SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCSA5202-WIRELESS SENSOR NETWORKS

Unit- I- WIRELESS SENSOR NETWORKS-SCSA5202

UNIT I

NETWORK ARCHITECTURE

Concept of sensor network – Introduction, Applications, Sensors. Single Node Architecture: Hardware and software component of a sensor node-Tiny OS operating system-C language. Wireless Sensor Network architecture: Typical network architectures-Data relaying strategies Aggregation-Role of energy in routing decisions

1. Introduction

Wireless Sensor Networks

Wireless Sensor Networks (WSNs) can be defined as a self-configured and infrastructure-less wireless networks to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants and to cooperatively pass their data through the network to a main location.A Wireless Sensor Network is a self-configuring network of small sensor nodes communicating among themselves using radio signals, and deployed in quantity to sense, monitor and understand the physical world. Wireless Sensor nodes are called motes Provide a bridge between the real physical and virtual worlds .Allow the ability to observe the previously unobservable at a fine resolution over large spatio-temporal scales. Have a wide range of potential applications to industry, science, transportation, civil infrastructure, and security.

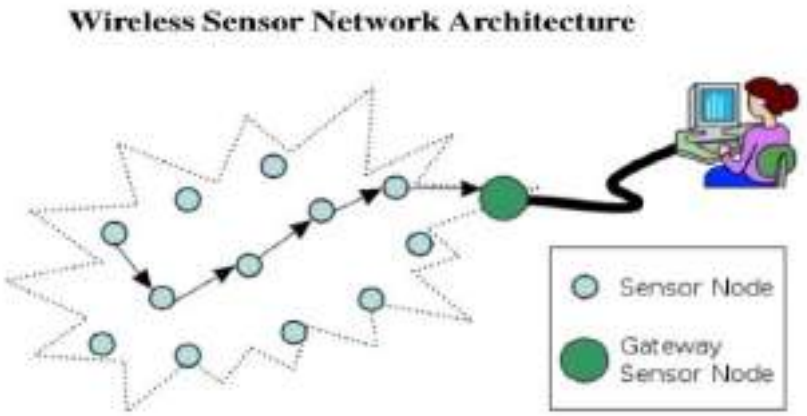


Figure 1: WSN Architecture

1.2 Applications of Wireless Sensor Networks

- Habitat and Ecosystem Monitoring
- Seismic Monitoring
- Civil Structural Health Monitoring
- Monitoring Groundwater Contamination
- Rapid Emergency Response
- Industrial Process Monitoring
- Perimeter Security and Surveillance
- Automated Building Climate Control

- Habitat Monitoring on Great Duck Island

1.2.1 FireBug

Wildfire Instrumentation System Using Networked Sensors.Allows predictive analysis of evolving fire behavior.Firebugs: GPS-enabled, wireless thermal sensor motes based on TinyOS that self-organize into networks for collecting real time data in wild fire environments.Software architecture: Several interacting layers (Sensors, Processing of sensor data, Command center)

1.2.2 Preventive Maintenance on an Oil Tanker in the North Sea: The BP ExperimentCollaboration of Intel & BP

Use of sensor networks to support preventive maintenance on board an oil tanker in the North Sea. A sensor network deployment onboard the ship. System gathered data reliably and recovered from errors when they occurred.

1.2.3 "Cricket" Mote

Basically a location-aware mote. Includes an Ultrasound transmitter and receiver.Uses the combination of RF and Ultrasound technologies to establish differential time of arrival and hence linear range estimates.

1.3 TinyOS

TinyOS is an embedded, component-based operating system and platform for low-power wireless devices, such as those used in wireless sensor networks (WSNs), smartdust, ubiquitous computing, personal area networks, building automation, and smart meters. It is written in the programming language nesC, as a set of cooperating tasks and processes. It began as a collaboration between the University of California, Berkeley, Intel Research, and Crossbow Technology, was released as free and open-source software under a BSD license, and has since grown into an international consortium, the TinyOS Alliance.

TinyOS has been used in space, being implemented in ESTCube-1. Low-power sensors, due to their limitations in scope, require efficient utilization of resources. TinyOS is essentially built on a "components-based architecture" to reduce code size to around 400 to 500 bytes and an "events-based design" which eliminates the need for even a command shell. The components-based architecture uses "nesC," which is a C programming language designed for networking embedded systems. Each code snippet consists of simple functions placed within components and complex functions integrating all the components together.

TinyOS also uses an "events-based design" whose objective is to put the CPU to rest when there are no pending tasks. An event can be something such as the triggering of an alert when the temperature of a thermostat rises or falls above a certain value. As soon as the event is over, the sensor motes can go to sleep.
The need for a design like TinyOS is mandatory in applications such as smart transit and smart factories. Because of thousands of sensors, it is important to have a very small memory footprint to reduce power requirements.

1.3.1 Applications of TinyOs

Environmental monitoring: since each TinyOS system can be embedded in a small sensor, they are useful in monitoring air pollution, forest fires, and natural disaster prevention.

Smart vehicles: smart vehicles are autonomous and can be understood as a network of sensors. These sensors communicate through low-power wireless area networks (LPWAN) which makes TinyOS a perfect fit.

Smart cities: TinyOS is a viable solution for the low-power sensor requirements of smart cities' utilities, power grids, Internet infrastructure and other applications.

Machine condition monitoring: machine-to-machine (M2M) applications have many sensor interfaces. It is impossible to assign a complete computing environment to each sensor. TinyOS can perform security, power management and debugging of the sensors.

1.3.2 Salient features of TinyOS

A simple event-based concurrency model and split-phase operations that influence the development phases and techniques when writing application code.

It has a component-based architecture which provides rapid innovation and implementation while reducing code size as required by the difficult memory constraints inherent in wireless sensor networks.

TinyOS's component library includes network protocols, distributed services, sensor drivers, and data acquisition tools.

TinyOS's event-driven execution model enables fine grained power management, yet allows the scheduling flexibility made necessary by the unpredictable nature of wireless communication and physical world interfaces.

1.4 Data Mule

Data mules have been used to offer internet connectivity to remote villages. Computers with a disk and wifi link are attached to buses on a bus route between villages. As a bus stops at the village to pick up passengers and cargo, the DTN router on the bus communicates with a DTN router in the bus station over Wi-Fi. DataMule – a mobile entity present in the environment that will pick up data from the mote when in range, buffer it, and drop off the data at base station.ex: People, Vehicles, Livestock.Data mules have been used to offer internet connectivity to remote villages. Computers with a disk and wifi link are attached to buses on a bus route between villages. As a bus stops at the village to pick up passengers and cargo, the DTN router on the bus communicates with a DTN router in the bus station over Wi-Fi. Email is down-loaded to the village and up-loaded for transport to the Internet or to other villages along the bus route.Data mules are a cost-effective mechanism for rural connectivity because

they use inexpensive commodity hardware, can be quickly installed, and can be piggy backed on existing transportation infrastructure.

1.5 Single Node Architecture

Choosing the hardware components for a wireless sensor node, obviously the applications has to consider size, costs, and energy consumption of the nodes. A basic sensor node comprises five main components such as Controller, Memory, Sensors and Actuators, Communication devices and Power supply Unit.

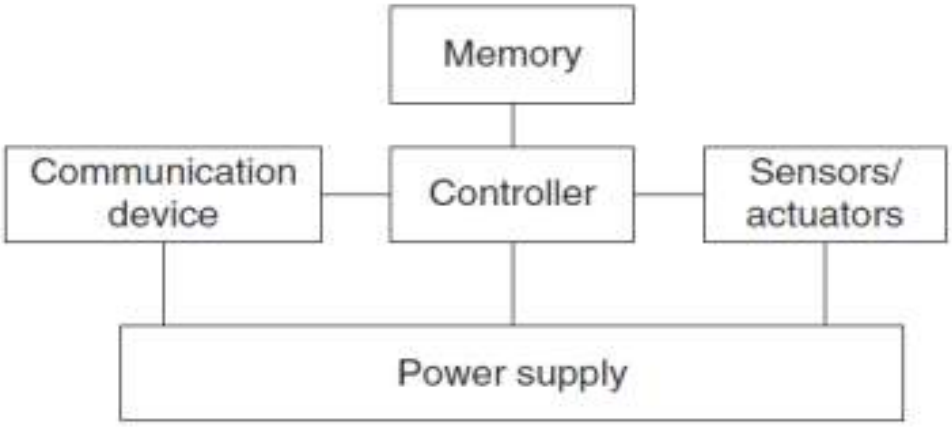


Figure 2: Single Node Architecture

A controller is used to process all the relevant data, capable of executing arbitrary code. 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 has to execute various programs, ranging from time critical signal processing and communication protocols to application programs; it is the Central Processing Unit (CPU) of the node.For General-purpose processors applications microcontrollers are used.

These are highly overpowered, and their energy consumption is excessive. These are used in embedded systems. key characteristics of microcontrollers are particularly suited to embedded systems are their flexibility in connecting with other devices like sensors and they are also convenient in that they often have memory built in. In a wireless sensor node, DSP could be used to process data coming from a simple analog, wireless communication device to extract a digital data stream. In broadband wireless communication, DSPs are an appropriate and successfully used platform.DSP-specifically geared, with respect to their architecture and their instruction set, for processing large amounts of vectorial data, as is typically the case in signal processing applications.Memory -to store programs and intermediate data.Different types of memory are used for programs and data.

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. Program code can be stored in Read-Only Memory (ROM) or, more typically, in Electrically Erasable Programmable Read-Only Memory (EEPROM) or flash memory (the later being similar to EEPROM but allowing data to be erased or written in blocks instead of only a byte at a time). Flash memory can also serve as intermediate storage of data in case

RAM is insufficient or when the power supply of RAM should be shut down for some time.Turning nodes into a network requires a device for sending and receiving inform.

Choice of transmission medium:

The communication device is used to exchange data between individual nodes. In some cases, wired communication can actually be the method of choice and is frequently applied in many sensor networks. The case of wireless communication is considerably more interesting because it include radio frequencies. Radio Frequency (RF)- based communication, best fits the requirements of most WSN applications.

Transceivers:

For Communication, both transmitter and receiver are required in a sensor node to convert a bit stream coming from a microcontroller and convert them to and from radio waves. For two tasks a combined device called transceiver is used over a wireless channel. Transceiver structure has two parts as Radio Frequency (RF) front end and the baseband part. The radio frequency front end performs analog signal processing in the actual radio frequency Band. The baseband processor performs all signal processing in the digital domain and communicates with a sensor node's processor or other digital circuitry.

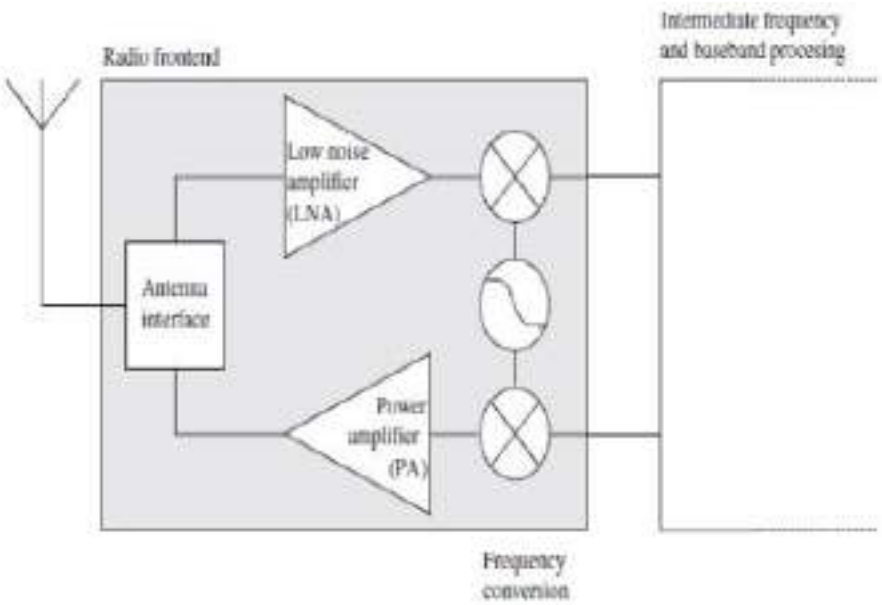


Figure 3:RF at front End

The Power Amplifier (PA) accepts upconverted signals from the IF or baseband part and amplifies them for transmission over the antenna. The Low Noise Amplifier (LNA) amplifies incoming signals up to levels suitable for further processing without significantly reducing the SNR. The range of powers of the incoming signals varies from very weak signals from nodes close to the reception boundary to strong signals from nearby nodes; this range can be up to 100 dB. Elements like local oscillators or voltage-controlled oscillators and mixers are used for frequency conversion from the RF spectrum to intermediate frequencies or to the baseband. The incoming signal at RF frequencies fRF is multiplied in a mixer with a fixedfrequency signal from the local oscillator (frequency fLO). The resulting

intermediate frequency signal has frequency fLO − fRF. Depending on the RF front end architecture, other elements like filters are also present

Transceiver tasks and Characteristics

Service to upper layer: A receiver has to offer certain services to the upper layers, 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: The simplest interpretation of energy efficiency is the energy required to transmit and receive a single bit

Carrier frequency and multiple channels: Transceivers are available for different carrier frequencies; evidently, it must match application requirements and regulatory restrictions. State change times and energy: A transceiver can operate in different modes: sending or receiving, use different channels, or be in different power-safe states.Data rates: Carrier frequency and used bandwidth together with modulation and coding determine the gross data rate.

- Modulations: The transceivers typically support one or several of on/off-keying, ASK, FSK, or similar modulations.
- Coding: Some transceivers allow various coding schemes to be selected.
- Transmission power control: Some transceivers can directly provide control over the transmission power to be used; some require some external circuitry for that purpose.
- Usually, only a discrete number of power levels are available from which the actual transmission power can be chosen. Maximum output power is usually determined by regulations.
- Noise figure: The noise figure NF of an element is defined as the ratio of the Signal-toNoise Ratio (SNR) ratio SNRI at the input of the element to the SNR ratio SNRO at the element's output.
- It describes the degradation of SNR due to the element's operation and is typically given in dB: NF dB= SNRI dB − SNRO dB
- Gain: The gain is the ratio of the output signal power to the input signal power and is typically given in dB. Amplifiers with high gain are desirable to achieve good energy efficiency.
- Power efficiency: The efficiency of the radio front end is given as the ratio of the radiated power to the overall power consumed by the front end; for a power amplifier, the efficiency describes the ratio of the output signal's power to the power consumed by the overall power amplifier.
- 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.
- Range: The range of a transmitter is clear. The range is considered in absence of interference; it evidently depends on the maximum transmission power, on the antenna characteristics.
- Blocking performance: The blocking performance of a receiver is its achieved bit error rate in the presence of an interferer
- Out of band emission: The inverse to adjacent channel suppression is the out of band emission of a transmitter. To limit disturbance of other systems, or of the WSN itself in a multichannel setup, the transmitter should produce as little as possible of transmission power outside of its prescribed bandwidth, centered around the carrier frequency
- Carrier sense and RSSI: In many medium access control protocols, sensing whether the wireless channel, the carrier, is busy (another node is transmitting) is a critical information.

- The receiver has to be able to provide that information. the signal strength at which an incoming data packet has been received can provide useful information a receiver has to provide this information in the Received Signal Strength Indicator (RSSI).
- Frequency stability: The frequency stability denotes the degree of variation from nominal center frequencies when environmental conditions of oscillators like temperature or pressure change.
- Voltage range: Transceivers should operate reliably over a range of supply voltages. Otherwise, inefficient voltage stabilization circuitry is required.

1.6 Sensors and Actuators

- The actual interface to the physical world: devices that can observe or control physical parameters of the environment.Sensors can be roughly categorized into three categories as
- Passive, omnidirectional sensors: These sensors can measure a physical quantity at the point of the sensor node without actually manipulating the environment by active probing – in this sense, they are passive. Moreover, some of these sensors actually are self-powered in the sense that they obtain the energy they need from the environment – energy is only needed to amplify their analog signal.
- Passive, narrow-beam sensors: These sensors are passive as well, but have a well defined notion of direction of measurement.
- Active sensors: This last group of sensors actively probes the environment, for example, a sonar or radar sensor or some types of seismic sensors, which generate shock waves by small explosions. These are quite specific – triggering an explosion is certainly not a lightly undertaken action – and require quite special attention. Actuators are just about as diverse as sensors

Purposes of designing a WSN - converts electrical signals into physical phenomenon.As usually no tethered power supply is available, some form of batteries are necessary to provide energy. Sometimes, some form of recharging by obtaining energy from the environment is available as well (e.g. solar cells). There are essentially two aspects: Storing energy and Energy scavenging.Traditional batteries: The power source of a sensor node is a battery, either nonrechargeable (“primary batteries”) or, if an energy scavenging device is present on the node, also rechargeable (“secondary batteries”).

Requirements of Battery

- Capacity: They should have high capacity at a small weight, small volume, and low price. The main metric is energy per volume, J/cm3.
- 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. Zinc-air batteries, for example, have only a very short lifetime (on the order of weeks).
- Efficient recharging: Recharging should be efficient even at low and intermittently available recharge power
- Relaxation: Their relaxation effect – the seeming self-recharging of an empty or almost empty battery when no current is drawn from it, based on chemical diffusion processes within the cell – should be clearly understood. Battery lifetime and usable capacity is considerably extended if this effect is leveraged.
- DC–DC Conversion: Unfortunately, batteries alone are not sufficient as a direct power source for a sensor node. One typical problem is the reduction of a battery’s voltage as its capacity drops.
- DC – DC converter can be used to overcome this problem by regulating the voltage delivered to the node’s circuitry. To ensure a constant voltage even though the battery’s supply voltage drops, the DC

– DC converter has to draw increasingly higher current from the battery when the battery is already becoming weak, speeding up battery death.

- The DC – DC converter does consume energy for its own operation, reducing overall efficiency

Energy Scavenging

- 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 are conceivable options.
- Photovoltaics: The well-known solar cells can be used to power sensor nodes. The available power depends on whether nodes are used outdoors or indoors, and on time of day and whether for outdoor usage.
- The resulting power is somewhere between 10 μW/cm2 indoors and 15 mW/cm2 outdoors. Single cells achieve a fairly stable output voltage of about 0.6 V (and have therefore to be used in series) as long as the drawn current does not exceed a critical threshold, which depends on the light intensity. Hence, solar cells are usually used to recharge secondary batteries.
- Temperature gradients: Differences in temperature can be directly converted to electrical energy.
- Vibrations: One almost pervasive form of mechanical energy is vibrations: walls or windows in buildings are resonating with cars or trucks passing in the streets, machinery often has low frequency vibrations. both amplitude and frequency of the vibration and ranges from about 0.1 μW/cm3 up to 10, 000 μW/cm3 for some extreme cases. Converting vibrations to electrical energy can be undertaken by various means, based on electromagnetic, electrostatic, or piezoelectric principles.
- Pressure variations: Variation of pressure can also be used as a power source.
- Flow of air/liquid: Another often-used power source is the flow of air or liquid in wind mills or turbines. The challenge here is again the miniaturization, but some of the work on millimeter scale MEMS gas turbines might be reusable.

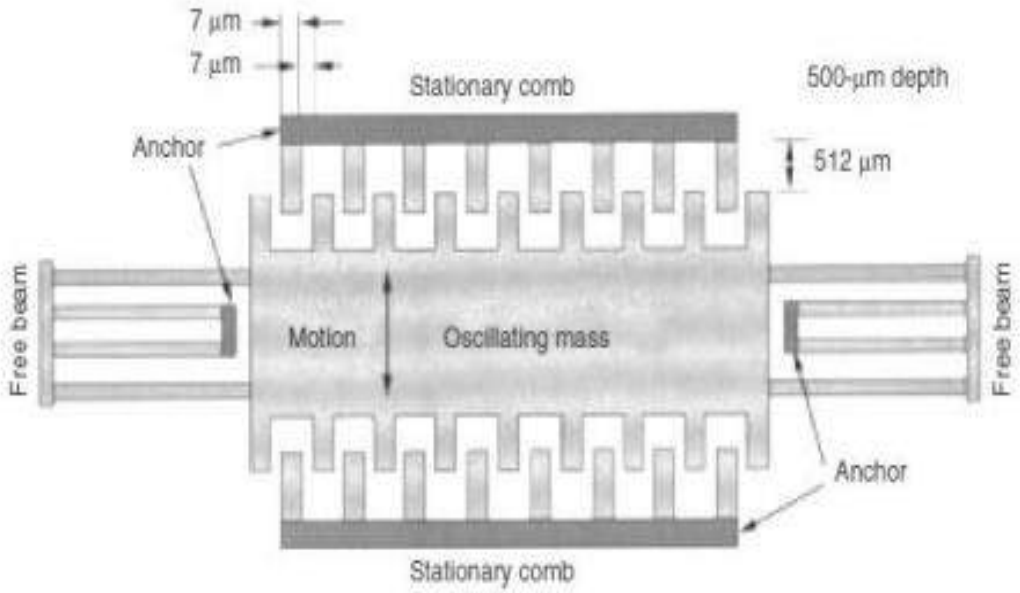


Figure 4: MEMS device for converting vibrations to electrical energy, based on a variable capacitor

SENSOR NETWORK SCENARIOS

- Source is any unit in the network that can provide information (sensor node). A sink is the unit where information is required, it could belong to the sensor network or outside this network to interact with another network or a gateway to another larger Internet.

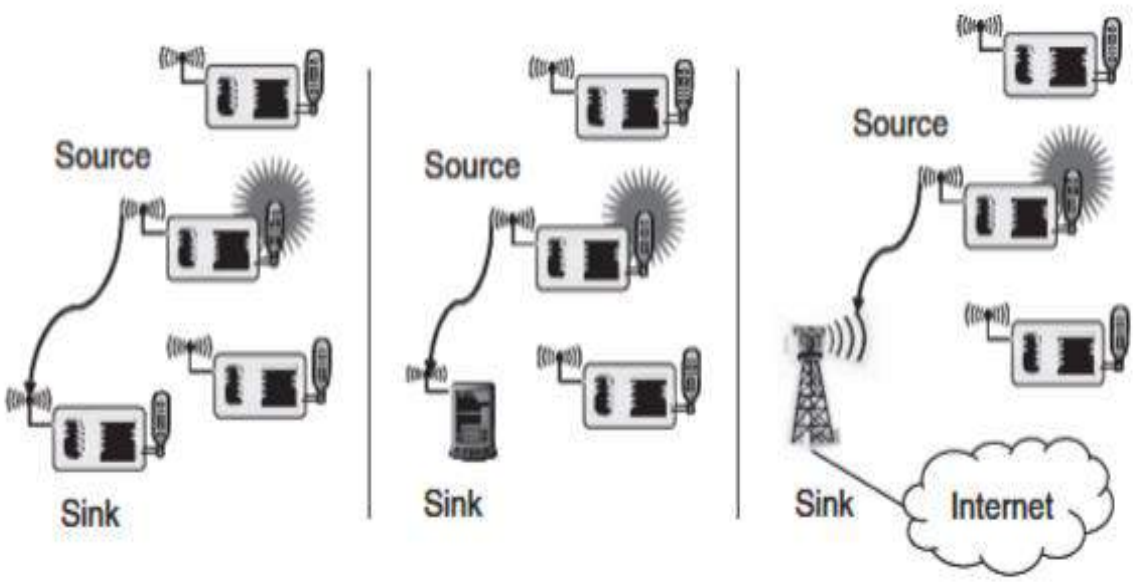


Figure 5:Sink Node in Network

Single-hop versus multi-hop Networks

Because of limited distance the direct communication between source and sink is not always possible. In WSNs, to cover a lot of environment the data packets taking multi hops from source to the sink. To overcome such limited distances it better to use relay stations. Depending on the particular application of having an intermediate sensor node at the right place is high. Multi-hopping also to improves the energy efficiency of communication as it consumes less energy to use relays instead of direct communication, the radiated energy required for direct communication over a distance d is cd^{α} (c some constant, $\alpha \geq 2$ the path loss coefficient) and using a relay at distance d/2 reduces this energy to $2c(d/2)^{\alpha}$ .This calculation considers only the radiated energy. It should be pointed out that only multihop networks operating in a store and forward fashion are considered here. In such a network, a node has to correctly receive a packet before it can forward it somewhere. Cooperative relaying (reconstruction in case of erroneous packet reception) techniques are not considered here.

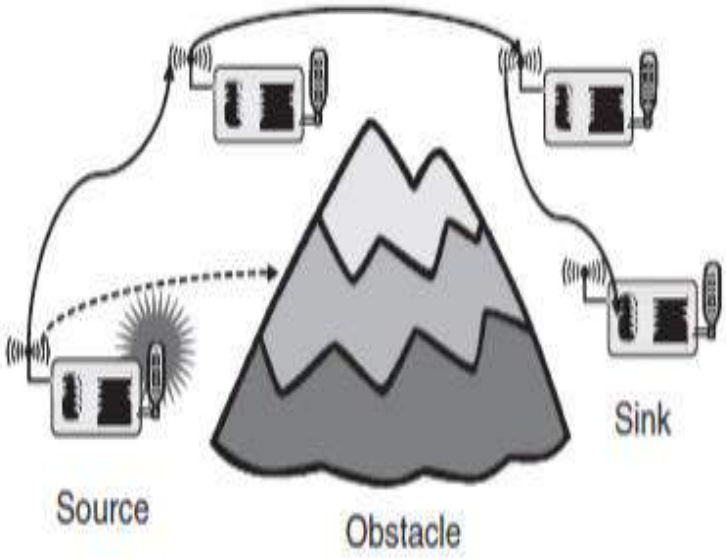


Figure 6: Single-hop versus multi-hop Networks

Multiple sinks and sources

- Multiple sources should send information to multiple sinks.
- Either all or some of the information has to reach all or some of the sinks.

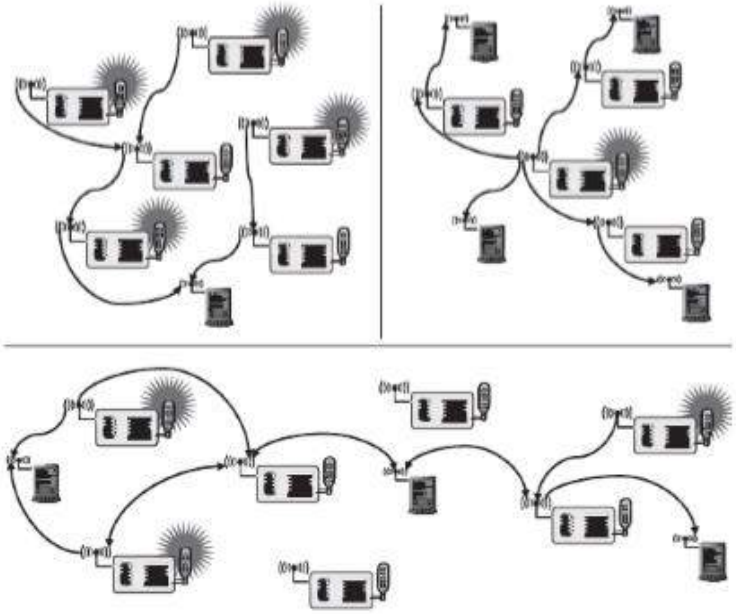


Figure 7: Multiple sinks and sources

Types of Mobility

All participants were stationary. But one of the main virtues of wireless communication is its ability to support mobile participants.In wireless sensor networks, mobility can appear in three main forms

a. Node mobility

b. Sink mobility

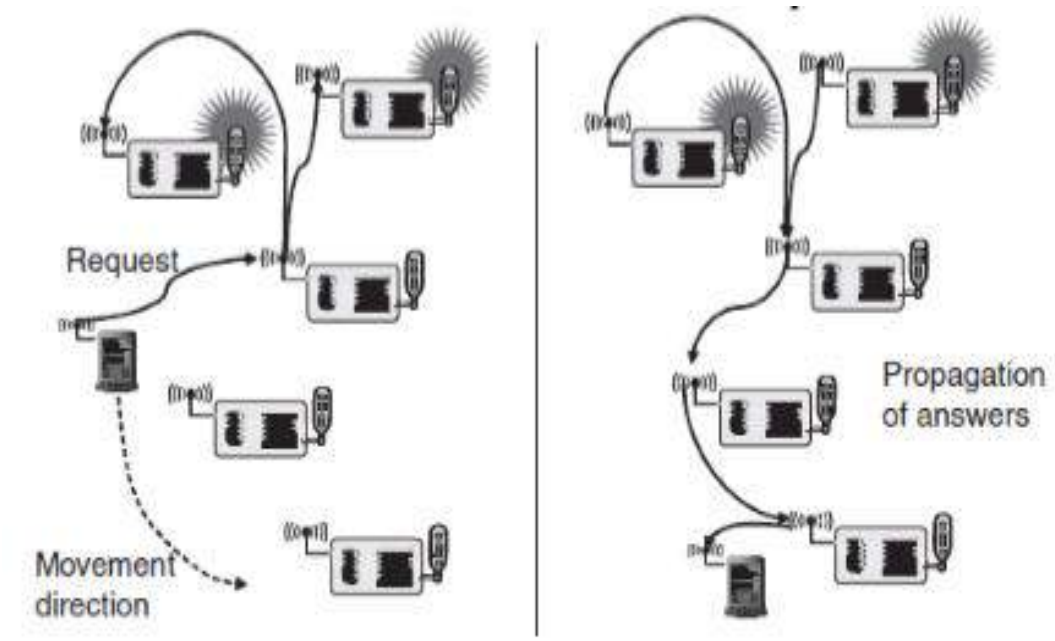
c. Event mobility

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 to function correctly.

Sink Mobility: The information sinks can be mobile. 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, In many cases, consecutive interactions can be treated as separate, unrelated requests.

Event Mobility: In tracking applications, the cause of the events or the objects to be tracked can be mobile. In such scenarios, it is (usually) important that the observed event is covered by a sufficient number of sensors at all time. 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 detect a moving elephant and to observe it as it moves around

Sink mobility: A mobile sink moves through a sensor network as information is being retrieved on its behalf . Area of sensor nodes detecting an event – an elephant– that moves through the network along with the event source (dashed line indicate the elephant's trajectory; shaded ellipse the activity area following or even preceding the elephant)



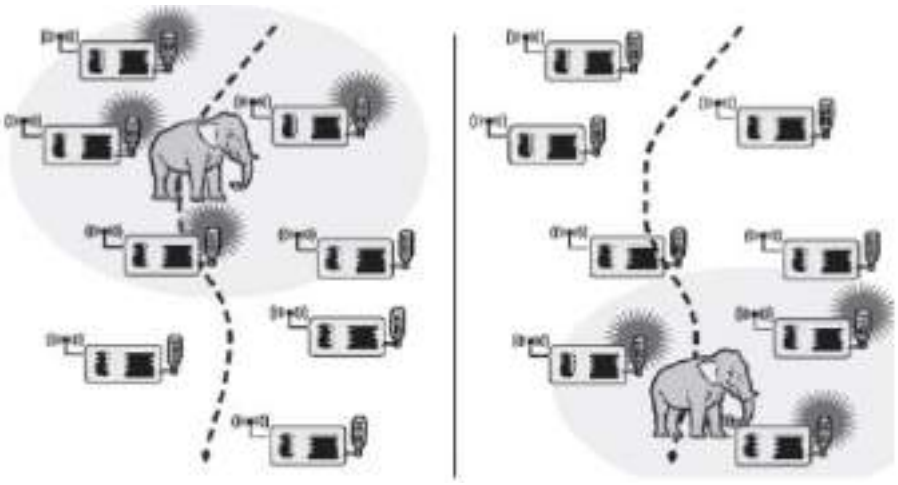


Figure 8, 9: Mobility

Data Aggregation

The nodes which are in same radio range may sense the redundant data and transmits the same to sink node. Then it is a challenging for the sink node to manage such large amount of data. This problem can be solved by a data driven approach called “Data Aggregation”. The approach data aggregation is the power-saving mechanism. It is the process of combining the data coming from various sources and en route them after removing redundancy, such as to improve overall network lifetime. This can significantly help to reduce the consumption by eliminating redundant data.The functionality of data aggregation is performed continuously in order to improve the bandwidth and energy utilization, but it may impact badly on other performance metrics such as delay, accuracy, fault tolerance, etc. However the objective of the data aggregation is to eliminate the redundant data transmission and improves the network lifetime.

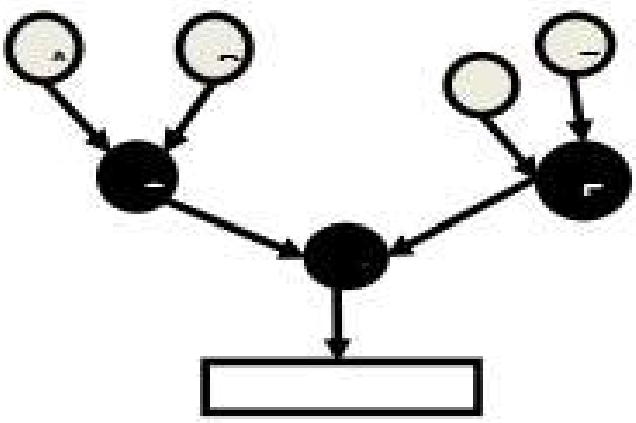


Figure 10: Data Aggregation

Data Aggregation Strategies

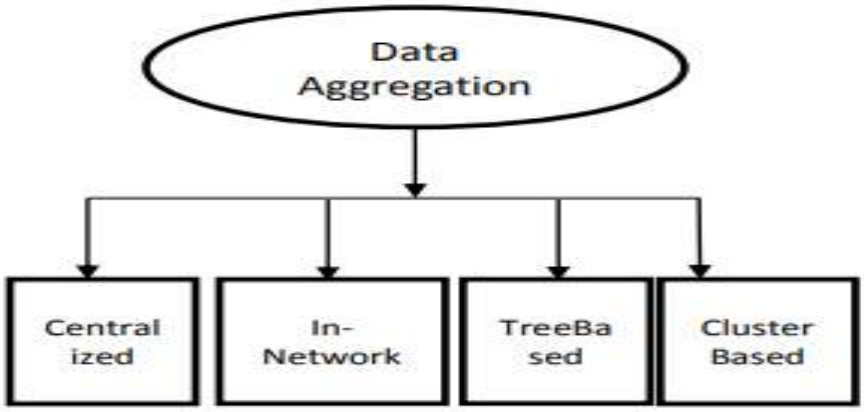


Figure 11: Data Aggregation Strategies

Tree Based Approach

In this approach, a Data Aggregation Tree (DAT) is framed and here for each data transmission a minimum spanning tree is constructed. Each node in a network has a parent-child relationship in which the data is forwarded in a bottom-up approach. The data starts flowing from leaf nodes to the sink node and the aggregation of the data is done by parent nodes in the network.

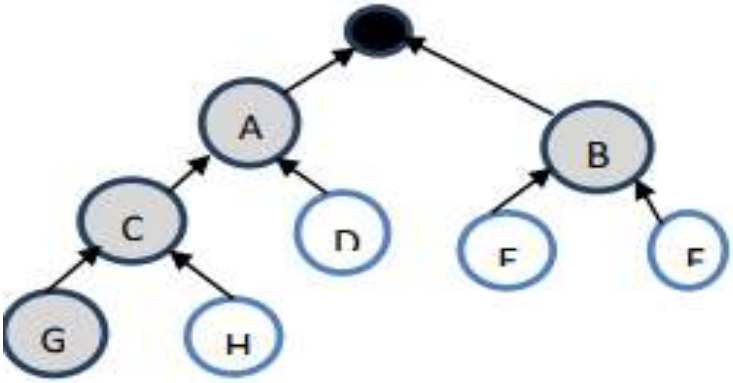


Figure 12: Tree Based Approach

Centralized Approach

In this approach each sensor node sends its sensed data to a central node (base station) via the shortest possible route. All the sensor nodes simply sends the data packets to a node, which is the powerful among all other nodes This node is called aggregator node or header node. This node aggregates the data coming from other nodes and the resultant data will be sent as a single packet.

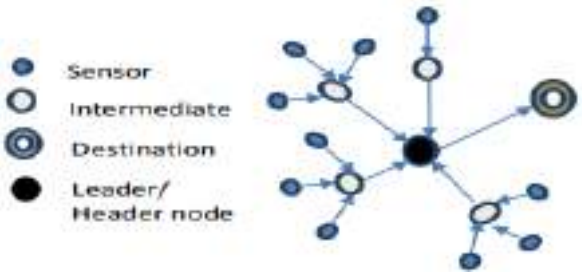


Figure 13: Centralized Approach

In-Network Approach

It is a global approach for gathering and processing the data at intermediate nodes and routing the information through a multi-hop network. The main of this approach is to reducing power consumption.There are two types of in-network aggregation: 1. With Size Reduction: Here the size of the packet to be transmitted to the sink node is reduced by combining and compressing the data packets received by sensor node from its neighbors. 2. Without Size reduction: Here, without processing the value of data the packets from the different neighboring nodes are merged into a single packet.

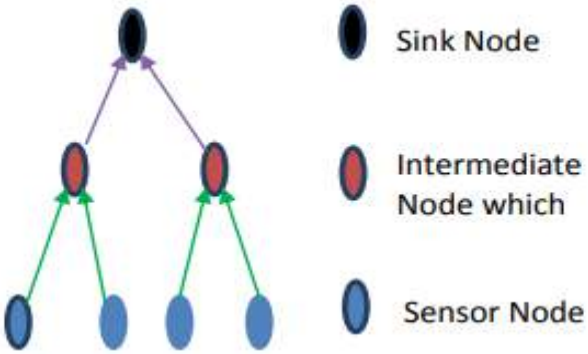


Figure 14: In-Network Approach

Cluster Based Approach

Here the whole network is split into several clusters. Each cluster is consisting of many sensor nodes. Cluster head is selected among the sensor nodes within a cluster. The aggregator role is performed by the Cluster head which aggregates the data received and send to the sink. By this approach the bandwidth overhead is minimized as total number of packets to be transmitted are less. Several clusters based approaches for data collection have been proposed for WSN. Clustering reduces direct transmission to the base station by in network data aggregation as well as decreases energy consumption by reducing the transmitting distance. Better aggregation for large number of nodes is provided by Hierarchical Clustering

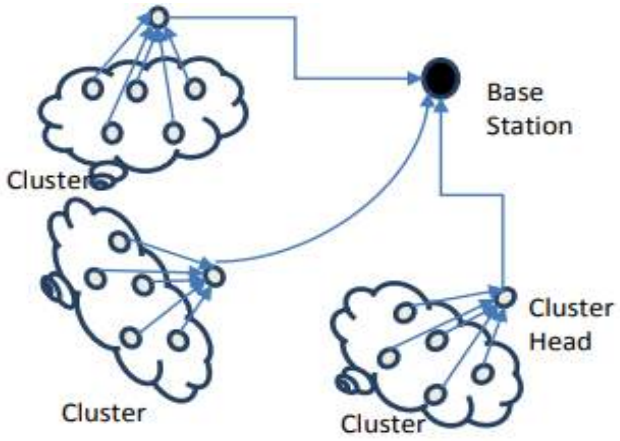


Figure 15: Cluster Based Approach

Routing Protocols in WSNs

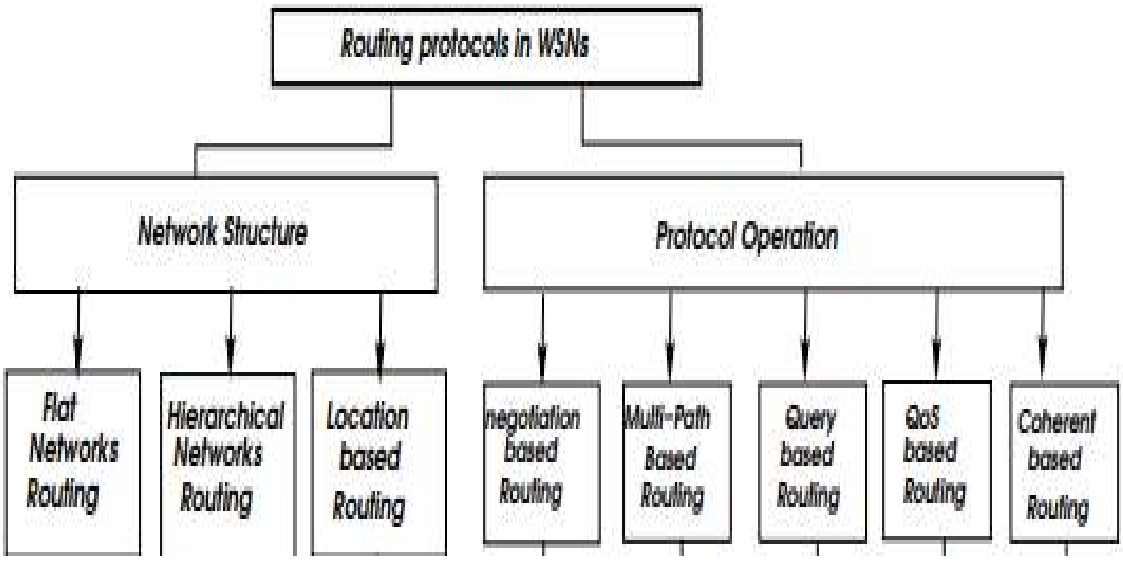


Figure 16: Routing Protocols in WSNs

Optimization Techniques for Routing in Wireless Sensor Networks

Attribute-based

The sink sends queries to certain regions and waits for the response from the sensors located in this area. Following an attribute-value scheme, the queries inform about the required data. The selection of the attributes depends on the application. An important characteristic of these schemes is that the content of the data messages is analyzed in each hop to make decisions about routing. Multiple routes can communicate a node and the sink. The aim of energy-aware algorithms is to select those routes that are expected to maximize the network lifetime. To do so, the routes composed of nodes with higher energy resources are preferred.Wireless sensor networks are formed by a significant number of nodes so the manual assignation of unique identifiers is infeasible. The use of the MAC address or the GPS coordinates is not recommended as it introduces a significant payload However, network-wide unique addresses are not needed to identify the destination node of a specific packet in wireless sensor networks. In fact, attribute-based addressing fits better with the specificities of wireless sensor networks. In this case, an attribute such as node location and sensor type is used to identify the final destination. Concerning these identifiers, two different approaches .

Firstly, the ID reuse scheme allows identifiers to be repeated in the network but keeping their uniqueness in close areas. In this way, a node knows that its identifier is unique in a k-hop neighborhood, being k a parameter to configure.

On the other hand, the field-wide unique ID schemes guarantee that the identifiers are unique in the whole application. With this assumption, other protocols such as routing, MAC or network configurations can be simultaneously used. Node decides the transmission route according to the localization of the final destination and the positions of some other nodes in the network.

Multipath Communication

- Nodes use multiple paths from an origin to a destination in the network. As multipath communications are intended to increase the reliability and the performance of the network, these paths should not share any link. Multipath communications can be accomplished in two ways. First, one path is established as the active communication routing while the other paths are stored for future need, i.e. when the current active path is broken. On the other hand, it is also possible to distribute the traffic among the multiple paths. The network application business and its functionalities prompt the need for ensuring a QoS (Quality of Service) in the data exchange.
- In particular, effective sample rate, delay bounded and temporary precision are often required. Satisfying them is not possible for all the routing protocols as the demands may be opposite to the protocol principles.
- For instance, a routing protocol could be designed to extend the network lifetime while an application may demand an effective sample rate which forces periodic transmissions and, in turn, periodic energy consumptions.

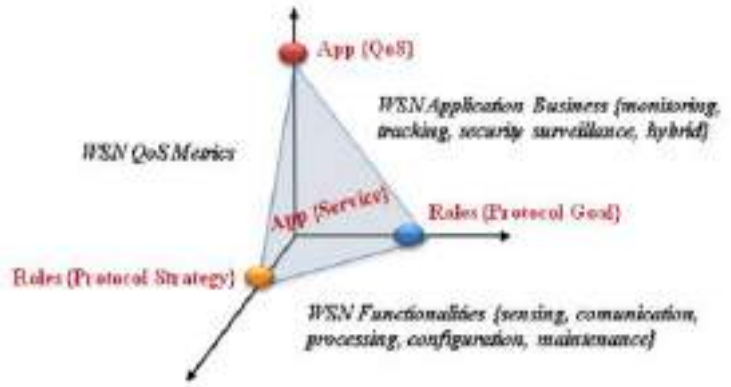


Figure 17: WSN Vs QOS Metrics

Significance for Study of Energy in Wireless Sensor Networks

To evaluate the network performance, consider parameters that evidence proper network operation directly influencing the energy consumption of each node. There are local and global parameters. Global parameters display the total energy costs for the network considering each type of energy for each specific activity. In contrast, local parameters provide total energy consumption rates for a single node. This energy depends on the location of the node within the topology regardless of how near or far they are located from the coordinator node and how much traffic is transmitted through it

An energy-efficient routing protocol decreases the consumption of the nodes by routing data through paths that display the least amount of energy. There are some special mechanisms to achieve this goal such as optimization of jumps to the destination node, maintenance of optimal and valid routes, reduction of transmission delays, and reduction of packet retransmissions and attempts to listen to the channel.Concerning the communication channel, it is a factor that significantly influences the energy consumption because the protocol executes a series of listening attempts to determine whether the channel is already busy with other information packets. The carrier senses multiple accesses with the collision avoidance (CSMA/CA) protocol.first, a node begins listening to the wireless channel and if it is free, the node begins transmitting. If the wireless channel is not free, the node recalculates a random delay, waits, and listens again. MAC-level protocol is used for all extensions of 802.15.4 (including the original version), which is the CSMA/CA that guarantees a high data rate.

A network recognition is being carried out at all times to check the status of the channel (carrier detection). Only when free, data can be sent. In the 802.11 standard, the physical layer polls the energy level over the radio frequency to determine whether or not there is transmission. If the channel is busy, a random timer starts (with a maximum of five back off periods), the timer only discovers time with free channel, transmits when it expires, and finally, if it does not receive ACK, it increases the back off. This metric is known as CSMA/CA retries. If these CSMA/CA retries are frequent, the channel is busy most of the time. Consequently, there might be several collisions due to overload. In addition, when the wireless channel is permanently busy with information packets, there are many collisions and retransmissions of packets. This fact influences energy consumption because the nodes spend more time and capacity retransmitting over and over.In a network layer, overloads are an important factor that influence energy consumption.

The efficiency of the routing protocol may also be measured by the number of packets the protocol needs to route to its destination. A protocol with many control packets will contribute to packet

collisions and overall performance reduction. In terms of route discovery, in all the protocols considered, the nodes exhibit capacity to know their neighbors.

Network energy consumption is directly related to the complexity in the administration of routing or neighbor tables. As sensors execute huge routing processes, energy consumption increases if these routes have not been properly updated. This is why it is also important to assess route delays; they are directly related to the number of jumps that a node takes to reach a destination.

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Unit- II- WIRELESS SENSOR NETWORKS-SCSA5202

Unit 2

MAC LAYER

MAC Layer Strategies: MAC Layer Protocols-Scheduling Sleep Cycles-Energy Management-Contention Based ProtocolsSchedule Based Protocols, 802.15.4 Standard. Naming and Addressing: Addressing Services - Publish-Subscribe Topologies. Clock Synchronization: Clustering For Synchronization-Sender-Receiver-Receiver Synchronization-Error Analysis. Power Management – Per Node -System-Wide-Sentry Services-Sensing Coverage

The wireless medium being inherently broadcast in nature and hence prone to interferences requires highly optimized medium access control (MAC) protocols. The prime role of the MAC is to coordinate access to and transmission over a medium common to several nodes

Issues in designing MAC protocol for Sensor networks

1. Bandwidth Efficiency

It is defined as the ratio of the bandwidth utilized for data transmission to the total available bandwidth. Bandwidth must be utilized in efficient manner. Control-overhead must be kept as minimal as possible.

2. Quality of Service support

This is essential for supporting time-critical traffic-sessions. • The protocol should have resource reservation mechanism that takes into considerations. 1) Nature of wireless-channel and 2) Mobility of nodes

3. Synchronization • This is very important for bandwidth (time-slot) reservation by nodes. • The protocol must consider synchronization between nodes in the network. • Exchange of control-packets may be required for achieving timesynchronization among nodes.

4. Hidden and Exposed Terminal Problems • The hidden-terminal problem refers to the collision of packets at a receivingnode due to the simultaneous transmission of those nodes that are not within the direct transmission-range of the sender but are within the transmissionrange of the receiver. • Collision occurs when both nodes transmit packets at the same time without knowing about the transmission of each other. • In figure, S1 and S2 are hidden from each other & they transmit simultaneously to R1 which leads to collision. • 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. • If S1 is already transmitting to R1, then S3 cannot interfere with on-going transmission & it cannot transmit to R2. • Hidden & exposed-terminal problems reduce the throughput of a network when traffic load is high. 5. Error-prone Shared Broadcast Channel • When a node is receiving data, no other node in its neighborhood (apart from the sender) should transmit. • A node should get access to the shared medium only when its transmission do not affect any ongoing session. • The protocol should grant channel access to nodes in such a manner that collisions are minimized. • Protocol should ensure fair bandwidth allocation. 6. Error-prone Shared Broadcast Channel • When a node is receiving data, no other node in its neighborhood (apart from the sender) should transmit. • A node should get access to the shared medium only when its transmission do not affect any ongoing session. • The protocol should grant channel access to nodes in such a manner that collisions are minimized. • Protocol should ensure fair bandwidth allocation. 7. Distributed Nature • There is no central point of coordination due to the mobility of the nodes. • Nodes must be scheduled in a distributed fashion for gaining access to the channel. 8. Mobility of Nodes • Nodes are mobile most of the time. • The protocol design must take

this mobility factor into consideration so that the performance of the system is not affected due to node mobility.

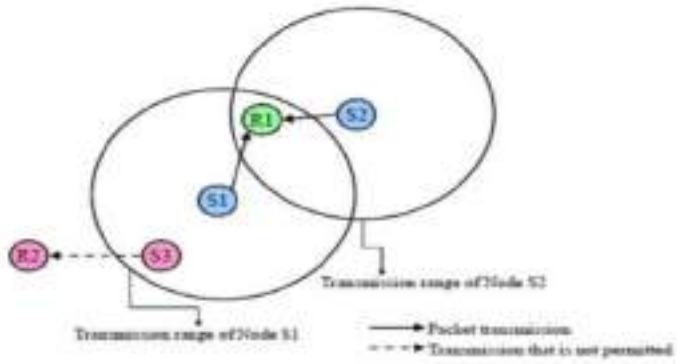


Figure 1: Hidden and Exposed Terminal

MAC Layer Protocols

There are two main categories of MAC protocols for WSNs, according to how the MAC manages when certain nodes can communicate on the channel:

- Time-division multiple access (TDMA) based: These protocols assign different time slots to nodes. Nodes can send messages only in their time slot, thus eliminating contention. Examples of this kind of MAC protocols include LMAC, TRAMA, etc.
- Carrier-sense multiple access (CSMA) based: These protocols use carrier sensing and backoffs to avoid collisions, similarly to IEEE 802.11. Examples include B-MAC, SMAC, TMAC, X-MAC.

B-MAC

B-MAC (short for Berkeley MAC) is a widely used WSN MAC protocol; it is part of TinyOS. It employs low-power listening (LPL) to minimize power consumption due to idle listening. Nodes have a sleep period, after which they wake up and sense the medium for preambles (clear channel assessment - CCA.) If none is detected, the nodes go back to sleep. If there is a preamble, the nodes stay awake and receive the data packet after the preamble. If a node wants to send a message, it first sends a preamble for at least the sleep period in order for all nodes to detect it. After the preamble, it sends the data packet. There are optional acknowledgments as well. After the data packet (or data packet + ACK) exchange, the nodes go back to sleep. Note that the preamble doesn't contain addressing information. Since the recipient's address is contained in

the data packet, all nodes receive the preamble and the data packet in the sender's communication range (not just the intended recipient of the data packet.)

X-MAC

X-MAC is a development on B-MAC and aims to improve on some of B-MAC's shortcomings. In B-MAC, the entire preamble is transmitted, regardless of whether the destination node awoke at the beginning of the preamble or the end. Furthermore, with B-MAC, all nodes receive both the preamble and the data packet. X-MAC employs a strobed preamble, i.e. sending the same length preamble as B-MAC, but as shorter bursts, with pauses in between. The pauses are long enough that the destination node can send an acknowledgment if it is already awake. When the sender receives the acknowledgment, it stops sending preambles and sends the data packet. This mechanism can save time because potentially, the sender doesn't have to send the whole length preamble. Also, the preamble contains the address of the destination node. Nodes can wake up, receive the preamble, and go back to sleep if the packet is not addressed to them. These features improve B-MAC's power efficiency by decreasing nodes' time spent in idle listening.

LMAC

LMAC (short for lightweight MAC) is a TDMA-based MAC protocol. There are data transfer timeframes, which are divided into time slots. The number of time slots in a timeframe is configurable according to the number of nodes in the network. Each node has its own time slot, in which only that particular node can transmit. This feature saves power, as there are no collisions or retransmissions. A transmission consists of a control message and a data unit. The control message contains the destination of the data, the length of the data unit, and information about which time slots are occupied. All nodes wake up at the beginning of each time slot. If there is no transmission, the time slot is assumed to be empty (not owned by any nodes), and the nodes go back to sleep. If there is a transmission, after receiving the control message, nodes that are not the recipient go back to sleep. The recipient node and the sender node goes back to sleep after receiving/sending the transmission. Only one message can be sent in each time slot. In the first five timeframes, the network is set up and no data packets are sent. The network is set up by nodes claiming a time slot. They send a control message in the time slot they want to reserve. If there are no collisions, nodes note that the time slot is claimed. If there are multiple nodes trying to claim the same time slot, and there is a collision, they randomly choose another unclaimed time slot.

The INET implementations

The three MACs are implemented in INET as the BMac, XMac, and LMac modules. They have parameters to adapt the MAC protocol to the size of the network and the traffic intensity, such as slot time, clear channel assessment duration, bitrate, etc. The parameters have default values, thus the MAC modules can be used without setting any of their parameters. Check the NED files of the MAC modules (BMac.ned, XMac.ned, and LMac.ned) to see all parameters.

The MACs don’t have corresponding physical layer models. They can be used with existing generic radio models in INET, such as UnitDiskRadio or ApskRadio.

Configuration

The showcase contains three example simulations, which demonstrate the three MACs in a wireless sensor network. The scenario is that there are wireless sensor nodes in a refrigerated warehouse, monitoring the temperature at their location. They periodically transmit temperature data wirelessly to a gateway node, which forwards the data to a server via a wired connection.

Note that in WSN terminology, the gateway would be called sink. Ideally, there should be a specific application in the gateway node called sink, which would receive the data from the WSN, and send it to the server over IP. Thus the node would act as a gateway between the WSN and the external IP network. In the example simulations, the gateway just forwards the data packets over IP.

Figure 2:Sensor sending data to gateway

In the network, the wireless sensor nodes are of the type SensorNode, named sensor1 up to sensor4, and gateway. The node named server is a StandardHost. The network also contains an Ipv4NetworkConfigurator, an IntegratedVisualizer, and an ApskScalarRadioMedium module. The nodes are placed against the backdrop of a warehouse floorplan. The scene size is 60x30 meters. The warehouse is just a background image providing context. Obstacle loss is not modeled, so the background image doesn’t affect the simulation in any way. Routes are set up according to a star topology, with the gateway at the center. This is achieved by dumping the full configuration of Ipv4NetworkConfigurator (which was generated with the configurator’s default settings), and then modifying it. The modified configuration is in the config.xml file. The following image shows the routes:

- The fewer bits per address, the better
- Global > Network-wide > Local
- Tradeoffs
    - Address length ↔ management overhead
- Typically, address negotiation runs only at the beginning
    - Except when there is mobility

Distributed Address Assignment

- Option 1: Random assignment
    - Unacceptable high risk of duplicate addresses
    - No-conflict probability for n addresses and k nodes is
    - $$P(n,k) = 1 \cdot \frac{n-1}{n} \cdot \ldots \cdot \frac{n-k+1}{n} = \frac{1}{n^k} \cdot \frac{n!}{(n-k)!} = \frac{k!}{n^k} \cdot \binom{n}{k}$$
    - By Stirlings approximation
    - $$P(n,k) \approx e^{-k} \cdot \left(\frac{n}{n-k}\right)^{(n-k)+1/2}$$
    - Similar to the birthday paradox
- Option 2: Still random, but avoid addresses used in local neighborhood
    - By overhearing exchanged packets
    - Good enough in many WSN apps where data sent to a certain sink
- Option 3: Repair any observed conflicts
    - Randomly pick a temporary address and a proposed fixed address
    - Send an address request to the proposed address, using temporary address
    - If address reply arrives, address already exists
    - Collisions in temporary address unlikely, as only used briefly
- Option 4: Similar to 3, but use a neighbor that already has a fixed address to perform requests

Issues with Asymmetric Links

- Assume nodes communicate with bidirectional neighbors only

- All bidirectional neighbors of each node must have distinct addresses
- The address of any inbound neighbor must be different from all bidirectional neighbors

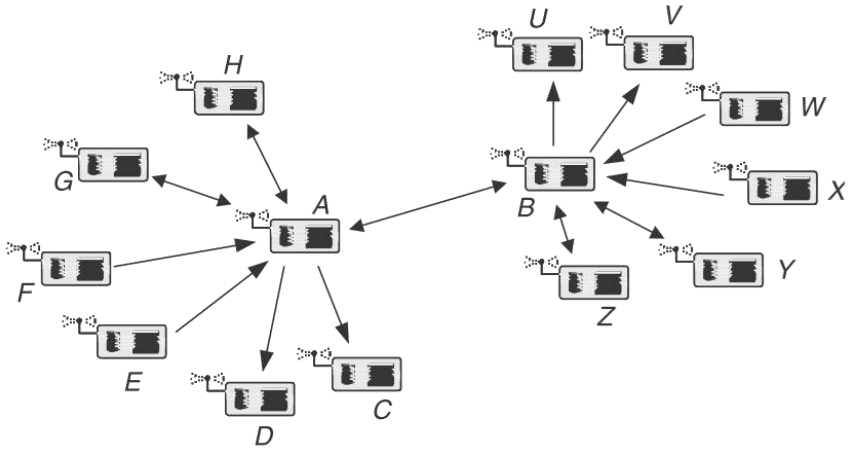


Figure 5:Distributed Addressing

Content-Based Addressing

- Recall: Paradigm change from id-centric to data-centric networking in WSN
- Supported by content-based names/addresses
  - Do not described involved nodes (not known anyway), but the content itself the interaction is about
- Classical option: Put a naming scheme on top of IP addresses
  - Done by some middleware systems

Geographic addressing

- Express addresses by denoting physical position of nodes
  - Considered a special case of content-based addresses
  - Attributes for x and y (and z) coordinates
- Options
  - Single point
  - Circle or sphere centered around given point
  - Rectangle by two corner points
  - Polygon

A Message-Oriented Middleware for Sensor Networks – Mires

In general, it facilitates the development of network-applications over the WSN and providing common application services.Problem: Thousands of sensor nodes and redundant data. Low availability of resources and processing capacity of the sensor nodes.How does it help: Message-oriented which

aggregate data, Multi-Hop routing and greatly reduce the among of transmissions, save lots of energy.Traditional request/response approach is not suitable for event-driven communication model. Publish/subscribe approach is used to query and extract data from the network.In applications: Use in habitat monitoring, object tracking, precision agriculture, building monitoring and military systems.

MIRES Architecture

- Publish/Subscribe service
    - communication between middleware services.
    - Advertising the topics available.
    - Maintaining the list of topics subscribed by the node application
    - Publishing messages.
- Routing
    - Multi-hop routing to the Sink
- 3 types of notification events:
    - TopicArrival,
        - event signals that the node application has submitted data collected from sensors.
    - StateArrival
        - Event signals that data received from the network.
    - TopicSetupArrival
        - the subscribe message broadcasted from the user application.

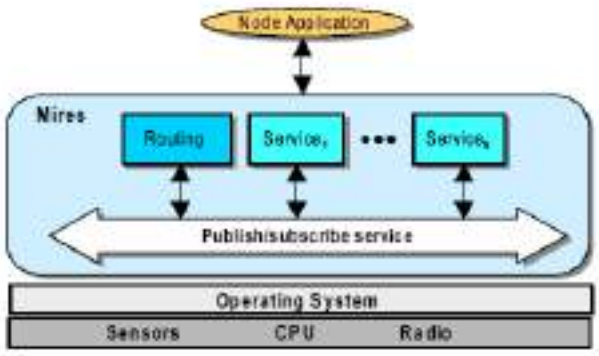


Figure 6:MIRES Architecture

Publish/Subscribe Service

- PublishState interface define the command used by ServiceX to publish their processing results.
- Notifier interface defines 3 events
- MultiHopRouter-route to the sink

- BCast-Boardcast Setup info.

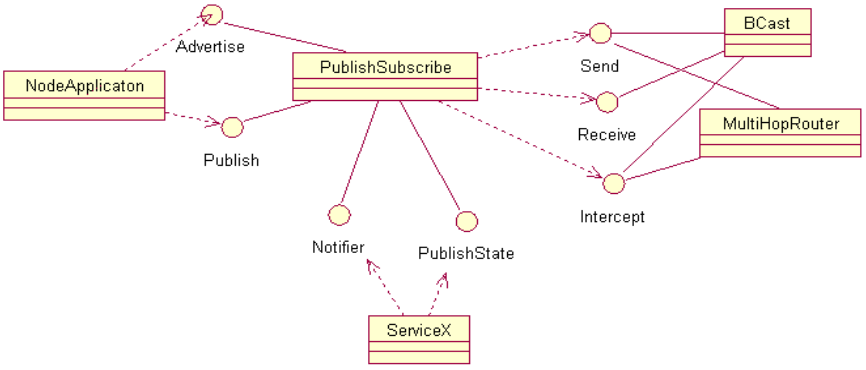


Figure 7: Publish Service

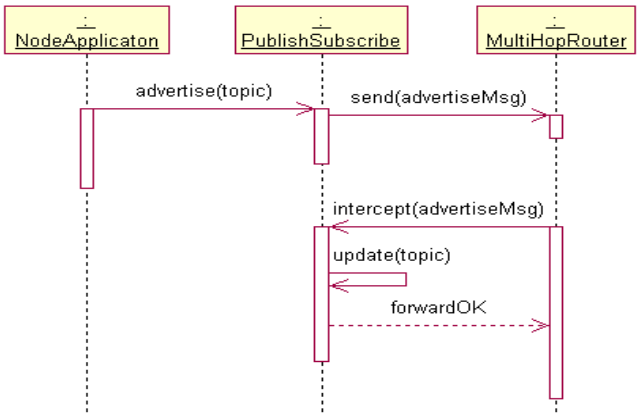


Figure 8: Advertisement

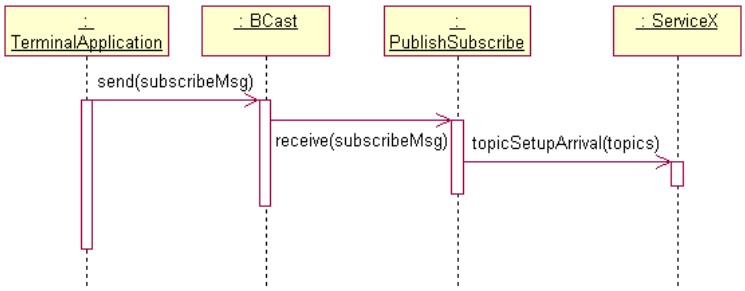


Figure 9: Subscibe Service

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Unit- III- WIRELESS SENSOR NETWORKS-SCSA5202

Unit III

NODE LOCALIZATION AND DATA GATHERING

Node Localization: Absolute and Relative Localization-Triangulation-Multi-Hop Localization and Error Analysis-Anchoring - Geographic Localization-Target Tracking - Localization and Identity Management-Walking GPS-Range Free Solutions. Data Gathering - Tree Construction Algorithms and Analysis - Asymptotic Capacity- Lifetime Optimization Formulations- Storage and Retrieval. Deployment & Configuration - Sensor deployment, scheduling and coverage issues-Self configuration and topology contro

Node Localization

Awareness of location is one of the important and critical issue and challenge in wireless sensor network. Knowledge of Location among the participating nodes is one of the crucial requirements in designing of solutions for various issues related to Wireless sensor networks. Wireless sensor networks are being used in environmental applications to perform the number of task such as environment monitoring, disaster relief, target tracking, defences and many more. In many such tasks, node localization is inherently one of the system parameters. Node localization is required to report the origin of events, assist group querying of sensors, routing and to answer questions on the network coverage. So, one of the fundamental challenges in wireless sensor network is node localization.

PARAMETERS FOR LOCALIZATION

Accuracy: Accuracy is very important in the localization of wireless sensor network. Higher accuracy is typically required in military installations, such as sensor network deployed for intrusion detection. However, for commercial networks which may use localization to send advertisements from neighboring shops, the required accuracy may not be lower. Cost: Cost is a very challenging issue in the localization of wireless sensor network. There are very few algorithms which give low cost but those algorithms don't give the high rate of accuracy. Power: Power is necessary for computation purpose. Power play a major role in wireless sensor network as each sensor device has limited power. Power supplied by battery. Static Nodes: All static sensor nodes are homogeneous in nature. This means that, all the nodes have identical sensing ability, computational ability, and the ability to communicate. We also assume that, the initial battery powers of the nodes are identical at deployment. Mobile Nodes: It is assumed that a few number of GPS enabled mobile nodes are part of the sensor network. These nodes are homogeneous in nature. But, are assumed to have more battery power as compared to the static nodes and do not drain out completely during the localization process. The communication range of mobile sensor nodes are assumed not to change drastically during the entire localization algorithm runtime and also not to change significantly within the reception of four beacon messages by a particular static node

Localization can be roughly divided into two categories: range-based and range-free. Range-based approach uses absolute distance estimate or angle estimate, meaning that a node in a network can measure the distances from itself to the beacons

In contrast, range-free approach means that it is impossible for a node to measure the direct distances from itself to beacons. Only through connectivity and proximity, a node can estimate its regions or areas where it

separately from each other. A balanced performance is crucial for most applications Different sensor deployment strategies can cause very different network topology, and thus different degrees of sensor redundancy A good sensor deployment with sufficient number of sensors which ensures a certain degree of redundancy in coverage so that sensors can rotate between active and sleep modes is required to balance the workload of sensors

Sensor Deployment 1) Sensor Deployment to Achieve 1-Coverage: Given a set of n targets T = {T1, T2,..., Tn} located in u×v region and m sensor nodes S = {S1, S2,..., Sm}, place the nodes such that each target is monitored by at least one sensor node and the network lifetime is maximum. The objective is to maximize U such that each target is monitored by at least one sensor node. 2) Sensor Deployment to Achieve k-Coverage: Given a set of n targets T = {T1, T2,..., Tn} located in u × v region and m sensor nodes S = {S1, S2,..., Sm}, place the nodes such that each target is monitored by at least k-sensor nodes and to maximize U. 3) Sensor Deployment to Achieve Q-Coverage: Given a set of n targets T = {T1, T2,..., Tn} located in u×v region and m sensor nodes S = {S1, S2,..., Sm}, place the nodes such that each target Tj , $1 \leq j \leq n$, is covered by at least q j sensor nodes and to maximize U

Sensor Deployment Since the upper bound of network lifetime can be computed, we have to find the deployment locations such that the network lifetime is maximum. First we use a heuristic to compute the deployment locations and then we use ABC and PSO algorithms to compute the locations. 1) A Heuristic for Sensor Deployment: Here , a heuristic for sensor deployment Initially, place the sensor nodes randomly. If any sensor node is idle (without monitoring any target), the node is moved to the least monitored targets' location. This is to ensure that all sensor nodes play their part in monitoring the targets. The sensor nodes are then sorted based on the number of targets it cover. The sensor node is placed at the middle of all the targets it cover. The next nearest target is identified and the sensor node is placed at the middle of all these targets. If it can cover this new target along with targets it was already monitoring, allow this move, else discard the move. This is done till the sensor node cannot cover any new target. At the end, upper bound is computed. The drawback of this approach is that it depends on the initial position of the sensor nodes. Though it may perform well for dense deployments, consistency cannot always be guaranteed

ABC Based Sensor Deployment:

Artificial Bee Colony (ABC) Algorithm is an optimization algorithm based on the intelligent behavior of honey bee swarm. The colony of bees contains three groups: employed bees, onlookers and scouts.The employed bee takes a load of nectar from the source and returns to the hive and unloads the nectar to a food store. After unloading the food, the bee performs a special form of dance called waggle dance which contains information about the direction in which the food will be found, its distance from the hive and its quality rating. Since information about all the current rich sources is available to an onlooker on the dance floor, an onlooker bee probably could watch numerous dances and choose to employ itself at the most qualitative source. There is a greater probability of onlookers choosing more qualitative sources since more information is circulating about the more qualitative sources. Employed foragers share their information with a probability, which is proportional to the quality of the food source. Hence, the recruitment is proportional to quality of a food source Exploitation is carried out by employed bees and onlookers, while exploration is carried out by scouts.

Sensor Deployment 1) Random Deployment: In random deployment, there is more chance of targets being not detected or targets not being covered with the required level of coverage. However, this may not hold true with dense deployment of nodes. But there is another possibility of some targets being monitored by many sensor nodes, and some by a few sensor nodes. This difference in the number of sensor nodes monitoring

each target will affect the network lifetime. The sensor nodes may be positioned in a better way so as to avoid this variation. This will yield better lifetime. Though random deployment has these drawbacks, there are applications where random deployment is the only feasible strategy.

Heuristic: The heuristic could consistently achieve better results compared to random deployment when the network size is increased

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCSA5202-WIRELESS SENSOR NETWORKS

Unit- I- WIRELESS SENSOR NETWORKS-SCSA5202

UNIT 4

ROUTING AND DISTRIBUTED COMPUTATION

Routing: Agent-Based Routing -Random Walk-Trace Routing Data Centric-Hierarchical - Location-Based – Energy Efficient-Routing Querying-Data Collection and Processing-Collaborative Information Processing And Group Connectivity.

ROUTING:

Routing in WSNs can be divided into flat-based routing, hierarchical-based routing, and location-based routing depending on the network structure. In flat-based routing, all nodes are typically assigned equal roles or functionality. In hierarchical-based routing, however, nodes will play different roles in the network. In location-based routing, sensor nodes' positions are exploited to route data in the network. A routing protocol is considered adaptive if certain system parameters can be controlled in order to adapt to the current network conditions and available energy levels. Furthermore, these protocols can be classified into multipath-based, query-based, negotiation-based, QoS-based, or coherent-based routing techniques depending on the protocol operation.

In addition to the above, routing protocols can be classified into three categories, namely, proactive, reactive, and hybrid protocols depending on how the source finds a route to the destination. In proactive protocols, all routes are computed before they are really needed, while in reactive protocols, routes are computed on demand. Hybrid protocols use a combination of these two ideas. When sensor nodes are static, it is preferable to have table driven routing protocols rather than using reactive protocols. A significant amount of energy is used in route discovery and setup of reactive protocols. Another class of routing protocol is called the cooperative routing protocols.

AGENT BASED ROUTING:

- Agent-based routing approach is One of the main objectives of WSNs is to report back the events of user's interest. The user interests are injected into a network by the Sink. Sink is a special node that acts like a server. The node that can identify the user requested interest is called source node. The source nodes report back the events to the sink. The WSN consists of uncountable nodes deployed with limited amount of banked energy, replenishment of which is a tedious task. This banked energy marks the life time on these nodes.
- Utilizing the energy of the nodes equitably and intelligently increases the life time of WSN into many folds. Hence, a scalable and intelligent routing approach is required. Towards this end, we propose AbR system that is scalable and intelligent enough to avoid continuously using and burning nodes energy along the shortest path to source node. Therefore, nodes along the shortest path will be provided a fair chance to rest by distributing their duty cycles to neighboring nodes.

- This may lead to exploration of energy expensive path toward source node, however in the long run network connectivity is maintained for longer period of time and abrupt node depletion is not witnessed. The sudden breakdown of aggressively used nodes in the optimal path creates connectivity holes in the network. This leads to the early segmentation of network with price payoff as inefficient and costly routing at later stage.
- To cut down on such sumptuous price payoff, developed two types of agent: stationary agent (SA) and mobile agent (MA).

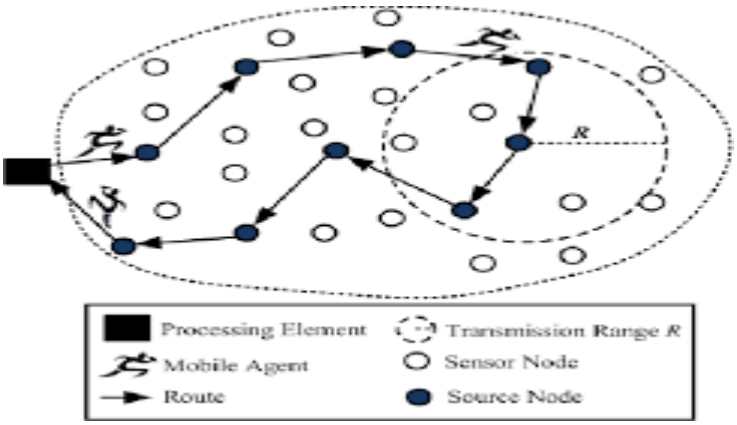


Figure 1: Agent Based Routing

- Every node on the sensor network is equipped with stationary agent. The role of SA is to acquire knowledge about its environment. The mobile agent is created and injected into the network by sink (SI). The MA benefits from the knowledge acquired by SA to select its next hop towards source node (SO), to distribute interest or processing code or report data to SI.

RANDOM WALK ROUTING:

The objective of random walks-based routing technique is to achieve load balancing in a statistical sense and by making use of multi-path routing in WSNs. This technique considers only large-scale networks where nodes have very limited mobility. In this protocol, it is assumed that sensor nodes can be turned on or off at random times. Further, each node has a unique identifier but no location information is needed. Nodes were arranged such that each node falls exactly on one crossing point of a regular grid on a plane, but the topology can be irregular. To find a route from a source to its destination, the location information or lattice coordination is obtained by computing distances between nodes using the distributed asynchronous version of the well-known Bellman-Ford algorithm.

Random walk- based routing is a probabilistic protocol in which each node selects randomly from its neighbor's nodes to forward the data packet. The path thus formed is a random walk (RW). RW based routing protocol is often proposed for very small devices, in large and dynamic networks due to being extremely simple to implement, requiring small memory footprints, and not requiring topology information of the network and load balancing property of the RW. On the other hand,

reactive (on-demand) routing protocols are also considered to be useful in resource constrained and dynamic WSNs.

However due to their inherent properties, increasing density of nodes badly affects performance of such protocols in terms of scalability. Furthermore, high mobility of sensor nodes and enabling low duty cycling make routing quite challenging. In such scenarios, random walk-based routing has not been studied widely. In this paper, we have put before a Lightweight Random Walk based Routing (LRWR) protocol in which each step follows a three messages exchange not only to discover neighbors but also to randomly select and forward the packets to the selected neighbor. We call this protocol lightweight since the number of messages required to achieve one step of RW are bare minimum. We applied the LRWR protocol in WSN with IEEE 802.15.4 standard and duty cycle enabled environments. By comparing its performance for low data rate with DYMO, a widely used protocol for WSN, we find that LRWR protocol offers a better alternative for duty cycle enabled mobile WSNs.

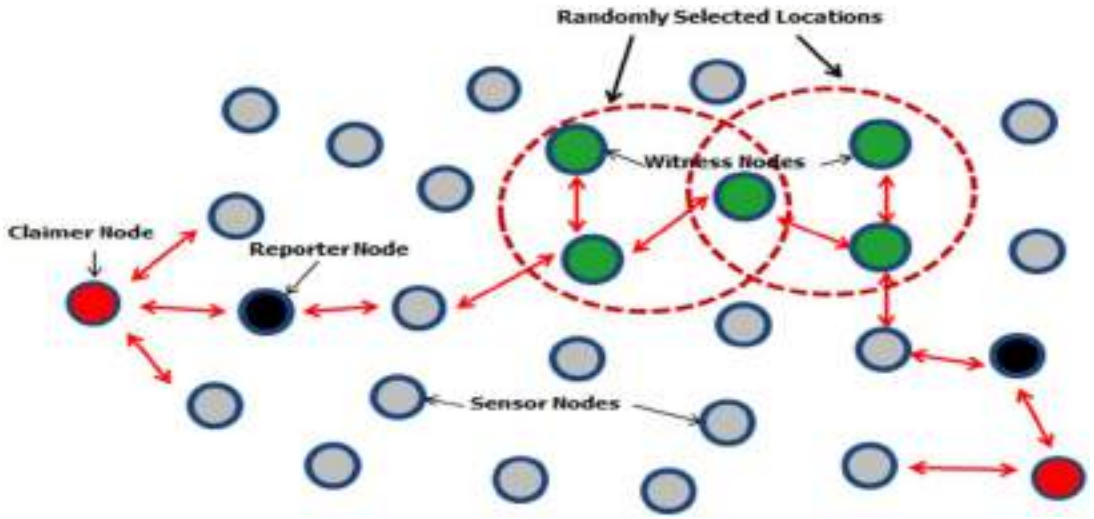


Figure 2: Random Walk

An intermediate node would select as the next hop the neighboring node that is closer to the destination according to a computed probability. By carefully manipulating this probability, some kind of load balancing can be obtained in the network. The routing algorithm is simple as nodes are required to maintain little state information. Moreover, different routes are chosen at different times even for the same pair of source and destination nodes. However, the main concern about this protocol is that the topology of the network may not be practical.

HIERARCHICAL ROUTING:

Hierarchical or cluster-based routing, originally proposed in wireline networks, are well-known techniques with special advantages related to scalability and efficient communication. As such, the concept of hierarchical routing is also utilized to perform energy-efficient routing in WSNs. In

a hierarchical architecture, higher energy nodes can be used to process and send the information while low energy nodes can be used to perform the sensing in the proximity of the target.

This means that creation of clusters and assigning special tasks to cluster heads can greatly contribute to overall system scalability, lifetime, and energy efficiency. Hierarchical routing is an efficient way to lower energy consumption within a cluster and by performing data aggregation and fusion in order to decrease the number of transmitted messages to the BS.

Hierarchical routing is mainly two-layer routing where one layer is used to select cluster heads and the other layer is used for routing. However, most techniques in this category are not about routing, rather on" who and when to send or process/aggregate" the information, channel allocation etc., which can be orthogonal to the multihop routing function.

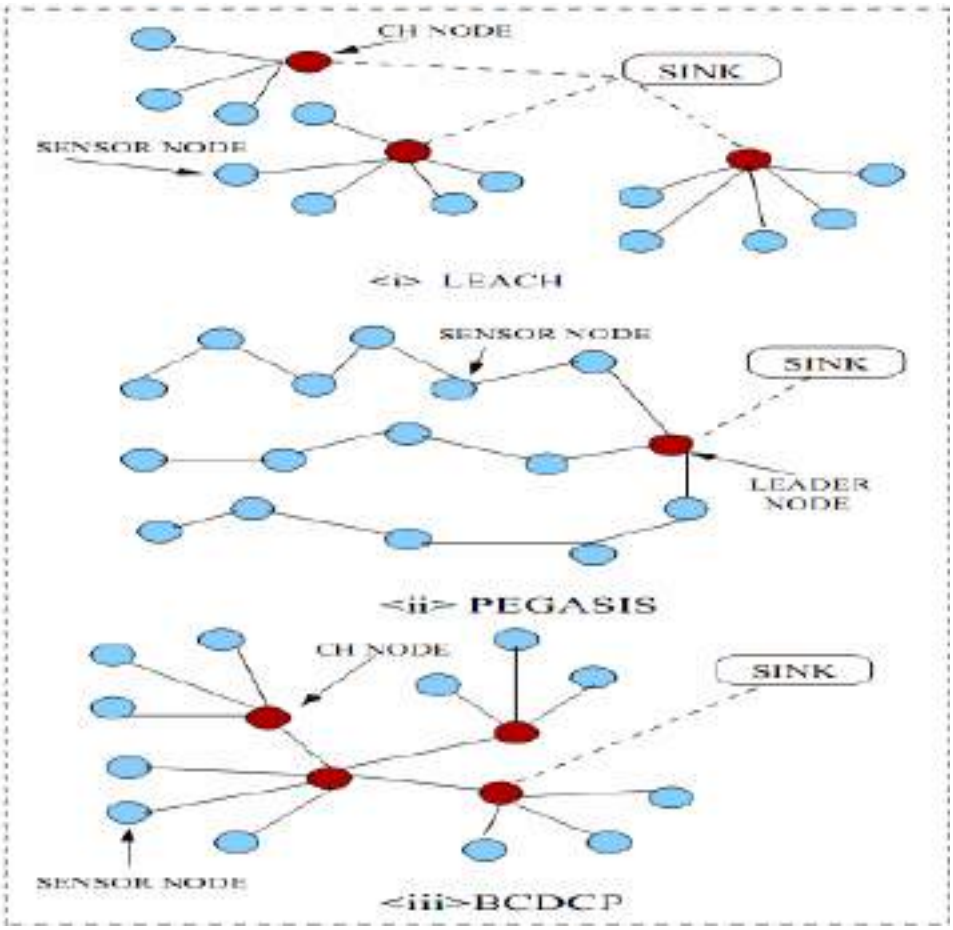


Figure 3: Hierarchical Routing

• **LEACH protocol**: Low Energy Adaptive Clustering Hierarchy (LEACH). LEACH is a cluster-based protocol, which includes distributed cluster formation. LEACH randomly selects a few sensor nodes as cluster heads (CHs) and rotate this role to evenly distribute the energy load among the sensors in the network. In LEACH, the cluster head (CH) nodes compress data arriving from nodes that belong to the respective cluster, and send an aggregated packet to the base station

• **Geographic Adaptive Fidelity (GAF)**: it is an energy-aware location-based routing algorithm designed primarily for mobile ad hoc networks, but may be applicable to sensor networks as well. The network area is first divided into fixed zones and form a virtual grid. Inside each zone, nodes collaborate with each other to play different roles. For example, nodes will elect one sensor node to stay awake for a certain period of time and then they go to sleep. This node is responsible for monitoring and reporting data to the BS on behalf of the nodes in the zone. Hence, GAF conserves energy by turning off unnecessary nodes in the network without affecting the level of routing fidelity.

Each node uses its GPS-indicated location to associate itself with a point in the virtual grid. Nodes associated with the same point on the grid are considered equivalent in terms of the cost of packet routing. Such equivalence is exploited in keeping some nodes located in a particular grid area in sleeping state in order to save energy.

ENERGY EFFICIENT ROUTING

In WSN Energy efficiency of a network is a significant concern in wireless sensor network (WSN). These days networks are becoming large, so information gathered is becoming even larger, which all consume a great amount of energy resulting in an early death of a node. Therefore, many energy efficient protocols are developed to lessen the power used in data sampling and collection to extend the lifetime of a network.

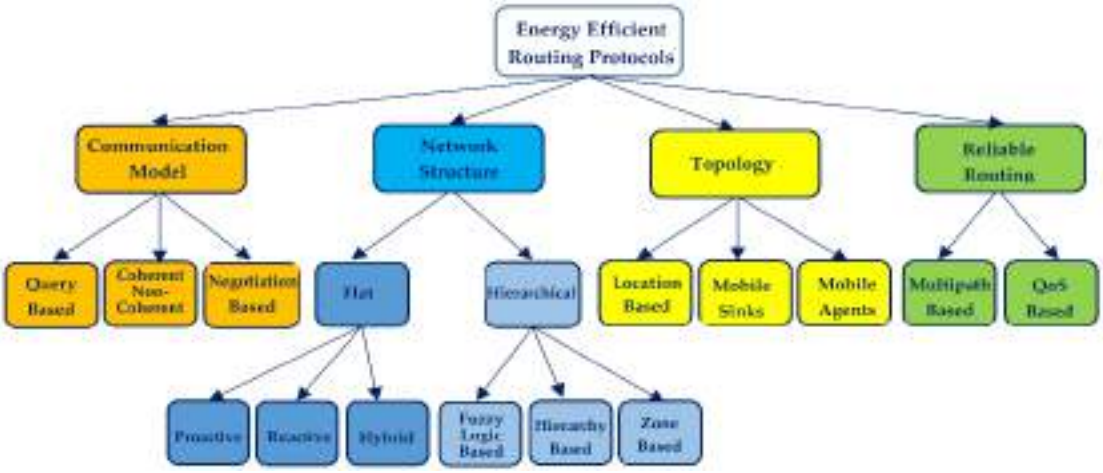


Figure 5: Taxonomy of Energy Efficient Routing

Energy efficient routing protocols:

Communication Model

Protocols of this category communication takes place from neighbor to neighbor, usually via single-hop routing. These data-centric protocols can convey more data for a certain quantity of energy. However, data delivery is not guaranteed. Protocols of this kind are classified into three subcategories depending on the method used in order to exchange data, which namely are: Query based, Coherent/Non-coherent, and Negotiation based.

1.Query Based:

Protocols of this subcategory use queries in order to route data. Whenever a node needs new data, it propagates a message (query) to ask for these data from the node that has them. Next, the node which owns the data requested sends them to the node that has applied the query. In what follows in this section, six typical examples of query based energy efficient routing protocols are described.

Directed Diffusion (DD) is a protocol that uses a naming scheme for data packets. It saves energy by diffusing data through the nodes and preventing unnecessary operations to run.DD uses a list of attribute-value pairs, with which it defines interests as object name, transmission, or geographic location. Interests are broadcasted from the BS to its neighbors and can be cached for later use. Interest caching includes gradients. Gradient is a reply link, from the neighbor sent the interest, which is described by data flow, duration, and expiring time generated from received interests. The nodes can do in-network data aggregation that is modeled as a minimum Steiner tree. Combining interests and gradients, multiple paths are generated between the BS and nodes, with one path been chosen using reinforcement.

To achieve this, the BS resends the initial interest from the selected path, in smaller intervals, resulting in reinforcement of the source node to send data more frequently. When a route failure occurs, DD tries to create a new or an alternative path by reinitiating reinforcement to search for new paths with lower casting ratios. The main advantages of DD are that node addressing mechanisms are not needed and there is no need for global knowledge of network topology. In addition, high energy efficiency is achieved.

COUGAR is a protocol that perceives the network as a distributed database system. It uses declarative queries to replace the network layer functions of query processing, as the selection of relevant nodes and utilizes in network data aggregation to save energy. To replace the network layer functions, it imports an additional layer, called query layer, between network and application layer. In COUGAR's architecture, nodes select a leader node for data aggregation and transmission to the BS. The main advantage of COUGAR is that it provides energy efficiency even with huge number of active nodes.

Active Query Forwarding In Sensor Networks (Acquire) uses a data centric mechanism for query sending and perceives the network as a distributed database, as COUGAR, which can divide complex queries in many sub-queries. The BS transmits a query, which is forwarded from every node that receives it. Upon query forwarding, nodes use their pre-cached information to reply to the query partially, updating pre-cached information from neighbor nodes, when needed, within a d hops distance. After the query is resolved, it can be sent back to the BS either from the reverse path or the shortest path. ACQUIRE provides efficient queries with proper setting of look-ahead parameter d. The traffic behaves like flooding when look-ahead parameter d is equal to network size but when the parameter is significant small queries have to travel more. ACQUIRE provides efficient querying when responses are collected from many nodes. However, if the look-ahead parameter is too small, query travels more hops.

Energy aware routing is a data centric routing protocol that constantly uses non optimal paths to maximize network lifetime. To pick one of these paths, it uses a probability function that depends on energy consumption of each path. This approach takes into consideration network lifetime as the only metric attribute. Instead of using the minimum energy path, it uses multiple routes with a certain probability to maximize network lifetime.

The operation of the routing protocol has three phases:

(i) Setup phase: Localized flooding is performed to find all paths from source to destination, calculate corresponding energy costs and create routing tables.

(ii) Data Communication phase: Based on the energy costs calculated, routing paths are chosen probabilistically and data are sent from source to destination.

(iii) Routing maintenance phase: With the intermittent use of localized flooding, routing paths are kept alive.

Gradient Based Routing (GBR) is a variant of Directed Diffusion. It combines the number of hops with interests and creates link heights and gradients to improve data communication. When an interest is diffused through the network, the number of hops is stored. Every node can find out the minimum number of hops to the BS, called node's height. A packet is transmitted through a link with a high gradient. The algorithm uniformly balances traffic over the network, which helps to balance nodes' load and prolong network lifetime, using techniques as data aggregation and traffic spread, as nodes act as relays for multiple paths. It uses three data spreading techniques:

(i) Stochastic design: the sender node picks one link in random in case there are two or more hops with the same gradient.

(ii) Energy based design: when a node has energy below a specified threshold, it increases its height to discourage other neighbors to transmit data.

(iii) Flow based design: flows from nodes that are part of other flows are prevented.

(iv)

Compared to DD, GBR has lower communication energy consumption

2.Coherent/Non-Coherent:

In this subcategory, nodes process collected data in the node level before they route them. In Coherent protocols, a node applies minimum processing only on the data it captures. On the other hand, in non-coherent routing protocols, nodes preprocess data they capture and send them to nodes, called aggregators, which further process them. In what follows in this section, two typical examples of this subcategory are described.

Single Winner Algorithm (SWE) an aggregator node, called Central Node (CN), to perform complex computations, depending on energy reservoirs and computational power. There are various message broadcasts before a CN can be elected. The first message is an announcement of nomination of each node and, when another node receives the message, it compares those candidates with itself. This comparison creates a second message broadcast, with the result of the comparison being sent again for another comparison, until a CN is elected. During the message broadcasts, better candidates create a minimum hop spanning tree, routed at the winning candidates, covering eventually the entire network.

Multiple Winner Algorithm (MWE) is an extension of SWE to prevent extra energy and computational overhead when multiple sources send data to CN. In MWE, each node keeps records of best candidate nodes and a set of minimum energy paths to each source. Thus, both energy and overhead are saved. Then, SWE is used to elect the best candidate for CN to aggregate data. Thus, energy consumption is reduced and a set of minimum energy routes to each source is found. However, long delays are caused and the scalability achieved is limited.

3.*Negotiation Based*

In this subcategory routing protocols, a source node exchanges data with their destination after negotiating. These protocols name data based on a naming scheme and use these names to advertise, negotiate and eventually reduce redundant data at destination. In what follows in this

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited "A" Grade by NAAC | 12B Status by UGC | Approved by AICTE
www.sathyabama.ac.in

SCHOOL OF COMPUTING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCSA5202-WIRELESS SENSOR NETWORKS

Unit- I- WIRELESS SENSOR NETWORKS-SCSA5202

Unit V

SENSOR NETWORK TOOLS

Sensor Network Platforms and Tools: Sensor node hardware- Programming challenges-Node level software platform-Node level simulators-Programming beyond individual nodes-Security-Privacy issues-Attacks and counter measures.

Sensor Node Hardware and programing

Sensor Node Hardware:

Sensor node hardware can be grouped into three categories. Augmented general-purpose computers, Dedicated embedded sensor nodes,System-on-chip (SoC).Berkley motes due to their small form factor, open source software development, and commercial availability, have gained wide popularity in the sensor network research community. 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 barely any operating system support. Typically support at least one programming language, such as C.zEx: mica, TinyOS.

Node-level software platforms

Node-centric design methodologies: Programmers think in terms of how a node should behave in the environment.zA node-level platform can be a node-centric OS, which provides hardware and networking abstractions of a sensor node to programmers.

TINY OS

Static memory allocation: analyzable, reduce memory management overheadzOnly parts of OS are compiled with the application. A program executed in TinyOShas two contexts, tasks and events.zTasks are posted by components to a task scheduler. Without preempting or being preempted by other taskszTriggered events can be preempted by other events and preempt tasks

nesC

An event call is a method call from a lower layer component to a higher layer component. (signal)zA command is the opposite. (call)zA component may use or provide the same interface multiple times. Give each interface inesCzAn application must contain the Main module which links the code to the scheduler at run time.zThe Main has a single StdControlinterface, which is the ultimate source of initialization of all componenstance a separate name using as notation.

Node-Level Simulators

- Wireless network are vulnerable to security attacks due to the broadcast nature of the transmission medium.

- Furthermore, WSN's have an additional vulnerability because nodes are often placed in a hostile or dangerous environment where the are not physically protected.
- For a large-scale sensor from physical or logical attack. Attackers may device different types of security threats to make the WSN system unstable.

Programming in wireless sensor network

```
interface AMSend {
  command error_t send(am_addr_t addr, message_t* msg, uint8_t len);
  command error_t cancel(message_t* msg);
  event void sendDone(message_t* msg, error_t error);
  command uint8_t maxPayloadLength();
  command void* getPayload(message_t* msg, uint8_t len);
}
```

```
module Sampler {
  uses interface Boot;
  uses interface TemperatureSensor;
  uses interface AMSend;
}

implementation {
  bool transmitLock;
  message_t msgBuffer;

  event void Boot.booted {
    call TemperatureSensor.read();
  }

  event void TemperatureSensor.readDone(uint16_t v){
    uint16_t* msg_payload = (uint16_t*) call AMSend.getPayload(&msgBuffer);
    *msg_payload = v;
    if (!transmitLock) {
      transmitLock = TRUE;
      if (!call AMSend.send(TOS_BCAST_ADDR, &msgBuffer, sizeof(message_t))) {
        transmitLock = FALSE;
      }
    }
  }

  event void AMSend.sendDone(message_t* msg, result_t success) {
    if(transmitLock && msg == &msgBuffer ) {
      transmitLock = FALSE;
    } else {
      // Error...
    }
  }
}
```

Figure 1: Code

WIRELESS SENSOR NETWORK APPLICATIONS

WSNs are being employed in a variety of scenarios. Such diversity translates into dif_x0002_ferent requirements and, in turn, different programming constructs supporting them.• Here we identify some common

CD simulators do not allow interdependencies within a single tick. Synchronous languages [91], which are typically used in control system designs rather than sensor network designs, do allow cyclic dependencies. They use a fixed-point semantics to define the behavior of a system at each tick.

Unlike cycle-driven simulators, a discrete-event (DE) simulator assumes that the time is continuous and an event may occur at any time. An event is a 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 a software module within a node and the events can be message passings among these modules.

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 event should not be earlier than that of the input event. Noncausal 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 the earliest time stamp) from the queue and triggers the component that reacts to that event.

In terms of timing behavior, 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. At an early stage of a design when only the asymptotic behaviors rather than timing properties are of concern, CD simulations usually require less complex components and give faster simulations. Partly because of the approximate timing behaviors, which make simulation results less comparable from application to application, there is no general CD simulator that fits all sensor network simulation tasks. We have come across a number of home grown simulators written in Matlab, Java, and C++. Many of them are developed for particular applications and exploit application-specific assumptions to gain efficiency.

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.8 The focus of these simulators is on network modeling, protocols 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. Examples include TOSSIM for Berkeley motes and Em* (pronounced em star)for Linux-based nodes such as Sensoria WINS NG platforms.

Node-Level Simulator: ns-2 &TOSSIM:

ns-2

-Originally developed for wired networks

-Extensions for sensor nodes

-Node locations vs. logical addresses

BASED ON CAPABILITY OF ATTACKER

- Outsider versus insider (Node Compromise) attack.
- Passive versus active attacks.
- Mote-class versus laptop-class attacks.

Attacks on Information in Transit

- Interruption
- Interception
- Modification
- Fabrication
- Replaying existing messages.

ISSUES WITH HIGH-LEVEL SECURITY MECHANISMS

CRYPTOGRAPH

To achieve security in WSNs, it is important to be able to perform various cryptographicoperations, including encryption, authentication, and so on. However, decision for Selecting the appropriate cryptography method depends on the computation and communication capability of the sensor nodes. Asymmetric cryptography is often too expensive for many applications. Thus, a promising approach is to use more efficient symmetric cryptographic alternatives. However, symmetric cryptography is not as versatile as public key cryptographic techniques, which complicates the design of secure applications. Applying any encryption scheme requires transmission of extra bits, hence extra processing, memory and battery power, which are very important resources for the sensors' longevity. Applying the security mechanisms such as encryption could also increase delay, jitter and packet loss in WSNs.

The process by which public key and symmetric key cryptography schemes should be selected is based on the following criteria:

- Energy

operations of the protocols and result in unfair bandwidth usage. In either way, the network performance is degraded. Eventually, the collisions and unfairness lead traffic distortion.

Identity Spoofing: MAC identity spoofing is another common attack in the MAC layer.Due to the broadcast nature of wireless communications, the MAC identity (such as a MAC address or a certificate) of a sensor is open to all the neighbors, including attackers. Without proper protection on it, an attacker can fake an identity and pretend to be a different one. A typical MAC identity spoofing attack is the Sybil attack.

CounterMeasures:

Misbehavior Detection Because attacks deviate from normal behaviors, it is possible to identify attackers by observing what has happened. Various data can be collected for this purpose, and various actions can be taken after detection. In a countering scheme for the IEEE 802.11 protocol, a receiver assigns and adjusts the backoff values to be used by the corresponding sender. Whenever detecting the sender's misbehavior in manipulating backoff value, the receiver may add some penalty to the next backoff value assigned to the sender. The idea was applied to ad hoc networks similarly can also be applied to WSNs. Another solution uses "watchdogs" on every node to monitor whether or not the neighbors of a node forward the packets sent out by this particular node. A neighbor not forwarding packets will be identified by the watchdog as a misbehaving node.