EC8702 AD HOC AND WIRELESS SENSOR NETWORKS

UNIT I AD HOC NETWORKS – INTRODUCTION AND ROUTING PROTOCOLS

Elements of Ad hoc Wireless Networks

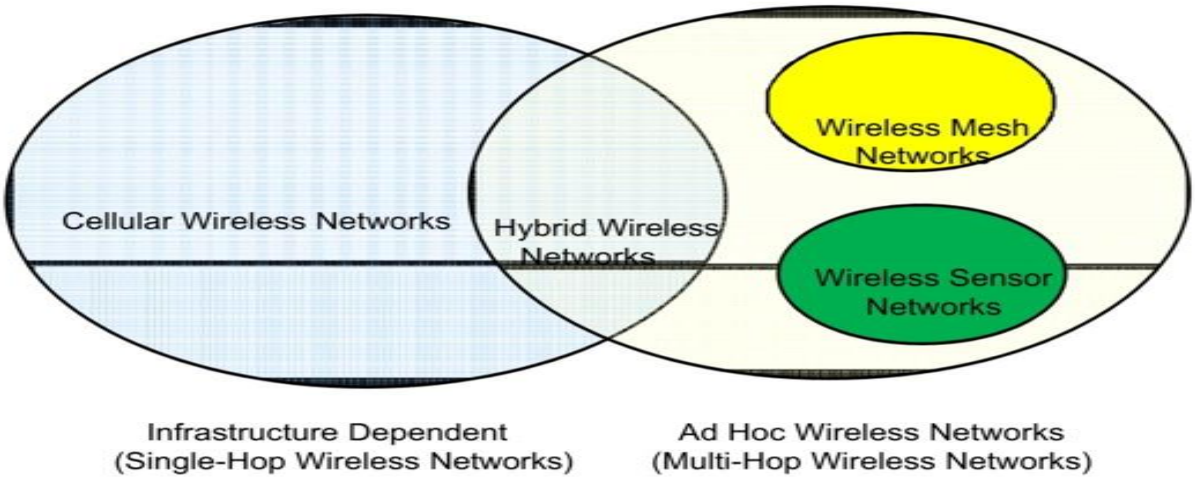


Figure 1.1. Cellular and ad hoc wireless networks.

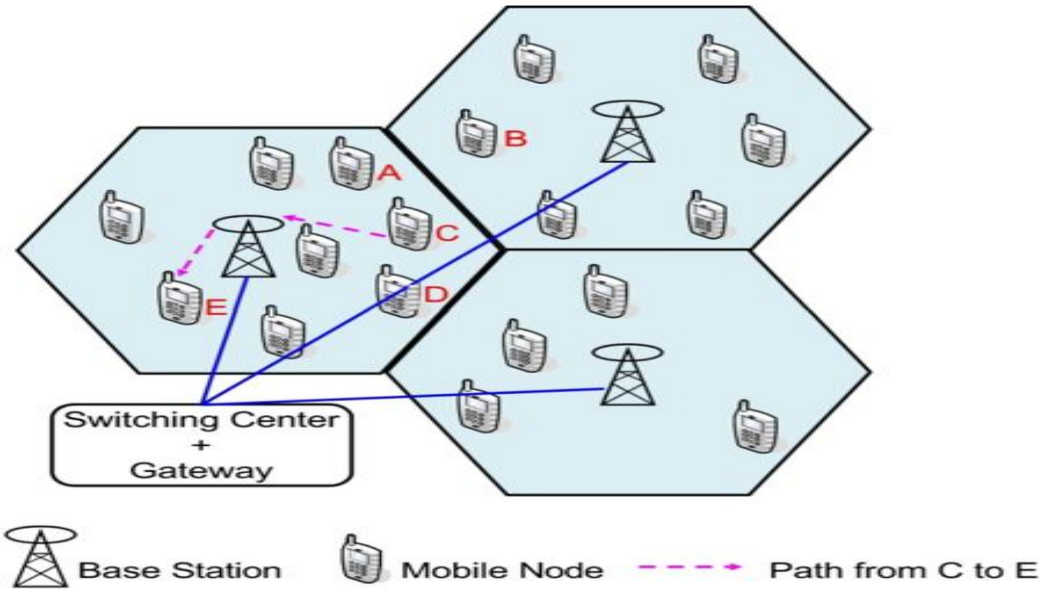
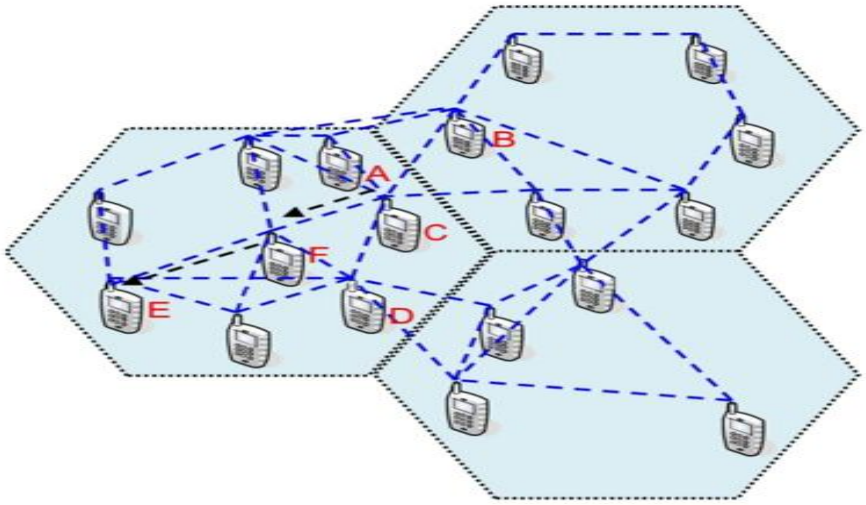


Figure 1.2. A cellular networks.



Mobile Node ------ Wireless Link ----→ Path from C to E

Figure 1.3. An ad hoc wireless networks

Table 1.1 Differences between cellular networks and ad hoc wireless networks

Cellular Networks	Ad Hoc Wireless Networks
Fixed infrastructure-based	Infrastructure-less
Single-hop wireless links	Multi-hop wireless links
Guaranteed bandwidth (designed for voice traffic)	Shared radio channel (more suitable for best-effort data traffic)
Centralized routing	Distributed routing
Circuit-switched (evolving toward packet switching)	Packet-switched (evolving toward emulation of circuit switching)
Seamless connectivity (low call drops during handoffs)	Frequency path break due to mobility
High cost and time of deployment	Quick and cost-effective deployment
Reuse of frequency spectrum through geographical channel reuse	Dynamic frequency reuse based on carrier sense mechanism

7

Table 1.1 Differences between cellular networks and ad hoc wireless networks (cont.)

Easier to achieve time synchronization	Time synchronization is difficult and consumes bandwidth
Easier to employ bandwidth reservation	Bandwidth reservation requires complex medium access control protocols
Application domains include mainly civilian and commercial sector	Application domains include battlefields, emergency search and rescue operation, and collaborative computing
High cost of network maintenance (backup power source, staffing, etc.)	Self-organization and maintenance properties are built into the network
Mobile hosts are of relatively low complexity	Mobile hosts require more intelligence (should have a transceiver as well as routing/switching capacity)
Major goals of routing and call admission are to maximize the call acceptance ratio and minimize the call drop ratio	Man aim of routing is to find paths with minimum overhead and also quick reconfiguration of broken paths
Widely deployed and currently in the third generation	Several issues are to be addressed for successful commercial deployment even though widespread use exists in defense

ISSUES IN AD HOC WIRELESS NETWORKS

The major issues that affect the design, deployment, & performance of an ad hoc wireless network system are :

- Medium Access Scheme.
- Transport Layer Protocol.
- Routing.
- Multicasting.
- Energy Management.
- Self-Organisation.
- Security.
- Addressing & Service discovery.
- Deployment considerations.
- Scalability.
- Pricing Scheme.
- Quality of Service Provisioning

1. Medium Access Scheme **

The primary responsibility of a Medium Access Control (MAC) protocol in adhoc wireless networks is the distributed arbitration for the shared channel for transmission of packets. **The major issues to be considered in designing a MAC protocol for adhoc wireless networks are as follows:**

1. Distributed Operation:

The ad hoc wireless networks need to operate in environments where no centralized coordination is possible.

The MAC protocol design should be fully distributed involving minimum control overhead.

2. Synchronization:

The MAC protocol design should take into account the requirement of time synchronization.

Synchronization is mandatory for TDMA-based systems for management of transmission and reception slots.

3. Hidden Terminals:

Hidden terminals are nodes that are hidden(or not reachable) from the sender of a data transmission session, but are reachable to the receiver of the session.

4. Exposed terminals:

Exposed terminals, the nodes that are in the transmission range of the sender of an on-going session, are prevented from making a transmission.

5. Throughput:

The MAC protocol employed in adhoc wireless networks should attempt to maximize the throughput of the system.

The important considerations for throughput enhancement are

- Minimizing the occurrence of collisions.
- Maximizing channel utilization and
- Minimizing control overhead.

6. Access delay:

The average delay that any packet experiences to get transmitted.

The MAC protocol should attempt to minimize the delay.

7. Fairness:

Fairness refers to the ability of the MAC protocol to provide an equal share or weighted share of the bandwidth to all competing nodes.

Fairness can be either node-based or flow-based.

8. Real-time Traffic support:

In a contention-based channel access environment, without any central coordination, with limited bandwidth, and with location-dependent contention,

supporting time- sensitive traffic such as voice, video, and real-time data requires explicit support from the MAC protocol.

9. Resource reservation:

The provisioning of QoS defined by parameters such as bandwidth, delay, and jitter requires reservation of resources such as bandwidth, buffer space, and processing power.

10. Ability to measure resource availability:

In order to handle the resources such as bandwidth efficiently and perform call admission control based on their availability, the MAC protocol should be able to provide an estimation of resource availability at every node.

This can also be used for making congestion control decisions.

11. Capability for power control:

The transmission power control reduces the energy consumption at the nodes, causes a decrease in interference at neighboring nodes, and increases frequency reuse.

12. Adaptive rate control:

This refers to the variation in the data bit rate achieved over a channel.

A MAC protocol that has adaptive rate control can make use of a high data rate when the sender and receiver are nearby & adaptively reduce the data rate as they move away from each other.

13. Use of directional antennas:

This has many advantages that include

- Increased spectrum reuse.
- Reduction in interference and
- Reduced power consumption.

2. Routing **

The responsibilities of a routing protocol include exchanging the route information; finding a feasible path to a destination. **The major challenges that a routing protocol faces are as follows:**

1. Mobility:

The Mobility of nodes results in frequent path breaks, packet collisions, transient loops, stale routing information, and difficulty in resource reservation.

2. Bandwidth constraint:

Since the channel is shared by all nodes in the broadcast region, the bandwidth available per wireless link depends on the number of nodes & traffic they handle.

3. Error-prone and shared channel:

The Bit Error Rate (BER) in a wireless channel is very high [10^{-5} to 10^{-3}] compared to that in its wired counterparts [10^{-12} to 10^{-9}].

Consideration of the state of the wireless link, signal-to-noise ratio, and path loss for routing in ad hoc wireless networks can improve the efficiency of the routing protocol.

4. Location-dependent contention:

The load on the wireless channel varies with the number of nodes present in a given geographical region.

This makes the contention for the channel high when the number of nodes increases.

The high contention for the channel results in a high number of collisions & a subsequent wastage of bandwidth.

5. Other resource constraints:

The constraints on resources such as computing power, battery power, and buffer storage also limit the capability of a routing protocol.

The major requirements of a routing protocol in adhoc wireless networks are the following.

1. Minimum route acquisition delay:

The route acquisition delay for a node that does not have a route to a particular destination node should be as minimal as possible.

The delay may vary with the size of the network and the network load.

2. Quick route reconfiguration:

The unpredictable changes in the topology of the network require that the routing protocol be able to quickly perform route reconfiguration in order to handle path breaks and subsequent packet losses.

3. Loop-free routing:

This is a fundamental requirement to avoid unnecessary wastage of network bandwidth

In adhoc wireless networks, due to the random movement of nodes, transient loops may form in the route thus established.

A routing protocol should detect such transient routing loops & take corrective actions.

4. Distributed routing approach:

An adhoc wireless network is a fully distributed wireless network & the use of centralized routing approaches in such a network may consume a large amount of bandwidth.

5. Minimum control overhead:

The control packets exchanged for finding a new route, and maintaining existing routes should be kept as minimal as possible.

6. Scalability:

Scalability is the ability of the routing protocol to scale well in a network with a large number of nodes.

This requires minimization of control overhead & adaptation of the routing protocol to the network size.

7. Provisioning of QoS:

The routing protocol should be able to provide a certain level of QoS as demanded by the nodes or the category of calls.

The QoS parameters can be bandwidth, delay, jitter, packet delivery ratio, & throughput.

8. Support for time-sensitive traffic:

Tactical communications & similar applications require support for time-sensitive traffic.

The routing protocol should be able to support both hard real-time & soft real-time traffic.

9. Security and privacy:

The routing protocol in adhoc wireless networks must be resilient to threats and vulnerabilities.

It must have inbuilt capability to avoid resource consumption, denial-of-service, impersonation, and similar attacks possible against an ad hoc wireless network.

3. Multicasting

It plays important role in emergency search & rescue operations & in military communication. Use of single- link connectivity among the nodes in a multicast group results in a tree-shaped multicast routing topology. Such a tree-shaped topology provides high multicast efficiency, with low packet delivery ratio due to the frequency tree breaks. The major issues in designing multicast routing protocols are as follows:

1. **Robustness:**

• The multicast routing protocol must be able to recover & reconfigure quickly from potential mobility-induced link breaks thus making it suitable for use in high dynamic environments.

2. **Efficiency:**

• A multicast protocol should make a minimum number of transmissions to deliver a data packet to all the group members.

3. **Control overhead:**

• The scarce bandwidth availability in ad hoc wireless networks demands minimal control overhead for the multicast session.

4. **Quality of Service:**

• QoS support is essential in multicast routing because, in most cases, the data transferred in a multicast session is time-sensitive.

5. **Efficient group management:**

• Group management refers to the process of accepting multicast session members and maintaining the connectivity among them until the session expires.

6. **Scalability:**

• The multicast routing protocol should be able to scale for a network with a large number of node

7. **Security:**

• Authentication of session members and prevention of non-members from gaining unauthorized information play a major role in military communications.

4. Transport Layer Protocol

The main objectives of the transport layer protocols include:

- Setting up & maintaining end-to-end connections,
- Reliable end-to-end delivery of packets,
- Flow control &
- Congestion control.

Examples of some transport layers protocols are,

a. UDP (User Datagram Protocol) :

- It is an unreliable connectionless transport layer protocol.
- It neither performs flow control & congestion control.
- It do not take into account the current network status such as congestion at the intermediate links, the rate of collision, or other similar factors affecting the network throughput.

b. TCP (Transmission Control Protocol):

- It is a reliable connection-oriented transport layer protocol.
- It performs flow control & congestion control.
- Here performance degradation arises due to frequent path breaks, presence of stale routing information, high channel error rate, and frequent network partitions.

5. Pricing Scheme

• Assume that an optimal route from node A to node B passes through node C, & node C is not powered on.

• Then node A will have to set up a costlier & non-optimal route to B.

• The non-optimal path consumes more resources & affects the throughput of the system.
• As the intermediate nodes in a path that relay the data packets expend their resources such as battery charge & computing power, they should be properly compensated.
• Hence, pricing schemes that incorporate service compensation or service reimbursement are required.

6. Quality of Service Provisioning (QoS)

QoS is the performance level of services offered by a service provider or a network to the user.
QoS provisioning often requires ,

- Negotiation between host & the network.
- Resource reservation schemes.
- Priority scheduling &
- Call admission control.

QoS parameters :

Applications	Corresponding QoS parameter
1. Multimedia application	1. Bandwidth & Delay.
2. Military application	2. Security & Reliability.
3. Defense application	3. Finding trustworthy intermediate hosts & routing
4. Emergency search and rescue operations	4.Availability.
5. Hybrid wireless network	5. Maximum available link life, delay, bandwidth & channel utilization.
6. Communication among the nodes in a sensor network	6. Minimum energy consumption, battery life & energy conservation

QoS-aware routing :

i. Finding the path is the first step toward a QoS-aware routing protocol.

ii. The parameters that can be considered for routing decisions are,

- Network throughput.
- Packet delivery ratio.
- Reliability.
- Delay.
- Delay jitter.
- Packet loss rate.
- Bit error rate.
- Path loss.

QoS framework :

I. A framework for QoS is a complete system that attempts to provide the promised services to each user or application.

II. The key component of QoS framework is a QoS service model which defines the way user requirements are served.

7. Self-Organization

• One very important property that an ad hoc wireless network should exhibit is organizing & maintaining the network by itself.

• The major activities that an ad hoc wireless network is required to perform for self-organization are,

- Neighbour discovery.
- Topology organization &
- Topology reorganization (updating topology information)

8. Security

1) Security is an important issue in ad hoc wireless network as the information can be hacked.

2) Attacks against network are of 2 types :

I. Passive attack → Made by malicious node to obtain information transacted in the network without disrupting the operation.

II. Active attack → They disrupt the operation of network. Further active attacks are of 2 types :

o External attack: The active attacks that are executed by nodes outside the network.

o Internal attack: The active attacks that are performed by nodes belonging to the same network.

3) The major security threats that exist in ad hoc wireless networks are as follows :
Denial of service – The attack affected by making the network resource unavailable for service to other nodes, either by consuming the bandwidth or by overloading the system.
Resource consumption – The scarce availability of resources in ad hoc wireless network makes it an easy target for internal attacks, particularly aiming at consuming resources available in the network.

The major types of resource consumption attacks are,
Energy depletion :
- Highly constrained by the energy source
- Aimed at depleting the battery power of critical nodes.

Buffer overflow :
- Carried out either by filling the routing table with unwanted routing entries or by consuming the data packet buffer space with unwanted data.
- Lead to a large number of data packets being dropped, leading to the loss of critical information.

Host impersonation – A compromised internal node can act as another node and respond with appropriate control packets to create wrong route entries, and can terminate the traffic meant for the intended destination node.
Information disclosure – A compromised node can act as an informer by deliberate disclosure of confidential information to unauthorized nodes.
Interference – A common attack in defense applications to jam the wireless communication by creating a wide spectrum noise.

9. Energy Management

Energy management is defined as the process of managing the sources & consumers of energy in a node or in the network for enhancing the lifetime of a network.
Features of energy management are :
→Shaping the energy discharge pattern of a node's battery to enhance battery life.
→Finding routes that consumes minimum energy.
→Using distributed scheduling schemes to improve battery life.
→Handling the processor & interface devices to minimize power consumption.
Energy management can be classified into the following categories :
a. Transmission power management:

The power consumed by the Radio Frequency (RF) module of a mobile node is determined by several factors such as

- The state of operation.
- The transmission power and
- The technology used for the RF circuitry.

The state of operation refers to transmit, receive, and sleep modes of the operation.
The transmission power is determined by

- Reach ability requirement of the network.
- Routing protocol and
- MAC protocol employed.

b. Battery energy management:

The battery management is aimed at extending the battery life of a node by taking advantage of its chemical properties, discharge patterns, and by the selection of a battery from a set of batteries that is available for redundancy.

c. Processor power management:

- The clock speed and the number of instructions executed per unit time are some of the processor parameters that affect power consumption.
- The CPU can be put into different power saving modes during low processing load conditions.
- The CPU power can be completely turned off if the machines is idle for a long time. In such a cases, interrupts can be used to turn on the CPU upon detection of user interaction or other events.

d. Devices power management:

- Intelligent device management can reduce power consumption of a mobile node significantly.
- This can be done by the operating system(OS) by selectively powering down interface devices that are not used or by putting devices into different power saving modes, depending on their usage.

10. Scalability

- Scalability is the ability of the routing protocol to scale well in a network with a large number of nodes.
- It requires minimization of control overhead & adaptation of the routing protocol to the network size.

11. Deployment Considerations

The deployment of a commercial ad hoc wireless network has the following benefits when compared to wired networks

a) Low cost of deployment:

- The use of multi-hop wireless relaying eliminates the requirement of cables & maintenance in deployment of communication infrastructure.
- The cost involved is much lower than that of wired networks.

b) Incremental deployment:

- Deployment can be performed incrementally over geographical regions of the city.
- The deployed part of the network starts functioning immediately after the minimum configuration is done.

c) Short deployment time:

Compared to wired networks, the deployment time is considerably less due to the absence of any wired links.

d) Reconfigurability:

The cost involved in reconfiguring a wired network covering a Metropolitan Area Network(MAN) is very high compared to that of an ad hoc wireless network covering the same service area.

12. Addressing and service discovery

- Addressing & service discovery assume significance in ad hoc wireless network due to the absence of any centralised coordinator.
- An address that is globally unique in the connected part of the ad hoc wireless network is required for a node in order to participate in communication.
- Auto-configuration of addresses is required to allocate non-duplicate addresses to the nodes.

AD HOC WIRELESS INTERNET

- Ad hoc wireless internet extends the services of the internet to the end users over an ad hoc wireless network.

Some of the applications of ad hoc wireless internet are:

- Wireless mesh network.
- Provisioning of temporary internet services to major conference venues.
- Sports venues.
- Temporary military settlements.
- Battlefields &
- Broadband internet services in rural regions.

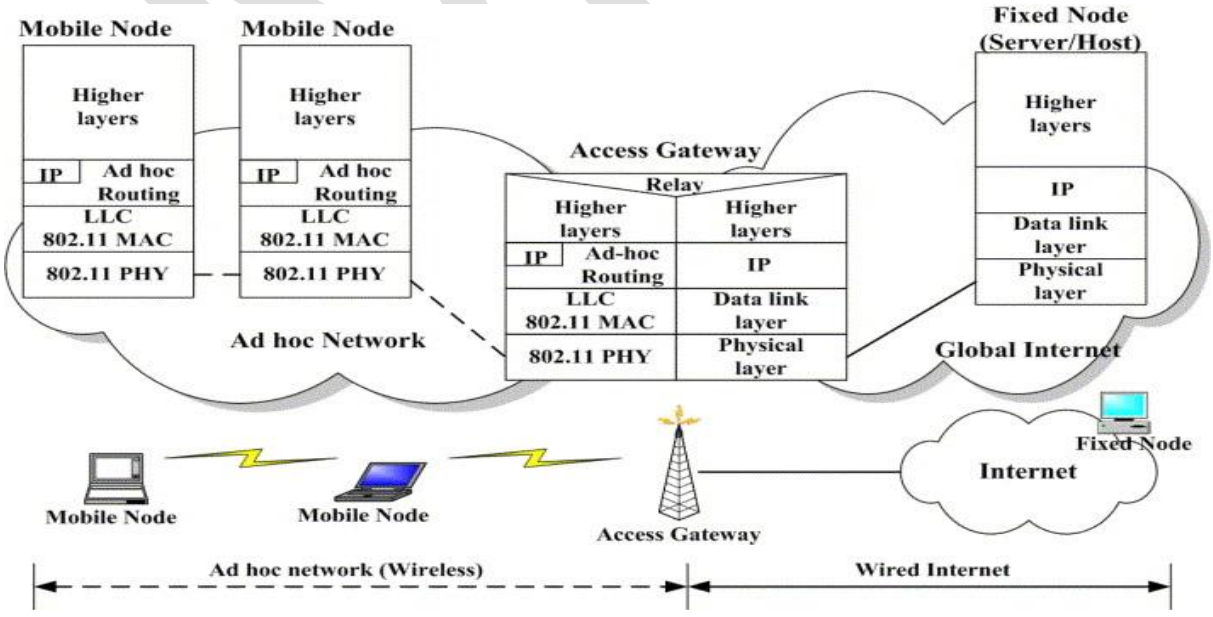
The major issues to be considered for a successful ad hoc wireless internet are the following:

Gateway:

- They are the entry points to the wired internet.
- Generally owned & operated by a service provider.

They perform following tasks,

- Keeping track of end users.
- Bandwidth management.
- Load balancing.
- Traffic shaping.
- Packet filtering.
- Width fairness &
- Address, service & location discovery.



Address mobility:
This problem is worse here as the nodes operate over multiple wireless hops.
Solution such as Mobile IP can provide temporary alternative.

Routing:

- It is a major problem in ad hoc wireless internet, due to dynamic topological changes, the presence of gateways, multi-hop relaying, & the hybrid character of the network.
- Possible solution is to use separate routing protocol for the wireless part of ad hoc wireless internet.

Transport layer protocol:

Several factors are to be considered here, the major one being the state maintenance overhead at the gateway nodes.

Load balancing:

They are essential to distribute the load so as to avoid the situation where the gateway nodes become bottleneck nodes.

Pricing / Billing:

Since internet bandwidth is expensive, it becomes very important to introduce pricing/billing strategies for the ad hoc wireless internet.

Provisioning of security:

Security is a prime concern since the end users can utilize the ad hoc wireless internet infrastructure to make e-commerce transaction.

QoS support:

With the widespread use of voice over IP(VOIP) & growing multimedia applications over the internet, provisioning of QoS support in the ad hoc wireless internet becomes a very important issue.

Service, address & location discovery:

- Service discovery refers to the activity of discovering or identifying the party which provides service or resource.
- Address discovery refers to the services such as those provided by Address Resolution Protocol (ARP) or Domain Name Service (DNS) operating within the wireless domain.
- Location discovery refers to different activities such as detecting the location of a particular mobile node in the network or detecting the geographical location of nodes.

APPLICATIONS OF AD HOC WIRELESS NETWORKS

1. Military Application

- Adhoc wireless networks can be very useful in establishing communication among a group of soldiers for tactical operations.
- Setting up of a fixed infrastructure for communication among group of soldiers in enemy territories or in inhospitable terrains may not be possible.
- In such a case, adhoc wireless networks provide required communication mechanism quickly.
- The primary nature of the communication required in a military environment enforces certain important requirements on adhoc wireless networks namely, Reliability, Efficiency, Secure communication & Support for multicast routing.

2. Collaborative & Distributed computing

- Adhoc wireless network helps in collaborative computing, by establishing temporary communication infrastructure for quick communication with minimal configuration among a group of people in a conference.
- In distributed file sharing application reliability is of high importance which would be provided by adhoc network.
- Other applications such as streaming of multimedia objects among participating nodes in ad hoc wireless networks require support for soft real-time communication
- Devices used for such applications could typically be laptops with add-on wireless interface cards, enhanced personal digital assistants (PDAs) or mobile devices with high processing power

3. Emergency Operations

- Ad hoc wireless networks are very useful in emergency operations such as search and rescue, crowd control and commando operations
- The major factors that favour ad hoc wireless networks for such tasks are self-configuration of the system with minimal overhead, independent of fixed or centralised infrastructure, the freedom and flexibility of mobility, and unavailability of conventional communication infrastructure.
- In environments, where the conventional infrastructure based communication facilities are destroyed due to a war or due to natural calamities, immediate deployment of adhoc wireless networks would be a good solution for co-ordinating rescue activities.

- They require minimum initial network configuration with very little or no delay

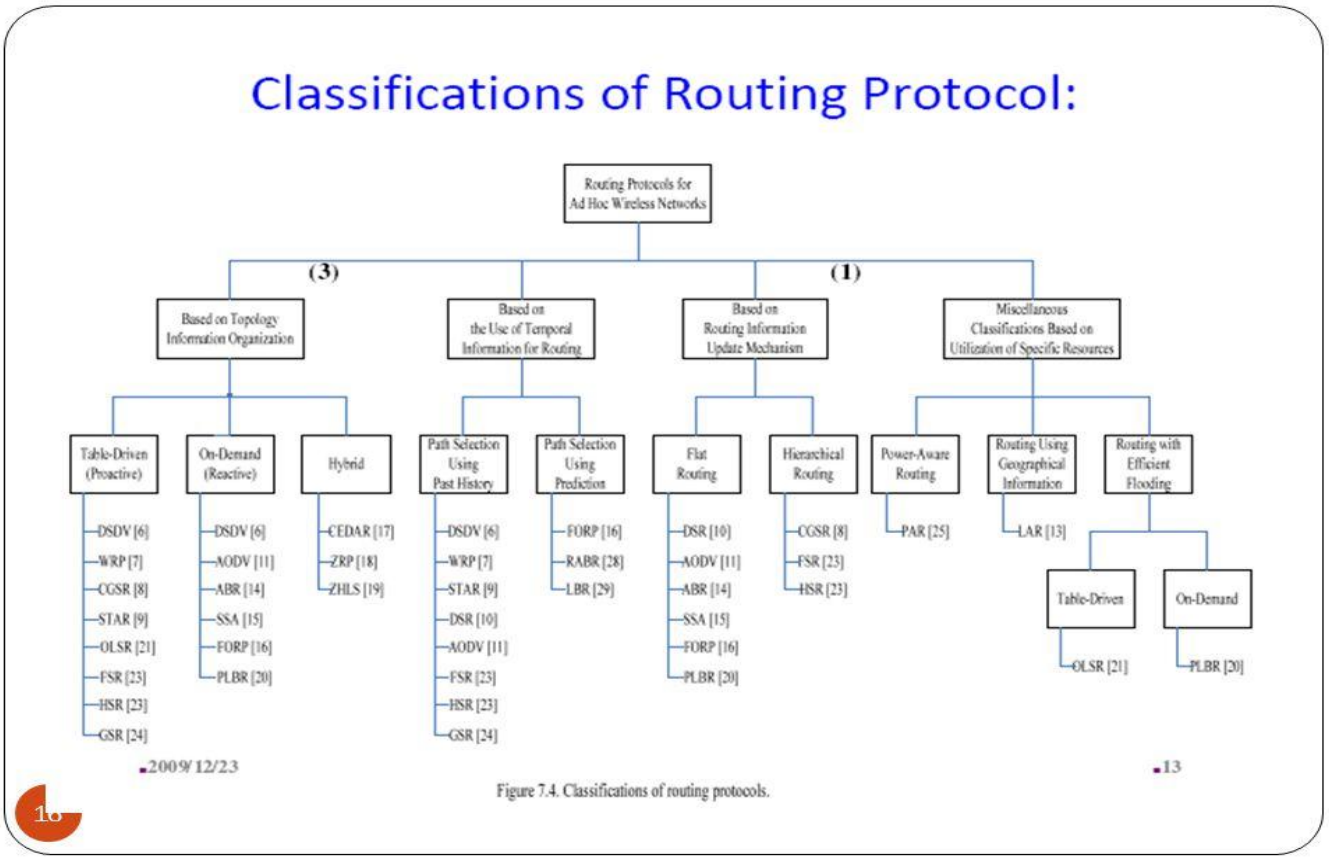
4. Wireless Mesh Network

- Wireless mesh networks are adhoc wireless network that are formed to provide an alternate communication infrastructure for mobile or fixed nodes/users, without the spectrum reuse constraint & requirement of network planning of cellular network.
- It provides many alternate paths for a data transfer session between a source & destination, resulting in quick reconfiguration of the path when the existing path fails due to node failure.
- Since the infrastructure built is in the form of small radio relaying devices, the investment required in wireless mesh networks is much less than what is required for the cellular network counterpart.
- The possible deployment scenarios of wireless mesh networks include: residential zones, highways, business zones, important civilian regions and university campuses
- Wireless mesh networks should be capable of self-organization and maintenance.
- It operates at license-free ISM band around 2.4 GHz & 5 GHz.
- It is scaled well to provide support to large number of points.
- Major advantage is the support for a high data rate, quick & low cost of deployment, enhanced services, high scalability, easy extendability, high availability & low cost per bit.

Classifications of routing protocols

The routing protocols for ad hoc wireless networks can be broadly classified into four categories based on

- Routing information update mechanism
- Use of temporal information for routing
- Routing topology
- Utilization of specific resources



Based on the Routing Information Update Mechanism

Ad hoc wireless network routing protocols can be classified into three major categories based on the routing information update mechanism. They are:

1. **Proactive or table-driven routing protocols:** In table-driven routing protocols, every node maintains the network topology information in the form of routing tables by periodically exchanging routing information. Routing information is generally flooded in the whole network. Whenever a node requires a path to a destination, it runs an appropriate path-finding algorithm on the topology information it maintains.
2. **Reactive or on-demand routing protocols:** Protocols that fall under this category do not maintain the network topology information. They obtain the

necessary path when it is required, by using a connection establishment process. Hence these protocols do not exchange routing information periodically.

3. **Hybrid routing protocols:** Protocols belonging to this category combine the best features of the above two categories. Nodes within a certain distance from the node concerned, or within a particular geographical region, are said to be within the routing zone of the given node. For routing within this zone, a table-driven approach is used. For nodes that are located beyond this zone, an on-demand approach is used.

Based on the Routing Topology

Routing topology being used in the Internet is hierarchical in order to reduce the state information maintained at the core routers. Ad hoc wireless networks, due to their relatively smaller number of nodes, can make use of either a flat topology or a hierarchical topology for routing.

1. **Flat topology routing protocols:** Protocols that fall under this category make use of a flat addressing scheme similar to the one used in IEEE 802.3 LANs. It assumes the presence of a globally unique (or at least unique to the connected part of the network) addressing mechanism for nodes in an ad hoc wireless network.

2. **Hierarchical topology routing protocols:** Protocols belonging to this category make use of a logical hierarchy in the network and an associated addressing scheme. The hierarchy could be based on geographical information or it could be based on hop distance.

Based on the Utilization of Specific Resources

1. **Power-aware routing:** This category of routing protocols aims at minimizing the consumption of a very important resource in the ad hoc wireless networks: the battery power. The routing decisions are based on minimizing the power consumption either locally or globally in the network.

2. **Geographical information assisted routing:** Protocols belonging to this category improve the performance of routing and reduce the control overhead by effectively utilizing the geographical information available.

TABLE-DRIVEN ROUTING PROTOCOLS

Destination sequenced distance-vector routing protocol

- It is an enhanced version of the distributed Bellman-Ford algorithm where each node maintains a table that contains the shortest distance and the first node on the shortest path to every other node in the network.
- It incorporates table updates with increasing sequence number tags to prevent loops, to counter the count-to-infinity problem, and for faster convergence.
- As it is a table-driven routing protocol, routes to all destinations are readily available at every node at all times.
- The tables are exchanged between neighbors at regular intervals to keep an up-to-date view of the network topology.
- The table updates are of two types:
- ***Incremental updates:*** Takes a single network data packet unit (NDPU). These are used when a node does not observe significant changes in the local topology.
- ***Full dumps:*** Takes multiple NDPUs. It is done either when the local topology changes significantly or when an incremental update requires more than a single NDPU.
- Table updates are initiated by a destination with a new sequence number which is always greater than the previous one.

- Consider the example as shown in figure one. Here node 1 is the source node and node 15 is the destination. As all the nodes maintain global topology information, the route is already available as shown in figure two.
- Here the routing table node 1 indicates that the shortest route to the destination node is available through node 5 and the distance to it is 4 hops, as depicted in figure two.
- The reconfiguration of a path used by an on-going data transfer session is handled by the protocol in the following way.
- The end node of the broken link initiates a table update message with the broken link's weight assigned to infinity (∞) and with a sequence number greater than the stored sequence number for that destination.
- Each node upon receiving an update with weight ∞, quickly disseminates it to its neighbors in order to propagate the broken-link information to the whole network.
- A node always assign an odd number to the link break update to differentiate it from the even sequence number generated by the destination.
- Figure shows the case when node 11 moves from its current position.

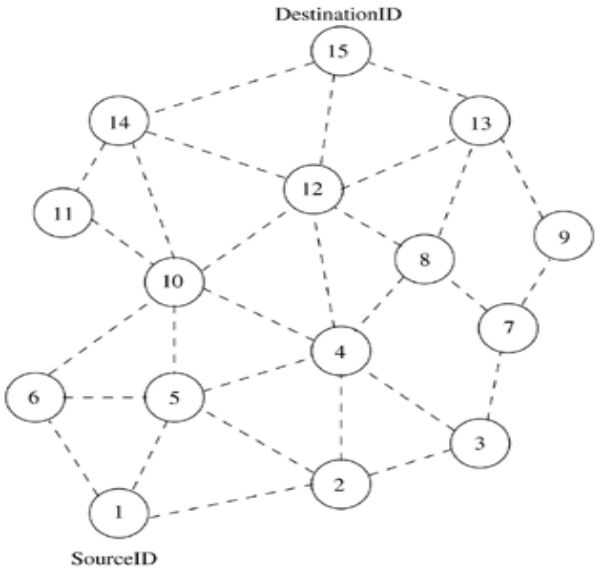
Advantages

- Less delay involved in the route setup process.
- Mechanism of incremental update with sequence number tags makes the existing wired network protocols adaptable to ad hoc wireless networks.
- The updates are propagated throughout the network in order to maintain an up-to-date view of the network topology at all nodes.

Disadvantages

- The updates due to broken links lead to a heavy control overhead during high mobility.
- Even a small network with high mobility or a large network with low mobility can completely choke the available bandwidth.
- Suffers from excessive control overhead.
- In order to obtain information about a particular destination node, a node has to wait for a table update message initiated by the same destination node.
- This delay could result in state routing information at nodes.

Route establishment in DSDV

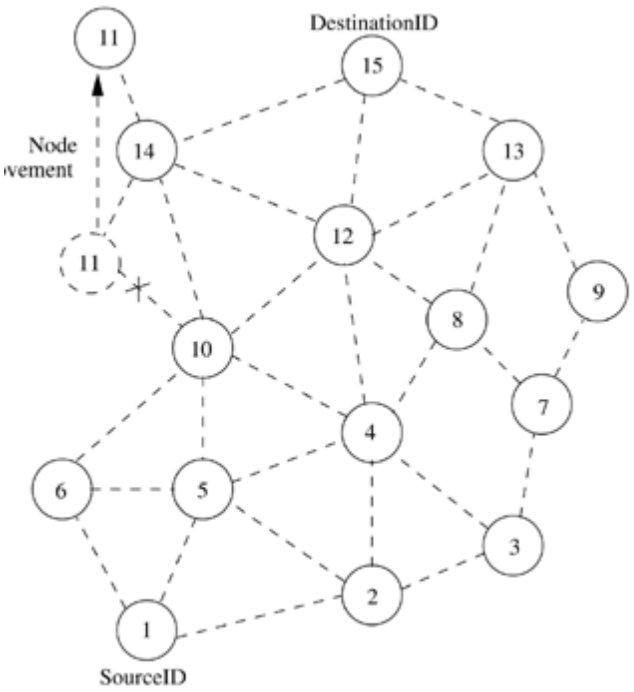


(a) Topology graph of the network

Dest	NextNode	Dist	SeqNo
2	2	1	22
3	2	2	26
4	5	2	32
5	5	1	134
6	6	1	144
7	2	3	162
8	5	3	170
9	2	4	186
10	6	2	142
11	6	3	176
12	5	3	190
13	5	4	198
14	6	3	214
15	5	4	256

(b) Routing table for Node 1

Route maintenance in DSDV



Routing Table for Node 1

Dest	NextNode	Dist	SeqNo
2	2	1	22
3	2	2	26
4	5	2	32
5	5	1	134
6	6	1	144
7	2	3	162
8	5	3	170
9	2	4	186
10	6	2	142
11	5	4	180
12	5	3	190
13	5	4	198
14	6	3	214
15	5	4	256

ON-DEMAND ROUTING PROTOCOLS

They execute the path-finding process and exchange routing information only when a path is required by a node to communicate with a destination

Dynamic Source Routing Protocol (DSR)

- Designed to restrict the bandwidth consumed by control packets in adhoc wireless networks by eliminating the periodic table update messages
- It is beacon-less and does not require periodic hello packet transmissions
- Basic approach to establish a route by flooding RouteRequest packets in the network
- Destination node responds by sending a RouteReply packet back to the source
- Each RouteRequest carries a sequence number generated by the source node and the path it has traversed
- A node checks the sequence number on the packet before forwarding it
- The packet is forwarded only if it is not a duplicate RouteRequest
- The sequence number on the packet is used to prevent loop formations and to avoid multiple transmissions
- Thus, all nodes except the destination forward a RouteRequest packet during the route construction phase
- In figure one, source node 1 initiates a RouteRequest packet to obtain a path for destination node 15
- This protocol uses a route cache that stores all possible information extracted from the source route contained in a data packet
- During network partitions, the affected nodes initiate RouteRequest packets
- DSR also allows piggy-backing of a data packet on the RouteRequest
- As a part of optimizations, if the intermediate nodes are also allowed to originate RouteReply packets, then a source node may receive multiple replies from intermediate nodes
- In figure two, if the intermediate node 10 has a route to the destination via node 14, it also sends the RouteReply to the source node
- The source node selects the latest and best route and uses that for sending data packets
- Each data packet carries the complete path to its destination
- If a link breaks, source node again initiates the route discovery process

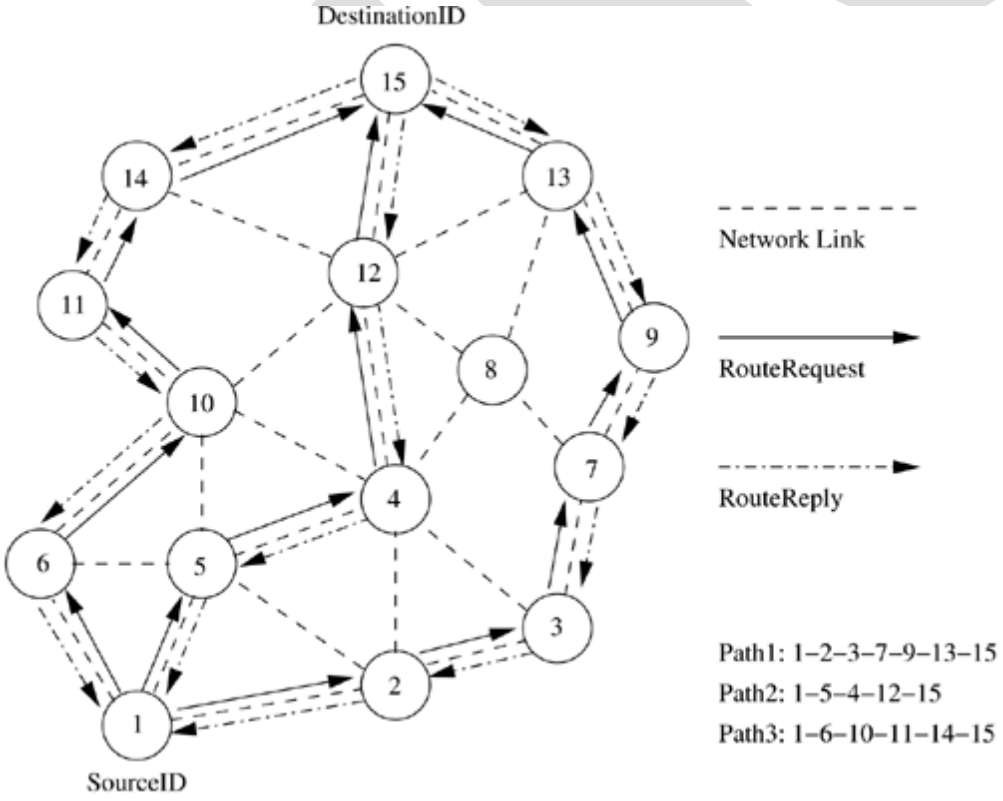
Advantages

- Uses a reactive approach which eliminates the need to periodically flood the network with table update messages
- Route is established only when required
- Reduce control overhead

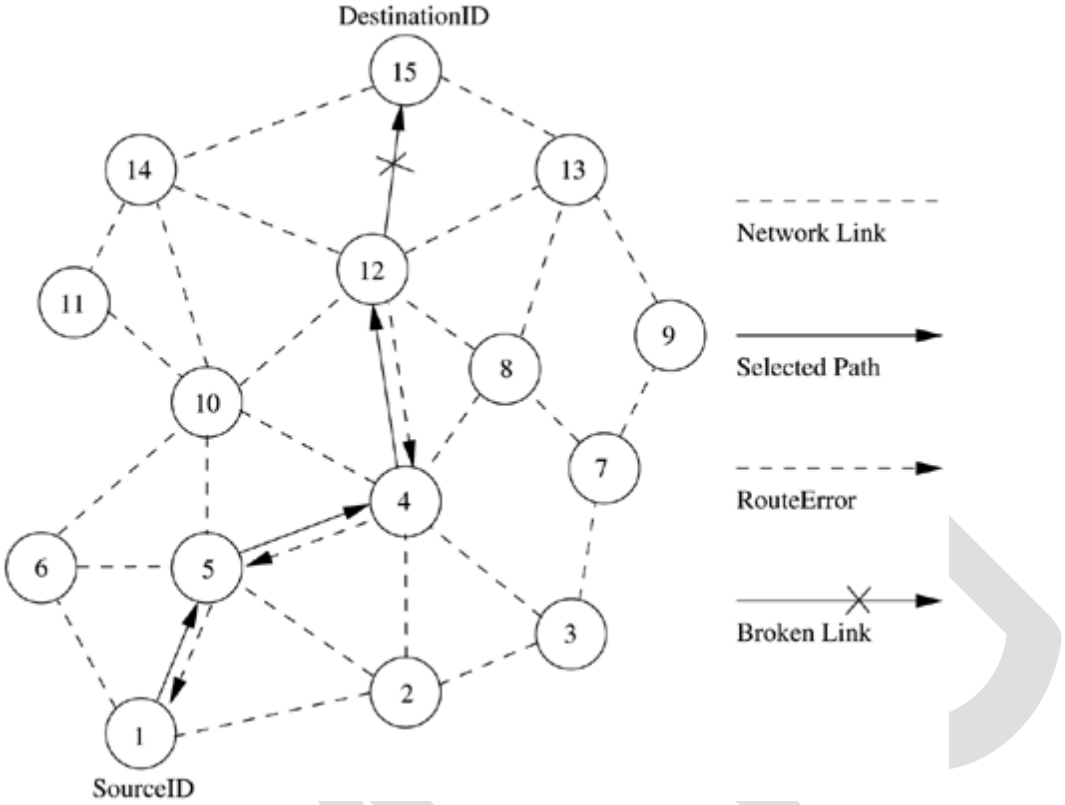
Disadvantages

- Route maintenance mechanism does not locally repair a broken link
- Stale route cache information could result in inconsistencies during route construction phase
- Connection set up delay is higher
- Performance degrades rapidly with increasing mobility
- Routing overhead is more & directly proportional to path length

Route establishment in DSR



Route maintenance in DSR



Ad Hoc On-Demand Distance Vector Routing Protocol

- Route is established only when it is required by a source node for transmitting data packets
- It employs destination sequence numbers to identify the most recent path
- Source node and intermediate nodes store the next hop information corresponding to each flow for data packet transmission
- Uses DestSeqNum to determine an up-to-date path to the destination
- A RouteRequest carries the source identifier, the destination identifier, the source sequence number, the destination sequence number, the broadcast identifier and the time to live field
- DestSeqNum indicates the freshness of the route that is accepted by the source
- When an intermediate node receives a RouteRequest, it either forwards it or prepares a RouteReply if it has a valid route to the destination
- The validity of the intermediate node is determined by comparing the sequence numbers
- If a RouteRequest is received multiple times, then duplicate copies are discarded
- Every intermediate node enters the previous node address and its BcastID
- A timer is used to delete this entry in case a RouteReply packet is not received
- AODV does not repair a broken path locally
- When a link breaks, the end nodes are notified
- Source node re-establishes the route to the destination if required

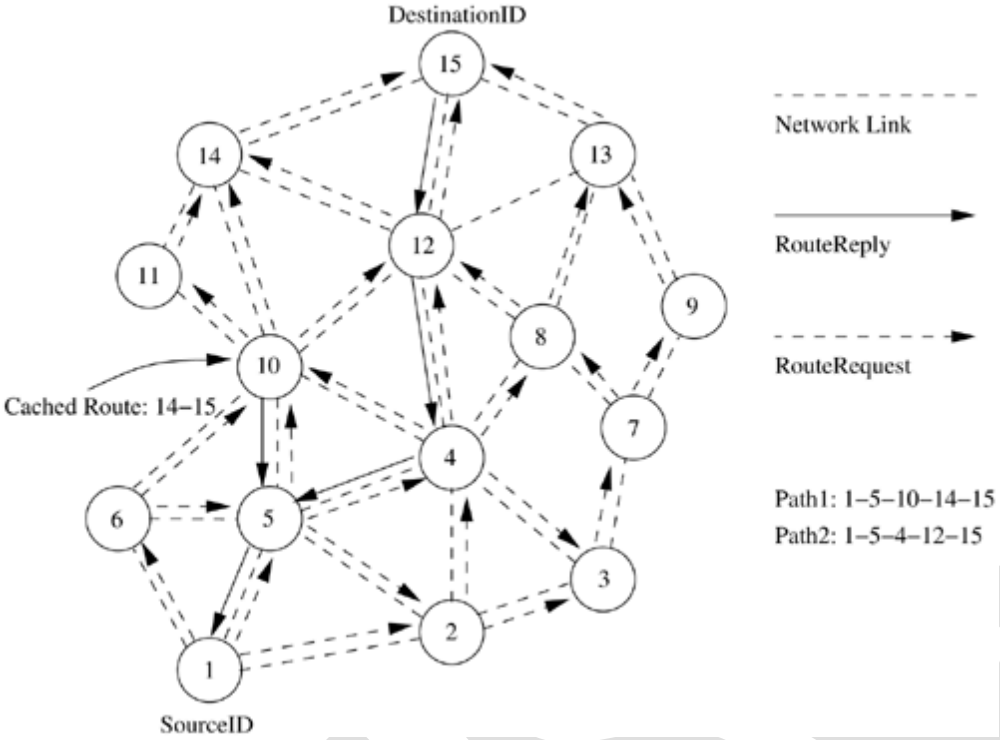
Advantage

- Routes are established on demand and DestSeqNum are used to find latest route to the destination
- Connection setup delay is less

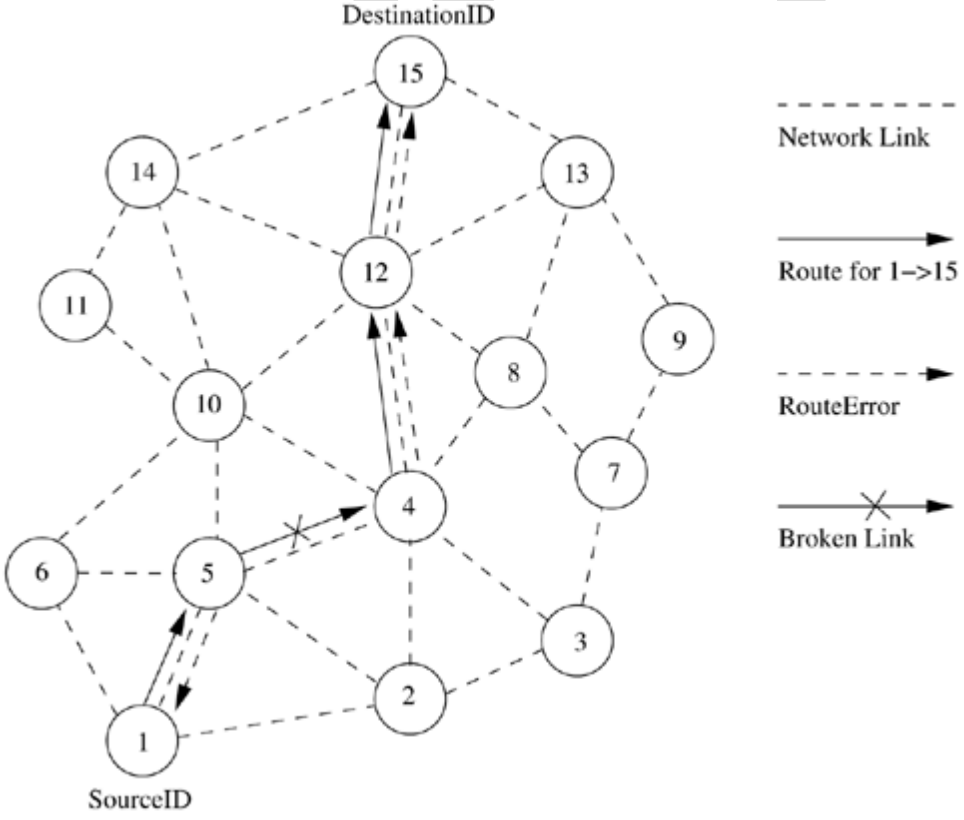
Disadvantages

- Intermediate nodes can lead to inconsistent routes if the source sequence number is very old
- Multiple RouteReply packets to single RouteRequest packet can lead to heavy control overhead
- Periodic beaconing leads to unnecessary bandwidth consumption

Route establishment in AODV



Route maintenance in AODV



UNIT II

SENSOR NETWORKS – INTRODUCTION & ARCHITECTURES

Challenges for Wireless Sensor Networks, Enabling Technologies for Wireless Sensor Networks, WSN application examples, Single-Node Architecture - Hardware Components, Energy Consumption of Sensor Nodes, Network Architecture - Sensor Network Scenarios, Transceiver Design Considerations, Optimization Goals and Figures of Merit.

Challenges for WSNs

1. **Explain the challenges for WSN.**

Characteristic requirements

In order to perform many applications in WSN, the following characteristics must be taken into consideration.

Type of service

- The service type rendered by a conventional communication network is evident – it moves bits from one place to another. For a WSN, moving bits is only a means to an end, but not the actual purpose.
- Additionally, concepts like *scoping* of interactions to specific geographic regions or to time intervals will become important.
- Hence, new paradigms of using such a network are required, along with new interfaces and new ways of thinking about the service of a network.

Quality of Service

- Traditional quality of service requirements
  - usually coming from multimedia-type applications
  - like bounded delay or minimum bandwidth are irrelevant when applications are tolerant to latency or the bandwidth of the transmitted data is very small in the first place.
- In some cases, only occasional delivery of a packet can be more than enough; in other cases, very high reliability requirements exist.
- In yet other cases, delay *is* important when actuators are to be controlled in a real-time fashion by the sensor network.

Fault tolerance

- Nodes may run out of energy or might be damaged, or since the wireless communication between two nodes can be permanently interrupted.
- It is important that the WSN as a whole is able to tolerate such faults.

Lifetime

- In many scenarios, nodes will have to rely on a limited supply of energy (using batteries).
- Replacing these energy sources in the field is usually not practicable, and simultaneously, a WSN must operate at least for a given mission time or as long as possible.

- Hence, the **lifetime** of a WSN becomes a very important figure of merit.
- Evidently, an energy-efficient way of operation of the WSN is necessary.
- The lifetime of a network also has direct trade-offs against quality of service: investing more energy can increase quality but decrease lifetime.
- The precise *definition of lifetime* depends on the application at hand. A simple option is to use the time until the first node fails (or runs out of energy) as the network lifetime.
- Other options include the time until the network is disconnected in two or more partitions.

Scalability

- Since a WSN might include a large number of nodes, the employed architectures and protocols must be able scale to these numbers.

Wide range of densities

- In a WSN, the number of nodes per unit area – the *density* of the network – can vary considerably. Different applications will have very different node densities.
- The network should adapt to such variations.

Programmability

- Nodes should be programmable, and their programming must be changeable during operation when new tasks become important.
- A fixed way of information processing is insufficient.

Maintainability

- As both the environment of a WSN and the WSN itself change (depleted batteries, failing nodes, new tasks), the system has to adapt.
- It has to monitor its own health and status to change operational parameters or to choose different trade-offs (e.g. to provide lower quality when energy resource become scarce).

Required mechanisms

- To realize these requirements, innovative mechanisms for a communication network have to be found, as well as new architectures, and protocol concepts.
- A particular challenge here is the need to find mechanisms that are sufficiently specific to the given application to support the specific quality of service, lifetime, and maintainability requirements .
- Some of the mechanisms that will form typical parts of WSNs are:

Multihop wireless communication

- In particular communication over long distances is only possible using prohibitively high transmission power.
- The use of intermediate nodes as relays can reduce the total required power.
- Hence *multihop communication* will be a necessary ingredient.

Energy-efficient operation

- To support long lifetimes, energy-efficient operation is a key technique.
- Energy-efficient data transport between two nodes (measured in J⁄bit) based on energy-efficient determination of requested information.

Auto-configuration

- A WSN will have to configure most of its operational parameters autonomously, independent of external configuration.
- The total number of nodes and simplified deployment will require that capability in most applications.

Collaboration and in-network processing

- In some applications, a single sensor is not able to decide whether an event has happened.
- But several sensors have to collaborate to detect an event and only the joint data of many sensors provides enough information.
- Information is processed in the network itself in various forms to achieve this collaboration, as opposed to having every node transmit all data to an external network and process it "at the edge" of the network.

Data centric

- In traditional communication networks the transfer of data between two specific devices, each equipped with (at least) one network address – the operation of such networks is thus **address-centric**.
- In **data-centric routing,** the sink which is responsible for gathering data and sending to the base station, issues a query for finding target data stored in the other nodes of WSN.

Locality

- Nodes, which are very limited in resources like memory, should attempt to limit the state that they accumulate during protocol processing to only information about their direct neighbors.

Exploit trade-offs

- WSNs will have to rely to a large degree on exploiting various inherent trade-offs between mutually contradictory goals, both during system/protocol design and at runtime.

****

Enabling technologies for wireless sensor networks

2. Explain the technologies used in WSN.

- Building such wireless sensor networks has only become possible with some fundamental advances in enabling technologies.

Miniaturization of hardware

- First and foremost among these technologies is the miniaturization of hardware.
- Smaller feature sizes in chips have driven down the power consumption of the basic components of a sensor node.
- This is particularly relevant to microcontrollers and memory chips as such, but also, the radio modems, responsible for wireless communication, have become much more energy efficient.
- Reduced chip size and improved energy efficiency is accompanied by reduced cost, which is necessary to make redundant deployment of nodes.

Sensing Equipment

- The actual sensing equipment is the third relevant technology.
- However, it is difficult to generalize because of the vast range of possible sensors.
- The basic parts of a sensor node have to accompanied by power supply.
- This requires, depending on application, high capacity batteries that last for long times, that is, have only a negligible self-discharge rate, and that can efficiently provide small amounts of current.
- Ideally, a sensor node also has a device for **energy scavenging**, recharging the battery with energy gathered from the environment – solar cells or vibration-based power generation.
- Such a concept requires the battery to be efficiently chargeable with small amounts of current, which is not a standard ability.
- Both batteries and energy scavenging are still objects of ongoing research.
- The counterpart to the basic hardware technologies is software.
- The division of tasks and functionalities in a single node is done by the architecture of the operating system or runtime environment.
- This environment has to support simple retasking, cross-layer information exchange, and modularity to allow for simple maintenance.
- This software architecture on a single node has to be extended to a network architecture, where the division of tasks between nodes, not only on a single node and also to structure interfaces for application programmers.

****

Applications of sensor networks

3. ***Explain the applications of WSN in detail.***

Some of the most important applications of WSN include:

Disaster relief applications

- A typical scenario is wildfire detection: Sensor nodes are equipped with thermometers and can determine their own location.
- These sensors are deployed over a wildfire, for example, a forest, from an airplane.
- They collectively produce a "temperature map" of the area or determine the perimeter of areas with high temperature that can be accessed from the outside by firefighters equipped with Personal Digital Assistants (PDAs).
- Similarly control of accidents in chemical factories.
- In military applications, where sensors should detect enemy troops rather than wildfires.
- In such an application, sensors should be cheap enough to be considered disposable since a large number is necessary; lifetime requirements are not particularly high.

Environment control and biodiversity mapping

- WSNs can be used to control the environment with respect to chemical pollutants – a possible application is garbage dump sites.
- Another example is the surveillance of the marine ground floor; an understanding of its erosion processes is important for the construction of offshore wind farms.
- Also to gain an understanding of the number of plant and animal species that live in a given habitat (biodiversity mapping).

Intelligent buildings

- Buildings waste vast amounts of energy by inefficient Humidity, Ventilation, Air Conditioning usage.
- A better, real-time, high-resolution monitoring of temperature airflow, humidity, and other physical parameters in a building by means of a WSN
- It can increase the comfort level of inhabitants and reduce the energy consumption.
- In addition, such sensor nodes can be used to monitor mechanical stress levels of buildings in seismically active zones.
- By measuring mechanical parameters like the bending load of girders, it is possible through WSN whether it is still safe to enter a given building after an earthquake. It is a considerable advantage for rescue personnel.
- Similar systems can be applied to bridges. Other types of sensors might be geared toward detecting people enclosed in a collapsed building and communicating such information to a rescue team.

Facility management

- In the management of facilities larger than a single building, WSNs also have a wide range of possible applications.
- Simple examples include keyless entry applications where people wear badges that allow a WSN to check which person is allowed to enter which areas of a larger company site.
- This example can be extended to the detection of intruders.
- Vehicles that pass a street outside of normal business hours. A wide area WSN could track such a vehicle's position and alert security personnel – this application shares many commonalities with corresponding military applications.
- WSN could be used in a chemical plant to scan for leaking chemicals.

Machine surveillance and preventive maintenance

- One idea is to fix sensor nodes to difficult to- reach areas of machinery where they can detect vibration patterns that indicate the need for maintenance.
- Examples for such machinery could be robotics or the axles of trains. Other applications in manufacturing are easily conceivable.
- The main advantage of WSNs here is the cable free operation, avoiding a maintenance problem in itself and allowing a cheap, often retrofitted installation of such sensors.

Precision agriculture

- Applying WSN to agriculture allows precise irrigation and fertilizing by placing humidity/soil composition sensors into the fields.
- Similarly, pest control can profit from a high-resolution surveillance of farm land.
- Also, livestock breeding can benefit from attaching a sensor to each pig or cow, which controls the health status of the animal (by checking body temperature, step counting, or similar means) and raises alarms if given thresholds are exceeded.

Medicine and health care

- The use of WSN in health care applications is a potentially very beneficial.
- Possibilities range from post operative and intensive care, where sensors are directly attached to patients.
- The advantage of doing away with cables is considerable to the long-term surveillance of (typically elderly) patients and to automatic drug administration (embedding sensors into drug packaging, raising alarms when applied to the wrong patient, is conceivable).
- Also, patient and doctor tracking systems within hospitals can be literally life saving.

Logistics

- In several logistics applications, it is possible to equip goods (individual parcels, for example) with simple sensors that allow a simple tracking of these objects during transportation or facilitate inventory tracking in stores or warehouses.

Telematics

- Partially related to logistics applications are applications for the telematics context, where sensors embedded in the streets or roadsides can gather information about traffic conditions. Such a so called "intelligent roadside"
- It could also interact with the cars to exchange danger warnings about road conditions or traffic jams ahead.

****

SINGLE-NODE ARCHITECTURE

4. Explain about the hardware components of sensor nodes (Nov/Dec 2018) (April/May 2018)

- Building a wireless sensor network first of all requires the constituting nodes to be developed and available.
- These nodes have to meet the requirements that come from the specific requirements of a given application:
- They might have to be small, cheap, or energy efficient, they have to be equipped with the right sensors, the necessary computation and memory resources, and they need adequate communication facilities.

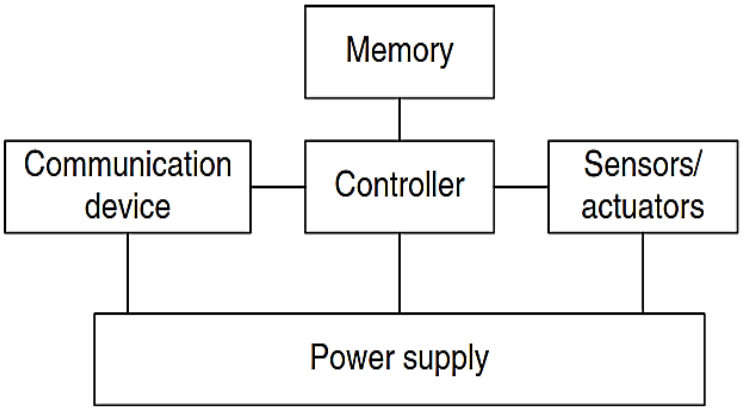


Figure 2.1 Overview of main sensor node hardware components

- **Controller: A** controller to process all the relevant data, capable of executing arbitrary code.
- **Memory:** Some memory to store programs and intermediate data; usually, different types of memory are used for programs and data.
- **Sensors and actuators: The** actual interface to the physical world: devices that can observe or control physical parameters of the environment.

- **Communication: Turning** nodes into a network requires a device for sending and receiving information over a wireless channel.
- **Power supply: Some** form of recharging by obtaining energy from the environment is available as well (e.g. solar cells).

CONTROLLER

- The controller is the core of a wireless sensor node.
- It collects data from the sensors, processes this data, decides when and where to send it, receives data from other sensor nodes, and decides on the actuator's behavior.
- It is the Central Processing Unit (CPU) of the node.it is representing trade-offs between flexibility, performance, energy efficiency, and costs.

❖ ***Microcontroller:***

- ✓ Flexibility suited to embedded systems.
- ✓ Instruction set amenable to time-critical signal processing
- ✓ Low power consumption
- ✓ Have memory built in
- ✓ Freely programmable

❖ ***Digital Signal Processors (DSPs)***

- ✓ Specialized processor
- ✓ Special architecture and their instruction set, for processing large amounts of vectorial data.
- ✓ It is used to process data coming from a simple analog, wireless communication device to extract a digital data stream.
- ✓ Another option for the controller is Field-Programmable Gate Arrays (FPGAs) or Application- Specific Integrated Circuits (ASICs).
- ✓ *FPGA*- Time and energy consumption for reprogrammable.
- ✓ *ASIC*- Less flexibility, costly hardware.

- In WSN application, the duties of the sensor nodes do not change over lifetime and where the number of nodes is big enough to warrant the investment in ASIC development is the superior solution.

MEMORY

- There is a need for Random Access Memory (RAM) to store intermediate sensor readings, packets from other nodes, and so on. While RAM is fast, its main disadvantage is that it loses its content if power supply is interrupted.
- ROM, PROM, EPROM, EEPROM can be used to store the data.
- Correctly dimensioning memory sizes, especially RAM, can be crucial with respect to manufacturing costs and power consumption.

COMMUNICATION DEVICE

- The communication device is used to exchange data between individual nodes.
- Radio Frequency (RF)-based communication provides relatively long range and high data rates, acceptable error rates at reasonable energy expenditure, and does not require line of sight between sender and receiver.

Transceivers:

- . The essential task is to convert a bit stream coming from a microcontroller (or a sequence of bytes or frames) and convert them to and from radio waves.
- It is usually convenient to use a device that combines these two tasks in a single entity. Such combined devices are called **transceivers**.
- A range of low-cost transceivers is commercially available that incorporate all the circuitry required for transmitting and receiving – modulation, demodulation, amplifiers, filters, mixers, and so on.

Transceiver tasks and characteristics

- To select appropriate transceivers, a number of characteristics should be taken into account.

❖ **Service to upper layer**

- ✓ Most notably to the Medium Access Control (MAC) layer. Sometimes, this service is **packet oriented**; sometimes, a transceiver only provides a **byte interface** or even only a **bit interface** to the microcontroller.

❖ **Power consumption and energy efficiency**

- ✓ Energy efficiency is the energy required to transmit and receive a single bit.
- ✓ Transceivers should be switchable between different states, for example, active and sleeping.
- ✓ The idle power consumption in each of these states and during switching between them is very important.

- **Carrier frequency and multiple channels**
  - ✓ Transceivers are available for different carrier frequencies; evidently, it must match application requirements and regulatory restrictions.
  - ✓ It is often useful if the transceiver provides several carrier frequencies to choose from, helping to alleviate some congestion problems in dense networks.
  - ✓ Such as FDMA or multichannel CSMA/ ALOHA techniques.
- **Data rates**
  - ✓ Carrier frequency and used bandwidth together with modulation and coding determine the gross data rate. Typical values are a few tens of kilobits per second.

- **Modulations** - Several of on/off-keying, ASK, FSK, or similar modulations.
- **Noise figure**
  - ✓ The **noise figure** is defined as the ratio of the Signal-to-Noise Ratio (SNR) ratio SNR$_{I\ at}$ the input of the element to the SNR ratio SNR$_{O\ at}$ the element's output.

NF= SNRi/SNRo

- **Receiver sensitivity**
  - ✓ The receiver sensitivity (given in dBm) specifies the minimum signal power at the receiver needed to achieve a prescribed Eb /N0 or a prescribed bit/packet error rate.
- **Blocking performance**
  - ✓ The blocking performance of a receiver is its achieved bit error rate in the presence of an interferer.
- **Frequency stability**
  - ✓ The **frequency stability** denotes the degree of variation from nominal center frequencies when environmental conditions of oscillators like temperature or pressure change.

Transceiver operational states

Many transceivers can distinguish four operational states

- **Transmit** - In the **transmit state**, the transmit part of the transceiver is active and the antenna radiates energy.
- **Receive** - In the **receive state** the receive part is active.
- **Idle**
  - ✓ A transceiver that is ready to receive but is not currently receiving anything is said to be in an **idle state**.
  - ✓ In this idle state, many parts of the receive circuitry are active, and others can be switched off.

- **Sleep**
  - ✓ In the **sleep state**, significant parts of the transceiver are switched off.
  - ✓ These sleep states differ in the amount of circuitry switched off and in the associated **recovery times** and **startup energy.**

Wakeup Receivers

- To keep this specialized receiver simple, it suffices for it to raise an event to notify other components of an incoming packet; upon such an event, the main receiver can be turned on and perform the actual reception of the packet.
- Such receiver concepts are called wakeup **receivers.**

SENSORS

- **Passive, omnidirectional sensors** -Thermometer, light sensors, vibration, microphones, humidity, mechanical stress or tension in materials
- **Passive, narrow-beam sensors -** Camera, which can "take measurements" in a given direction, but has to be rotated if need be.
- **Active sensors -** a sonar or radar sensor or some types of seismic sensors.

  Each sensor node has a certain **area of coverage** for which it can reliably and accurately report the particular quantity that it is observing.

ACTUATORS

- Actuators are just about as diverse as sensors,
- This controls a motor, a light bulb, or some other physical object is not really of concern to the way communication protocols are designed.

POWER SUPPLY OF SENSOR NODES

- **Traditional batteries**
  - ✓ The power source of a sensor node is a battery, either non rechargeable ("primary batteries") or rechargeable ("secondary batteries").
- **Capacity**
  - ✓ They should have high capacity at a small weight, small volume, and low price. The main metric is energy per volume, J/cm^3
- **Capacity under load**
  - ✓ They should withstand various usage patterns as a sensor node can consume quite different levels of power over time and actually draw high current in certain operation modes.

- **Self-discharge**
  - ✓ Their self-discharge should be low; they might also have to last for a long time
- **Efficient recharging**
  - ✓ Recharging should be efficient even at low and intermittently available recharge power;
- **Energy scavenging**
  - ✓ **E**nergy from a node's environment must be tapped into and made available to the node – **energy scavenging** should take place.
- **Photovoltaic** -The well-known solar cells can be used to power sensor nodes.
- **Vibrations**- One almost pervasive form of mechanical energy is vibrations:

****

ISSUES AND CHALLENGES IN DESIGNING A SENSOR NETWORKS

5. Explain the issues and Challenges in designing a sensor networks.

Sensor networks pose certain design challenges due to the following reasons:

- ✓ Sensor nodes are randomly deployed and hence do not fit into any regular topology. Once deployed, they usually do not require any human intervention n. Hence, the setup and maintenance of the network should be entirely autonomous.
- ✓ Sensor networks are infrastructure-less. Therefore, all routing and maintenance algorithms need to be distributed.
- ✓ Sensors usually rely only on their battery for power, which in many cases cannot be recharged or replaced. Hence, the available energy at the nodes should be considered as a major constraint while designing protocols.
- ✓ Hardware design for sensor nodes should also consider energy efficiency as a primary requirement. The micro-controller, operating system, and application software should be designed to conserve power.
- ✓ Sensor nodes should be able to synchronize with each other in a completely distributed manner, so that TDMA schedules can be imposed and temporal ordering of detected events can be performed without ambiguity.
- ✓ A sensor network should also be capable of adapting to changing connectivity due to the failure of nodes, or new nodes powering up.
- ✓ Provisions must be made for secure communication over sensor networks, especially for military applications which carry sensitive data.

****

SENSOR NETWORK ARCHITECTURE

6. Explain the architecture of wireless sensor networks. (May 2019)

- The design of sensor networks is influenced by factors such as scalability, fault tolerance, and power consumption.
- The two basic kinds of sensor network architecture are
  - ✓ Layered
  - ✓ Clustered.

Layered Architecture

- A layered architecture has a single powerful base station (BS), and the layers of sensor nodes around it correspond to the nodes that have the same hop-count to the BS.

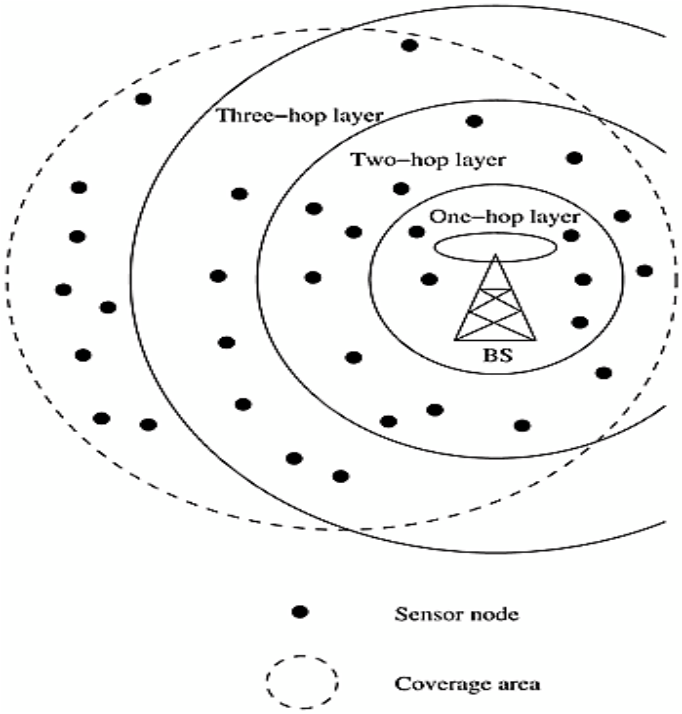


Figure: Layered architecture.

- It is used with in-building wireless backbones, and in military sensor-based infrastructure, such as the Multi-Hop Infrastructure Network Architecture (MINA).
- . In the in-building scenario, the BS acts an access point to a wired network, and small nodes form a wireless backbone to provide wireless connectivity.
- The users of the network have hand-held devices such as PDAs which communicate via the small nodes to the BS.
- Similarly, in a military operation, the BS is a data-gathering and processing entity with a communication link to a larger network.
- A set of wireless sensor nodes is accessed by the hand-held devices of the soldiers.

- ***Advantage:***
  - ✓ Each node is involved only in short-distance.
  - ✓ Low-power transmissions to nodes of the neighboring layers.

Clustered Architecture

- A clustered architecture organizes the sensor nodes into clusters, each governed by a cluster-head. The nodes in each cluster are involved in message exchanges with their cluster-heads, and these heads send message to a BS.
- Clustered architecture is useful for sensor networks because of its inherent suitability for data fusion. The data gathered by all member of the cluster can be fused at the cluster-head, and only the resulting information needs to be communicated to the BS.
- The cluster formation and election of cluster-heads must be an autonomous, distributed process.

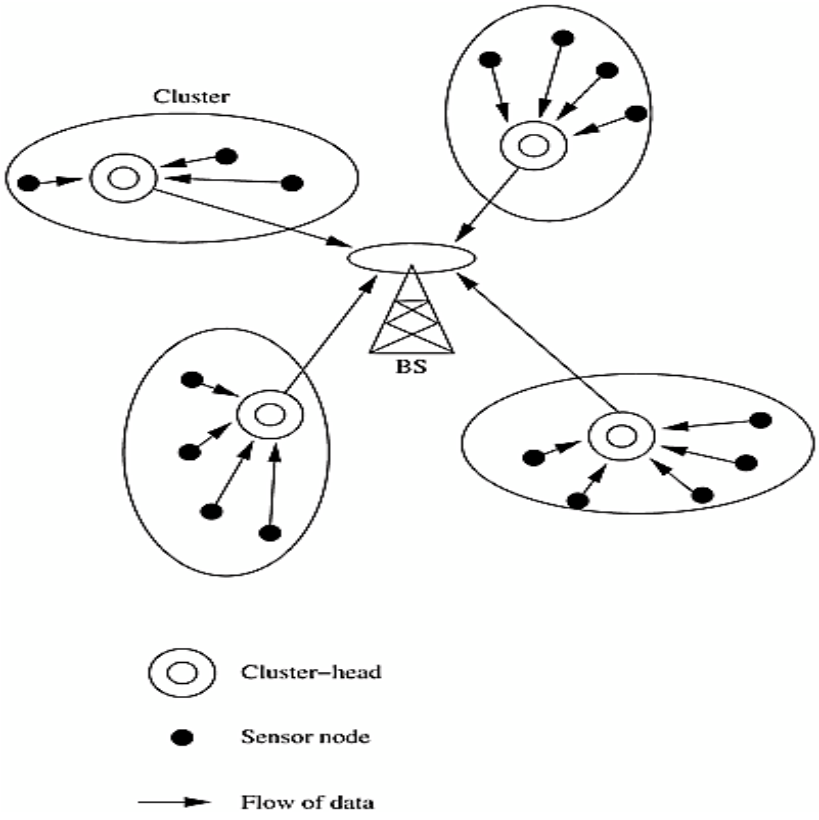


Figure: Clustered architecture.

****

ENERGY CONSUMPTION OF SENSOR NODES

7. Explain the Operation States With Different Power Consumption.

Operational states with power consumption

- Energy supply for a sensor node is at a premium: batteries have small capacity, and recharging by energy scavenging is complicated and volatile.
- Hence, the energy consumption of a sensor node must be tightly controlled.
- The main consumers of energy are the controller, the radio front ends, to some degree the memory, and, depending on the type, the sensors.

How to reduce consumption?

- To reduce power consumption of these components comes from chip-level and lower technologies:
- Designing low-power chips is the best starting point for an energy-efficient sensor node.
- Introducing and using multiple states of operation with reduced energy consumption in return for reduced functionality is the core technique for energy-efficient wireless sensor node.
- For a controller, typical states are "active", "idle", and "sleep"; a radio modem could turn transmitter, receiver, or both on or off; sensors and memory could also be turned on or off.
- The usual terminology is to speak of a "deeper" sleep state if less power is consumed.

Drawbacks of transition of states:

- Transitions between states take both time and energy.
- The usual assumption is that the deeper the sleep state, the more time and energy it takes to wake up again to fully operational state (or to another, less deep sleep state).
- Hence, it may be worthwhile to remain in an idle state instead of going to deeper sleep states.

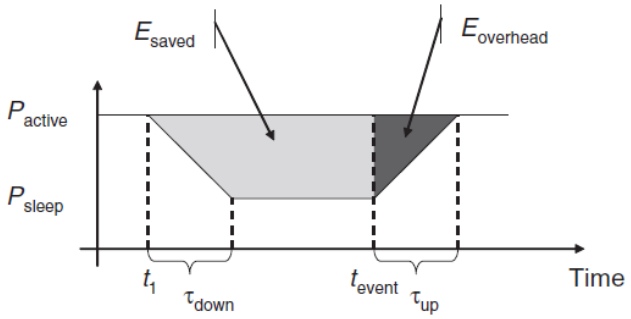


Figure Energy savings and overheads for sleep modes

- Figure illustrates this notion based on a commonly used model .

- At time t1, the decision whether or not a component (say, the microcontroller) is to be put into sleep mode should be taken to reduce power consumption from Pactive to Psleep.
- If it remains active and the next event occurs at time tevent,
  - then a total energy of Eactive = Pactive(tevent − t1) has be spent uselessly idling.
- Putting the component into sleep mode, on the other hand, requires a time τdown until sleep mode has been reached; as a simplification, assume that the average power consumption during this phase is (Pactive + Psleep)/2.
- Then, *Psleep* is consumed until *tevent.*
- In total, τdown(Pactive + Psleep)/2 + (tevent − t1 − τdown)Psleep energy is required in sleep mode as opposed to (tevent − t1)Pactive when remaining active.
- The energy saving is thus

$$E_{saved} = (t_{event} - t_1)P_{active} - (\tau_{down}(P_{active} + P_{sleep})/2 + (t_{event} - t_1 - \tau_{down})P_{sleep}).$$

- Once the event to be processed occurs, however, an additional overhead of is incurred to come back to operational state before the event can be processed, again making a simplifying assumption about average power consumption during makeup. This energy is indeed an overhead since no useful activity can be undertaken during this time.

$$E_{overhead} = \tau_{up}(P_{active} + P_{sleep})/2,$$

- switching to a sleep mode is only beneficial if E*overhead* < E*saved* or, equivalently, if the time to the next event is sufficiently large:

$$(t_{event} - t_1) > \frac{1}{2}\left(\tau_{down} + \frac{P_{active} + P_{sleep}}{P_{active} - P_{sleep}}\tau_{up}\right).$$

Microcontroller Energy Consumption

Basic power consumption in discrete operation states:

Intel StrongARM

The Intel StrongARM provides *three sleep modes:*

- In *normal mode,* all parts of the processor are fully powered. Power consumption is up to 400 mW.
- In *idle mode,* clocks to the CPU are stopped; clocks that pertain to peripherals are active.
- Any interrupt will cause return to normal mode. Power consumption is up to 100 mW.
- In *sleep mode*, only the real-time clock remains active. Wakeup occurs after a timer interrupt and takes up to 160 ms. Power consumption is up to 50 ⌈W.

Texas Instruments MSP 430

- The MSP430 family features a wider range of operation modes:
- One fully operational mode, which consumes about 1.2 mW (all power values given at 1 MHz and 3 V).
- There are four sleep modes in total.
- The deepest sleep mode, LPM4, only consumes 0.3 µW, but the controller is only woken up by external interrupts in this mode.
- In the next higher mode, LPM3, a clock is also still running, which can be used for scheduled wake ups, and still consumes only about 6 µW.

Atmel ATmega

- The Atmel ATmega 128L has six different modes of power consumption, which are in principle similar to the MSP 430.
- Its power consumption varies between 6 mW and 15 mW in idle and active modes and is about 75µW in power-down modes.

Dynamic voltage scaling

- A more sophisticated possibility than discrete operational states is to use a continuous notion of functionality/power adaptation by *adapting the speed* with which a controller operates.
- The idea is to choose the best *possible speed* with which to compute a task that has to be completed by a *given deadline.*
- One obvious solution is to switch the controller in full operation mode, compute the task at highest speed, and go back to a sleep mode as quickly as possible.
- *The alternative approach is to compute the task only at the speed that is required to finish it before the deadline. The rationale is the fact that a controller running at lower speed, that is, lower clock rates, consumes less power than at full speed. This is due to the fact that the supply voltage can be reduced at lower clock rates while still guaranteeing correct operation. This technique is called Dynamic Voltage Scaling (DVS).*
- This technique is actually beneficial for CMOS chips: reducing the voltage is a very efficient way to reduce power consumption.
- Power consumption also depends on the frequency f, hence

$$P \propto f \cdot V_{2DD.}$$

- Consequently, dynamic voltage scaling also reduces energy consumption

Memory

- From an energy perspective, the most relevant kinds of memory are on-chip memory of a microcontroller and FLASH memory – off-chip RAM is rarely if ever used.
- In fact, the power needed to drive on-chip memory is usually included in the power consumption numbers given for the controllers.
- Hence, the most relevant part is FLASH memory – in fact, the construction and usage of FLASH memory can heavily influence node lifetime.
- The relevant metrics are the read and write times and energy consumption.
- Writing is somewhat more complicated, as it depends on the granularity with which data can be accessed.

Radio Transceivers

- A radio transceiver has essentially two tasks: transmitting and receiving data between a pair of nodes.
- Similar to microcontrollers, radio transceivers can operate in different modes,
- The simplest ones are being turned on or turned off. To accommodate the necessary low total energy consumption.
- The transceivers should be turned off most of the time and only be activated when necessary – they work at a low **duty cycle**.

Modeling energy consumption during transmission

- In principle, the energy consumed by a transmitter is due to two sources :
    - One part is due to RF signal generation, which mostly depends on chosen modulation and target distance and hence on the transmission power Ptx, that is, the power radiated by the antenna.
    - A second part is due to electronic components necessary for frequency synthesis, frequency conversion, filters, and so on.

Modeling energy consumption during reception

- Similar to the transmitter, the receiver can be either turned off or turned on.
- While being turned on, it can either actively receive a packet or can be idle, observing the channel and ready to receive.
- Evidently, the power consumption while it is turned off is negligible. Even the difference between idling and actually receiving is very small and can, for most purposes, be assumed to be zero.

****

Sensor Network Scenarios

8. Explain the Sensor Network Scenarios with neat diagram.

Types of sources and sinks

- Several typical interaction patterns found in WSNs – event detection, periodic measurements, function approximation and edge detection, or tracking.
- A source is any entity in the network that can provide information, that is, typically a sensor node; it could also be an actuator node that provides feedback about an operation.
- A sink, on the other hand, is the entity where information is required.
- There are essentially three options for a sink: it could belong to the sensor network as such and be just another sensor/actuator node or it could be an entity outside this network.
- For this second case, the sink could be an actual device, for example, a handheld or PDA used to interact with the sensor network.
- It could also be merely a gateway to another larger network such as the Internet, where the actual request for the information comes from some node "far away" and only indirectly connected to such a sensor network.
- These main types of sinks are illustrated by below figure, showing sources and sinks in direct communication.

Single-hop versus multihop networks

- The inherent power limitation of radio communication follows a limitation on the feasible distance between a sender and a receiver.

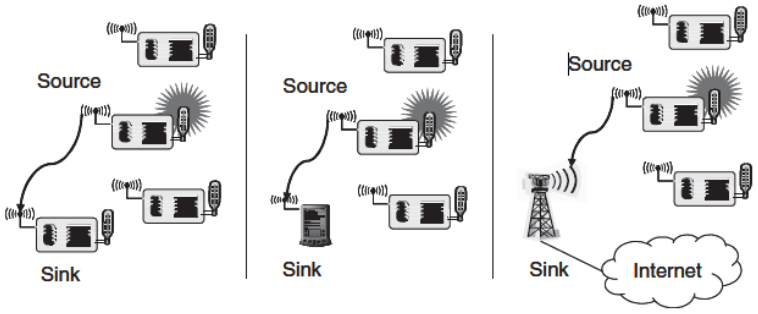


Figure 3.1 Three types of sinks in a very simple, single-hop sensor network

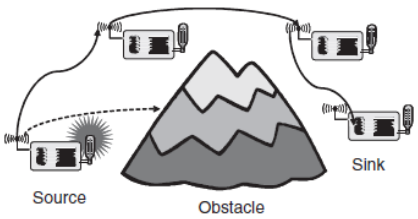


Figure 3.2 Multihop networks: As direct communication is impossible because of distance and/or obstacles, multihop communication can circumvent the problem

- Because of this limited distance, the direct communication between source and sink is not always possible, specifically in WSNs, which are intended to cover a lot of ground (e.g. in environmental

or agriculture applications) or that operate in difficult radio environments with strong attenuation (e.g. in buildings).

- To overcome such limited distances, an obvious way out is to use relay stations, with the data packets taking multi hops from the source to the sink.
- This concept of multihop networks for WSNs as the sensor nodes themselves can act as such relay nodes.
- Depending on the particular application, the likelihood of having an intermediate sensor node at the right place can actually be quite high.
- While multihopping is the solution to overcome problems with large distances or obstacles, it has been claimed to improve the energy efficiency of communication.
- The attenuation of radio signals is at least quadratic in most environments (and usually larger), it consumes less energy to use relays instead of direct communication.
- When targeting for a constant SNR at all receivers, the *radiated* energy required for direct communication over a distance d is cd^{α} (c some constant, $\alpha \geq 2$ the path loss coefficient).
- Using a relay at distance $d/2$ reduces this energy to $2c(d/2)^{\alpha}$.
- But this calculation considers only the radiated energy, not the actually *consumed* energy – in particular, the energy consumed in the intermediate relay node.
- Only for large d does the radiated energy dominate the fixed energy costs consumed in transmitter and receiver electronics.
- The concrete distance where direct and multihop communication are in balance depends on a lot of device-specific and environment-specific parameters.
- It should be pointed out that only multihop networks operating in a **store and forward** fashion. In such a network, a node has to correctly receive a packet before it can forward it somewhere.

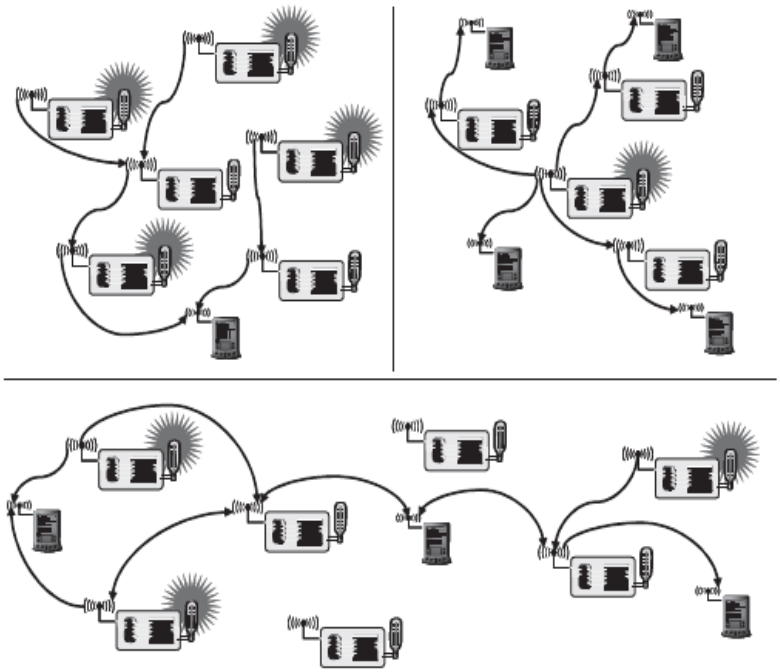


Figure 3.3 Multiple sources and/or multiple sinks. Note how in the scenario in the lower half, both sinks and active sources are used to forward data to the sinks at the left and right end of the network

Multiple sinks and sources

- In many cases, there are multiple sources and/or multiple sinks present.
- In the most challenging case, multiple sources should send information to multiple sinks, where either all or some of the information has to reach all or some of the sinks. The above figure illustrates these combinations.

Three types of mobility

- In wireless sensor networks, mobility can appear in three main forms:

Node mobility

- The wireless sensor nodes themselves can be mobile. The meaning of such mobility is highly application dependent.
- In examples like environmental control, node mobility should not happen; in livestock surveillance (sensor nodes attached to cattle, for example), it is the common rule.
- In the face of node mobility, the network has to reorganize itself frequently enough to be able to function correctly.
- It is clear that there are trade-offs between the frequency and speed of node movement on the one hand and the energy required to maintain a desired level of functionality in the network on the other hand.

Sink mobility

- The information sinks can be mobile.
- The important aspect is the mobility of an information sink that is not part of the sensor network, for example, a human user requested information via a PDA while walking in an intelligent building.
- In a simple case, such a requester can interact with the WSN at one point and complete its interactions before moving on.

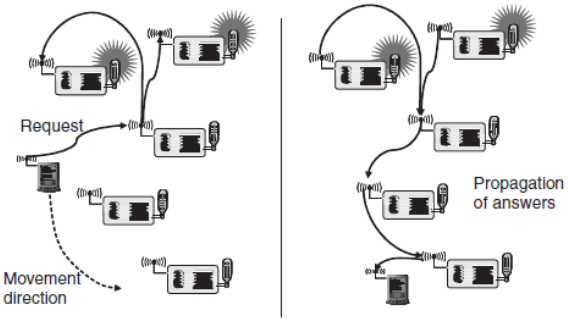


Figure 3.4 A mobile sink moves through a sensor network as information is being retrieved on its behalf

- A mobile requester is particularly interesting, however, if the requested data is not locally available but must be retrieved from some remote part of the network.
- Hence, while the requester would likely communicate only with nodes in its surrounding area, it might have moved to some other place.
- The network, possibly with the assistance of the mobile requester, must make provisions that the requested data actually follows and reaches the requester despite its movements .

Event mobility

- In applications like event detection and in particular in tracking applications, the cause of the events or the objects to be tracked can be mobile.
- In such scenarios, it is important that the observed event is covered by a sufficient number of sensors at all time.
- Hence, sensors will wake up around the object, engaged in higher activity to observe the present object, and then go back to sleep.
- As the event source moves through the network, it is accompanied by an area of activity within the network – this has been called the *frisbee* model.

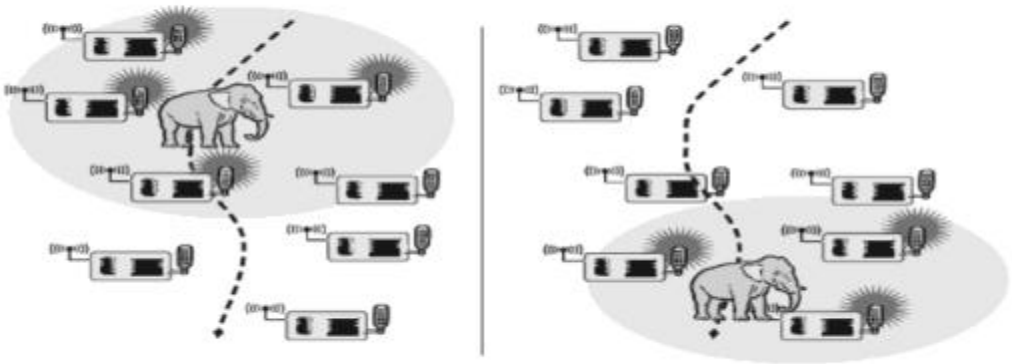


Fig.2.10 Detection of an event – an elephant that moves through the network along with the event source

****

Physical layer and transceiver design considerations in WSNs:

9. Explain the physical layer and transceiver design considerations in WSNs.

- Some of the most crucial points influencing PHY design in WSNs are:
  - Low power consumption;
    - Consequence 1: small transmit power and thus a small transmission range;
    - Consequence 2: low duty cycle; most hardware should be switched off or operated in a low power standby mode most of the time;
  - Low data rates (tens to hundreds kb/s);
  - Low implementation complexity and costs;
  - Low degree of mobility;
  - A small form factor for the overall node;
  - Low cost;

- **Energy usage profile:**
  - The radiated energy is small but the overall transceiver consumes much more energy than is actually radiated; for ex. for the Mica motes, 21 mW are consumed in transmit mode and 15 mW in received mode for a radiated power of 1 mW;
  - For small transmit powers the transmit and receive modes consume more or less the same power; therefore it is important to put the transceiver into sleep state instead of idle state;
  - This rises the problem of startup energy/ startup time which a transceiver has to spend upon waking up from sleep mode, for example, to ramp up phase – locked loops or voltage – controlled oscillators; during this startup time, no transfer of data is possible; for example, the μAMPS-1 transcveiver needs 466 μs and a power dissipation of 58 mW; therefore, going into sleep mode is unfavorable when the next wakeup comes fast;
  - Computation is cheaper than communication: the ratio is hundreds to thousands of instructions/ 1 transmitted bit;

- **Choice of modulation scheme:**
  - The choice of modulation scheme depends on several aspects, including technological factors, packet size, target error rate and channel error model;
  - The power consumption of a modulation scheme depends much more on the symbol rate than on the data rate; it leads to desire of high data rates at low symbol rates which ends to m – ary modulation schemes; trade – offs:
    - M – ary modulation schemes require more hardware than 2 – ary schemes;
    - M – ary modulation schemes require for increasing m an increased E_b/N_0 ratio;
    - Generally, in WSN applications most packets are short; for them, the startup time dominates overall energy consumption making the other efforts irrelevant;
  - Dynamic modulation scaling is necessary;

- **Antenna considerations:**
  - The small form factor of the overall sensor restricts the size and the number of antennas;
  - If the antenna is much smaller than the carrier's wavelength, it is hard to achieve good antenna efficiency and transmitted energy must increase;
  - In case of multiple antennas, they should be spaced apart at least 40 – 50% of the wavelength used to achieve good effects; for ex. for 2.4 GHz, a spacing of 5 – 6 cm between the antennas is necessary, which is difficult to be accepted;

- Radio waves emitted from antennas close to the ground, typical in some applications, are faced with higher path – loss coefficients than the common value of $\alpha = 2$; a typical value, considering the obstacles too, is $\alpha = 4$;
- Nodes randomly scattered on the ground, deployed from an aircraft, will land in random orientations, with the antennas facing the ground or being otherwise obstructed; this can lead to nonisotropic propagation of the radio wave, with considerable differences in the strength of the emitted signal in different directions.

****

Optimization Goals and Figures of Merit

10. Explain the optimization goals and figure of merit.

- For all these scenarios and application types, different forms of networking solutions can be found.
- The challenging question is how to optimize a network, how to compare these solutions, how to decide which approach better supports a given application, and how to turn relatively imprecise optimization goals into measurable figures of merit?

1. Quality of service

- WSNs differ from other conventional communication networks mainly in the type of service they offer. These networks essentially only move bits from one place to another.
- Possibly, additional requirements about the offered Quality of Service (QoS) are made, especially in the context of multimedia applications.
- Such QoS can be regarded as a low-level, networking-device-observable attribute – bandwidth, delay, jitter, packet loss rate – or as a high-level, user-observable, so-called subjective attribute like the perceived quality of a voice communication or a video transmission.
- But just like in traditional networks, high-level QoS attributes in WSN highly depend on the application. Some generic possibilities are:

Event detection/reporting probability

- What is the probability that an event that actually occurred is not detected or, more precisely, not reported to an information sink that is interested in such an event? For example, not reporting a fire alarm to a surveillance station would be a severe shortcoming.
- Clearly, this probability can depend on/be traded off against the overhead spent in setting up structures in the network that support the reporting of such an event (e.g. routing tables) or against the run-time overhead (e.g. sampling frequencies).

Event classification error

- If events are not only to be detected but also to be classified, the error in classification must be small.

Event detection delay

- The delay between detecting an event and reporting it to any/all interested sinks.

Missing reports

- In applications that require periodic reporting, the probability of undelivered reports should be small.

Approximation accuracy

- For function approximation applications (e.g. approximating the temperature as a function of location for a given area), what is the average/maximum absolute or relative error with respect to the actual function? Similarly, for edge detection applications, what is the accuracy of edge descriptions; are some missed at all?

Tracking accuracy

- Tracking applications must not miss an object to be tracked, the reported position should be as close to the real position as possible, and the error should be small.

2 Energy efficiency

- Energy is a precious resource in WSN that energy efficiency should therefore make an evident optimization goal.
- It is clear that with an arbitrary amount of energy; most of the QoS metrics can be increased.
- Hence, putting the delivered QoS and the energy required to do so into perspective should give a first, reasonable understanding of the term energy efficiency.
- The most commonly considered aspects are:

Energy per correctly received bit

- How much energy, counting all sources of energy consumption at all possible intermediate hops, is spent on average to transport one bit of information (payload) from the source to the destination? This is often a useful metric for periodic monitoring applications.

Energy per reported (unique) event

- Similarly, what is the average energy spent to report one event? Since the same event is sometimes reported from various sources, it is usual to normalize this metric to only the unique events.

Delay/energy trade-offs

- Some applications can increase energy investment for a speedy reporting of such events. Here, the trade-off between delay and energy overhead is interesting.

Network lifetime

- The time for which the network is operational or, put another way, the time during which it is able to fulfill its tasks. It is not quite clear, however, when this time ends.

Time to first node death

- When does the first node in the network run out of energy or fail and stop operating?

Network half-life

- When have 50% of the nodes run out of energy and stopped operating. Any other fixed percentile is applicable as well.

Time to partition

- When the first partition of the network in two (or more) disconnected parts occur.
- This can be as early as the death of the first node or occur very late if the network topology is robust.

Time to loss of coverage

- A possible figure of merit is thus the time when for the first time any spot in the deployment region is no longer covered by any node's observations.

Time to failure of first event notification

- A network partition can be seen as irrelevant if the unreachable part of the network does not want to report any events in the first place.
- This can be due to an event not being noticed because the responsible sensor is dead or because a partition between source and sink has occurred.

3 Scalability

- The ability to maintain performance characteristics irrespective of the size of the network is referred to as scalability.

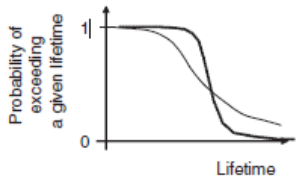


Figure 3.6 Two probability curves of a node exceeding a given lifetime – the dotted curve trades off better minimal lifetime against reduced maximum lifetime

4 Robustness

- WSN should exhibit an appropriate robustness.
- They should not fail just because a limited number of nodes run out of energy, or because their environment changes, these failures have to be compensated by finding other routes.

****

TWO MARKS

1. **What is sensor network?**

Sensor networks are highly distributed networks of small, lightweight wireless node, deployed in large numbers to monitor the environment or system.

2. **List the three subsystems of the sensor networks.**

- ✓ Sensor subsystem: senses the environment
- ✓ Processing subsystem: performs local computations on the sensed data
- ✓ Communication subsystem: responsible for message exchange with neighboring sensor nodes

3. **What are the components of WSN? (Dec 2019)**

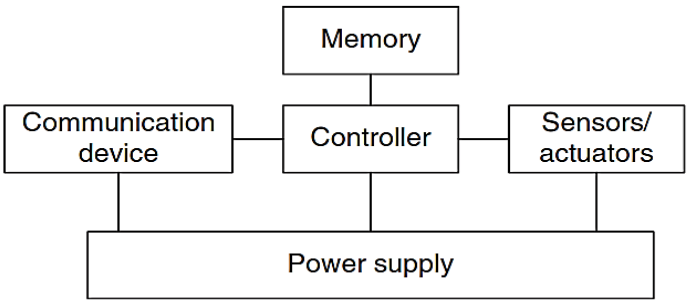


Figure 2.1 Overview of main sensor node hardware components

4. **Write short notes on memory devices in WSN.**

There is a need for Random Access Memory (RAM) to store intermediate sensor readings, packets from other nodes, and so on. While RAM is fast, its main disadvantage is that it loses its content if power supply is interrupted. ROM, PROM, EPROM, EEPROM can be used to store the data.

5. **Define transceivers in WSN.**

The essential task is to convert a bit stream coming from a microcontroller (or a sequence of bytes or frames) and convert them to and from radio waves.

It is usually convenient to use a device that combines these two tasks in a single entity. Such combined devices are called **transceivers**.

6. **Define noise figure.**

The **noise figure** NF of an element is defined as the ratio of the Signal-to-Noise Ratio (SNR) ratio $SNR_{I\ at}$ the input of the element to the SNR ratio $SNR_{O\ at}$ the element's output.

NF= SNRi/SNRo

7. What is Receiver sensitivity?

The receiver sensitivity (given in dBm) specifies the minimum signal power at the receiver needed to achieve a prescribed Eb /N0 or a prescribed bit/packet error rate.

8. What is meant by Frequency stability?

The **frequency stability** denotes the degree of variation from nominal center frequencies when environmental conditions of oscillators like temperature or pressure change.

9. Write short note on different operational states of transceiver in WSN.

Many transceivers can distinguish four operational states

- ✓ **Transmit -** In the **transmit state**, the transmit part of the transceiver is active and the antenna radiates energy.
- ✓ **Receive -** In the **receive state** the receive part is active.
- ✓ **Idle**
  - A transceiver that is ready to receive but is not currently receiving anything is said to be in an **idle state**.
- ✓ **Sleep**
  - In the **sleep state**, significant parts of the transceiver are switched off.

10. What are wakeup receivers?

To keep this specialized receiver simple, it suffices for it to raise an event to notify other components of an incoming packet; upon such an event, the main receiver can be turned on and perform the actual reception of the packet. Such receiver concepts are called wakeup **receivers.**

11. List the **issues and challenges in designing a sensor networks(April/May 2018)**

- ✓ Sensor networks pose certain design challenges due to the following reasons:
- ✓ Sensor nodes are randomly deployed and hence do not fit into any regular topology.
- ✓ Sensor networks are infrastructure-less.
- ✓ Power constraints.
- ✓ A sensor network should also be capable of adapting to changing connectivity due to the failure of nodes, or new nodes powering up.

12. List the two kinds of sensor network architecture

The two basic kinds of sensor network architecture are

- ✓ Layered
- ✓ Clustered.

13. List the advantages of layered architecture.

- ✓ Each node is involved only in short-distance.
- ✓ Low-power transmissions to nodes of the neighboring layers.

14. What is Clustered Architecture?

A clustered architecture organizes the sensor nodes into clusters, each governed by a cluster-head. The nodes in each cluster are involved in message exchanges with their cluster-heads, and these heads send message to a BS.

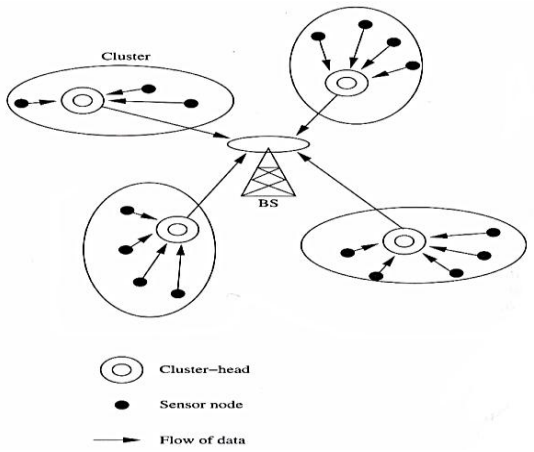


Figure: Clustered architecture.

15. What is event & Sink?

The node that generates data is call source and the information to be reported is called an event. A node which interested in an event is called sink.

16. What is QoS in WSN?

QoS can be regarded as a low-level, networking-device-observable attribute – bandwidth, delay, jitter, packet loss rate – or as a high-level, user-observable, so-called subjective attribute like the perceived quality of a voice communication or a video transmission.

17. Define Scalability.

The ability to maintain performance characteristics irrespective of the size of the network is referred to as scalability.

18. What is address and data centric in WSN?

In traditional communication networks the transfer of data between two specific devices, each equipped with (at least) one network address – the operation of such networks is thus **address-centric**.

In **data-centric routing,** the sink which is responsible for gathering data and sending to the base station, issues a query for finding target data stored in the other nodes of WSN.

19. List the three main categories of sensors.

Passive, omnidirectional sensors -Thermometer, light sensors, vibration, microphones, humidity, mechanical stress or tension in materials

Passive, narrow-beam sensors - Camera, which can "take measurements" in a given direction, but has to be rotated if need be.

Active sensors - a sonar or radar sensor or some types of seismic sensors.

20. List the types of Mobility.

- Node mobility
- Event Mobility
- Sink Mobility

21. What is ?Network half-life

When have 50% of the nodes run out of energy and stopped operating in sensor networks.

22. List the applications of sensor networks.

Disaster relief applications

Environment control and biodiversity mapping

Intelligent buildings

Facility management

Machine surveillance and preventive maintenance

Precision agriculture

Medicine and health care

23. How to reduce consumption in sensor node level?

- To reduce power consumption of these components comes from chip-level and lower technologies:
- Designing low-power chips is the best starting point for an energy-efficient sensor node.

24. What is Dynamic Voltage Scaling(DVS)?

The controller running at lower speed, that is, lower clock rates, consumes less power than at full speed. This is due to the fact that the supply voltage can be reduced at lower clock rates while still guaranteeing correct operation. This technique is called Dynamic Voltage Scaling (DVS).

25. What are the types of sink available in WSN?

There are essentially three options for a sink: it could belong to the sensor network as such and be just another sensor/actuator node or it could be an entity outside this network.

UNIT III
WSN NETWORKING CONCEPTS AND PROTOCOLS

MAC Protocols for Wireless Sensor Networks, Low Duty Cycle Protocols And Wakeup Concepts - S-MAC, The Mediation Device Protocol, Contention based protocols - PAMAS, Schedule based protocols – LEACH, IEEE 802.15.4 MAC protocol, Routing Protocols- Energy Efficient Routing, Challenges and Issues in Transport layer protocol.

MAC protocols for wireless sensor networks

1. Explain the design considerations for MAC protocols in wireless sensor networks.

Balance of requirements

- The importance of energy efficiency for the design of MAC protocols is relatively new and many of the "classical" protocols like ALOHA and CSMA contain no provisions toward this goal.
- Other typical performance figures like fairness, throughput, or delay tend to play a minor role in sensor networks.
- Further important requirements for MAC protocols are scalability and robustness against frequent topology changes.
- It is caused by nodes powering down temporarily to replenish their batteries by energy scavenging, mobility, deployment of new nodes, or death of existing nodes.

Energy problems on the MAC layer

- A nodes transceiver consumes a significant share of energy.
- The transceiver has four main states: transmitting, receiving, idling, or sleeping.
- Transmitting is costly, receive costs often have the same order of magnitude as transmit costs, idling can be significantly cheaper but also about as expensive as receiving, and sleeping costs almost nothing but results in a "deaf" node.
- Some **energy problems** and design goals are mentioned below:

Collisions

- Collisions incur useless receive costs at the destination node, useless transmit costs at the source node, and the prospect to expend further energy upon packet retransmission.

- Hence, collisions should be avoided, either by design (fixed assignment/TDMA or demand assignment protocols) or by appropriate collision avoidance/hidden-terminal procedures in CSMA protocols.

Overhearing

- Unicast frames have one source and one destination node.
- However, the wireless medium is a broadcast medium and all the source's neighbors that are in receive state hear a packet and drop it when it is not destined to them; these nodes overhear the packet.

Protocol overhead

- Protocol overhead is induced by MAC-related control frames like, RTS and CTS packets or request packets in demand assignment protocols.

Idle listening

- A node being in idle state is ready to receive a packet but is not currently receiving anything.
- This readiness is costly and useless in case of low network loads; the idle state still consumes significant energy.
- Switching off the transceiver is a solution
- A design constraint somewhat related to energy concerns is the requirement for **low complexity operation**.
- Sensor nodes shall be simple and cheap and cannot offer plentiful resources in terms of processing power, memory, or energy.
- Therefore, computationally expensive operations like complex scheduling algorithms should be avoided.

****

Low duty cycle protocols and wakeup concepts

2. Explain about Low duty protocols in WSN with neat diagram.

- **Low duty cycle protocols** try to avoid spending time in the idle state and to reduce the communication activities of a sensor node to a minimum.
- In an ideal case, the sleep state is left only when a node is about to transmit or receive packets.
- A concept for achieving this is called wakeup radio.
- In several protocols, a **periodic wakeup** scheme is used. Such schemes exist in different flavors. One is the **cycled receiver** approach is illustrated in below Figure.

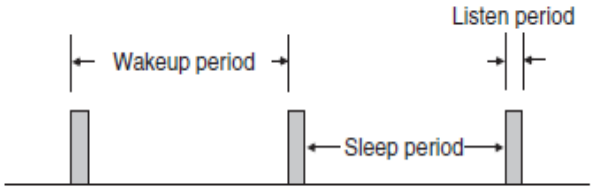


Figure 5.4 Periodic wakeup scheme

- In this approach, nodes spend most of their time in the sleep mode and wake up periodically to *receive* packets from other nodes.
- Specifically, a node *A* listens onto the channel during its **listen period** and goes back into sleep mode when no other node takes the opportunity to direct a packet to *A*.
- A potential transmitter *B* must acquire knowledge about *A*'s listen periods to send its packet at the right time – this task corresponds to a *rendezvous.*
- This rendezvous can be accomplished by letting node *A* transmit a short beacon at the beginning of its listen period to indicate its willingness to receive packets.
- Another method is to let node *B* send frequent request packets until one of them hits *A*'s listen period and is really answered by *A*.
- However, in either case, node *A* only *receives* packets during its listen period.
- If node *A* itself wants to transmit packets, it must acquire the target's listen period.
- A whole cycle consisting of sleep period and listen period is also called a **wakeup period**.
- The ratio of the listen period length to the wakeup period length is also called the node's **duty cycle**.

- By choosing a small duty cycle, the transceiver is in sleep mode most of the time, avoiding idle listening and conserving energy.
- By choosing a small duty cycle, the traffic directed from neighboring nodes to a given node concentrates on a small time window (the listen period) and in heavy load situations significant competition can occur.
- Choosing a long sleep period induces significant **per-hop latency.** In the multihop case, the per-hop latencies add up and create significant end-to-end latencies.
- Sleep phases should not be too short lest the start-up costs outweigh the benefits.
- In other protocols like S-MAC, there is also a periodic wakeup but nodes can both *transmit and receive* during their wakeup phases.
- When nodes have their wakeup phases at the same time, there is no necessity for a node wanting to transmit a packet to be awake *outside* these phases to rendezvous its receiver.

****

S-MAC

3. Explain about S-MAC protocol in WSN with neat diagram.

- The S-MAC (Sensor-MAC) protocol provides mechanisms to circumvent idle listening, collisions, and overhearing.
- S-MAC adopts a periodic wakeup scheme, that is, each node alternates between a fixed-length listen period and a fixed-length sleep period according to its **schedule.**
- The listen period of S-MAC can be used to receive *and transmit* packets.
- S-MAC attempts to coordinate the schedules of neighboring nodes such that their listen periods start at the same time.

Phases in listen period:

- A node *x*'s listen period is subdivided into three different phases:

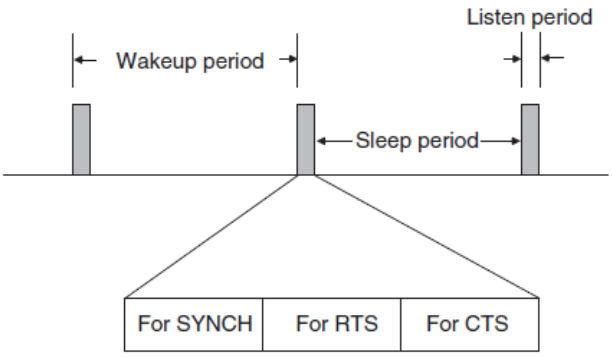


Figure 5.6 S-MAC principle

1. ***first phase***
   - In the first phase (**SYNCH phase**), node *x* accepts SYNCH packets from its neighbors.
   - In these packets, the neighbors describe their own schedule and *x* stores their schedule in a table (the **schedule table**).
   - Node *x*'s SYNCH phase is subdivided into time slots and *x*'s neighbors contend according to a CSMA scheme with additional backoff.
   - Each neighbor *y* wishing to transmit a SYNCH packet picks one of the time slots randomly and starts to transmit if no signal was received in any of the previous slots.
   - In the other case, *y* goes back into sleep mode and waits for *x*'s next wakeup.
   - In the other direction, since *x* knows a neighbor *y*'s schedule, *x* can wake at appropriate times and send its own SYNCH packet to *y* (in broadcast mode).
   - It is not required that *x* broadcasts its schedule in every of *y*'s wakeup periods.
   - However, for reasons of time synchronization and to allow new nodes to learn their local network topology, *x* should send SYNCH packets periodically. The according period is called **synchronization period**.

2. ***Second phase***
   - In the second phase (**RTS phase**), *x* listens for RTS packets from neighboring nodes.
   - In S-MAC, the RTS/CTS handshake is used to reduce collisions of data packets due to hidden-terminal situations.
   - Again, interested neighbors contend in this phase according to a CSMA scheme with additional backoff.

3. ***Third Phase***
   - In the third phase (**CTS phase**), node *x* transmits a CTS packet if an RTS packet was received in the previous phase. After this, the packet exchange continues, extending into *x*'s nominal sleep time.

<u>Working of S-MAC Protocol</u>

- When competing for the medium, the nodes use the RTS/CTS handshake, including the virtual carrier-sense mechanism.
- When transmitting in a broadcast mode (for example SYNCH packets), the RTS and CTS packets are dropped and the nodes use CSMA with backoff.

- If we can arrange that the schedules of node *x* and its neighbors are synchronized, node *x* and all its neighbors wake up at the same time and *x* can reach all of them with a single SYNCH packet.
- The S-MAC protocol allows neighboring nodes to agree on the same schedule and to create **virtual clusters**.
- The clustering structure refers solely to the exchange of schedules; the transfer of data packets is not influenced by virtual clustering.
- The S-MAC protocol proceeds as follows to form the virtual clusters:
  - A node *x*, newly switched on, listens for a time of at least the synchronization period.
  - If *x* receives any SYNCH packet from a neighbor, it adopts the announced schedule and broadcasts it in one of the neighbors' next listen periods.
  - In the other case, node *x* picks a schedule and broadcasts it.
  - If *x* receives another node's schedule during the broadcast packet's contention period, it drops its own schedule and follows the other one.
  - It might also happen that a node *x* receives a different schedule after it already has chosen one, for example, because bit errors destroyed previous SYNCH packets.
  - If node *x* already knows about the existence of neighbors who adopted its own schedule, it keeps its schedule and in the future has to transmit its SYNCH and data packets according to both schedules.
  - On the other hand, if *x* has no neighbor sharing its schedule, it drops its own and adopts the other one.
  - Since there is always a chance to receive SYNCH packets in error, node *x* periodically listens for a whole synchronization period.

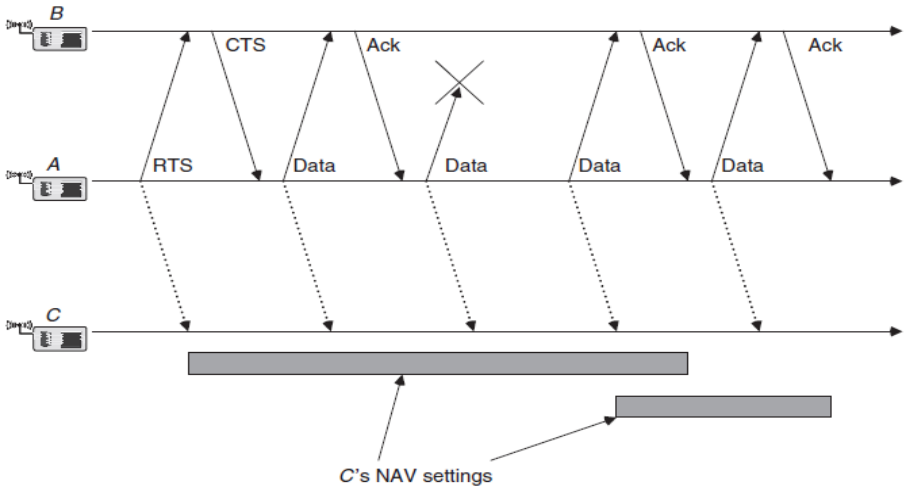


Figure 5.7 S-MAC fragmentation and NAV setting

S-MAC includes a fragmentation scheme

- ✓ A series of fragments is transmitted with only one RTS/CTS exchange between the transmitting node *A* and receiving node *B*.
- ✓ After each fragment, *B* has to answer with an acknowledgment packet.
- ✓ All the packets (data, ack, RTS, CTS) have a duration field and a neighboring node *C* is required to set its NAV field accordingly.
- ✓ In S-MAC, the duration field of all packets carries the remaining length of the whole transaction, including all fragments and their acknowledgments. Therefore, the whole message shall be passed at once.
- ✓ If one fragment needs to be retransmitted, the remaining duration is incremented by the length of a data plus ack packet, and the medium is reserved for this prolonged time.
- ✓ However, there is the problem of how a nonparticipating node shall learn about the elongation of the transaction when he has only heard the initial RTS or CTS packets.

Drawbacks:

- ✓ It is hard to adapt the length of the wakeup period to changing load situations, since this length is essentially fixed, as is the length of the listen period.

****

The mediation device protocol

4. Explain the mediation device protocol with neat diagram.

- ✓ The mediation device protocol is compatible with the peer-to-peer communication mode of the IEEE 802.15.4 low-rate WPAN standard.
- ✓ It allows each node in a WSN to go into sleep mode periodically and to wake up only for short times to receive packets from neighbor nodes.
- ✓ There is no global time reference, each node has its own sleeping schedule, and does not take care of its neighbors sleep schedules.
- ✓ Upon each periodic wakeup, a node transmits a short **query beacon**, indicating its node address and its willingness to accept packets from other nodes.
- ✓ The node stays awake for some short time following the query beacon, to open up a window for incoming packets.
- ✓ If no packet is received during this window, the node goes back into sleep mode.

- ✓ When a node wants to transmit a packet to a neighbor, it has to synchronize with it.
- ✓ The **dynamic synchronization** approach achieves this synchronization without requiring the transmitter to be awake permanently to detect the destinations query beacon.
- ✓ To achieve this, a **mediation device** (MD) is used.

Working of Mediation Device:

- ✓ Consider the scenario, mediation device is not energy constrained and can be active all the time.
- ✓ Because of its full duty cycle, the mediation device can receive the query beacons from all nodes in its vicinity and learn their wakeup periods.

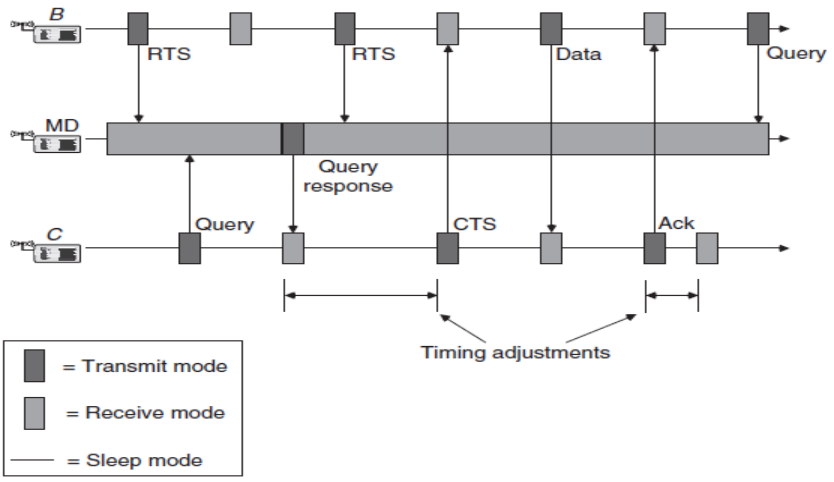


Figure 5.8 Mediation device protocol with unconstrained mediators [115, Chap. 4, Fig. 3]

- ❖ Suppose that node *A* wants to transmit a packet to node *B*.
- ❖ Node *A* announces this to the mediation device by sending periodically **request to send** (RTS) packets, which the MD captures.
- ❖ Node *A* sends its RTS packets instead of its query beacons and thus they have the same period.
- ❖ Again, there is a short answer window after the RTS packets, where *A* listens for answers.
- ❖ After the MD has received *A*'s RTS packet, it waits for *B*'s next query beacon.
- ❖ The MD answers this with a **query response** packet, indicating *A*'s address and a timing offset, which lets *B* know when to send the answering **clear to send** (CTS) to *A* such that the CTS packet hits the short answer window after *A*'s next RTS packet.
- ❖ Therefore, *B* has learned *A*'s period. After *A* has received the CTS packet, it can send its data packet and wait for *B*'s immediate acknowledgment.

- After the transaction has finished, *A* restores its periodic wakeup cycle and starts to emit query beacons again.
- Node *B* also restores its own periodic cycle and thus *decouples* from *A*'s period.

Advantages:

- It does not require any time synchronization between the nodes, only the mediation device has to learn the periods of the nodes.
- The protocol is asymmetric in the sense that most of the energy burden is shifted to the mediation device, which so far is assumed to be power unconstrained.
- The other nodes can be in the sleep state most of the time and have to spend energy only for the periodic beacons.

Drawbacks:

- The nodes transmit their query beacons without checking for ongoing transmissions and, thus, the beacons of different nodes may collide repeatedly when nodes have the same period and their wakeup periods overlap.
- However, in case of higher node densities or unwanted synchronization between the nodes, the number of collisions can be significant.

****

Wakeup radio concepts

5. Explain the wakeup radio concepts in WSN.

- If a node were always in the receiving state when a packet is transmitted to it, in the transmitting state when it transmits a packet, and in the sleep state at all other times; the idle state should be avoided.
- The **wakeup radio** concept strives to achieve this goal by a simple, "powerless" receiver that can trigger a main receiver if necessary.
- One proposed wakeup MAC protocol assumes the presence of several parallel data channels, separated either in frequency (FDMA) or by choosing different codes in a CDMA schemes.
- A node wishing to transmit a data packet randomly picks one of the channels and performs a carrier sensing operation.
- If the channel is busy, the node makes another random channel choice and repeats the carrier-sensing operation.

- After a certain number of unsuccessful trials, the node backs off for a random time and starts again.
- If the channel is idle, the node sends a wakeup signal to the intended receiver, indicating both the receiver identification and the channel to use.
- The receiver wakes up its data transceiver, tunes to the indicated channel, and the data packet transmission can proceed. Afterward, the receiver can switch its data transceiver back into sleep mode.
- It has the significant advantage that only the low-power wakeup transceiver has to be switched on all the time while the much more energy consuming data transceiver is nonsleeping if and only if the node is involved in data transmissions.
- Furthermore, this scheme is naturally **traffic adaptive**, that is, the MAC becomes more and more active as the traffic load increases.
- Periodic wakeup schemes do not have this property. However, there are also some drawbacks.
- *First*, there is no real hardware yet for such an ultralow power wakeup transceiver.
- *Second*, the range of the wakeup radio and the data radio should be the same.
- If the range of the wakeup radio is smaller than the range of the data radio, possibly not all neighbor nodes can be woken up.
- On the other hand, if the range of the wakeup radio is significantly larger, there can be a problem with local addressing schemes.
- These schemes do not use globally or network wide-unique addresses but only locally unique addresses, such that no node has two or more one-hop neighbors with the same address.
- Since the packets exchanged in the neighbor discovery phase have to use the data channel, the two hop neighborhood as seen on the data channel might be different from the two-hop neighborhood on the wakeup channel.
- *Third*, this scheme critically relies on the wakeup channel's ability to transport useful information like node addresses and channel identifications;
- This might not always be feasible for transceiver complexity reasons and additionally requires methods to handle collisions or transmission errors on the wakeup channel.
- If the wakeup channel does not support this feature, the transmitter wakes up *all* its neighbors when it emits a wakeup signal, creating an overhearing situation for most of them.

- If the transmitting node is about to transmit a long data packet, it might be worthwhile to prepend the data packet with a short **filter packet** announcing the receiving node's address.
- All the other nodes can go back to sleep mode after receiving the filter packet. Instead of using an extra packet, all nodes can read the bits of the data packet until the destination address appeared.
- If the packet's address is not identical to its own address, the node can go back into sleep mode.

****

Contention-based protocols

6. Explain the contention based protocol PAMAS with neat diagram.

- In contention-based protocols, a given transmit opportunity toward a receiver node can in principle be taken by any of its neighbors.
- If only one neighbor tries its luck, the packet goes through the channel.
- If two or more neighbors try their luck, these have to compete with each other and in unlucky cases due to hidden-terminal situations, a collision might occur, wasting energy for both transmitter and receiver.

PAMAS

- The PAMAS protocol (Power Aware Multiaccess with Signaling) originally designed for ad hoc networks.
- It provides a detailed overhearing avoidance mechanism while it does not consider the idle listening problem.
- The protocol combines the busy-tone solution and RTS/CTS handshake similar to the MACA protocol

Features of PAMAS:

- It uses two channels: a **data channel** and a **control channel**.
- All the signaling packets (RTS, CTS, busy tones) are transmitted on the control channel, while the data channel is reserved for data packets.

Protocol operation of PAMAS:

- Let us consider an idle node *x* to which a new packet destined to a neighboring node *y* arrives.
- First, *x* sends an RTS packet on the control channel without doing any carrier sensing. This packet carries both *x*'s and *y*'s MAC addresses.
- If *y* receives this packet, it answers with a CTS packet if *y* does not know of any ongoing transmission in its vicinity.
- Upon receiving the CTS, *x* starts to transmit the packet to *y* on the data channel. When *y* starts to receive the data, it sends out a **busy-tone** packet on the control channel.
- If *x* fails to receive a CTS packet within some time window, it enters the backoff mode, where a binary exponential backoff scheme is used.
- The backoff time is uniformly chosen from a time interval that is doubled after each failure to receive a CTS.
- Now, let us look at the nodes receiving *x*'s RTS packet on the control channel. There is the intended receiver *y* and there are other nodes; let *z* be one of them.
- If *z* is currently receiving a packet, it reacts by sending a busy-tone packet, which overlaps with *y*'s CTS at node *x* and effectively destroys the CTS.
- Therefore, *x* cannot start transmission and *z*'s packet reception is not disturbed. Since the busy-tone packet is longer than the CTS, we can be sure that the CTS is really destroyed.
- Next, we consider the intended receiver *y*. If *y* knows about an ongoing transmission in its vicinity, it suppresses its CTS, causing *x* to back off.
- Node *y* can obtain this knowledge by either sensing the data channel or by checking whether there was some noise on the control channel immediately after receiving the RTS.
- This noise can be an RTS or CTS of another node colliding at *y*.
- In the other case, *y* answers with a CTS packet and starts to send out a busy-tone packet as soon as *x*'s transmission has started.
- Furthermore, *y* sends out busy-tone packets each time it receives some noise or a valid packet on the control channel, to prevent its neighborhood from any activities.

****

Schedule-based protocols

7. Write short notes on advantages and disadvantages of scheduled based protocols.

Advantages:

- Schedule-based protocols that do not explicitly address idle listening avoidance but do so implicitly, for example, by employing TDMA schemes, which explicitly assign transmission and reception opportunities to nodes and let them sleep at all other times.
- In schedule-based protocols is that transmission schedules can be computed such that no collisions occur at receivers and hence no special mechanisms are needed to avoid hidden-terminal situations.

Disadvantages:

- First, the setup and maintenance of schedules involves signaling traffic, especially when faced to variable topologies.
- Second, if a TDMA variant is employed, time is divided into comparably small slots, and both transmitter and receiver have to agree to slot boundaries to actually meet each other and to avoid overlaps with other slots, which would lead to collisions.
- However, maintaining time synchronization involves some extra signaling traffic.
- Third drawback is that such schedules are not easily adapted to different load situations on small timescales. Specifically, in TDMA, it is difficult for a node to give up unused time slots to its neighbors.
- Fourth drawback is that the schedule of a node may require a significant amount of memory, which is a scarce resource in several sensor node designs.
- Finally, distributed assignment of conflict-free TDMA schedules is a difficult problem in itself.

LEACH

8. Explain the operation of LEACH protocol.

- The LEACH protocol (Low-energy Adaptive Clustering Hierarchy) assumes a dense sensor network of homogeneous, energy-constrained nodes, which shall report their data to a sink node.
- In LEACH, a TDMA based MAC protocol is integrated with clustering and a simple "routing" protocol.

- LEACH partitions the nodes into **clusters** and in each cluster a dedicated node, the **clusterhead**, is responsible for creating and maintaining a TDMA schedule; all the other nodes of a cluster are **member nodes**.
- To all member nodes, TDMA slots are assigned, which can be used to exchange data between the member and the clusterhead; there is no peer-to-peer communication.
- With the exception of their time slots, the members can spend their time in sleep state.
- The clusterhead aggregates the data of its members and transmits it to the sink node or to other nodes for further relaying.
- Since the sink is often far away, the clusterhead must spend significant energy for this transmission.
- For a member, it is typically much cheaper to reach the clusterhead than to transmit directly to the sink.
- The clusterheads role is energy consuming since it is always switched on and is responsible for the long-range transmissions.
- If a fixed node has this role, it would burn its energy quickly, and after it died, all its members would be "headless" and therefore useless.
- Therefore, this burden is rotated among the nodes. Specifically, each node decides independent of other nodes whether it becomes a clusterhead, and therefore there is no signaling traffic related to clusterhead election.
- This decision takes into account when the node served as clusterhead the last time, such that a node that has not been a clusterhead for a long time is more likely to elect itself than a node serving just recently.
- The protocol is round based, that is, all nodes make their decisions whether to become a clusterhead at the same time and the nonclusterhead nodes have to associate to a clusterhead subsequently.
- The nonclusterheads choose their clusterhead based on received signal strengths.
- The network partitioning into clusters is time variable and the protocol assumes global time synchronization.
- After the clusters have been formed, each clusterhead picks a random CDMA code for its cluster, which it broadcasts and which its member nodes have to use subsequently.
- This avoids a situation where a border node belonging to clusterhead *A* distorts transmissions directed to clusterhead *B*.

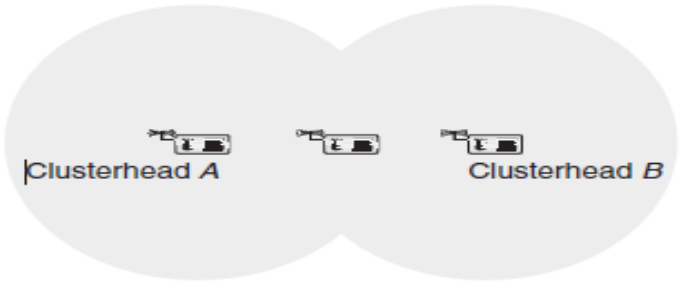


Figure 5.10 Intercluster interference

Stages of LEACH protocol:

- The protocol is organized in **rounds** and each round is subdivided into a setup phase and a steady-state phase.

Setup Phase:

- The **setup phase** starts with the self-election of nodes to clusterheads.
- In the following **advertisement phase**, the clusterheads inform their neighborhood with an advertisement packet.
- The clusterheads contend for the medium using a CSMA protocol with no further provision against the hidden-terminal problem.
- The nonclusterhead nodes pick the advertisement packet with the strongest received signal strength.
- In the following cluster-setup phase, the members inform their clusterhead ("join"), again using a CSMA protocol.
- After the cluster setup-phase, the clusterhead knows the number of members and their identifiers.
- It constructs a TDMA schedule, picks a CDMA code randomly, and broadcasts this information in the broadcast schedule subphase.

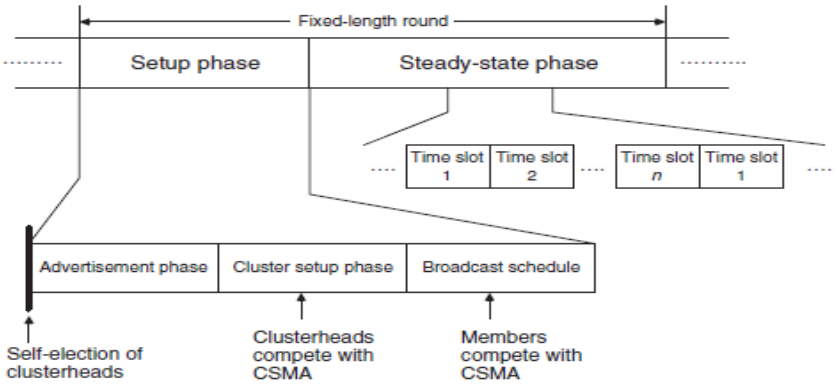


Figure 5.11 Organization of LEACH rounds

Steady state phase:

- After this, the TDMA steady-state phase begins. Because of collisions of advertisement or join packets, the protocol cannot guarantee that each non clusterhead node belongs to a cluster.
- However, it can guarantee that nodes belong to at most one cluster.
- The clusterhead is switched on during the whole round and the member nodes have to be switched on during the setup phase and occasionally in the steady-state phase, according to their position in the cluster's TDMA schedule.

Drawback:

- unable to cover large geographical areas because a clusterhead two miles away from the sink likely does not have enough energy to reach the sink at all, not to mention achieving a low BER.

Solution:

- If it can be arranged that a clusterhead can use other clusterheads for forwarding, this limitation can be mitigated.

****

IEEE 802.15.4

(Low-rate WPANs)

9. Explain about the MAC protocol in WSN (April/May 2018)(Dec 2019)

- IEEE 802.15.4: The fourth working group goes in the opposite direction for data rates.
- This group standardizes low-rate wireless personal area networks (LR-WPAN.
- The ZigBee consortium tries to standardize the higher layers of 802.15.4 similar to the activities of the Bluetooth consortium for 802.15.1 (ZigBee, 2002).
- IEEE 802.15.4 – Low-rate WPANs The reason for having low data rates is the focus of the working group on extremely low power consumption enabling multi-year battery life.
- Compared to 802.11 or Bluetooth, the new system should have a much lower complexity making it suitable for low-cost wireless communication.
- Example applications include industrial control and monitoring, smart badges, interconnection of environmental sensors, interconnection of peripherals, remote controls etc.
- The new standard should offer data rates between 20 and 250 Kbit/s as maximum and latencies down to 15 ms.

- This is enough for many home automation and consumer electronics applications.
- IEEE 802.15.4 offers two different PHY options using DSSS.
- The 868/915 MHz PHY operates in Europe at 868.0–868.6 MHz and in the US at 902–928 MHz. At 868 MHz one channel is available offering a data rate of 20 kbit/s.
- At 915 MHz 10 channels with 40 kbit/s per channel are available (in Europe GSM uses these frequencies).
- The advantages of the lower frequencies are better propagation conditions.
- However, there is also interference in these bands as many analog transmission systems use them. The 2.4 GHz PHY operates at 2.4–2.4835 GHz and offers 16 channels with 250 kbit/s per channel.
- This PHY offers worldwide operation but suffers from interference in the 2.4 GHz ISM band and higher propagation loss.
- Typical devices with 1 mW output power are expected to cover a 10–20 m range. All PHY PDUs start with a 32 bit preamble for synchronization.
- After a start-of-packet delimiter, the PHY header indicates the length of the payload (maximum 127 bytes).
- Compared to Bluetooth the MAC layer of 802.15.4 is much simpler. \

Network architecture and types/roles of nodes

- The standard distinguishes on the MAC layer two types of nodes:
  - ✓ A Full Function Device (FFD) can operate in three different roles: it can be a PAN coordinator (PAN = Personal Area Network), a simple coordinator or a device.
  - ✓ A Reduced Function Device (RFD) can operate only as a device.

Superframe structure

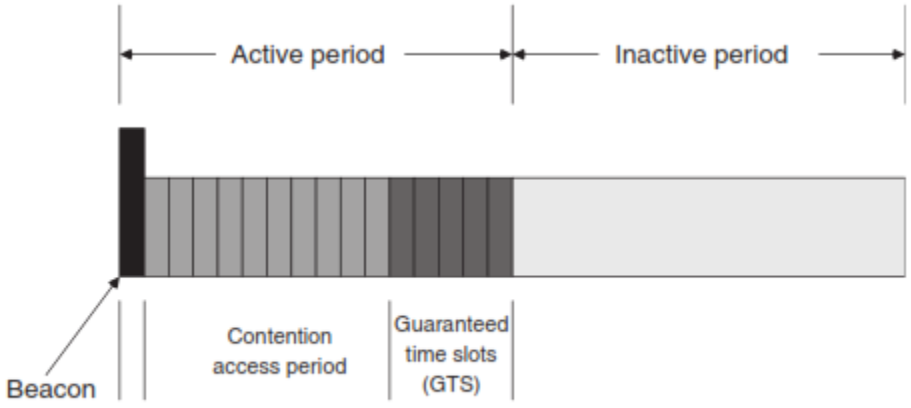
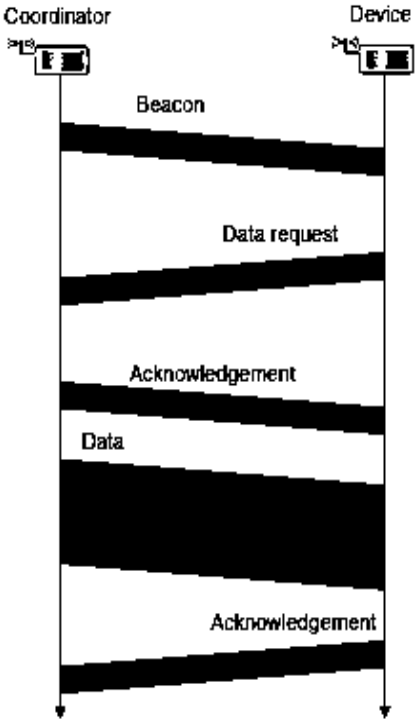


Figure 5.14 Superframe structure of IEEE 802.15.4

- ✓ The coordinator of a star network operating in the beaconed mode organizes channel access and
  data transmission with the help of a superframe structure displayed in Figure 5.14.
- ✓ All superframes have the same length. The coordinator starts each superframe by sending a frame beacon packet. The frame beacon includes a superframe specification describing the length of the
  various components of the following superframe:
- ✓ The superframe is subdivided into an active period and an inactive period. During the inactive period, all nodes including the coordinator can switch off their transceivers and go into sleep state. The nodes have to wake up immediately before the inactive period ends to receive the next beacon. The inactive period may be void.
- ✓ The active period is subdivided into 16 time slots. The first time slot is occupied by the beacon frame and the remaining time slots are partitioned into a Contention Access Period (CAP) followed by a number (maximal seven) of contiguous Guaranteed Time Slots (GTSs).

Slotted CSMA-CA protocol

- ❖ No synchronous voice links are supported. MAC frames start with a 2-byte frame control field, which specifies how the rest of the frame looks and what it contains.



Handshake between coordinator and device when the device retrieves a packet

- The following 1-byte sequence number is needed to match acknowledgements with a previous data transmission. The variable address field (0–20 bytes) may contain source and/or destination addresses in various formats.
- The payload is variable in length; however, the whole MAC frame may not exceed 127 bytes in length.
- A 16-bit FCS protects the frame. Four different MAC frames have been defined: beacon, data, acknowledgement, and MAC command.
- The time slots making up the CAP are subdivided into smaller time slots, called backoff periods.
- Optionally, this LR-WPAN offers a superframe mode. In this mode, a PAN coordinator transmits beacons in predetermined intervals (15ms–245s).
- With the help of beacons, the medium access scheme can have a period when contention is possible and a period which is contention free.
- Furthermore, with beacons a slotted CSMA/CA is available. Without beacons standard CSMA/CA is used for medium access.
- Acknowledgement frames confirming a previous transmission do not use the CSMA mechanism. These frames are sent immediately following the previous packet.
- IEEE 802.15.4 specifies three levels of security: no security, access control lists, and symmetric encryption using AES-128. Key distribution is not specified further.
- Security is a must for home automation or industry control applications. Up to now, the success of this standard is unclear as it is squeezed between Bluetooth, which also aims at cable replacement, and enhanced RFIDs/RF controllers.

****

Energy-Efficient Routing

10. Explain the Energy efficient unicast routing protocol with an example.

- Energy-efficient unicast routing appears to be a simple problem: take the network graph, assign to each link a cost value that reflects the energy consumption across this link, and pick any algorithm that computes least-cost paths in a graph.
- Figure shows an example scenario for a communication between nodes A and H including link energy costs and available battery capacity per node.
- The minimum energy route is A-B-E-H, requiring 3 units of energy.
- The minimum hop count route would be A-D-H, requiring 6 units of energy.

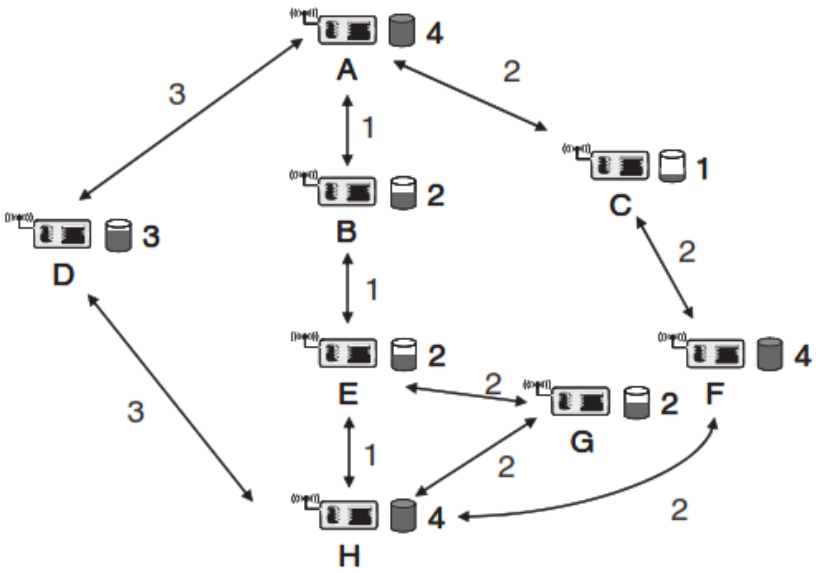


Fig: Communication between A and H through different routes.

Minimize energy per packet (or per bit):

- The total energy required to transport a packet over a multihop path from source to destination.
- The goal is then to minimize, for each packet, this total amount of energy by selecting a good route.

Maximize network lifetime:

- The network should be able to fulfill its duty for as long as possible.

Routing considering available battery energy:

- As the finite energy supply in nodes' batteries is the limiting factor to network lifetime, it stands to reason to use information about battery status in routing decisions.

Maximum Total Available Battery Capacity:

- Choose that route where the sum of the available battery capacity is maximized, without taking "maximum available power".
- Looking only at the intermediate nodes in above Figure, route A-B-E-G-H has a total available capacity of 6 units, but that is only because of the extra node G that is not really needed – such detours can of course arbitrarily increase this metric.

- Hence, AB-E-G-H should be discarded as it contains A-B-E-H as a proper subset. Eventually, route A-C-F-H is selected.

Minimum Battery Cost Routing (MBCR):

- MBCR looks at the "reluctance" of a node to route traffic instead of looking directly into the sum available battery capacities.
- This reluctance increases as its battery is drained; for example, reluctance or routing cost can be measured as the reciprocal of the battery capacity.
- Then, the cost of a path is the sum of this reciprocals and the rule is to pick that path with the smallest cost.
- Since the reciprocal function assigns high costs to nodes with low battery capacity, this will automatically shift traffic away from routes with nodes about to run out of energy.
- In the example of Figure, route A-C-F-H is assigned a cost of 1/1 + 1/4 = 1.25, but route A-D-H only has cost 1/3.
- Consequently, this route is chosen, protecting node C from needless effort.

Min–Max Battery Cost Routing (MMBCR)

- The main idea behind this routing is to protect nodes with low energy battery resources.
- Instead of using the sum of reciprocal battery levels, simply the largest reciprocal level of all nodes along a path is used as the cost for this path.
- Then, again the path with the smallest cost is used.
- In the example of Figure, route A-D-H will be selected.

Conditional Max–Min Battery Capacity Routing (CMMBCR):

- If there are routes along which all nodes have a battery level exceeding a given threshold, then select the route that requires the lowest energy per bit.
- If there is no such route, then pick that route which maximizes the minimum battery level.

Minimize variance in power levels:

- To ensure a long network lifetime, one strategy is to use up all the batteries uniformly to avoid some nodes prematurely running out of energy and disrupting the network.

- Hence, routes should be chosen such that the variance in battery levels between different routes is reduced.

Minimum Total Transmission Power Routing (MTPR) :

- A given transmission is successful if its SINR exceeds a given threshold.
- The goal is to find an assignment of transmission power values for each transmitter such that all transmissions are successful and that the sum of all power values is minimized.

****

Issues in Designing a Transport Layer Protocol For Ad Hoc Wireless Networks

11. Explain the issues in designing a transport layer protocol for adhoc wireless networks.(May/ June 2013 (NOV/DEC 2018))

1. ***Induced Traffic:***
   - In a path having multiple link, the traffic at any given link (or path) due to the traffic through neighbouring links (or paths) is referred to as induced traffic.
   - This is due to the broadcast nature of the channel and the location-dependent contention on the channel.
   - Induced Traffic affects the throughput achieved by the transport layer protocol.
2. ***Induced throughput unfairness:***
   - It refers to the throughput unfairness at the transport layer due to the throughput/delay unfairness existing at the lower layer such as the n/w and MAC layers.
   - A transport layer should consider these in order to provide a fair share of throughput across contending flows.
3. ***Separation of congestion control, reliability and flow control:***
   - A transport layer protocol can provide better performance if end-to-end reliability, flow control and congestion control are handled separately.
   - Reliability and flow control are end-to-end activities, whereas congestion can at times be a local activity.
   - The Objective is minimisation of the additional control overhead generated by them.

4. ***Power and Band width constraints:***
    - Nodes in ad hoc wireless networks face resource constraints including the two most important resources: (i) power source and (ii) bandwidth.
    - The performance of a Transport layer protocol is significantly affected by these resource constraints.
5. ***Interpretation of congestion:***
    - Interpretation of network congestion as used in traditional networks is not appropriate in ad hoc networks.
    - This is because the high error rates of wireless channel, location-dependent contention, hidden terminal problem, packet collisions in the network, path breaks due to mobility of nodes, and node failure due to drained battery can also lead to packet loss in ad hoc wireless networks.
6. ***Completely decoupled transport layer:***
    - Another challenge faced by Transport layer protocol is the interaction with the lower layers.
    - Cross-layer interaction between the transport layer and lower layers is important to adapt to the changing network environment
7. ***Dynamic topology:***
    - Experience rapidly changing network topology due to mobility of nodes.
    - Leads to frequent path breaks, partitioning and remerging of networks & high delay in re-establishment of paths.
    - Performance is affected by rapid changes in network topology.

****

Design Goals Of A Transport Layer Protocol For Ad Hoc Wireless Networks

Explain the significance and design goals of transport layer protocol for adhoc network.

- The protocol should maximize the throughput per connection.
- It should provide throughput fairness across contending flows.
- It should incur minimum connection set up and connection maintenance overheads.
- It should have mechanisms for congestion control and flow control in the network.
- It should be able to provide both reliable and unreliable connections as per the requirements of the application layer.
- It should be able to adapt to the dynamics of the network such as rapid changes in topology.

- Bandwidth must be used efficiently.
- It should be aware of resource constraints such as battery power and buffer sizes and make efficient use of them.
- It should make use of information from the lower layers for improving network throughput.
- It should have a well-defined cross-layer interaction framework.
- It should maintain End-to-End Semantics.

****

TWO MARKS

1. 1. **List the features of 802.15 standards.** **(Dec 2019)**
   - ✓ Data rates of 250 kbps, 40 kbps, and 20 kbps.
   - ✓ Two addressing modes; 16-bit short and 64-bit IEEE addressing.
   - ✓ Support for critical latency devices, such as joysticks.
   - ✓ CSMA-CA channel access.
   - ✓ Automatic network establishment by the coordinator.
   - ✓ Fully handshaked protocol for transfer reliability.
   - ✓ Power management to ensure low power consumption.
   - ✓ 16 channels in the 2.4GHz ISM band, 10 channels in the 915MHz I and one channel in the 868MHz band.
2. **What is Low-rate WPANs?**

   IEEE 802.15.4 – Low-rate WPANs The reason for having low data rates is the focus of the working group on extremely low power consumption enabling multi-year battery life.
3. **List the type of nodes that distinguish on the MAC layer.**

   The standard distinguishes on the MAC layer two types of nodes:

   A *Full Function Device (FFD*) can operate in three different roles: it can be a PAN coordinator (PAN = Personal Area Network), a simple coordinator or a device.

   A *Reduced Function Device* (RFD) can operate only as a device.
4. **Draw the Super frame structure of IEEE 802.15.4.**

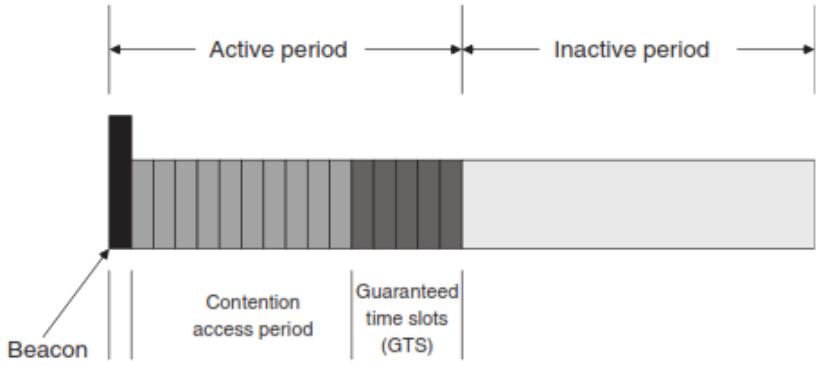


Figure 5.14 Superframe structure of IEEE 802.15.4

5. **List the challenges of MAC Protocols for Sensor Networks.**
   - ✓ No single controlling authority, so global synchronization is difficult
   - ✓ Power efficiency issue
   - ✓ Frequent topology changes due to mobility and failure

6. **Write the challenges posed by sensor network MAC protocol**
   - ✓ No single controlling authority, so global synchronization is difficult
   - ✓ Power efficiency issue
   - ✓ Frequent topology changes due to mobility and failure
7. **List the three kinds of MAC protocols used in sensor network.**

   There are three kinds of MAC protocols used in sensor network:
   - ✓ Fixed-allocation
   - ✓ Demand-based
   - ✓ Contention-based
8. **What are the mechanisms used in MAC layer? (Dec 2019)**

   The MAC protocol provides a channel of access and an addressing mechanism, so that each available node on the network may communicate with other nodes which are available – either on the same network, or on others.
9. **What is hidden terminal problem?**

   Collision occurs when both nodes transmit packets at the same time without knowing about transmission of each other.
10. **What is exposed terminal problem?**

    The exposed terminal problem refers to the inability of a node, which is blocked due to transmission by a nearby transmitting node, to transmit to another node.
11. **Mention the design considerations for MAC protocols in wireless sensor networks.**
    - ✓ Balance of requirements
    - ✓ Energy problems on the MAC layer
    - ✓ Collisions
    - ✓ Overhearing
    - ✓ Protocol overhead
    - ✓ Idle listening
12. **What is the purpose of Low duty cycle protocols?**

    It tries to avoid spending time in the idle state and to reduce the communication activities of a sensor node to a minimum.

13. **What is duty cycle?**

    The ratio of the listen period length to the wakeup period length is also called the node's **duty cycle**.

14. What is SMAC protocol?

The S-MAC (Sensor-MAC) protocol provides mechanisms to circumvent idle listening, collisions, and overhearing.

S-MAC adopts a periodic wakeup scheme, that is, each node alternates between a fixed-length listen period and a fixed-length sleep period according to its **schedule.**

15. What is the drawback of SMAC protocol?

It is hard to adapt the length of the wakeup period to changing load situations, since this length is essentially fixed, as is the length of the listen period.

16. When mediation device can be used in WSN?

- ✓ When a node wants to transmit a packet to a neighbor, it has to synchronize with it.
- ✓ The **dynamic synchronization** approach achieves this synchronization without requiring the transmitter to be awake permanently to detect the destinations query beacon.
- ✓ To achieve this, a **mediation device** (MD) is used.

17. What is wakeup radio concept?

The wakeup radio concept strives to achieve this goal by a simple, "powerless" receiver that can trigger a main receiver if necessary. One proposed wakeup MAC protocol assumes the presence of several parallel data channels, separated either

18. What is a contention protocol?

- ✓ In contention-based protocols, a given transmit opportunity toward a receiver node can in principle be taken by any of its neighbors.
- ✓ If only one neighbor tries its luck, the packet goes through the channel.

19. List the Features of PAMAS.

- ❖ It uses two channels: a **data channel** and a **control channel**.
- ❖ All the signaling packets (RTS, CTS, busy tones) are transmitted on the control channel, while the data channel is reserved for data packets.

20. What is scheduling based protocols?

Schedule-based protocols that do not explicitly address idle listening avoidance but do so implicitly, for example, by employing TDMA schemes, which explicitly assign transmission and reception opportunities to nodes and let them sleep at all other times.

21. What is LEACH protocol?

- ✓ The LEACH protocol (Low-energy Adaptive Clustering Hierarchy) assumes a dense sensor network of homogeneous, energy-constrained nodes, which shall report their data to a sink node.
- ✓ In LEACH, a TDMA based MAC protocol is integrated with clustering and a simple "routing" protocol.

22. List the issues in designing a transport layer protocol for adhoc wireless networks.

- ✓ Induced Traffic:
- ✓ Induced throughput unfairness:
- ✓ Separation of congestion control, reliability and flow control:
- ✓ Power and Band width constraints:
- ✓ Interpretation of congestion:
- ✓ Completely decoupled transport layer:
- ✓ Dynamic topology:

23. List the design goals of transport layer protocol for adhoc network.

- ✓ The protocol should maximize the throughput per connection.
- ✓ It should provide throughput fairness across contending flows.
- ✓ It should incur minimum connection set up and connection maintenance overheads.
- ✓ It should have mechanisms for congestion control and flow control in the network.
- ✓ It should be able to provide both reliable and unreliable connections as per the requirements of the application layer.

24. Draw the frame structure of LEACH.

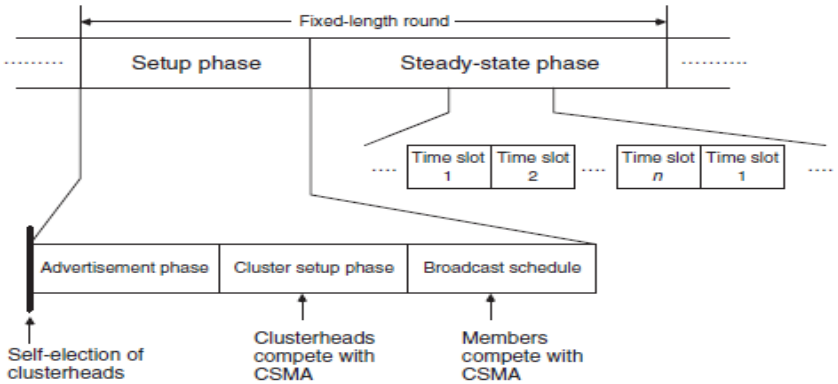


Figure 5.11 Organization of LEACH rounds

25. Draw the frame structure of SMAC.

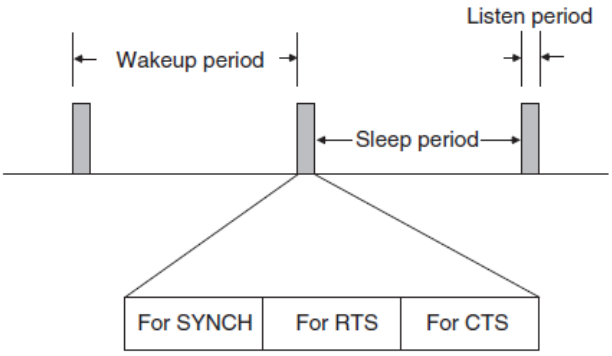


Figure 5.6 S-MAC principle

UNIT IV

SENSOR NETWORK SECURITY

Network Security Requirements, Issues and Challenges in Security Provisioning, Network Security Attacks, Layer wise attacks in wireless sensor networks, possible solutions for jamming, tampering, black hole attack, flooding attack. Key Distribution and Management, Secure Routing – SPINS, reliability requirements in sensor networks.

NETWORK SECURITY REQUIREMENTS

1. Explain the requirements of security in Adhoc networks.

A security protocol for ad hoc wireless networks should satisfy the following requirements

- ✓ ***Confidentiality:***
  - ➢ The data sent by the sender must be comprehensible only to the intended receiver.
  - ➢ Though an intruder might get hold of the data being sent, he / she must not be able to derive any useful information out of the data.
  - ➢ One of the popular techniques used for ensuring confidentiality is data encryption.
- ✓ ***Integrity:***
  - ➢ The data sent by the source node should reach the destination node without being altered.
  - ➢ It should not be possible for any malicious node in the network to tamper with the data during transmission.
- ✓ ***Availability:***
  - ➢ The network should remain operational all the time.
  - ➢ It must be robust enough to tolerate link failures and also be capable of surviving various attacks mounted on it.
  - ➢ It should be able to provide guaranteed services whether an authorized user requires them
- ✓ ***Non-Repudiation:***
  - ➢ It is a mechanism to guarantee that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message.
  - ➢ Digital signatures are used for this purpose.

****

ISSUES AND CHALLENGES IN SECURITY PROVISIONING

2. ***Explain how the security provisioning in adhoc network differs from that in infrastructure based network.***

Shared broadcast radio channel :

- The radio channel used for communication in adhoc wireless networks is broadcast in nature & is shared by all nodes within its direct transmission range.
- Data transmitted by a node is received by all nodes within its direct transmission range. So a malicious node could easily obtain data being transmitted in the network.
- This problem can be minimized to a certain extent by using directional antennas.

3. ***Limited resource availability :***
   - Resources such as bandwidth, battery power, & computational power are scarce in adhoc wireless networks.
   - Hence it is difficult to implement complex cryptography-based security mechanisms in networks.
4. ***Insecure operational environment :***
   - The operating environments where adhoc wireless is used may not always be secure.
   - One important application of such networks is in battlefields.
5. ***Physical Vulnerability :***
   - Nodes in these networks are usually compact & hand-held in nature.
   - They could get damaged easily & are also vulnerable to theft.
6. ***Lack of central authority:***
   - In wired networks & infrastructure-based wireless networks, it would be possible to monitor the traffic on the network through certain important central points & implement security mechanisms at such points.
   - Since adhoc –wireless networks do not have central points, these mechanisms cannot be applied in ad hoc wireless networks.
7. ***Lack of associations:***
   - Since these networks are dynamic in nature, a node can join or leave the network at any point of time.
   - If no proper authentication mechanism is used for associating nodes in a network, an intruder would be able to join into the network quite easily & carry out his/her attacks.

8. **Limited Resource availability:**
   - Resources such as Bandwidth, battery power and computational power are scarce in WSN.
   - Hence It is difficult to implement complex cryptography based security mechanisms in such networks.
9. **Physical Vulnerability:**
   - Nodes in these networks are usually compact and handheld in nature.
   - They could get damaged easily and are also vulnerable to theft.

****

NETWORK SECURITY ATTACKS

3. Explain various network and application layer security attacks in detail.

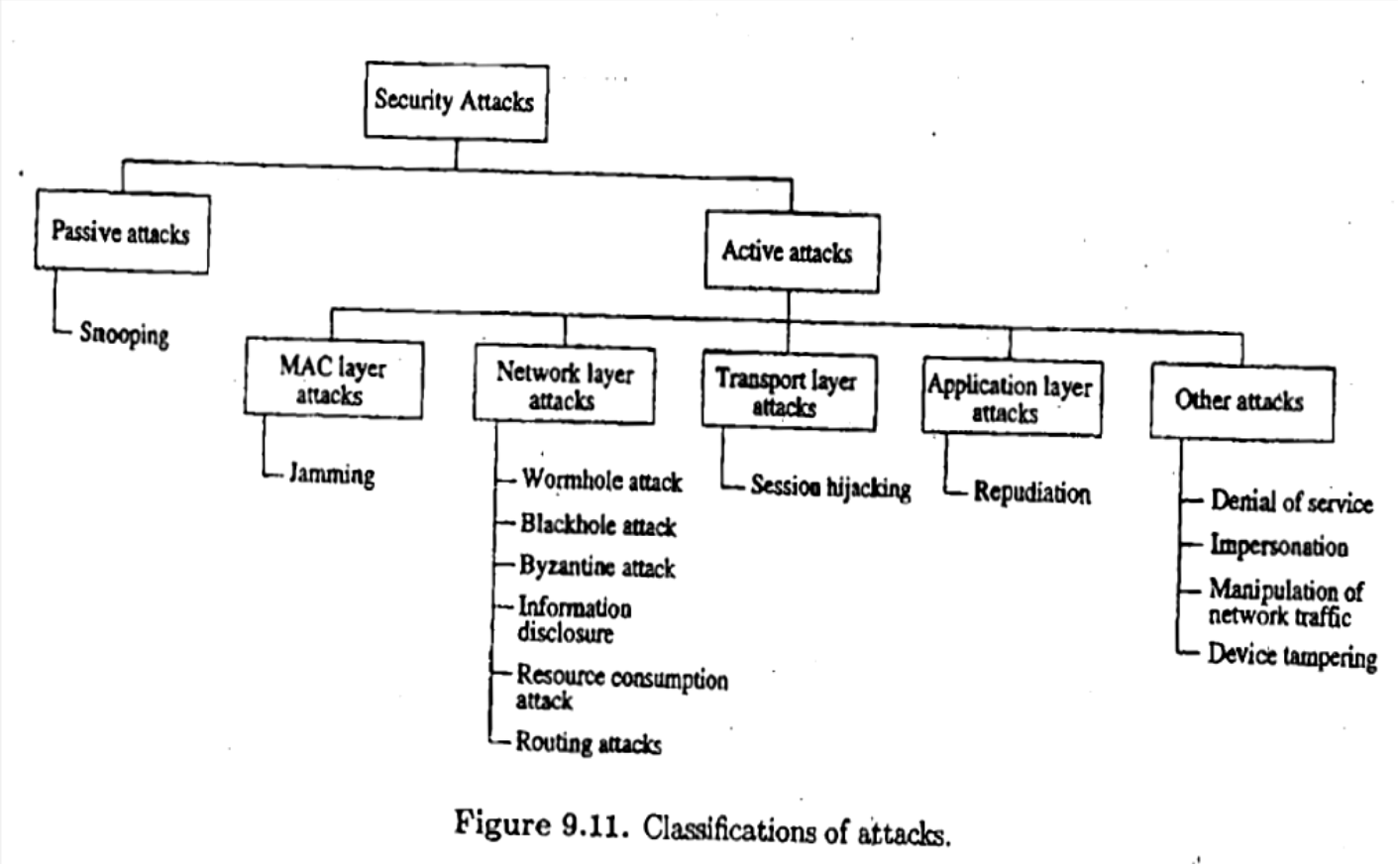


Figure 9.11. Classifications of attacks.

Attacks on adhoc wireless networks can be classified into 2 broad categories, namely:

1. Passive attack

- It does not disrupt the operation of the network; the adversary snoops the data exchanged in the network without altering it.

- One way to overcome such problems is to use powerful encryption mechanisms to encrypt the data being transmitted.

2. Active attack

- An active attack attempts to alter or destroy the data being exchanged in the network, thereby disrupting the normal functioning of the network.
- They can be further classified into 2 categories :
    i. *External attacks*, which are carried out by nodes that do not belong to the network. They can be prevented using standard encryption techniques and firewalls.
    ii. *Internal attacks* are from compromised nodes that are actually part of the network.

Network Layer Attacks

There are many types of attacks pertaining to the network layer in network protocol stack. Some of them are as follows:

1.*Wormhole attack:*

- In this attack, an attacker receives packets at one location in the network & tunnels them (possibly selectively) to another location in the network, where the packets are resent into the network. This tunnel between 2 colliding attackers is referred to as a wormhole.
- If proper mechanisms are not employed to defend the network against wormhole attacks, existing routing protocols for adhoc wireless networks may fail to find valid routes.

2. *Blackhole attack:*

- In this attack, a malicious node falsely advertises good paths to destination node during path-finding process or in route update messages.
- The intention of malicious node could be to hinder the path-finding process or to intercept all data packets being sent to the destination node.

3. *Byzantine attack:*

- Here, a compromised intermediate note or a set of compromised intermediate nodes work in collision & carries out attack such as creating routing loops, routing packets on non-optimal paths & selectively dropping packets.

4. *Information disclosure:*

- A compromised node may leak confidential or important information to unauthorized nodes in the network.

5. *Resource consumption attack:*

- In this attack, a malicious node tries to consume/waste resources of other nodes present in the network.

- ❖ The resources targeted are battery power, bandwidth & computational power, which are limitedly available in adhoc wireless networks.

6. *Routing attacks:*

There are several types of attacks mounted on routing protocol & they are as follows:

i. Routing table overflow:

- ➢ In this type of attack, an adversary node advertises routes to non-existent nodes, to the authorized nodes present in the network.
- ➢ The main objective of this attack is to cause an overflow of routing tables, which would in turn prevent the creation of entries corresponding to new routes to authorized nodes.

ii. *Routing table poisoning*:

- ➢ Here, the compromised nodes in the networks send fictitious routing updates or modify genuine route update packets sent to other uncompromised nodes.
- ➢ This may result in sub-optimal routing, congestion in network or even make some parts of network inaccessible.

iii. *Packet replication:*

- ➢ In this attack, an adversary node would replicate state packets.

iv. *Route cache poisoning:*

- ➢ Similar to routing table poisoning, an adversary can also poison the route cache to achieve similar activities.

v. Rushing attack:

- ➢ On-demand routing protocols that use duplicate suppression during the route discovery process are vulnerable to this attack.

Transport Layer Attacks:

1. *Session Hijacking:*

- ❖ Here, an adversary takes control over a session between 2 nodes.
- ❖ Since most authentication processes are carried out only at the start of session, once the session between 2 nodes get established, the adversary node masquerades as one of the end-nodes of the session & hijacks the sessions.

Application Layer Attacks:

1. Repudiation:

- ✓ It refers to the denial or attempted denial by a node involved in a communication of having participated in all or part of the communication

Other Attacks:

It discusses security attacks that cannot strictly be associated with any specific layer in the network protocol stack

Multi-layer attacks

Multi-layer attacks are those that could occur in any layer of the network protocol stack. Some of the multi-layer attacks in adhoc wireless networks are:

1. Denial of Service

- ✓ In this type of attack, an adversary attempts to prevent legitimate & authorized users of services offered by the network from accessing those services.
- ✓ This may lead to a failure in the delivery of guaranteed services to the end users.
- ✓ Some of the DoS attacks are as follows:

a. ***Jamming*** – in this form of attack, the adversary initially keeps monitoring the wireless medium in order to determine the frequency at which the receiver node is receiving signals from the sender. Frequency hopping spread spectrum(FHSS) and direct sequence spread spectrum (DSSS) are two commonly used techniques that overcome jamming attacks

b. ***SYN flooding*** – here, an adversary sends a large number of SYN packets to a victim node, spoofing the return addresses of the SYN packets. The victim node builds up a table/data structure for holding information regarding all pending connections. Since the maximum possible size of the table is limited, the increasing number of half-connections results in an overflow in the table.

c. ***Distributed DoS attack*** – Several adversaries that are distributed throughout the network collide and prevent legitimate users from accessing the services offered by the network.

2. ***Impersonation***

- ✓ In these attacks, an adversary assumes the identity & privileges of an authorized node, either to make use of network resources that may not be available to it under normal circumstances, or to disrupt the normal functioning of the network by injecting false routing information into the network.
- ✓ A *man-in-the-middle* attack is another type of impersonation attack.

3. Device Tampering - Mobile devices get damaged or stolen easily.

****

KEY DISTRIBUTION AND MANAGEMENT

4. Explain in detail about Key Management approaches which includes symmetric and Asymmetric algorithms.

- Inorder to overcome the attacks. various techniques are employed.
- **CRYPTOGRAPHY** is one of the most common & reliable means to ensure security & can be applied to any communication network.
- In the parlance of cryptography, the original information to be sent from one person to another is called ***plaintext***.
- The plaintext is converted into ***ciphertext*** by the process of ***encryption.***
- An authentic receiver can decrypt / decode the ciphertext back into plaintext by the process of ***decryption.***
- The process of encryption and decryption are governed by keys, which are small amounts of information used by the cryptographic algorithms. When the keys is to be kept secret to ensure the security of the system, it is called a ***secret key.***
- The secure administration of cryptographic keys is called ***Key Management.***
- The 4 main goals of cryptography are confidentiality, integrity, authentication & non-repudiation.
- There are 2 major kinds of cryptographic algorithms:
    1. ***Symmetric*** *key algorithms*, which use the same key for encryption & decryption.
    2. ***Asymmetric*** *key algorithms,* which use two different keys for encryption & decryption.
- The asymmetric key algorithms are based on some mathematical principles which make it feasible or impossible to obtain one key from another; therefore, one of the keys can be made public while the others is kept secret (private). This is called *public key cryptography*

SYMMETRIC KEY ALGORITHMS

- Symmetric key algorithms rely on the presence of shared key at both the sender & receiver, which has been exchanged by some previous arrangement.
- There are 2 kinds of symmetric key algorithms:
    - One involving block ciphers &
    - The stream ciphers.
- A block cipher is an encryption scheme in which plaintext is broken into fixed-length segments called blocks, & the blocks are encrypted one at a time.
- The simplest example includes substitution & transposition.
- In *substitution*, each alphabet of plaintext is substituted by another in the cipher text,& this table mapping of the original & the substituted alphabet is available at both the sender & receiver.

- A *Transposition cipher,* permutes the alphabet in plaintext to produce the cipher text.

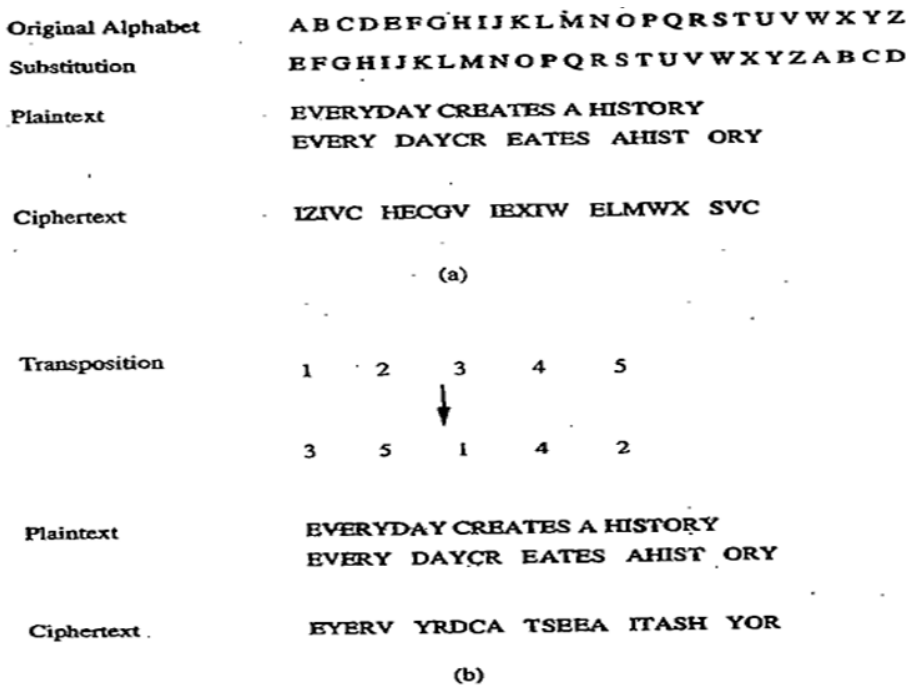


Figure 9.12. Substitution and transposition.

- Fig (a) shows encryption using substitution & fig (b) shows a transposition cipher.
- The block length used is 5.
- A stream cipher is, in effect, a block cipher of block length one.
- One of the simplest stream ciphers is vernam cipher, which uses a key of same length as plaintext for encryption.
- For example : If the plaintext is the binary string 10010100 & key is 01011001.then the encrypted string is given by the XOR of the plaintext & key, to be 11001101.
- The plaintext is again recovered by XOR-ing the cipher text with the same key.

ASYMMETRIC KEY ALGORITHMS

- Asymmetric key (or public key) algorithms use different keys at the sender end & receiver ends for encryption & decryption, respectively.
- Let the encryption process be represented by a function E, & decryption by D.
- Then plaintext 'm' is transformed into the ciphertext 'c' as

$$C = E(m)$$

- The receiver then decodes c by applying D. Hence, D is such that

$$m = D(c) = D(E(m))$$

- When this asymmetric key concept is used in public key algorithms, the key E is made public, while D is made private, known only to the intended receiver.

RSA algorithm

- RSA algorithm is the best example of public key cryptography.
- Digital signatures scheme are also based on public key encryption.
- These are called reversible public key systems
- In this case, the person who wishes to sign a document encrypts it using his/her private key D, which is known only to him/her.
- Anybody who has his/her public key E can decrypt it and obtain the original document
- A trusted third party is responsible for issuing these digital signatures and for resolving any disputes regarding the signatures
- This is usually a governmental or business organisation

KEY MANAGEMENT APPROACHES

- The primary goal of key management is to share a secret (some information) among a specified set of participants.
- The main approaches to key management are key predistribution, key transport, key arbitration and key agreement.

1. KEY PREDISTRIBUTION:

- Key predistribution, as the name suggests, involves distributing key to all interested parties before the start of communication.
- This method involves much less communication & computation, but all participants must be known *a priori*, during the initial configuration.
- Once deployed, there is no mechanism to include new members in the group or to change the key.
- As an improvement over predistribution scheme, sub-groups may be formed within a group, and some communication may be restricted to a subgroup.
- However, formation of subgroups is also an *a priori* decision.

2. KEY TRANSPORT:

- In key transport systems, one of the communicating entities generates keys & transports them to the other members.
- The simplest scheme assumes that a shared key already exists among the participating members. This shared key is used to encrypt a new key & is transmitted to all corresponding nodes.

- Only those nodes which have the prior shared key can decrypt it.
- This is called the *Key Encrypting Key (KEK) method.*
- An interesting method for key transport without prior shared keys is the shamir's three-pass protocol. The scheme is based on a special-type of encryption called communicative Encryption schemes.
- Consider 2 nodes X & Y which wish to communicate. Node X selects a key K which it wants to use in its communication with node Y.
- It then generates a random key K_X ,using which it encrypts K with f, & sends to node Y. Node Y encrypts this with a random key k_Y using g,& sends this back to node X.
- Now, node X decrypts this message with its key K_X, & after applying inverse function f^{-1},sends it to node y. finally, node Y decrypts the message using K_Y & g^{-1} to obtain key K.

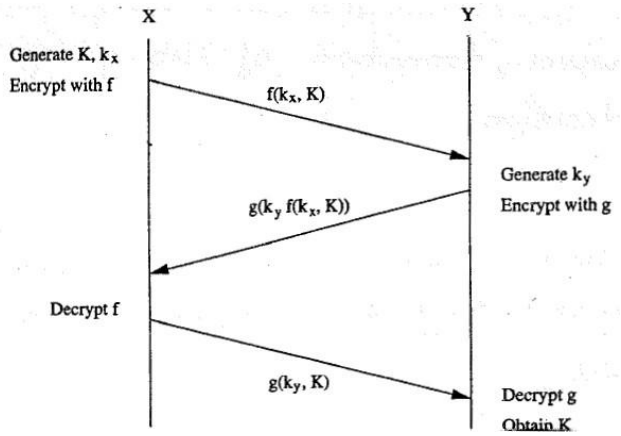


Figure: Shamir's three-pass protocol

3. KEY ARBITRATION:

- Key arbitration schemes use a central arbitrator to create & distribute keys among all participants.
- Hence, they are a class of key transport schemes.
- In ad hoc wireless networks, the problem with implementation of arbitrated protocols is that the arbitrator has to be powered on at all times to be accessible to all nodes
- This leads to a power drain on that particular node
- Alternative is to make the keying service distributed
- If any one of the replicated arbitrators is attacked, the security of the whole system breaks down

4. KEY AGREEMENT:

- Key agreement protocols are used to establish a secure context over which a session can be run, starting with many parties who wish to communicate & an insecure channel.
- In group key agreement schemes, each participant contributes a part to the secret key.

- ❖ Require least amount of preconfiguration
- ❖ Have high computational capability
- ❖ The most popular key agreement schemes use the Diffie-Hellman exchange, an asymmetric key algorithm based on discrete logarithms.

****

Key Management techniques for Ad Hoc Networks

5. What are the different Key Management techniques used for Ad Hoc Networks?

- ➢ Adhoc networks pose certain specific challenges in key management, due to the lack of infrastructure in such networks.
- ➢ 3 types of infrastructure have been identified, which are absent in adhoc networks:

  →The first is the network infrastructure, such as dedicated routers & stable links, which ensure communication with all nodes.

  →The second missing infrastructure is services, such as name resolution, directory & TTP's.

  →The third missing infrastructure is the administrative support of certifying authorities.

Password-Based Group Systems

- ➢ A long string is given as the password for users for one session.
- ➢ However, human beings tend to favour natural language phrases as passwords, over randomly generated strings.
- ➢ Such passwords, if used as keys directly during a session, are very week & open to attack directly during a high redundancy, & the possibility of reuse over different sessions.
- ➢ Hence, protocols have been proposed to derive a strong key (not vulnerable to attacks).
- ➢ This password-based system could be two-party, with a separate exchange between any 2 participants, or it could be for the whole group, with a leader being elected to preside over the session.
- ➢ The protocol used is as follows:

  →Each participant generates a random number, & sends it to all others

  →When every node has received the random number of every other node, a common pre-decided function is applied on all the numbers to calculate a reference value.

  →The nodes are ordered based on the difference between their random number & the reference value.

Threshold Cryptography

- ➢ Public Key Infrastructure(PKI) enables the easy distribution of keys & is a scalable method.

- Each node has a public/private key pair.
- A certifying authority(CA) can bind the keys to a particular node.
- But CA has to be present at all times, which may not be feasible in Adhoc networks.
- A scheme based on threshold cryptography has been proposed by which n servers exist in an adhoc network, out of which any (t+1) servers can jointly perform arbitration or authorization successfully, but t servers cannot perform the same. This is called an (n, t+1) configuration, where n >= 3t +1.
- To sign a certificate, each server generates a partial signature using its private key & submits it to a combiner. The combiner can be any one of the servers.

  →In order to ensure that the key is combined correctly, t+1 combiners can be used to account for at most t malicious servers.

  →Using t+1 partial signatures, the combiner computes a signature & verifies its validity using a public key.

  →If verification fails, it means that at least one of the t+1 keys is not valid, so another subset of t+1 partial signature is tried. If combiner itself is malicious, it cannot get a valid key, because partial key itself is always invalid.

Self-Organized Public Key Management for Mobile Adhoc Networks

- This makes use of absolutely no infrastructure.
- The users in the adhoc network issue certificates to each other based on personal acquaintance.
- A certificate is binding between a node & its public-key.
- The certificates are stored & distributed by the users themselves.
- Certificates are issued only for specific period of time, before it expires; the certificate is updated by the user who had issued the certificate.
- Each certificate is initially stored twice, by the issuer & by the person for whom it is issued.
- If any of the certificates are conflicting (e.g: the same public key to different users, or the same user having different pubic keys), it is possible that a malicious-node has issued a false certificate.
- A node then enables such certificates as conflicting & tries to resolve the conflict.
- If the certificates issued by some node are found to be wrong, then that node may be assumed to be malicious.
- A certificate graph is a graph whose vertices are public keys of some nodes and whose edges are public key certificates issued by users.

****

SECURE ROUTING

Requirements of A Secure Routing Protocol For Adhoc Wireless Networks

6. Explain the Requirements of A Secure Routing Protocol For Adhoc Wireless Networks or WSN.

The fundamental requirements for a secure routing protocol for adhoc wireless networks are listed as below:

- ***Detection of malicious nodes:***
  - A secure routing protocol should be able to detect the presence of any malicious node in the network & should avoid the participation of such nodes in the routing process.
- ***Guarantee of correct route discovery:***
  - If a route between the source & destination node exist, the routing protocol should be able to find the route, & should also ensure the correctness of the selected route.
- ***Confidentiality of network topology:***
  - Once the network topology is known, the attacker may try to study the traffic pattern in the network. If some of the nodes are found to be more active compared to others, the attacker may try to mount attacks.
  - This may ultimately affect the ongoing routing process. Hence, confidentiality of network topology is important.
- ***Stability against attacks:***
  - The routing protocols must be self-stable in the sense that it must be able to revert to its normal operating state within a finite amount of time after passive or an active attack.
  - Some of the security-aware routing protocols proposed for adhoc wireless networks are discussed.

****

SECURE ROUTING-SPINS

Explain about secure routing- SPINS.

- Security protocols for sensor networks (SPINS) consists of a suite of security protocols that are optimized for highly resource constrained sensor networks.
- SPINS consists of two main modules:
  - Sensor Network Encrption Protocol (SNEP)
  - a micro version of timed, efficient, streaming , loss- tolerant authentication protocol (µ TESLA).
- SNEP provides data authentication, protection from replay attacks and semantic security, all with low communication overhead of eight bytes per message.

- Semantic Security means that an adversary cannot get any idea about the plaintext even by seeing multiple encrypted versions of the same plaintext.
- Encryption of the plaintext uses a shared counter (shared between sender and receiver).
- Hence the same message is encrypted differently at different instances in time.
- Message integrity and confidentiality are maintained using a message authentication code (MAC).
- This is similar to a check sum derived by applying an authentication scheme with a secret shared key to the message.
- This message can e decrypted only if the same shared key is present.
- The message also carries the counter value at the instance of transmission (like a time stamp), to protect against replay attacks.
- µ TESLA ensures an authenticated broadcast, that is, nodes which receive a packet can be assured of its senders identity.
- It requires a loose time synchronization between BS and nodes, with an upper bound on maximum synchronization error.
- The MAC keys are derived from a chain of keys, obtained by applying a one way function F (a one way function is one whose inverse is not easily computable).
- All nodes have an initial key K_0 , which is some key in the key chain.
- The relationship between the keys proceeds as $K_0 = F(K_1)$, $K_1 = F(K_2)$, and in general,

$$K_i = F(K_i + 1).$$

- Given $K_0, K_1, \ldots K_i$, it is not possible to compute $K_i + 1$.
- The key to be used changes periodically , and since nodes are synchronized to a common time within a bounded error, they can detect which key is to be used to encrypt/ decrypt a packet at any time instant.
- The BS periodically discloses the next verification key to all the nodes and this period is known to all nodes.
- There is also a specified lag of certain intervals between the usage of a key for encryption and its disclosure to all receivers.
- When the BS transmits a packet, It uses a MAC key which is still secret.
- The nodes which receive this packet buffer it until the appropriate verification key is disclosed.
- But, as soon as a packet is received, the MAC is checked to ensure that the key used in the MAC has not yet been disclosed which implies that only the BS which knows that yet disclosed key could have sent the packet.
- The packets are decrypted once the key disclosure packet is received from the BS.

- If one of the key disclosure packets is missed, the data packets is received from the next time interval, and then authenticated.
- For instance, suppose the disclosure packet of K_j does not reach a node.
- It waits till it receives $K_j = F(K_j + 1)$ and decrypts the packets received in the previous time interval.

****

TWO MARKS

1. **What are the requirements of network security?**
   - ✓ Confidentiality
   - ✓ Integrity
   - ✓ Availability
   - ✓ Non-repudiation

2. **List the issues and challenges in security provisioning:-**
   - ✓ Shared broadcast radio channel
   - ✓ Insecure operational environment
   - ✓ Lack of central authority
   - ✓ Lack of association
   - ✓ Limited resource availability
   - ✓ Physical vulnerability

3. **What are the two types of security attacks?**
   - ✓ Active attack
   - ✓ Passive attack

4. **List the types of attacks are included in active attacks:-**
   - ✓ MAC layer attacks
   - ✓ Network layer attacks
   - ✓ Transport layer attacks
   - ✓ Application layer attacks
   - ✓ Other attacks

5. **What is an active and passive attack of network security?**

Passive attacks:

Advisory snoops the data exchanged the network without altering it.

Active attacks:

It alters or destroys the data being exchanged in the network thereby disrupting the normal functions of the network.

6. What is difference between wormhole and black hole attacks?

Wormhole attacks	Blackhole attacks
An attacker receives packet at one location in the network tunnels them to another location in the network. This tunneling between two colliding attackers is referred as wormhole attacks	A malicious node falsely advertises good paths to the destination node during the path finding process (or)in the route update messages.

7. What is resource consumption attack?

- ✓ A malicious node tries to consume waste away resource of other nodes present in the network.
- ✓ The resources that are targeted are battery power, bandwidth and computational power which are only limitedly available in adhoc wireless networks.

8. What is the term of session hijacking?

An adversary takes control over a session between two nodes. Since most authentication process are carried out only at the start of the session, once the session between two nodes gets established, the adversary node masquerades as one of the end nodes of the session and hijacks the session.

9. What is repudiation in application layer attacks in network security?

Repudiation refers to the denial (or) attempted denial by a node involved in a communication of having participated in all parts of the communication.

10. What are the various attacks are involved in the routing attacks?

- ✓ Routing table overflow
- ✓ Routing table poisoning
- ✓ Packet replication
- ✓ Route cache poisoning
- ✓ Rushing attacks

11. What is denial of service attacks?

An adversary attempts to prevent legitimate and authorized users of services offered by the network from processing those services.

12. Some of the DOS attacks are included in the network security.

- ✓ Jamming
- ✓ SYN flooding
- ✓ Distributed DOS attacks.

13. What are the requirements of a secure routing protocol for adhoc wireless networks?

- ✓ Detection of malicious nodes
- ✓ Guarantee of correct route discovery
- ✓ Confidentiality of network topology
- ✓ Stability against attacks.

14. What is blackhole attacks and how to overcome it?

A malicious intermediate node could advertise that enhance the shortest path to the destination, thereby redirecting all the packets through itself. This is called blackhole attacks.

15. How to overcome blackhole attacks?

To restrict the intermediate nodes from originating route reply packets. Only the destination node would be permitted to initiate route reply packets. Security is still not completely assured, since the malicious node may lie in the path chosen by the destination node.

16. What is Byzantine attack? (May/June 2012)

A compromised intermediate node could create routing loops and leads to wastage of power and bandwidth.

17. What is Distributed DoS attack?

Several adversaries that are distributed throughout the network collide and prevent legitimate users from accessing the services offered by the network.

18. What is SYN flooding?

An adversary sends a large number of SYN packets to a victim node, spoofing the return addresses of the SYN packets. The victim node builds up a table/data structure for holding information regarding all pending connections.

19. What is jamming? How to overcome it?

The adversary initially keeps monitoring the wireless medium in order to determine the frequency at which the receiver node is receiving signals from the sender. Frequency hopping spread spectrum(FHSS) and direct sequence spread spectrum (DSSS) are two commonly used techniques that overcome jamming attacks

20. What is Routing table poisoning attack?

The compromised nodes in the networks send fictitious routing updates or modify genuine route update packets sent to other uncompromised nodes.

21. What is wormhole attack?

An attacker receives packets at one location in the network & tunnels them (possibly selectively) to another location in the network, where the packets are resent into the network. This tunnel between 2 colliding attackers is referred to as a wormhole.

22. What are the different attacks possible over adhoc networks?

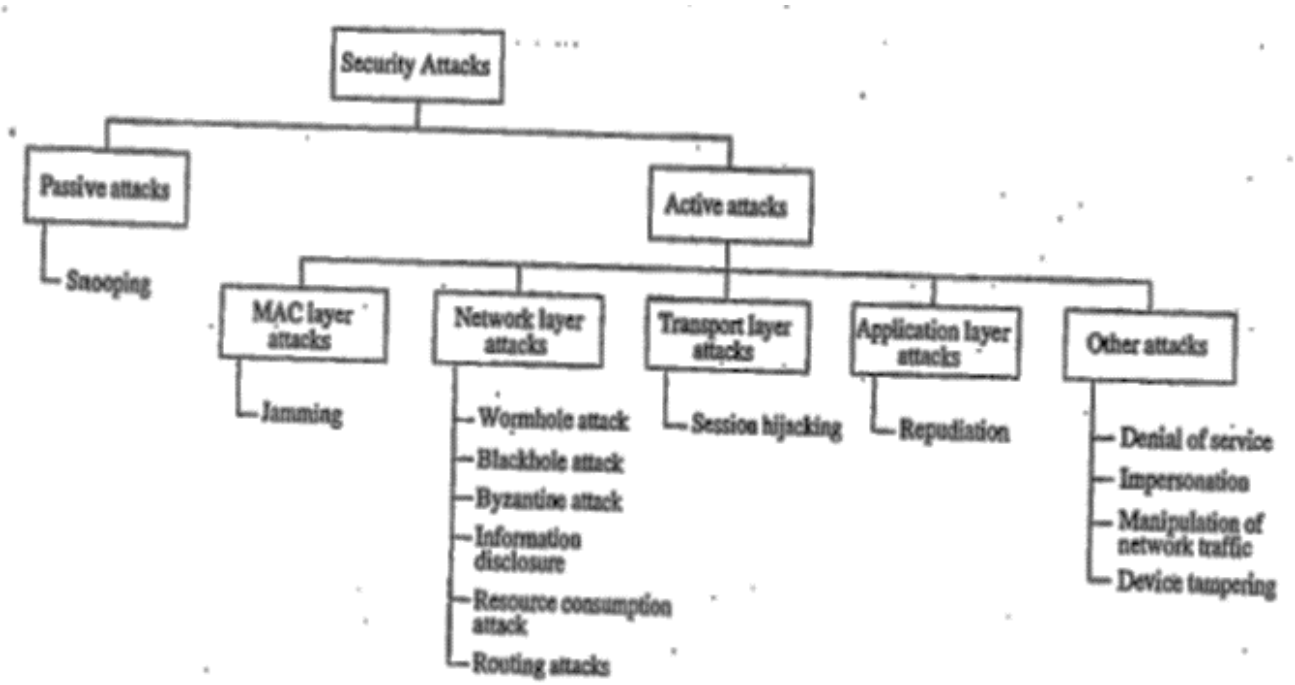


Figure 9.11. Classifications of attacks.

23. What is Symmetric and Asymmetric key algorithms?

1. ***Symmetric*** *key algorithms,* which use the same key for encryption & decryption.
2. ***Asymmetric*** *key algorithms,* which use two different keys for encryption & decryption.

24. What is key management?

The secure administration of cryptographic keys[private,public] is called as key manangement.

25. What are the two major kinds of cryptographic algorithms?

- symmetric key algorithms
- asymmetric key algorithms.

26. what are various approaches in key management?

- Key predistribution
- Key transport
- Key arbitration
- Key agreement

27. What are the requirements of a secure routing protocol for adhoc wireless networks?

- Detection of malicious nodes
- Guarantee of correct route discovery
- Confidentiality of network topology
- Stability against attacks.

28. What are the modules in SPINS protocol?

SPINS consists of two main modules:

- Sensor Network Encrption Protocol (SNEP)
- a micro version of timed, efficient, streaming , loss- tolerant authentication protocol (μ TESLA).

UNIT – 5 SENSOR NETWORK PLATFORMS AND TOOLS

Sensor Node Hardware – Berkeley Motes, Programming Challenges, Node-level software platforms – TinyOS, nesC, CONTIKIOS, Node-level Simulators – NS2 and its extension to sensor networks, COOJA, TOSSIM, Programming beyond individual nodes – State centric programming.

TABLE OF CONTENTS

5.1 Sensor Node Hardware

- Sensor node hardware can be grouped into three categories, each of which entails a different trade-offs in the design choices.
  - Augmented general-purpose computers
  - Dedicated embedded sensor nodes
  - System on-chip (SoC) nodes

5.1.1 Augmented general-purpose computers

- These nodes typically run off-the-shelf operating systems such as WinCE, Linux, or real-time operating systems and use standard wireless communication protocols such as IEEE 802.11, Bluetooth, Zigbee etc.

- Because of their relatively higher processing capability, they can accommodate wide variety of sensors, ranging from simple microphones to more sophisticated video cameras. It is fully supported for popular programming languages.
- Examples include low-power PCs, embedded PCs (e.g. PC104), custom-designed PCs, (e.g. Sensoria WINS NG nodes), and various personal digital assistants (PDA).

5.1.2 Dedicated embedded sensor nodes

- These platforms typically use commercial off-the-shelf (COTS) chip sets with emphasis on small form factor, low power processing and communication, and simple sensor interfaces.

- Because of their COTS CPU, these platforms typically support at least one programming language, such as C. However, in order to keep the program footprint small to accommodate their small memory size, programmers of these platforms are given full access to hardware but rarely any operating system support.
- Examples include the Berkeley mote family, the UCLA Medusa family, Ember nodes and MIT µAMP. A classical example is the TinyOS platform and its companion programming language, nesC, mica.

5.1.3 System on-chip (SoC) nodes

- These platforms try to push the hardware limits by fundamentally rethinking the hardware architecture trade-offs for a sensor node at the chip design level.
- The goal is to find new ways of integrating CMOS, MEMS, and RF technologies to build extremely low power and small footprint sensor nodes that still provide certain sensing, computation, and communication capabilities.
- Examples of SoC hardware include smart dust the BWRC picoradio node, and the PASTA node.

5.2 Berkeley Motes

- Berkeley Mote platform as it is an open hardware/software, smart-sensing platform with a large user community.
- The Berkeley Mote platform was developed under the Networked Embedded Systems Technology (NEST) program with the quantitative target of building dependable, real-time, distributed, embedded applications comprising 100 to 100 000 simple computing nodes.
- Berkeley motes tiny, self-contained, battery powered computers with radio links, which enable to communicate and exchange data with one other, and to self- organize into ad hoc networks.
- Motes form the building blocks of wireless sensor networks.
- The platform consists of four basic components: Power, sensors, computation, and communication. These motes are autonomous and connectable to other motes.
- The main advantages are small physical size, low cost, modest power consumption, and diversity in design and usage. The latest versions of the Berkeley Mote include the MicaZ, Mica2, and Mica2dot processor boards (Fig. 5.1). The Motes have

improvements in memory and radio over predecessors and specifications are summarised in Table 5.1. The same sensor board can be also used for the MicaZ and modified for use with the Mica2dot

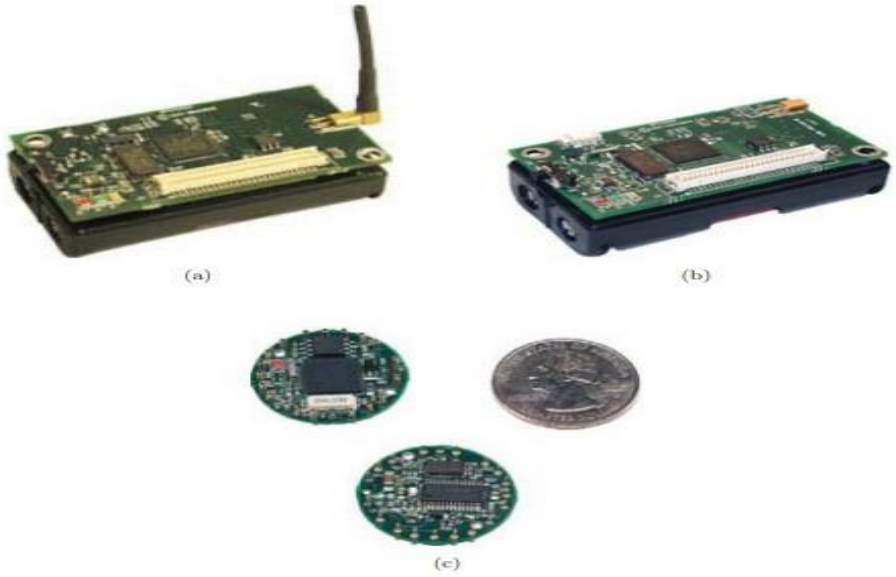


Figure 5.1 Berkeley Mote processor boards: (a) MicaZ, (b) Mica2, and (c) Mica2dot.
Source : Protocol and Architecture for Wireless Sensor Networks by Holger Karl , Andreas willig

Table 5.1 Characteristics of the MicaZ, Mica2, and Mica2dot processor boards

	MicaZ	Mica2	Mica2dot
Flash memory	128 K bytes	128 K bytes	128 K bytes
Measurement memory	512 K bytes	512 K bytes	512 K bytes
EEPROM	4 K bytes	4 K bytes	4 K bytes
A/D (Channels)	10 bits (8)	10 bits (8)	10 bits (6)
Frequency	1400 MHz–2483.5 MHz	433/868/916 MHz	433/868/916 MHz
Data rate	250 K bps	19.2 K bps	19.2 K bps
Outdoor range	100 m	300 m	300 m
Size	6×3×1 cm	6×3×1 cm	2.5×0.6 cm

- The Motes have the versatility to connect different Printed Circuit Boards (PCB); that is, the Motes have the modularity to support different types of sensors. Users can switch the sensor board or customise it, independent of the other hardware components. Table 5. 2 provides a description of the sensors that are used on the respective boards.

- The current sensor board designs for the Mica2 platform includes: MTS101CA, MTS300CA, MTS310CA, MTS400CA, and MTS420CA

Table 5. 2. Description of the available sensor boards

Name	Sensors
MTS101CA	Photo resistor and thermistor.
MTS300CA	Photo resistor, thermistor, acoustic sensor and acoustic actuator.
MTS310CA	Photo resistor, thermistor, acoustic sensor, acoustic actuator, biaxial accelerometer and magnetometer.
MTS400/420CA	Thermistor, hygrometer, barometer, Photo resister, accelerometer, GPS(only on MTS420CA)

5.3 Sensor Network Programming Challenges

- Traditional programming technologies rely on operating systems to provide abstraction for processing, I/O, networking, and user interaction hardware. When applying such a model to programming networked embedded systems, such as sensor networks, the application programmers need to explicitly deal with message passing, event synchronization, interrupt handling, and sensor reading.
- As a result, an application is typically implemented as a finite state machine (FSM) that covers all extreme cases: unreliable communication channels, long delays, irregular arrival of messages, simultaneous events etc.
- For resource-constrained embedded systems with real-time requirements, several mechanisms are used in embedded operating systems to reduce code size, improve response time, and reduce energy consumption.
- The microkernel technologies modularize the operating system so that only the necessary parts are deployed with the application. Real-time scheduling allocates resources to more urgent tasks so that they can be finished early.
- Event-driven execution allows the system to fall into low-power sleep mode when no interesting events need to be processed.
- At the extreme, embedded operating systems tend to expose more hardware controls to the programmers, who now have to directly face device drivers and scheduling algorithms, and optimize code at the assembly level.
- Although these techniques may work well for small, stand-alone embedded systems, they do not scale up for the programming of sensor networks for two reasons:

- **Sensor networks are large-scale distributed systems**, where global properties are derivable from program execution in a massive number of distributed nodes. Distributed algorithms themselves are hard to implement, especially when infrastructure support is limited due to the ad hoc formation of the system and constrained power, memory, and bandwidth resources.

- **As sensor nodes deeply embed into the physical world,** a sensor network should be able to respond to multiple concurrent stimuli at the speed of changes of the physical phenomena of interest.

➢ There no single universal design methodology for all applications. Depending on the specific tasks of a sensor network and the way the sensor nodes are organized, certain methodologies and platforms may be better choices than others.

➢ For example, if the network is used for monitoring a small set of phenomena and the sensor nodes are organized in a simple star topology, then a client-server software model would be sufficient.

➢ If the network is used for monitoring a large area from a single access point (i.e., the base station), and if user queries can be decoupled into aggregations of sensor readings from a subset of nodes, then a tree structure that is rooted at the base station is a better choice. However, if the phenomena to be monitored are moving targets, as in the target tracking, then neither the simple client-server model nor the tree organization is optimal. More sophisticated design and methodologies and platforms are required.

5.4 Node-Level Software Platforms

➢ Most design methodologies for sensor network software are node-centric, where programmers think in terms of how a node should behave in the environment.

➢ A node level platform can be node-centric operating system, which provides hardware and networking abstractions of a sensor node to programmers, or it can be a language platform, which provides a library of components to programmers.

➢ A typical operating system abstracts the hardware platform by providing a set of services for applications, including file management, memory allocation, task scheduling, peripheral device drivers, and networking.

➢ For embedded systems, due to their highly specialized applications and limited resources, their operating systems make different trade-offs when providing these services.

- For example, if there is no file management requirement, then a file system is obviously not needed. If there is no dynamic memory allocation, then memory management can be simplified. If prioritization among tasks is critical, then a more elaborate priority scheduling mechanism may be added.

5.5 Operating System Design Issues

- Traditional operating systems are system software, including programs that manage computing resources, control peripheral devices, and provide software abstraction to the application software.

- Traditional OS functions are therefore to manage processes, memory, CPU time, file system, and devices. This is often implemented in a modular and layered fashion, including a lower layer of kernels and a higher layer of system libraries.

- Traditional OSs are not suitable for wireless sensor networks because WSNs have constrained resources and diverse data-centric applications, in addition to a variable topology.

- Hence, WSNs need a new type of operating system, considering their special characteristics. There are several issues to consider when designing operating systems for wireless sensor networks.

- The **first issue** is process management and scheduling. The traditional OS provides process protection by allocating a separate memory space (stack) for each process. Each process maintains data and information in its own space. But this approach usually causes multiple data copying and context switching between processes. This is obviously not energy efficient for WSNs. For some real-time applications in WSNs, a real-time scheduler such as earliest deadline first (EDF) or its variants may be a good choice, but the number of processes should be confined since that would determine the time complexity of the EDF scheduler.

- The **second issue** is memory management. Memory is often allocated exclusively for each process/task in traditional operating systems, which is helpful for protection and security of the tasks. Since sensor nodes have small memory, another approach, sharing, can reduce memory requirements.

- The **third issue** is the kernel model. The event-driven and finite state machine(FSM) models have been used to design microkernels for WSNs. The event-driven model may serve WSNs well because they look like event-driven systems. An event may comprise receiving a packet, transmitting a packet, detection of an event of interest,

alarms about energy depletion of a sensor node, and so on. The FSM-based model is convenient to realize concurrency, reactivity, and synchronization.

- The **fourth issue** is the application program interface (API). Sensor nodes need to provide modular and general APIs for their applications. The APIs should enable applications access the underlying hardware.

- The **fifth issue** is code upgrade and reprogramming. Since the behavior of sensor nodes and their algorithms may need to be adjusted either for their functionality or for energy conservation, the operating system should be able to reprogram and upgrade.

- **Finally,** because sensor nodes generally have no external disk, the operating system for WSNs cannot have a file system. These issues should be considered carefully in the design of WSN OSs and to meet their constrained resources, network behavior, and data-centric application requirements.

- Sensor operating systems (SOS) should represent the following functions, bearing in mind the limited resource of sensor nodes:

  - Should be compact and small in size since the sensor nodes have very small memory. The sensor nodes often have memories of only tens or hundreds of kilobytes.

  - Should provide real-time support, since there are real-time applications, especially when actuators are involved. The information received may become outdated rather quickly. Therefore, information should be collected and reported as quickly as possible.

  - Should provide efficient resource management mechanisms in order to allocate microprocessor time and limited memory. The CPU time and limited memory must be scheduled and allocated for processes carefully to guarantee fairness (or priority if required).

  - Should support reliable and efficient code distribution since the functionality performed by the sensor nodes may need to be changed after deployment. The code distribution must keep WSNs running normally and use as little wireless bandwidth as possible.

  - Should support power management, which helps to extend the system lifetime and improve its performance. For example, the operating system may schedule the process to sleep when the system is idle, and to wake up with the advent of an incoming event or an interrupt from the hardware.

- Should provide a generic programming interface up to sensor middleware or application software. This may allow access and control of hardware directly, to optimize system performance.

5.6 Operating System: TinyOS

- The design of TinyOS allows application software to access hardware directly when required. TinyOS is a tiny micro threaded OS that attempts to address two issues:
  - How to guarantee concurrent data flows among hardware devices, and
  - How to provide modularized components with little processing and storage overhead.
- These issues are important since TinyOS is required to manage hardware capabilities and resources effectively while supporting concurrent operation in an efficient manner.
- TinyOS uses an event-based model to support high levels of concurrent application in a very small amount of memory. Compared with a stack-based threaded approach, which would require that stack space be reserved for each execution context, and because the switching rate of execution context is slower than in an event-based approach, TinyOS achieves higher throughput.
- It can rapidly create tasks associated with an event, with no blocking or polling. When CPU is idle, the process is maintained in a sleep state to conserve energy. TinyOS includes a tiny scheduler and a set of components. The scheduler schedules operation of those components.
- Each component consists of four parts: command handlers, event handlers, an encapsulated fixed-size frame, and a group of tasks
- Commands and tasks are executed in the context of the frame and operate on its state. Each component will declare its commands and events to enable modularity and easy interaction with other components.
- The current task scheduler in TinyOS is a simple FIFO mechanism whose scheduling data structure is very small, but it is power efficient since it allows a processor to sleep when the task queue is empty and while the peripheral devices are still running. The frame is fixed in size and is assigned statically. It specifies the memory requirements of a component at compile time and removes the overhead from dynamic assignment. Commands are non-blocking requests made to the low-level components. Therefore, commands do not have to wait a long time to be executed.

- A command provides feedback by returning status indicating whether it was successful (e.g., in the case of buffer overrun or of timeout). A command often stores request parameters into its frame and conditionally assigns a task for later execution.
- The occurrence of a hardware event will invoke event handlers. An event handler can store information in its frame, assign tasks, and issue high-level events or call low-level commands. Both commands and events can be used to perform a small and usually fixed amount of work as well as to pre-empt tasks.
- Tasks are a major part of components. Like events, tasks can call low-level commands, issue high-level events, and assign other tasks. Through groups of tasks, TinyOS can realize arbitrary computation in an event-based model.
- The design of components makes it easy to connect various components in the form of function calls. The architecture of TinyOS shown in Figure 5.2.
- This WNS operating system defines three type of components:
  - Hardware abstractions
  - Synthetic hardware
  - High-level software components
- **Hardware abstraction components** are the lowest-level components. They are actually the mapping of physical hardware such as I/O devices, a radio transceiver, and sensors. Each component is mapped to a certain hardware abstraction.
- **Synthetic hardware components** are used to map the behavior of advanced hardware and often sit on the hardware abstraction components. TinyOS designs a hardware abstract component called the radio-frequency module (RFM) for the radio transceiver, and a synthetic hardware component called radio byte, which handles data into or out of the underlying RFM.
- **Higher-level components** encapsulate software functionality, but with a similar abstraction. They provide commands, signal events, and have internal handlers, task threads, and state variables

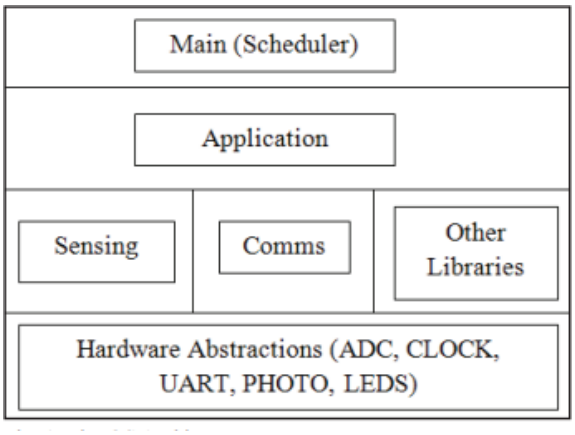


Figure 5.2 TinyOS Architecture

Source : Protocol and Architecture for Wireless Sensor Networks by Holger Karl , Andreas willig

- An evaluation of TinyOS shows that it achieves the following performance gains or advantages:
  - It requires very little code and a small amount of data.
  - Events are propagated quickly and the rate of posting a task and switching the corresponding context is very high.
  - It enjoys efficient modularity.

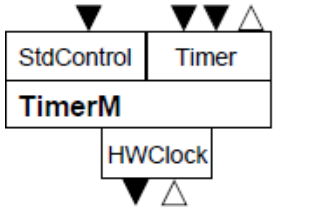
5.7 nesC

- nesC is a component-based, event-driven programming language used to build applications for the TinyOS platform. TinyOS is an operating environment designed to run on embedded devices used in distributed wireless sensor networks.
- The name nesC is an abbreviation of "network embedded systems C". nesC is an extension of C.
- nesC programs are subject to whole program analysis (for safety) and optimization (for performance). Therefore we do not consider separate compilation in nesC's design. The limited program size on motes makes this approach tractable.
- nesC is a "static language". There is no dynamic memory allocation and the call-graph is fully known at compile-time. These restrictions make whole program analysis and optimization significantly simpler and more accurate. nesC's component model and parameterized interfaces eliminate many needs for dynamic memory allocation and dynamic dispatch.

- nesC is based on the concept of components, and directly supports TinyOS's event based concurrency model. Additionally, nesC explicitly addresses the issue of concurrent access to shared data. In practice, nesC resolved many ambiguities in the TinyOS concepts of components and concurrency.

Component Specification

- nesC applications are built by writing and assembling components. A component provides and uses interfaces. These interfaces are the only point of access to the component. An interface generally models some service (e.g., sending a message) and is specified by an interface type. Figure 5.3 shows the TimerM component, part of the TinyOS timer service that provides the StdControl and Timer interfaces and uses a Clock interface (all shown in Figure 5.4).




```
module TimerM {
  provides {
    interface StdControl;
    interface Timer;
  }
  uses interface Clock as Clk;
} ...
```

Figure 5.3 Specification and graphical depiction of the TimerM component

Source : Protocol and Architecture for Wireless Sensor Networks by Holger Karl , Andreas willig

```
interface StdControl {
  command result_t init();
}

interface Timer {
  command result_t start(char type, uint32_t interval);
  command result_t stop();
  event result_t fired();
}

interface Clock {
  command result_t setRate(char interval, char scale);
  event result_t fire();
}

interface Send {
  command result_t send(TOS_Msg *msg, uint16_t length);
  event result_t sendDone(TOS_Msg *msg, result_t success);
}

interface ADC {
  command result_t getData();
  event result_t dataReady(uint16_t data);
}
```

Figure 5.4 Some Interface Types

- TimerM provides the logic that maps from a hardware clock (Clock) into TinyOS's timer abstraction (Timer).
- Interfaces in nesC are bidirectional. They contain commands and events, both of which are essentially functions. The providers or an interface implement the commands, while the users implements the events. For instance, the Timer interface (Figure 5.4) defines start and stop commands and a fired event.
- In Figure 5.3 provided interfaces are shown above the TimerM component and used interfaces are below; downward-pointing arrows depict commands and upward-pointing arrows depict events. Although this same interaction between the timer and its client could have been provided via two separate interfaces (one for start and stop, and one for fired), grouping these commands and events in the same interface makes the specification much clearer and helps prevent bugs when wiring components together.
- Split-phase operations are cleanly modelled by placing the command request and event response in the same interface. Figure 5.4 shows two examples of this.
- The Send interface has the send command and sendDone event of the split-phased packet send. The ADC interface is similarly used to model split-phase sensor value reads. The separation of interface type definitions from their use in components promotes the definition of standard interfaces, making components more reusable and flexible.
- A component can provide and use the same interface type (e.g., when interposing a component between a client and service), or provide the same interface multiple times. In these cases, the component must give each interface instance a separate name using the as notation shown for Clk in Figure 5.3.
- The components are also a clean way to abstract the boundary between hardware and software. For instance, on one sensor board, the temperature sensor (accessed via a component named Temp) is mostly in hardware; Temp is a thin layer of software accessing on-chip hardware registers.

Component Implementation

- There are two types of components in nesC: modules and configurations. Modules provide application code, implementing one or more interfaces. Configurations are used to wire other components together, connecting interfaces used by components to interfaces provided by others.

Concurrency and Atomicity

- nesC detects the data races at compile time. Data races occur due to concurrent updates to shared state. In order to prevent them, a compiler must
  - Understand the concurrency model,
  - Determine the target of every update

5.8 ContikiOS

- ContikiOS is open source operating system for resource constraint hardware devices with low power and less memory. It was developed by Adam Dunkels in 2002. This OS is fully GUI based system requires only 30 KB ROM and 10 KB RAM. It also provide multitasking feature and have the built in TCP/IP suit.
- The working environments of the WSNs are often energy-limited. This is one of the most important constraint for WSNs. Likewise, tiny and simple designs of the nodes are the other constraints. For this reason, WSNs should have some important hardware and software features to cope with these constraints.
- Contiki OS is one of the convenient solutions to cope with mentioned constraints to its flexibility and support of lightweight and low-powered networks.
- Contiki can provide communication over IPv4, IPv6 and Rime Network Stack. Contiki Network Stack shown in Figure 5.5 gives more details for its structure.

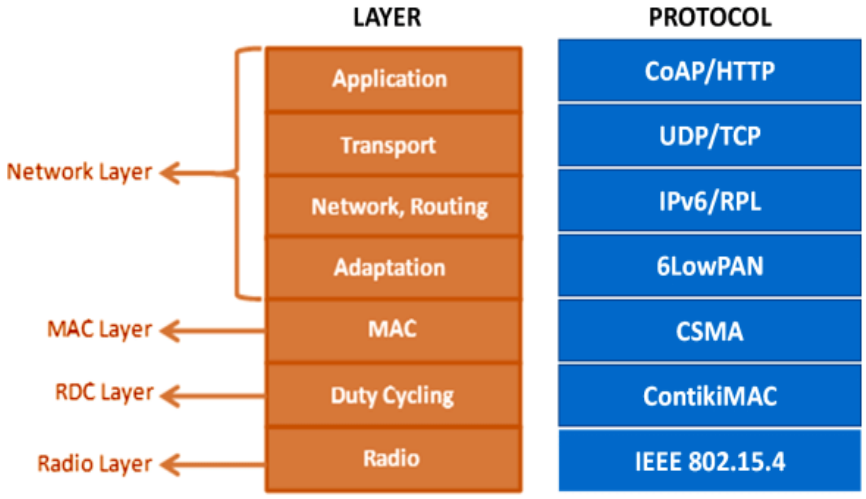


Figure 5.5 Contiki Network Stack

Source : Protocol and Architecture for Wireless Sensor Networks by Holger Karl , Andreas willig

- Many Contiki systems are severely power-constrained. Battery operated wireless sensors may need to provide years of unattended operation and with little means to recharge or replace batteries.

- Contiki provides a set of mechanisms to reduce the power consumption of systems on which it runs. The default mechanism for attaining low-power operation of the radio is called ContikiMAC. With ContikiMAC, nodes can be running in low-power mode and still be able to receive and relay radio messages.

- The Contiki programming model is based on protothreads. A protothread is a memory-efficient programming abstraction that shares features of both multithreading and event-driven programming to attain a low memory overhead of each protothread.

- The kernel invokes the protothread of a process in response to an internal or external event. Examples of internal events are timers that fire or messages being posted from other processes. Examples of external events are sensors that trigger or incoming packets from a radio neighbour.

5.9 Node-Level Simulators

- Node-level design methodologies are usually associated with simulators that simulate the behavior of a sensor network on a per-node basis. Using simulation, designers can quickly study the performance (in terms of timing, power, bandwidth, and scalability) of potential algorithms without implementing them on actual hardware and dealing with the vagaries of actual physical phenomena. A node-level simulator typically has the following components:

  - **Sensor node model:** A node in a simulator acts as a software execution platform, a sensor host, as well as a communication terminal. In order for designers to focus on the application-level code, a node model typically provides or simulates a communication protocol stack, sensor behaviours (e.g., sensing noise), and operating system services. If the nodes are mobile, then the positions and motion properties of the nodes need to be modelled. If energy characteristics are part of the design considerations, then the power consumption of the nodes needs to be modelled.

  - **Communication model:** Depending on the details of modelling, communication may be captured at different layers. The most elaborate simulators model the communication media at the physical layer, simulating the RF propagation delay and collision of simultaneous transmissions. Alternately, the communication may

be simulated at the MAC layer or network layer, using, for example, stochastic processes to represent low-level behaviours.

- **Physical environment model:** A key element of the environment within a sensor network operates is the physical phenomenon of interest. The environment can also be simulated at various levels of details. For example, a moving object in the physical world may be abstracted into a point signal source. The motion of the point signal source may be modelled by differential equations or interpolated from a trajectory profile.

- **Statistics and visualization:** The simulation results need to be collected for analysis. Since the goal of a simulation is typically to derive global properties from the execution of individual nodes, visualizing global behaviours is extremely important. An ideal visualization tool should allow users to easily observe on demand the spatial distribution and mobility of the nodes, the connectivity among nodes, link qualities, end-to-end communication routes and delays, phenomena and their spatio-temporal dynamics, sensor readings on each node, sensor nodes states, and node lifetime parameters (e.g., battery power).

➢ A sensor network simulator simulates the behavior of a subset of the sensor nodes with respect to time. Depending on how the time is advanced in the simulation, there are two types of execution models:

  - Cycle-Driven Simulation
  - Discrete-Event Simulation

5.9.1 Cycle-Driven Simulation

- A cycle-driven (CD) simulation discretizes the continuous notion of real time into (typically regularly spaced) ticks and simulates the system behavior at these ticks. At each tick, the physical phenomena are first simulated, and then all nodes are checked to see if they have anything to sense, process, or communicate.

- Sensing and computation are assumed to be finished before the next tick. Sending a packet is also assumed to be completed by then. However, the packet will not be available for the destination node until next tick. This split-phase communication is a key mechanism to reduce cyclic dependencies that may occur in cycle-driven simulations.

5.9.2 Discrete-Event Simulation

- A Discrete-Event (DE) simulator assumes that the time is continuous and an event may occur at any time. As event is 2-tuple with a value and a time stamp indicating when the event is supposed to be handled. Components in a DE simulation react to input events and produce output events. In node-level simulators, a component can be a sensor node, and the events can be communication packets; or a component can be software module within and the events can be message passing among these nodes.

- Typically, components are causal, in the sense that if an output event is computed from an input event, then the time stamp of the output should not be earlier than that of the input event. Non-causal components require the simulators to be able to roll back in time, and worse, they may not define a deterministic behavior of a system.

- A DE simulator typically requires a global event queue. All events passing between nodes or modules are put in the event queue and sorted according to their chronological order. At each iteration of the simulation, the simulator removes the first event (the one with earliest time stamp) from the queue and triggers the component that reacts to that event.

➢ In terms of timing behaviour, a DE simulator is more accurate than a CD simulator, and as a consequence, DE simulators run slower. The overhead of ordering all events and computation, in addition to the values and time stamps of events, usually dominates the computation time.

➢ CD simulations usually require less complex components and give faster simulations. DE simulations are sometimes considered as good as actual implementations, because of their continuous notion of time and discrete notion of events.

➢ There are several open source or commercial simulators available. One class of these simulators comprises extensions of classical network simulators, such as ns-2, J-Sim (previously known as JavaSim), and GloMoSim/ Qualnet. The focus of these simulators is on network modelling, protocol stacks, and simulation performance.

➢ Another class of simulators, sometimes called software-in-the-loop simulators, incorporate the actual node software into the simulation. For this reason, they are typically attached to particular hardware platforms and are less portable. Example include TOSSIM for Berkeley motes.

5.10 NS2 and its Extension to Sensor Networks

- The NS-2 (Network Simulator-2) is a well-known network simulator for discrete event simulation. Simulations are based on a combination of C++ and OTcl.
- NS-2 includes a large number of simulated network protocols and tools used for simulating transport control protocol (TCP), routing algorithm, multicast protocol over the wired or wireless (local connection or via satellite connection) networks.
- NS-2 is committed to OSI model simulation, including the behaviour of physical layer and it is a free open source software and available for free download.

Limitations of NS-2

- It puts some restrictions on the customisation of packet formats, energy models, MAC protocols, and the sensing hardware models, which limits its flexibility.
- The lack of an application model makes it ineffective in environments that require interaction between applications and the network protocols.
- It does not run real hardware code.
- It has been built by many developers and contains several inherent known and unknown bugs.
- It does not scale well for WSNs due to its object-oriented design.
- Using C++ code and OTcl scripts make it difficult to use.

- Actually, NS-2 was not initially designed to simulate wireless sensor network, but a few research groups had extended NS-2 in order to enable it to support wireless sensor network simulation, including sensor model, battery model, a small stack, and hybrid simulation tools.
- It is extensible, but not very scalable because of the split programming model and object-oriented structure. In addition, because NS-2 can simulate very detailed data packet close to the exact number of running packets, it is unable to carry out large-scale network simulation.
- To overcome the above drawbacks the improved NS-3 simulator was developed. NS-3 supports simulation and emulation. It is totally written in C++, while users can use python scripts to define simulations.

- Hence, transferring NS-2 implementation to NS-3 require manual intervention. Besides the scalability and performance improvements, simulation nodes have the ability to support multiple radio interfaces and multiple channels.
- Furthermore, NS-3 supports a real-time schedule that makes it possible to interact with real systems. For example, a real network device can emit and receive NS-3 generated packets.

5.11 COOJA

- Cooja simulator is the efficient simulate wireless sensor networks. Cooja is the default simulator of Contiki operating system that helps to simulate the wireless sensor networks in addition it helps to do the performance evolution.
- Contiki is a light weight operating system that is developed mainly for wireless nodes. The motes that are developed by the contiki offers many advantages.
- Contiki offers a java based simulator called as cooja which is used to simulate the wireless sensors. Cooja simulator is more flexible so that many parts of the simulator is replaceable and extendable. The parts of the simulator like simulated node hardware, plug-ins and radio medium can be replaceable.

Characteristics of Cooja

- Scalability
- Efficiency
- Extensibility
- Flexibility

- Wireless sensor network has the powerful tool called tool in which it can be simulate the idea before it is implementing in real time. Contiki Cooja WSN Simulator mainly used to simulate many wireless scenario.

Contiki Cooja WSN simulator

- Contiki cooja is the best simulator to simulate any wireless sensors with its own property. For example, if we are designing a wireless sensor network that detects the earth quake, the sensor has its own property like lifetime, withstand ability, capacity, etc.

- We can design this wireless sensors with the same property in contiki cooja. When compared to other simulators cooja is developed purely for wireless sensor networks.

- In addition cooja is more flexible to change the properties of a node so that we could implement our own idea exactly. Wireless sensors play important role in IOT (Internet of Thins), where contiki Operating system was developed mainly for IOT devices, cooja is a simulator comes with the Contiki. So we can use the Cooja simulator for simulating any wireless sensor networks.

5.12 TOSSIM

- TOSSIM (TinyOS Mote Simulator) is an open-source operating system specially developed for the wireless embedded sensor networks. There are few hardware platforms available for TinyOS, some commercial and some non-commercial.

- TinyOS release includes a simulator called TOSSIM. It is built especially for Berkeley Mica Mote platform. TOSSIM is an emulator rather than a simulator, as it runs actual application code. Simulated application code can be transferred directly to the platform, but it might not run in a mote as it runs in a simulation due to the simplifying assumptions in TOSSIM.

- Figure 5.6 shows the working flow of TOSSIM. The TOSSIM architecture is consisted of five segments: Frames, Components, Models, Services and Events.

- TOSSIM is a very simple but powerful emulator for WSN. Each node can be evaluated under perfect transmission conditions, and using this emulator can capture the hidden terminal problems.

- As a specific network emulator, TOSSIM can support thousands of nodes simulation. This is a very good feature, because it can more accurately simulate the real world situation. Besides network, TOSSIM can emulate radio models and code executions. This emulator may be provided more precise simulation result at component levels because of compiling directly to native codes.

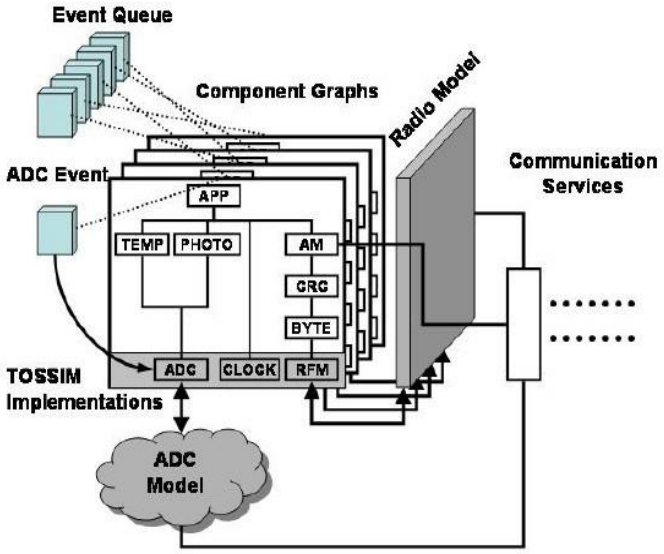


Figure 5.6 TOSSIM Architecture

Source : Protocol and Architecture for Wireless Sensor Networks by Holger Karl , Andreas willig

- TOSSIM is a bit-level discrete event network emulator built in Python, a high-level programming language emphasizing code readability, and C++. It can run TOSSIM on Linux Operating Systems or on Cygwin on Windows.
- TOSSIM also provides open sources and online documents. Developers had set four requirements for TOSSIM: scalability, completeness, fidelity and bridging.
- To be scalable, a simulator should manage networks of thousands of nodes in a wide variety of configurations. To achieve this, each node in TOSSIM is connected in a directed graph where each edge has a probabilistic biterror.
- For completeness, a simulator must capture behavior and interactions of a system at a wide variety of levels. And for fidelity, a simulator must capture behavior of a network with a subtle timing of interactions on a mote and between motes. Requirement for bridging is met as the simulated code runs directly in a real mote.
- The goal of TOSSIM is to study the behavior of TinyOS and its applications rather than performance metrics of some new protocol. Hence, it has some limitations, for instance, it does not capture energy consumption. Another drawback of this framework is that every node must run the same code. Therefore, TOSSIM cannot be used to evaluate some types of heterogeneous applications.

5.13 Programming Beyond Individual Nodes

- Sensor-actuator network systems offer some unique advantages. Dense networks of distributed sensors can improve perceived signal-to-noise ratio by reducing average distances from sensor to physical phenomena.
- In-network processing and actuation shorten the feedback chain and improve the timeliness of observation and response. Untethered network nodes and infrastructure less mesh network topologies reduce deployment costs. However, the greatest advantages of networked systems are improved robustness and scalability.
- A decentralized system is inherently more robust against individual node or link failures because of network redundancy. Decentralized algorithms are also far more scalable in practical deployment; they might be the only way to achieve the large scales needed for some applications. Because of decentralized systems spatial coverage and multiplicity in sensing aspect and modality, the detection, classification, and tracking of moving, nonlocal, or low-observable events require cross-node collaboration among sensors.

Target tracking as a motivating example

- Tracking is a canonical problem for sensor networks and essential for many commercial and military applications such as traffic monitoring, facility security, and battlefield situational awareness.
- Given a moving point signal source or target in a 2D sensor field, a tracking system's goal is to estimate target state histories, such as spatial trajectory, on the basis of sensor measurements.
- From a tracking expert's point of view, each sensor node provides a local measurement useful in estimating the target state. However, in most cases, only a relatively small subset of sensors contribute significantly to the estimation, owing to sensing-range limitations. In this case, a good solution is a leader-based tracking scheme, such as Information-Driven Sensor Querying (IDSQ), to fuse information from only the sensors that provide high-quality measurements.
- As Figure 5.7 illustrates, at any time instant t, IDSQ designates a single node, located close to the target, as leader. The leader node fuses these high signal-to-noise ratio measurements and updates its current target location estimate, referred to as the belief.
- For most sensor types, owing to the physical properties of signal propagation, the sensors with high signal-to-noise ratio will be within a limited range of the leader node. So, we can minimize the communication cost and latency for gathering sensor data.

- As the target traverses the sensor field and the belief evolves to follow its motion, the most "informative" sensors might no longer be those closest to the current leader. A nearby sensor might then be selected to replace this leader on the basis of the updated belief and a criterion combining resource constraints with some measure of sensing utility (such as mutual information). The current leader then hands off the belief to this sensor, which becomes the next leader at time $t + \delta$, where δ is the communication delay. The process of sensing, estimation, and leader selection repeats.

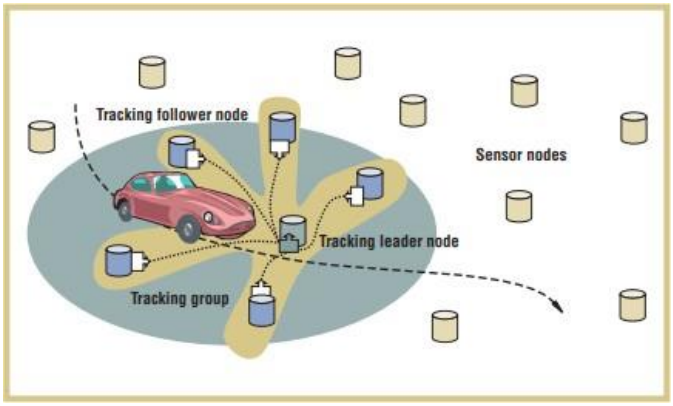


Figure 5.7. Collaborative processing in a leader-based object-tracking scenario.
Source : Protocol and Architecture for Wireless Sensor Networks by Holger Karl , Andreas willig

- As a vehicle moves through a sensor field, nearby sensors detect it. An elected leader node aggregates data from the active sensors and migrates the information from node to node as the vehicle moves.
- The sensor nodes collaborate primarily to improve sensing accuracy, and acceptable estimation quality might be achieved using only a subset of the sensors.
- One node, the leader, plays a key role in fusing others' sensor measurements. If no leader is present, all sensors that form the contour are equally important. Each node might locally update and repair its observation of a contour section, but the global state can only be assembled from observations of many nodes along the entire contour. Hence, System designers must explicitly write code to
  - Maintain sensor connectivities in a neighbourhood
  - Discover the best node for handoff
  - Invite neighbour nodes into the group
  - Handle communication delays and failures

5.14 State Centric Programming

- Many sensor network applications, such as target tracking (Discussed in previous section 5.13), are not simply generic distributed programs over an ad hoc network of energy-constrained nodes.
- Deeply rooted in these applications is the notion of states of physical phenomena and models of their evolution over space and time. Some of these states may be represented on a small number of nodes and evolve over time, as in the target tracking problem, while others may be represented over a large and spatially distributed number of nodes, as in tracking a temperature contour.
- A distinctive property of physical states, such as location, shape, and motion of objects, is their continuity in space and time. Their sensing and control is typically done through sequential state updates. System theories, the basis for most signal and information processing algorithms, provide abstractions for state updates, such as:

  $x_{k+1} = f(x_k, u_k)$

  $y_k = g(x_k, u_k)$
- Where **x** is the state of a system, u is the system input, **y** is the output and k is an integer update index over space and/or time, f is the state update function, and g is the output or observation function.
- This formulation is broad enough to capture a wide variety of algorithms in sensor fusion, signal processing, and control (e.g., Kalman filtering, Bayesian estimation, system identification, feedback control laws, and finite-state automata).
- However, in distributed real-time embedded systems such as sensor networks, the formulation is not as clean as represented in the above equations. The relationships among subsystems can be highly complex and dynamic over space and time.
- The following issues must be properly addressed during the design to ensure the correctness and efficiency of the system.
  - Where are the state variables stored?
  - Where do the inputs come from?
  - Where do the outputs go?
  - Where are the functions f and g evaluated?
  - How long does the acquisition of input take?
  - Are the inputs in uk collected synchronously?
  - Do the inputs arrive in the correct order through communication?
  - What is the time duration between indices k and k +1? Is it a constant?

- These issues, addressing where and when, rather than how, to perform sensing, computation, and communication, play a central role in the overall system performance.
- However, these 'non-functional" aspects of computation, related to concurrency, responsiveness, networking, and resource management, are not well supported by traditional programming models and languages.
- **State-centric programming aims** at providing design methodologies and frameworks that give meaningful abstractions for these issues, so that system designers can continue to write algorithms on top of an intuitive understanding of where and when the operations are performed.
- A collaborative group is such an abstraction. A collaborative group is a set of entities that contribute to a state update. These entities can be physical sensor nodes, or they can be more abstract system components such as virtual sensors or mobile agents hopping among sensors. These are all referred to as agents.
- Intuitively, a collaboration groups provides two abstractions: its scope to encapsulate network topologies and its structure to encapsulate communication protocols. The scope of a group defines the membership of the nodes with respect to the group.
- A software agent that hops among the sensor nodes to track a target is a virtual node, while a real node is physical sensor. Limiting the scope of a group to a subset of the entire space of all agents improves scalability.
- Grouping nodes according to some physical attributes rather than node addresses is an important and distinguishing characteristic of sensor networks. The structure of a group defines the "roles" each member plays in the group, and thus the flow of data.
  - Are all members in the group equal peers?
  - Is there a "leader" member in the group that consumes data?
  - Do members in the group form a tree with parent and children relations?
- For example, a group may have a leader node that collects certain sensor readings from all followers. By mapping the leader and the followers onto concrete sensor nodes, one can effectively define the flow of data from the hosts of followers to the host of the leader. The notion of roles also shields programmers from addressing individual nodes either by name or address.
- Furthermore, having multiple members with the same role provides some degree of redundancy and improves robustness of the application in the presence of node and link failures.