connectivity.

↳ 6LowPAN (IPV6 over Low Power Wireless Personal Area Networks): Enables IPV6 communication over IEEE 802.15.4 networks.

Example: Environmental sensors in a smart-agriculture setup.

Physical Layer (3 marks)

Purpose: The physical layer deals with the transmission and reception of raw data over a physical medium.

Protocols:

↳LTE-A (Long Term Evolution Advanced): Provides high-speed wireless communication for mobile devices and data terminals.

Example: Cellular IoT devices such as connected cars or smart city infrastructure.

EPC global: Standards for RFID (Radio Frequency Identification) and the Internet of Things.

Example: Inventory Management Systems using RFID tags.

IEEE 802.15.4: Standard defining the physical layer for low rate wireless personal area networks (LR-WPANs)

Example: Basis for zigbee and 6LowPAN communication.

Z-wave: Wireless Communication Protocol Primarily used for home automation.

Example: Smart home devices like light switches and door locks.

---

12. b Identify the Protocols available for IoT service discovery. With suitable Sketches Illustrate the working of these Protocols (16) [CO2 K2]

IoT Service Discovery Protocol (1 mark)

* DNS Service Discovery (DNS-SD)
* Multicast Domain Name System (mDNS)
* Simple Service Discovery Protocol (Part of UPnP)

DNS Service Discovery (DNS-SD): (5 marks)

DNS Service Discovery (DNS-SD) allows clients to discover services on a network using standard DNS message, typically leveraging mDNS for multicast DNS queries.

Key Points:

Function: Helps clients discover desired services in a network

Process: Two-step: Finding host names and pairing IP addresses with host names using mDNS

Advantages: No external administration or configuration needed. Keeps hostnames constant even if IP addresses change.

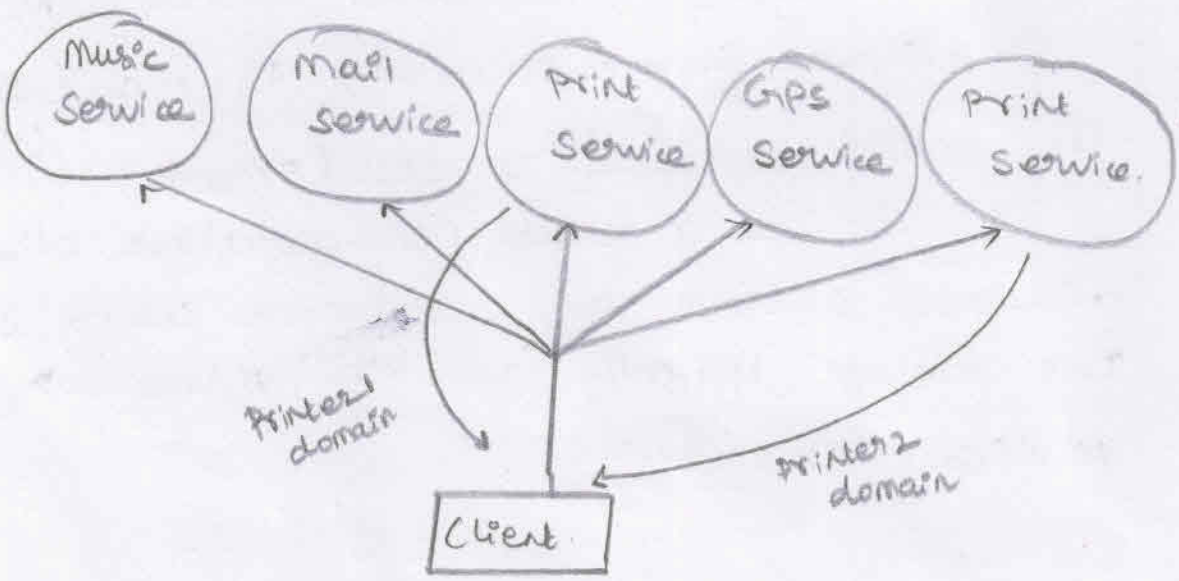
Working of DNS-SD:

1. Service Registration
   ↳ Devices register their services with a DNS server.

2. Service Discovery:
   ↳ Clients send DNS queries to discover services.
   ↳ DNS SD uses mDNS to send these queries to a multicast address.

3. Service Resolution:
   ↳ The DNS server or mDNS responds with the service's IP address and Port number.

Diagram



2. Multicast DNS (mDNS): (5 Marks)

mDNS allows devices to perform DNS-like operations on a local link without requiring a DNS Server. It uses IP multicast to send DNS Packets to all devices on the local network

key Points:

Function: Works like a unicast DNS server, but on a local network without additional configuration.

Advantages: No manual configuration needed, high fault tolerance, no additional infrastructure needed.

Working:

1) Service Query:

A device sends a multicast query to discover services.

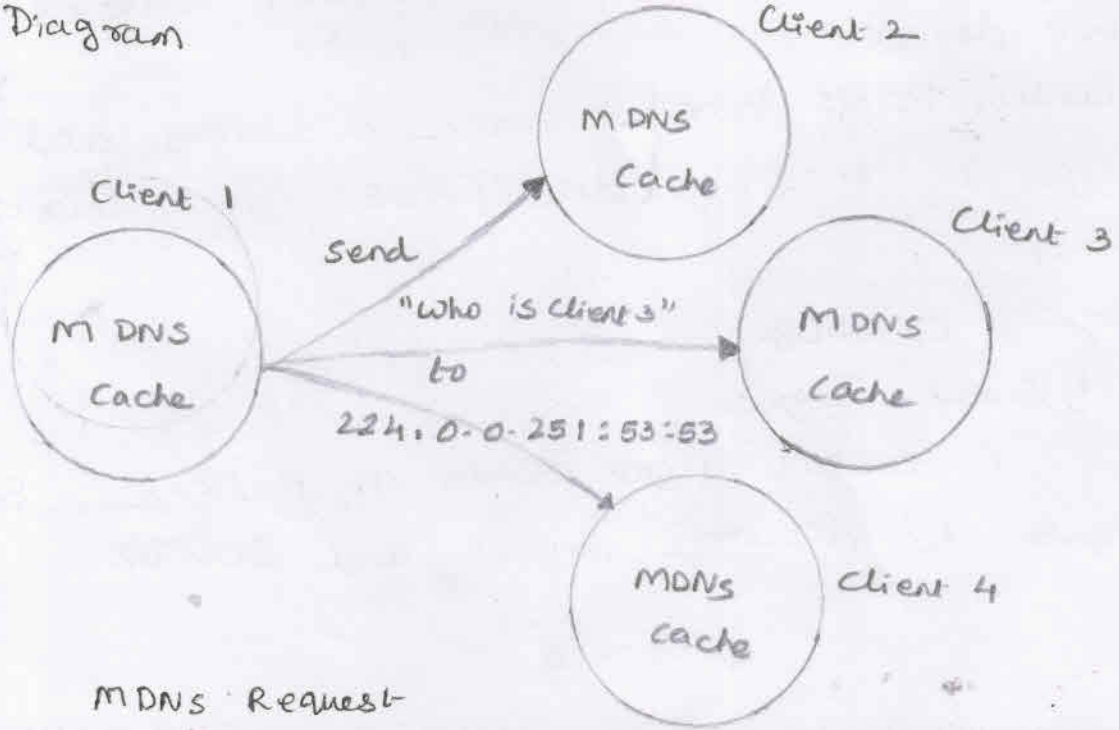
2) Service Response:

Target device responds with its name and IP address

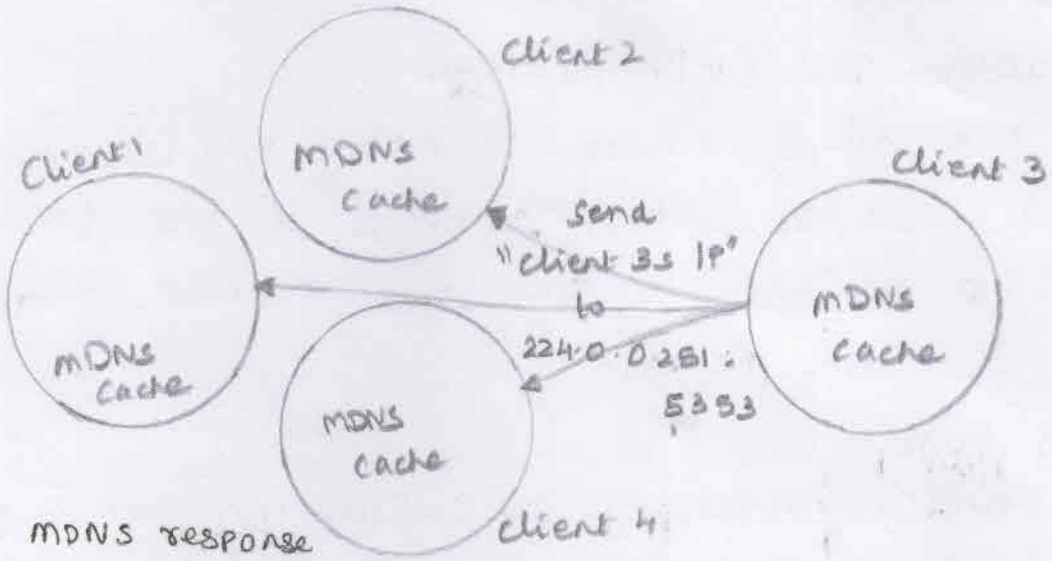
3) Cache Update:

All device respond like update their local cables with the new information.

Diagram



MDNS Request-



3. Universal Plug and Play (UPnP) with SSDP (5 Marks)

UPnP enables devices to join a network discover each other and share do services with Zero Configuration. SSDP is the protocol that handles Service discovery in UPnP networks.

Key Points:

Function: Allows dynamic discovery and interaction of devices and Services.

Advantages: Zero configuration, Supports remote access, flexible Connectivity.

Components:

Devices: Containers for Services and other devices

Services: Granular units of control offered by UPnP devices.

Control Points: Provide device discovery and control by receiving descriptions and invoking actions

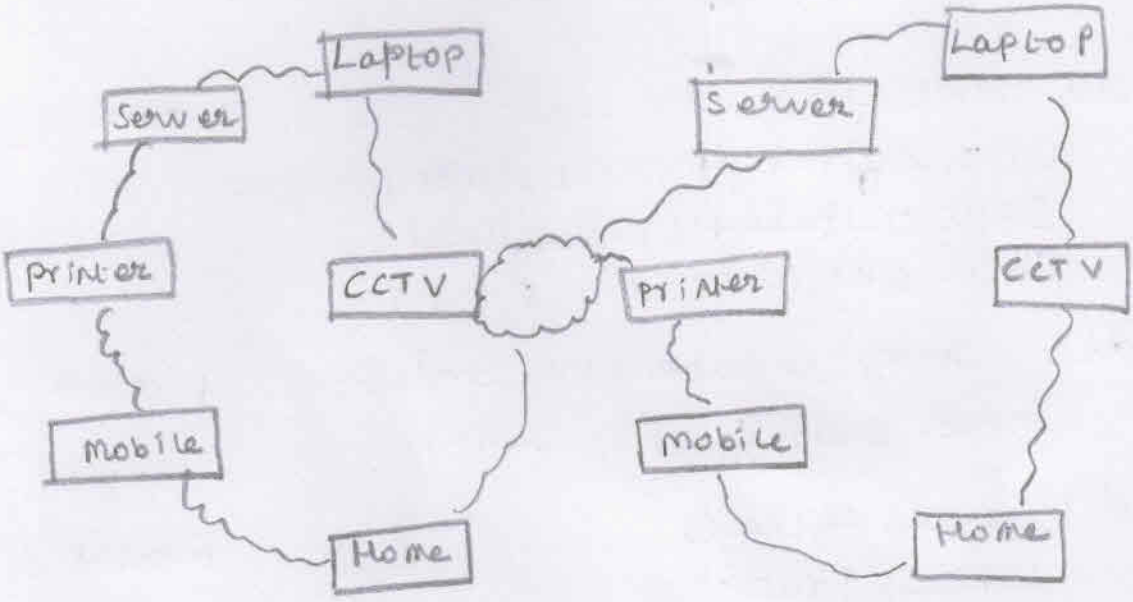
Working of SSDP.

1) Search Request:

A control Point Sends a multicast Search request to discover devices and Services.

2. Search Response:

Devices respond to the search request with their capabilities and services.



13

a. Write a python Program that simulates traffic light Controller (16)

```
# Initialize ..... 6 marks
import time
import RPi.GPIO as GPIO

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)          } 2 marks

GREEN = 8
YELLOW = 10                      } 2 marks
RED = 12

GPIO.setup(GREEN, GPIO.OUT)
GPIO.setup(YELLOW, GPIO.OUT)     | 2 marks
GPIO.setup(RED, GPIO.OUT)
```

```
cycle_time = {
    "GREEN": 1000, #seconds
    "YELLOW": 2000,
    "RED": 2000
    }
def initialize_lights():  ..... 1 mark
    GPIO.output(GREEN, False)
    GPIO.output(YELLOW, False)
    GPIO.output(RED, True)
def change_state(Pin, State): ... 1 mark
    GPIO.output(Pin, State)

def run_cycle(): ..... 6 marks
    While True:
        Change_state(GREEN, True)
        Change_state(YELLOW, False)
        Change_state(RED, False)
        time.sleep(cycle_time["GREEN"])
        Change_state(GREEN, False)
        Change_state(YELLOW, True)
        time.sleep(cycle_time["YELLOW"])
        Change_state(YELLOW, False)
        Change_state(RED, True)
        time_sleep(cycle_time["RED"])

if __name__ == "__main__": .... 2 marks
    try:
        initialize_lights()
        run_cycle()
    except keyboard Interrupt:
        Print("Simulation Stopped")      (or)
    finally:
        GPIO.cleanup()
```

(7 mark)

```
def run_cycle()
    while True:
        Change_State (GREEN, True)
        Change_State (YELLOW, False)
        Change_State (RED, False)
        time.sleep (1000)
        Change_State (GREEN, False)
        Change_State (YELLOW, True)
        time_sleep (2000)
        Change_State (YELLOW, False)
        Change_State (RED, True)
        time.sleep (3000)
```

13
b. Write a python program that reads the current state of led (on/off) and upload 0 of led is off and 1 if led is on to a cloud of your choice (16 marks)

```
# Initialize ..... [3 marks]
import RPI.GPIO as GPIO
import requests
import time

GPIO.setmode (GPIO.BOARD)
LED-PIN=8
GPIO.setup ( LED_PIN, GPIO.IN )

THINGSPEAK_API-KEY = "Your_Thingspeak_API_Key"  ... [4 marks]
THINGSPEAK_URL = f"https://api.thingspeak.com/
        update?api_key={THINGSPEAK_API_KEY}"
def read_led_state():
    return GPIO.input(LED-PIN)
def upload_to_thingspeak (state): ... [6 marks]
    response = requests.get(f"{THINGSPEAK_URL}
                & field1={state}")
    if response.status_code = 200:
        Print ("Dat uploaded to Thing speak
                successfully")
    else.
        Print ("failed to upload data to
                Thingspeak successfully")
```

```
def main():                                    [3 marks]
    try:
        while True:
            led_state = read_led_state()
            upload_to_thingspeak(led_state)
            time.sleep(10)
    except keyboard Interrupt:
        Print("program terminated by user")
    finally:
        GPIO.cleanup()
```

14

a.i) Design a Sensor-embedded Smart environment [7]

System explanation → [3 marks]

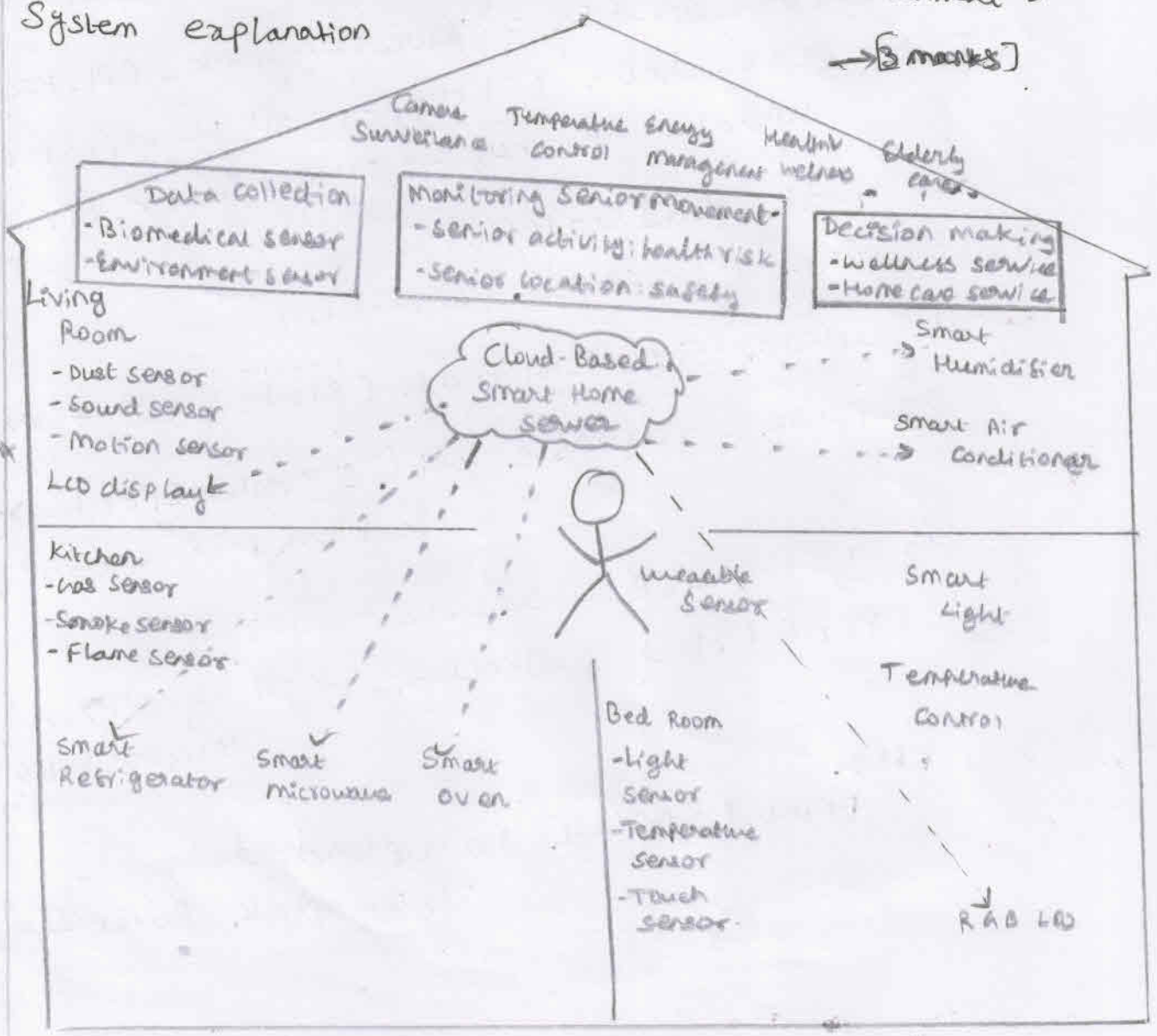


diagram 3 explanation

Explanation [4 Marks]

* Effective approach is to connect the devices in the cloud.

* Reduces the complexity of managing software in the home bound devices and simplifies the interoperability of devices

↳ Bridging the services throughout the network through services interfaces

↳ Translating the different device protocols to a common platform

↳ Connecting the devices through the network cloud

* Consumers consume the services through their connected devices

* Home automation elements, industry-strength and open standards trendy electronics, application development platforms, dynamic, virtualized and converged infrastructures and proven processes

* Futeristic home environments constitutes the following:

↳ Wireless broadband communication, ambient, agile and adaptive sensors and actuators, smart heating, lighting, ventilation and air control systems sophisticated, energy-efficient and connectable edutainment and infotainment electronics, home security

14
a.ii. Classify the cloud Federation approaches. List their advantages and disadvantages (9 marks)

Approaches: 3 marks

diagram - 2
explani - 1

* Centralized Approach
* Decentralized Approach
* Hybrid Approach

3 + 1 = 3 marks 3 * 1 = 3 marks

Approach	Advantages	Dis Advantages
Centralized	• Unified Control for easier management • Simplified administration and governance • Potential for optimized resource allocation and cost savings	• Single Point of failure introduces vulnerabilities • Scalability may become challenging
Decentralized	• Increased resilience due to no single point of failure • Flexibility for individual nodes to adapt independently • Easier horizontal scaling by adding more nodes	• Complexity in managing a distributed system • coordination challenges among decentralized components • Potential for inconsistencies without centralized oversight

Hybrid	• Balanced Control, Combining benefits of Centralized and decentralized approaches • Flexibility to tailor the model to specific requirements. • Balance between Scalability and resilence.	• Increased Complexity in design and management • Integration challenges between Centralized and decentralized Components. • Additional overhead for coordination and Communication.

14 b. i) List the key architectural elements and Capabilities of Software defined Storage [9 marks]

Elements: (3 marks)

1. Commodity Hardware:
   - Storage intelligence centralized in software layer
   - Storage solutions become cheap, off-the-shelf and Commoditized.
   - Interconnecting and intermediate fabric also commoditized

2. Scale-out Architecture:
   - SDS enables fluid, flexible and elastic Configuration of storage resources through Software.
   - Facilities dynamic pooling of heterogeneous resources
   - Traditional architecture hinders dynamic addition and release of storage resources.

Capabilities: (6 marks)

1. Resource Pooling:
   * Storage resources pooled into a unified logical entity.
   * Central management and control plane Provide visibility and control over resources.

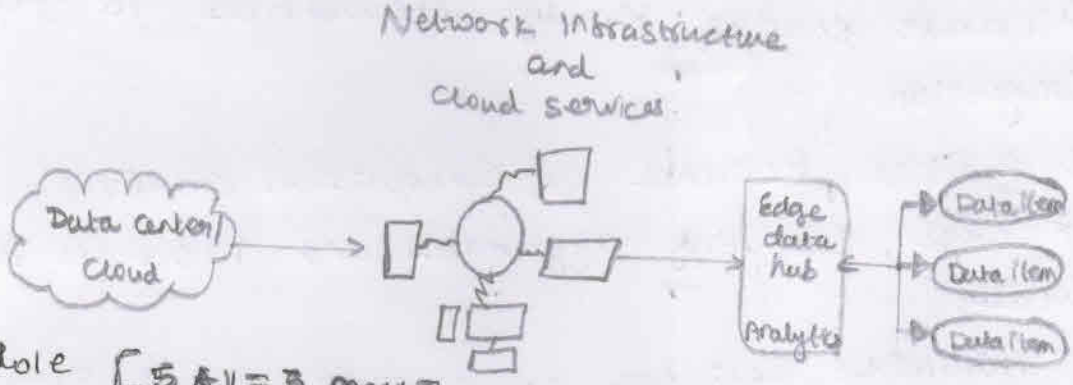
2. Abstraction:
   * Physical storage resources virtualized and presented to control plane.
   * Control plane configures and delivers tiered storage services.

3. Automation:
   * Extensive automation enables one-click and policy-based provisioning of storage resources.
   * Storage delivered based on application needs rather than specific configurations
   * Continuous monitoring and reconfiguration to meet SLAs.

4. Programmability:
   * Rich APIs provide fine-grained visibility and control of underlying resources.
   * Integration across storage, network and compute layers for workflow automation.
   * SDS integration with other infrastructure layers for end-end application-focused automation.

14. b. ii) Illustrate the role of edge computing in Smarter traffic system. [7 marks]

Diagram [2 marks]



Role [5 * 1 = 5 marks]

1. Real time Decision making:

* Immediate processing of data enables quick decision-making.

* Analysis of traffic patterns, vehicle speeds, and road conditions allows for rapid adjustments to prevent accidents

2. Low-Latency Responses:

* Processing data at the edge minimizes latency.

* Immediate feedback to smart traffic lights and connected vehicles ensures swift adjustments to traffic signals and routes.

3. Edge Device Collaboration:

* Smart traffic lights and roadside unit collaborate to collect and process data locally.

* Distributed decision-making reduces reliance on centralized systems and improves responsiveness.

4. Stream Data Processing:

* Edge computing processes data streams in real-time.

* Accurate detection of complex events, such as accidents, ensures timely interventions to prevent congestion

5. Reduced Reliance on Centralized cloud:

* Edge computing aggregates and processes data locally.

* Minimizes reliance on remote cloud servers, ensuring critical decisions can be made quickly, even with limited network connectivity.

15

a i) List iot core services and give the significance of each service (6)

[6*1 = 6 marks]

1) Device management:

* Registers, Monitors and Controls IoT devices at scale.

* Ensures reliability and security of deployments

2) Data Ingestion and Storage

* Collects and stores vast amount of IoT data

* Facilitates real-time analytics and decision-making

3) Data Processing and Analytics.

* Derives actionable insights from IoT data

* Optimizes operations and identifies trends.

4 marks
EC2
Lambda

4) Security and Identity Management:

* Protects devices, data and communications from threats.
* Maintains compliance and safeguards sensitive information.

5) Connectivity Management:

* Manages diverse IoT device connections.
* Optimizes network performance and reliability

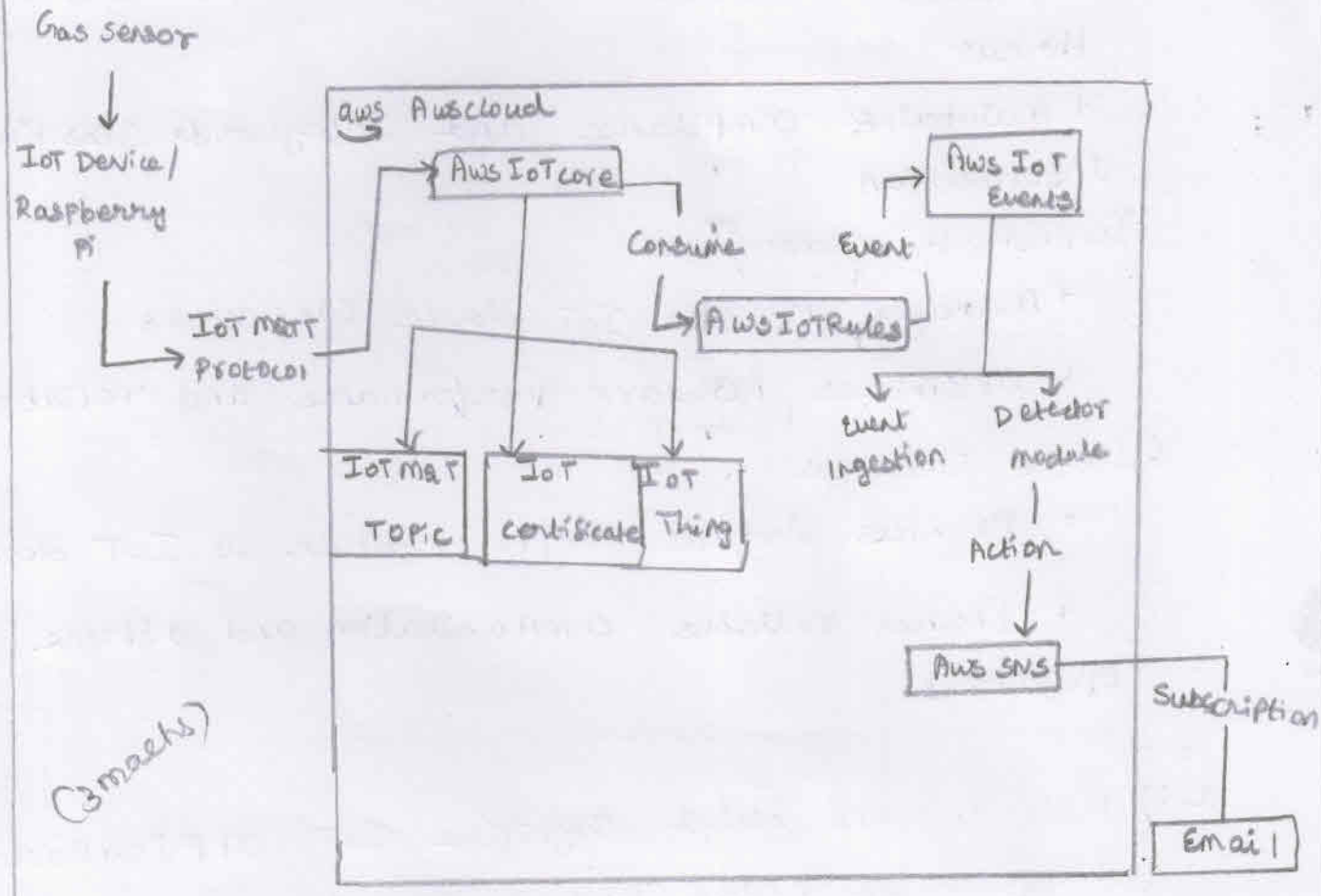
6) Device Shadows:

* Provides virtual representation of IoT devices
* Enables reliable communication and offline operations.

15

a ii) How do we build cognitive IoT applications using rules engine explain with examples (10 marks). Suitable Aws service: [2 marks]

* Aws IoT core service can be used to read data from various IoT devices and provide appropriate response based on the emission level of the gases.
* IoT devices such as Gas sensors and smoke detectors connected with Raspberry Pi send notifications to an Aws IoT core which triggers events to Aws IoT events.
* The detector model will then decide the pattern using States and trigger action based on the gas emission level and events



(3 marks)

AWS Components used - ~~2 marks~~

* AWS IoT core: IoT Policy, IoT Things, Rule
* AWS IoT Events: Input, Detector Model
* AWS Detector Model
* AWS SNS

Steps to be followed: ~~5 marks~~ (7 marks)

Step 1: Create an AWS IoT Policy.

* AWS IoT Console → Secure → Policies → Create a policy.

→ Create an AWS IoT Policy document that will authorize your device to interact with AWS IoT services. Select the following actions: IoT: connect, IoT: Receive, IoT: Publish, IoT: Subscribe or IoT: *

→ Under Effect, choose Allow, and then choose Create.

Step 2: Create a thing in AWS IoT

* A thing is a representation of a specific device or logical entity. In this case, it's the Pi.

→ AWS IoT Console → Manage → Things → Create → Create a single thing

→ Generate a certificate for the thing. Create thing and download the certificate, private key, and the root CA cert for AWS.

Step 3: SNS Topic

SNS → TOPICS → Create TOPICS in SNS → Subscriptions → Create Subscription

Step 4: Create an IoT Event Input

AWS IoT Event Console → Inputs → Create Input

Step 5: Create a Detector model

AWS IoT Events Console → Detector models → Create detector model

Step 6: Create an IoT rule

AWS IoT Console → Act → Rules → Create

* Rule query: SELECT * from 'sensors'

* Action: Send a message to an IoT Events Input, choose the Input created in the previous step.

5

b. i) List the benefits of device Shadow (6 marks)

[6*1] = 6 marks

1) Remote Device Control

Allows for remote control and management.

2) Consistent Device Interaction

Ensure consistent interaction with devices

3) Offline Operations

Enable offline operations by storing the last state.

4) Reliable Communication

Facilitate reliable communication between devices and cloud application

5) Enhanced User Experience

Providing uninterrupted access to device functionalities and data

6) Seamless Integration

Simplified development and deployment processes while improving reliability and performance

15 & ii) What are the protocols for communication with and between devices provide a brief note of them [10 marks]

(Any 5) [5*2=10]

1) MQTT (Message Queuing Telemetry Transport)

* Lightweight and efficient protocol designed for constrained devices
* Suitable for low bandwidth, high latency or unreliable networks

2) HTTP (Hyper Text Transfer Protocol):

* Widely used protocol for communication over the internet
* Operates in a client-server architecture with request-response messages

3) CoAP (Constrained Application Protocol):

* Lightweight Protocol designed for constrained devices and Low Power Networks
* Similar to HTTP but optimized for IoT applications.

4) AMQP (Advanced Message Queuing Protocol):

* Messaging protocol supporting reliable, asynchronous communication between devices.
* Provides features like message queuing, routing and delivery assurance.

5) DDS (Data Distribution Service):

* Data-centric Publish-Subscribe Protocol for real time, distributed systems.

* Enables high performance, Scalable Communication between devices.

6) WebSockets:

* Communication protocol providing full-duplex channels over a single TCP connection.

* Enables bi-directional, low-latency Communication between devices and servers

7) Bluetooth (BLE - Bluetooth Low Energy)

* Wireless communication Protocol Optimized for low power devices with short range Communication requirements

* Enables energy efficient communication between devices.