EC8702 – AD HOC AND WIRELESS SENSOR NETWORKS

ALL UNITS – 16 MARKS QUESTIONS WITH ANSWERS

Subject Code	**EC8702**
Branch	**B.E. ECE – 7th Semester**
Regulation	**2017 (Anna University)**
Covers PYQs	**2020, 2021, 2022, 2023, 2024**

■■■ HIGH PRIORITY	**Appeared 3+ times – Must Prepare**
■■ MEDIUM PRIORITY	**Appeared 2 times – Likely to appear**
■ LOW PRIORITY	**Appeared once – Prepare if time allows**

UNIT I: AD HOC NETWORKS – INTRODUCTION AND ROUTING PROTOCOLS

Q1. Discuss in detail about the various ISSUES AND CHALLENGES in designing a Routing Protocol for Ad hoc Wireless Networks. (13/16 Marks) ■■■

Definition:

A **routing protocol** in an Ad hoc network determines the path for data packets from source to destination without fixed infrastructure.

Key Issues and Challenges:

1. Dynamically Changing Topology: Nodes move freely causing frequent link changes. Routes become invalid quickly. Protocol must handle rapid topology updates efficiently.

2. Bandwidth Constraint: Wireless links have limited and shared bandwidth. Control packets for routing consume useful bandwidth. Protocol must minimize overhead.

3. Error-Prone Shared Broadcast Radio Channel: Wireless signals face interference, fading, hidden terminal problem. Packet loss is higher than wired networks.

4. Hidden Terminal Problem: Node A and C cannot hear each other but both send to B, causing collision. CSMA/CA with RTS/CTS partially solves this.

5. Energy Constrained Operation: Nodes run on batteries. Routing protocol must be energy-aware. Minimize transmissions to extend network lifetime.

6. Limited Physical Security: Ad hoc networks are more vulnerable to attacks (eavesdropping, spoofing, DoS). Secure routing is difficult without infrastructure.

7. Scalability: As number of nodes increases, routing table size, control overhead, and convergence time also increase. Protocol must scale well.

8. Uni-directional Links: Due to different transmission power levels, some links may be one-way. Most routing protocols assume bidirectional links.

9. QoS Support: Hard to guarantee Quality of Service (delay, bandwidth, reliability) due to dynamic topology and shared medium.

10. Interoperability: Multiple nodes from different vendors need to communicate. Standard protocols must be followed.

Design Goals of Routing Protocol:

- Distributed operation (no central controller)
- Loop-free routing
- Demand-based operation
- Security and authentication
- Minimal control overhead
- Support for unidirectional links
- Power-aware routing

■ **PYQ: Nov/Dec 2020, Nov/Dec 2022, Apr/May 2024**

Q2. Explain AODV Route Establishment and Route Maintenance with Diagram. Consider 'A' as Source Node. (15/16 Marks) ■■■

AODV – Ad hoc On-Demand Distance Vector Routing

AODV is a reactive (on-demand) protocol. Routes are created only when needed. It uses sequence numbers to avoid loops and stale routes.

Key Messages:

- **RREQ (Route Request):** Broadcast by source to find route
- **RREP (Route Reply):** Unicast reply from destination or intermediate node
- **RERR (Route Error):** Sent when link breaks to notify route failure
- **HELLO Messages:** Periodic broadcasts to detect neighbor nodes

Phase 1 – Route Discovery:

1. Source A generates RREQ with: Source IP, Source Seq No, Broadcast ID, Dest IP, Dest Seq No, Hop Count=0
2. RREQ is flooded via broadcast to all neighbors
3. Each intermediate node checks if route to destination exists. If yes, sends RREP back to source
4. If not, re-broadcasts RREQ (increments hop count) and records reverse route to A
5. Destination receives RREQ → sends unicast RREP back along reverse path
6. Each node receiving RREP records forward route to destination
7. Source A receives RREP → route established

Phase 2 – Route Maintenance:

- Nodes monitor active routes using HELLO messages
- If link breaks → node sends RERR upstream to source
- Source re-initiates route discovery (new RREQ)
- Sequence numbers ensure freshest route is selected
- Destination Sequence Number prevents loops and stale routes

RREQ Packet Format:

Field	Description
Type	RREQ = 1
Hop Count	No. of hops from source
RREQ ID	Unique ID (Source IP + Seq No)
Dest IP	Target destination address
Dest Seq No	Last known dest sequence no.
Source IP	Originator of RREQ

Source Seq No	Source sequence number

Network Example: A→B→D→F→I→H (source A wants route to H)

- A broadcasts RREQ → B,C receive it → forward to D,F → ... → H receives → RREP sent back A←B←D←F←I←H

■ **PYQ: ALL 5 years (2020–2024) – MOST IMPORTANT**

Q3. Explain DSDV Routing Protocol with Example and Routing Table Construction. (13 Marks) ■■■

DSDV – Destination Sequenced Distance Vector

DSDV is a table-driven (proactive) routing protocol based on Bellman-Ford algorithm. Every node maintains a routing table with routes to ALL destinations.

Routing Table Fields:

Field	Description
Destination	IP of destination node
Next Hop	Immediate next node on path
Metric	Number of hops to reach dest
Sequence Number	Even = valid, Odd = invalid/broken
Install Time	When route was installed
Stable Data	Time to settle/stabilize

Working Mechanism:

1. Each node broadcasts its routing table periodically
2. Neighbors update their tables using received info (Bellman-Ford)
3. Higher sequence number = more recent (fresher) route
4. Same seq no. → choose shorter metric (fewer hops)
5. When link breaks: node sets metric = infinity and seq no. = odd
6. Link repair: send new update with even seq no.

Types of Updates:

- **Full Dump:** Complete routing table sent periodically
- **Incremental Update:** Only changed entries sent (saves bandwidth)

Comparison: DSDV vs AODV

Parameter	DSDV (Proactive)	AODV (Reactive)
Route availability	Always available	On-demand only
Control overhead	High (periodic updates)	Low (only when needed)
Delay	Low (route pre-computed)	High (route discovery delay)
Best for	Small, stable networks	Large, mobile networks
Loop-free	Yes (seq numbers)	Yes (seq numbers)

■ **PYQ: Nov/Dec 2020, 2022 | AODV vs DSDV: Apr/May 2024**

UNIT II: SENSOR NETWORKS – INTRODUCTION AND ARCHITECTURES

Q4. Explain WSN Architecture – Single-Node and Network Architecture with Diagrams. (13 Marks) ■■■

What is WSN?

A **Wireless Sensor Network (WSN)** consists of spatially distributed autonomous sensor nodes that monitor physical/environmental conditions and cooperatively transmit data to a base station.

A. Single-Node Architecture (Hardware Components):

- **Sensor Unit:** Converts physical quantity (temp/pressure) to electrical signal. Contains sensors + ADC (Analog to Digital Converter).
- **Processing Unit:** Microcontroller/microprocessor + memory (Flash, RAM, ROM). Runs TinyOS or CONTIKI. Processes sensor data.
- **Communication Unit:** Transceiver (radio module) for wireless communication. Most power-hungry component in active mode.
- **Power Unit:** Battery (primary source). May include energy harvesting (solar, vibration). Determines node lifetime.
- **Mobilizer (Optional):** Allows physical movement of the node for targeted sensing.
- **Location Finding (Optional):** GPS or localization protocol to determine node position.

B. Network Architecture Scenarios:

- **Star Topology:** All sensor nodes directly communicate with base station. Simple but limited range.
- **Cluster Topology:** Nodes grouped into clusters. Cluster head aggregates data. Used in LEACH protocol.
- **Multi-hop Mesh:** Data relayed through intermediate nodes. Extends network coverage. Most common in WSN.
- **Hybrid Topology:** Combination of star and mesh. Flexible and robust.

Network Architecture Elements:

Sensor Nodes → Gateway/Sink Node → Internet/Satellite → End User

■ **PYQ: Nov/Dec 2021, Apr/May 2024**

Q5. Discuss Energy Consumption Model of a Sensor Node with Derivation and Diagram. (13 Marks) ■■■

Four Operational States of a Sensor Node:

State	Description	Typical Current
Transmit (TX)	Highest power consumption. Sending data packets.	~12–20 mA
Receive (RX)	Active listening. High power but less than TX.	~10–15 mA
Idle	Radio on but not transmitting/receiving.	~8–12 mA
Sleep	Radio off. Minimum power. Only timer/interrupt active.	~0.001–0.01 mA

Energy Consumption Formula:

For transmitting k-bit message over distance d:

$$E_{TX}(k,d) = E_{elec} \times k + E_{amp} \times k \times d^2$$

For receiving:

$$E_{RX}(k) = E_{elec} \times k$$

Where:

- E_{elec} = Energy dissipated per bit in transmit/receive circuit (~50 nJ/bit)
- E_{amp} = Amplifier energy (~100 pJ/bit/m²)
- d = Distance between nodes
- k = Number of bits
- d^2 = Free-space path loss model

Energy Saving Techniques:

- Duty cycling (sleep/wake scheduling) – S-MAC uses this
- Data aggregation at cluster heads – LEACH uses this
- Power-aware routing – choose energy-efficient paths
- Dynamic voltage scaling

• Energy harvesting (solar, piezoelectric)

■ **PYQ: Nov/Dec 2020, 2022, 2023 – Formula asked frequently!**

Q6. Explain Transceiver Design Considerations in WSN with Operational States. (13 Marks) ■■■

Transceiver: The radio module responsible for sending and receiving data wirelessly. It is the most energy-consuming component.

Design Considerations:

- **Energy Efficiency:** Must support multiple power modes. Transition between TX/RX/Idle/Sleep should be fast.
- **Operating Frequency:** 2.4 GHz (IEEE 802.15.4), 868 MHz, 915 MHz. Higher freq = more bandwidth, higher attenuation.
- **Modulation Scheme:** BPSK, QPSK, O-QPSK. Choose based on energy vs. data rate trade-off.
- **Data Rate:** 250 kbps (IEEE 802.15.4). Higher rate = shorter TX time = less energy.
- **Sensitivity:** Ability to detect weak signals. Better sensitivity = longer range.
- **Startup Time:** Time to switch from sleep to active. Lower startup time saves energy.
- **Dynamic Modulation Scaling:** Adjust modulation based on channel quality to save energy.
- **Noise Figure:** NF = SNR_in / SNR_out. Lower NF = better receiver performance.

Noise Figure Formula:

NF = 10 × log■■(SNR_input / SNR_output) dB

Lower NF → Better performance | NF is minimized when noise temperature is low

■ **PYQ: Nov/Dec 2020, 2022, 2023**

UNIT III: WSN NETWORKING CONCEPTS AND PROTOCOLS

Q7. Explain S-MAC Protocol with Neat Diagram. Explain Duty Cycle and Energy Savings. (13 Marks) ■■■

S-MAC (Sensor-MAC):

S-MAC is a **low duty cycle MAC protocol** designed specifically for WSN to reduce energy consumption by periodically sleeping.

Key Features:

- **Periodic Listen and Sleep:** Nodes alternate between active (listen) and sleep periods. Sleep = radio OFF → saves energy.
- **Synchronization:** Neighboring nodes synchronize their sleep/wake schedules using SYNC packets.
- **Duty Cycle:** Ratio of listen time to total frame time. Typical: 10%. Formula: DC = T_listen / (T_listen + T_sleep)
- **Virtual Clustering:** Nodes with same schedule form a virtual cluster. Boundary nodes follow two schedules.
- **Adaptive Listening:** Nodes briefly wake up after overhearing a transmission to receive data meant for them.
- **RTS/CTS/DATA/ACK:** Uses 4-way handshake to avoid collisions and hidden terminal problem.

S-MAC Frame Structure:

| SYNC | Listen Period | Sleep Period | ... repeat ...|

Duty Cycle Formula: DC = T_listen / (T_listen + T_sleep)

Limitations: Synchronization overhead, latency increases for multi-hop, not adaptive to traffic changes.

■ **PYQ: Apr/May 2024, Nov/Dec 2020**

Q8. Explain LEACH Protocol – Setup Phase and Steady-State Phase with Diagrams. (13 Marks) ■■■

LEACH – Low Energy Adaptive Clustering Hierarchy

LEACH is a **cluster-based routing protocol** for WSN that rotates cluster head (CH) role among nodes to balance energy consumption and extend network lifetime.

Phase 1 – Setup Phase:

1. Each node generates a random number between 0 and 1
2. If random number < T(n) → node becomes Cluster Head (CH)
3. CH advertises itself by broadcasting ADV message
4. Non-CH nodes join nearest CH and send JOIN-REQ message
5. CH creates TDMA schedule and broadcasts to cluster members
6. T(n) = p/(1-p × (r mod 1/p)) — where p=desired CH fraction, r=current round

Phase 2 – Steady-State Phase:

1. Each member node transmits in its assigned TDMA slot
2. CH aggregates data from all members
3. CH compresses and sends aggregated data to Base Station (BS)
4. Non-CH nodes sleep when not in their slot (saves energy)
5. After fixed time, new round begins → new CH elected

Key Advantages of LEACH:

- Rotates CH → balances energy load
- Data aggregation reduces transmissions to BS
- Nodes sleep during other slots → saves energy
- Self-organizing – no central controller needed

Network Lifetime Extension: LEACH can extend lifetime by 8x compared to direct transmission.

■ PYQ: Nov/Dec 2021, Apr/May 2024

Q9. Discuss IEEE 802.15.4 MAC Protocol – Super-frame Structure, GTS, Beacon Modes. (13 Marks) ■■■

IEEE 802.15.4:

IEEE 802.15.4 is the standard MAC and PHY layer protocol for **Low-Rate Wireless Personal Area Networks (LR-WPAN)** – the foundation for ZigBee.

Operating Frequencies:

Band	Frequency	Data Rate	Channels
ISM (global)	2.4 GHz	250 kbps	16 channels
ISM (Americas)	915 MHz	40 kbps	10 channels
Europe	868 MHz	20 kbps	1 channel

Super-frame Structure (Beacon-enabled mode):

| Beacon | CAP (Contention Access Period) | CFP (Contention-Free Period/GTS) | Inactive |

Components:

- **Beacon:** Sent by coordinator to synchronize devices and define super-frame boundaries
- **CAP:** CSMA/CA based access. Any device can transmit. Used for regular data.
- **CFP/GTS:** Guaranteed Time Slots. Dedicated slots for specific devices. No collisions.
- **Inactive Period:** Devices sleep to save energy. No transmissions.
- **BO (Beacon Order):** Controls beacon interval. Beacon Interval = aBaseSuperframeDuration × 2^BO
- **SO (Superframe Order):** Controls active portion. Active period = aBaseSuperframeDuration × 2^SO

Beacon vs Non-Beacon Mode:

Feature	Beacon-Enabled	Non-Beacon
Synchronization	Yes (coordinator sends beacons)	No

Power saving	High (sleep in inactive period)	Low
Access method	CSMA/CA + GTS	Unslotted CSMA/CA
Suitable for	Periodic, low-latency apps	Irregular traffic

■ **PYQ: Nov/Dec 2020, 2023 | Super-frame diagram often asked!**

Q10. Explain PAMAS – Power Aware Multi-Access Protocol with Signaling Protocol. (13 Marks) ■■■

PAMAS – Power Aware Multi-Access Protocol with Signaling:

PAMAS is a MAC protocol that reduces energy waste by **turning off nodes that are overhearing** irrelevant transmissions (called 'overhearing problem').

Two Key Problems PAMAS Solves:

1. **Idle Listening:** Radio ON but no data → wastes energy
2. **Overhearing:** Receiving packets not meant for you → wastes energy

PAMAS Architecture – Two Separate Channels:

- **Data Channel:** For actual data transmission (uses CSMA/CA)
- **Signaling Channel:** Separate channel for RTS/CTS/SNDR/RCVR control messages

PAMAS Signaling Packet Types:

- **SNDR (Sender):** Node broadcasts SNDR when it wants to transmit
- **RCVR (Receiver):** Receiver broadcasts RCVR to signal it is busy
- **RTS:** Request to Send – used with SNDR
- **CTS:** Clear to Send – used with RCVR

Node Sleep Decision Rules:

- Node turns OFF if neighbor is transmitting AND node has nothing to send
- Node turns OFF if another node is receiving AND current node has nothing to send
- Node stays awake if it has pending data
- Sleep duration = estimated transmission time of ongoing transfer

Advantage: Significantly reduces energy waste from overhearing.

Disadvantage: Separate signaling channel → hardware complexity.

■ **PYQ: Apr/May 2024, Nov/Dec 2023**

UNIT IV: SENSOR NETWORK SECURITY

Q11. Explain Network Security Attacks and Layer-wise Attacks in WSN with Solutions. (13 Marks) ■■■

Security Requirements in WSN:

- Data Confidentiality – Only authorized parties read data
- Data Integrity – Data not modified in transit
- Authentication – Verify identity of communicating nodes
- Availability – Network services available when needed
- Freshness – Data is recent (no replay attacks)

Layer-wise Attacks in WSN:

Layer	Attack	Effect	Solution
Physical	Jamming	Disrupts radio communication	Spread spectrum, FHSS
Physical	Tampering	Physical node destruction/capture	Tamper-proof casing
Data Link	Collision	Packet corruption	Error correction codes

Data Link	Exhaustion	Repeated RTS → battery drain	Rate limiting
Network	Sinkhole	Attract traffic via false routing	Authenticated routing
Network	Wormhole	Tunnel packets to far away	Packet leashes
Network	Sybil	One node fakes multiple identities	Identity authentication
Network	Black Hole	Drop all packets selectively	Multipath routing
Transport	Flooding	SYN flood exhausts resources	Puzzles/authentication
Transport	De-sync	Disrupt connection states	Sequence verification

■ **PYQ: Apr/May 2024, Nov/Dec 2021**

Q12. As an Attacker, Write Steps for DoS Attack and Explain Effects in WSN. (13/15 Marks) ■■■

DoS (Denial of Service) Attack – Step by Step:

- Step 1 – Reconnaissance: Attacker scans WSN to identify nodes, topology, protocols used
- Step 2 – Select Attack Vector: Choose physical jamming / packet flooding / resource exhaustion
- Step 3 – Deploy Attack Node: Introduce malicious node into network vicinity
- Step 4 – Jamming: Broadcast high-power noise on same frequency to block communication (Physical DoS)
- Step 5 – Flooding: Send massive number of packets (RREQ flood) to overwhelm routing tables
- Step 6 – Sinkhole/Black Hole: Attacker node advertises shortest/best route to attract traffic
- Step 7 – Selective Forwarding: Drop packets selectively to cause data loss while appearing normal
- Step 8 – Resource Exhaustion: Send continuous requests to drain battery of victim nodes

Effects of DoS Attack on WSN:

- Network partitioning – nodes cannot communicate with base station
- Battery drain – nodes die prematurely → network lifetime reduced
- Data loss – critical sensor readings lost (dangerous in medical WSN)
- Routing disruption – false routing information spreads
- False alarms – attacker injects false sensor readings

Mitigation Strategies:

- Spread spectrum (FHSS/DSSS) against jamming
- Rate limiting against flooding
- Authentication to prevent malicious node injection
- Multipath routing to bypass black hole nodes
- Intrusion detection systems (IDS) in WSN

■ **PYQ: Nov/Dec 2020, Apr/May 2024 – VERY IMPORTANT!**

Q13. Explain SPINS – Security Protocol for Sensor Networks. How does SPIN provide Authenticated Broadcast? (13 Marks) ■■■

SPINS – Security Protocol for Sensor Networks:

SPINS provides a suite of security protocols designed for **resource-constrained sensor nodes**. It has two main components:

1. SNEP (Sensor Network Encryption Protocol):

- Provides: Data Confidentiality, Data Integrity, Data Freshness
- Uses: AES block cipher in counter mode
- Counter mode avoids IV synchronization issues
- Uses MAC (Message Authentication Code) for integrity
- Low overhead – adds only 8 bytes per message

2. μTESLA (Micro Timed Efficient Streaming Loss-tolerant Authentication):

- Provides: Authenticated Broadcast to multiple nodes

- Based on one-way hash chain of keys
- Keys are revealed after time delay (bootstrapping trust)
- Sender buffers messages; receiver buffers too
- Solves the problem: broadcast authentication without asymmetric cryptography

Three Main Design Issues of SPINS:

1. **Resource Constraints:** Low memory (4KB RAM), low CPU, limited battery
2. **Bootstrapping Trust:** How to establish shared keys initially? Use pre-loaded keys.
3. **Broadcast Authentication:** Authenticating broadcast without public key crypto

μTESLA Authenticated Broadcast Process:

1. BS sets up key chain: K0 ← K1 ← K2 ← ... (using one-way hash)
2. Each message authenticated with current key Ki
3. Key Ki revealed after delay → nodes verify broadcast authenticity

■ PYQ: Nov/Dec 2020, Nov/Dec 2022

UNIT V: SENSOR NETWORK PLATFORMS AND TOOLS

Q14. Explain TinyOS Features, Limitations, and How it Supports Berkeley Mote Hardware. (13 Marks) ■■■

Berkeley Motes:

Berkeley Motes are miniaturized sensor node platforms developed at UC Berkeley. Common motes: MICA, MICA2, MICAz, TelosB.

TinyOS – Overview:

TinyOS is an **open-source, event-driven operating system** designed for wireless embedded sensor networks. Written in nesC language.

Key Features of TinyOS:

- **Component-based Architecture:** Programs built from components. Each component has Interface, Module, Configuration.
- **Event-driven Programming:** No threads. Uses events and tasks. Efficient for low-power hardware.
- **Task Scheduler:** Non-preemptive FIFO scheduler. Tasks run to completion. Simple, predictable.
- **Split-phase Operations:** Long operations (radio TX) split into start + completion event. Non-blocking.
- **Small Footprint:** Only ~400 bytes RAM needed. Fits on 4KB RAM nodes.
- **Active Messages:** Message-passing model for communication between components.
- **TinyDB:** SQL-like query language to extract sensor data as database queries.

nesC Language:

nesC (network embedded systems C) is a C dialect for TinyOS. Key concepts:

- Interface – defines set of functions (commands + events)
- Module – implements interface functions
- Configuration – wires components together
- Component wiring enables modular, reusable code

TinyOS Limitations:

- No dynamic memory allocation (no malloc/free)
- Non-preemptive: one task blocks all others
- Complex programming model for beginners
- Limited debugging support
- No virtual memory or process isolation

TinyOS on Berkeley Mote:

TinyOS hardware abstraction layer (HAL) maps software components directly to MICA mote hardware: ADC, radio (CC1000/CC2420), UART, timer.

■ **PYQ: Apr/May 2024, Nov/Dec 2021, 2022**

Q15. Illustrate NS-2 for Sensor Network Design and Evaluation. Write Sample NS-2 Script. (13 Marks) ■■■

NS-2 (Network Simulator 2):

NS-2 is an open-source, discrete event network simulator widely used for research and evaluation of ad hoc and sensor network protocols.

Key Components of NS-2:

- **Programming Language:** OTcl (Object Tcl) for scripting + C++ for core simulation
- **Graphical Interface:** NAM (Network Animator) – visualizes packet transmission
- **Animation Window:** NAM trace window shows node positions and packet flows
- **Trace Files:** Text files (.tr) recording all simulation events
- **Protocol Support:** AODV, DSDV, DSR, IEEE 802.11, 802.15.4
- **Mobility Model:** Random Waypoint, Gauss-Markov models

Sample NS-2 Simulation Script (Tcl):

```
set ns [new Simulator]
set nf [open out.nam w]
$ns namtrace-all $nf
set tr [open out.tr w]
$ns trace-all $tr

# Create 3 nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]

# Create duplex links
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
$ns duplex-link $n1 $n2 1Mb 10ms DropTail

# Create UDP agent and attach to n0
set udp [new Agent/UDP]
$ns attach-agent $n0 $udp
set null [new Agent/Null]
$ns attach-agent $n2 $null
$ns connect $udp $null

# Create CBR traffic
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
$cbr set rate_ 1Mb

# Schedule events
$ns at 0.5 "$cbr start"
$ns at 4.5 "$cbr stop"
$ns at 5.0 "finish"
proc finish {} { global ns nf; $ns flush-trace; close $nf; exec nam out.nam &; exit 0 }
$ns run
```

NS-2 Simulation Process:

Write Tcl Script → Run NS-2 → Generate .tr trace file → Analyze with AWK → Visualize in NAM

■ **PYQ: Nov/Dec 2020, 2023 – Script writing often asked!**

Q16. How are TinyOS, nesC, and CONTIKI OS Designed for Sensor Nodes? Discuss Design Challenges. (15 Marks) ■■■

Comparison of OS Platforms:

Feature	TinyOS	CONTIKI OS
Language	nesC (C dialect)	C
Model	Event-driven	Event-driven + multi-threading
Threading	No (tasks only)	Protothreads
Memory	~400 bytes RAM	~2KB RAM
Dynamic alloc.	No	Yes
Best for	Tiny motes (MICA)	Slightly bigger nodes
Toolchain	TOSSIM simulator	COOJA simulator

CONTIKI OS Key Features:

- Supports protothreads (lightweight threads without stack overhead)
- Has uIP (micro IP stack) for Internet connectivity
- Supports IPv6 and 6LoWPAN
- Used with COOJA simulator
- Supports coffee filesystem for flash storage

Design Challenges for Sensor OS:

- **Memory Constraint:** Nodes have 4–8KB RAM. OS must have minimal footprint.
- **Concurrency:** Multiple events (sensor, radio) need handling. No preemption → tricky.
- **Real-time Requirements:** Timing-critical operations (TDMA slots, duty cycles).
- **Energy Management:** OS must aggressively power-gate unused components.
- **Robustness:** Nodes must recover from errors without human intervention.
- **Reprogrammability:** Update software over-the-air (OTA) without physical access.

■ **PYQ: Nov/Dec 2022, Nov/Dec 2021**

QUICK REVISION SUMMARY – MUST-PREPARE TOPICS

Unit	Must-Prepare Topics	Priority
Unit I	AODV (with diagram), DSDV routing table, Routing protocol design challenges, AODV vs DSDV	■■■
Unit II	WSN Architecture (single-node + network), Energy consumption derivation, Transceiver states	■■■
Unit III	S-MAC protocol, LEACH (setup + steady-state), IEEE 802.15.4 super-frame, PAMAS	■■■
Unit IV	DoS attack steps + effects, Layer-wise attacks, SPINS protocol, Black hole + flooding attacks	■■■
Unit V	TinyOS + nesC features, NS-2 simulation script, CONTIKI OS design, COOJA/TOSSIM simulators	■■■

Prepared based on PYQs: Nov/Dec 2020 | 2021 | 2022 | 2023 | Apr/May 2024