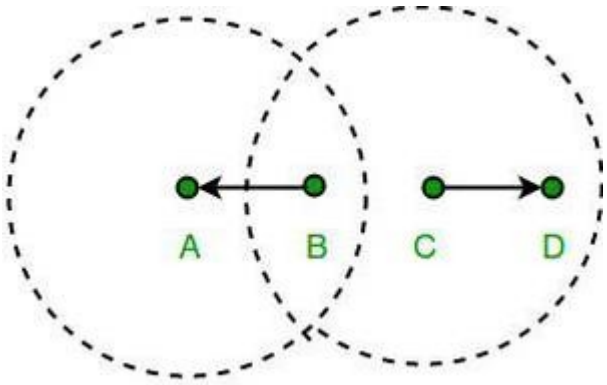
IOT ASSIGNMENT – 2

1) Explain about hidden and exposed problems with the help of diagram and how it can be resolved.

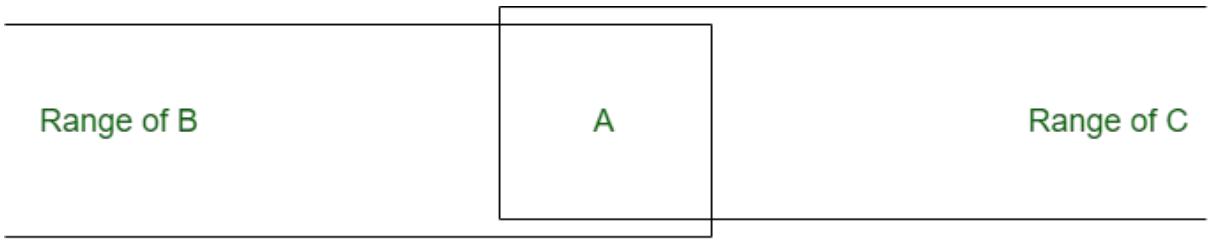
Exposed Terminal Problem:
Occurs when a node (C) unnecessarily defers transmitting to its intended receiver (D) because it overhears a nearby node (B) transmitting to a *different* receiver (A). Even if C's transmission wouldn't interfere with A receiving from B, C waits after hearing B, thus reducing network throughput.

- *Example:* B transmits to A. C wants to transmit to D. C hears B and waits, although C transmitting to D would not cause interference at A.



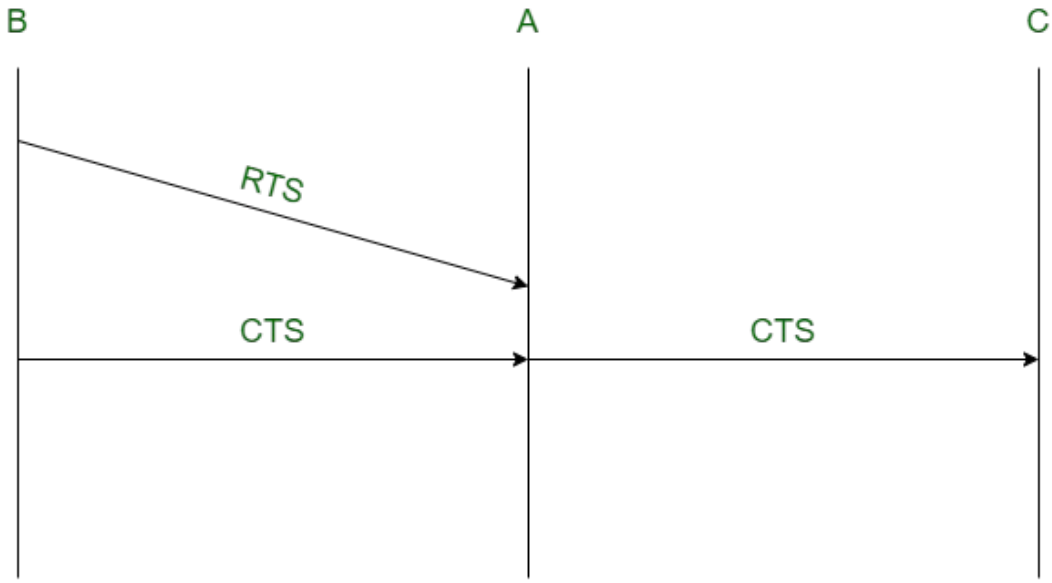
Hidden Terminal Problem:
Happens when two nodes (B and C) are out of range of each other, but both are within range of a common receiver (A). If B and C transmit to A simultaneously without hearing each other, their signals collide at A, reducing network capacity.

- *Example:* B sends to A. C, unable to hear B, also sends to A, causing a collision.



Hidden Station Problem

- *Resolution:* Often solved using **RTS/CTS** (Request to Send/Clear to Send). B sends RTS to A; A replies with CTS. Node C hears the CTS and knows the channel around A is reserved, thus preventing the collision.



Use of handshaking to prevent hidden station problem

2) What are the design goals of a MAC protocol for an ad hoc wireless sensor network?

1. **Energy Efficiency:** Paramount goal; minimize energy waste from idle listening, collisions, overhearing, and protocol overhead. Enable sleep/duty cycles to maximize network lifetime.
2. **Scalability:** Efficiently handle potentially large numbers of nodes and high density.
3. **Reliability:** Ensure dependable data delivery despite unreliable wireless links and node failures.
4. **Latency:** Provide timely data delivery, especially for critical applications, balanced against energy costs.
5. **Throughput:** Maximize the rate of successful data transmission relative to available bandwidth and energy.
6. **Fairness:** Offer equitable channel access opportunities to all nodes.
7. **Adaptability:** Adjust effectively to dynamic changes in network topology, traffic load, and channel quality.

3) Discuss the contention-based MAC protocol with scheduling mechanisms.

These **hybrid MAC protocols** combine elements of both **contention-based access** (like CSMA/CA, where nodes compete for the channel) and **scheduled access** (like TDMA, where resources are pre-allocated).

The goal is to gain the flexibility of contention for bursty or low-priority traffic while using scheduling's efficiency and collision-free nature for regular data or traffic requiring Quality of Service (QoS) guarantees.

Typically, this involves dividing channel access time:

1. A **Contention Period:** Nodes use contention (e.g., CSMA/CA) to transmit small amounts of data, send signaling/requests, or compete for reservations.
2. A **Scheduled Period:** Time slots or other resources are pre-allocated or reserved (often based on requests made during the contention period) for specific nodes to transmit without collisions.

A key example is **IEEE 802.15.4**

4) What is an embedded system on an IoT device? What are the primary hardware components that make up an embedded system?

An **embedded system in an IoT device** is its integrated, specialized computer "brain." It's dedicated to controlling the device's functions: reading sensors, processing data, controlling actuators, and managing network communication, often under power and memory constraints.

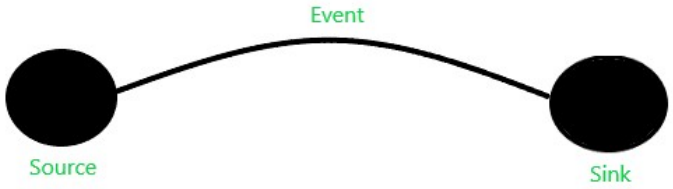
The **primary hardware components** are:

1. **Processor (MCU/MPU):** Executes the code.
2. **Memory:** Flash/ROM (for storing code) and RAM (for running code).
3. **Input/Output (I/O) Interfaces:** Pins (Digital/Analog) and converters (ADC/DAC) to connect with sensors/actuators.
4. **Communication Interfaces:** Modules for wired (Ethernet, USB, Serial) or wireless (Wi-Fi, BLE, LoRaWAN, etc.) connectivity.
5. **Power Supply:** Manages and regulates power (from battery or mains).

5) What is Data aggregation & dissemination?

Data Aggregation: This is the process in Wireless Sensor Networks (WSNs) where intermediate nodes (like routers or cluster heads) collect data from multiple sensor nodes. If the data relates to the same phenomenon or attribute, the intermediate node processes it (e.g., averages it, finds the minimum/maximum) using techniques like LEACH or TAG. This combined or summarized data, rather than all the raw data, is then forwarded towards the sink node. The primary goals are to reduce data redundancy, minimize traffic (solving implosion/overlap), save energy, and improve bandwidth utilization. Security and integrity are important considerations during this process.

Data Dissemination: This refers to the methods used to transmit sensor data from source nodes to interested sink nodes (or other relevant nodes) within the WSN. It often involves a two-step process: first, sinks propagate their "interest" in certain data/events throughout the network; second, when a source detects relevant data, it sends it towards the interested sinks, often following paths established by the interest propagation. Common methods include basic Flooding (broadcasting to all neighbors) and Gossiping (random forwarding), which are simple but can be inefficient, and more advanced techniques designed to reduce redundancy and energy use.



6) Explain survey routing protocols

Routing protocols in WSN/IoT determine data paths from sensors to a sink, focusing on energy efficiency due to device constraints. They are classified mainly by:

1. **Network Structure:**
    - **Flat:** All nodes are peers; data hops node-by-node (e.g., Flooding, SPIN). Simple but can be inefficient and scale poorly.
    - **Hierarchical:** Nodes form clusters managed by Cluster Heads (CHs) that aggregate data and route it (e.g., LEACH). Improves scalability and energy efficiency but adds overhead for cluster management.
    - **Geographical:** Uses node location for forwarding decisions (e.g., Greedy Forwarding). Avoids route maintenance overhead but requires location services and can fail in sparse areas ("voids").
2. **Protocol Operation (Route Discovery):**
    - **Proactive (Table-Driven):** Continuously maintain routes; low latency but high overhead, less common in WSNs.
    - **Reactive (On-Demand):** Find routes only when needed (e.g., AODV, DSR); lower steady-state overhead but higher initial latency.
    - **Hybrid:** Combine proactive (local) and reactive (remote) approaches to balance latency and overhead.

7) Explain the following IOT application in details:

Smart Metering

Smart metering replaces traditional utility meters with digital ones that automatically transmit consumption data (electricity, gas, water) wirelessly to the utility provider using technologies like Cellular, LPWAN, or RF Mesh. This eliminates manual readings, enables accurate billing, allows consumers real-time usage monitoring.

Smart Cards
Smart cards embed a chip into a plastic card for secure data storage, processing, and often wireless communication via NFC or RFID (contactless) or through physical contacts. Offering enhanced security over magnetic stripes, they are used for secure payments (EMV).

E-Health (Connected Health)
E-Health utilizes IoT to improve healthcare delivery by connecting patients, devices, and providers. It involves wearable sensors for remote patient monitoring (RPM) transmitting vital signs, connected medical equipment in hospitals, smart implants, and telehealth platforms.

Home Automation (Smart Home)
Home automation integrates household devices like lighting, thermostats, security systems, and appliances onto a network (using Wi-Fi, Zigbee, Z-Wave, etc.). This allows users remote control via apps or voice assistants and enables automation based on schedules or sensor data (e.g., motion, temperature).

8) Explain the difference between a "void setup()" function and a "void loop()" function in Arduino programming.

Feature	void setup()	void loop()
Purpose	Initialization and one-time setup	Main program logic, continuous execution
Execution	Runs **once** at power-on or reset	Runs **repeatedly** after setup() finishes
Common Uses	- Set pin modes (pinMode())	- Read sensors (digitalRead/analogRead)
	- Start serial (Serial.begin())	- Control actuators (digitalWrite/analogWrite)

9) What is a library in Arduino? Name some commonly used Arduino libraries?

What is a Library: Pre-written, reusable Arduino code that simplifies complex tasks or hardware control (like sensors or networking). Libraries encapsulate functionality, allowing programmers to use powerful features with just a few lines of code in their sketch, rather than writing everything from scratch.

Common Libraries: Servo (motors), Wire (I2C), SPI, WiFi/Ethernet (networking), LiquidCrystal (LCDs), SD (storage).

10) Write a program to turn on built in LED of the arduino Uno for 1 sec and off for 1 sec.

```
void setup() {

  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  digitalWrite(LED_BUILTIN, HIGH);
  delay(1000);
  digitalWrite(LED_BUILTIN, LOW);
  delay(1000);
}
```