

INFRASTRUCTURE ESTABLISHMENT &

SECURITY IN WSN

SECURITY IN AD-HOC WIRELESS NETWORKS:

Ad-hoc wireless networks are highly vulnerable to security attacks compared to wired networks or infrastructure-based wireless networks. The following sections discuss the various security requirements in ad-hoc wireless networks, the different types of attacks possible in such networks and some of the solutions proposed for ensuring network security.

Network Security Requirements:

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

1. Confidentiality:

The data sent by the sender must be understandable 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. Data Encryption can be used to ensure confidentiality.

2. 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 to tamper with the data during transmission.

3. Availability:

- The network should remain operational all the time.
- It must be robust enough to tolerate link-failures and capable of surviving various attacks mounted on it.
- It should be able to provide guaranteed services whenever an authorized-user requires them.

4. Non-Repudiation: (Assurance)

- It is a mechanism to guarantee that the sender of a message cannot later deny having sent the message & that the recipient cannot deny having received the message.
- Digital signatures are used for this purpose.

Issues And Challenges In Security Provisioning:

1. Shared - Broadcast Radio channel:

- The radio channel used for communication is broadcast in nature and It 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 transmitted-data in the network.
- This problem can be minimized to a certain extent by using directional antennas,

2. Limited Resource Availability:

- Resources such as bandwidth, battery-power & Computational power are limited in ad-hoc networks.

∴ Hence, it is difficult to implement complex cryptography based security mechanisms in networks.

3. Insecure Operational Environment:

- The operating environments where ad-hoc wireless is used may not always be secure.
- One important application of such networks is in battlefields.

A. Physical Vulnerability:

- Nodes in the networks are usually compact & hand-held in nature.
- They could get damaged easily & are also vulnerable to theft.

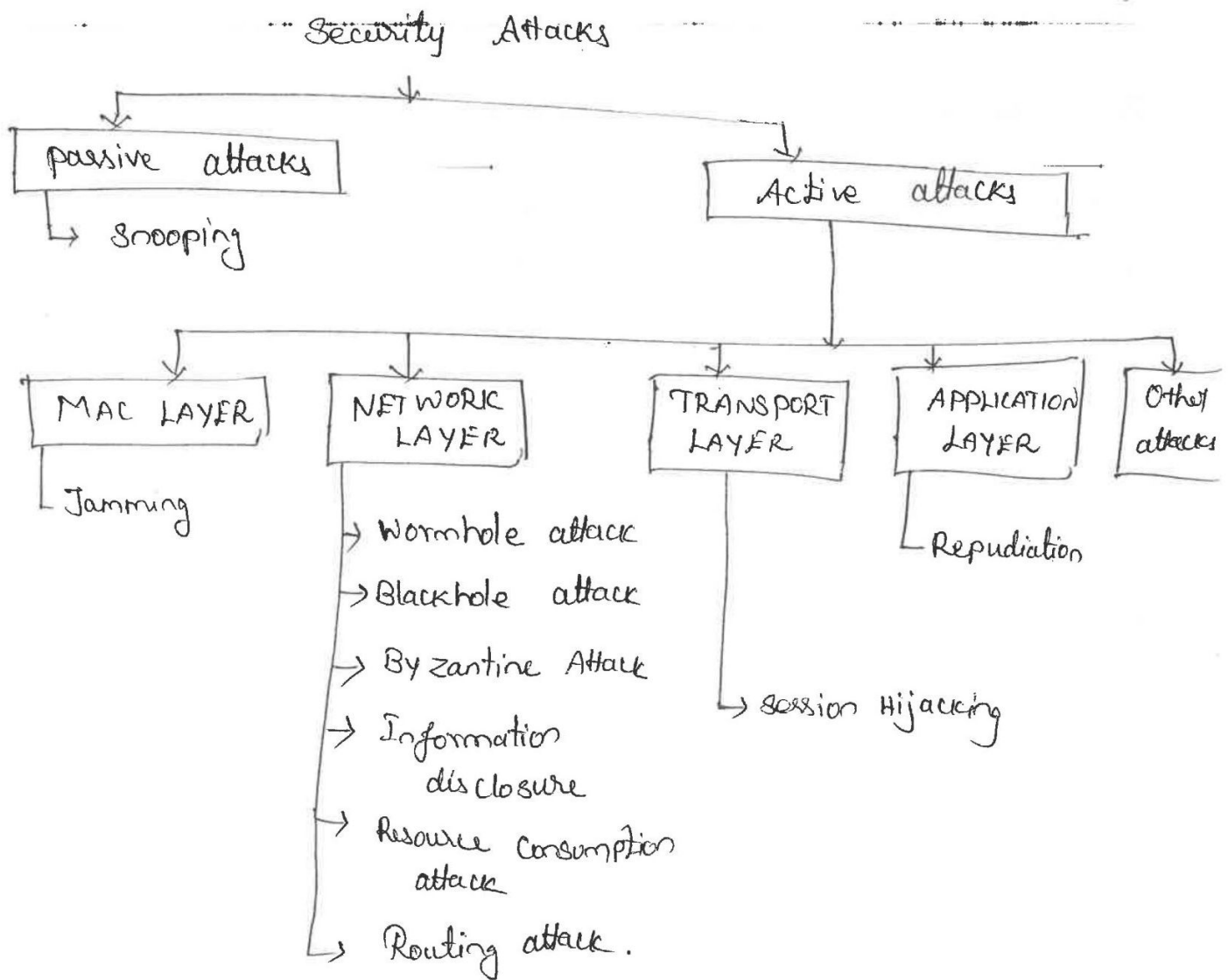
B. Lack of Central Authority:

- In wired-networks & infrastructure-based wireless network it would be possible to monitor the traffic on the network through certain important central points and implement security mechanisms at such points.
- Since ad-hoc networks do not have central points, these mechanisms cannot be applied in adhoc networks.

6. 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 attacks.

Network Security Attacks:



Attacks on adhoc networks can be Classified into 2 broad categories, namely

1. Passive attacks:

- It does not disrupt the normal operation of the network
- The adversary Snoops the data exchanged in the networks, with-out altering it.
- Data encryption can be used to overcome this problem.

2. Active Attacks:

- It disrupts the normal functioning of the network.

• It attempts to alter (or destroy) the data being exchanged in the network.

• They can be further classified into 2 categories:

→ External Attacks: They are carried out by nodes that do not belong to the network. They can be prevented using standard encryption techniques and firewalls.

→ 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 Attacks:

An Attacker receives packets at one location in the network and tunnels them to another location in the network, where the packets are resent into the network. This tunneling between two attackers is referred to as a wormhole.

If proper mechanism is not employed to defend the network against wormhole attacks, existing routing protocols for adhoc networks may fail to find valid routes.

2. Blackhole Attacks:

• A malicious-node falsely advertises good path to destination-node falsely advertising during path finding process.

• 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

(secret agreement)

- A set of compromised-node work in collusion & 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 information to unauthorized-nodes in the network.

5. Resource consumption Attack:

- A malicious-node tries to consume or waste resources of other nodes present in the network.
- The resources targeted are battery-power, bandwidth & computational-power.

6. Routing attacks:

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

i) Routing table overflow:

- Adversary-node advertises routes to non-existent nodes, to the authorized-nodes present in the network,
- The main objective is to cause an overflow of routing table

ii) Routing table Poisoning:

- The compromised-nodes send wrong routing update (or) modify genuine route update packets.
- This may result in sub-optimal routing, congesting in network and even make some part of network unreachable

- iii) Packet Replication
- iv) Route cache poisoning
- v) Rushing Attack

Transport Layer Attacks:

Session Hijacking:

- 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:

Repudiation:

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

Other Attacks:

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

Multilayer attacks:

Multilayer attacks are those that could occur in any layer of the network protocol stack. Some of the multilayer attacks in adhoc networks are:

1. Denial of Service (Dos):

- An Adversary attempts to prevent authorized-user 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: —

Jamming, SYN flooding, Distributed DoS Attacks.

2. Impersonation:

- 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.

- A man-in-the-middle attack is another type of impersonation attack.

Device Tempering:

- Unlike nodes in a wired network, nodes in adhoc networks are usually compact, soft and hand-held in nature.

- They could get damaged or stolen easily.

— x —

Key Management:

- CRYPTOGRAPHY is one of the most common & reliable means to ensure security & can be applied to any communication network.

- 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 the ciphertext back into plaintext by the process of decryption.

- The process of encryption & decryption are governed by keys. Keys are small amounts of information used by the cryptographic algorithms. When the keys are to be kept ~~security~~ secret to ensure the security of the system, it is called a secret key.

- The secure administration of cryptographic key is called key management.

- The 4 main goals of cryptography are confidentiality, integrity, authentication & non-repudiation.

- There are 2 major kinds of cryptographic algorithms:

- Symmetric key algorithms, which use the same key for encryption & decryption.

- Asymmetric key algorithms, which use two different keys for encryption & decryption.

Symmetric Key Algorithms:

- These 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

- Block ciphers

- Stream ciphers.

- A block cipher is an encryption scheme in which plaintext is broken into fixed-length segments called blocks, and 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.

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

original Alphabet	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Substitution	E F G H I J K L M N O P Q R S T U V W X Y Z A B C D
plaintext	EVERYDAY CREATES A HISTORY EVERY DAYCR EATES AHIST ORY
Cipher text	IZIVC HECGV IEXIW ELMWX SVC

fig: (a) encryption using substitution.

Transposition	1 2 3 4 5 3 5 1 4 2
plaintext	EVERYDAY CREATES A HISTORY EVERY DAYCR EATES AHIST ORY
Transposition	EYERV YRDCA TSEEA ITASH YOR

fig: (b) transposition cipher.

- The block length used is 5.
- One of the simplest stream ciphers is vernam cipher, which uses a key of same length as plaintext for encryption.
For example :

If plaintext is the binary string = 10010100
key = 01011001

then the encrypted string is given by the XOR of the plaintext & key.

plain text $\Rightarrow$ 10010100

key $\Rightarrow$ 01011001

$\oplus$ 11001101 $\Leftarrow$ encrypted key/ cipher text

plaintext is again recovered by XOR-ing the cipher text with the same key.

Decryption:

Cipher Text = 11001101

key = 01011001

10010100 $\Rightarrow$ plain Message.

Asymmetric Key Algorithms:

- These use different keys at the sender-end & receiver end for encryption and decryption, respectively.
- Let the encryption process be represented by a function E and decryption by D .

Then the plaintext 'm' is transmitted into the ciphertext 'c'

$$c = E(m)$$

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

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

The key E is made public, while D is made private.

Only to the intended receiver.

- RSA algorithm is the best example of public key cryptography.
- Digital signatures (DS) schemes are also based on public key encryption.
 - In DS, the person who wishes to sign a document encrypts it using his private key D , which is known only to him.
 - Anybody who has his public key E can decrypt it and obtain the original document.
 - This is usually a governmental or business organization.

— x —

Secure Routing in ad-hoc wireless networks:

Ensuring secure communication in ad-hoc networks includes the mobility of nodes, a promiscuous mode of operation, limited processing power & limited availability of resources such as battery power, bandwidth & memory.

Requirements of a secure Routing Protocol for Adhoc networks.

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

i) Detection of Malicious Nodes:

A routing protocol should be able to detect the presence of any malicious-node in the network, and should avoid

The participation of such nodes in the routing process.

2) Guarantee of ~~Correct~~ Route Discovery:

If a route between the source & destination node exist, the routing protocol should be able to find the route & also ensure the correctness of the selected route.

3) 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.

4) Stability against Attacks:

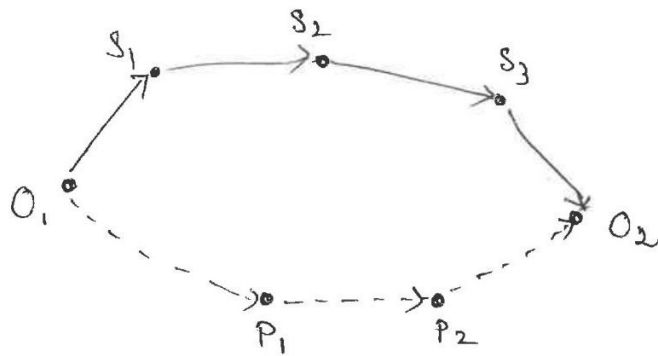
The routing protocols must be self-stable in the sense that it must be able to revert to its normal operating state within a finite amount of time after passive or an active attack.

— x —

Security Aware Routing Protocol (SAR)

- This uses security as one of the key metrics in path find.
- In adhoc networks, communication between end-nodes through possibly multiple intermediate-node is based on the fact that the two end-nodes trust the intermediate-nodes.

- This defines Level of Trust as a metric for routing & as one of the attributes for Security to be taken into consideration while routing.



where

---> Shortest path
 —> Secure path
 O_i → officer-node
 S_i → Secure-node
 P_i → Private-node

Two paths exist between the two officers O₁ and O₂ who want to communicate with each other.

One of these paths is a shorter path which runs through private-node whose trust levels are very low.

Hence, the protocol chooses a longer but secure path which passes through other secure-nodes.

Nodes of equal levels of trust distribute a common key among themselves and with those nodes having higher levels of trust.

This could be incorporated into both on-demand and table-driven routing protocols.

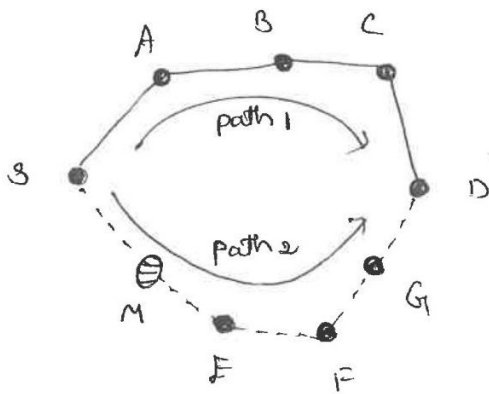
This allow the application to choose the level of security it requires.

SECURITY AWARE AODV PROTOCOL:

- AODV is an on-demand routing protocol where the route discovery process is initiated by sending RouteRequest packet only

When data packets arrived at a node for transmission,

- A malicious-node could advertise that it has the shortest path to the destination, thereby redirecting all the packets through itself. This is known as blackhole attack.

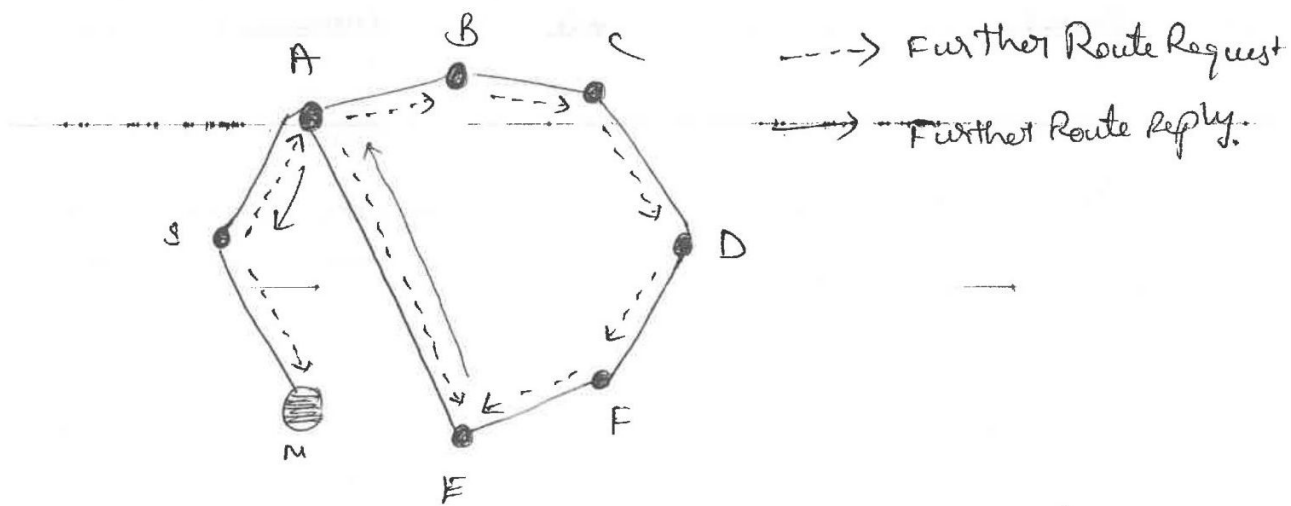


- Let node M be the malicious-node that enters the network.
- It advertises that it has the shortest path to the destination node D when it receives the RouteRequest packet sent by node S.
- The attacker may not be able to succeed if node A, which also receives the RouteRequest packet from node S, replies earlier than node M.

Advantage: malicious-node does not have to search its routing table for a route to the destination. Hence the malicious-node would be able to reply faster than node A.

Solutions FOR THE BLACK HOLE PROBLEM:

- One of the solutions for the blackhole problem is to restrict the intermediate-nodes from originating RouteReply packets.
- only the destination-node would be permitted to initiate RouteReply packets.
- Security is not completely assured.
- The delay involved in the route discovery process increases as the size of the network increases.



- The source node s sends Further Route Request packets to this neighbour - node E .
- Node E responds by sending a Further Route Reply packet to source node.
- Since node M is a malicious-node which is not present in the routing list of node E , the Further Route Reply packet sent by node E will not contain a route to the malicious - node M .
- This protocol completely eliminates the blackhole attack caused by a single attacker.
- Disadvantage: Control overhead of the routing protocol increases considerably.
- If the malicious-nodes work in a group, this protocol fails miserably.

INFRASTRUCTURE Establishment:

When a sensor network is first activated, various tasks must be performed to establish the necessary infrastructure that will allow useful collaborative work to be performed. In particular, each node must discover with other nodes it can directly communicate with, and its radio power must be set appropriately to ensure adequate connectivity.

Nodes near one another may wish to organize themselves into clusters, so that sensing redundancy can be avoided and scarce resources, such as radio frequency, may be reused across non-overlapping clusters. In this chapter, we survey some common techniques used to establish infrastructure: topology control, clustering, time synchronization and location for the network node.

Topology Control:

A sensor network node that first wakes up executes a protocol to discover which other nodes it can communicate with bidirectionally. This set of neighbors is determined by the radio power of the node as well as the local topography and other conditions that may degrade radio links.

However, nodes in a wireless sensor network can change the topology of the network by choosing to broadcast at less than their maximum possible power. This can be advantageous in situations where there is dense node deployment, as radio power is the main drain on a node's batteries. The problem of topology control for a sensor network is how to set the radio range for each node so as

to minimize energy usage, while still ensuring that the communication graph of the nodes remains connected and satisfies other desirable communication properties.

Although in principle the transmitting range of each node can be set independently, let us first examine the simpler case where all nodes must use the same transmission range: in expensive radio transmitters, for example, may not allow the range to be adjusted. We also ignore all effects of interference or multipath, so that any pair of nodes within range of each other can communicate. This homogeneous topology control setting defines the critical transmitting range (CTR) Problem: compute the minimum common transmitting range r such that the network is connected.

Clustering:

The nodes in a sensor network often need to organize themselves into clusters. Clustering allows hierarchical structures to be built on the nodes and enables more efficient use of scarce resources, such as frequency spectrum, bandwidth and power.

For example, if the cluster size corresponds roughly with the direct communication range of the nodes, much simpler protocols can be used for routing and broadcasting within cluster; furthermore, the same time or frequency division multiplexing can be reused across non-overlapping clusters. Clustering also allows the health of the network to be monitored and misbehaving nodes to be identified, as some nodes in the cluster can play watchdog role over other nodes.

Finally, network can be comprised of mixtures of nodes, including some that are more powerful or have special capabilities, such as increased communication range, GPS, and the like. These more capable nodes can naturally play the role of cluster-head.

Time Synchronization:

Since the nodes in a sensor network operate independently, their clocks may not be, or stay, synchronized with one another. This can cause difficulties when trying to integrate and interpret information sensed at different nodes. For example, if a moving car is detected at two different times along a road, before we can even tell in what direction the car is moving, we have to be able to meaningfully compare the detection times. And clearly we must be able to transform the two times reading into a common frame of reference before we can estimate the speed of the vehicle. Estimating time differences across nodes accurately is also important during node localization.

For instance, many localized algorithms use ranging technologies to estimate internode distance; in these technologies, synchronization is needed for time-of-flight measurements that are then transformed into distances by multiplying with the medium propagation speed for the type of signal used. Configuring a beam-forming array or setting a TDMA radio schedule are just two more examples of situations in which collaborative sensing requires the nodes

involved to agreed on a common time frame,

The latency in channel can be decomposed into four components

Send time: This is the time taken by the sender to construct the message, including delays introduced by operating system calls, context switching and the data access to the network interface.

Access time: This is the delay incurred while waiting for access to the transmission channel due to contention, collisions and the like. The details of that are very MAC-specific.

Propagation time: This is the time for the message to travel across the channel to the destination node. It can be highly variable from negligible for single-hop wireless transmission to very long in multihop wire-area transmission.

Receive time: This is the time for the network interface on the receiver side to get the message and notify the host of its arrival. This delay can be kept small by time-stamping the incoming packet inside the network driver's interrupt handler.

Localization and Positioning:

An attractive feature of sensor network is that it can provide information about the world that is highly localized in space and/or time. For several sensor net applications, including target tracking and habitat monitoring, knowing exact location where information was collected is critical. In fact, for almost all sensor net applications, the value of the information collected can be enhanced if the

Location of the sensors where readings were made is also available. With current technologies, most sensor nodes remain static. Thus one way to know the node positions is to have the network installer measure these locations during network deployment.

Self-Localization is a method, that allow the nodes in a network to determine their geographic positions on their own as much as possible, during the network initialization process. We also describe location service algorithms - methods that allow other nodes to obtain the location of a desired node, after the initial phase in which each node discovers its own location. Such location services are important for geographic routing, location-aware query processing and many other tasks in a sensor network.

Sensor Tasking and Control:

To efficiently and optimally utilize scarce resources in a sensor network, such as limited on board battery power supply and limited communication bandwidth, nodes in a sensor network must be carefully tasked and controlled to carry out the required set of tasks while consuming only a modest amount of resources.

For example, a camera sensor may be tasked to look for animals of a particular size and colour, or an acoustic sensor may be tasked to detect the presence of a particular type of vehicle. To detect and track to detect the presence of a particular type of vehicle.

To detect and track a moving vehicle, a pan-and-tilt camera may be tasked to anticipate and follow the vehicle object. It should be noted that to achieve scalability and autonomy, sensor tasking and control have to be carried out in a distributed fashion, largely using only information available to each sensor.

As a sensor network's primary function is to collect information from a physical environment, we must rethink the role of routing. A collaborative processing group is an important abstraction of physical sensors, since individual sensors are ephemeral and hence less important, and sensors collectively support a set of tasks. The challenge is to efficiently create, maintain and migrate groups as tasks and physical environments change.

A major benefit of establishing the collaborative group is in enabling the programming of sensor networks to move from addressing individual nodes to addressing collectives.

— x —

UNIT - 8

SENSOR NETWORK PLATFORMS AND TOOLS

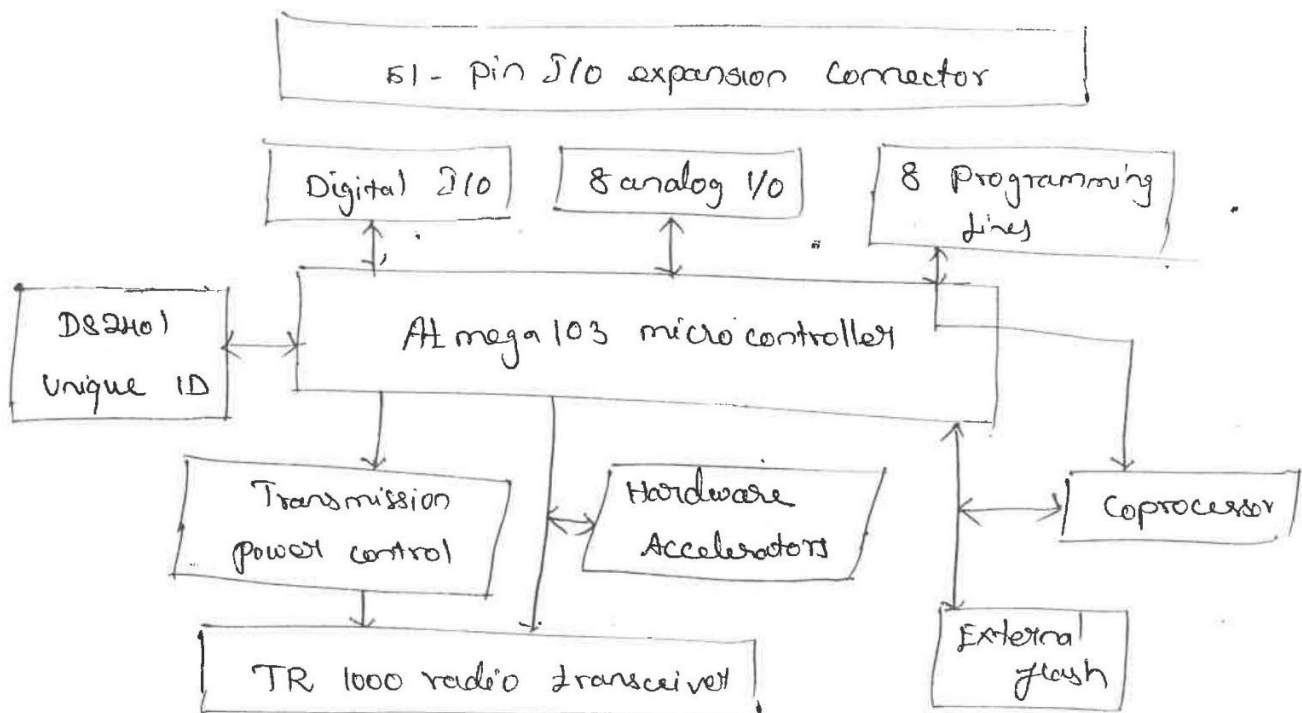
Sensor Node Hardware :

Sensor node Hardware can be grouped into three categories, each of which entails a different set of trade-offs in the design choices.

- Augmented general-purpose computers
- Dedicated embedded sensor nodes
- System-on-chip nodes

Berkeley Notes :

The Berkeley notes are a family of embedded sensor nodes sharing roughly the same architectures. Let us take the MICA nodes as an example. The MICA nodes have a two-cpu design as shown in figure 7.2.

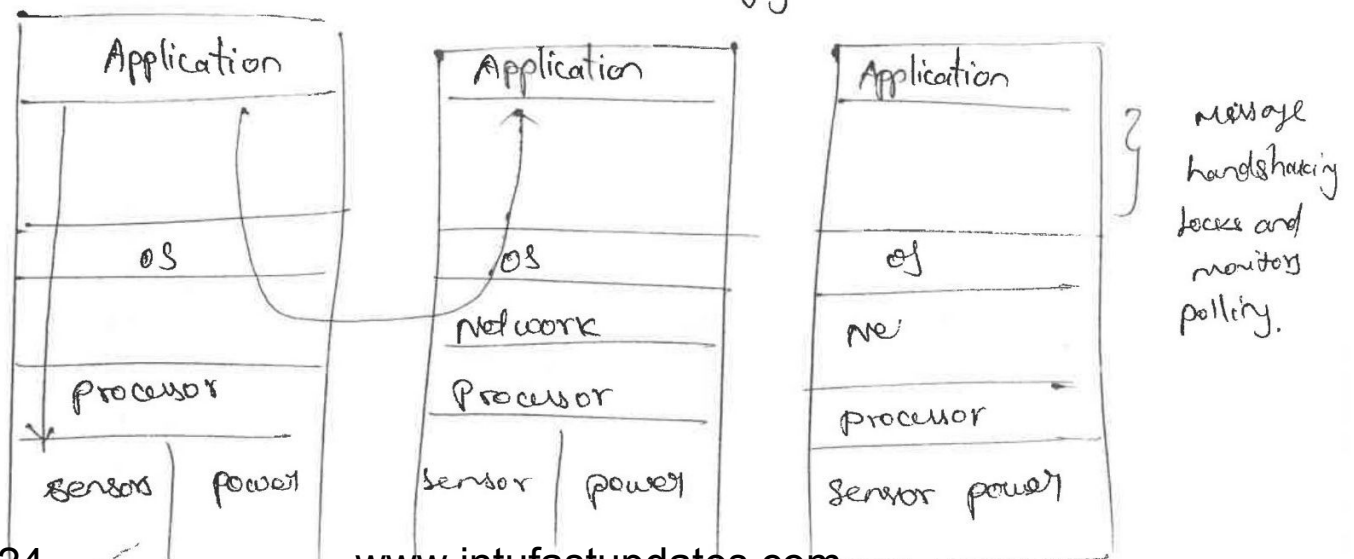


The main microcontroller, an Atmel ATmega 1032, takes care of regular processing. A separate and much less capable coprocessor is only active when the MCU is being reprogrammed. The ATmega 1032 MCU has integrated 512 KB flash memory and 1 KB of data memory.

In addition to the memory inside the MCU, a MICA mote also has a separate 512 KB flash memory unit that can hold data. Since the connection between the MCU and this external memory is via a low-speed serial peripheral interface protocol, the external memory is more suited for storing data for later batch processing than for storing programs. MICA motes implement a 40 Kbps transmission rate. The transmission power can be digitally adjusted by software through a potentiometer. The max transmission range is about 300 feet in open space.

Sensor Network Programming Challenges:

Traditional programming technologies rely on operating system to provide abstraction for processing, I/O, networking and user interaction hardware as illustrated in figure 7.4.



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 that covers all extreme cases: unreliable communication channels, long delays, irregular arrival of messages, simultaneous events and so on.

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 a node centric operating system, which provides hardware and networking abstraction of a sensor node to programmers or it can be a language platform, 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.

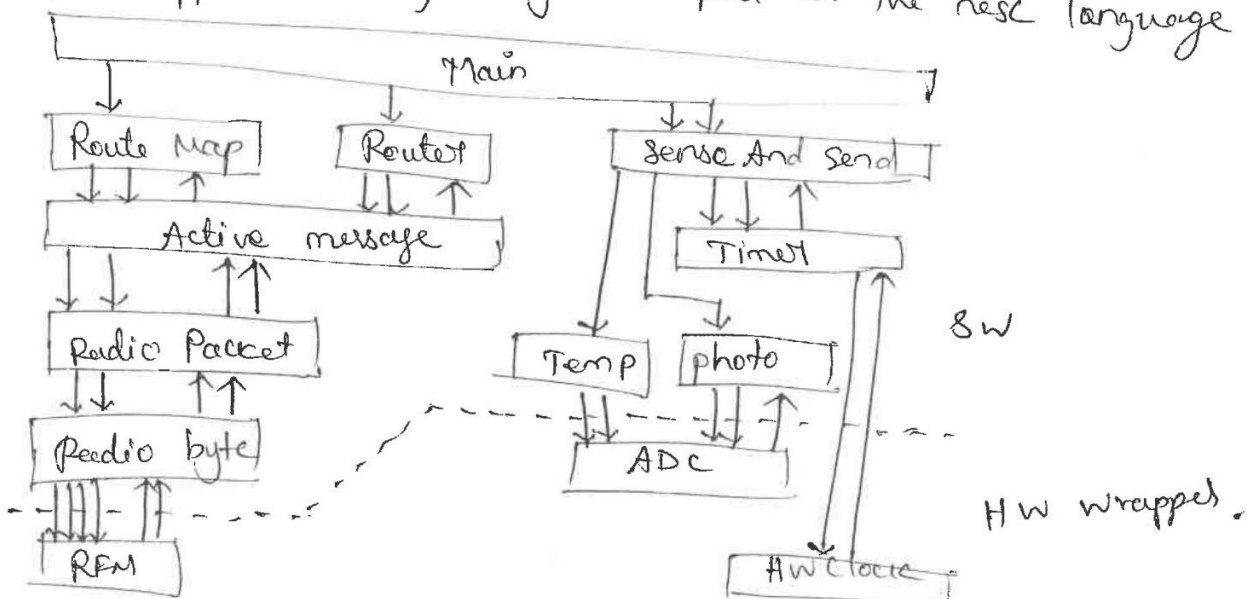
A typical operating system abstracts the hardware platform by providing a set of services for applications, including file management, memory allocation, task scheduling, peripheral device drivers and networking. TinyOS and TinyGALS are two representative examples of node level programming tools that we will cover in detail in this section.

Operating System : TinyOS aims at supporting sensor

network applications on resource constrained hardware platforms such as the Berkeley motes.

To ensure that an application code has an extremely small foot print, Tinyos chooses to have no file system, support only static memory allocation, implements a simple task model and provides minimal device and networking abstractions. Furthermore, it takes a language-based application development approach, to be discussed compiled with the application. To a certain extent, each Tinyos application is built into the operating system.

Like many operating systems, Tinyos organizes components into layers. Intuitively, the lower a layer is, the "closer" it is to the hardware; the higher a layer is, the "closer" it is to the application. In addition to the layers, Tinyos has a unique component architecture and provides as a library a set of system software components. A component specification is independent of the component implementation. Although most components encapsulate software functionalities, some are just thin wrappers around hardware. An application, typically developed in the nesc language covered.



Node - Level Simulators:

Node-Level design methodologies are usually associated with simulators that simulate the behaviour of a sensor network on a per-node basis. Using simulation, designers can quickly study the performance 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.

Communication model: Depending on the details of modeling, 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 transmission. Alternately, the communication may be simulated at the MAC layer or network layer, using, for example, stochastic processes to represent low-level behaviour.

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 behaviour is extremely important.

The NS-2 simulator and its sensor Network Extensions:

The simulator ns-2 is an open-source network simulator that was originally designed for wired, IP networks. Extensions have been made to simulate wireless/mobile networks and more recently sensor networks, while the original ns-2 only supports logical addresses for each node and a simple wireless channel model. This is not a trivial extension, since once the nodes move, the simulator needs to check for each physical layer event whether the destination node is within the communication range.

The main functionality of ns-2 is implemented in C++, while the dynamics of the simulation is controlled by Tcl scripts. Basic components in ns-2 are the layers in the protocol stack. They implement the handlers interface, indicating that they handle events. Events are communication packets that are passed between consecutive layers within one node, or between the same layers across nodes.

The key advantage of ns-2 is its rich libraries of protocol for nearly all network layers and for many routing mechanisms. These protocols are modeled in fair details, so that they closely resemble the actual protocol implementations.

TCP: reno, Tahoe, vegas and SACK implementations.

MAC: 802.3, 802.11 and TDMA

Ad-hoc routing: DSDV, DSR, AODV and TORA.

Sensor N/w routing: Directed diffusion, geographical routing
and GAF routing

The Simulator (Tossim) TOSSIM :

TOSSIM is a dedicated simulator for TinyOS applications running on one or more Berkeley nodes. The key design decision on building TOSSIM were to make it scalable to a network of potentially thousands of nodes, and to be able to use the actual software code in the simulation.

TOSSIM uses a simple but powerful abstraction to model a wireless network. A network is a directed graph, where each vertex is a sensor node and each directed edge has a bit-error rate. Each node has a private piece of state representing what it hears on the radio channel. By setting connections among the vertices in the graph and a bit error rate on each connection, wireless channel characteristics, such as imperfect channels, hidden terminal problems and asymmetric links, can be easily modeled. Wireless transmissions are simulated at the bit level. If a bit error occurs, the simulator flips the bit.

Programming Beyond Individual Nodes: State-Centric Programming.

Many sensor network applications, such as target tracking, 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.

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 algorithm, provide abstractions for state update, 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 are the inputs, y are the outputs, 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.

1) Collaboration groups:

A Collaboration 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. In this context, they are all referred to as agents.

$$A = (A, L, P, R)$$

where

A — set of Agents

$L \rightarrow$ Labels

$P: A \rightarrow L$ is a function that assigns each agent a role.

$R \subset L \times L \rightarrow$ connectivity relations among roles

ii) Geographically constrained Group:

A geographically constrained group consists of members within a prespecified geographical extent. Since physical signals, especially the ones from point targets, may propagate only to a limited extent in an environment, this kind of group naturally represents all the sensor nodes that can possibly "sense" a phenomenon. A GCG may have a leader, which fuses information from all other members in the group.

N-hop Neighborhood Group: When the communication topology is more important than geographical topology extent, hop count is more useful to constrain group membership. An n-hop neighborhood group has an anchor node and defines that all nodes within n communication hops are members of the group.

publish/subscribe Group: A Group may also be defined more dynamically, by all entities that can provide certain data or services, or that can satisfy certain predicates over their observations or internal states. A publish/subscribe group comprises consumers expressing interest in specific types of data or services.

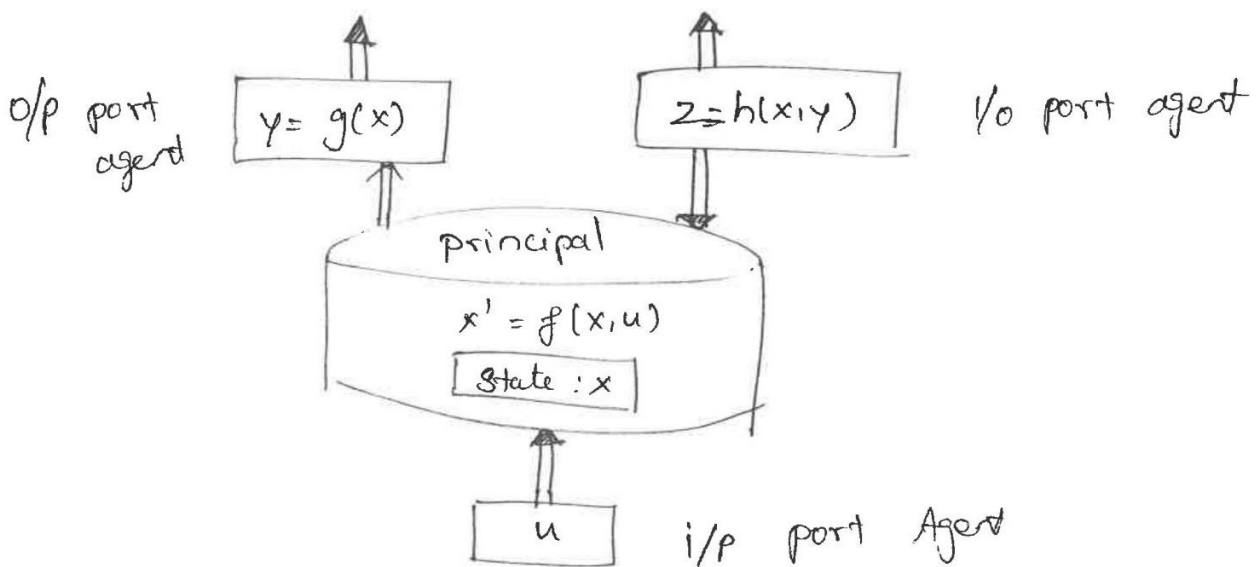
PIECES: A State-Centric Design Framework:

PIECES (Programming and Interaction Environment for collaborative Embedded Systems) is a

Software framework that implements the methodology of state-centric programming over collaboration groups to support the modeling, simulation and design of sensor network applications. It is implemented in a mixed JAVA-Matlab environment.

Principals and port Agents:

PIECES comprises principals and port Agents.



A principal is the key component for maintaining a piece of state. Typically, a principal maintains state corresponding to certain aspects of the physical phenomenon of interest. The role of a principal is to update its state, a computation corresponding to evaluating function f shown in figure. A principal also accepts other principals' queries of certain view on its own state, a computation corresponding to evaluating function g .

The execution of principals and port agents can be either time driven or event driven, where events may

may include physical events that are pushed to them or query events from other principals or agents. Principals maintain state, reflecting the physical phenomena. These state can be updated, rather than rediscovered, because the underlying physical states are typically continuous in time.

— x —

Applications of WSN:

Ultra wide band radio communication:

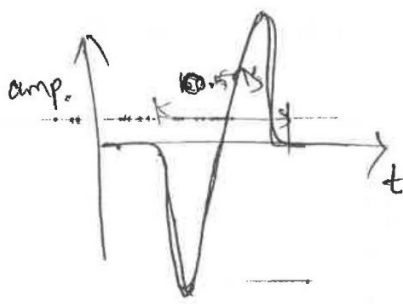
The major differences between the Ultra-wide band technology and the existing narrow-band and wide band technologies are the following:

1) The bandwidth of UWB system

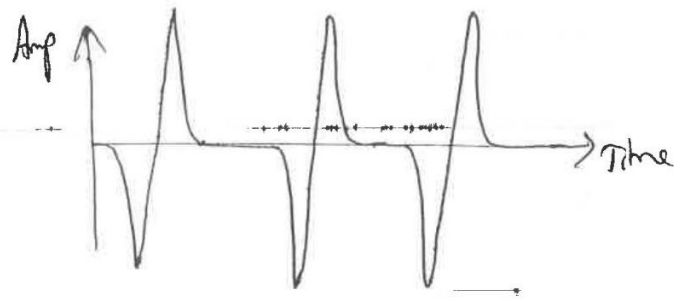
2) UWB systems are implemented in a carrier-less fashion in which the modulation scheme can directly modulate base band signals into an impulse with very sharp rise and fall time, thus resulting in a waveform ranging several GHz of bandwidth.

operation of UWB system:

The operation of UWB systems is based on transmission of ultra-short pulses which are also called mono cycles. Each mono cycle is similar to a single cycle of an ultra-high frequency sine wave and is a single ultra-short pulse. The pulse width of mono cycles ranges from 0.10 to 1.6 ns with interval 2 ns - 1000 ns.



a) A monopulse



b) A pulse train

Advantages and Disadvantages of UWB:

Promising applications of UWB in communication include cable-free audio/video devices, broadband WPANs and high-speed wireless links. UWB systems are potential candidates for high data rate, low power and short-to medium-range communication applications. UWB systems have a wide range of applications other than communications, some of which are automobile collision-detection devices, medical imaging similar to x-rays and ultrasound scans, through wall imaging for detecting people and objects in law-enforcement applications and ground penetrating radars in construction applications.

The major advantages of UWB systems include simplicity of implementation, high data rate, inherent robustness to multipath fading, flexibility of operation, low power consumption and low power and low cost implementation.

The disadvantages of UWB systems include stringent design requirements of communication subsystems and chances of interference from the existing technologies.

Wireless Fidelity Systems:

Wireless Fidelity (Wi-Fi) system is the high speed wireless LAN that was originally intended to extend the wired Ethernet in offices to wireless client. The coverage area and ability to support high bit rate are the two major reasons behind the name Wi-Fi. IEEE 802.11 a and IEEE 802.11 b are considered as the standard wi-fi candidates.

The service provider Models for wi-Fi systems:

- The Wi-Fi micro carrier models
- The franchisee - franchisor model
- The wi-fi carrier model
- The wi-fi aggregator model
- The extended service provider model.

Issues in Wi-Fi systems:

Security:

Security in Wi-Fi systems is important and requires viable solutions. The IEEE has proposed a long-term security architecture for 802.11, which is referred to as robust security network. There are three main states in the transition process of an association.

- i) unauthenticated and unassociated
- ii) authenticated and unassociated
- iii) authenticated and Associated.

The major elements in Wi-Fi system that can contribute to its vulnerability are discussed below.

1. The eavesdropper: Wireless medium is inherently broadcast, making it trivial for the adversary with a good receiver to eavesdrop on other stations' traffic. The use of a single key for encryption proved inadequate for this purpose. Dynamic changing of the encryption key is a must for making the eavesdropping a difficult task.

The rogue client: The authenticated client that has intentions of acting against the organization is another problem aggravated by the wireless environment. A Wi-Fi client running a routing protocol and an additional wired interface can easily use its authentication information on its corporate wireless network infrastructure to transfer valuable information. This problem is extremely difficult to trace and prevent as the loss incurred cannot easily be detected.

The rogue AP: The rogue AP is the unauthorized AP attached to an organization's network, either on its premises or in its authorized employee's home. An AP without the necessary security provisions, or in situations where the security is disabled temporarily, can act as a rogue AP.

The two major types of attacks:

Session hijack attack: In this case, a hijacker node waits until the successful completion of a node's authentication

Process with an AP. On completion of the Authentication, the hijacker sends a forged disassociate message that appears to be originated by the AP. On reception of the message, the original node gets disassociated. The AP continues to consider the node as part of the Network. Now the hijacker, using the originally authenticated node's address, can utilize network service until the session expires.

Man-in-the-middle attack:

When an adversary spoofs messages to represent itself as AP to a node under Attack, it can obtain the victim's authentication information. Using this information, the adversary authenticates with another AP masquerading as the victim node. This attack is known as man-in-the-middle Attack.