UNIT 3

Q1) **Low Energy Adaptive Clustering Hierarchy (LEACH)**  
LEACH is a hierarchical, energy-efficient routing protocol designed for wireless sensor networks. It organizes nodes into clusters to minimize energy consumption.  
**Key Features and Operation:**  
1. **Clustering:** LEACH divides the sensor nodes into several clusters.  
2. **Cluster Head Selection:** Nodes within each cluster take turns assuming the role of a cluster head. This rotation helps distribute the energy load among the nodes.  
3. **Hierarchical Structure:** The protocol follows a hierarchical approach where nodes communicate with their designated cluster head, and the cluster head then communicates with the base station.

4. **Intra-Cluster Communication (Node to Cluster Head):**   
- LEACH utilizes Time Division Multiple Access (TDMA) for communication between nodes and their respective cluster heads.  
- The cluster head is responsible for creating and broadcasting a TDMA schedule to all nodes within its cluster.  
- This schedule assigns specific time slots to each node for data transmission, preventing collisions and allowing nodes to turn off their radios during other nodes' transmission slots. This leads to a low-duty cycle operation, conserving energy.

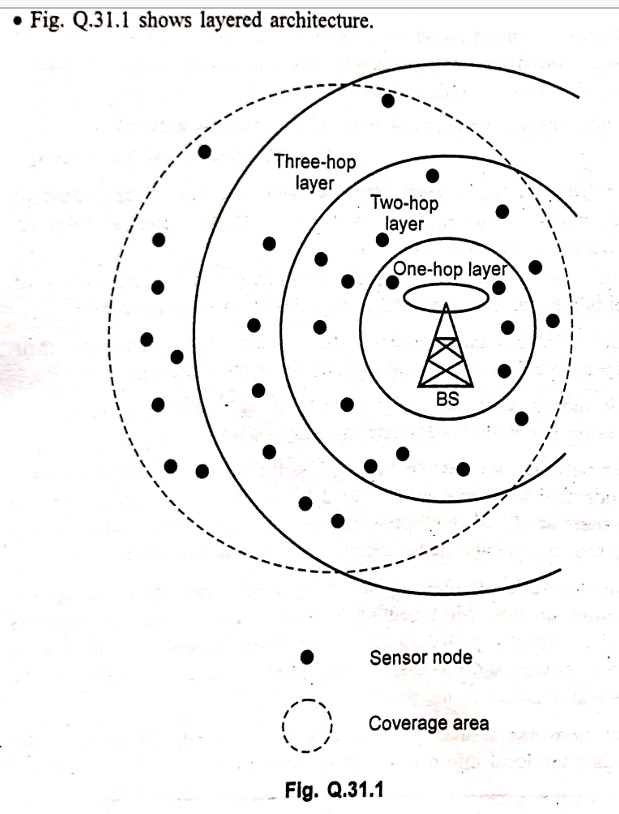
5. **Inter-Cluster Communication (Reducing Interference):**   
To reduce interference between different clusters, LEACH employs a transmitter-based code assignment scheme using Direct-Sequence Spread Spectrum (DSSS).  
Each cluster is assigned a unique spreading code, which all nodes in that cluster use to transmit data to their cluster head.

6. **Cluster Head to Base Station Communication:**   
Cluster heads aggregate the data received from their cluster members.  
They then communicate with the base station using a fixed spreading code and Carrier Sense Multiple Access (CSMA).  
Before transmitting, a cluster head senses the channel to ensure no other cluster head is currently transmitting data using the base station spreading code. If the channel is busy, the cluster head delays its transmission until the channel is idle.

7. **Synchronization:** LEACH assumes that cluster nodes start the cluster setup phase simultaneously and remain synchronized thereafter. The base station can assist in synchronization by sending out synchronization pulses to all nodes.

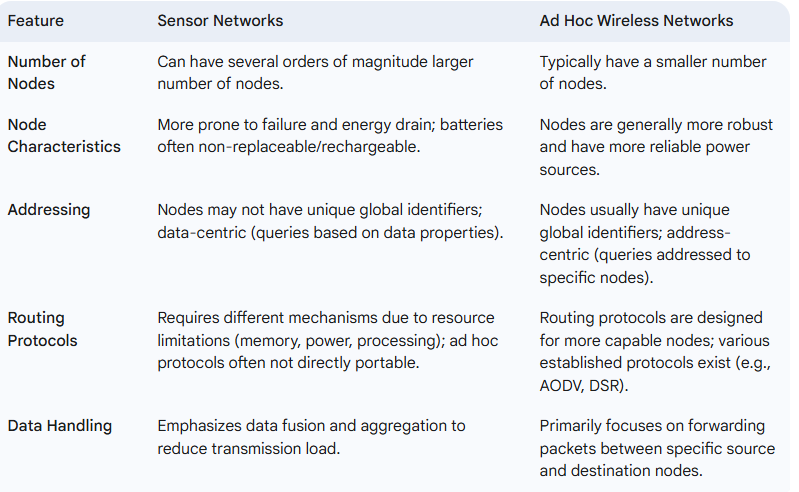
8. **Energy Efficiency:** The use of clustering, rotated cluster head roles, and TDMA scheduling significantly reduces energy consumption by allowing nodes to power down their radios when not actively sending or receiving data.

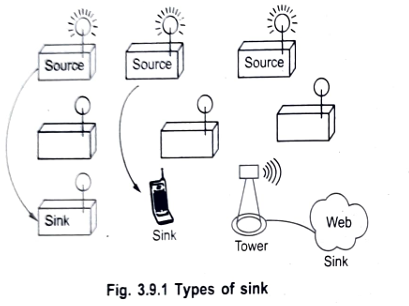
**Advantages (related to schedule-based nature):**  
**Contention-Free:** The scheduled nature of communication within clusters (TDMA) eliminates collisions.  
**Energy Saving:** Nodes can turn off their radios during allocated idle slots, significantly reducing energy waste and extending the network lifetime.

**Disadvantages (as a schedule-based MAC protocol):**  
**Limited Peer-to-Peer Communication:** The hierarchical structure primarily supports communication between nodes and their cluster heads, making direct peer-to-peer communication difficult unless nodes listen during all time slots.  
**Synchronization Challenges:** Achieving and maintaining time synchronization among distributed sensor nodes can be difficult and costly, especially in energy-constrained networks.  
**Requires Additional Mechanisms:** Overcoming inter-cluster communication issues and interference may require additional mechanisms like FADMA or CDMA.  
**Limited Scalability and Adaptability:** TDMA-based MAC protocols can have limited scalability and may not easily adapt to node mobility or changes in network traffic and topology.  
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  
Q2) **Layered Architecture & Clustered Architecture**  
In a layered architecture, the sensor network typically features a single, powerful base station (BS). The sensor nodes are organized into distinct layers based on their hop count from the base station. Nodes at the same hop count form a layer.  
**Structure:** A central base station with sensor nodes arranged in concentric layers, where each layer represents a specific number of hops away from the base station.  
**Communication:** Nodes in each layer communicate with nodes in adjacent layers or directly with the base station depending on their hop count. Data is typically relayed hop-by-hop towards the base station.  
**Applications:** Layered architectures have been used in contexts like in-building wireless backbones and military sensor-based infrastructure (such as the multi-hop infrastructure network architecture - MINA).  
**Related Framework:** The Unified Network Protocol Framework (UNPF) is mentioned as a set of protocols for implementing sensor networks with potentially a layered structure. UNPF integrates network initialization and maintenance, MAC, and routing protocols.   
**Network Initialization and Maintenance Protocol:** Organizes nodes into different layers based on the BS's broadcast capability. The BS broadcasts its identifier, and nodes record the BS ID upon receiving the broadcast.  
**MAC Protocol:** Network initialization occurs on a common control channel. During data transmission, the Distributed TDMA Receiver Oriented Channel (DTROC) assignment MAC protocol is used for reception, with channel reuse to avoid collisions.  
**Routing Protocol:** Downlink from the BS is a direct broadcast. The layered architecture facilitates multi-hop data forwarding from sensor nodes to the BS.  
Below is a diagram illustrating the layered architecture (Fig. Q.31.1):  
  
**Clustered Architecture**  
In a clustered architecture, the sensor nodes are organized into groups called clusters. Each cluster is managed by a designated node known as the clusterhead. The clusterhead acts as an intermediary between the sensor nodes in its cluster and the base station (BS).  
**Structure:** The network is divided into multiple clusters. Within each cluster, there are regular sensor nodes and a clusterhead. The clusterheads communicate with the base station, which often serves as an access point connected to a wired network.  
**Communication:** Sensor nodes within a cluster typically send their data to their respective clusterhead. The clusterhead is responsible for collecting data from all the nodes in its cluster, potentially performing data aggregation or fusion, and then transmitting the aggregated data to the base station.  
**Suitability:** Clustered architecture is particularly well-suited for sensor networks because it allows for efficient data aggregation and fusion at the clusterheads. By combining data from multiple nodes before sending it to the base station, the amount of data transmission is reduced, which saves energy and network bandwidth.  
**Self-Organization:** Sensor networks employing clustered architectures should ideally be self-organizing. This means that the formation of clusters and the election of clusterheads should be an autonomous process carried out by the nodes themselves, rather than requiring manual configuration. Protocols like LEACH (Low Energy Adaptive Clustering Hierarchy) are examples of algorithms that enable this type of autonomous cluster formation and cluster head rotation.

Below is a diagram illustrating the clustered architecture (Fig. Q.31.2):  
A diagram of a cluster

AI-generated content may be incorrect.

Q3)   
  
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  
Q4**) the key issues and challenges in designing sensor networks are as follows:**  
1. **Infrastructure-less Nature:** Sensor networks typically operate without a fixed infrastructure. Therefore, all routing and network maintenance algorithms need to be distributed among the nodes.  
2. **Energy Constraints:** Sensor nodes heavily rely on their limited battery power, which is often not rechargeable or replaceable. The available energy at each node is a major constraint that must be carefully considered when designing protocols.  
3. **Hardware Design:** The hardware design for sensor nodes must prioritize energy efficiency. The microcontroller, operating system, and application software should all be designed to conserve power.  
4. **Synchronization:** Sensor nodes need to be able to synchronize with each other in a completely distributed manner to coordinate operations.  
5. **Adaptability to Changing Connectivity:** A sensor network should be capable of adapting to changes in network connectivity, such as those caused by node failures or the addition of new nodes powering up. Routing protocols, in particular, need to dynamically adjust to include or avoid sensor nodes in their paths.  
6. **Real-time Communication and QoS:** Supporting real-time communication over sensor networks is a challenge. This requires providing guarantees on factors like maximum delay, minimum bandwidth, and other Quality of Service (QoS) parameters.  
------------------------------------------**-------------------------------------------------------------------------------------------------------------------------------------  
Q5) Sensor Network Architecture**Sensor network architectures uses two main concept source and sink.

Source provides information to network and sink receive information from network. Sink receives the information from source entity. Example of source is sensor node acts as source entity. Examples of sink are sensor network, PDA and entity outside home network. Fig. 3.9.1 shows three types of sink.   
 Fig. 3.9.1 Types of sink

If direct communication is not possible because of distance limit or obstacles, then multihop communication method is used. Multihop communication uses store and forward fashion. Node has to receive a packet properly before it can forward next entity.   
Proper placing of intermediate sensor node is necessary. Fig. 3.9.2 shows multihop communication network.

Fig. 3.9.2 Multihop communication  
A diagram of a block diagram

AI-generated content may be incorrect.  
Traditional transport protocols such as UDP and TCP cannot be directly implemented in sensor networks because if a sensor node is far away from the sink then the flow and congestion control mechanism cannot be applied for those nodes.   
UDP on the other hand has a reputation of not providing reliable data delivery and has no congestion or flow control mechanisms which are needed for sensor networks.

**1. Components of Sensor Network Architecture:  
a. Sensor Nodes:**Basic building blocks of WSN.Each node contains: **Sensing Unit** – to sense data (e.g., temperature, pressure). **Processing Unit** – usually a microcontroller for data processing. **Transceiver Unit** – for wireless communication. **Power Unit** – battery or energy-harvesting source.

**b. Sink/Base Station:**Collects data from sensor nodes.Acts as a gateway to other networks (e.g., the Internet).Has more power, memory, and processing capability.

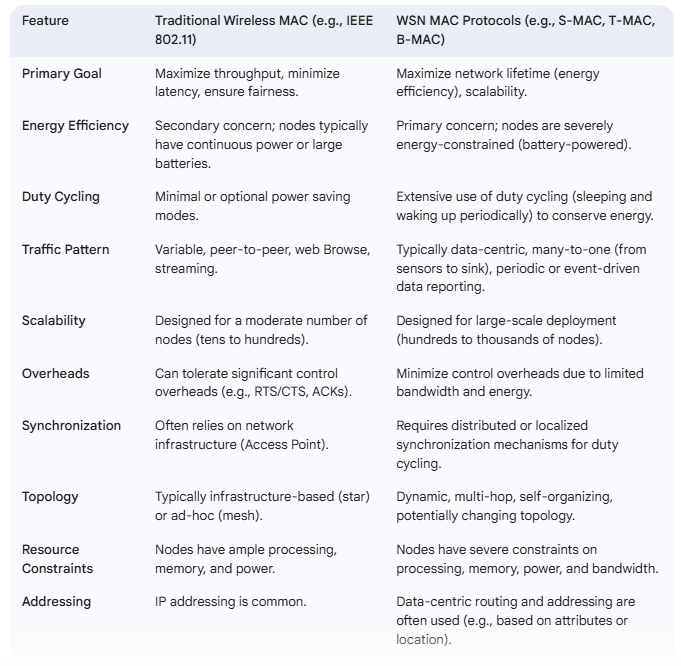
**c. Task Manager Node/User Interface:**Allows users to query or control the network.Can be part of the base station or a separate system.

**2. Architectural Models:  
a. Flat Architecture:**All sensor nodes are equal and perform the same functions.  
Data is routed in a peer-to-peer manner or multi-hop fashion.Suitable for small networks but not energy-efficient for large deployments.

**b. Hierarchical Architecture (Clustered):**Nodes are grouped into clusters  
Each cluster has a Cluster Head (CH) responsible for collecting and forwarding data to the base station.Reduces communication overhead and conserves energy.Example: LEACH (Low-Energy Adaptive Clustering Hierarchy).

**c. Tiered Architecture:**Nodes are organized in multiple tiers with different roles (e.g., sensors, aggregators, gateways).Useful in large-scale and heterogeneous WSNs.  
------------------------------------------**-------------------------------------------------------------------------------------------------------------------------------------  
Q6) Differentiation between MAC Protocols of WSN and Traditional Wireless MAC Protocols:**

Medium Access Control (MAC) protocols govern how nodes in a shared wireless medium access the channel to transmit data, avoiding collisions. MAC protocols for WSNs differ significantly from traditional wireless MAC protocols (like those used in Wi-Fi - IEEE 802.11) due to the unique characteristics and constraints of sensor networks:

****------------------------------------------**-------------------------------------------------------------------------------------------------------------------------------------  
Q7)** **short note on Wireless Sensor Networks (WSNs):**A wireless sensor network is a collection of nodes organized into a co-operative network. Each node consists of processing capability, may contain multiple types of memory, have a RF transceiver and a power source and accommodate various sensors and actuators. The nodes communicate wirelessly and often self-organize after being deployed in an ad-hoc fashion.

WSN is a wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental, conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations.

A communication network is composed of nodes, each of which has computing power and can transmit and receive messages over communication links, wireless or cabled.

The development of wireless sensor networks was originally motivated by military applications such as battlefield surveillance. However, wireless sensor networks are now used in many civilian application areas, including environment and habitat monitoring, healthcare applications, home automation and traffic control.Possible applications

1. Military Battlefield surveillance, biological attack detection, targeting.

2. Ecological: Fire detection, flood detection, agricultural uses.

3. Health related: Human physiological data monitoring.

4. Miscellaneous Car theft detection, inventory control, home applications.

Sensor network development rely on advances in sensing, communication and computing. To manage scarce WSN resources adequately, routing protocols for WENs need to be energy-aware.

Data-centric routing and in-network processing are important concepts that are associated intrinsically with sensor networks. The end-to-end routing schemes that have been proposed in the literature for mobile ad-hoc networks are not appropriate WSNs; data-centric technologies are needed that perform in-network aggregation of data to yield energy efficient dissemination.

A sensor node typically has embedded processing capabilities and onboard storage; the node can have one or more sensors operating in the acoustic, seismic, radio (radar), infrared, optical, magnetic and chemical or biological domains. The node has communication interfaces, typically wireless links, to neighbouring domains. The sensor node also often has location and positioning knowledge that is acquired through a Global Positioning System (GPS) or local positioning algorithm.

Sensor nodes are scattered in a special domain called a sensor field. Each of the distributed sensor nodes typically has the capability to collect data, analyze them and route them to a designated sink point.  
------------------------------------------**-------------------------------------------------------------------------------------------------------------------------------------  
Q8)** **What are the design issues in wireless sensor network?**1. **Fault tolerance:** The possibility of node failure and changes in network topology is high in WSNs. Therefore, the network design must be robust and reliable to handle node failures and topology changes effectively.

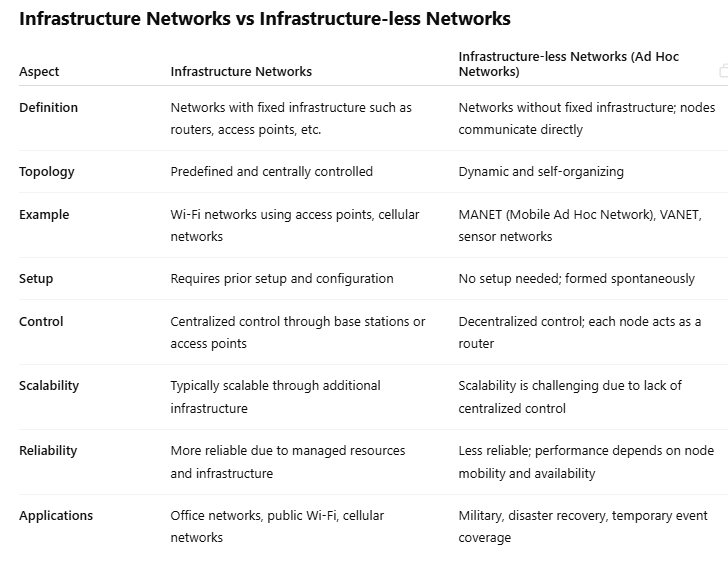
2. **Scalability:** The design of a WSN should support the addition of new nodes at any time and be capable of supporting a large number of nodes.

3. **Environment:** The WSN should be designed to survive and operate regardless of the environmental conditions in which it is deployed.

4.**Heterogeneity support:** The protocols designed for WSNs should support different kinds of sensor nodes and be able to accommodate a variety of applications.

5. **Autonomous operations:** The WSN should be capable of organizing, reorganizing, and operating autonomously, especially since WSNs are often deployed in locations where human intervention is not feasible.

6. **Limited memory and processing capability:** Sensor nodes have very limited memory, processing power, and energy. Consequently, all design aspects of a WSN should not be demanding in terms of processing requirements or memory usage.  
------------------------------------------**-------------------------------------------------------------------------------------------------------------------------------------**  
Q9)

  
------------------------------------------**-------------------------------------------------------------------------------------------------------------------------------------  
Q10)** **Dynamic Source Routing (DSR) protocol:**The Dynamic Source Routing (DSR) protocol is a key routing protocol designed for use in multi-hop wireless ad hoc networks, particularly those with mobile nodes. Unlike traditional routing protocols that rely on pre-established routes or central control, DSR is a **reactive** or **on-demand** protocol, meaning that routes are discovered only when they are needed by a source node to send data to a destination.

A fundamental characteristic of DSR is that the network is **completely self-organizing and self-configuring**. This eliminates the need for any existing network infrastructure or central administration, making it suitable for dynamic environments where the network topology changes frequently.

**The Concept of Source Routing**DSR employs the concept of **source routing**. In this approach, the sender of a data packet determines the complete sequence of intermediate nodes that the packet must traverse to reach its destination. This complete path information is then included in the header of the data packet itself. Intermediate nodes forwarding the packet do not need to perform a routing lookup; they simply follow the explicit path specified in the packet header.

The primary advantage of source routing highlighted in the text is that **intermediate nodes do not need to maintain up-to-date routing information** for all possible destinations to forward packets. Their role is primarily to forward the packet along the specified path.

**DSR Mechanisms: Route Discovery and Route Maintenance**DSR is primarily composed of two essential mechanisms:

1. **Route Discovery:** This process is initiated by a source node when it needs to send a packet to a destination for which it does not currently have a valid route in its route cache. The Route Discovery process works as follows:
   * The source node broadcasts a **Route REQUEST (RREQ)** message. The RREQ message contains the destination address, the source address, and a unique identification number to prevent processing duplicate RREQs.
   * When an intermediate node receives an RREQ, if it has not processed the same RREQ before (checked using the source address and identification number) and is not the destination, it appends its own address to the list of traversed nodes in the RREQ header and re-broadcasts the RREQ.
   * This flooding process continues until the RREQ reaches the destination node or an intermediate node that has a valid route to the destination in its route cache.
   * Once the RREQ reaches the destination node (or an intermediate node with a valid route), the destination (or the intermediate node) sends a **Route REPLY (RREP)** packet back to the source.
   * The RREP packet carries the path information collected by the RREQ (the sequence of nodes from the source to the destination). This path information is used by the source to learn the route to the destination. The RREP traverses backward along the path recorded in the RREQ packet header.
2. **Route Maintenance:** This mechanism is responsible for handling situations where the network topology changes, causing an existing route to become invalid (e.g., due to a link break or node failure).
   * Route failure is typically detected by the failure of message transmissions (e.g., lack of acknowledgment at the link layer).
   * When a node (either an intermediate node or the destination) detects a link break or failure on a path it is using or forwarding packets on, it initiates a **Route ERROR (RERR)** message.
   * The RERR message is sent back towards the source of the packets that were using the failed route.
   * When the source and any intermediate nodes receive the RERR message, they remove the broken link and any routes that utilize that link from their respective route caches.
   * If the source needs to send data to the destination again after receiving an RERR, it will need to initiate a new Route Discovery process to find a fresh route. A route maintenance procedure might also be initiated if a route from the cache is found but is no longer valid.

**Route Cache**Each node participating in the DSR protocol maintains a **route cache**. This cache stores the source routes that the node has learned, either through initiating Route Discovery or by overhearing RREP packets or data packets that contain source routes. Before transmitting a packet, a source node checks its route cache for a valid route to the destination. If a valid route exists, it uses that route. If not, it initiates Route Discovery.In summary, DSR is a dynamic, reactive routing protocol for wireless ad hoc networks that leverages source routing to avoid the need for intermediate nodes to maintain global routing tables. Its two main components, Route Discovery and Route Maintenance, enable it to adapt to the constantly changing topology of mobile ad hoc environments.

Q11) **Destination Sequenced Distance Vector (DSDV) Routing Protocol**DSDV was one of the first **proactive** (or table-driven) routing protocols developed for Ad-hoc networks. Unlike reactive protocols (like DSR) that discover routes on demand, proactive protocols maintain routes to all possible destinations at all times, regardless of whether they are currently needed.  
**A diagram of a diagram

AI-generated content may be incorrect.**

**Algorithm Basis**The DSDV protocol is based on the classic **Bellman-Ford algorithm**. However, it incorporates enhancements to make it suitable for mobile ad-hoc networks, primarily by using sequence numbers to prevent routing loops and stale route information.

**Routing Table Structure**In DSDV, each node in the network maintains a **routing table**. This table contains an entry for every reachable destination within the network. Each entry in the routing table includes the following information:

* **Destination:** The address of the node that is the target of the route.
* **Next Hop:** The address of the neighbor node to which a packet for the destination should be forwarded.
* **Metric:** The cost of the route to the destination, typically represented by the number of hops.
* **Sequence Number:** A time indication originated by the destination node. This sequence number is crucial for maintaining route freshness and preventing loops.

**Routing Table Updates**Routing tables in DSDV are updated through periodic exchanges between neighboring nodes. Each node periodically broadcasts its entire routing table or recent updates to its immediate neighbors. There are two main ways these updates can be performed:

1. **Full Dump:** The node sends its complete routing table to its neighbors. This is typically done periodically and is preferred when there are frequent changes in the network topology.
2. **Incremental Updates:** The node only sends entries that have been recently updated. This is more efficient in stable topologies as it reduces the amount of routing traffic.

Nodes receive these updates and use the information to update their own routing tables. When a node receives an update from a neighbor, it compares the received route information with the entries in its own table. The decision to update an entry is based on the sequence number and the metric:

* If a node receives an update for a destination with a higher sequence number than the one it currently has in its table, it always updates its entry to use this new route, regardless of the metric. A higher sequence number indicates a fresher route originated by the destination.
* If a node receives an update with the same sequence number, it then chooses the route with the better metric (e.g., a lower hop count).
* If a node receives an update with a lower sequence number, it ignores the update as it indicates stale route information.

Updates are performed on a regular basis (periodically), and they are instantly scheduled if a new event, such as a link failure or a new link establishment, is detected in the topology. This allows the network to react relatively quickly to topology changes.

**Route Selection**When multiple routes to the same destination are known, DSDV uses a combination of the **metric** and the **sequence number** to select the best route. The primary criterion is the sequence number; the route with the highest sequence number is always preferred as it is considered the freshest route. If two routes have the same sequence number, the one with the better metric (fewer hops) is chosen. This use of sequence numbers guarantees a **loop-free path**.

**Advantages and Disadvantages  
Advantages:**DSDV was an early and fundamental algorithm for ad-hoc networks.It guarantees loop-free paths. **Disadvantages:**DSDV requires regular updates of its routing tables, which consumes battery power and a significant amount of bandwidth even when the network is idle. This makes it less energy-efficient compared to reactive protocols.Whenever the topology changes, a new sequence number is necessary before the network reconverges. This means DSDV is **not suitable for highly dynamic networks** where topology changes are very frequent, as the constant updates can lead to high overhead and slow convergence.  
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  
Q12) **Ad-hoc On-Demand Distance Vector (AODV) Routing Protocol**The Ad-hoc On-Demand Distance Vector (AODV) protocol is a widely used **stateless, on-demand distance vector routing protocol** designed for mobile ad-hoc networks (MANETs). Being an on-demand (reactive) protocol, AODV establishes routes only when they are requested by a source node for sending data, thus reducing overhead compared to proactive protocols in networks with infrequent traffic.

AODV is a **distance vector routing protocol**, meaning that routing decisions are based on the number of hops to the destination. A key feature of AODV is its support for **unicast, broadcast, and multicast communication**.

The performance of AODV is improved by keeping routing information in each node. While it is a distance vector protocol, it is considered "stateless" in the sense that intermediate nodes primarily maintain information about the *next hop* towards a destination, rather than the full source route like in DSR.  
A paper with text and symbols on it

AI-generated content may be incorrect.

AODV's functionality is primarily managed through two major processes:  
1. **Route Discovery:** This process is initiated by a source node when it needs to send data to a destination for which it does not have an active route.

* + The source node broadcasts a **Route Request (RREQ)** message to its neighbors. The RREQ contains the source address, destination address, the source's current sequence number, and the last known sequence number for the destination.
  + When an intermediate node receives an RREQ, it first checks if it has already processed this RREQ (using a unique ID and the initiator's address). If not, and if it does not have a valid route to the destination with a sequence number at least as great as the one in the RREQ, it increments the hop count in the RREQ and re-broadcasts it.
  + Importantly, as the RREQ travels through the network, each node that forwards it creates a **reverse path** entry in its routing table pointing back towards the source of the RREQ. This reverse path is used later for sending the Route Reply (RREP).
  + If an intermediate node has an active route to the destination with a destination sequence number greater than or equal to the one in the RREQ, or if the RREQ reaches the destination node itself, a **Route Reply (RREP)** message is generated.
  + The RREP is then unicasted back to the source node along the established reverse path. The RREP contains the destination address, the source address, the destination's current sequence number, and the hop count to the destination.
  + When the source receives the RREP, it records the route in its routing table and can begin sending data packets. Multiple RREP packets might arrive at the source if the RREQ traveled along different paths.

**2. Route Maintenance:** This process handles link breakages or node failures that occur on active routes.

* + AODV uses **periodic HELLO messages** to allow nodes to detect the connectivity of their neighbors. If a node fails to receive HELLO messages from a neighbor for a certain period, it assumes the link to that neighbor is broken.
  + When a node detects a link failure on an active route it is using or forwarding packets for, it propagates a **Route Error (RERR)** message to its upstream neighbors (neighbors closer to the source).
  + The RERR message indicates which destination(s) are now unreachable through this broken link.
  + Nodes receiving an RERR invalidate the routes associated with the specified destinations and further propagate the RERR towards the sources that were using those routes.
  + When the source node receives an RERR, it marks the route as invalid. If it still needs to communicate with that destination, it will initiate a new Route Discovery process.

**Characteristics of AODV**Support for unicast, broadcast, and multicast communication.Performs on-demand route establishment with relatively small delay.Uses multicast trees for connecting group members, which are maintained for the lifetime of the multicast group.Link breakages in active routes are efficiently repaired using RERR messages.All routes established by AODV are loop-free through the effective use of **sequence numbers**. Sequence numbers are crucial for tracking the freshness and accuracy of routing information.Nodes only keep track of the **next hop** for a route instead of the entire source route (unlike DSR).Uses periodic HELLO messages to track the connectivity of neighbors.

**Advantages and Disadvantages of AODV  
Advantages:**Routes are established on demand, which helps in finding the latest route to the destination.The connection setup delay for a route is relatively lower compared to some other protocols. **Disadvantages:**Multiple Route Reply (RREP) packets received in response to a single Route Request can lead to heavy control overhead.  
The periodic beaconing using HELLO messages leads to unnecessary bandwidth consumption, even when the network is idle.  
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------  
Q13) **the key issues in designing a routing protocol for Adhoc wireless networks are:  
1. Mobility:** Adhoc networks are highly dynamic due to the movement of nodes. This node movement causes frequent path breaks in the network. Repairing these broken paths can be slow, leading to slow convergence of the routing protocol. **2. Bandwidth constraint:** Wireless communication operates within a limited radio band, resulting in less available bandwidth compared to wired networks. Maintaining topology information with high data rates is difficult in this limited bandwidth. Frequent changes in topology further increase the overhead associated with topology maintenance. Therefore, routing protocols need to optimize bandwidth usage and design topology update mechanisms with minimal overhead. **3. Error-prone shared broadcast radio channel:** Wireless links exhibit time-varying characteristics in terms of link capacity and link-error probability. The shared broadcast nature of the radio channel also makes it susceptible to issues like the hidden terminal problem, which causes packet collisions. Routing protocols need to interact effectively with the MAC layer to identify and utilize better-quality links and mitigate the effects of collisions. **4. Resource constraints:** Adhoc network nodes, especially in scenarios involving sensor networks, often have limited battery life and limited processing power. Routing protocols must be designed to optimally manage these scarce resources to maximize network lifetime and performance.  
---------------------------------------------------------------------------------------------------**------------------------------------------------------------------------------Q14) MACAW (Multiple Access with Collision Avoidance for Wireless) protocol:**  
MACAW is a **Media Access Protocol designed for Wireless LANs**, built upon the foundation of the MACA (Multiple Access Collision Avoidance) protocol. Its primary goal is to address the challenges of collision avoidance in wireless environments, particularly the hidden and exposed terminal problems.

**Message Exchange Sequence:** MACAW extends the basic RTS/CTS handshake of MACA with additional control packets. The standard successful message exchange in MACAW typically involves the following sequence: **1. RTS (Request-To-Send):** The sender initiates communication by sending an RTS packet to the intended receiver. This packet includes the duration of the upcoming data transmission. **2. CTS (Clear-To-Send):** If the receiver is ready and able to receive, it replies with a CTS packet addressed to the sender. This packet also includes the duration of the transmission. Nodes hearing the CTS should defer their transmissions. **3. DS (Data Sending):** After receiving the CTS, the sender immediately transmits a DS (Data Sending) message. This signal ensures a 3-way handshake between the sender and receiver, informing stations within hearing distance of either the sender or the receiver about the impending data transmission. This helps to alleviate the hidden terminal problem. **4. DATA:** The sender transmits the actual data packet. **5. ACK (Acknowledgment):** Upon successful reception of the DATA packet, the receiver sends an ACK packet to the sender. This acknowledgment confirms successful delivery and helps with error recovery at the link layer. An extra ACK at the end, as opposed to relying solely on transport layer recovery, allows for faster retransmission if needed.

The basic exchange is described as replacing RTS-CTS-DATA-ACK with RTS-CTS-DS-DATA-ACK.

**Role of Control Packets  
RTS:** Sent by the sender to request permission to transmit and reserve the channel.  
**CTS:** Sent by the receiver to grant permission to transmit and signal other nodes within its range to defer.  
**DS:** Sent by the sender after receiving CTS to further inform nodes around both the sender and receiver about the ongoing transmission, aiding in synchronization and preventing starvation.  
**ACK:** Sent by the receiver to confirm successful data reception.  
**RRTS (Request-for-Request-to-Send):** This is an additional control packet used in MACAW. It is transmitted by a *receiver* on behalf of a sender to help it out of starvation. It essentially acts as a proxy RTS. However, the text notes that even RRTS cannot guarantee perfect fair contention in all scenarios, especially when the actual RTS sender is too far away.

**Design Observations :** The design of MACAW was based on four key observations: **1.** Relevant contention occurs *at the receiver*, meaning sensing the carrier at the sender (as in CSMA) is often insufficient to avoid collisions at the receiver's end due to the hidden terminal problem. **2.** Congestion is *location dependent*. Congestion experienced by one node might not be the same as that experienced by another. **3.** For fair allocation of the wireless medium, **collision (congestion) information must be shared among devices**. **4.** Information related to the contention period must be **synchronized among devices** to promote fair contention.

**Backoff Algorithm**MACAW employs a backoff algorithm called **MILD (Multiplicative Increase and Linear Decrease)**, which replaces the Binary Exponential Backoff (BEB) used in protocols like Ethernet. In MILD, the backoff interval grows relatively slowly (multiplicatively) when collisions occur but shrinks quickly (linearly to a minimum value) upon successful transmissions. To enable better congestion detection and fairer access, MACAW shares backoff timers among stations by putting this information in packet headers.

**Multiple Streams and Multicast  
Multiple Stream Model:** MACAW supports a multiple stream model where separate queues are maintained for each stream (or data flow) at each node for increased fairness. Each queue runs independent backoff algorithms, but stations attempting to communicate with the *same receiver* should use the same backoff value. **Multicast:** MACAW handles multicast by sending data packets right away after the RTS packet, without waiting for a CTS from multiple receivers. However, the text indicates that this approach suffers from the same problems as CSMA, leaving it as an open challenge according to the authors. **A diagram of a computer network

AI-generated content may be incorrect.**

Q15) **Multiple Access Collision Avoidance (MACA) Protocol**Multiple Access Collision Avoidance (MACA) is a foundational medium access control (MAC) protocol for wireless networks, proposed by Phil Karn in 1990. A key characteristic of MACA is that it **does not rely on carrier sensing** before transmission, which is a common technique in many other MAC protocols (like CSMA). Instead, MACA uses a handshake mechanism involving control packets to avoid collisions.

The core idea behind MACA is to address the **hidden terminal problem**, where a sender might not detect another transmission occurring at the receiver's location because the other sender is outside the first sender's radio range. This can lead to collisions at the receiver.

MACA utilizes a two-way handshake involving two control messages:  
1. **RTS (Request-To-Send):** When a node wants to transmit a data packet, it first sends an RTS packet to the intended receiver. The RTS packet typically includes the size of the data packet the sender intends to transmit.  
2. **CTS (Clear-To-Send):** If the receiver receives the RTS and is ready to receive, it replies with a CTS packet addressed back to the sender. The CTS packet also carries the duration of the upcoming data transmission.  
Before transmitting a data packet, nodes in MACA operate in an **RTS-CTS mode** to reserve the channel. When a node successfully completes the RTS/CTS handshake, it gains the right to transmit its data packet.

**How RTS/CTS helps with the Hidden Terminal Problem:**  
When a node (say, Node A) sends an RTS to a receiver (Node B), nodes within Node A's transmission range *hear* the RTS.  
When Node B replies with a CTS, nodes within Node B's transmission range *hear* the CTS.  
Crucially, a node that hears a CTS (even if it didn't hear the initial RTS from the sender) knows that a transmission is about to occur involving the receiver and should **defer** its own transmission to avoid colliding at the receiver. This is how MACA helps mitigate the hidden terminal problem.

**Limitations of Basic MACA:** The provided text points out some limitations of the basic MACA protocol:  
1. There is **no acknowledgment (ACK) packet** in the basic MACA scheme to confirm successful data reception.  
2. MACA **does not fully solve the hidden and exposed terminal problems**. While it helps with the hidden terminal problem, it does not completely eliminate it, and it does nothing to address the exposed terminal problem (where a node that could transmit safely is unnecessarily silenced).  
3. MACA **does nothing regarding the receiver blocked problem**, a scenario where a receiver is busy and cannot receive the intended transmission.  
4. The use of a simple **binary exponential backoff (BEB)** algorithm in MACA might lead to **starvation** for some nodes, a problem that is noted to be solved by the MACAW protocol (an enhancement of MACA).  
A diagram of a diagram

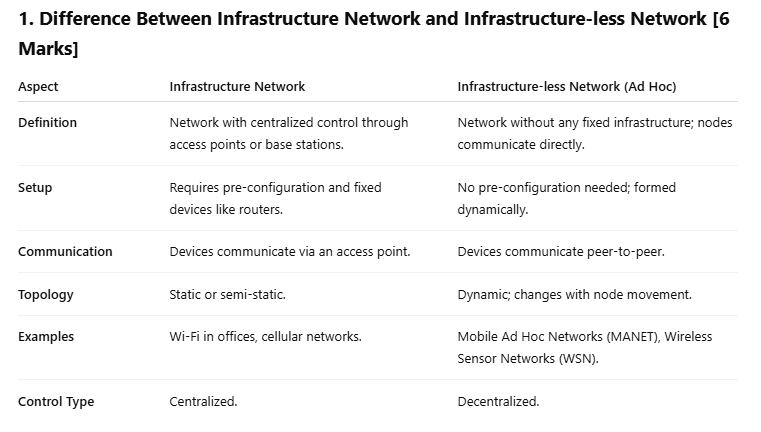
AI-generated content may be incorrect.  
---------------------------------------------------------------------------------------------------**------------------------------------------------------------------------------  
Q16)  
Design Goals for Ad hoc Network MAC layer**1. Protocol operation should be distributed through all the nodes. 2. In real time traffic, the protocol should provide QoS. 3. The average delay for packet transmission should be as small as possible. 4. The bandwidth should be utilized efficiently.   
5. Each node must have a fair share of the available bandwidth. 6. Control overhead should be minimized.   
7. The hidden and exposed terminal problem should be minimized. 8. The protocol must be scalable to large networks. 9. Power control mechanisms are needed for efficient management of the energy consumption of the nodes. 10. Time synchronization between the nodes should be provided.

**Design Issues for Ad hoc Network MAC layer   
Bandwidth efficiency** is defined at the ratio of the bandwidth used for actual data transmission to the total available bandwidth. The MAC protocol for ad-hoc networks should maximize it.  **Quality of service suppor**t is essential for time-critical applications. The MAC protocol for ad-hoc networks should consider the constraint of ad-hoc networks.  **Synchronization** can be achieved by exchange of control packets. Some mechanism has to be found in order to provide synchronization among the nodes. Synchronization is important or regulating the bandwidth reservation.

**Hidden and exposed terminal problems**: The reason for these two problems is the broadcast nature of the radio channel, namely, all the nodes within a node's transmission range receive its transmission.

**Hidden terminal problem:** Two nodes that are outside each-other's range perform simultaneous transmission to a node that is within the range of each of them, hence, there is a packet collision.

Exposed terminal problem the node is within the range of a node that is transmitting, and it cannot transmit to any node.

**Error-prone shared broadcast channel**: In radio transmission, a node can listen to all traffic within its range. Therefore, when there is communication going on no other node should transmit, otherwise there would be interferences. Access to the physical medium should be granted only if there is no session going on. Nodes will often compete for the channel at the same time; therefore, there is high probability of collisions. The aim of a MAC protocol will be to minimize them, while maintaining fairness.   
**No central coordination** In ad hoc networks, there is no central point of coordination due to the mobility of the nodes. Therefore, the control of the access  
---------------------------------------------------------------------------------------------------**------------------------------------------------------------------------------  
Q17)**---------------------------------------------------------------------------------------------------**------------------------------------------------------------------------------  
Q18)Distributed Denial of Service (DDoS) Attacks**A Distributed Denial of Service (DDoS) attack is a malicious attempt to disrupt the normal traffic of a targeted server, service, or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic.

The key characteristic of a DDoS attack is that the attack traffic originates from multiple compromised computer systems (often referred to as a "botnet" or "zombie network") that are distributed across different geographical locations. This distribution makes it significantly harder to stop the attack compared to a single-source DoS attack.

**How it Works:  
1. Botnet Creation:** Attackers infect a large number of computers with malware, turning them into "bots" or "zombies" that can be controlled remotely. **2. Command and Control:** The attacker communicates with the botnet through a command and control (C&C) server, instructing the bots on when and how to launch the attack. **3. Attack Launch:** On command, all the bots in the botnet simultaneously send a massive volume of traffic (e.g., requests, packets) towards the target.

**Impact of a DDoS Attack:  
1. Resource Exhaustion:** The sheer volume of traffic overwhelms the target's resources, such as server bandwidth, processing power, or network capacity. **2. Service Disruption:** As a result, legitimate users are unable to access the targeted service or website, leading to service outages.  
**3. Financial Loss:** Businesses can suffer significant financial losses due to lost revenue, damage to reputation, and the cost of mitigating the attack.

**Types of DDoS Attacks:**DDoS attacks can target different layers of the network and can be categorized based on the type of traffic used:  
**Volume-based Attacks:** Aim to flood the target with a massive amount of traffic, consuming bandwidth (e.g., UDP floods, ICMP floods).  
**Protocol Attacks:** Exploit weaknesses in network protocols to consume server resources (e.g., SYN floods, fragmented packet attacks).     
**Application-layer Attacks:** Target specific vulnerabilities in applications to crash or overwhelm the server (e.g., HTTP floods, DNS query floods).

Mitigating DDoS attacks is challenging due to the distributed nature of the sources and the difficulty in distinguishing malicious traffic from legitimate traffic. Defense strategies often involve traffic filtering, rate limiting, anomaly detection, and using specialized DDoS mitigation services.  
---------------------------------------------------------------------------------------------------**------------------------------------------------------------------------------  
Q19)** **Hidden Station and Exposed Station Problems in WLAN**The root cause of both problems lies in the difference between a node's **transmission range** and its **sensing range**, and the fact that not all nodes can hear each other in a distributed wireless network.

**Hidden Station Problem:** The Hidden Station problem occurs when a station is "hidden" from another station with which it can potentially interfere at a shared receiver.

**Scenario:**  
Consider three stations:  
**Station A:** Wants to transmit to Station B.  
**Station B:** The receiver.  
**Station C:** Wants to transmit to Station B.

Assume the following conditions:  
Station A is within the transmission range of Station B.  
Station C is within the transmission range of Station B.  
**Crucially, Station A is *outside* the transmission range of Station C, and Station C is *outside* the transmission range of Station A.**

**Why it Happens:**  
Both Station A and Station C use carrier sensing to determine if the wireless channel is busy before transmitting.  
1. Station A wants to send data to B. It senses the channel. If C is not transmitting (or if A cannot hear C's transmission), A will find the channel idle.  
2. Simultaneously, Station C wants to send data to B. It senses the channel. If A is not transmitting (or if C cannot hear A's transmission), C will also find the channel idle.

Since A and C cannot hear each other, they are unaware of each other's potential transmissions to B. Believing the channel is free, both A and C might transmit their data packets to Station B at the same time.

**Consequences:**  
**Collisions at the Receiver:** The transmissions from A and C collide at Station B because B is within the range of both. This collision corrupts the data packets from both A and C.  
**Reduced Throughput:** The colliding packets are lost, requiring retransmissions, which reduces the effective data rate or throughput of the network.  
**Increased Latency:** Packets experience delays due to collisions and the need for retransmissions.

**Analogy:** Imagine two people in separate rooms trying to talk to a person in a third room. Neither of the first two people can hear each other, so they might both start talking at the same time, causing their voices to overlap and become unintelligible to the person in the third room.

**Exposed Station Problem:** The Exposed Station problem occurs when a station is prevented from transmitting, even though its transmission would not cause a collision at the intended receiver.

**Scenario:**  
Consider four stations:  
**Station A:** The sender.  
**Station B:** The receiver of A's transmission.  
**Station C:** Wants to transmit to Station D.  
**Station D:** The receiver of C's transmission.

Assume the following conditions:  
Station A is transmitting to Station B.  
Station C is within the transmission range of Station B (i.e., C can hear B's transmissions).  
**Crucially, Station C is *outside* the transmission range of Station A, and Station D is *outside* the transmission range of Station B.**

**Why it Happens:**  
1. Station A is transmitting to Station B.  
2. Station C wants to transmit to Station D.  
3. Station C senses the channel. Since C is within the range of B, C hears B's transmission (even though B is receiving data, acknowledgments, or control signals related to its communication with A).  
4. Based on carrier sensing, Station C detects that the channel is busy because it hears B.  
5. According to standard MAC protocols that rely heavily on carrier sensing (like CSMA/CA without additional mechanisms), Station C will defer its transmission to Station D, believing it might cause a collision.

**Consequences:**  
**Reduced Spatial Reuse:** Station C's transmission to Station D would *not* have interfered with A's transmission to B because D is outside B's range. By deferring, C unnecessarily leaves the channel idle in its vicinity, reducing the overall capacity and spatial reuse of the network.  
**Decreased Efficiency:** Bandwidth is wasted because potential simultaneous transmissions that would not collide are prevented.

**Analogy:** Imagine one person talking on a phone. Another person nearby hears them talking. This second person wants to have a separate conversation with someone far away, who cannot hear the first conversation at all. If the second person decides *not* to make their call just because they hear the first person talking, even though their call wouldn't interfere, that's analogous to the exposed terminal problem.

**Impact on WLAN Performance**Both the Hidden and Exposed Station problems negatively impact the performance of WLANs: **Hidden Terminals:** Directly lead to collisions, requiring retransmissions, and significantly reducing throughput and increasing delay. **Exposed Terminals:** Reduce the potential for parallel transmissions in different parts of the network, limiting spatial reuse and overall network capacity.  
---------------------------------------------------------------------------------------------------**------------------------------------------------------------------------------  
Q20)**