SRI CHANDRASEKHARENDRA SARASWATHI VISWA MAHAVIDYALAYA

(University established under section 3of UGC Act 1956) (Accredited with 'A' Grade by NAAC)

Enathur, Kanchipuram – 631 561

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

Course Material for

Wireless Sensor Networks

FULL TIME B.E., IV YEAR / VII SEMESTER

Prepared By:Dr.S.Omkumar,ECE

Approved by: Prof.V.Swaminathan

PEC3 WIRELESS SENSOR NETWORKS VII SEMESTER

PRE-REQUISITE:

Basic knowledge of Data Communication Networks

OBJECTIVES:

- To understand the basics of Wireless sensor Networks
- To learn the Architecture of WSN
- To understand the concept of Networking and Networking in WSN

UNIT I OVERVIEW OF WIRELESS SENSOR NETWORKS (9 Hrs)

Single-Node Architecture - Hardware Components- Network Characteristics- unique constraints and challenges, Enabling Technologies for Wireless Sensor Networks- Types of wireless sensor networks.

UNIT II ARCHITECTURES (9 Hrs)

Network Architecture- Sensor Networks-Scenarios- Design Principle, Physical Layer and Transceiver Design Considerations, Optimization Goals and Figures of Merit, Gateway Concepts, Operating Systems and Execution Environments- Introduction to TinyOS and nesC- Internet to WSN Communication

UNIT III NETWORKING SENSORS (10 Hrs)

MAC Protocols for Wireless Sensor Networks, Low Duty Cycle Protocols And Wakeup Concepts - SMAC, - B-MAC Protocol, IEEE 802.15.4 standard and ZigBee, the Mediation Device Protocol, Wakeup Radio Concepts, Address and Name Management, Assignment of MAC Addresses, Routing Protocols Energy-Efficient Routing, Geographic Routing.

UNIT IV INFRASTRUCTURE ESTABLISHMENT (8 Hrs)

Topology Control, Clustering, Time Synchronization, Localization and Positioning, Sensor Tasking and Control.

UNIT V SENSOR NETWORK PLATFORMS AND TOOLS (9 Hrs)

Sensor Node Hardware – Berkeley Motes, Programming Challenges, Node-level software platforms, Node level Simulators, State-centric programming.

OUTCOMES: Total: 45 Hrs

Upon completion of the course, students will be able to:

- Understand challenges and technologies for wireless networks
- Understand architecture and sensors
- Establishing infrastructure and simulations

TEXT BOOKS:

1. Holger Karl & Andreas Willig, "Protocols and Architectures for Wireless Sensor Networks", John Wiley, 2005.
2. Feng Zhao & Leonidas J.Guibas, "Wireless Sensor Networks-An Information Processing Approach", Elsevier, 2007
3. Waltenegus Dargie , Christian Poellabauer, "Fundamentals Of Wireless Sensor Networks - Theory And Practice", By John Wiley & Sons Publications, 2011

REFERENCES:

1. KazemSohraby, Daniel Minoli, & TaiebZnati, "Wireless Sensor Networks-Technology, Protocols, and Applications", John Wiley, 2007.
2. Anna Hac, "Wireless Sensor Network Designs", John Wiley, 2003

Wireless Sensor Networks

Unit I / Overview of WSN

Prepared
By

Dr.S.Omkumar/Associate Prof
Department of ECE, SCSVMV

Syllabus / Unit - I

- Overview of WSN:
- Single-Node Architecture - Hardware Components - Network Characteristics - Unique constraints and challenges - Enabling Technologies for Wireless Sensor Networks- Types of wireless sensor networks.

Topic 1

Introduction to WSN

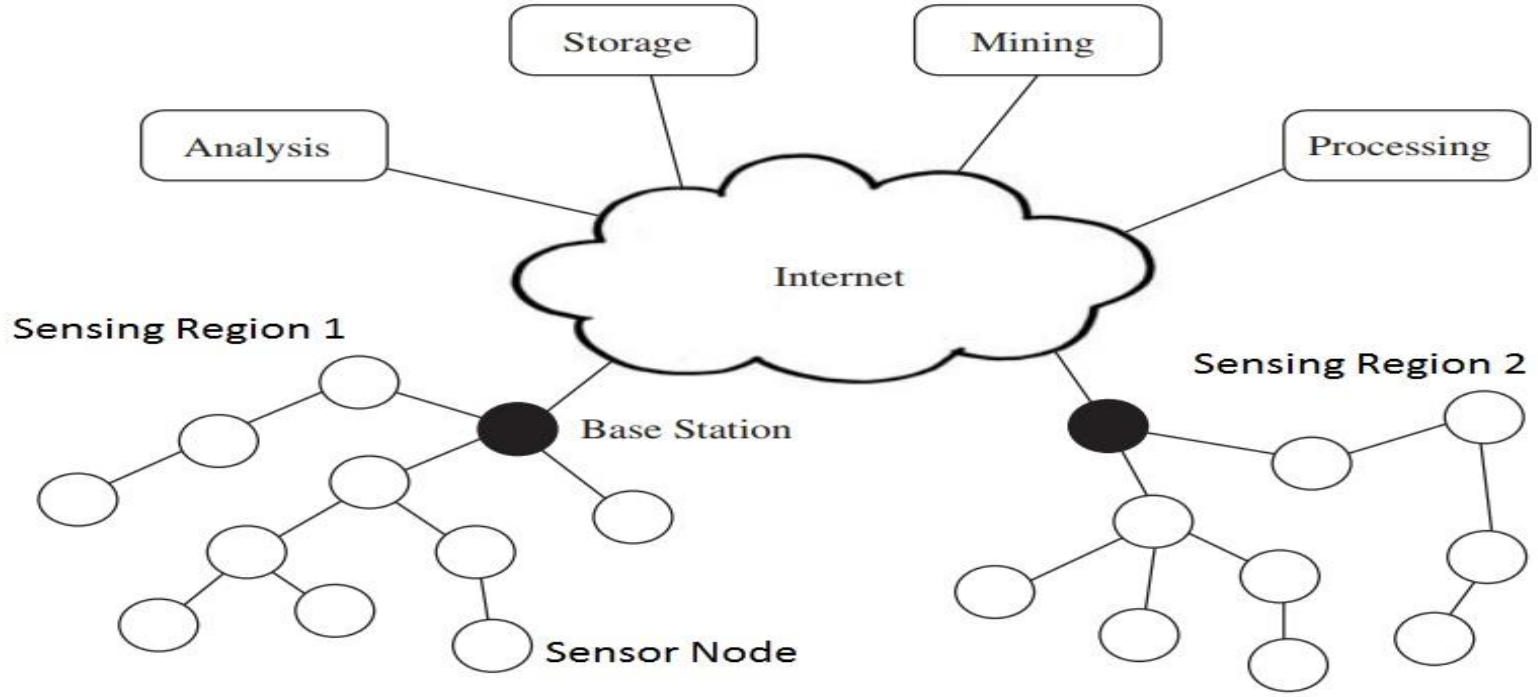
Introduction

- A Sensor is a device used to gather information about a physical process and translate into electrical signals that can be processed, measured and analyzed.
- The physical process can be any real-world information like temperature, pressure, light, sound, motion, position, flow, humidity, radiation etc.
- A Sensor Network is a structure consisting of sensors, computational units and communication elements for the purpose of recording, observing and reacting to an event or a phenomenon.
- The events can like physical world, an industrial environment, a biological system while the controlling or observing body can be a consumer application, government, civil, military, or an industrial entity.

- Such Sensor Networks can be used for remote sensing, medical telemetry, surveillance, monitoring, data collection etc.

Wireless Sensor Networks

- A typical sensor network consists of sensors, controller and a communication system. If the communication system in a Sensor Network is implemented using a Wireless protocol, then the networks are known as Wireless Sensor Networks.



- According to technologists, Wireless Sensor Networks is an important technology for the twenty first century.
- Recent developments in MEMS Sensors (Micro Electro Mechanical System) and Wireless Communication has enabled cheap, low power, tiny and smart sensors, deployed in a wide area and interconnected through wireless links for various civilian and military applications.
- A Wireless Sensor Network consists of Sensor Nodes deployed in large quantities and support sensing, data processing, embedded computing and connectivity.

Motivation for WSN

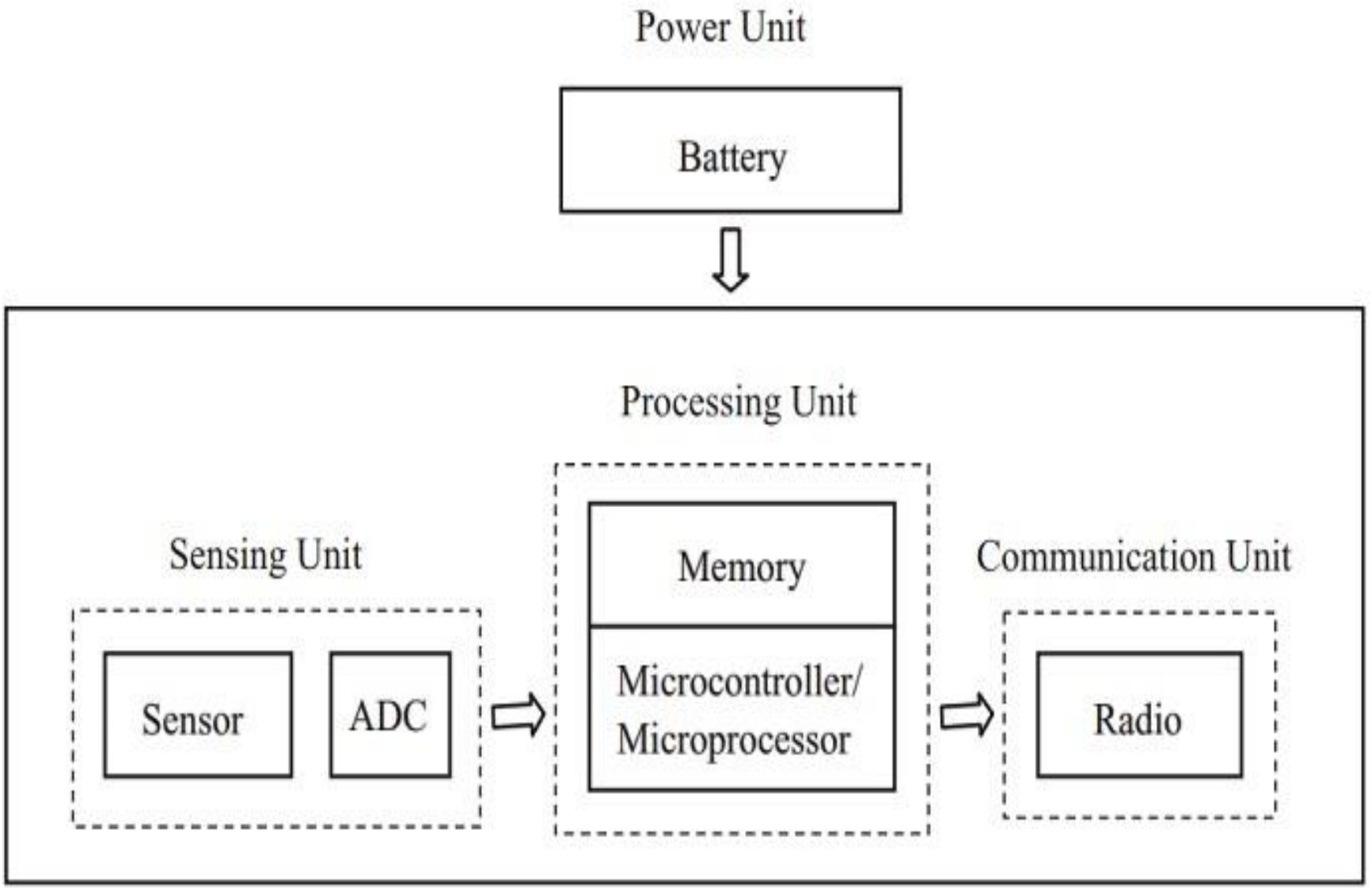
- The recent developments in engineering, communication and networking led to new sensor designs, information technologies and wireless systems.
- Such advanced sensors can be used as a bridge between the physical world and the digital world.
- Sensors are used in numerous devices, industries, machines and help in avoiding infrastructure failures, accidents, conserving natural resources, preserving wildlife, increase productivity, provide security etc.
- The use of distributed sensor network contributed by the technological advances in VLSI, MEMS and Wireless Communication.

- With the help of modern semiconductor technology, powerful microprocessors can be developed, smaller in size when compared to the previous generation products. This miniaturization of processing, computing and sensing technologies led to tiny, low-power and cheap sensors, controllers and actuators.

Elements of WSN

- A typical wireless sensor network can be divided into two elements. They are:
  - Sensor Node
  - Network Architecture
- A Sensor Node in a WSN consists of four basic components. They are:
  - Power Supply
  - Sensor
  - Processing Unit
  - Communication System

Fig 2 / Basic Components of WSN



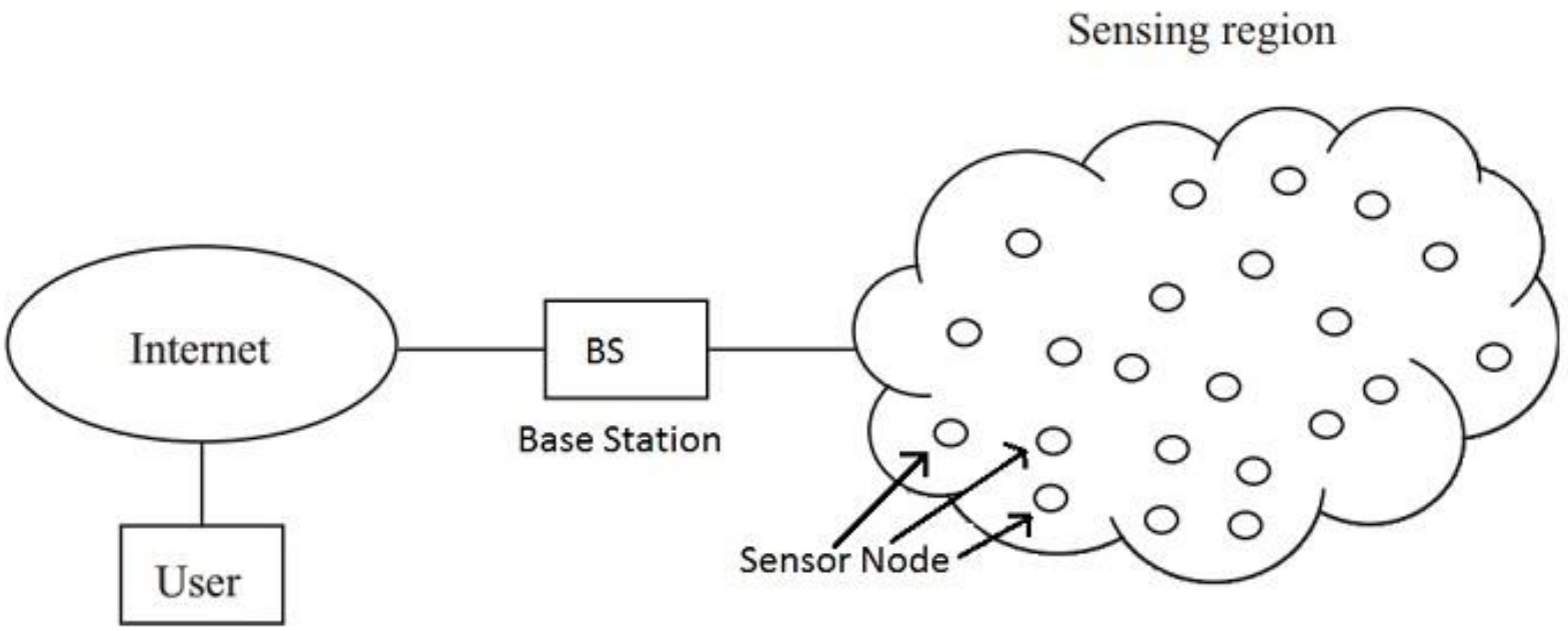
Elements of WSN (Cont)

- The sensor collects the analog data from the physical world and an ADC converts this data to digital data.
- The main processing unit a microprocessor or a microcontroller, performs an intelligent data processing and manipulation. Communication system consists of radio system, a short-range radio for data transmission and reception.
- As all the components are low-power devices, a small battery like CR-2032, is used to power the entire system.
- A Sensor Node consists of not only the sensing component but also other important features like processing, communication and storage units.

- With all these features, components and enhancements, a Sensor Node is responsible for physical world data collection, network analysis, data correlation and fusion of data from other sensor with its own data.

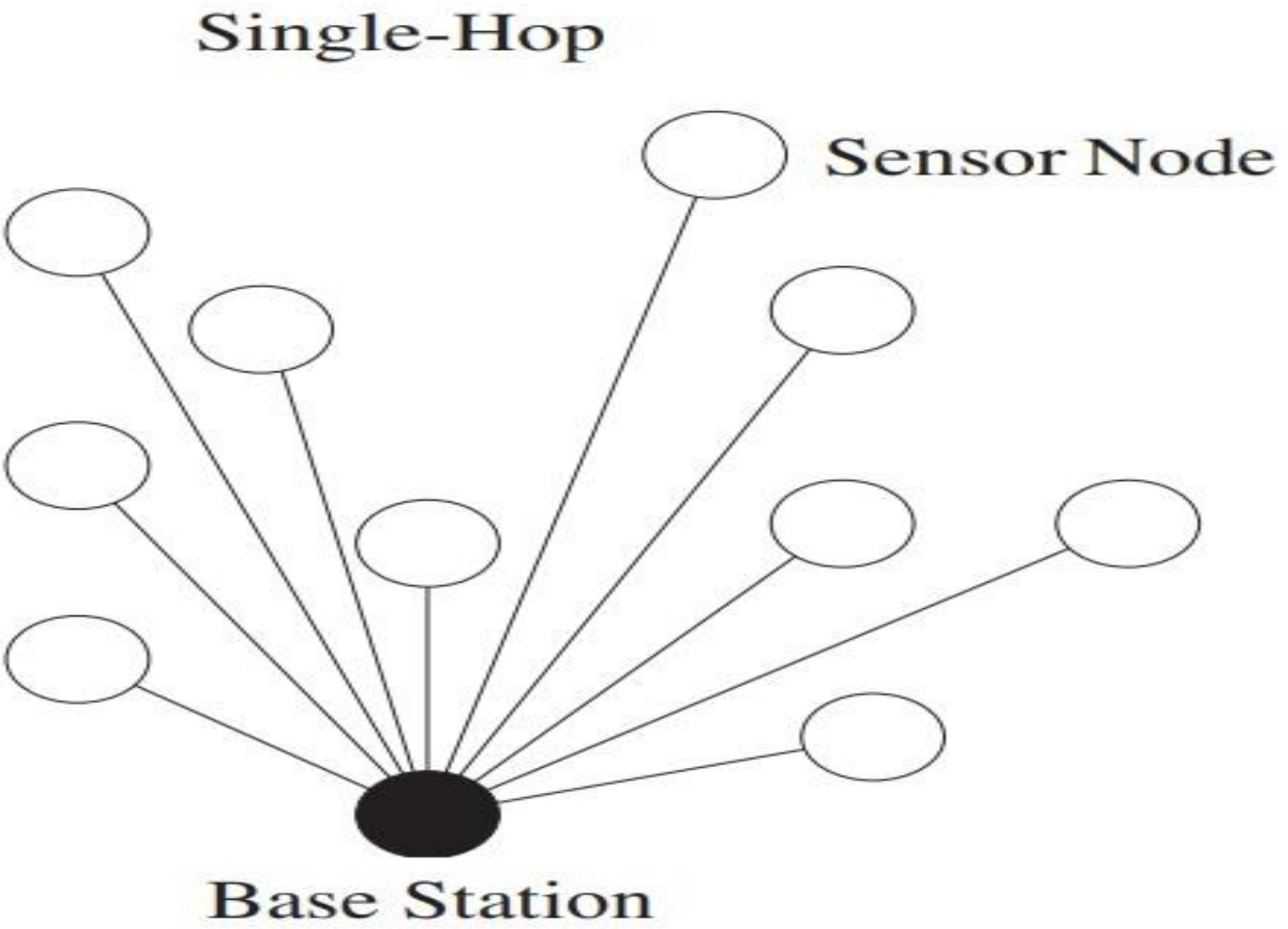
Network Architecture

- When a large number of sensor nodes are deployed in a large area to monitor a physical environment, the networking of these sensor nodes is equally important. A sensor node in a WSN not only communicates with other sensor nodes but also with a Base Station (BS) using wireless communication.



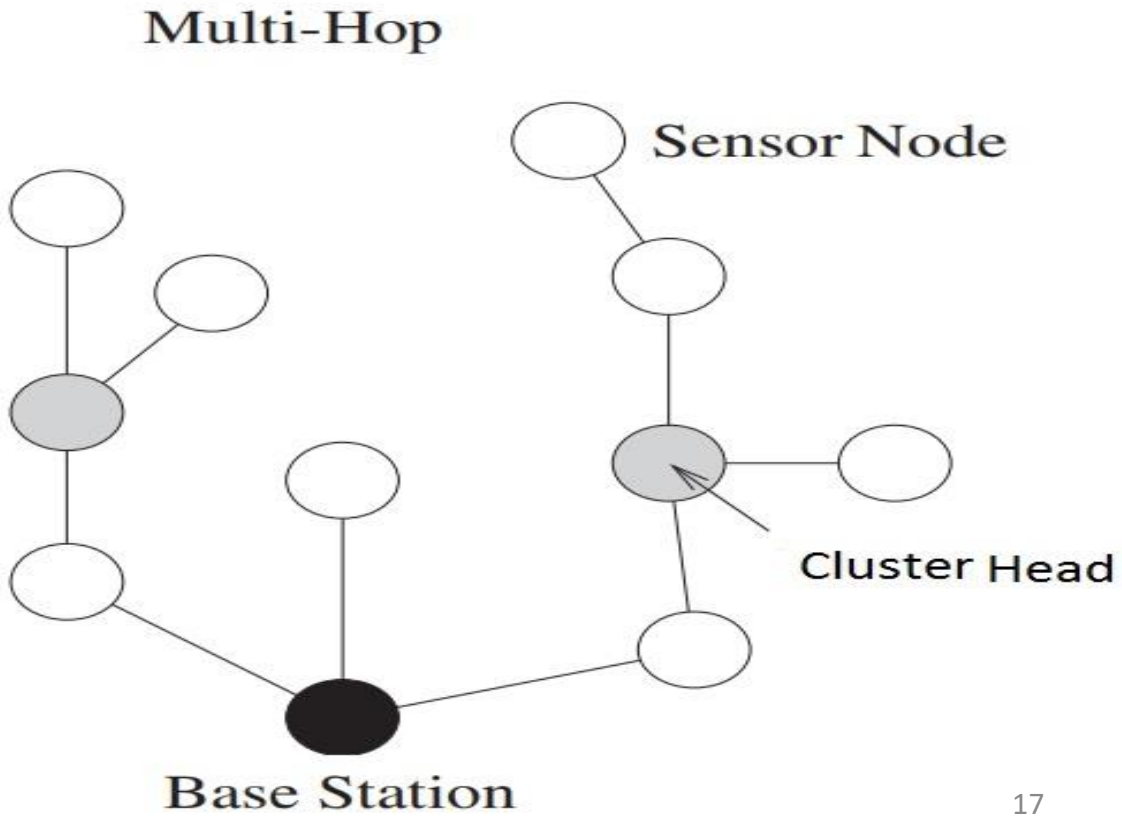
- The base station sends commands to the sensor nodes and the sensor node perform the task by collaborating with each other.
- The sensor nodes in turn send the data back to the base station. A base station also acts as a gateway to other networks through the internet.
- After receiving the data from the sensor nodes, a base station performs simple data processing and sends the updated information to the user using internet.
- If each sensor node is connected to the base station, it is known as Single-hop network architecture.
- Although long distance transmission is possible, the energy consumption for communication will be significantly higher than data collection and computation.

Fig 4 / Single Hop Architecture



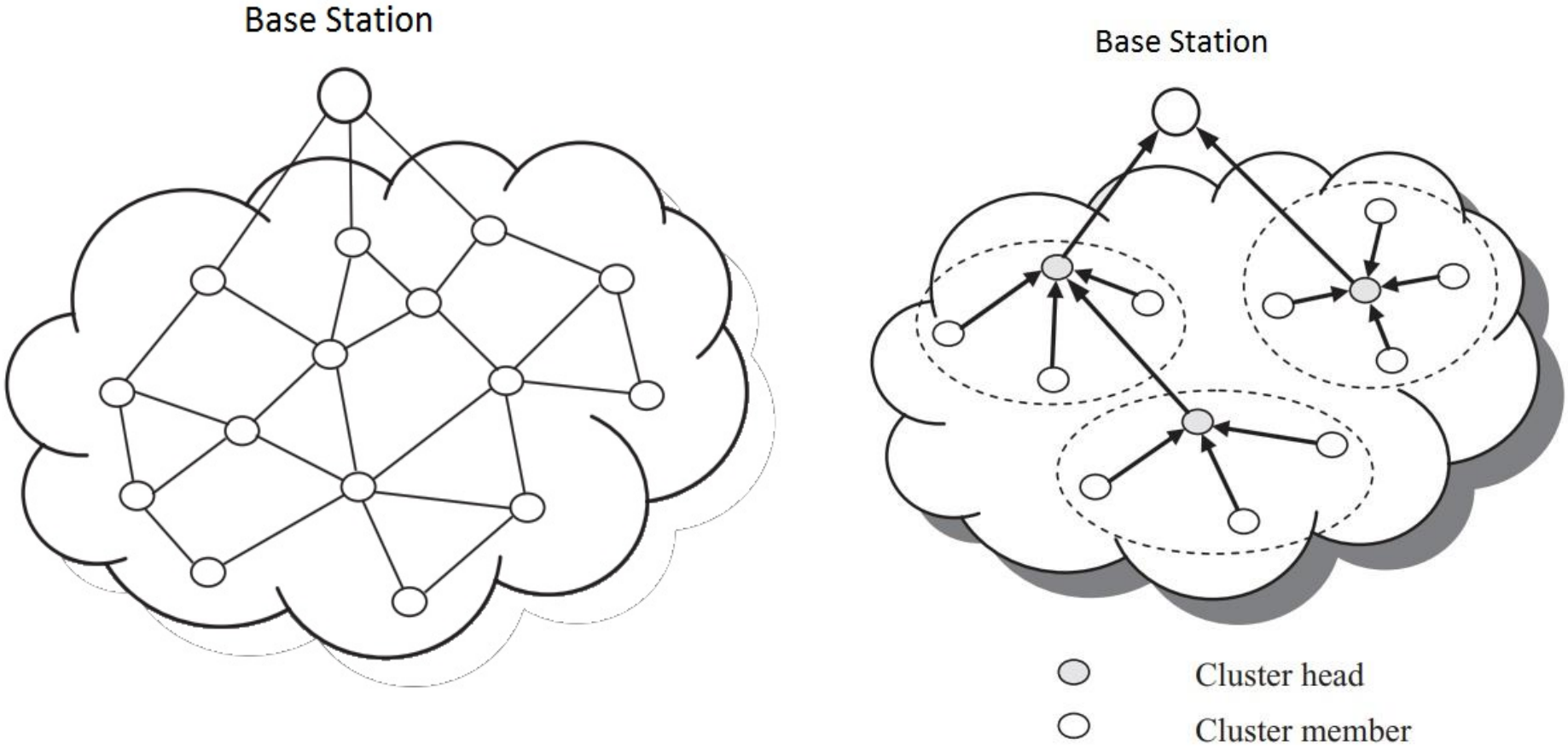
Multi-hop Architecture

- Hence, Multi-hop network architecture is usually used. Instead of one single link between the sensor node and the base station, the data is transmitted through one or more intermediate node.



- This can be implemented in two ways. Flat network architecture and Hierarchical network architecture.
- In flat architecture, the base station sends commands to all the sensor nodes but the sensor node with matching query will respond using its peer nodes via a multi-hop path.
- In hierarchical architecture, a group of sensor nodes are formed as a cluster and the sensor nodes transmit data to corresponding cluster heads.
- The cluster heads can then relay the data to the base station

Fig 6 / Flat and Hierarchical Network Architectures



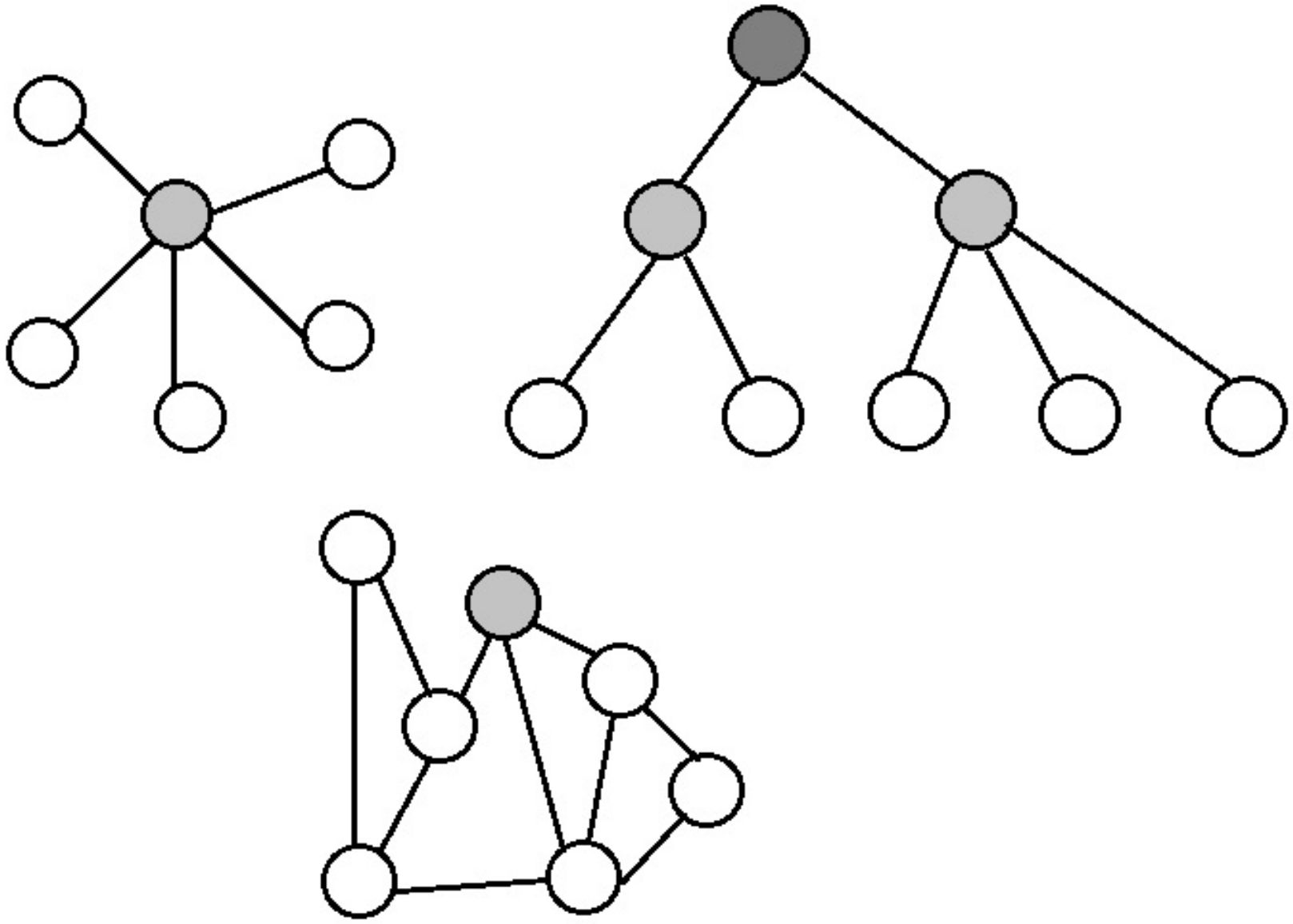
Network Topologies in WSN

- A WSN can be either a single-hop network or a multi-hop network. The following are a few different network topologies that are used in WSNs.
- Star Topology
- In star topology, there is a single central node known as hub or switch and every node in the network is connected to this hub. Star topology is very easy to implement, design and expand. The data flows through the hub and plays an important role in the network and a failure in the hub can result in failure of entire network.

- Tree Topology
- A tree topology is a hierarchical network where there is a single root node at the top and this node is connected to many nodes in the next level and continues. The processing power and energy consumption is highest at the root node and keeps on decreasing as we go down the hierarchical order.

- Mesh Topology
- In mesh topology, apart from transmitting its own data, each node also acts as a relay for transmitting data of other connected nodes. Mesh topologies are further divided into Fully Connected Mesh and Partially Connected Mesh. In fully connected mesh topology, each node is connected to every other node while in partially connected mesh topology, a node is connected one or more neighboring nodes.

Fig 7 / Network Topologies in WSN



Applications of WSN

- Air Traffic Control (ATC)
- Heating Ventilation and Air Conditioning (HVAC)
- Industrial Assembly Line
- Automotive Sensors
- Battlefield Management and Surveillance
- Biomedical Applications
- Bridge and Highway Monitoring
- Disaster Management
- Earthquake Detection
- Electricity Load Management

- Environment Control and Monitoring
- Industrial Automation
- Inventory Management
- Personal Health Care
- Security Systems

Topic 2

Single Node Architecture – Hardware Components

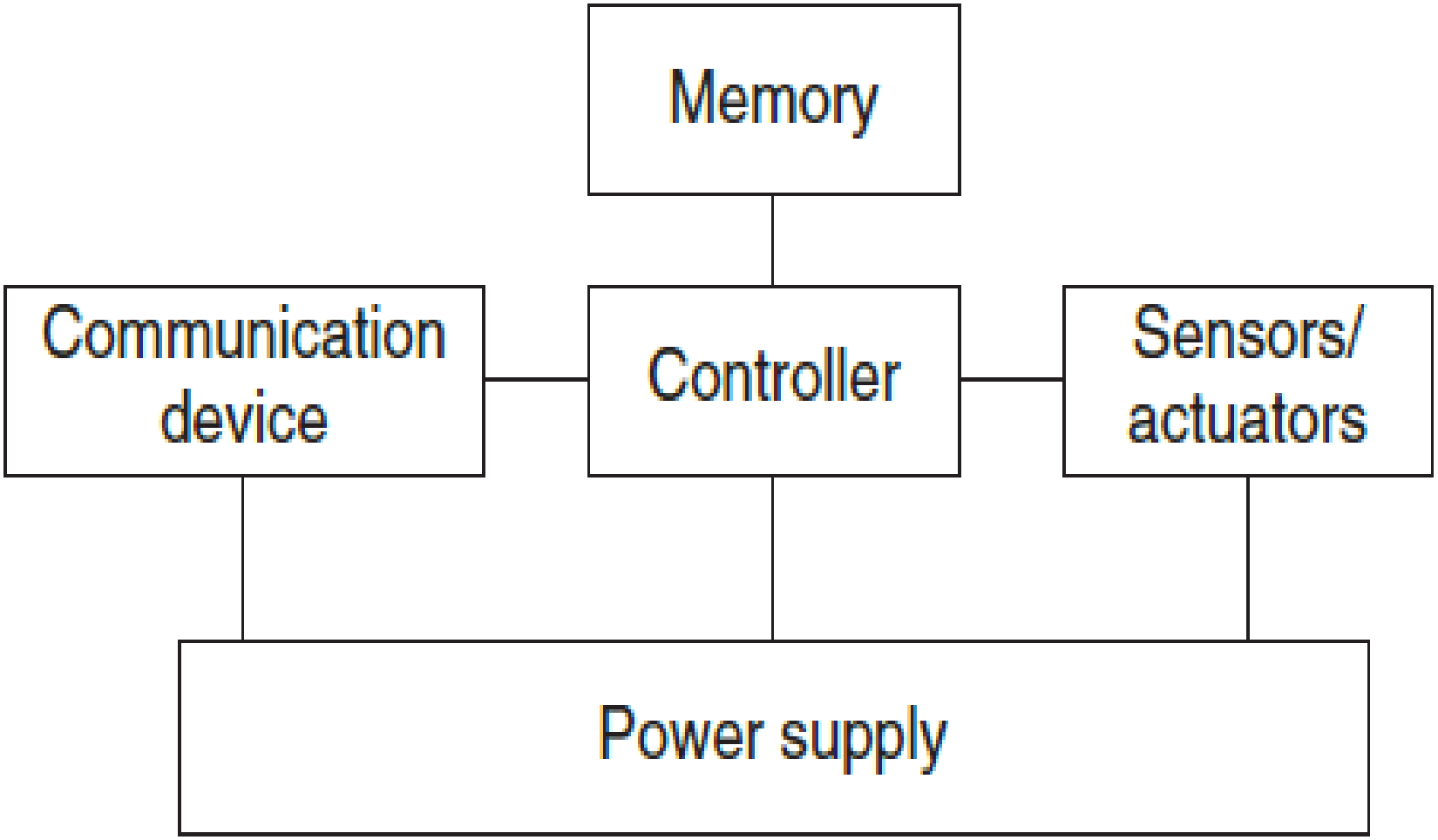
Introduction

- Building a wireless sensor network requires the constituting nodes to be developed. These nodes have to meet the requirements from a given application. They have to be small, cheap, energy efficient, equipped with the right sensors, memory resources and sufficient communication facilities. The hardware components of the functioning node are explained as follows.

Overview of Sensor Node

- A basic sensor node comprises five main components are shown in the Figure.
- **Controller:** To process all relevant data
- **Memory: T**o store programs and intermediate data.
- **Sensors and actuators:** Actual interface to the physical world to observe or control physical parameters of the environment.
- **Communication: D**evice for sending and receiving information over a wireless channel
- **Power supply: S**ome form of batteries necessary to provide energy and some form of recharging by obtaining energy from the environment as well.

Fig 8 / Basic Components of a Sensor Node



Controllers

- The controller is the core of a wireless sensor node.
- It is the Central Processing Unit (CPU) of the node
- It collects data from sensors, processes this data, 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.
- Such a variety of processing tasks can be performed on various controller architectures, representing trade-offs between flexibility, performance, energy efficiency, and costs.

- Microcontrollers are suitable for WSNs since they can reduce their power consumption by going into **sleep states** where only parts of the controller are active.
- One of the main differences to general-purpose systems is that microcontroller-based systems do not include a memory management unit – for example, protected or virtual memory is difficult.
- In a wireless sensor node, DSP can be used to process incoming data. But the advantages of a DSP are not required in a WSN node and they are usually not used.
- Another option for the controller is to use Field-Programmable Gate Arrays (FPGAs) or Application-Specific Integrated Circuits (ASICs) instead of microcontrollers.

- An FPGA can be reprogrammed in the field to adapt to a changing set of requirements , but this can take time and energy.
- An ASIC is a specialized processor, designed for a given application such as high-speed routers and switches.
- The typical trade-off here is loss of flexibility in return for a considerably better energy efficiency and performance.

Memory

- There is a need for Random Access Memory (RAM) to store intermediate sensor readings, packets from other nodes etc.
- RAM is fast, but it loses its contents if power supply is interrupted.
- The program code can be stored in Read-Only Memory (ROM) or in Electrically Erasable Programmable Read-Only Memory (EEPROM) or flash memory.
- Flash memory can also serve as intermediate storage of data when the power supply goes off for some time.
- The long read and write access delays of flash memory should be taken into account as well as the high required energy.

Communication Module

1. Choice of transmission medium

- The first choice is the transmission medium and usual choices include radio frequencies, optical communication, and ultrasound.
- Radio Frequency (RF)-based communication is vital requirement of most WSN applications.
- It provides long range and high data rates, acceptable error rates at reasonable energy expenditure, and does not require line of sight between sender and receiver.
- For a practical wireless, RF-based system, the carrier frequency has to be carefully chosen. The wireless sensor networks use communication frequencies between about 433 MHz and 2.4 GHz.

2. Transceivers

- For actual communication, both a transmitter and a receiver are required in a sensor node to convert a bit stream coming from a microcontroller and convert them to and from radio waves. Such combined devices are called **transceivers**.
- Usually, half-duplex operation is realized since transmitting and receiving at the same time on a wireless medium is impractical in most cases. A range of low-cost transceivers is available that incorporate all the circuitry required for transmitting and receiving, modulation, demodulation, amplifiers, filters, mixers etc..

3. Transceiver tasks and characteristics

- The following are the some of the important characteristics of a transceiver which should be taken into account.
  - Service to upper layer
  - Power Consumption and Energy Efficiency
  - Carrier Frequency & Multiple channels
  - Transmission Power Control
  - Data Rates
  - Modulation
  - Noise Figure
  - Power Efficiency
  - Frequency Stability etc

4. Transceiver States

- **Transmit State: T**he transmit part of the transceiver is active and the antenna radiates energy.
- **Receive State: T**he receive part is active.
- **Idle State:** A transceiver that is ready to receive but not currently receiving anything is said to be in an **idle state**.
- **Sleep State: The** significant parts of the transceiver are switched off. There are transceivers offering several different sleep states.

Sensors & Actuators

- Sensors can be categorized into the following three categories -

1. Passive Omni-directional sensors:

- They can measure a physical quantity at the point of the sensor node without manipulating the environment by active probing. They obtain the energy directly from the environment – energy is only needed to amplify their analog signal. There is no notion of "direction in these measurements. Typical examples include thermometer, light sensors, vibration, microphones, humidity, chemical sensors etc

2. Passive narrow-beam sensors: They are passive but have a well-defined notion of direction of measurement. A typical example is a camera, which can “take measurements” in a given direction, but has to be rotated if need be.

3. Active sensors: They probe the environment, for example, a sonar or radar sensor or some types of seismic sensors, which generate shock waves by small explosions.

Power Supply of Sensor Nodes

1. Traditional batteries

- The power source of a sensor node is a battery, either non-rechargeable (primary batteries) or, if an energy scavenging device is present on the node, also rechargeable (secondary batteries).
- In some form or other, batteries are electro-chemical stores for energy – the chemicals being the main determining factor of battery technology.

2. Energy scavenging

- Some of the unconventional energy sources like fuel cells, micro heat engines and radioactivity – convert energy from stored secondary form into electricity in a easy way than a normal battery would do.
- The entire energy supply is stored on the node itself – once the fuel supply is exhausted, the node fails.
- The energy from a node's environment must be tapped into and made available to the node – **energy scavenging** should take place.

3. Photo-voltaics

The 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. The resulting power ranges between 10 mW/cm^2 indoors and 15 mW/cm^2 outdoors. Single cells achieve a fairly stable output voltage of about 0.6 V. Hence, solar cells are used to recharge secondary batteries.

4. Temperature gradients

Differences in temperature can be directly converted to electrical energy. Theoretically, even small difference for example, 5 K can produce considerable power, but practical devices fall very short of theoretical upper limits.

5. Vibrations

Walls or windows in buildings are resonating with cars or trucks passing in the streets, machinery often has low- frequency vibrations, ventilations also cause it, and so on. The available energy depends on amplitude and frequency of the vibration and ranges between 0.1 mW/cm^3 and 10, 000 mW/cm^3 for some extreme cases.

Topic 3

Network Characteristics

- The following are the characteristics of Wireless Sensor Networks:

1. Type of service

- The service type provided by a conventional communication network is to move bits from one place to another.. A WSN is expected to provide meaningful information and actions about a given task. The concepts like *scoping* of interactions to specific geographic regions or to time intervals are important. Hence using such a network along with new interfaces and new ways of thinking about the service of a network are required.

2. Quality of Service

- The quality of service **is c**losely related to the type of a network's service. The traditional quality of service requirements coming from multimedia-type applications like bounded delay or minimum bandwidth are irrelevant when applications are tolerant to latency or the bandwidth of the transmitted data.
- In some cases, the occasional delivery of a packet can be more than enough and in other cases, very high reliability requirements exist. In some other cases, delay *is* important when actuators are to be controlled in a real-time fashion by the sensor network. The packet delivery ratio is an insufficient metric.

- The vital requirement is the amount and quality of information that can be extracted at given sinks about the observed objects or area.
- Therefore, adapted quality concepts like reliable detection of events or the approximation quality of a, say, temperature map is important.

3. Fault tolerance

- The nodes may run out of energy or get damaged, or even interrupt the wireless communication between two nodes permanently. The redundant deployment is necessary for WSN to tolerate the node failure and using more number of nodes will be necessary even if all nodes functioned correctly.

4. Lifetime

- In many cases, the nodes will have to depend on a limited supply of energy using batteries. Replacing these energy sources in the field is usually not practicable and simultaneously, a WSN must operate at least for a given mission time. Hence, the lifetime of a WSN becomes a very important figure of merit. Hence an energy-efficient way of operation of the WSN is necessary. As an alternative to energy supplies, a limited power source must also be available on a sensor node.

- These sources are not powerful enough to ensure continuous operation but can provide some recharging of batteries.
- Under such conditions, the lifetime of the network should ideally be infinite. The lifetime of a network also has direct trade-offs against quality of service: investing more energy can increase quality but decrease lifetime.
- The precise *definition of lifetime* depends on the application. The simple option is to use the time until the first node fails as the network lifetime. Other options include the time until the network is disconnected in two or more partitions, the time until 50 % of nodes have failed etc.

5. Scalability

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

6. Wide range of densities

- In a WSN, the number of nodes per unit area that is the *density* of the network can vary considerably. Different applications will have very different node densities. Even within a given application, density can vary over time and space and density also does not have to homogeneous in the entire and the network should adapt to such variations.

7. Programmability

- The nodes need to process information and react flexibly on changes in their tasks. These nodes should be programmable and their programming must be changeable during operation when new tasks become important.

8. Maintainability

- WSN has to monitor its own health and status to change operational parameters or to choose different trade-offs. The network can also be able to interact with external maintenance mechanisms to ensure its extended operation at a required quality.

Topic 4

Challenges of WSN

Introduction

- To realize the characteristics requirements, the innovative mechanisms for a communication network have to be found.
- The particular challenge is the need to find mechanisms specific to the idiosyncrasies of a given application to support the specific quality of service, and maintainability requirements.
- These mechanisms also have to generalize to a wider range of applications and implementation of a WSN becomes necessary for every individual application.
- Some of the mechanisms that will form typical parts of WSNs are:

1. Multi-hop Wireless Communication

- Since wireless communication is a core technique, a direct communication between a sender and a receiver is faced with limitations.
- In particular, communication over long distances is only possible using high transmission power.
- The use of intermediate nodes as relays can reduce the total required power.
- Hence, for many forms of WSNs, multi-hop communication will be a necessary ingredient.

Energy Efficient Operation & Auto-configuration

- **2. Energy-efficient Operation: It** is a key technique for supporting long life time. The other options include energy-efficient data transport between two nodes or the energy-efficient determination of requested information. The non-homogeneous energy consumption – the forming of "hotspots" is an issue.
- **3. Auto-configuration:** A WSN will have to configure most of its operational parameters, independent of external configuration. As an example, nodes should be able to determine their geographical positions only using other nodes of the network so- called "self-location". The network should be able to tolerate failing nodes or to integrate new nodes.

4. Collaboration & In-network Processing

- In some applications, a single sensor is not able to decide whether an event has happened but several sensors have to collaborate to detect an event and only the joint data of many sensors provides enough information.
- Information is processed in the network in various forms to achieve this collaboration. This is opposite to having every node transmit all data to an external network and process it “at the edge” of the network.
- An example is to determine the highest or the average temperature within an area and to report that value to a sink. To solve such tasks, readings from individual sensors can be *aggregated* reducing the amount of data to be transmitted and hence improving the energy efficiency.

5. Data Centric

- Traditional communication networks are centered around the transfer of data between two specific devices, each equipped with one network address – the operation of such networks is thus **address-centric**.
- In a WSN, the nodes are deployed to protect against node failures or to compensate for the low quality of a single node's actual sensing equipment. Hence, switching from an address-centric paradigm to a **data-centric** paradigm in designing architecture and communication protocols is promising.
- An example for such a data-centric interaction will be to request the average temperature in a given location area, as opposed to requiring temperature readings from individual nodes.

6. Locality

- **T**he principle of locality will have to be embraced to ensure in particular, scalability.
- Nodes with limited should attempt to limit the state that they accumulate during protocol processing to only information about their direct neighbors.
- This will allow the network to scale to large numbers of nodes without having to depend on powerful processing at each single node.

7. Exploit Trade-offs

- Similar to locality principle, WSNs will have to depend to a large degree on exploiting various trade-offs between contradictory goals, both during system design and runtime.
- Examples for such trade-offs are - higher energy expenditure allows higher result accuracy, longer lifetime of the entire network trades off against lifetime of individual nodes and node density.
- If there is a depart from an address-centric view of the network, it may require new programming interfaces beyond the simple semantics of the conventional socket interface and allow concepts like required accuracy, energy/accuracy trade-offs etc.

Topic 5

Enabling Technologies for WSN

Introduction

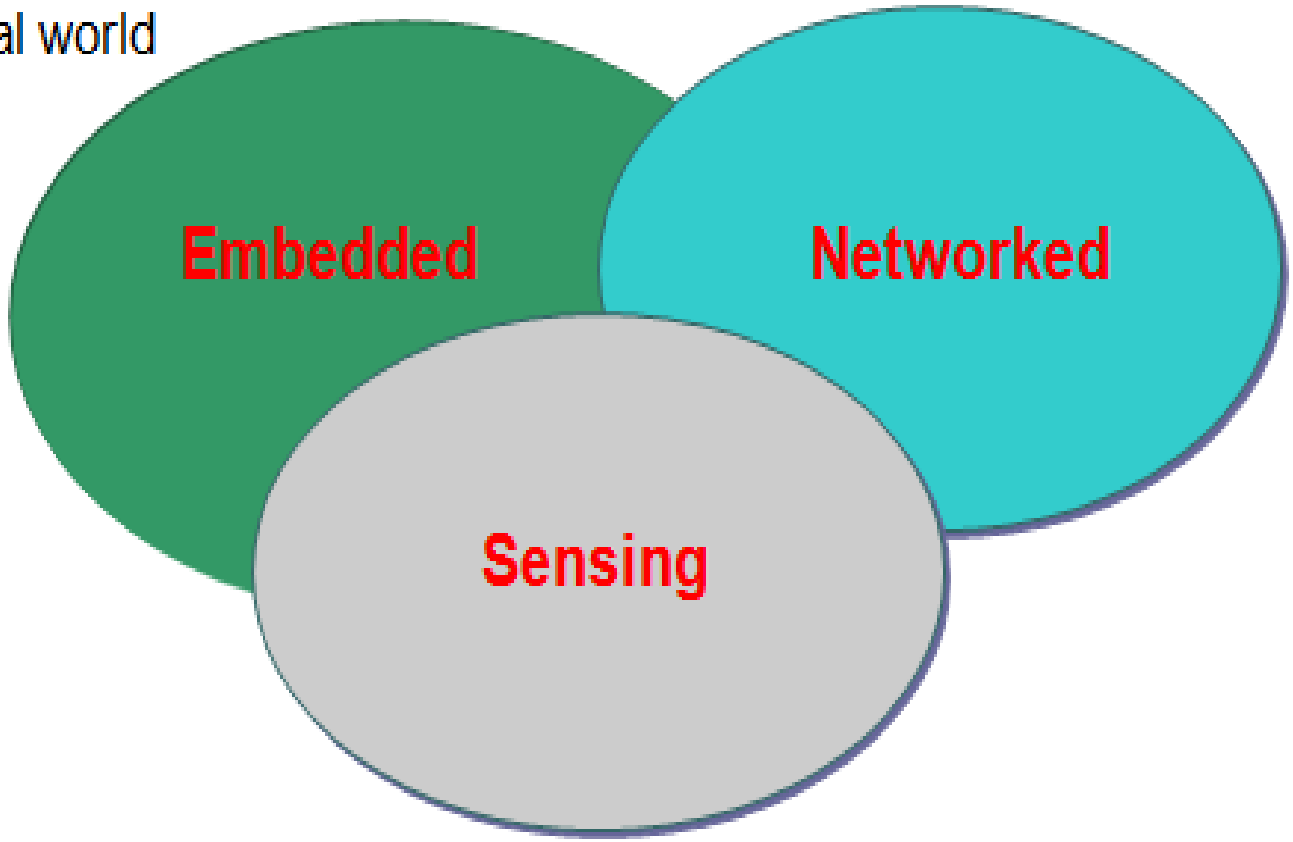
- It has only become possible to build wireless sensor networks with some fundamental advances in enabling technologies.
- First and foremost among these technologies is the miniaturization of hardware.
- Smaller feature sizes in chips have driven down the power consumption of the basic components of a sensor node to a level that the constructions of WSNs can be contemplated.
- This is particularly relevant to microcontrollers and memory chips and the radio modems responsible for wireless communication.

- Reduced chip size and improved energy efficiency is accompanied by reduced cost, which is necessary to make redundant deployment of nodes affordable.
- The actual sensing equipment is the third relevant technology next to processing and communication.
- However, it is difficult to generalize because of the vast range of possible sensors.

Fig 9 / Enabling Technologies for WSN

Embed numerous distributed devices to monitor and interact with physical world

Network devices to coordinate and perform higher-level tasks



Energy Scavenging

- These three basic parts of a sensor node have to be accompanied by power supply.
- This requirement depends on application, high capacity batteries lasting for long times and can efficiently provide small amounts of current.
- 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.
- Such a concept requires the battery to be efficiently chargeable with small amounts of current, which is not a standard ability.
- The counterpart to the basic hardware technologies is software.

- The architecture of the operating system or runtime environment has to support simple re-tasking, cross-layer information exchange and modularity to allow for simple maintenance.
- This software architecture on a single node has to be extended to a network architecture, where the division of tasks between nodes is considered.
- The third part to solve is how to design appropriate communication protocols.
- Figure 9 shows the enabling technologies for WSN.

Topic 6

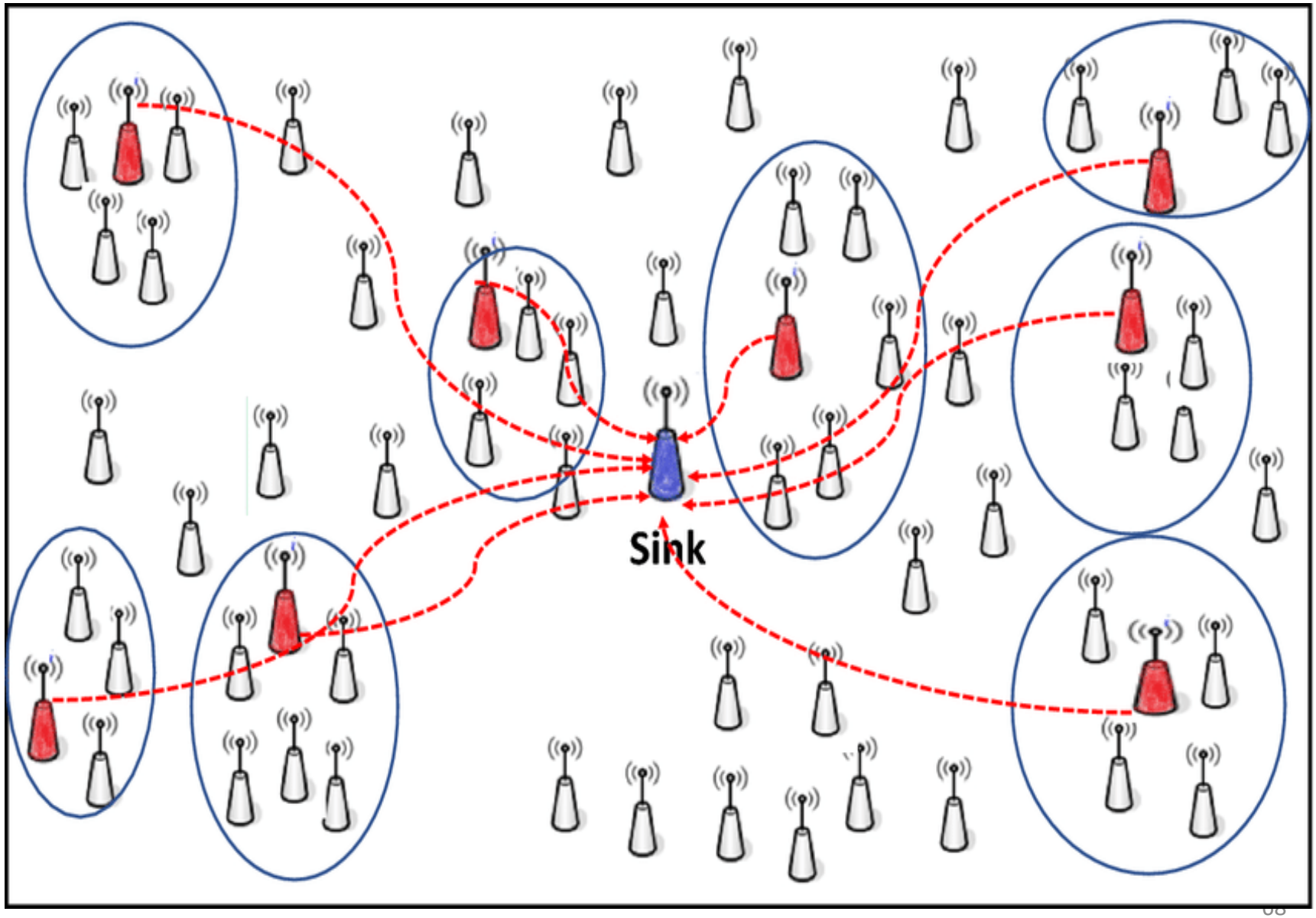
Types of Wireless Sensor Networks

Introduction

- The types of networks are decided based upon the environment so that they can be deployed underwater, underground, on land and so on. Different types of WSNs include:
  - Terrestrial WSNs
  - Underground WSNs
  - Underwater WSNs
  - Multimedia WSNs
  - Mobile WSNs

Terrestrial WSN's

- Terrestrial WSNs are capable of communicating base stations efficiently and consist of hundreds to thousands of wireless sensor nodes deployed either in an unstructured or structured manner.
- In an unstructured mode, the sensor nodes are randomly distributed within the target area dropped from a fixed plane.
- The preplanned or structured mode considers optimal placement, grid placement, and 2D, 3D placement models. In this WSN, the battery power is limited but equipped with solar cells as a secondary power source.
- The energy conservation of these WSNs is achieved by using low duty cycle operations, minimizing delays, and optimal routing, and so on.

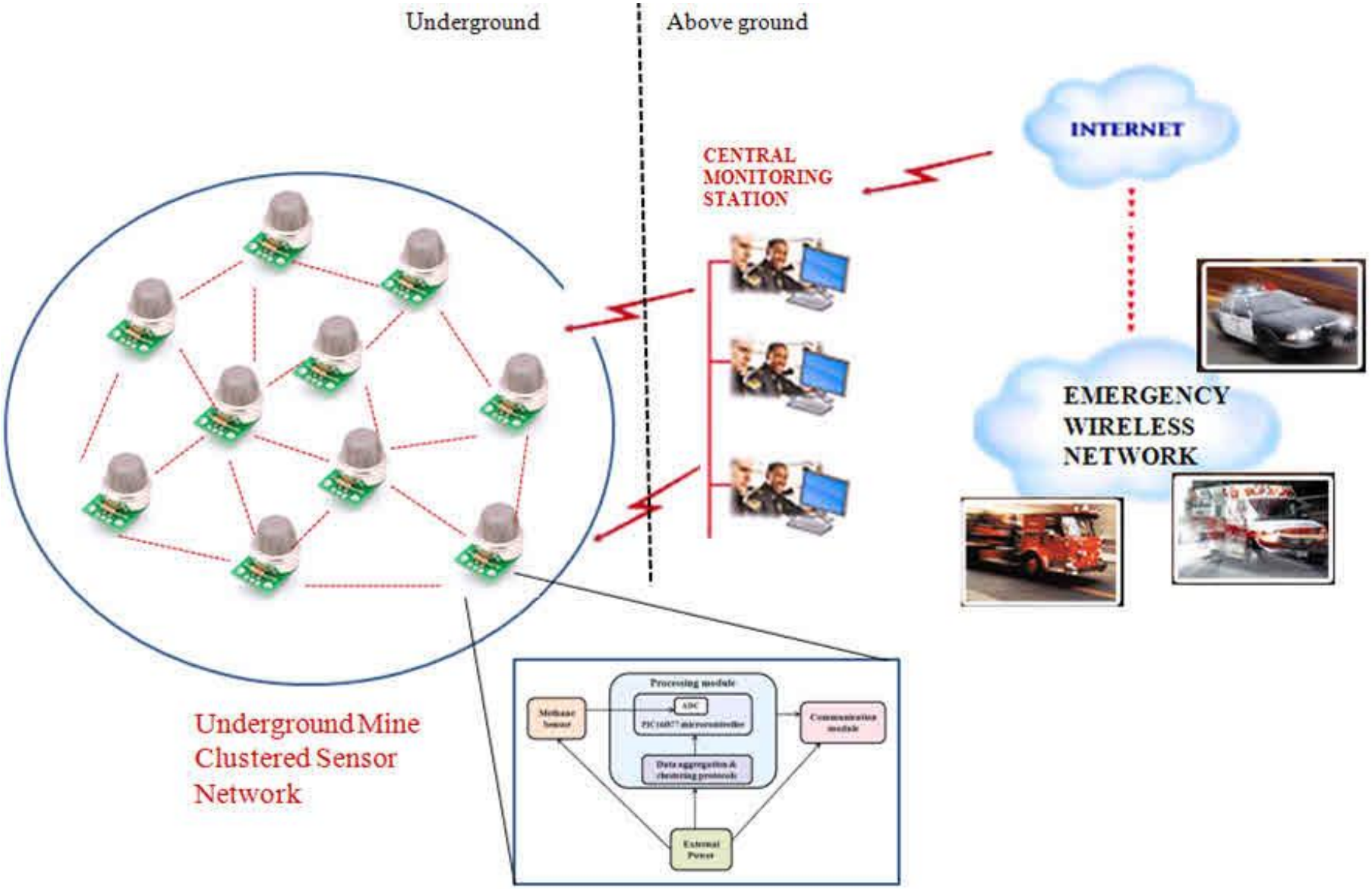

Sink

Underground WSN

- The underground wireless sensor networks are more expensive than the terrestrial WSNs in terms of deployment, maintenance, and equipment cost considerations and careful planning.
- The WSNs networks consist of several sensor nodes hidden in the ground to monitor underground conditions.
- To relay information from the sensor nodes to the base station, additional sink nodes are located above the ground.
- The underground wireless sensor networks deployed into the ground are difficult to recharge.

- The sensor battery nodes equipped with limited battery power are difficult to recharge
- In addition to this, the underground environment makes wireless communication a challenge due to the high level of attenuation and signal loss.

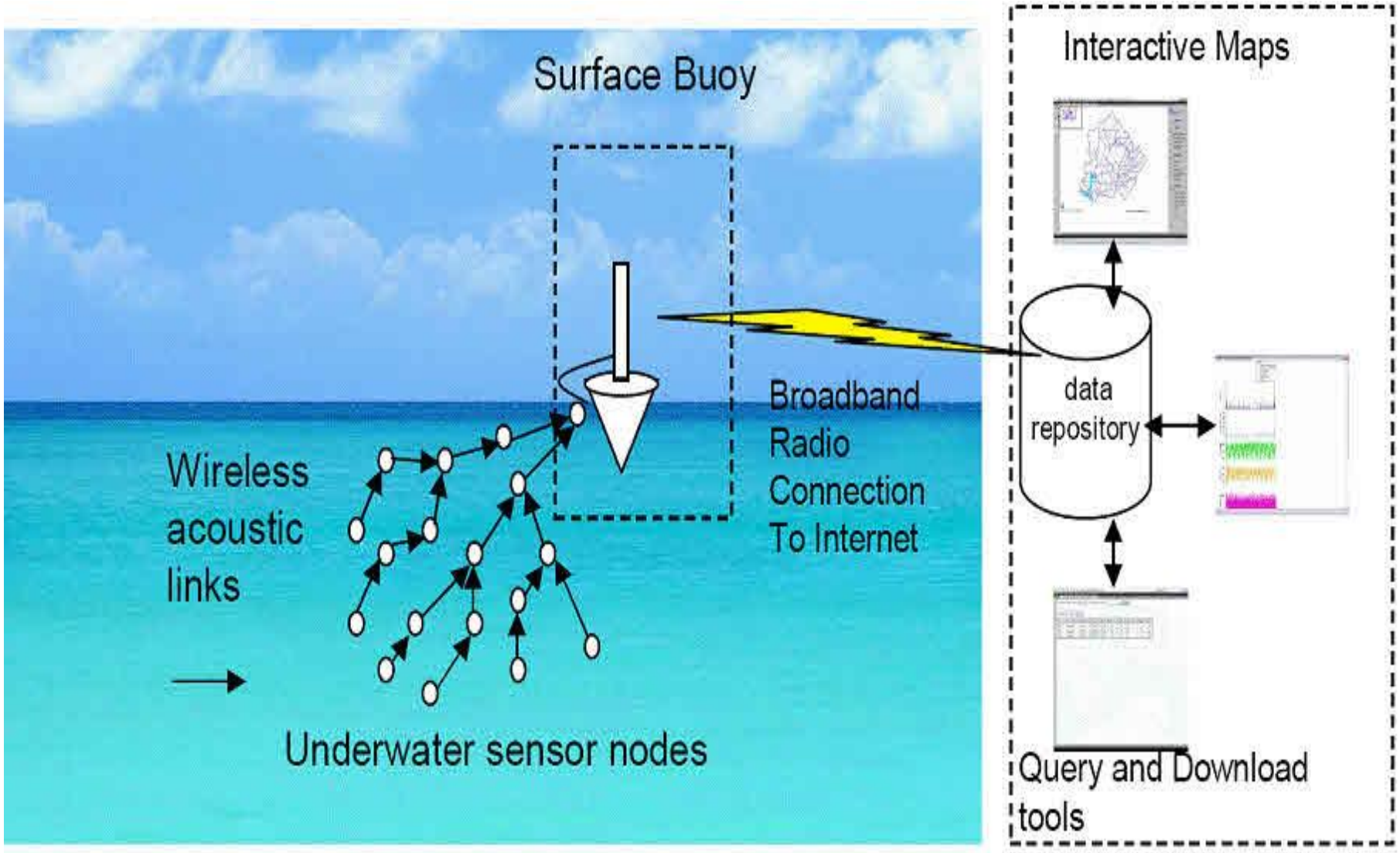
Fig 10 / Underground WSN



Under Water WSN

- More than 70% of the earth is occupied with water. These networks consist of several sensor nodes and vehicles deployed underwater.
- Autonomous underwater vehicles are used for gathering data from these sensor nodes. A challenge of underwater communication is a long propagation delay, and bandwidth and sensor failures.
- Underwater, WSNs are equipped with a limited battery that cannot be recharged or replaced.
- The issue of energy conservation for underwater WSNs involves the development of underwater communication and networking techniques.

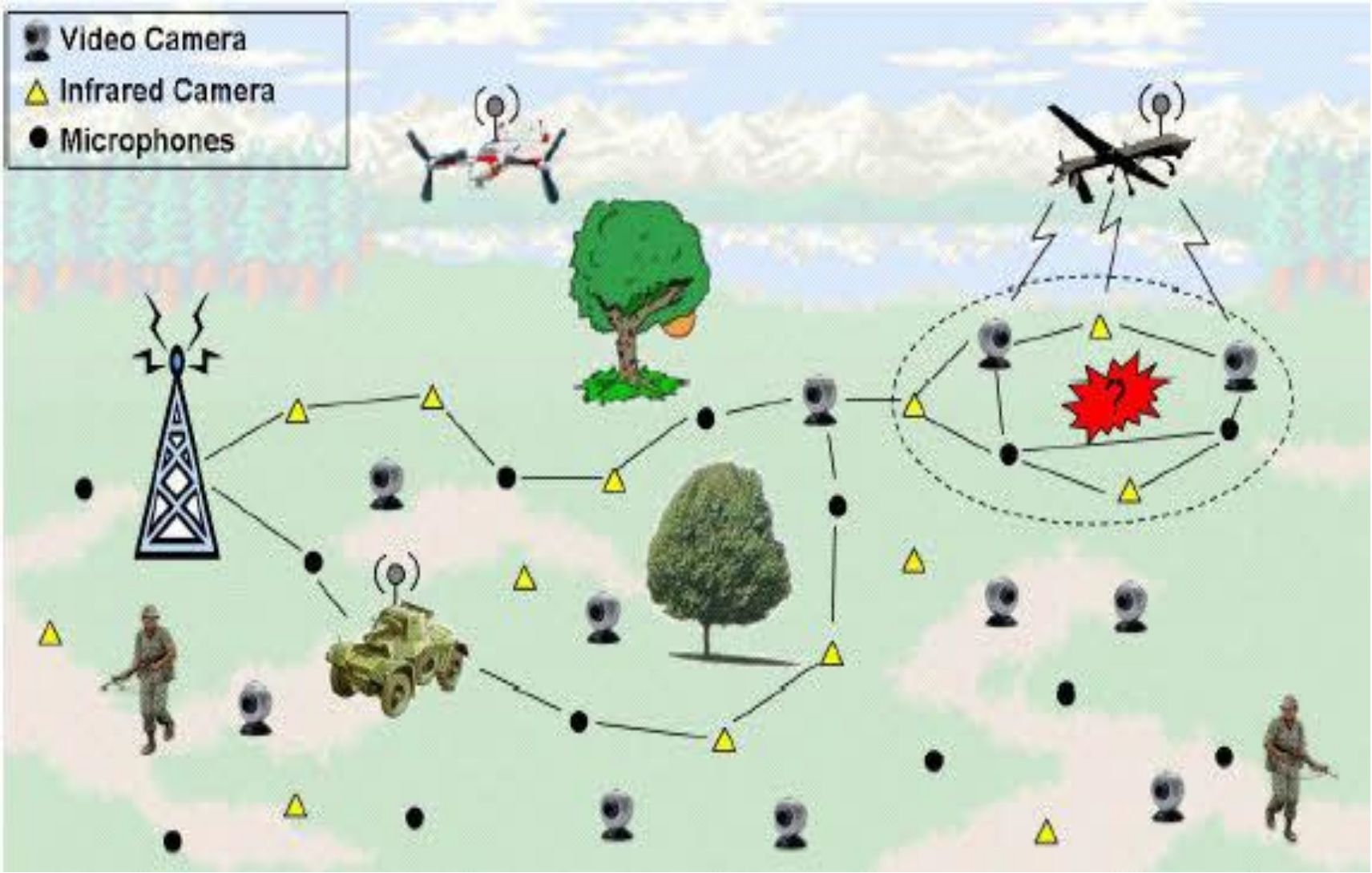
Fig 11 / Underwater WSN



Multimedia WSN

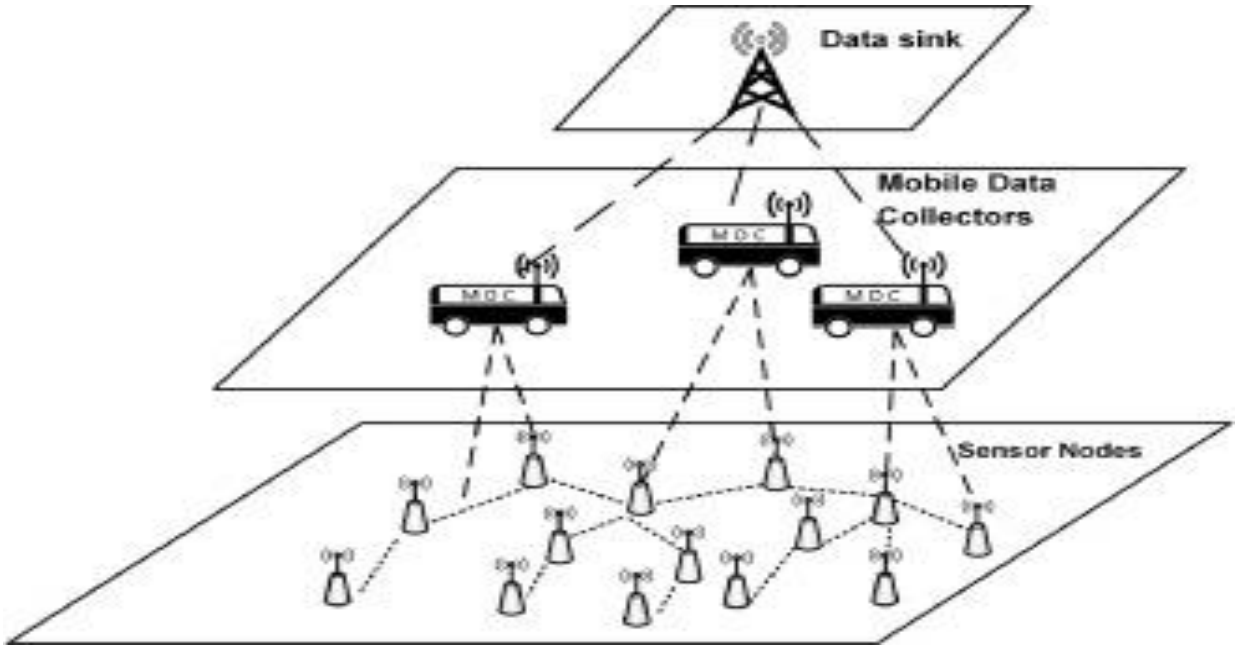
- Multimedia wireless sensor networks have been proposed to enable tracking and monitoring of events in the form of multimedia such as imaging, video, and audio.
- These networks consist of low-cost sensor nodes equipped with microphones and cameras.
- These nodes are interconnected with each other over a wireless connection for data compression, data retrieval, and correlation.
- The challenges with the multimedia WSN include high energy consumption, high bandwidth requirements, data processing, and compressing techniques.
- In addition to this, multimedia contents require high bandwidth for the content to be delivered properly and easily.

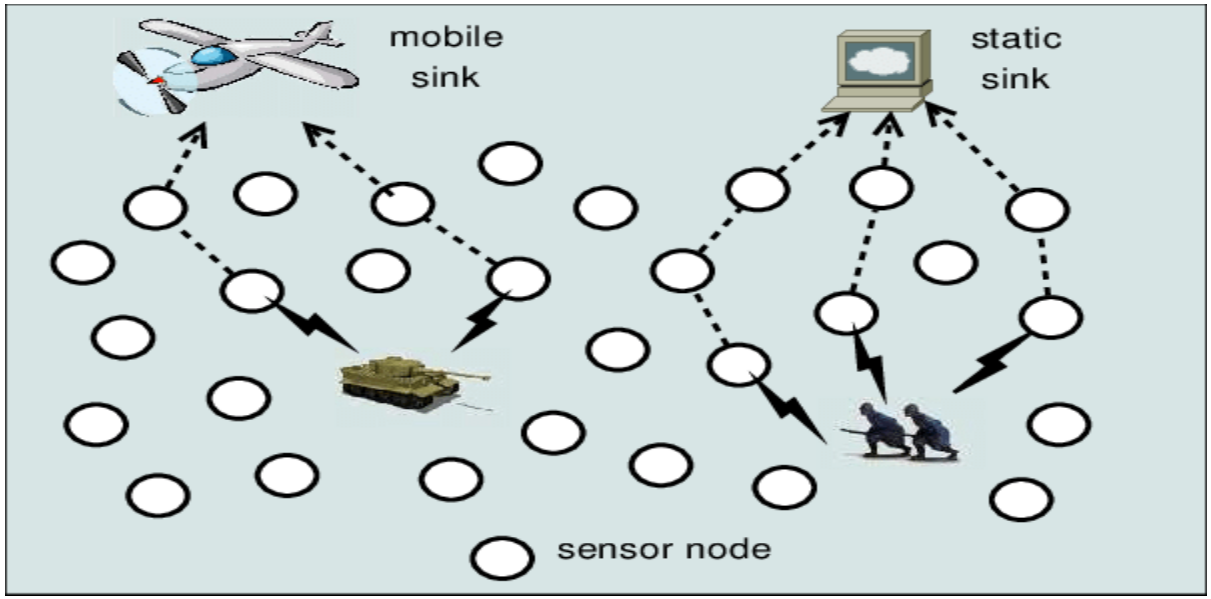
Fig 12 / Multimedia WSN



Mobile WSN

- These networks consist of a collection of sensor nodes that can be moved on their own and can be interacted with the physical environment.
- The mobile nodes can compute sense and communicate. Mobile wireless sensor networks are much more versatile than static sensor networks.
- The advantages of MWSN over static wireless sensor networks include better and improved coverage, better energy efficiency, superior channel capacity, and so on.


Data sink
Mobile Data Collectors
MDC
MDC
MDC
Sensor Nodes


mobile sink
static sink
sensor node

Classification of WSN's

- The classification of WSNs can be done based on the application but its characteristics mainly change based on the type.
- Generally, WSNs are classified into different categories like the following.
  - Static & Mobile
  - Deterministic & Nondeterministic
  - Single Base Station & Multi Base Station
  - Static Base Station & Mobile Base Station
  - Single-hop & Multi-hop WSN
  - Self Reconfigurable & Non-Self Configurable
  - Homogeneous & Heterogeneous

1. Static & Mobile WSN

- All the sensor nodes in several applications can be set without movement so these networks are static WSNs. Especially in some applications like biological systems uses mobile sensor nodes which are called mobile networks. The best example of a mobile network is the monitoring of animals.

2. Deterministic & Nondeterministic WSN

- In a deterministic type of network, the sensor node arrangement can be fixed and calculated. This sensor node's pre-planned operation is possible in simply some applications. In most applications, the location of sensor nodes cannot be determined because of different factors like hostile operating conditions and harsh environment, so these networks are called non-deterministic.

3. Single Base Station & Multi Base Station

- In a single base station network, a single base station is used and it can be arranged very close to the region of the sensor node. The interaction between sensor nodes can be done through the base station. In a multi-base station type network, multiple base stations are used and a sensor node is used to move data toward the nearby base station.

4. Static Base Station & Mobile Base Station

- Base stations are either mobile or static similar to sensor nodes. The static type base station includes a stable position close to the sensing area whereas the mobile base station moves in the region of the sensor so that the sensor nodes load can be balanced.

5. Single-hop & Multi-hop WSN

- In a single-hop type network, the arrangement of sensor nodes can be done directly toward the base station whereas, in a multi-hop network, both the cluster heads and peer nodes are utilized to transmit the data to reduce the energy consumption.

6. Self Reconfigurable & Non-Self Configurable

- In a non-self configurable network, the arrangement of sensor networks cannot be done by them within a network and depends on a control unit for gathering data. In wireless sensor networks, the sensor nodes maintain and organize the network and collaboratively work by using other sensor nodes to accomplish the task.

7. Homogeneous and Heterogeneous

- In a homogeneous wireless sensor network, all the sensor nodes mainly include similar energy utilization, storage capabilities and computational power.
- In heterogeneous network, some sensor nodes include high computational power as well as energy necessities as compared to others.
- The processing and communication tasks are separated consequently.

Model Question Bank

PART A

1. What is a sensor?
2. What is a sensor network?
3. Give the elements of WSN.
4. What are the basic components of a sensor node?
5. Differentiate between single hop and multi-hop networks.
6. Differentiate between flat and hierarchical network architectures.
7. What are the various topologies used in WSN?
8. Give any four applications of WSN.
9. How does wireless sensor network work?
10. What is the need for wireless sensor network?

11. Mention the challenges of wireless sensor networks.
12. What is event detection?
13. What is energy scavenging?
14. Differentiate between sensor and actuator.
15. What is quality of service?
16. List the types of WSN.
17. What is Multi-hop wireless communication?
18. What is data centric?
19. What is an active sensor?
20. State the deployment options.

PART B

1. Discuss briefly the various hardware components used in Single node architecture of WSN.
2. Explain the characteristics, constraints and challenges of WSN.
3. Write a short note on enabling technologies for wireless sensor networks.
4. Describe the types of wireless sensor networks in a brief manner.

Wireless Sensor Networks

Unit 2 / Architectures

Prepared By

Dr.S.Omkumar/Associate Prof

Syllabus / Unit 2

- Network Architecture- Sensor Networks- Scenarios- Design Principle, Physical Layer and Transceiver Design Considerations, Optimization Goals and Figures of Merit, Gateway Concepts, Operating Systems and Execution Environments- Introduction to TinyOS and nesC- Internet to WSN Communication.

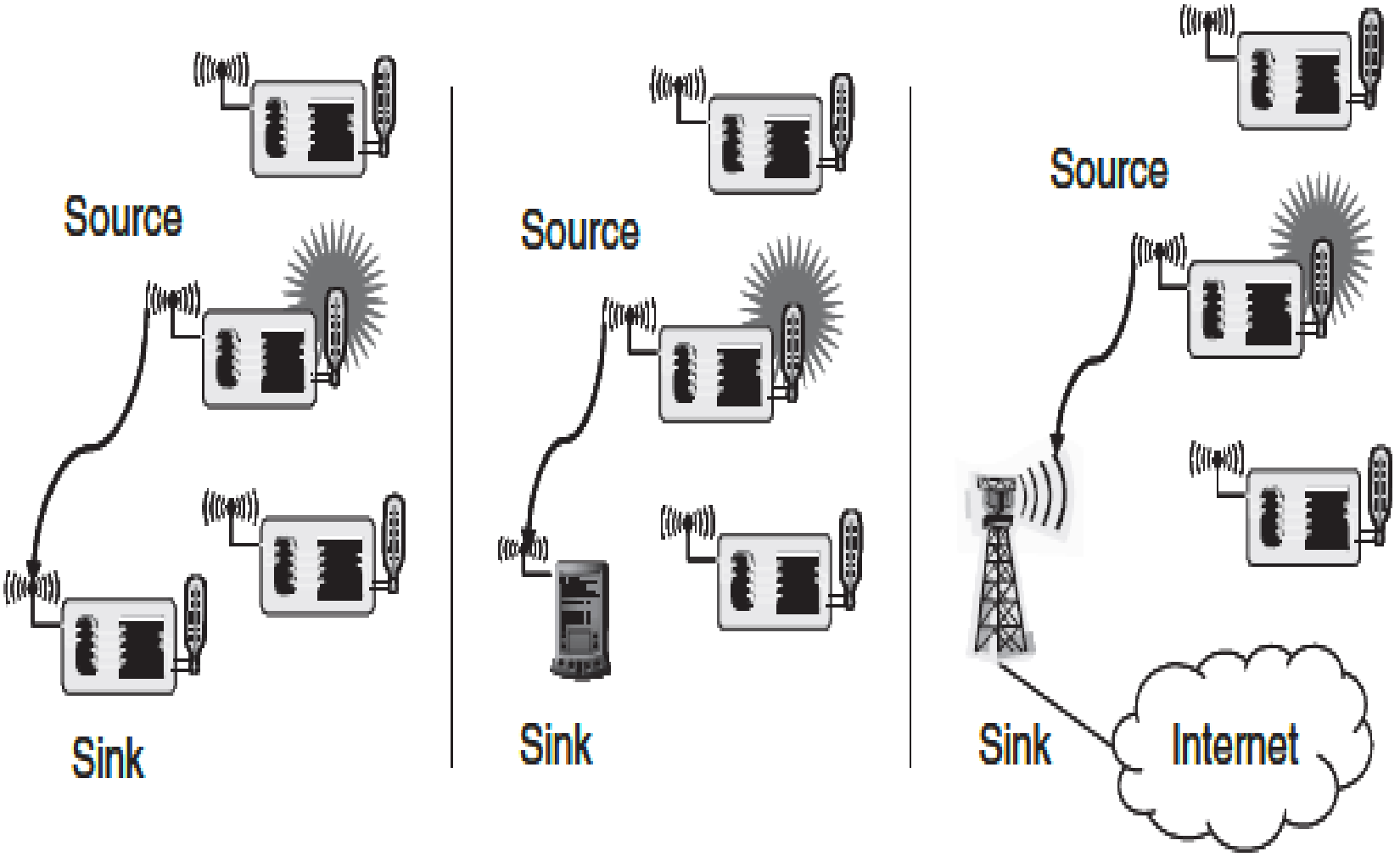
Topic 1

Sensor Networks Scenario

Types of Sources & Sinks

- A source is any entity in the network that provide information typically a sensor node and also be an actuator node that provides feedback about an operation.
- A sink is the entity where information is required.
- There are three options for a sink - it can belong to the sensor network or just another sensor/actuator node or can be an entity outside this network.
- For the second case, the sink can be an actual device to interact with the sensor network or can also be a gateway to another larger network such as Internet.
- These main types of sinks are shown in Figure 2.1, showing sources and sinks in direct communication.

Fig 1 / Three Types of Sinks

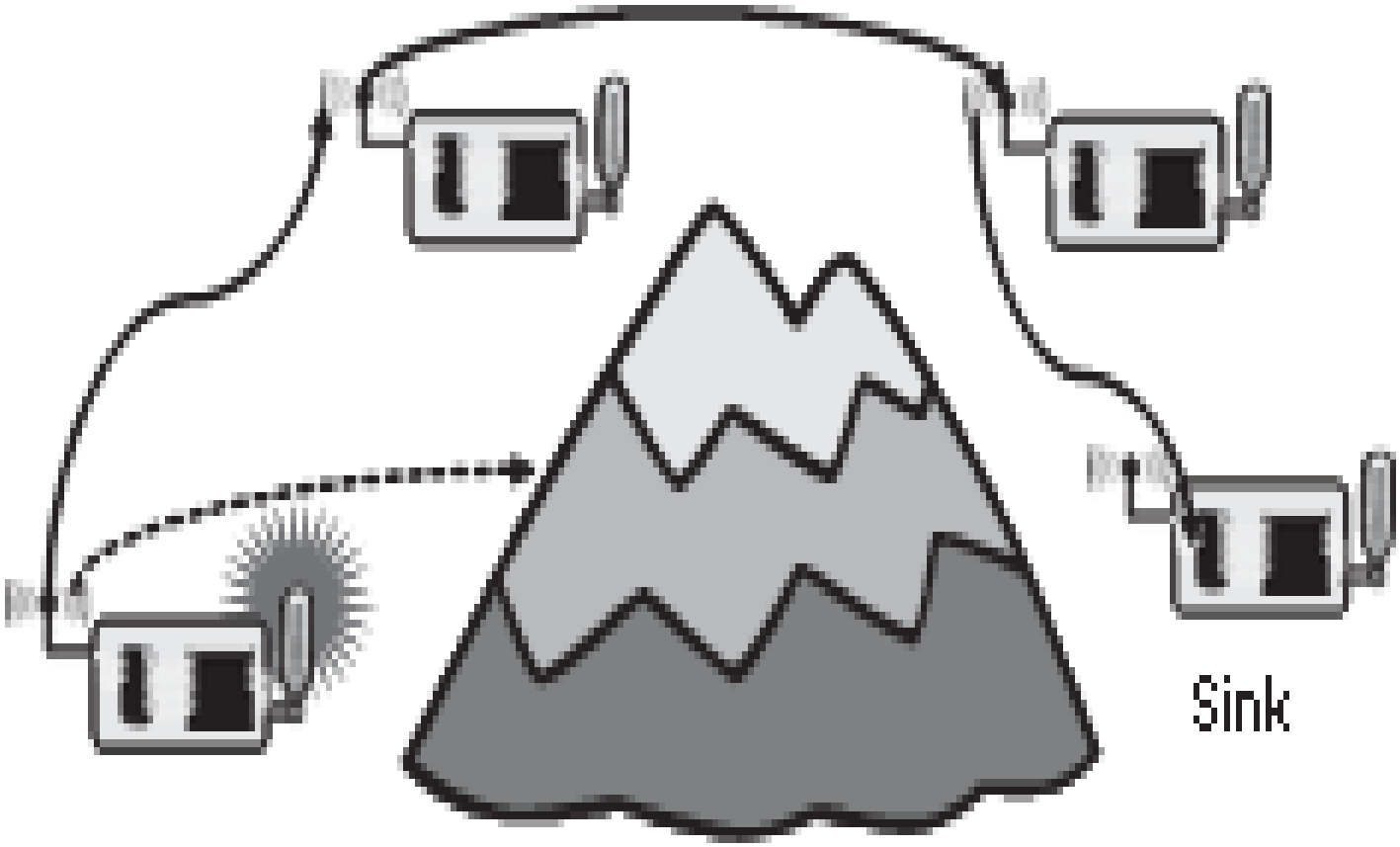


Single Hop Vs Multiple Hop Networks

- Because of limited distance, the simple direct communication between source and sink is not possible in WSN, which are intended to cover a lot environmental or agriculture applications.
- To overcome such limited distances, the relay stations are used with the data packets taking multi hops from the source to the sink.
- The multi-hopping is a working solution to overcome problems with large distances and can also improve the energy efficiency of communication.
- It consumes less energy to use relays instead of direct communication.

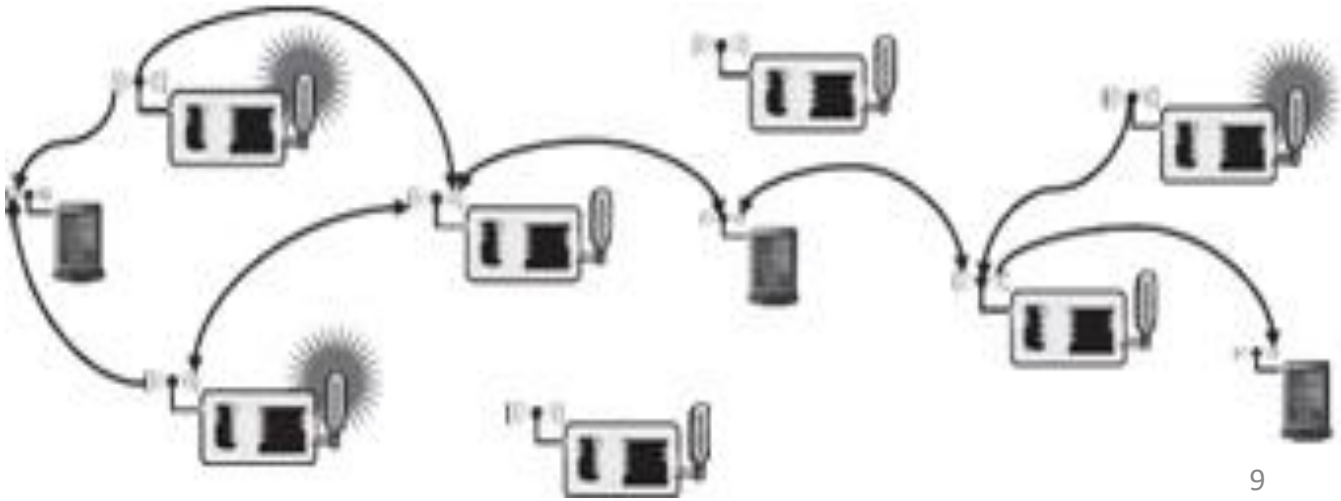
- The energy is actually wasted if intermediate relays are used for short distances and for large distance, the radiated energy dominate the fixed energy costs consumed in transmitter and receiver electronics.
- Moreover multi-hop networks operate in a store and forward fashion.

Fig 2 / Multihop Network



Multiple Sinks & Sources

- So far, only networks with a single source and a single sink have been explained.
- In many cases, there are multiple sources and/or multiple sinks present. In the most challenging case, multiple sources should send information to multiple sinks, where either all or some of the information has to reach all or some of the sinks. Figure 2.3 illustrates these combinations.



Types of Mobility

- 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:

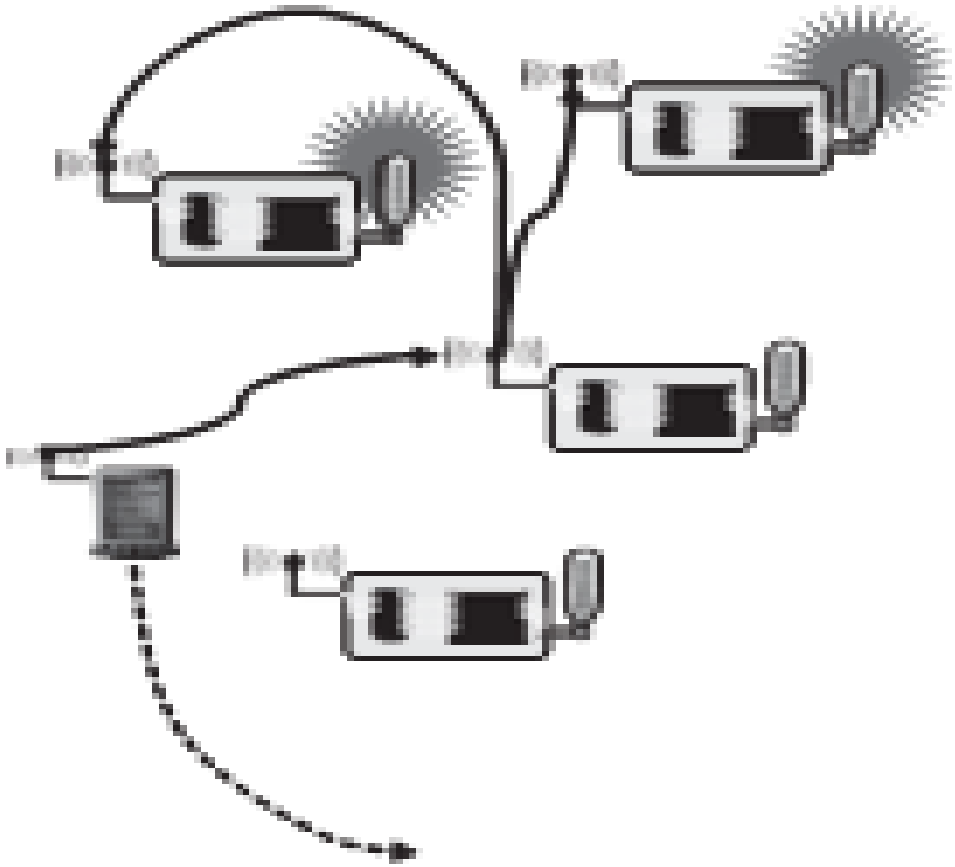
1. Node mobility:

The wireless sensor nodes can be mobile. The meaning of such mobility is highly application dependent. In node mobility, the network has to reorganize itself frequently enough to be able to function correctly. There are trade-offs between the frequency and speed of node movement on one hand and the energy required to maintain a desired level of functionality in the network on the other hand.

2. Sink mobility:

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

Fig 4 / Mobile Sink through Sensor Network



3. Event mobility:

- In applications like event detection and in tracking, the cause of the events or the objects to be tracked can be mobile. In such scenarios, the observed event is covered by a sufficient number of sensors at all time.
- Hence, sensors will wake up around the object, engaged in higher activity to observe the present object, and then go back to sleep. As the event source moves through the network, it is accompanied by an area of activity within the network. This is called as Frisbee Model as shown in Figure 2.4

Topic 2

Design Principles for WSN

Distributed Organization

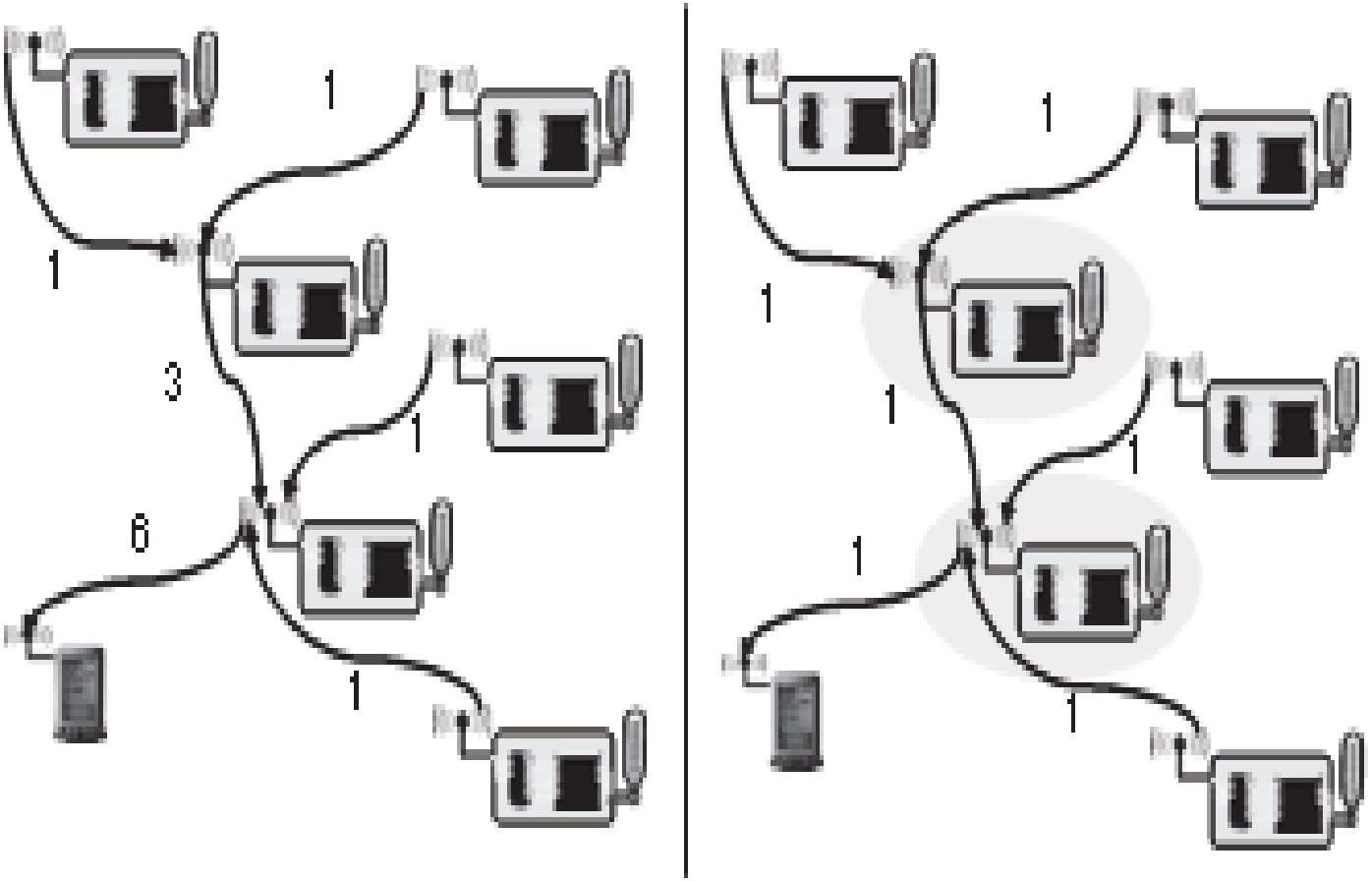
- Both the scalability and the robustness optimization goal are required to organize the network in a distributed fashion.
- When organizing a network in a distributed fashion, it is necessary to know potential shortcomings of this approach.
- In many cases, a centralized approach can produce solutions that perform better or require fewer resources.
- One possibility is to use centralized principles in a localized fashion by electing, out of set of equal nodes.
- Such elections result in a dynamic hierarchy.
- The election process should be repeated continuously until the elected node runs out of energy

In Network Processing Techniques

1. Aggregation:

- The simplest in-network processing technique is aggregation. The term aggregation means that information is aggregated into a condensed form in nodes intermediate between sources and sinks out of information provided by nodes further away from the sink. The aggregation function must be applied in the intermediate nodes as shown in Figure 2.5.

Fig 5 / Aggregation as an Example



2. Distributed Source Coding and Distributed Compression:

- The objective is to encode the information provided by several sensors by using traditional coding schemes, which may be complex for simple sensor nodes.
- The readings of adjacent sensors are going to be quite similar and correlated.
- Such **correlation** can be exploited instead of sending the sum of the data so that the overhead can be reduced.

3. Distributed and collaborative signal processing

- When complex computations on a certain amount of data is to be done, it can be more energy efficient to compute these functions on the sensor nodes using Fast Fourier Transform (FFT). In principle, this is similar to algorithm design for parallel computers. However the energy consumption of communication and computation are relevant parameters to decide between various algorithms.

4. Mobile code/Agent-based networking

- The idea of mobile code is to have a small, compact representation of program code to be sent from node to node. This code is executed locally for collecting measurements and then decides where to be sent next. This idea has been used in various environments

Adaptive Fidelity & Accuracy

- The idea of making fidelity of computation depends upon the amount of energy available for that particular computation.
- This concept can be extended from a single node to an entire network. As an example, consider a function approximation application.
- When more sensors participate in the approximation, the function is sampled at more points and the approximation is better. But more energy has to be invested.
- Hence, it is up to an application to define the degree of accuracy of the results and the task of the communication protocols to achieve this accuracy.

Data Eccentricity

- In traditional communication networks, the focus will be on the pair of communicating peers, the sender and the receiver of data.
- In a wireless sensor network, the interest of an application is actual information reported about the physical environment. This is applicable when a WSN is redundantly deployed such that any given event can be reported by multiple nodes.
- This method of concentrating on the data rather than identity of nodes is called data-centric networking.
- For an application, this means that an interface is exposed by the network where data only is addressed in requests.

Exploit Local Information

- Another useful technique is to exploit location information in the communication protocols when-ever such information is present.
- Since the location of an event is crucial information for many applications, mechanisms must be available to determine the location of sensor nodes.
- It can simplify the design and operation of communication protocols and can improve their energy efficiency.

Exploit Activity Patterns

- Activity patterns in a wireless sensor network are quite different from that of traditional networks.
- The data rate averaged over a long time can be very small.
- This can be detected by a larger number of sensors, breaking into a frenzy of activity, causing a well-known event shower effect.
- Hence, the protocol design should be able to handle such bursts of traffic by switching between modes of quiescence and of high activity.

Exploit Heterogeneity

- Sensor nodes can be heterogeneous by constructions, that is, they have larger batteries, farther-reaching communication devices, or more processing power.
- They can also be heterogeneous by evolution, that is, they started from an equal state, but scavenge energy from the environment due to overloading.
- Heterogeneity in the network is both a burden and an opportunity.
- The opportunity is an asymmetric assignment of tasks, giving nodes with more resources or more capabilities the more demanding tasks.
- The burden is asymmetric task assignments cannot be static but have to be reevaluated.

Component Based Protocol Stacks

- The concept is a collection of components which can form a basic “toolbox” of protocols and algorithms to build upon.
- All wireless sensor networks will require some form of physical, MAC, Link layer protocols, routing and transport layer functionalities.
- Moreover, “helper modules” like time synchronization, topology control can be useful.
- On top of these basic components, more abstract functionalities can then be built.
- The set of components active on a sensor node can be complex and will change from application to application.
- Protocol components will also interact with each other either by using simple exchange of data packets or by exchange of cross-layer information.

Topic 3

Physical Layer and Transceiver Considerations

Introduction

- Some of the crucial points influencing the Physical Layer design in wireless sensor networks are -
  - Low power consumption
  - Small transmit power and a small transmission range
  - Low duty cycle
  - Low data rates in the order of tens to hundreds kilobits per second
  - Low implementation complexity and costs
  - Low degree of mobility
  - Small form factor for the overall node

Energy Usage Profile

- The choice of a small transmission power leads to an energy consumption profile different from other wireless devices like cell phones.
- The radiated energy is small and the overall transceiver consumes much more energy than actually radiated.
- Then for small transmit powers, transmit and receive modes consume more or less the same power depending on the transceiver architecture.
- To reduce average power consumption in a low-traffic wireless sensor network, the transceiver must go into sleep state instead of just idling.
- During this startup time, no transmission or reception of data is possible.

- The third key observation is the relative costs of communications versus computation in a sensor node.
- A comparison of these costs depends for the communication part on BER requirements, range, transceiver type etc.

Choice of Modulation Scheme

- The following factors have to be balanced for the choice of modulation scheme -
  - Required data rate and symbol rate
  - Implementation complexity
  - Relationship between radiated power and target BER
  - Expected channel characteristics
- To maximize the time of transceiver in sleep mode, the transmit times should be minimized. The higher the data rate offered by a modulation, the smaller the time needed to transmit a given amount of data and the smaller the energy consumption. Moreover, the power consumption of a modulation scheme depends much more on the symbol rate than on the data rate.

Dynamic Modulation Scaling

- To adapt the modulation scheme to the current situation, an approach called dynamic modulation scaling is employed.
- For the case of m-ary QAM, a model has been developed with the symbol rate ‘B’ and the number of levels per symbol ‘m’ as parameters.
- This model expresses the energy required per bit and also the achieved delay per bit, taking into account the higher levels of modulation.
- Hence the bit delay decreases for increasing values of ‘B’ and ‘m’. The energy per bit depends much more on ‘m’ than on ‘B’.

- For the particular parameters chosen, both energy per bit and delay per bit can be minimized for the maximum symbol rate.
- With modulation scaling, a packet is equipped with a delay constraint, from which directly a minimum required data rate can be derived.

Antenna Considerations

- The desired small form factor of the overall sensor nodes restricts the size and the number of antennas.
- If the antenna is much smaller than the carrier's wavelength, it is difficult to achieve good antenna efficiency.
- In case of small sensor node cases, it will be difficult to place two antennas with suitable distance to achieve receive diversity.
- The antennas should be spaced apart at least 40–50% of the wavelength used to achieve good effects from diversity.

- In addition, radio waves emitted from an antenna close to the ground are faced with higher path-loss coefficients than the common value $\alpha = 2$ for free-space communication.
- Moreover, depending on the application, antennas must not protrude from the casing of a node to avoid possible damage to it.
- These restrictions limit the quality and characteristics of an antenna for wireless sensor nodes.

Topic 4

Optimization of Goals & Figure of Merit

Introduction

- The following techniques will optimize a network, compare solutions, decide a better approach for a given application, and turn optimization goals into measurable figures of merit.

Quality of Service

- WSNs differ from other conventional communication networks in the type of services they offer.
- These networks only move bits from one place to another.
- Such QoS can be regarded as a low-level, networking-device attributes like bandwidth, delay, jitter or as a high-level, user attributes like perceived quality of a voice communication or a video transmission.
- But high-level QoS attributes in WSN highly depend on the application.
- Some generic possibilities are:

1. Event detection/reporting probability

   The probability of an event that actually occurred is not detected or not reported to an information sink

2. Event classification error

   If events are to be both detected and classified, the error in classification must be small.

3. Event detection delay

   The delay between detecting an event and reporting to all interested sinks

4. Missing reports

   The probability of undelivered reports should be small in periodic reporting applications.

5. Approximation accuracy

   For function approximation applications, the average/maximum absolute error with respect to the actual function.

6. Tracking accuracy

   In Tracking applications, the reported position should be as close to the real position and the error should be small.

Energy Efficiency

- The most commonly considered aspects of energy efficiency are:

1. Energy per correctly received bit

   The average amount **of** energy to transport one bit of information from the source to the destination.

2. Energy per reported event

   The average energy spent to report one event

3. Delay/energy trade-offs

   The notion of “urgent” events to justify the increased energy investment for a speedy reporting of events.

4. Network lifetime

- The time for which the network is operational to fulfill its tasks starting from a given amount of stored energy.
  - **Time to first node death: F**irst node in the network run out of energy and stop operating
  - **Network half-life:** When 50% of the nodes run out of energy and stopped operating.
  - **Time to partition: F**irst partition of the network in two or more disconnected parts occur
  - **Time to Loss of Coverage: F**or the first time any spot in the deployment region is no longer covered by any node's observations.
  - **Time to failure of first event notification:** The unreachable part of the network does not want to report any events in the first place.

Scalability

- The ability to maintain performance characteristics irrespective of the size of the network is called scalability.
- Scalability requires consistent state such as addresses or routing table entries to be maintained.
- Hence, the need to restrict such information is enforced with the resource limitations of sensor nodes with respect to memory.
- The need for extreme scalability has direct consequences on the protocol design.
- Architectures and protocols should implement appropriate scalability support rather than trying to be as scalable as possible.

Robustness

- Related to QoS and scalability requirements, wireless sensor networks should also exhibit an appropriate robustness.
- They should not fail just because a limited number of nodes run out of energy, or because their environment changes.
- These failures have to be compensated by finding other routes.
- A precise evaluation of robustness is difficult in practice and depends mostly on failure models for both nodes and communication links

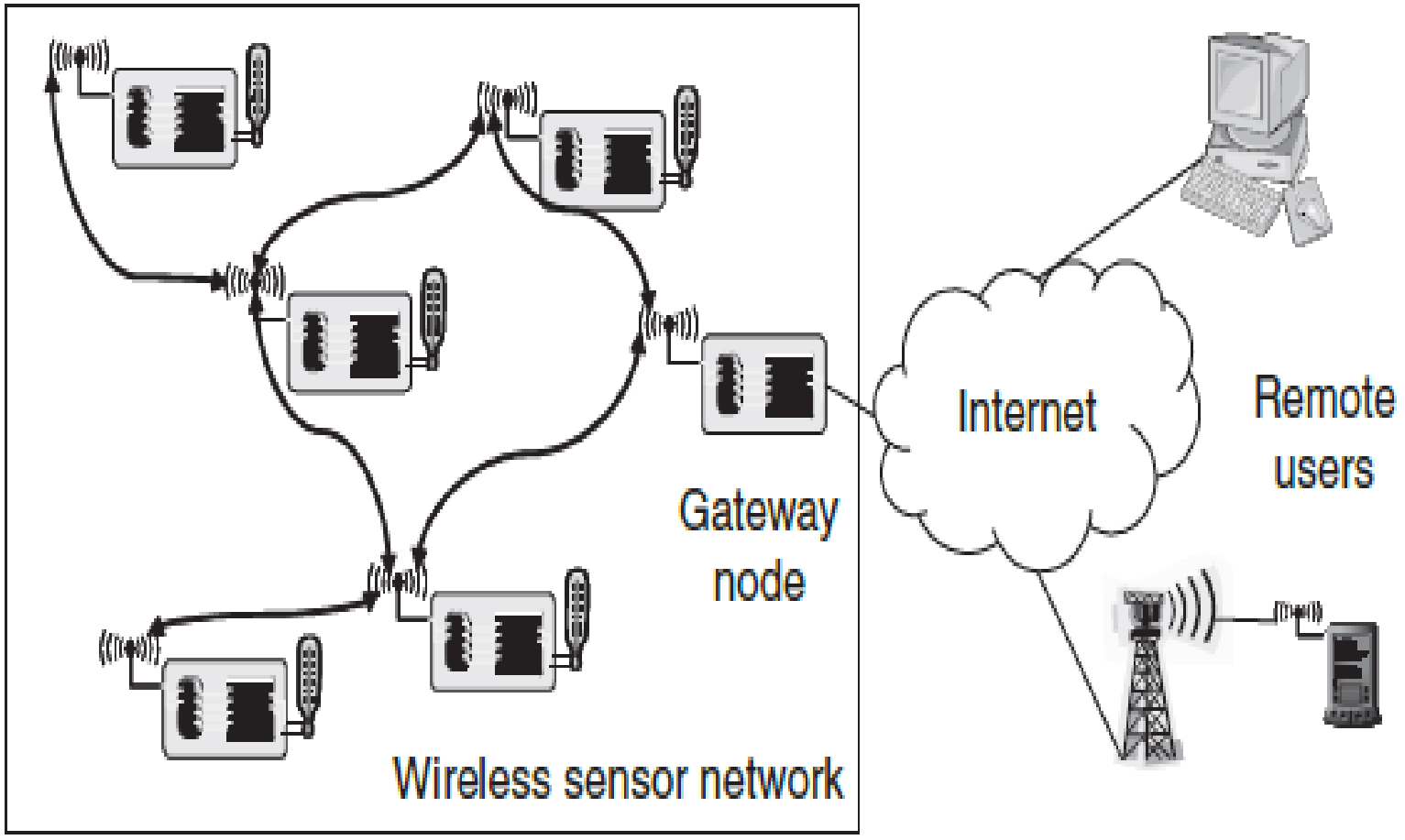
Topic 5

Gateway Concepts

Need for Gateways

- For practical deployment, the sensor network has to interact with other information devices. The standard example is to read the temperature sensors in one's home while traveling. Figure 2.6 shows the networking scenario.
- The WSN has to exchange data with such a mobile device or with some sort of gateway which provides the physical connection to the Internet.
- The first option is to regard a gateway as a simple router between Internet and sensor network. This will entail the use of Internet protocols within the sensor network.
- The next option is to design the gateway as an actual application-level gateway on the basis of the application-level information.

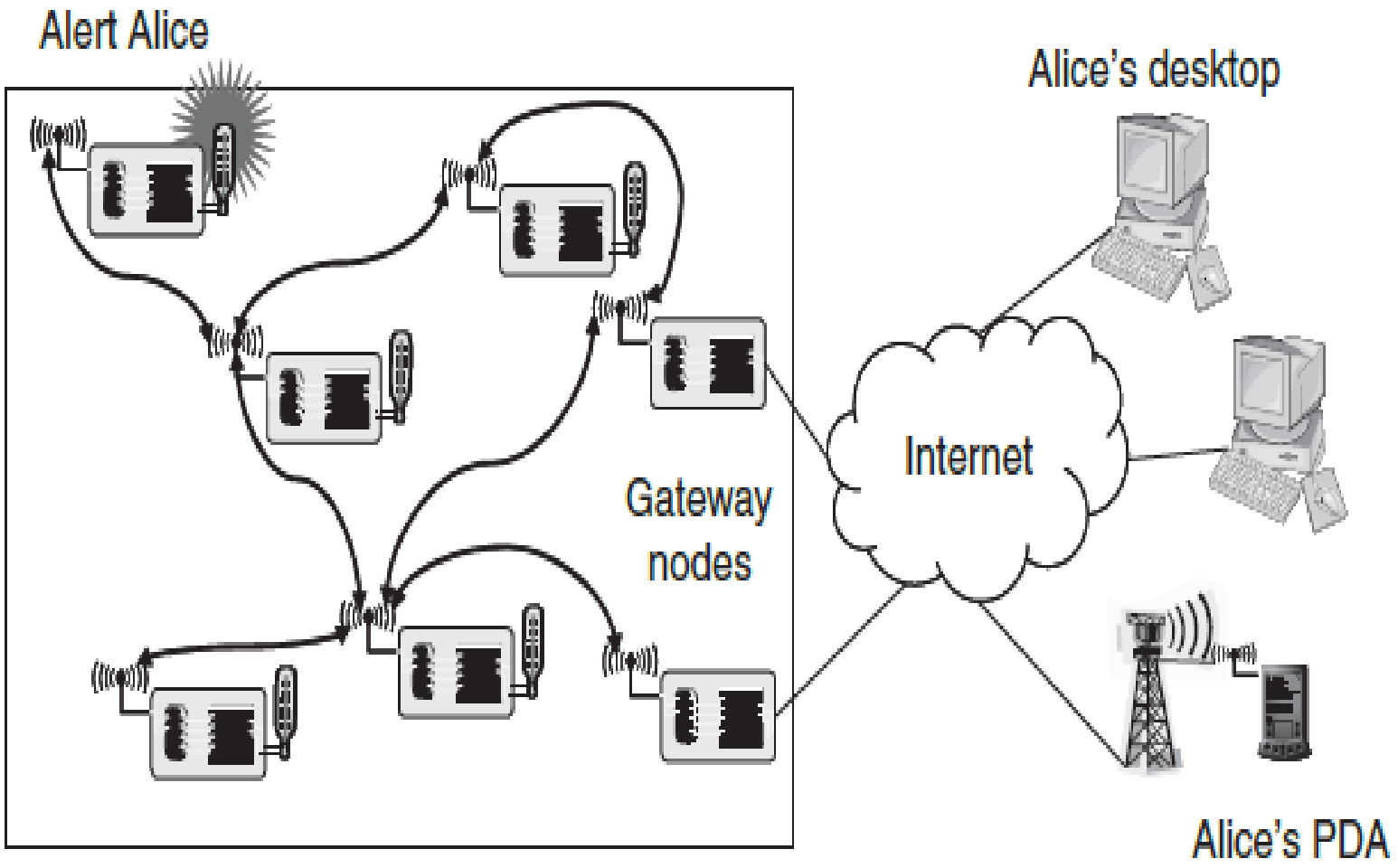
Fig 6 / WSN with Gateway Node



WSN to Internet Communication

- For example, a sensor node wants to deliver an alarm message to some Internet host.
- The first problem to solve is to find the gateway from within the network.
- If several gateways are available, the selection of the particular route and gateway for a given destination have to be done.
- To handle several gateways the option is to build an IP overlay network on top of the sensor network. Figure 2.7 shows the mapping of Alice to a concrete IP address.
- The sensor node has to include sufficient information such as IP address and port number in its own packets.
- The gateway in turn will extract this information and translate it into IP packets.

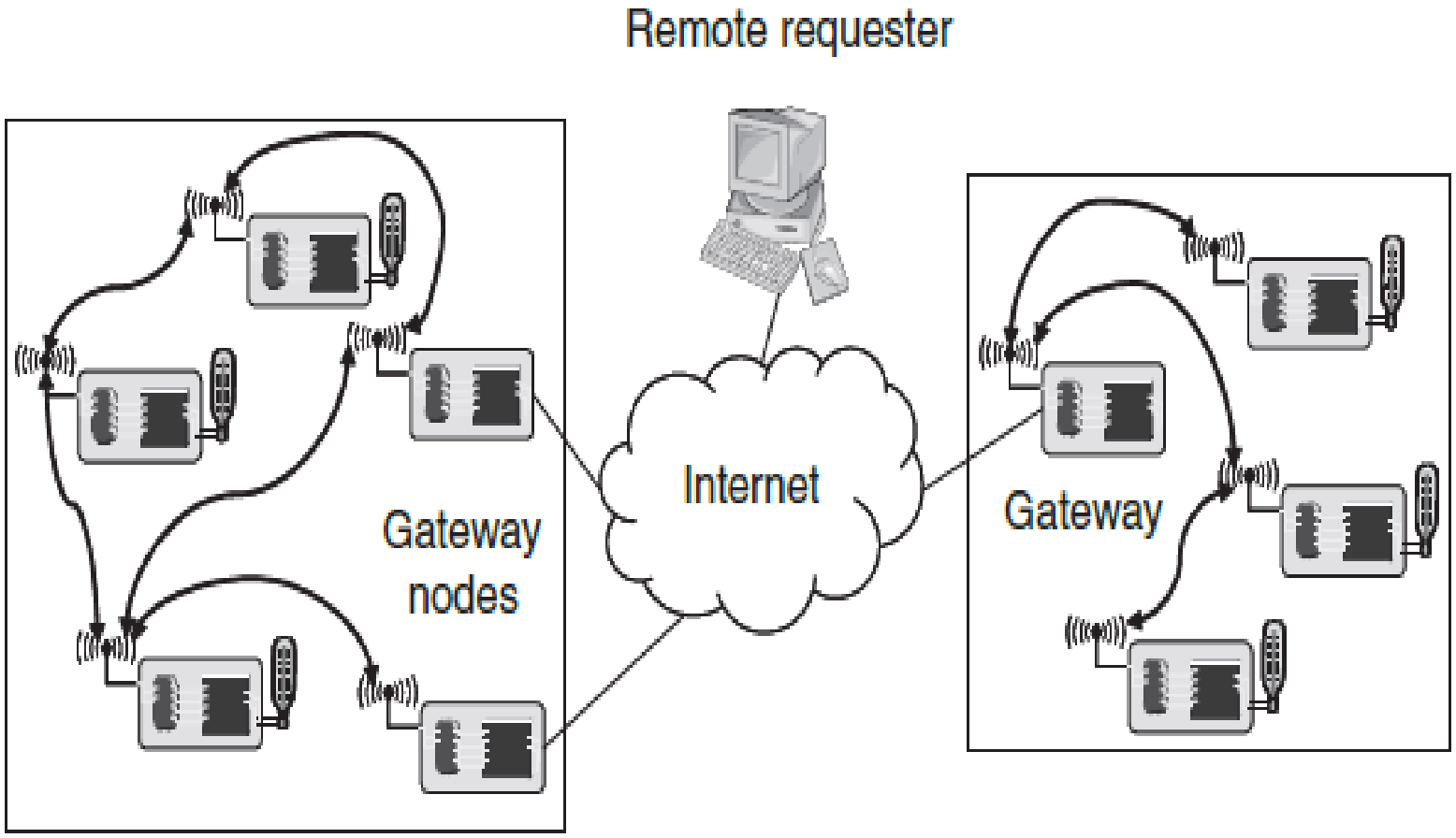
Fig 7 / WSN to Internet Communication



Internet to WSN Communication

- For example, a mobile requester is equipped with a WSN transceiver which has all the necessary protocol components.
- In this case, the requesting terminal can be a direct part of the WSN.
- First of all, identification of the sensor network in the desired location and existence of a gateway node has to be done.
- Once the requesting terminal has obtained this information, then the actual services can be accessed.
- The requesting terminal can instead send a properly formatted request to this gateway which acts as an application-level gateway that can answer this request.

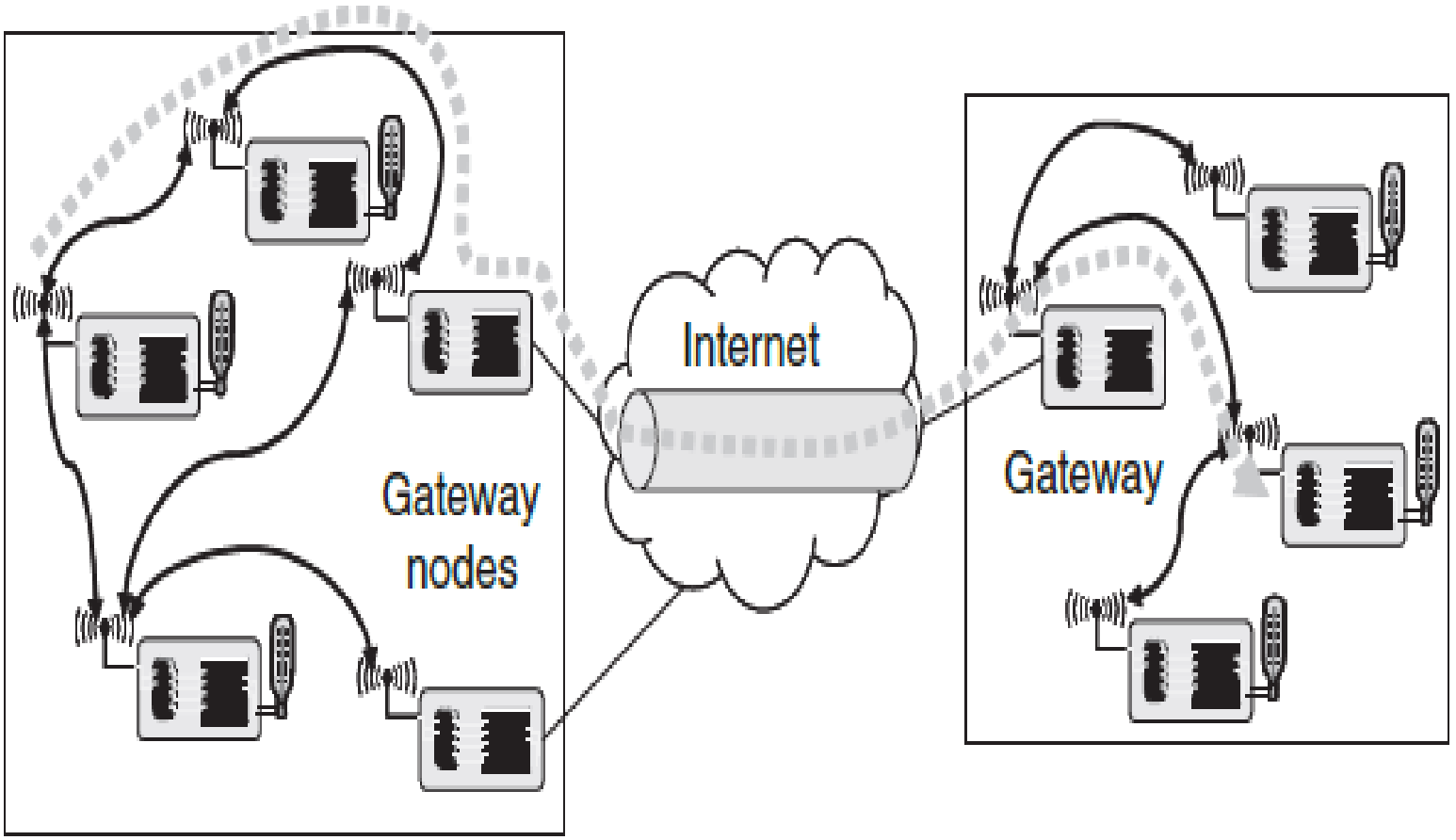
- The gateway translates this request into the proper intra sensor network protocol interactions. Figure 8 shows the scenario.



WSN Tunneling

- The gateways can also act as simple extensions of one WSN to another WSN.
- The idea is to build a larger virtual WSN “tunneling” all protocol messages between these two networks and simply using the Internet as a transport network as shown in Figure 2.9.
- But care has to be taken not to confuse the virtual link between two gateway nodes with a real link.
- Otherwise, protocols that depend on physical properties of a communication link can get confused.

Fig 9 / WSN Tunneling



Topic 6

Operating Systems & Execution Environment

Embedded Operating Systems

- The traditional tasks of an operating system are controlling and protecting the access to resources, managing their allocation to users and support for concurrent execution of processes.
- These tasks are only partially required in an embedded system and these systems do not have required resources to support a full-blown operating system.
- In particular, the need for energy-efficient execution requires support for energy management or Dynamic Voltage Scaling (DVS) techniques.
- Also, external components like sensors, the radio modem, or timers should be handled easily and efficiently.
- All this requires an appropriate programming model to structure a protocol stack and explicit support for energy management.

Programming Paradigms

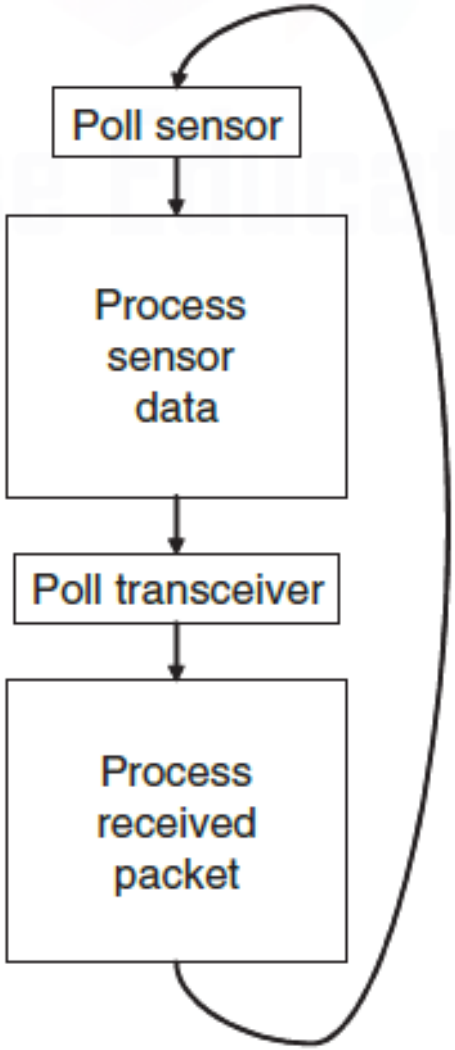
1. Concurrent Programming

- The support for concurrent execution is crucial for WSN nodes to handle data coming from arbitrary sources like multiple sensors or the radio transceiver at arbitrary points in time.
- For example, a system can poll a sensor to decide whether data is available and process the data, then poll the transceiver to check whether a packet is available and then immediately process the packet and so on.

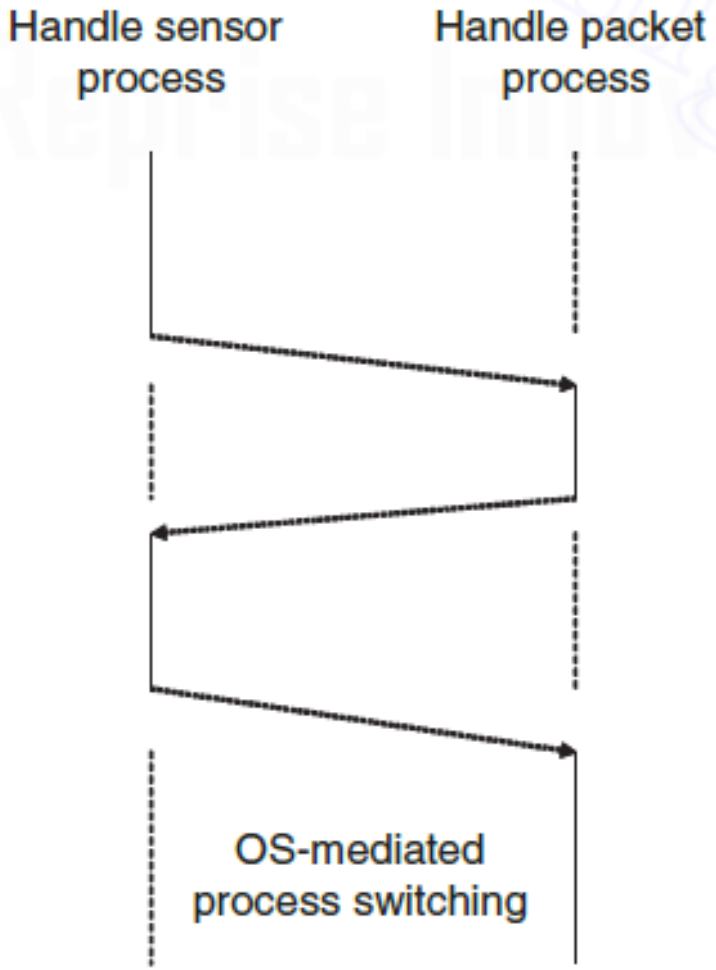
2. Process Based Concurrency

- Most general-purpose operating systems support concurrent execution of multiple processes on a single CPU. Hence such a process-based approach can be used to support concurrency in a sensor node as illustrated in (b) of Figure 2.10.
- Mapping such an execution model of concurrent processes to a sensor node shows that there are some granularity mismatches.
- This problem is severe for smaller tasks to be executed when compared to overhead.

Fig 10 / Programming Models for WSN



Sequential programming model

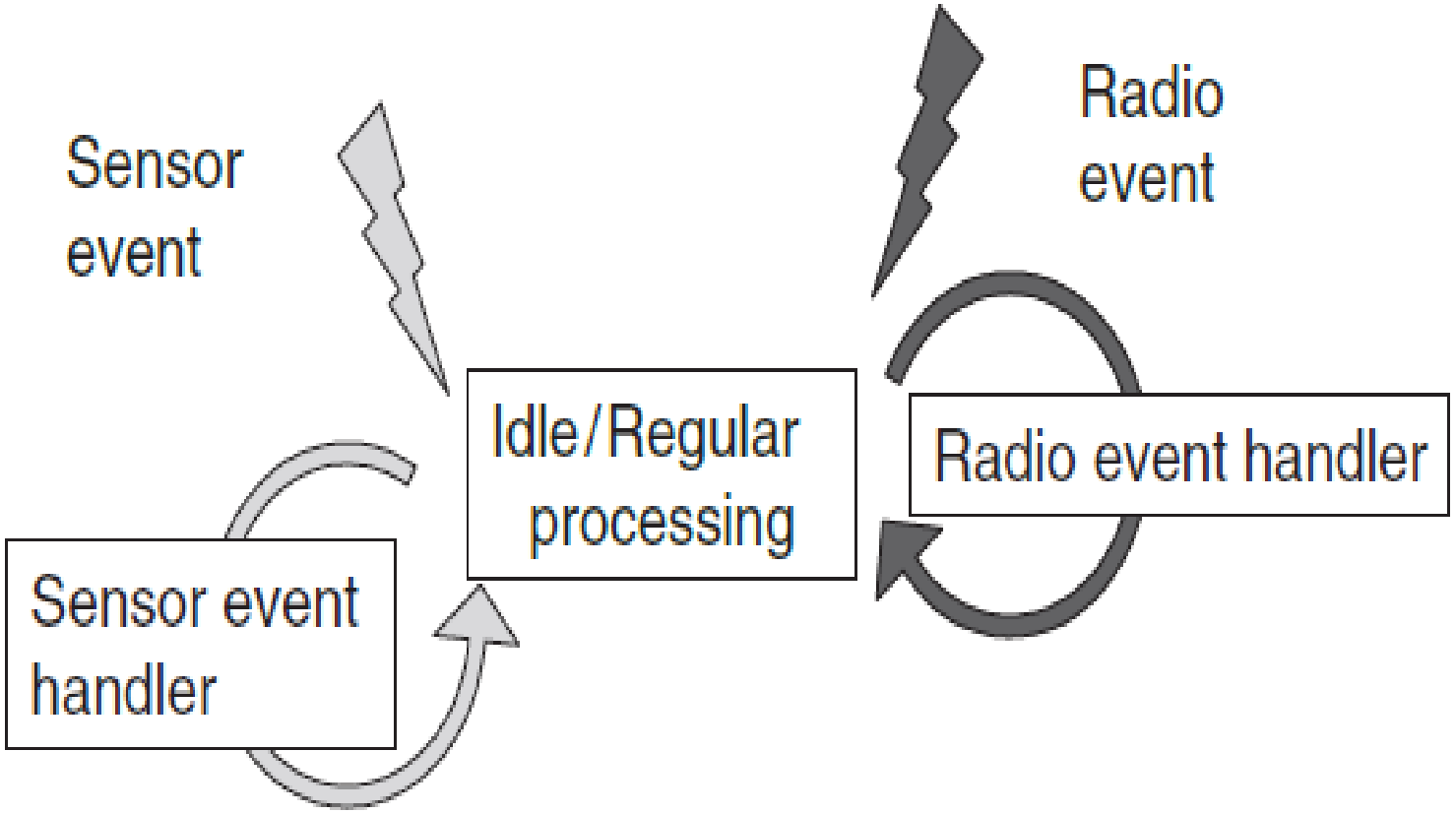


Process-based programming model

3. Event-based Programming

- The system waits for any event to happen, where an event can be the availability of data from a sensor, or arrival of a packet.
- Such an event is then handled by a short sequence of instructions that stores the occurrence of event and necessary information.
- This is called event based programming model as shown in Figure 2.11.
- This programming model distinguishes between two different “contexts”: - time-critical event handlers (execution cannot be interrupted) and for the processing of normal code (only triggered by the event handlers).

Fig 11 / Event Based Programming Model



4. Interfaces to Operating System

- In WSNs, the interfaces should be accessible from protocol implementations.
- This interface is closely tied with the structure of protocol stacks.
- For example Application Programming Interface (API) comprises, a “functional interface, object abstractions, and detailed behavioral semantics”.
- Abstractions are wireless links, nodes and so on.
- The possible functions include state inquiry, manipulation, transmitting of data, access to hardware and setting of policies.

Operating System & Protocols Stack

- In communication protocol structuring, the individual protocols are stacked on top of each other, each layer only using functions of the layer directly below.
- This layered approach has multiple benefits in keeping the entire protocol stack manageable.
- As an example, consider the use of information about the strength of the signal received from a communication partner.
- This physical layer information can be used to assist in networking protocols to decide about routing changes.
- Hence, one single source of information can be used by many other protocols not directly associated with the source of this information.
- Such **cross-layer information exchange** is one way to loosen the strict confinements of the layered approach.

Dynamic Energy & Power Management

1. Probabilistic State Transition Policies

- These policies regulate the transition between various sleep states.
- They start out by considering sensors randomly distributed over a fixed area and events arrive with certain temporal distributions and spatial distributions.
- This allows them to compute probabilities for the time to the next event, once an event has been processed.

2. Controlling Dynamic Voltage Scaling

- For example, only a single task has to be run in an operating system. Hence, a clever scheduler is required to decide exact clock rate to use in that situation to meet all deadlines. This can require feedback from applications for example, video playback in reference.

3. Trading off fidelity against energy consumption

- There are certain tasks that can be computed with a higher or lower level of accuracy. The fidelity achieved by such tasks is a candidate for trading off against other resources. In a WSN, the natural trade-off is against energy required to compute a task.

Topic 7

Introduction to TinyOS & NesC

Introduction

- The event-based programming model is the only feasible way to support the concurrency required for sensor node software with simple hardware provided by these nodes.
- In addition, modularity should be supported to easily exchange one state machine against another.
- The operating system TinyOS along with the programming language nesC addresses these challenges as follows.

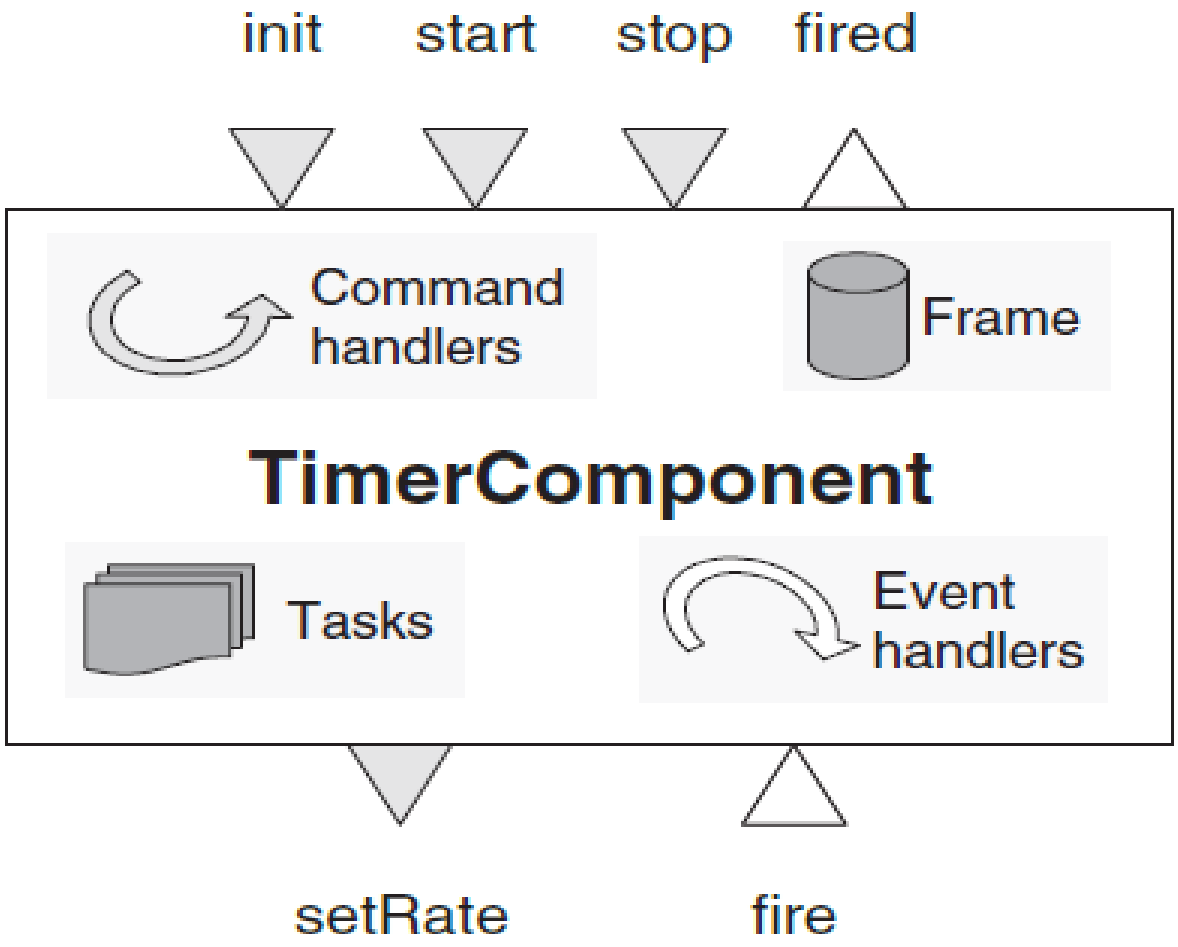
Events & Commands

- TinyOS supports modularity and event-based programming by the concept of components.
- A component contains related functionality, for example, for handling a radio interface.
- Such a component comprises the required state information in a frame, program code for normal tasks and handlers for events & commands.
- Both events and commands are exchanged between different components.
- Components are arranged from low-level components close to the hardware to high-level components making up the actual application.
- Events originate in the hardware and pass upward from low-level to high-level components whereas commands are passed from high-level to low-level components.

Event Handlers

- Figure 2.12 shows a timer component. It understands three commands (“init”, “start”, and “stop”) and can handle one event (“fire”) from another component.
- It issues “setRate” commands to this component and can emit a “fired” event.
- In event-based paradigm, both command and event handlers must run to conclusion and supposed to perform very simple triggering duties.
- Commands must not block or wait for an indeterminate amount of time.
- Similarly, an event handler only leaves information in its component’s frame and arranges for a task to be executed later.
- The actual computational work is done in the tasks.
- In TinyOS, they have to run to completion, but can be interrupted by handlers.

Fig 12 / Example Timer Component



FIFO Scheduler

- There are two advantages - there is no need for stack management and tasks are atomic with respect to each other.
- The arbitration between tasks can be triggered by several events and are ready to execute.
- This is done by a simple First In First Out (FIFO) scheduler, which shuts the node down when there is no task executing or waiting.

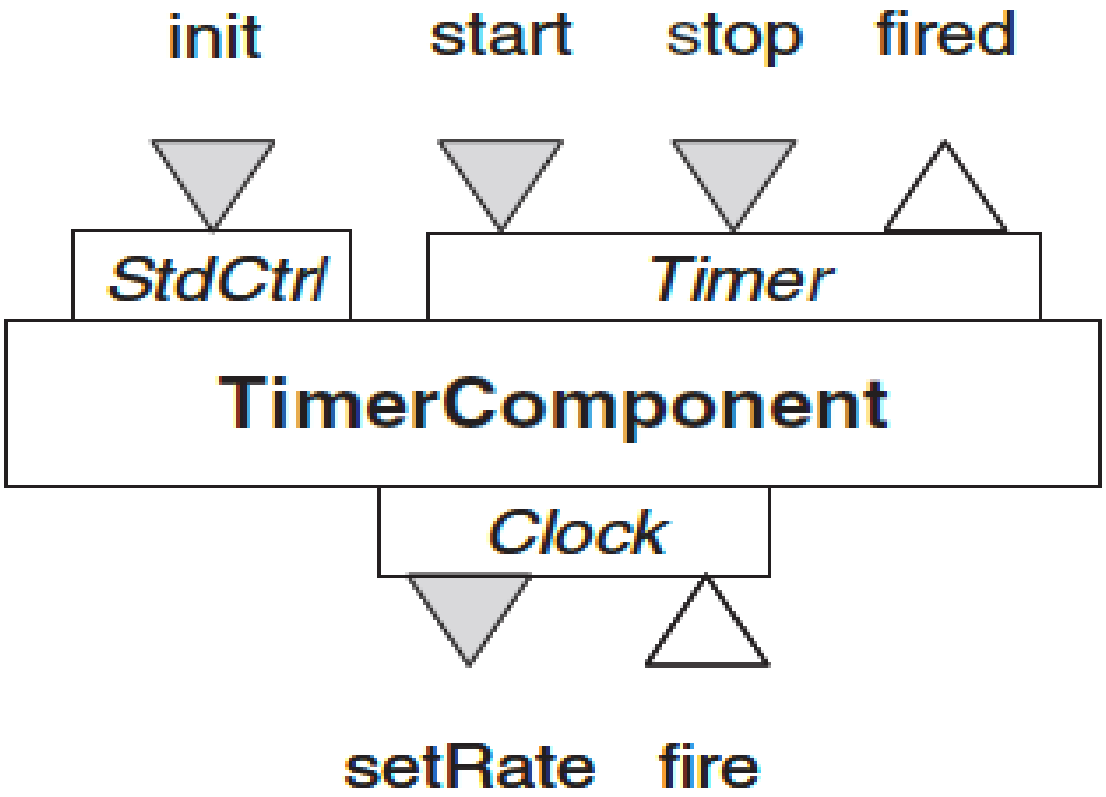
Split Phase Programming

- The first phase is the sending of the command and the second is explicit information about the outcome of the operation, delivered by a separate event.
- This split-phase programming approach requires for each command a matching event that enables concurrency.
- When using split-phase programming, a large number of commands and events are combined in a large program. Hence, an abstraction is required to organize them.
- The set of commands that a component understands and the set of events that a component may emit are its interface to the components of a hierarchically higher layer.
- Therefore, structuring commands and events forms an **interface** between two components.

NesC Language

- The nesC language allows a programmer to define interface types that define commands and events belong together.
- This allows split-phase programming style to put commands and their corresponding completion events into the same interface.
- Components then provide certain interfaces to their users and in turn *use* other interfaces from underlying components.
- Figure 2.13 shows the Timer component reorganized into using a *clock* interface and providing two interfaces StdCtrl and Timer.

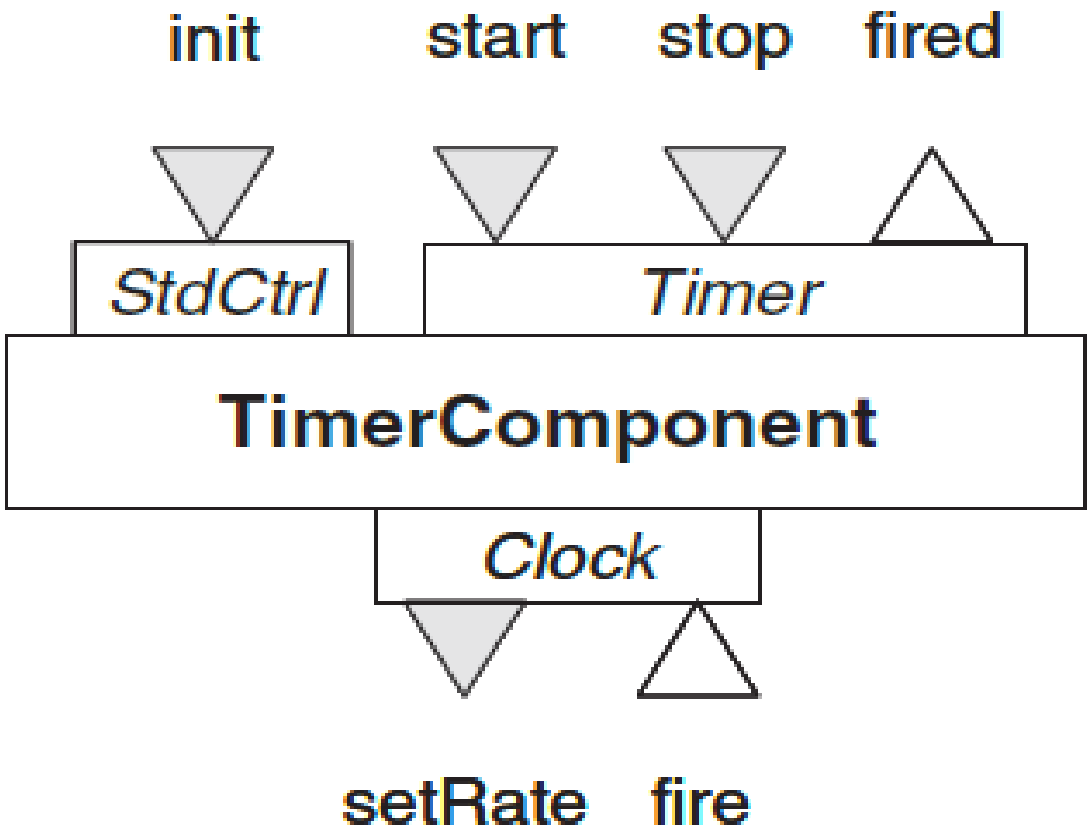
Fig 13 / Timer Component using Interfaces



Timer Component

- The TimerComponent is defined as a primitive component containing handlers and tasks.
- Such primitive components or modules can be combined into larger *configurations* by simply "wiring" appropriate interfaces together.
- For this wiring only components that have the correct interface types can be plugged together.
- Figure 2.14 shows how the TimerComponent and an additional component HW Clock can be wired together to form a new component Complete Timer.

Fig 14 / Larger Configuration



Model Question Bank

PART A

1. Differentiate between a source and sink
2. Mention the three options for a sink.
3. What is multiple sink?
4. What is multi-hopping?
5. Give the types of mobility in WSN.
6. What is event mobility?
7. What is sink mobility?
8. Mention the techniques used for in-network processing.
9. What is aggregation?
10. What is the use of mobile code?

11. What is meant by data centric networking?
12. Mention any ‘4’ crucial points influencing design of physical layer in WSN.
13. Give the factors to be balanced for the choice of modulation scheme.
14. Define figure of merit.
15. What is dynamic modulation scaling?
16. What are the various aspects of Energy efficiency?
17. What is scalability?
18. What are gateway concepts?
19. What is called tunneling?
20. What is concurrent programming?
21. What is TinyOS?
22. What is the use of NesC language?

PART B

1. Explain the various scenarios of Sensor Networks.
2. Discuss in detail, the design principles of WSN.
3. Describe about optimization goals of a WSN and figures of merit in detail.
4. Write a short note on Gateway Concepts.
5. Discuss in detail the characteristics and structure of Transceivers.
6. Write a short note on TinyOS and NesC.

Wireless Sensor Networks

Unit 3 / Networking Sensors

Prepared

By

Dr.S.Omkumar

Syllabus / Unit 3

- NETWORKING SENSORS:
- MAC Protocols for Wireless Sensor Networks, Low Duty Cycle Protocols And Wakeup Concepts – SMAC - B-MAC Protocol, IEEE 802.15.4 standard and ZigBee, the Mediation Device Protocol, Wakeup Radio Concepts, Address and Name Management, Assignment of MAC Addresses, Routing Protocols Energy-Efficient Routing, Geographic Routing

Topic 1

Fundamentals of MAC Protocols

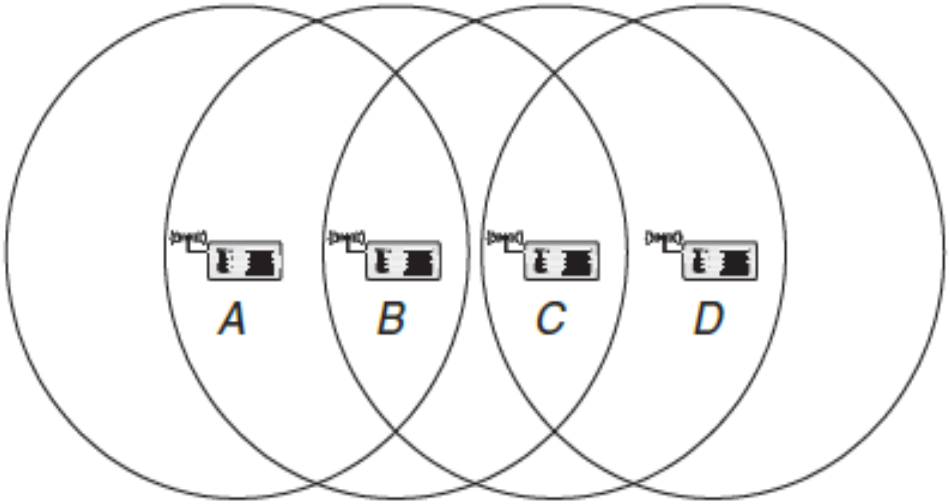
Introduction to MAC Protocols

- The MAC protocol determines the points in time to transmit a data, control or manage packet to another node (unicast) or to a set of nodes (multicast, broadcast).
- Two important responsibilities of the DLL are error control and flow control.
- Error control is used to ensure correctness of transmission and take appropriate actions in case of transmission errors and flow control regulates the rate of transmission.

- The important performance requirements for MAC protocols are throughput efficiency, stability, fairness, low access, low transmission delay and low overhead.
- The overhead can result from per-packet overhead collisions, or from exchange of extra control packets.
- Collisions can happen if MAC protocol allows two or more nodes to send packets at the same time.
- Collisions can result in the inability of the receiver to decode a packet correctly, causing the upper layers to perform a retransmission.

Hidden Terminal Scenario

- If two nodes are out of reach, they cannot hear each other. This gives rise to the hidden-terminal problems.
- The hidden-terminal problem occurs for Carrier Sense Multiple Access (CSMA) protocols, where a node senses the medium before starting to transmit a packet.
- If the medium is found to be busy, the node defers its packet to avoid a collision and a subsequent retransmission.



- Consider the example shown in Figure 3.1.
  - Three nodes *A*, *B*, *C* arranged such that *A* and *B* are in mutual range, *B* and *C* are in mutual range, but *A* and *C* cannot hear each other.
  - *Now A* starts to transmit a packet to *B* and sometime later node *C* also decides to start a packet transmission.
  - A carrier-sensing operation by *C* shows an idle medium since *C* cannot hear *A*'s signals.
  - When *C* starts its packet, the signals collide at *B* and both packets are useless.
  - Using simple CSMA in a hidden-terminal scenario thus leads to needless collisions.

Exposed Terminal Scenario

- *B* transmits a packet to *A* and some moment later, *C* wants to transmit a packet to *D*.
- This will be possible since both *A* and *D* will receive their packets without distortions.
- The carrier-sense operation performed by *C* suppresses *C*'s transmission and bandwidth is wasted.

- Using simple CSMA in an exposed terminal scenario thus leads to needless waiting.
- Two solutions to the hidden-terminal and exposed-terminal problems are busy-tone solutions and the RTS/CTS handshake used in the IEEE 802.11 WLAN standard.

Classes of MAC Protocols

1. Fixed Assignment Protocols:

- The available resources are divided between the nodes such that resource assignment is long term without the risk of collisions.
- Long term means that the assignment is for durations of minutes, hours, or even longer.
- To account for changes in topology due to nodes dying or new nodes being deployed, signaling mechanisms are needed in fixed assignment protocols to rectify the assignment of resources to nodes.
- Typical protocols of this class are TDMA, FDMA, CDMA, and SDMA.

2. Demand Assignment Protocols

- The allocation of resources to nodes is made on a short-term basis, typically the duration of a data burst.
- This class of protocols can be further divided into centralized and distributed protocols.
- In central control protocols, the nodes send out requests for bandwidth allocation to a central node that either accepts or rejects the requests.
- In case of successful allocation, a confirmation is transmitted back to the requesting node along with a description of the allocated resource.

3. Random Access Protocols

- Random access protocols incorporate a random element by exploiting random packet arrival times, setting timers to random values and so on.
- Typical random access protocols are pure ALOHA or slotted ALOHA protocol, developed at the University of Hawaii.
- In pure ALOHA protocol, a node willing for transmission transmit a new packet it immediately.
- There is no coordination with other nodes and the protocol thus accepts the risk of collisions at the receiver.
- To detect this, the receiver is required to send an immediate acknowledgment for a properly received packet.
- If no acknowledgement, the transmitter backs off for a random time and starts the next trial.

Topic 2

MAC Protocols for WSN

Introduction

- The specific requirements and design considerations for MAC protocols in wireless sensor networks are explained below.
  1. Balance of Requirements
  2. Energy Problems on MAC Layer
  - Collisions
  - Overhearing
  - Protocol Overhead
  - Idle Listening

Balance of Requirements

- The typical performance figures of WSN are –

1. Energy Efficiency
   - New parameter
2. Fairness
   - Not important because individual nodes do not compete for bandwidth
3. Transmission delay
   - Traded against energy conservation
4. Scalability & Robustness
   - Important against changes in network topology

Energy Problems on MAC Layer

- A node transceiver consumes a significant share of energy.
- Moreover a transceiver can be in one of the four main states - transmitting, receiving, idling, or sleeping.
- The important features are –
  - Transmitting & receive costs similar
  - Idling cheaper but as expensive as receiving
  - Sleeping costs almost nothing but results in a “deaf” node.
- The above features are applied to the operations of a MAC protocol and the following energy problems and design goals are derived.

1. Collisions

- Collisions incur useless receive costs at the destination node, useless transmit costs at the source node, and expend further energy upon packet retransmission. Hence, collisions should be avoided, either by design or by appropriate collision avoidance or hidden-terminal procedures in CSMA protocols.

2. Overhearing

- Unicast frames have one source and one destination node. But all the source's neighbors in receive state can hear a packet even though not destined to them. Hence these nodes overhear the packet. For higher node densities overhearing avoidance can save significant amounts of energy.

3. Protocol overhead

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

4. Idle listening

- A node being in idle state is ready to receive a packet but not currently receiving anything. This readiness is costly and useless in case of low network loads. Switching off the transceiver is a solution but mode changes also cost energy, and their frequency should be kept at “reasonable” levels. In case of TDMA-based protocols, a node can exchange data *only* during assigned time slot and switch off its transceiver in all other time slots.

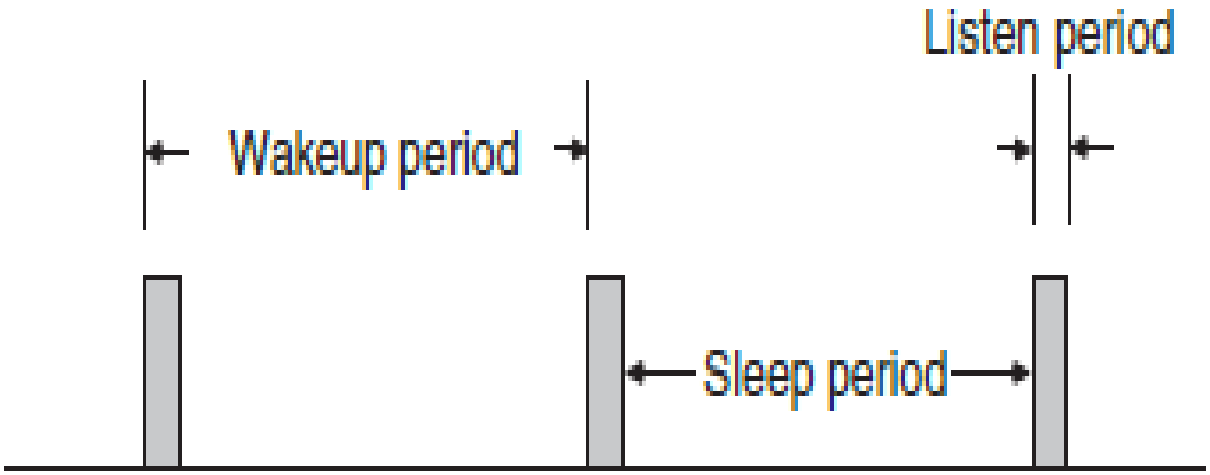
- In order to reduce the energy consumption with MAC protocols in WSN –
  - Expensive operations like complex scheduling algorithms should be avoided.
  - The desire to use cheap node hardware includes components like oscillators and clocks.
  - Frequent resynchronization of neighboring nodes which can consume significant energy.

Topic 3

Low Duty Cycle Protocols & Wakeup Concepts

Introduction

- **Low duty cycle protocols** avoid spending much time in the idle state and reduce the communication activities of a sensor node to a minimum.
- In an ideal case, the sleep state is left only when a node is about to transmit or receive packets. A concept for achieving this is the wakeup radio.



Periodic Wakeup Scheme – Method 1

- First method is the **cycled receiver** approach illustrated in Figure 3.2. In this approach, nodes spend most of their time in the sleep mode and wake up periodically to *receive* packets from other nodes.
- A node '*A*' listens onto the channel during its **listen period** and goes back into sleep mode when no other node takes the opportunity to transmit a packet to *A*.
- A potential transmitter *B* must acquire knowledge about *A*'s listen periods to send its packet at the right time – this task corresponds to a *rendezvous*.
- This can be accomplished by letting node *A* transmit a short beacon at the beginning of listen period to indicate its willingness to receive packets.

Periodic Wakeup Scheme - Method 2

- Second method is to let node *B* send frequent request packets until one of them hits *A*'s listen period and is answered by *A*.
- However, in both methods, node *A* only *receives* packets during its listen period.
- If node *A* wants to transmit packets, it must acquire the target's listen period.
- The whole cycle consisting of sleep period and listen period is also called a **wakeup period**.
- The ratio of the listen period length to the wakeup period length is also called the node's **duty cycle**.

Important Observations from 2 Methods

- By choosing a small duty cycle, the transceiver is in sleep mode most of the time, avoiding idle listening and conserving energy.
- By choosing a small duty cycle, the traffic directed from neighboring nodes to a given node concentrates on a small listen period and in heavy load situations significant competition can occur.
- Choosing a long sleep period leads to significant **per-hop latency**.
- Sleep phases should not be too short, otherwise the start-up costs outweigh the benefits.

Comparison with other Protocols

- In other protocols like S-MAC, there is also a periodic wakeup but nodes can both transmit and receive during their wakeup phases.
- When nodes have their wakeup phases at the same time, there is no necessity for a node wanting to transmit a packet to be awake outside these phases to rendezvous its receiver.

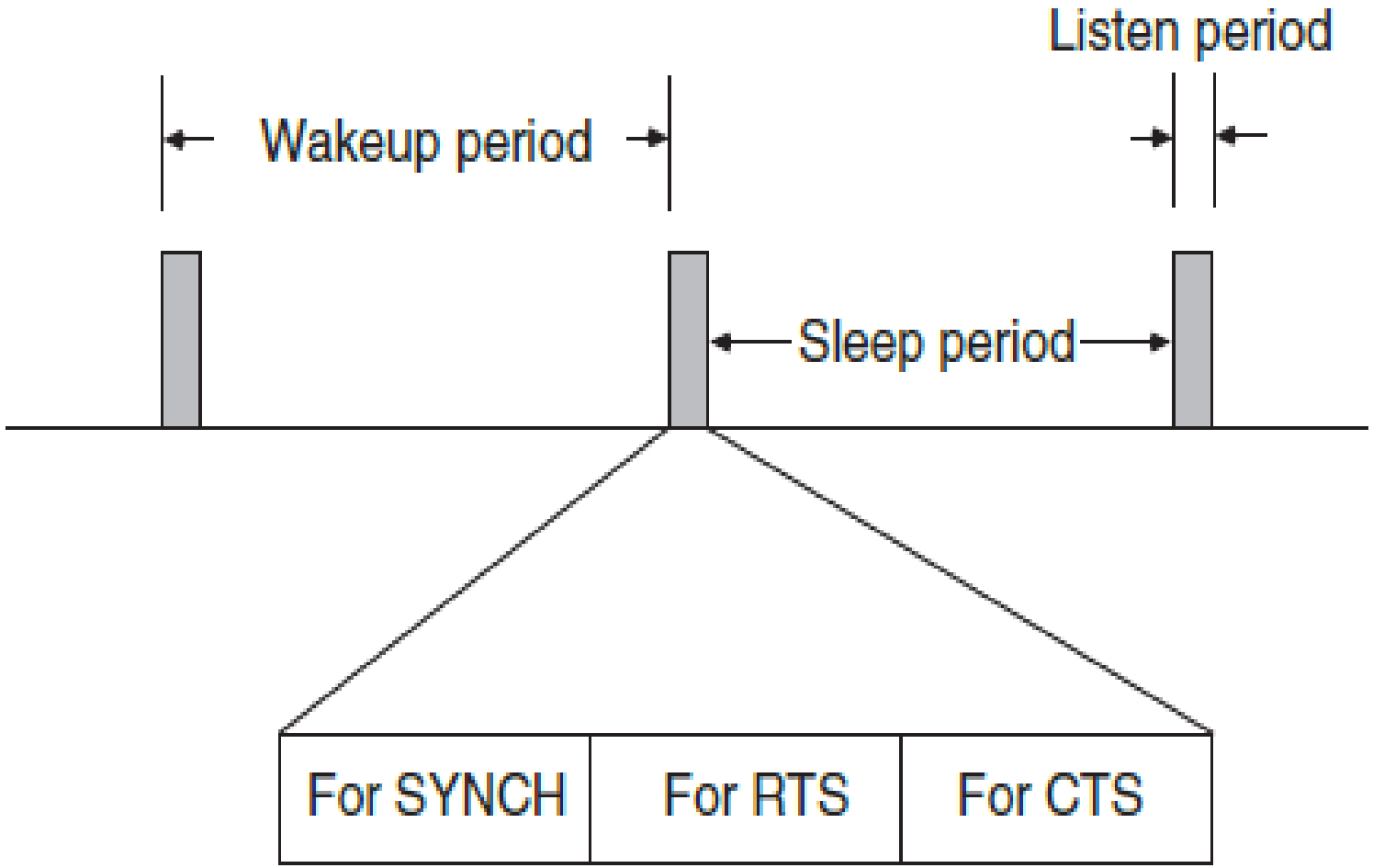
Topic 4

S-MAC Protocol

Introduction

- The S-MAC (Sensor-MAC) protocol provides mechanisms to circumvent idle listening, collisions, and overhearing.
- S-MAC adopts a periodic wakeup scheme - each node alternates between a fixed-length listen period and a fixed-length sleep period as shown in Figure 3.3.
- However, the listen period of S-MAC can be used to receive *and transmit* packets.
- S-MAC attempts to coordinate the schedules of neighboring nodes such that their listen periods start at the same time. A node *x*'s listen period is subdivided into three different phases:
  - Wakeup period
  - Listen period
  - Sleep period

Fig 3 / Principle of SMAC



First Phase - Synch Phase

- During this phase, node *x* accepts SYNCH packets from its neighbors.
- In these packets, the neighbors describe their own schedule and *x* stores their schedule in a schedule table.
- Node *x*'s SYNCH phase is subdivided into time slots according to a CSMA scheme.
- That is each neighbor *y* wishing to transmit a SYNCH packet picks one of the time slots randomly and starts to transmit if no signal was received in any of the previous slots.
- In the other case, *y* goes back into sleep mode and waits for *x*'s next wakeup.
- In the other direction, since *x* knows a neighbor *y*'s schedule, *x* can wake at appropriate times and send its own SYNCH packet to *y*.

Second Phase - RTS Phase

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

Third Phase – CTS Phase

- In the third phase (**CTS phase**), node *x* transmits a CTS packet if an RTS packet was received in the previous phase. After this, the packet exchange continues, extending into *x*'s nominal sleep time.
- In general, when competing for the medium, the nodes use the RTS/CTS handshake whereby a node maintains a NAV variable.
- The NAV mechanism can be used to switch off the node during ongoing transmissions to avoid overhearing.

Virtual Cluster Approach

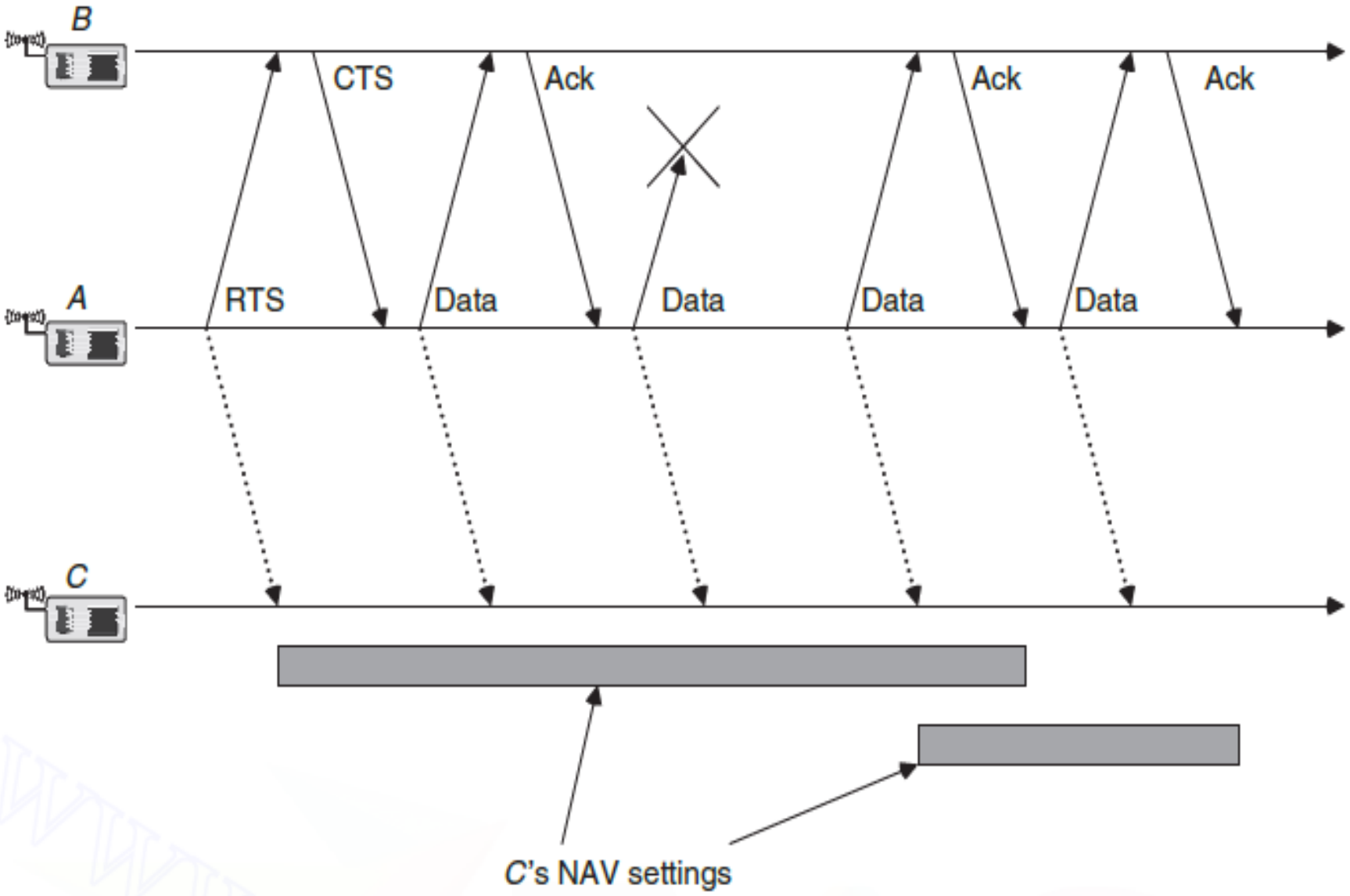
- The S-MAC protocol allows neighboring nodes to agree on the same schedule and to create virtual clusters.
- The clustering structure refers to the exchange of schedules and the transfer of data packets is not influenced by virtual clustering.
- The S-MAC protocol proceeds as follows to form the virtual clusters -

1. A node *x* listens for a time at least the synchronization period.
2. If *x* receives any SYNCH packet from a neighbor, it adopts the announced schedule and broadcasts in one of the neighbors' next listen periods.
3. In the other case, node *x* picks a schedule and broadcasts it.
4. If *x* receives another node's schedule during the broadcast packet's contention period, it drops its own schedule and follows the other one.
5. If node *x* already knows about the existence of neighbors who adopted its own schedule, it keeps its schedule.

Message Passing Approach

- S-MAC also adopts a message-passing approach as shown in Figure 3.4. In wireless media, it is advisable to break a longer packet into several shorter ones. S-MAC includes a fragmentation scheme working as follows.
  - A series of fragments is transmitted with only one RTS/CTS exchange between the transmitting node *A* and receiving node *B*.
  - After each fragment, *B* has to answer with an acknowledgment packet. All the packets have a duration field and a neighboring node *C* is required to set its NAV field accordingly.

Fig 4 / SMAC Fragmentation & NAV Setting



- In S-MAC, the duration field of all packets carries the remaining length of the whole transaction, including all fragments and their acknowledgments.
- Therefore, the whole message shall be passed at once.
- If one fragment needs to be retransmitted, the remaining duration is incremented by the length of a data plus acknowledgement packet.

Topic 5

The Mediation Device Protocol

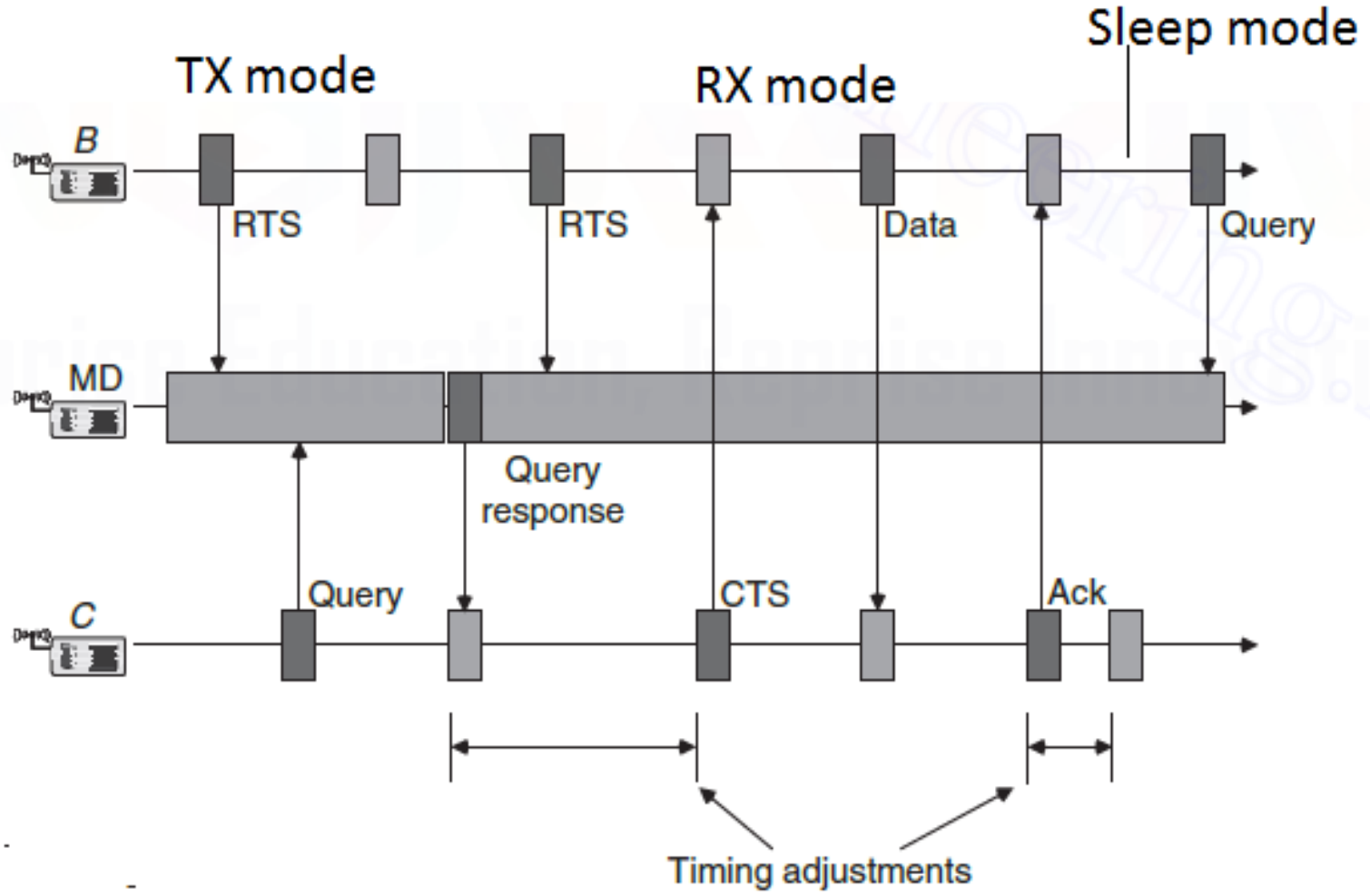
Introduction

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

Mediation Device

- When a node wants to transmit a packet to a neighbor, it has to synchronize with it.
- The dynamic synchronization approach achieves this synchronization without requiring the transmitter to be awake permanently to detect the destinations query beacon.
- To achieve this, a mediation device (MD) is used.
- The mediation device is not energy constrained and can be active all the time as shown in Figure 3.5.
- Because of its full duty cycle, the mediation device can receive the query beacons from all nodes and learn their wakeup periods.

The Mediation Device Protocol



Dynamic Synchronization Approach

- Suppose node *A* wants to transmit a packet to node *B*. The dynamic synchronization approach is given below-
  - Node *A* announces this to the mediation device by sending **request to send** (RTS) packets, which the MD captures.
  - There is a short answer window after the RTS packets, where *A* listens for answers.
  - After MD has received *A*'s RTS packet, it waits for *B*'s next query beacon.
  - The MD answers with a **query response** packet, indicating *A*'s address and a timing offset, which lets *B* know when to send the answering **clear to send** (CTS) to *A*.
  - Therefore, *B* has learned *A*'s period.

- After *A* has received the CTS packet, it can send its data packet and wait for *B*'s acknowledgment.
- After transaction has finished, *A* restores its periodic wakeup cycle and starts to emit query beacons again.
- Node *B* also restores its own periodic cycle and thus *decouples* from *A*'s period.

Advantages

- It does not require any time synchronization between nodes.
- The protocol is asymmetric that most of the energy burden is shifted to the mediation device.
- The other nodes can be in the sleep state most of the time and spend energy only for the periodic beacons.

Disadvantages

- The nodes transmit their query beacons without checking for ongoing transmissions.
- The beacons of different nodes may collide when nodes have the same period and their wakeup periods overlap.

Topic 6

Wakeup Radio Concepts

Introduction

- The ideal situation will be if a node is always in the receiving state when a packet is transmitted to it, in the transmitting state when it transmits a packet, and in the sleep state at all other times, the idle state should be avoided.
- The requirement can be achieved by the concept of **wakeup radio** by a simple, “powerless” receiver that can trigger a main receiver if necessary.

Proposed Wakeup Protocol

- The proposed wakeup MAC protocol comprises several parallel data channels separated either in frequency (FDMA) or in CDMA schemes.
- A node wishing to transmit a data packet randomly picks one of the channels and performs a carrier sensing operation.
- If the channel is busy, the node picks another random channel and repeats the carrier-sensing operation.
- After a certain number of unsuccessful trials, the node waits for a random time and starts again.
- If the channel is idle, the node sends a wakeup signal to the intended receiver, indicating both the receiver identification and the channel to use.

- The receiver wakes up its data transceiver, tunes to the indicated channel, and the data packet transmission can proceed.
- Afterward, the receiver can switch its data transceiver back into sleep mode.

Advantages

- Only low-power wakeup transceiver has to be switched on all the time while more energy consuming data transceiver can be non-sleeping particular node involved in data transmissions.
- This scheme is naturally traffic adaptive, the MAC becomes more and more active as the traffic load increases.

Disadvantages

- There is no real hardware for such low power wakeup transceiver.
- The range of the wakeup radio and the data radio should be the same.
- If the range of the wakeup radio is smaller than that of data radio, all neighbor nodes cannot be woken up.
- If the range of the wakeup radio is larger, there can be a problem with local addressing schemes
- This scheme depends on the wakeup channel's ability to transport useful information like node addresses and channel identifications

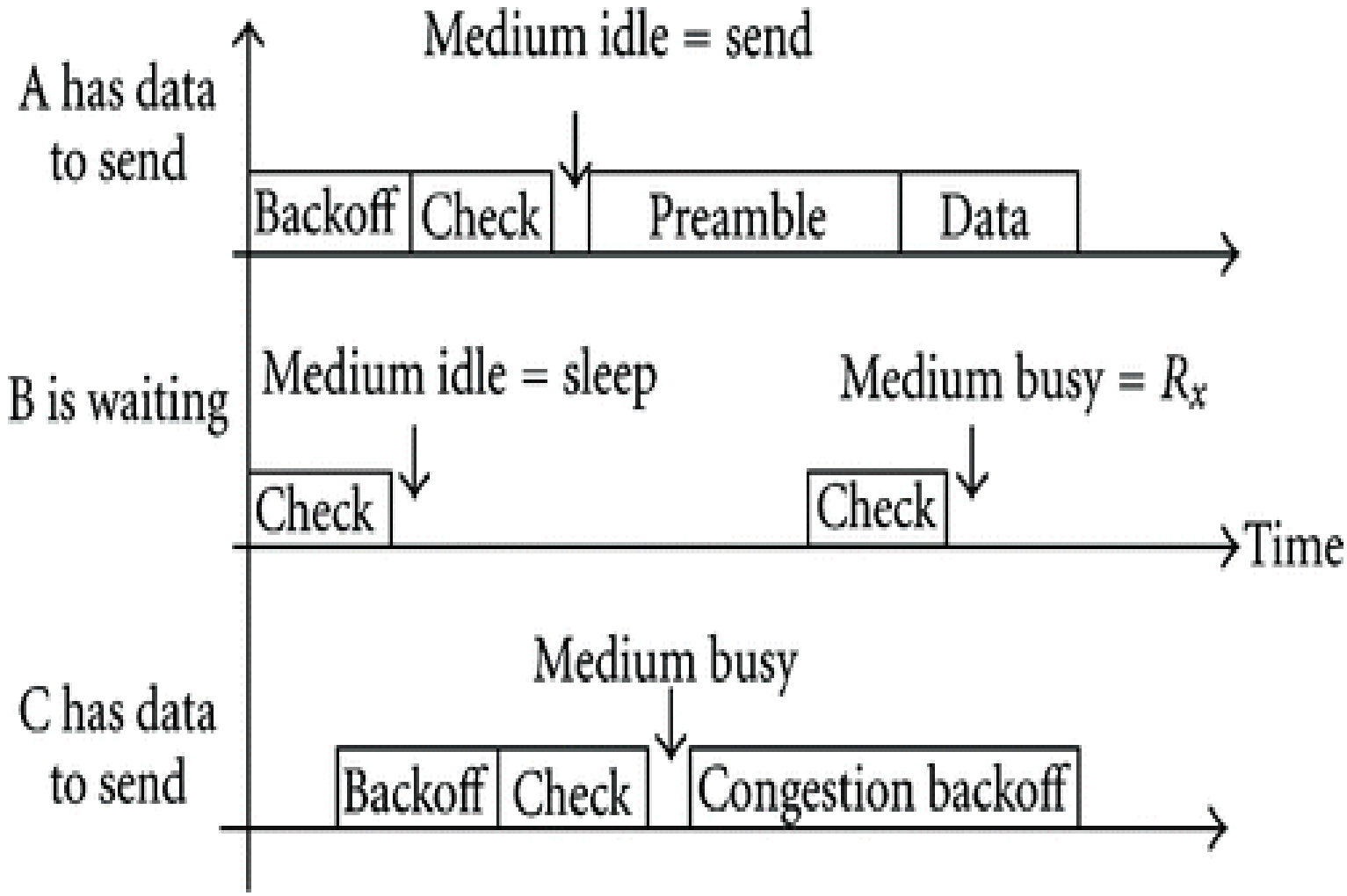
Topic 7

B-MAC Protocol
(Berkeley MAC Protocol)

Introduction

- B-MAC is a widely used WSN MAC protocol. It is a part of TinyOS.
- It employs low-power listening to minimize power consumption due to idle listening.
- Nodes have a sleep period, after which they wake up and sense the medium for preambles.
- 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 the sleep period for all nodes to detect it. After the preamble, it sends the data packet.
- There are optional acknowledgments also.
- After the data packet exchange, the nodes go back to sleep.
- The preamble doesn't contain addressing information. Figure 3.6 shows the example transmission using B-MAC.

Fig 6 / Example for BMAC Communication



Preamble Sampling Scheme

- The B-MAC preamble sampling scheme checks the channel for adjusting the time interval equal to frame preamble size.
- As an example, if the medium is checked every 100 ms, the preamble of the packet must be available for 100 ms at the minimum, in order for the receiver to detect the packet.
- Upper layers may change the preamble duration, according to the application requirements.

Advantages

- It does not use RTS, CTS, ACK, or any other control frame by default, but they can be added.
- It is one of the few specialized MAC protocols whose implementation was tested in hardware.
- No synchronization is required and the protocol performance can be tuned by higher layers to meet the needs of various applications.

Disadvantages

- The preamble creates large overhead.
- One example presents 271 bytes of preamble to send 36 bytes of data.

Topic 8

IEEE 802.15.4. Standard

Introduction

- The standard covers the physical layer and the MAC layer of a low-rate Wireless Personal Area Network (WPAN).
- The targeted applications for IEEE 802.15.4 are in the area of wireless sensor networks, home automation, home networking, connect devices to PC etc.
- Most of these applications require only low-to-medium bitrates, moderate delays and minimum energy consumption.
- The physical layer offers bitrates of 20 kbps (a single channel in frequency range 868–868.6 MHz), 40 kbps (ten channels in range between 905 and 928 MHz) and 250 kbps (16 channels in range between 2.4 and 2.485 GHz).
- There are a total of 27 channels available, but the MAC protocol uses only one of these channels at a time.
- The MAC protocol combines both schedule-based as well as contention-based schemes.

Network Architecture

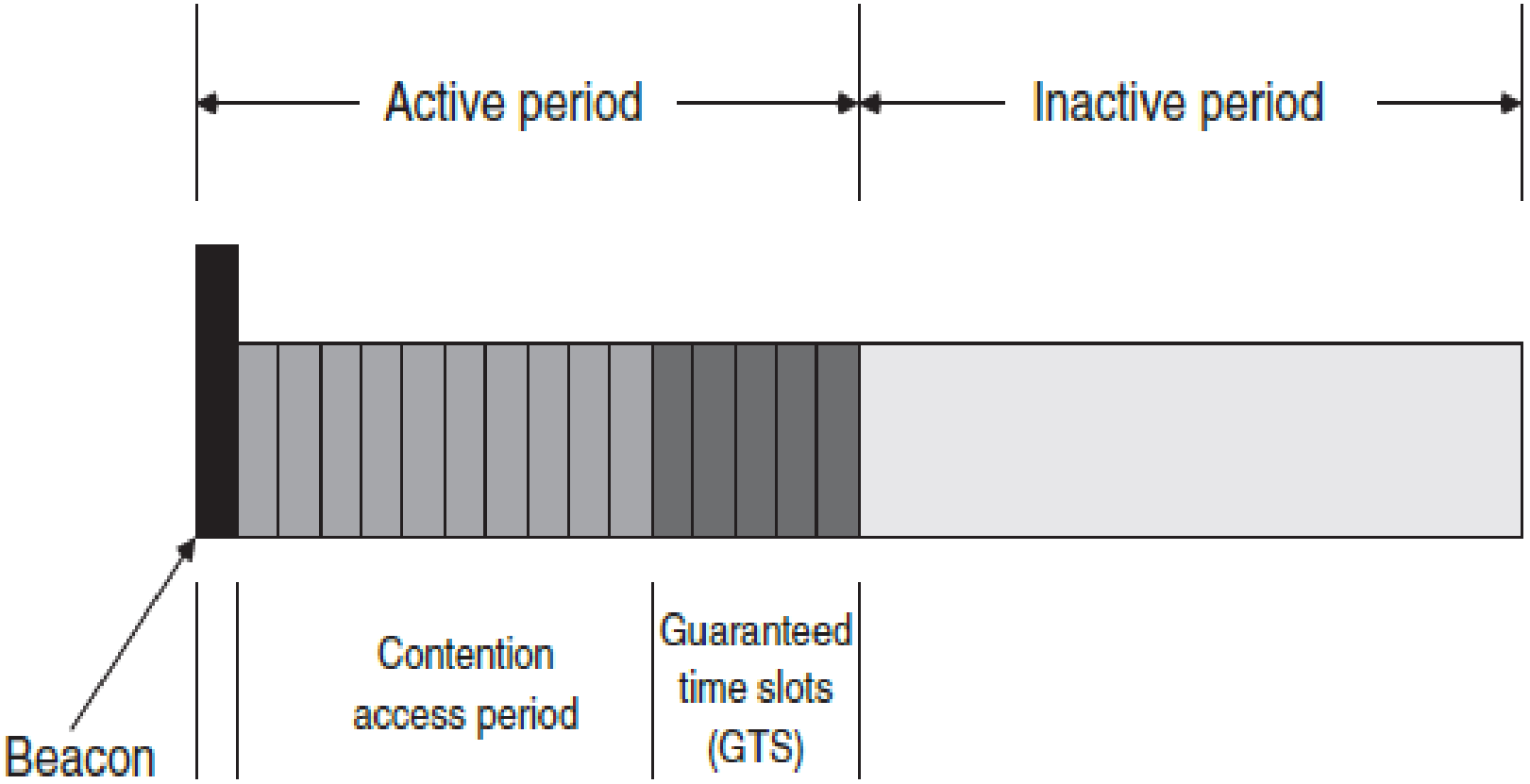
- The standard distinguishes on the MAC layer two types of nodes:
  - A Full Function Device (FFD) can operate in three different roles - a PAN coordinator, a simple coordinator and a device.
  - A Reduced Function Device (RFD) can operate only as a device.
- A device must be associated to a coordinator node and communicates only with this, forming a star network.
- Coordinators can operate in a peer-to-peer fashion and multiple coordinators can form a Personal Area Network (PAN).
- The PAN is identified by a 16-bit PAN Identifier and one of its coordinators is designated as a PAN coordinator.

- A coordinator handles the following tasks:
  - It manages a list of associated devices.
  - It allocates short addresses to its devices
  - In the beaconed mode, it transmits regularly frame beacon packets announcing the PAN identifier, a list of outstanding frames and other parameters.
  - It exchanges data packets with devices and with peer coordinators.

Super Frame Structure

- The coordinator of a star network operating in the beaconed mode organizes data transmission with the help of a super-frame structure displayed in Figure 3.7.
- All super-frames have the same length. The coordinator starts each super-frame by sending a frame beacon packet. The various components of the following super-frame are as follows-
  - The super-frame is subdivided into an active period and an inactive period. During the inactive period, all nodes including the coordinator can switch off their transceivers and go into sleep state.
  - The active period is subdivided into 16 time slots.

Fig 7 / Super Frame Structure of IEEE 802.15.4



- The first time slot is occupied by the beacon frame and the remaining time slots are partitioned into a Contention Access Period (CAP) followed by a number of contiguous Guaranteed Time Slots (GTSs).
- The length of the active period, inactive period, length of a single time slot and the usage of GTS slots are configurable.
- The coordinator is active during the entire active period.
- The associated devices are active in the GTS phase only in their allocated time slots.

GTS Management

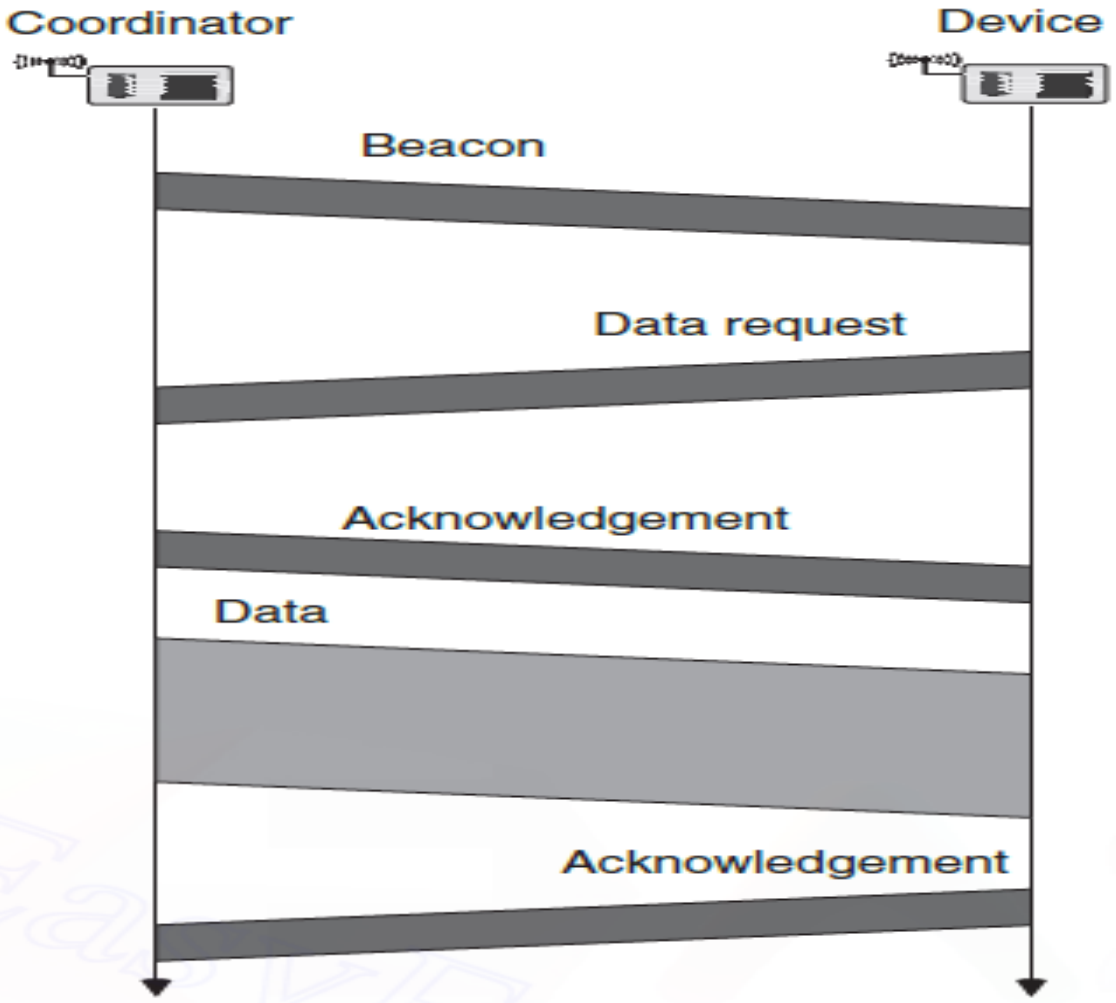
- The coordinator allocates GTS to devices only when the latter send appropriate request packets during the CAP.
- One flag in the request indicates whether requested time slot is a transmit slot or a receive slot.
- In a transmit slot, the device transmits packets to the coordinator and in a receive slot the data flows in the reverse direction.
- The coordinator answers the request packet in two steps:
- An immediate acknowledgment packet with no information about success or failure of the request.
- If resources available, the coordinator inserts a GTS descriptor into one of the next beacon frames.

Data Transfer Procedures

- Case 1: Device transmits data packet to coordinator
  - If the device has an allocated transmit GTS, it wakes up and sends its packet immediately without running any carrier-sense operations.
  - However, the device can do the full transaction only if allocated time slots are available.
  - When the device does not have any allocated slots, it sends its data packet during the CAP using a slotted CSMA protocol.
  - The coordinator sends an immediate acknowledgment for the data packet

- Case 2: Coordinator sends data packer to Device.
  - If the device has allocated receive GTS and when the packet/acknowledgment fits into these, the coordinator simply transmits the packet in the allocated time slot.
  - The device has to acknowledge the data packet.
  - When the coordinator is not able to use a receive GTS, the handshake between device and coordinator will happen as shown in Figure 3.8.

Fig 8 / Handshaking Operation



Non-Beaconed Mode

- The coordinator does *not* send beacon frames nor any GTS mechanism. No time synchronization exists.
- All packets from devices are transmitted without using time slots because of lack of time synchronization.
- Coordinators must be switched on constantly but devices can follow their own sleep schedule.

Topic 9

Address and Name Management

Introduction

- Naming and addressing are two fundamental issues in networking.
- The **names** are used to denote things (for example, nodes, data) whereas **addresses** supply the information needed to *find* these things, for example, with routing in a multi-hop network.
- Sometimes addresses are used to denote things too – an IP address contains information to both find a node and to identify a node more precisely a network interface within a node.

Use of Addresses & Names in Networks

1. Unique node identifier:

- **A** persistent data item unique for every node. An example of a UID might be a combination of a vendor name, a product name etc assigned at the time of manufacturing.

2. MAC address:

- To distinguish between one-hop neighbors of a node. This is important in wireless sensor networks using contention-based MAC protocols

3. Network address:

- **It** is used to find and denote a node over multiple hops and hence network addresses are often connected to routing.

4. Network identifiers:

- To distinguish geographically overlapping wireless networks of the same type and working in the same frequency band.

5. Resource identifiers:

- **It** is represented in user-understandable terms. For example, upon reading the name www.xemacs.org, an experienced user knows that (i) the thing the name refers to is likely a web server and (ii) the user can find information about a great text editor.

Address Management Tasks

1. Address allocation: **A**ssignment of an address to an entity from an address pool.
2. Address de-allocation: **I**f the addresses of the dying nodes were not put back into the address pool for **reuse**, the address pool will be exhausted and no addresses can be allocated to new nodes. Address de-allocation can be either graceful or abrupt.
   - Graceful de-allocation: **A** node sends out control packets to give up its address.
   - Abrupt de-allocation: **T**he node disappears and does not send appropriate control packets, leaving the responsibility to the network.

3. Address representation: A format for representing addresses needs to be negotiated and implemented.
4. Conflict Detection/Resolution: Address conflicts can occur in networks with distributed assignment of on-demand addresses or in case of **mergers** of so-far distinct networks.
5. Binding: If several addressing layers are used, a mapping between the different layers has to be provided. For example, in IP networks, an IP address has to be mapped to a MAC address using the ARP protocol.

Uniqueness of Addresses

- Globally unique: A globally unique address is supposed to occur at most once all over the world. An example is 48-bit IEEE MAC addresses used in Ethernet and Token Ring networks.
- Network wide unique: A network wide unique address is supposed to be unique within a given network, but the same address can be used in different networks.
- Locally unique: A locally unique address might occur several times in the same network, but it should be unique within a suitably defined neighborhood.

Address Allocation & Assignment

- The address assignment can happen **a priori** during the manufacturing process or **on demand**, by using an address assignment protocol.
- Such an on-demand address assignment protocol can be either centralized or distributed.
- In a centralized solution, there is one single authority/node taking care of the address pool, whereas in distributed solutions, there is no such exposed node.
- The distinction between strong and weak Duplicate Address Detection (DAD) are as follows:

1. Strong DAD:

- **I**f address *x* is already assigned to node *A* at time *t*0 and subsequently assigned to node *B* at time *t*1, then this duplicate assignment must be detected latest at time *t*1 + *T* where *T* is some fixed time bound.

2. Weak DAD:

- **D**uplicate addresses are tolerated as long as they do not distort ongoing sessions. For example, if two networks *A* and *B* merge and one address *x* is assigned in both networks, no action should be taken as long as all packets from nodes of the former network *A* destined to *x* reach the node in *A* with address *x* and not the node with the same address in the other network.

Topic 10

Assignment of MAC Addresses

Introduction

- The assignment of globally unique MAC addresses is undesirable in sensor networks with mostly small packets.
- A priori assignment of network wide unique addresses is feasible only if it can be done with reasonable effort.
- But the overhead required to represent addresses is not much large as in globally unique addresses.
- For example, up to 16,384 nodes can be addressed with 14 bits and this number is much friendlier than 48 bits used for globally unique IEEE addresses.

Distributed Assignment of Network Wide Addresses

- A node chooses its address without any prior information to use a uniform distribution on the address range since this has maximum entropy.

1. Random address assignment

- Suppose that we have k nodes and each of these nodes picks uniformly and independently a random address from 0 to 2m − 1. The probability that these nodes choose a conflict-free assignment has to be computed.
  - For k = 1 this probability is one.
  - For k = 2, the second node picks with probability (n−1/ n) an address different from that of first node
  - For k = 3, the third node picks with probability [(n−1).(n−2) / n2] an address different from the first two and so on.

- Hence the probability P (n, k) of the conflict-free assignment is given by –

$$P(n,k) = 1 \cdot \frac{n-1}{n} \cdot \ldots \cdot \frac{n-k+1}{n} = \frac{1}{n^k} \cdot \frac{n!}{(n-k)!} = \frac{k!}{n^k} \cdot \binom{n}{k},$$

- Therefore, this method of random assignment quickly leads to address conflicts.
- To preserve network wide uniqueness, either a conflict-resolution protocol or more clever assignment schemes should be chosen.

2. Techniques to deal with address collisions:

a. Auto-configuration Technique:

- A node starts by randomly selecting a temporary address and a proposed fixed address and sends the address request control packet.
- The temporary address is allocated from a dedicated address pool and the routing protocol will find a path to a node having the same fixed address.
- For such a node, an address reply packet is generated and sent toward the temporary address.
- Upon receiving this reply, the node knows that the selected fixed address is allocated and tries another address.
- If no address reply is received within a certain time, the node repeats the address request packet a number of times to compensate for possibly lost address reply packets.
- If still no address reply is received after all trials, the node accepts the chosen IP address.

b. Initiator Technique:

- The initiator keeps a table of all known address assignments and picks an unused address.
- The initiator then disseminates the proposed new address to all nodes in the network and collects the answers. All nodes put the proposed address into a list of candidate addresses.
- If a node finds the address in the candidate list, it answers with a reject packet, otherwise it answers with an accept packet.
- If all known nodes have answered with an accept packet, the initiator assigns the address to the requester and informs all other nodes in the network that the assignment now is permanent.
- Otherwise, the initiator picks another address and tries again.

Distributed Assignment Of Locally Unique Addresses

- This protocol assigns locally unique MAC addresses to nodes by which a node communicates only with immediate neighbors.
- Hence fewer bits are needed for address representation than for network wide or globally unique addresses.
- By using locally unique addresses, the same address can be used several times in the overall network. This opportunity is taken by not transmitting addresses directly but by encoding them and transmitting the code words.
- The mapping from addresses to codes is called the codebook and must be known a priori to nodes.

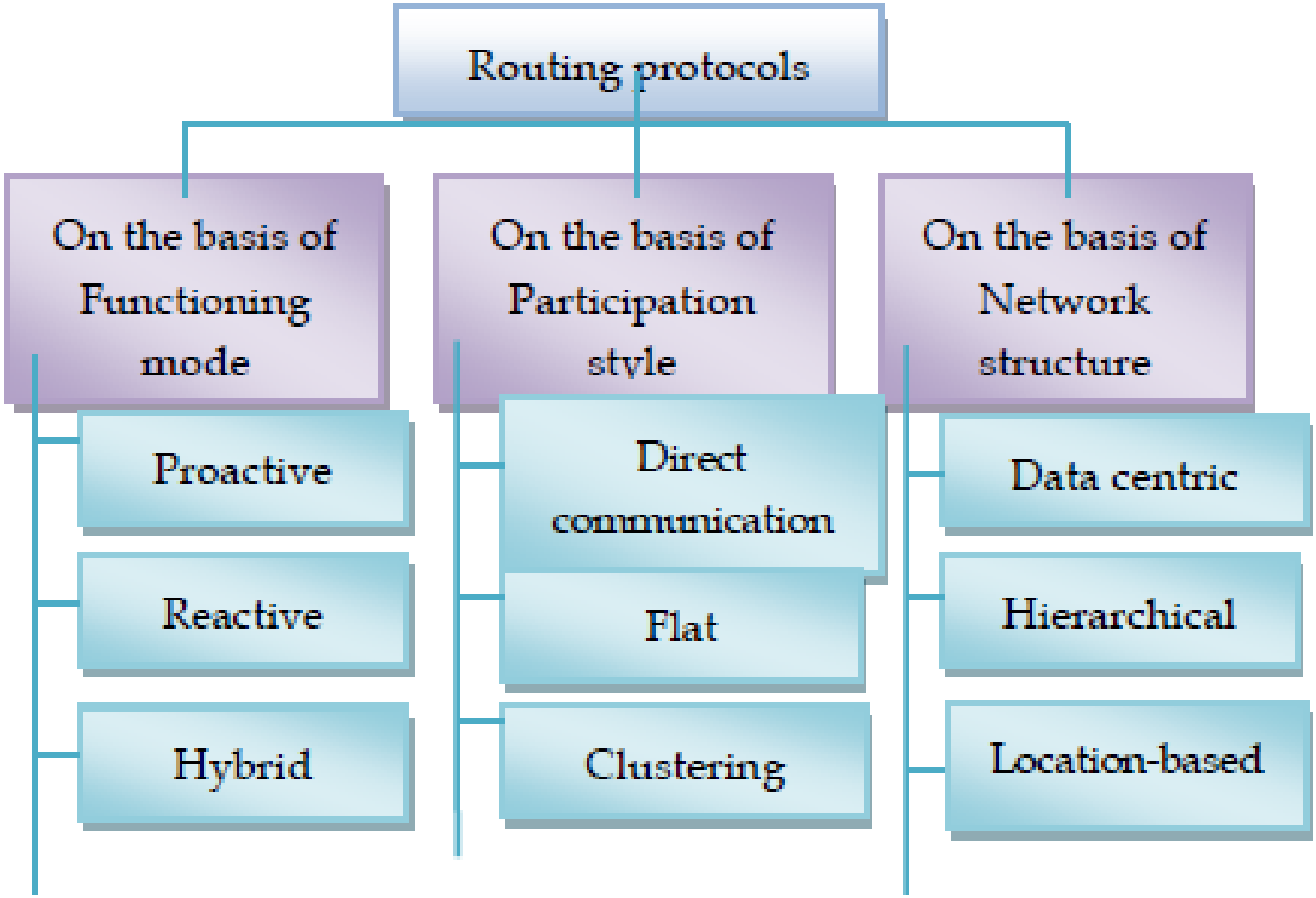
Topic 11

Routing Protocols

Introduction

- Routing strategies are required for transferring data between the sensor nodes and the base station.
- Routing in WSN is different than traditional IP network routing because it exhibits a number of unique characteristics to build a global addressing scheme for a large number of sensor nodes.
- Different routing techniques are proposed for remote sensor network and these conventions can be classified as per different parameters.
- The classification of routing methods is shown in Figure 3.10.

Fig 10 / Classification of Routing Protocols



Functioning Mode Based Routing Protocols

- The function of a wireless sensor network specifies its application. Hence routing protocols can be categorized according to the operation used to satisfy a WSN function as follows:
- Proactive Protocols: These protocols are also called as table-driven protocols. In Proactive, the data is transmitted to a BS through the predefined route. Examples: LEACH, PEGASIS.
- Reactive Protocol: In Reactive Protocol the route is established on demand. The route is established dynamically when needed. Examples: TEEN, AODV, DSR
- Hybrid protocols: All the routes are found initially and then improved at the time of sending data. These protocols possess the concepts of both reactive and proactive. For example APTEEN.

Participation Style Based Routing Protocols

- Some WSNs consist of homogeneous nodes, whereas some consist of heterogeneous nodes and these nodes participate differently in every network according to remaining energy of nodes, cluster head etc. Based on this concept the protocols are classified as:
- Direct Communication protocols: In this type of protocols the information sensed by nodes is sent directly to Base Station (BS). Example: SPIN
- Flat protocols: In this, the nodes search for the valid path and then transmit it to Base station. Example: Rumor routing protocol.
- Clustering Protocols: In this, the area is divided into clusters and Cluster heads are assigned to each cluster. All the nodes in the cluster send data to corresponding cluster heads and then cluster head sends it to Base station. Example: TEEN

Network Based Routing Protocols

- Network-based routing protocols depend on the strategy how the network is prearranged. Such protocols fall under three categories:
- Data Centric protocols: These are query based and they depend on the naming of the desired data. The BS sends queries within a certain region to get information and waits for a reply from the nodes. Nodes in a particular region collect the specific data based upon the queries. Example: SPIN.
- Hierarchical protocols: In this, the nodes with lower energy are used to capture information and nodes with higher energies are used to process, transfer it and it is used to perform energy efficient routing. Example: TEEN, APTEEN.

- Location Based: In these, the location of nodes must be known to find an optimal path using flooding. To get the information about location of a particular node GPS is used. Example: GEAR.

Topic 12

Energy Efficient Routing

Introduction

- Energy efficiency of a network is a significant concern in wireless sensor network.
- These days networks are becoming large, information gathered is becoming 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.
- The following are some of energy efficient routing protocols:

1. LEACH – Low Energy Adaptive Clustering Hierarchy

- In this type of hierarchical protocol, most of the nodes communicate to cluster heads. It consists of two phases:
- Setup Phase: In this phase, the clusters are ordered and then Cluster Head (CH) has been selected. The task of CH is to cumulate, wrapping, and forward the information to the base station (Sink).
- Study State Phase: In this phase, the data is communicated to the base station (Sink). To minimize the overhead, the duration of this phase has been increased. Each node in the network, contacts with the cluster head, and transfer the data to it. Then CH will develop the schedule to transfer the data of each node to base station.

2. PEGASIS [Power-Efficient Gathering in Sensor Information Systems]

- It is a “chain-basis protocol” and an upgrading of the “LEACH”.
- In “PEGASIS” every node transfers only with a close neighbor to direct and obtain information. It turns communicating to the BS, thus decreasing the quantity of energy consumed per round.
- A chain should be developed, which can be completed by the sensor nodes along with using an algorithm.
- On the other hand, the BS can compute this chain and transmit it to all the sensor nodes.
- To develop the chain, all nodes have universal information of the system and a greedy algorithm is engaged

3. Threshold sensitive Energy Efficient sensor Network protocol

- The TEEN is a hierarchical protocol designed for the conditions like sudden changes in the sensed attributes such as temperature.
- The reduction of the number of transmissions is the purpose of a hard threshold done by allowing the nodes to transmit only when the sensed attribute is in the range of interest.
- TEEN is well applicable for time important problems and quite efficient in terms of saving energy and response time.
- It also allows the user to manage the power utilization and accurateness to suit the application.

4. Adaptive Threshold sensitive Energy Efficient Sensor Network

- The “APTEEN” is an expansion of “TEEN” and goals at both taking episodic data gatherings and replying to time-critical events.
- As soon as the BS formulates the clusters, the CH transmits the features, the values of threshold and schedule of transmission to all nodes.
- After that, CH performs information accumulation in order to preserve power.
- The main advantage of “APTEEN” in contrast to “TEEN”, is that nodes utilize a smaller amount of power.
- The primary disadvantages of APTEEN are the complication which results in lengthier deferment times.

5. Directed Diffusion

- Directed diffusion is data-centric routing protocol for collecting and publishing the information in WSNs.
- It has been developed to address the requirement of data flowing from the sink toward the sensors.
- Its main objective is extending the network life time by realizing essential energy saving.
- In order to achieve this, it has to keep the interactions among the nodes within a limited environment by message exchange.
- A localized interaction that provides multipath delivery is a unique feature of this protocol.

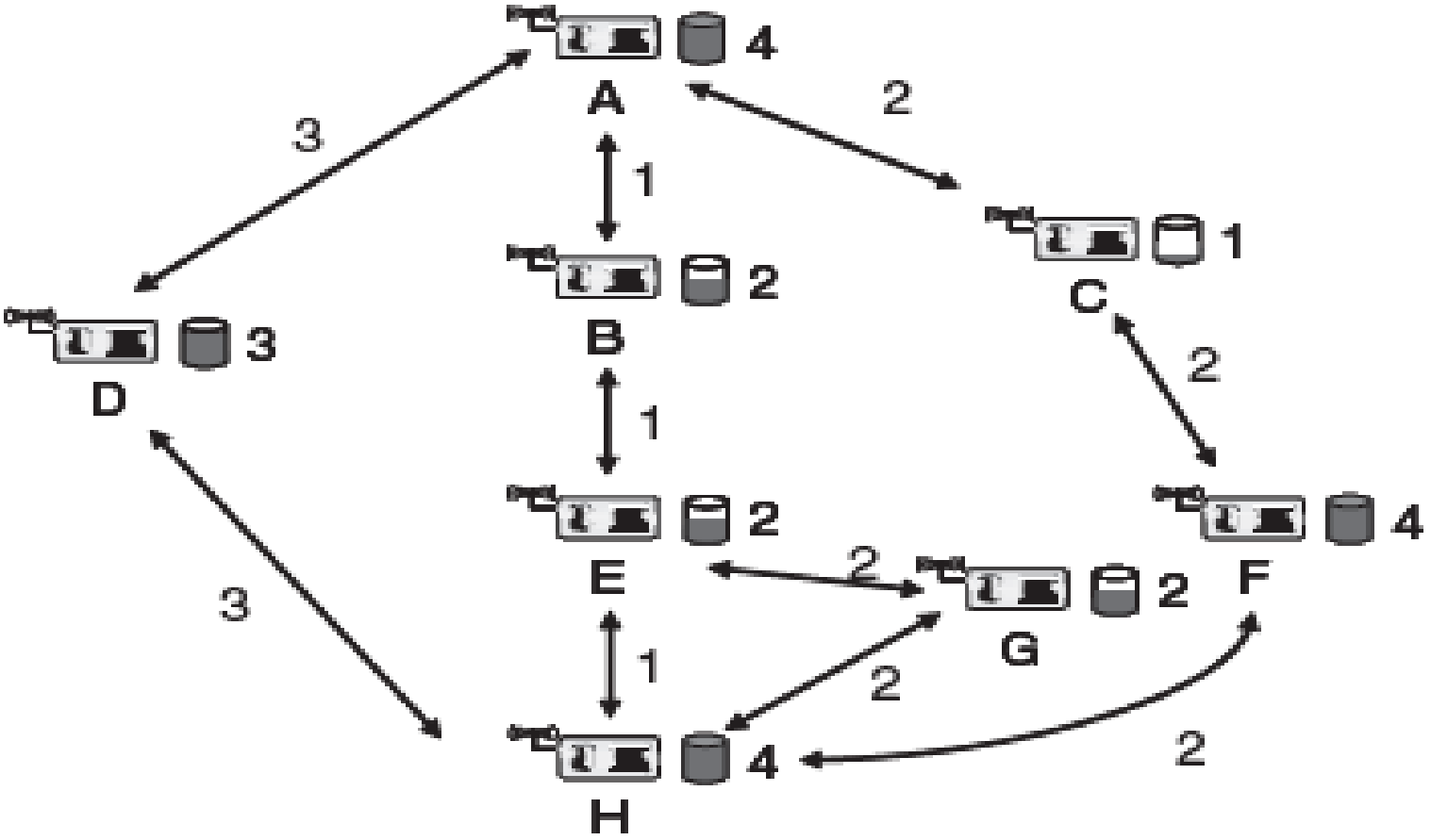
6. Energy Efficient Sensor Routing

- EESR is a flat routing algorithm proposed to decrease the power utilization, data latency and to give scalability in the WSN. It consists of Gateway, Base Station, Manager Nodes, and Sensor Nodes. Their duties are-
- Gateway Delivers messages from Manager Nodes to the Base Station, which has extra specification than normal sensor nodes.
- It sends and receives messages to/from Gateway. Moreover, it sends queries and collects data to/from sensor nodes.
- Manager Nodes and Sensor Nodes collect data from the environment and send it to each other in 1-Hop distance till the Base Station.

Typical Example for Energy Efficient Routing

- Energy-efficient unicast routing is a simple technique.
- It assigns to each link a cost value that reflects the energy consumption across this link.
- It picks any one algorithm that computes least-cost paths in a graph. (For Example: Shortest path algorithm to obtain routes with minimal total transmission power.)
- An example scenario for a communication between nodes ‘A’ and ‘H’ including link energy costs and available battery capacity per node is shown in Figure 3.11.
- Various routes for communication between nodes ‘A’ and ‘H’ also show energy costs per packet for each link and available battery capacity for each node

Fig 11 / Typical Example for EE Routing



- In the above example, it is found that -
- The minimum energy route is A-B-E-H which requires 3 units of energy only. The minimum hop count route would be A-D-H which requires 6 units of energy.
- The following are the important parameters of Energy efficient routing.
  - Minimize Energy per packet
  - Maximize network lifetime
  - Routing considering available battery energy
  - Maximum Total Available Battery Capacity
  - Minimum Battery Cost Routing (MBCR

Topic 13

Geographic Routing

Introduction

- For many applications, it is necessary to address physical locations as “any node in a given region” or “the node at/closest to a given point”.
- When the position of source, destination and the positions of intermediate nodes, are known, this information can be used to assist in the routing process.
- The destination node has to be specified either geographically or as some form of mapping.
- A source node knows the geographic area of the destination and makes an impression on the destination is called geographic routing.
- The area of the nodes is accessible through different techniques like GPS, radio signal and so on.

Importance of Geographic Routing

- Geographic routing (GR) is one in which a node forwards a packet to the neighbor closest to the destination.
- GR is an attractive approach for routing in WSNs due to its low overhead and localized interactions.
- In GR, nodes will interact with their one-hop neighbors to exchange the location information and make localized forwarding decisions.

Aspects of Geographic Routing

- First aspect: sending data to arbitrary nodes in a given region, referred to as geo-casting.
- Second aspect: is called position-based routing or “Cartesian routing”.

- In wireless sensor networks, usually the geo-casting aspect of geographic routing is more important.
- Since nodes are considered as interchangeable and distinguished by external aspects a location service is not necessary.
- Hence, this concentrates on the geo-casting aspect, with position-based routing aspects treated.

Basics of Position Based Routing

- Assume a node wants to send a data packet to a node at known position and every node in the network knows its own position and that of its neighbors.
- In a simple greedy forwarding approach, the packet is forwarded to that neighbor closest to the destination, minimizing the remaining distance for the packet to travel.

Types of Geographic Routing

- Geographic routing has two parts-
  - Geographic Forwarding
  - Face Routing.

1. Geographic Forwarding:

- Geographic forwarding is a greedy routing algorithm based on geography.
- For a given node, all its one-hop neighbors closer to the sink belong to the forwarding set (FS).
- The node forwards an incoming data packet to the neighbor in the FS closest to the sink.
- GR is attractive because it only requires nodes to maintain the location information of their one-hop neighbors.
- Also, routing decisions can be made locally and

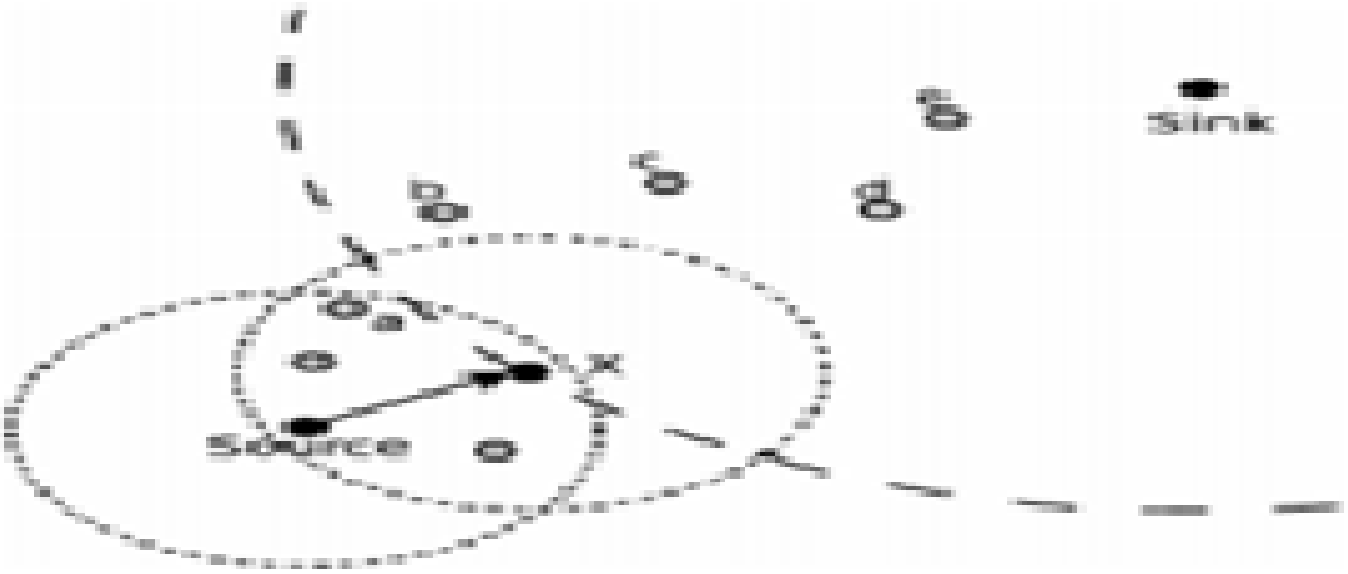
Fig 12 / Geographic Forwarding

X is the neighbor closest to the sink

2. Face Routing:

- However, Geographic Routing does not always succeed in the greedy phase.
- When the forwarding node, e.g., node x in Figure 3.12, has no one hop neighbor closer to the sink than itself, it cannot further forward the incoming packet.
- Thus, the packet is stuck in a local minimum, called a void, where the FS is empty.
- In such a case, a complementary mechanism called face routing or backtracking towards a beacon is used to route around the void.
- Utilizing the area data, the destination region is selected and the packet is sent to that selected region

- This scheme overcomes the issue of limited power in WSN as in some schemes nodes enters into the sleep mode when they are not in used.
- The energy saving depends upon the number of dozing nodes in the network. Fig 13 shows face routing below.



Void: X is a local minimum.

List of Geographic Routing Protocols

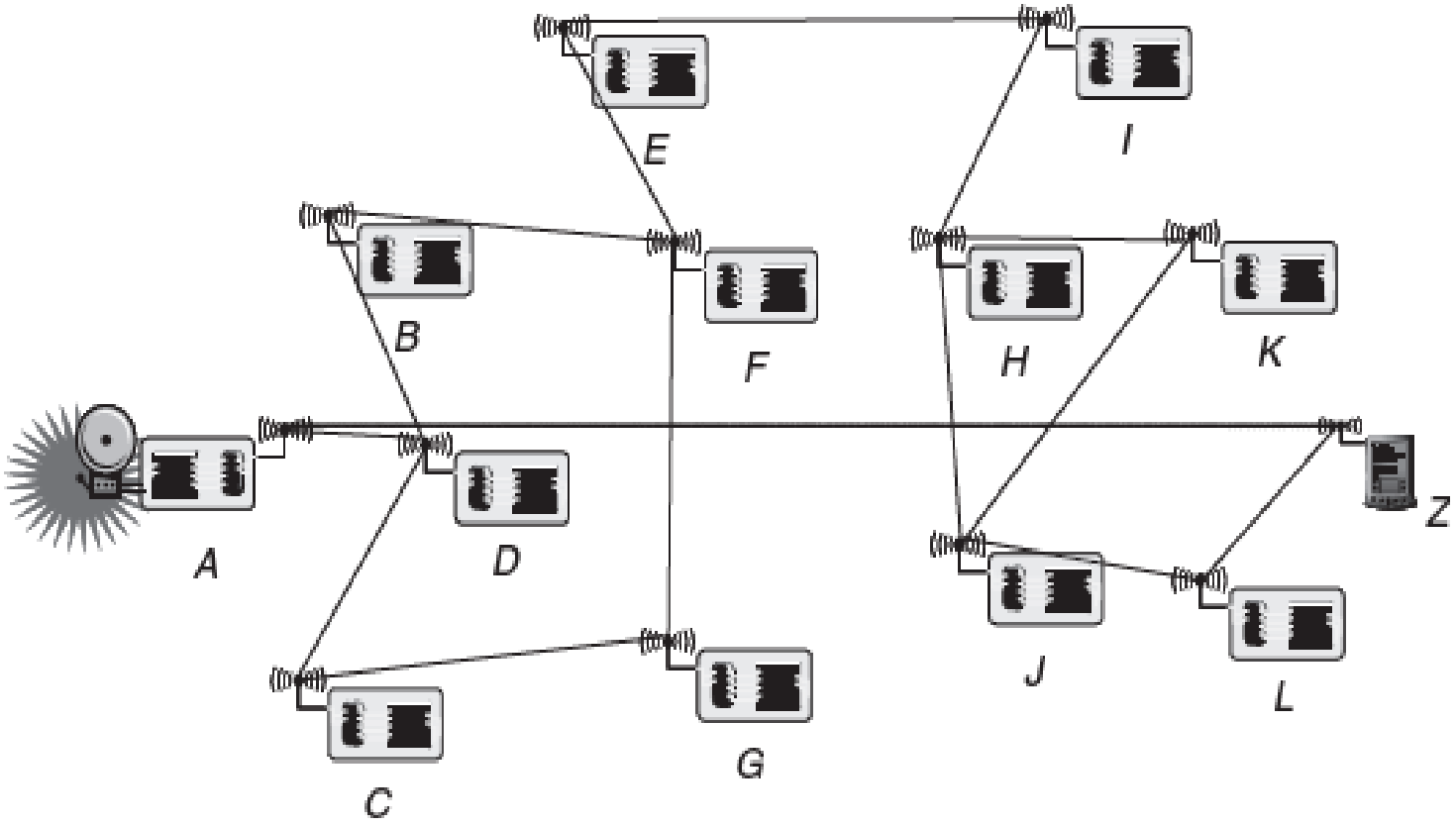
- There are different sorts of geographic routing protocols in wireless sensor networks with each having a different feature of its own. Some of them are given as follows:
  - MECN - Minimum Energy Communication Network
  - GPSR - Greedy Perimeter Stateless Routing Protocol
  - SMECN- Small Minimum Energy Communication Network
  - GEAR - Geographic Energy Aware Routing
  - GAF - Geographic Adaptive Fidelity
  - LAR – Location Aided Routing
  - GOAFR - The Greedy Other Adaptive Face Routing
  - TBF - Trajectory Based Forwarding
  - SPAN- Coordination of Power Saving with Routing

Greedy Perimeter Stateless Routing Protocol

- GPSR forwards a packet using greedy forwarding with the “most forward” rule.
- If a packet cannot make any more progress, the packet is switched to another routing mode called as perimeter routing. A perimeter is a set of nodes defining a face.
- The perimeter routing consists of sending the packet around the face using the right-hand rule.
- To do so, the packet carries information where it entered a given face.
- This node ‘v’, the connecting line between ‘v’ and the destination are used to decide whether the packet should leave and proceed to the next one.

- The packet can return to greedy forwarding if the distance of the current node to the destination and node v has been effectively reduced.
- Figure 3.14 illustrates how a packet will be routed from node A to node Z.
- While at node A, the packet can be greedily forwarded to node D.
- At node D, greedy forwarding fails (both B and C are further away from Z than D itself), so the packet has to be routed round the perimeter of the interior face defined by BFGCD.
- That is, it is forwarded to B and from there to F. Here, edges F, G intersects line DZ and routing can proceed to the next face.

Fig 14 / GPSR Protocol



- The packet proceeds around the perimeter of the exterior face via E and I to H, from there via K to J and then to L and Z.
- Since this face-based procedure is based on properties of the plane, it only applies to planar graphs.
- In general, wireless network graphs are not planar.
- The performance guarantees of combined greedy/face routing.
- When combining face routing and greedy routing, face routing is tasked with routing around obstacles or out of dead ends while greedy routing tries to make quick progress toward the destination

Model Question Bank

PART A

1. What is MAC protocol?
2. What is geographic addressing?
3. State the fundamental tasks of address management in WSN.
4. Differentiate WSN routing with ad hoc routing.
5. What is energy efficient routing?
6. What is geographic routing?
7. Define Assignment of MAC address.
8. What is a Routing protocol?
9. Give the classes of MAC protocols.
10. What is Overhearing?

11. What is idle listening?
12. What is SMAC protocol?
13. What is a mediation device?
14. Mention the advantages of Mediation device protocol.
15. Give the demerits of Wake-up protocol.
16. Give the advantages of BMAC protocol.
17. What is IEEE 802.15.4 standard?
18. What is GTS management?
19. What are the techniques used for address collisions?
20. What is an Initiator?
21. What is LEACH?
22. Give the aspects of geographic routing.
23. Manton any four geographic routing.
24. What is GPSR protocol?

PART B

1. Describe the MAC protocols for WSN in detail.
2. Write short notes on (i) SMAC and (ii) BMAC protocols.
3. Explain the IEEE 802.15.4 standard used for Wireless Personal Area Network and its correlation with Zigbee.
4. Describe briefly the address and name management in WSN.
5. Explain in detail about Energy efficient routing in WSN.
6. Explain in detail about Geographic routing in WSN.

Wireless Sensor Networks

Unit 4 / Infrastructure Establishment

Prepared By

Dr. S.Omkumar

Syllabus / Unit 4

- INFRASTRUCTURE ESTABLISHMENT:
- Topology Control, Clustering, Time Synchronization, Localization and Positioning, Sensor Tasking and Control.

Topic 1

Topology Control

Introduction

- In a dense wireless network, a single node has many neighboring nodes with which direct communication possible when using large transmission power.
- In turn, high transmission power requires lots of energy.
- Many neighbors are a burden for a MAC protocol and routing protocols suffer from volatility in the network.
- To overcome these problems, topology control can be applied.
- The idea is to restrict the set of neighbors of a given node.
- This can be done by controlling transmission power, introducing hierarchies in the network and turning off some nodes for a certain time.

Basic Ideas

- In a crowded network many wireless networking problems are aggravated by the large number of neighbors.
- Many nodes interfere with each other, a lot of possible routes, nodes use large transmission power to talk to distant nodes directly and routing protocols may re-compute their routes.
- The above problems can be overcome by topology-control techniques.
- The topology of a network is determined by the subset of active nodes and the set of active links along which direct communication can occur.

- A topology-control algorithm takes a graph G = (V, E) representing the network where V is the set of all nodes in the network.
- There is an edge $(v_1, v_2) \in E$ if and only if nodes v_1 and v_2 can directly communicate with each other and transforms it to a graph $T = (V_T, E_T)$.
- Figure 1 shows the topology for a dense WSN.

Options for Topology Control

- To compute a modified graph *T* out of a graph *G* representing the original network, a topology control algorithm has the following options:
  - The set of active nodes can be reduced by periodically switching off nodes with low energy and activating other nodes
  - The set of active links or set of neighbors for a node can be controlled.
  - Active links/neighbors can also be rearranged in a *hierarchical* network topology where some nodes assume special roles.

Aspects of Topology Control

- Connectivity: Topology control should not disconnect a connected graph *G*
- Stretch factors: Removing links from a graph will increase the length of a path between any two nodes *u* and *v*. The **hop stretch factor** is defined as the worst increase in path length for any pair of nodes *u* and *v* between the original graph *G* and the topology-controlled path *T*.
- Graph metrics: The importance of a small number of edges in *T* and a low maximum degree (number of neighbors) for each node.
- Throughput: The reduced network topology should be able to sustain a comparable amount of traffic as the original network.

- Robustness to mobility: When neighborhood relationships change in the original graph *G* some other nodes might have to change their topology information.
- Algorithm overhead: The overhead imposed by the algorithm should be small.

Controlling Topology – Power Control

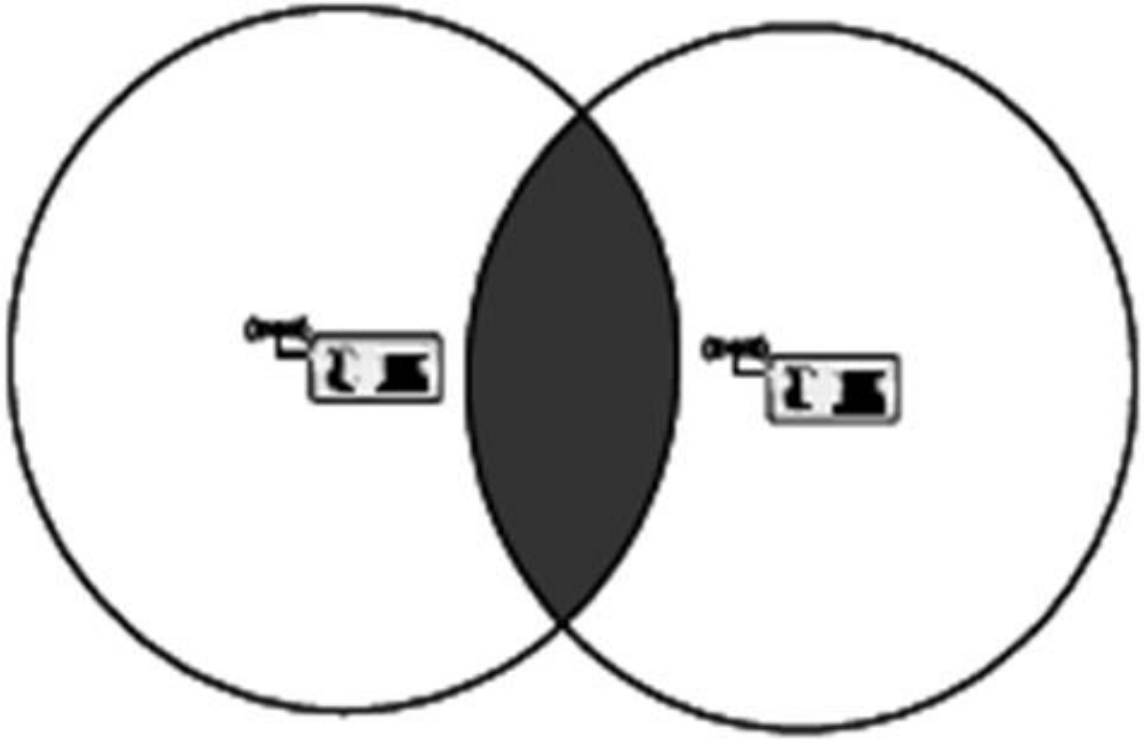
- Controlling the set of neighbors to a particular node is the basic approach of topology control.
- A flat topology is considered where all nodes are operational and have the same tasks. This problem is closely linked to controlling the transmission power of nodes.
- When looking at the connectivity problem, there are two options to approach the problem.
  - Transmission range of a node
  - Number of neighbors
- Under certain assumptions, these two options are equivalent, but lead to different styles of proofs and results.
- The first option, controlling the transmission range, is purely geometric which leads to a uniform distribution of nodes in a given area of size A.

Some Example Protocols

1. Relative Neighborhood Graph (RNG)

- The Relative Neighborhood Graph (RNG) ‘T’ of a graph G = (V,E) is defined as T = (V,E) where there is an edge between nodes ‘u’ and ‘v’ if and only if there is no other node ‘w’ closer to either ‘u’ or ‘v’ than ‘u’ and ‘v’ are apart from each other.
- The RNG is easy to determine with a local algorithm.
- It is also necessarily connected if the original graph G is connected.

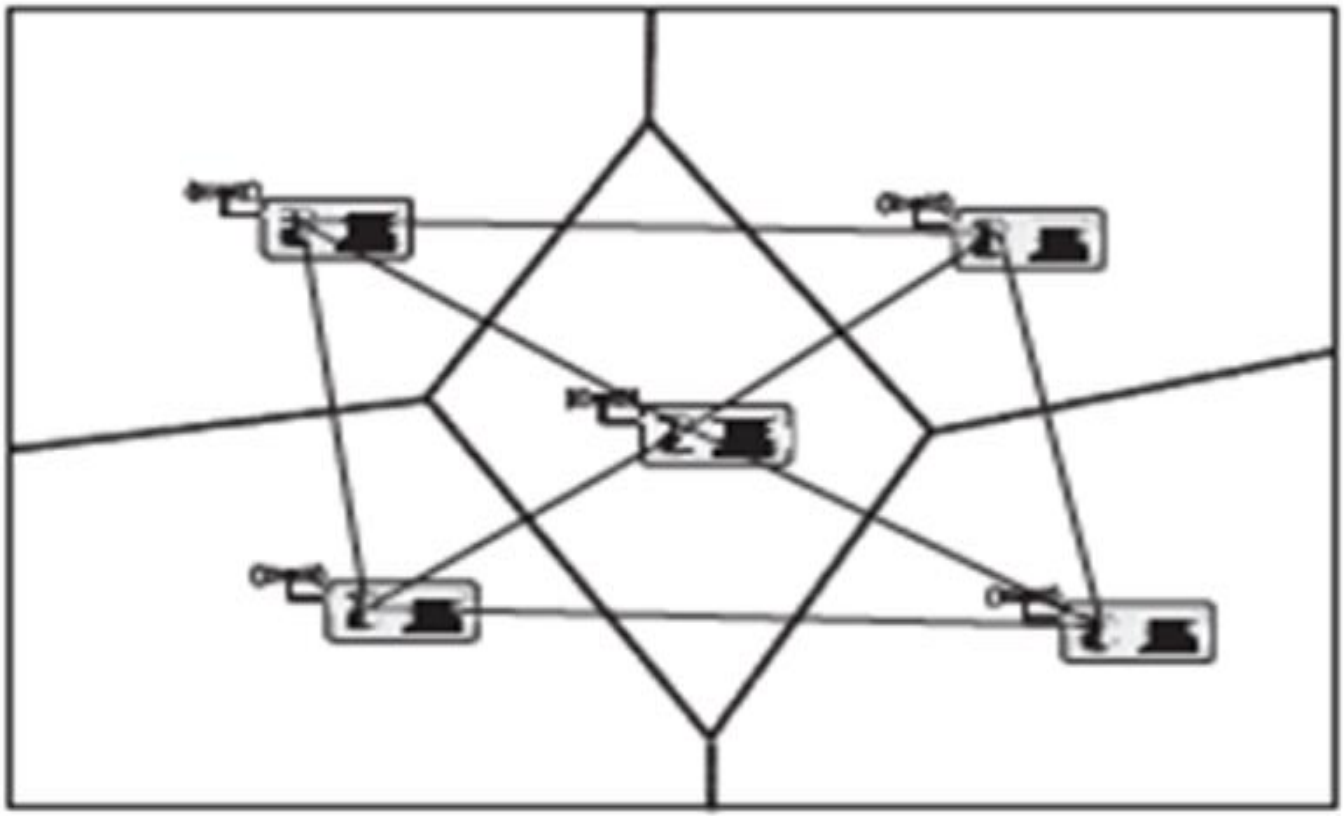
Fig 2 / Construction of RNG



2. Spanning Tree–Based Construction

- Each node will collect information about its neighboring nodes at maximum transmission power.
- A minimum spanning tree can be constructed for these nodes, with energy costs used as link weights.
- The key is to maintain those edges in the reduced topology correspond to direct neighbors in the minimal spanning tree.
- This construction preserves the connectivity of the original graph, and the maximum degree of each node.

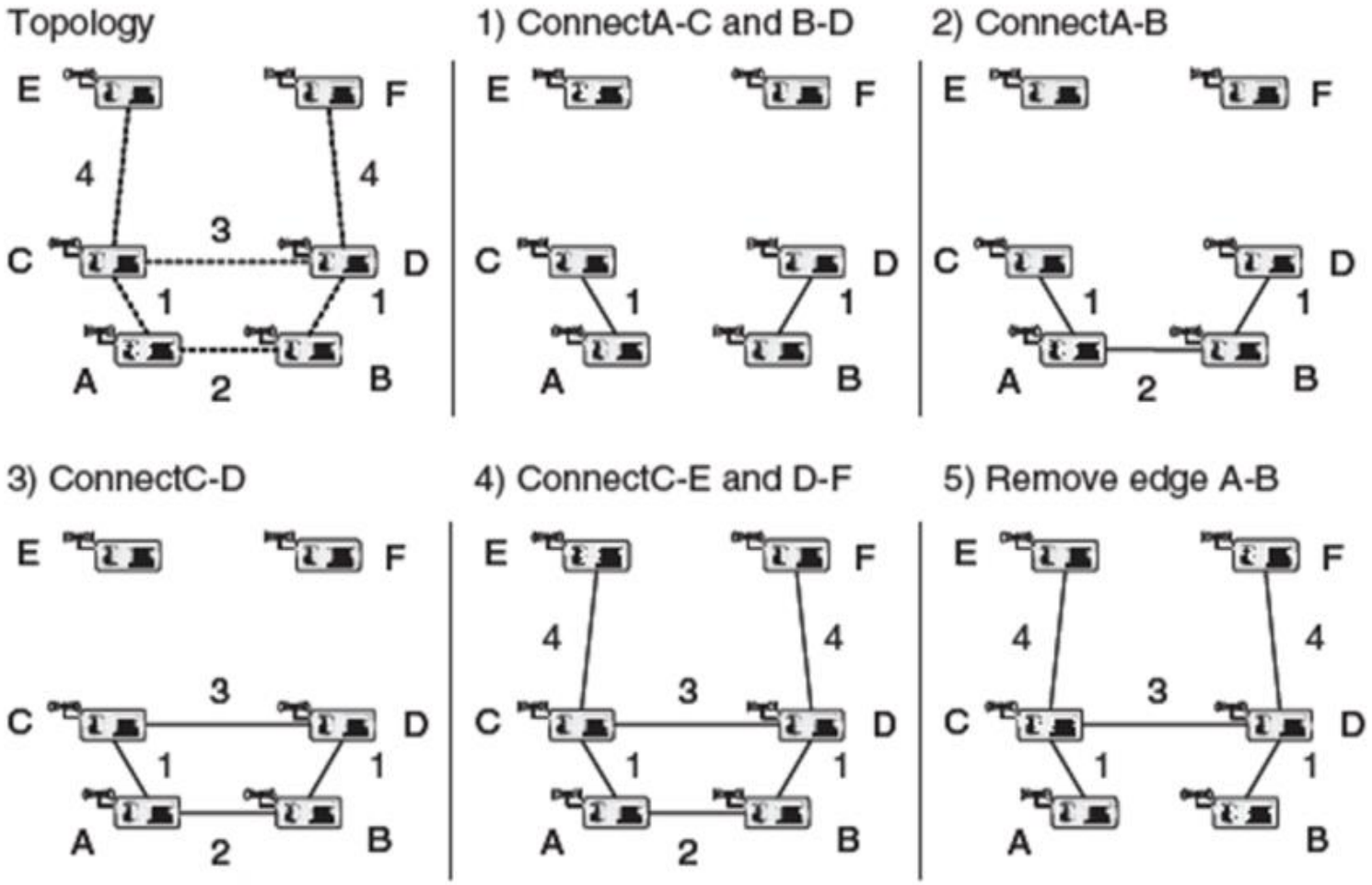
Fig 3 / Spanning Tree Based Construction



3. A Distributed Common Power Protocol – COMPOW:

- The first observation is assigning the identical transmission power to all nodes.
- The second observation is the need to keep the transmission power level low for good connectivity.
- Each node determines routing tables for each transmission power level.
- A node will use the smallest transmission power for which the associated routing table has the same number of entries as the table for the maximum transmission power

Fig 4 / Greedy Algorithm for Minimum Power



Topic 2

Clustering

Introduction

- Clustering allows hierarchical structures to be built on the nodes and enables more efficient use of resources such as frequency spectrum, bandwidth, and power.
- For example, if the cluster size corresponds with the direct communication range of the nodes then simpler protocols can be used for routing and broadcasting within a cluster.
- The same time or frequency division multiplexing can be reused across non-overlapping clusters.

Cluster Heads

- Clustering helps to monitor the health of the network and misbehaving nodes.
- Networks can be comprised of mixtures of nodes having special capabilities such as increased communication range, GPS etc.
- These more capable nodes can play the role of **cluster-heads.**
- **H**owever, the nodes are identical and their common communication range is a natural cluster size.
- There exist several distributed protocols for cluster-head election based on node unique identifiers (UIDs).

Nominators

- It has a higher ID than all its “uncovered” neighbors—neighbors already not claimed by another cluster-head.
- In others, each node nominates a cluster-head, the highest ID node for communication.
- Nominated nodes then form clusters with their nominators.

Gateways

- Nodes that can communicate with two or more cluster-heads may become gateways.
- They aid in passing traffic from one cluster to another and also useful to view the IDs as weights for selecting the cluster-heads.
- The number of clusters obtained is compared with the minimum possible, even in a randomized setting.
- A constant approximation bound can be shown if the leader election protocol is used hierarchically, with increasing node ranges.

Advantages of Clustering

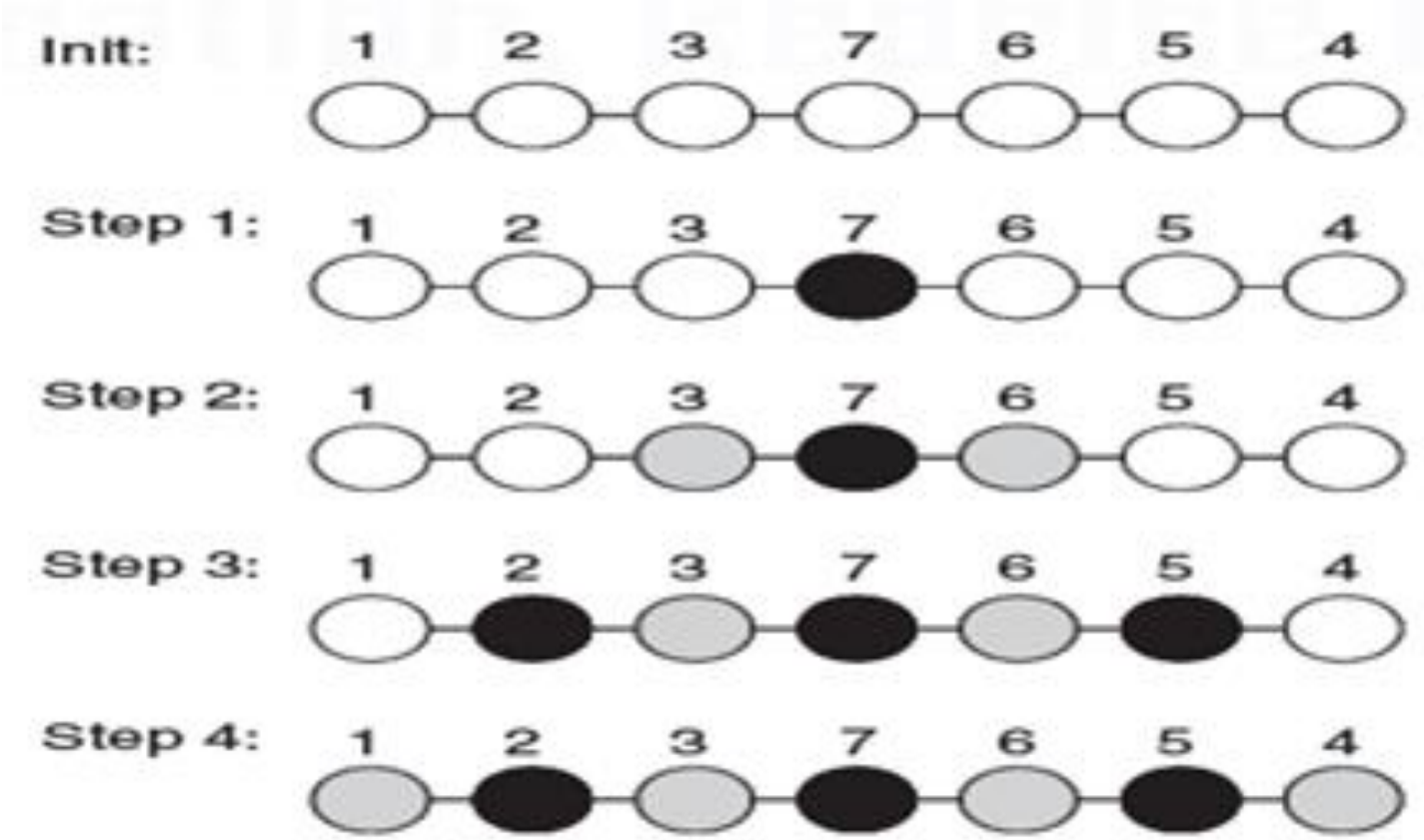
- Clustering can be used to thin out parts of the network where an excessive number of nodes present.
- Used to set up a simplified long range communication network.
- Cluster-heads can have the minimum separation comparable to the node communication range. This property ensures that each cluster-head has a bounded number of cluster-head neighbors.
- Several local communication protocols can become simpler.

Algorithm for determining Independent sets

- Consider a simple linear network shown as Figure 4.5.
- In step 1, nodes 2 and 5 cannot become cluster-heads because their neighboring nodes 3 and 6 have not yet decided and will take precedence over them.
- Once nodes 3 and 6 have learned about node 7 being a cluster-head, they decide to become cluster members and propagate this information to nodes 2 and 5.
- Then, these nodes can become cluster-heads in step 3.
- This essential algorithm has to be modified with the following variations for useful in mobile networks.

- Whether to hold back nodes from forming clusters as long as the cluster-head decision is revised, or to allow intermediate clusters to be formed.
- Which nodes will re-clustered later
- Which may join another cluster-head

Fig 5 / Algorithm for determining Independent Sets



Topic 3

Time Synchronization

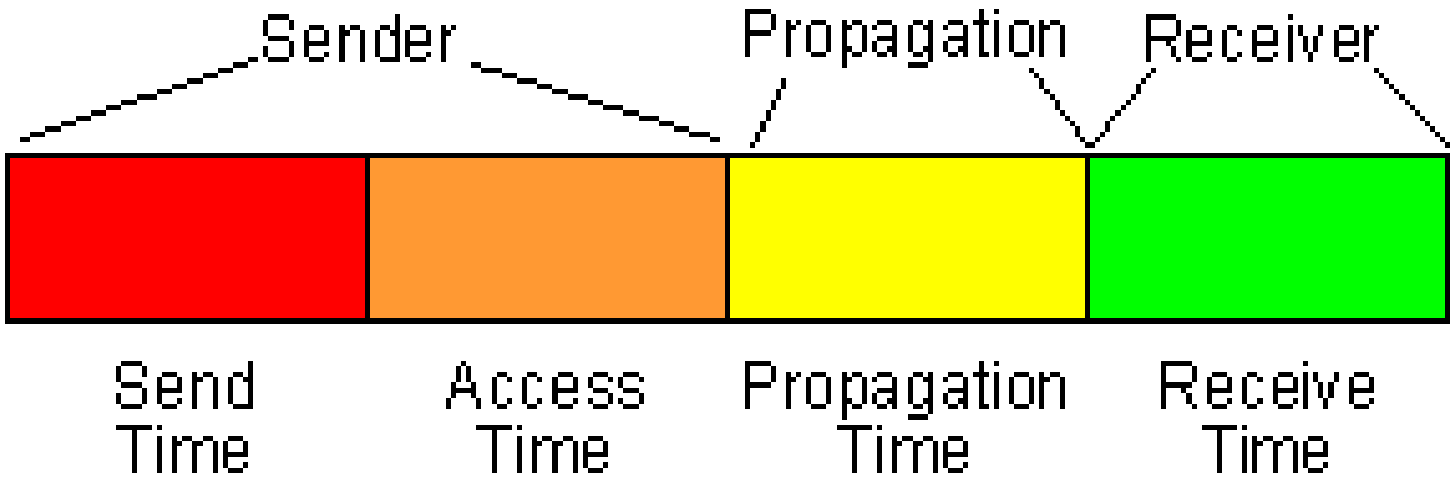
Introduction

- Time synchronization allows for successful communication between nodes on the network.
- Synchronization in wireless nodes implements a TDMA algorithm over a multi-hop wireless network.
- Wireless time synchronization is used for many different purposes including location, proximity, energy efficiency and mobility.
- Time synchronization is used to save energy.
- It will allow the nodes to sleep for a given time and then awaken periodically to receive a beacon signal.
- Many wireless nodes are battery powered, so energy efficient protocols are necessary.

Wireless Network Synchronization

- There are three basic types of synchronization methods for wireless networks.
  - The first is relative timing and the simplest which depends on the ordering of messages and events.
  - The next method is relative timing in which the network clocks are independent of each other and the nodes keep track of drift and offset.
  - The last method is global synchronization where there is a constant global timescale throughout the network.

Fig 6 / Breakdown of Packet Delay Components



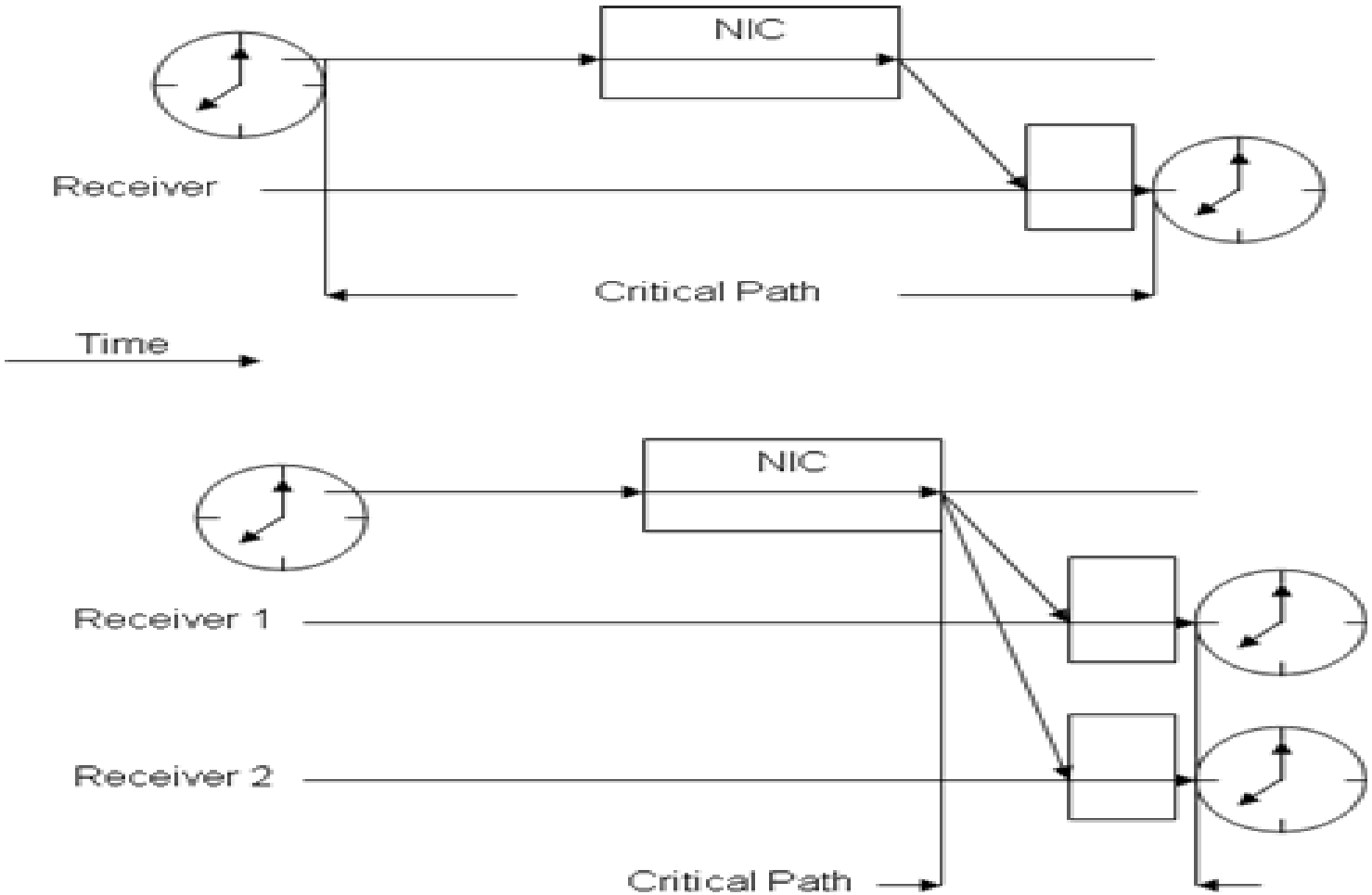
- As shown in Figure 6, all the wireless synchronization schemes have four basic packet delay components:
- send time, access time, propagation time, and receive time.
  - The send time is the sender constructing the time message to transmit on the network.
  - The access time is the MAC layer delay in accessing the network.
  - The receive time is the receiving node processing the message and transferring it to the host.

- There are many synchronization protocols which do not differ much from each other.
  - Reference Broadcast Synchronization (RBS),
  - Timing-sync Protocol for Sensor Networks (TPSN)
  - Flooding Time Synchronization Protocol (FTSP).
  - Lightweight Time Synchronization Protocol (LTS)
- These protocols are the major timing protocols currently in use for wireless networks.
- These protocols cover sender to receiver synchronization as well as receiver to receiver.
- Also, they cover single hop and multi hop synchronization schemes.

Reference Broadcast Synchronization (RBS)

- RBS uses receiver to receiver synchronization and a third party will broadcast a beacon to all the receivers.
- The beacon does not contain any timing information and instead the receivers will compare their clocks to one another to calculate their relative phase offsets.
- The timing is based on when the node receives the reference beacon.
- The simplest form of RBS is one broadcast beacon and two receivers.
- The timing packet will be broadcasted to the two receivers. The receivers will record when the packet was received according to their local clocks.

Fig 7 / Traditional Synch with RBS



- Then, the two receivers will exchange their timing information and be able to calculate the offset.
- RBS can be expanded from the simplest form of one broadcast and two receivers to synchronization between ‘n’ receivers; where n is greater than two.
- The main advantage of RBS is it eliminates the uncertainty of the sender by removing the sender from the critical path.
- By removing the sender, the only uncertainties are the propagation and receive time.

Time Synch Protocol for Sensor Networks

- TPSN is a traditional *sender-receiver* based synchronization that uses a tree to organize the network topology.
- The concept is broken up into two phases, the level discovery phase and the synchronization phase.

1. Level Discovery Phase:

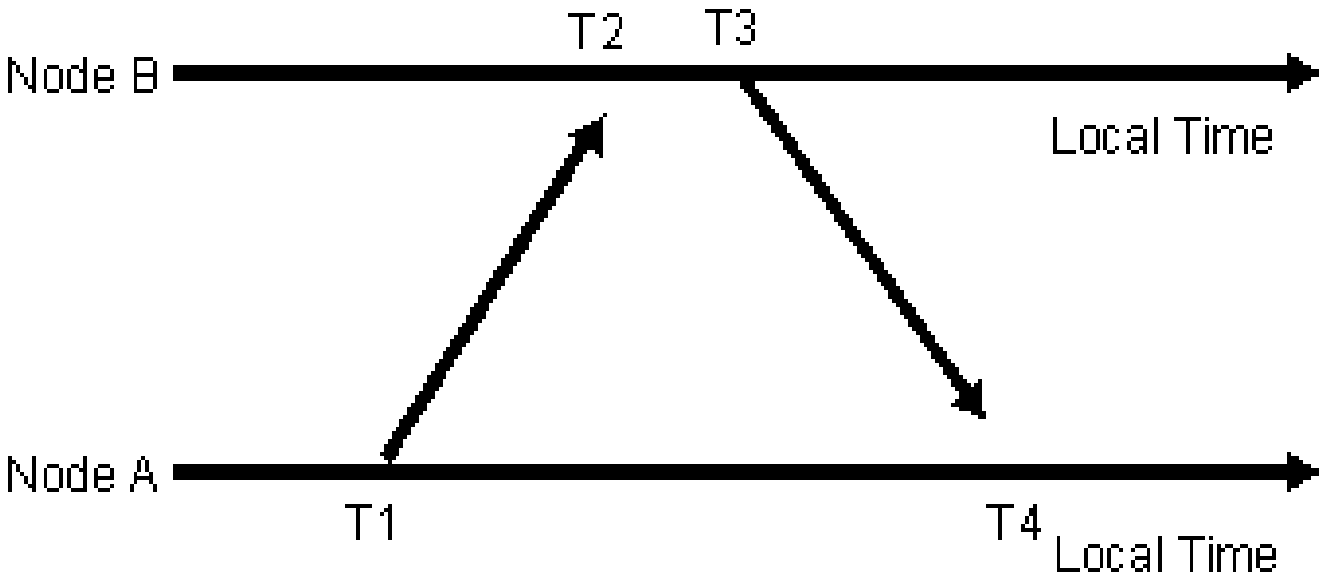
- First, the root node should be assigned. If one node is equipped with a GPS receiver, then it can be the root node
- Now the root node will send out the *level discovery* packet to its neighboring nodes. The identity and level of the sending node will be included in the packet.
- The neighbors of the root node will then assign themselves as level one. They will in turn send out the *level discovery* packet to their neighboring nodes.

- This process continues until all nodes received the *level discovery* packet and are assigned a level

2. Synchronization Phase:

- Similar to the level discovery phase, the synchronization phase begins at the root node and propagates through the network.
- Figure 4.8 illustrates the two-way messaging between a pair of nodes by following this method.
- The times T1, T2, T3, and T4 are all measured times.

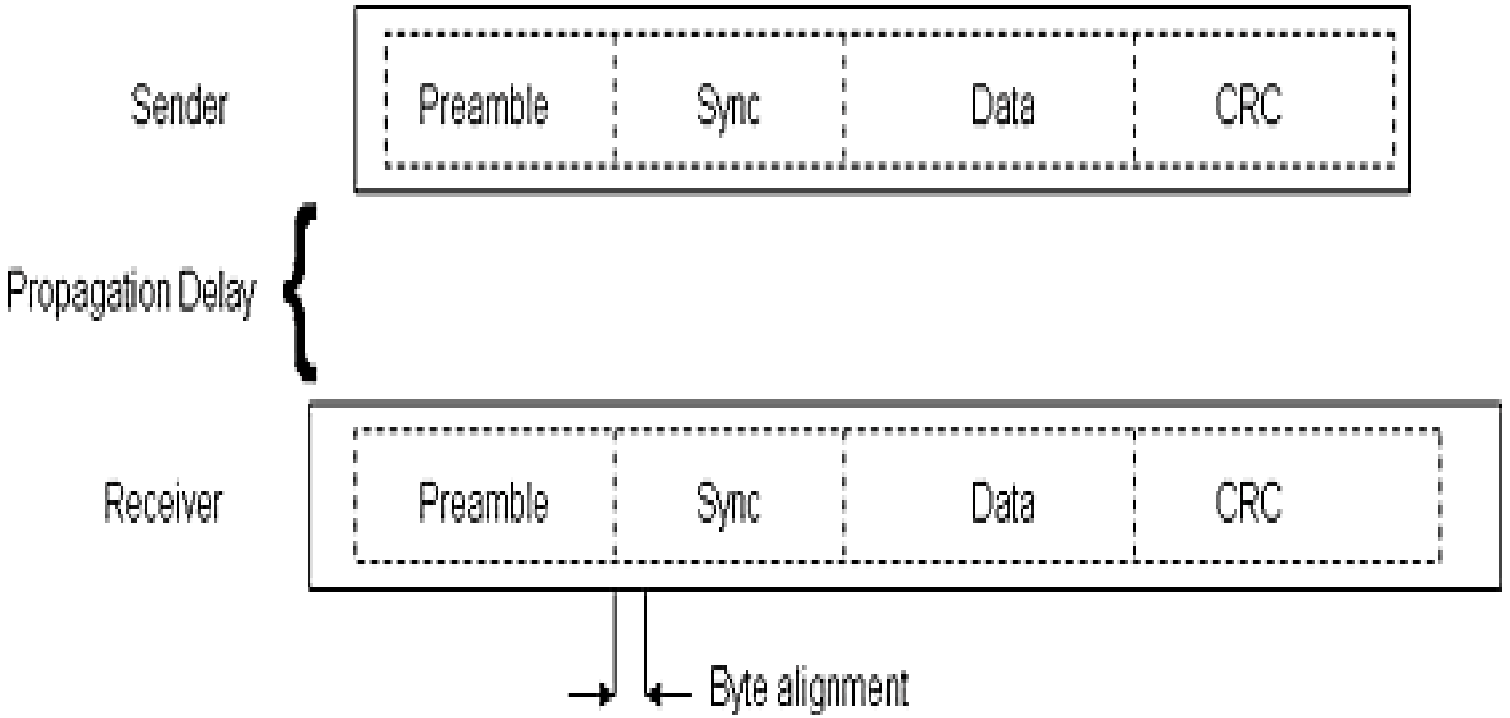
Fig 8 / Two Communication between Nodes



Flooding Time Synch Protocol (FTSP)

- It is similar to TPSN that it has a structure with a root node and that all nodes are synchronized to the root.
- The root node will transmit the time synchronization information with a single radio message to all participating receivers.
- The message contains the sender's time stamp at transmission. The receiver notes its local time when the message is received.
- Having the sender's transmission time and the reception time, the receiver can estimate the clock offset.
- FTSP was designed for large multi-hop networks. The root is elected dynamically and responsible for keeping the global time of the network.
- The network structure is mesh type topology instead of a tree topology as in TPSN.

Fig 9 / Packets transmitted using FTSP



Light Weight Time Synch Protocol

- The lightweight time synchronization (LTS) protocol is meant to synchronize the clocks of a sensor network to the clocks held by certain reference nodes having GPS receivers.
- The protocol has control knobs to trade off energy expenditure and achievable accuracy.
- LTS makes no restrictions with respect to the local clock and not try to estimate actual drift rates.
- LTS subdivides time synchronization into two building blocks:
  - A pair-wise synchronization protocol synchronizes two neighboring nodes.
  - To keep all nodes synchronized to a common reference, a spanning tree from the reference node to all nodes is constructed.

Topic 4

Localization & Positioning

Introduction

- Localization is an important aspect in the field of wireless sensor networks.
- The task of determining physical coordinates of sensor nodes in WSN is known as localization or positioning and is a key factor in today's communication systems to estimate the place of origin of events.
- Different localization methods are used in different applications and there are several challenges in some special scenarios such as forest fire detection.
- There are three approaches available to determine the position of a node.
- They are –

1. Proximity:

   Using information about the neighborhood of a node.

2. Triangulation and Trilateration:

   Exploiting geometric properties of a given scenario

3. Scene Analysis:

   To analyze characteristic properties of the position of a node in comparison with premeasured

Proximity

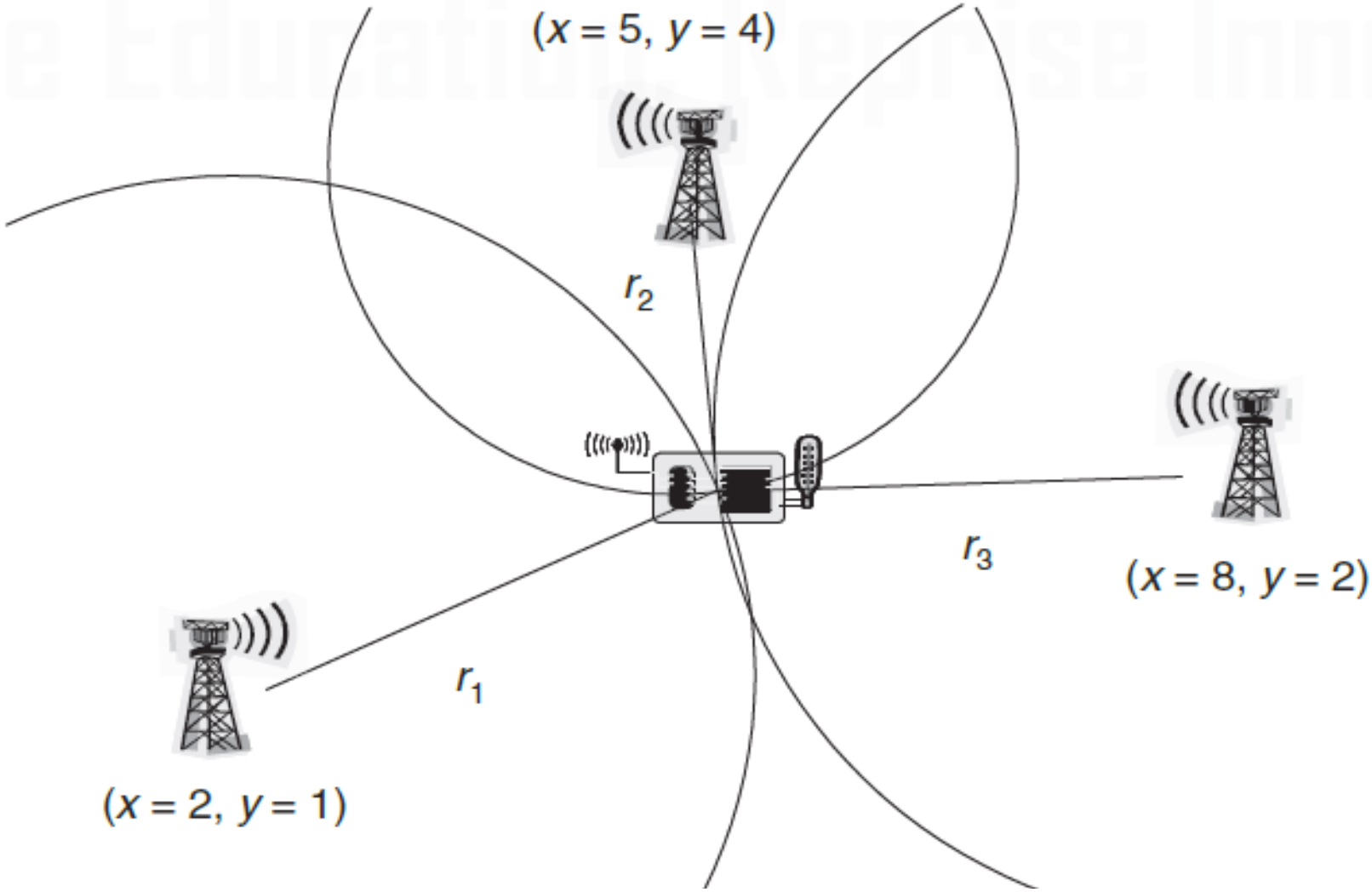
- It can be used to decide whether a node wants to determine its position or location in the proximity of an anchor.
- Typical example is the natural restriction of infrared communication by walls, which can be used to provide a node with simple location information about the room.
- Proximity-based systems are quite sophisticated and used for approximate positioning when a node can analyze proximity information of several overlapping anchors .

Trilateration (Determining the distances)

- The geometrical information can be used to derive information about node positions.
- When distances between entities are used, the approach is called **Lateration** and when angles between nodes are used, it is called A**ngulation**.
- In order to achieve perfections, the distance measurements from more than three anchors can be used, resulting in an approach called m**ultilateration.**
- To use multi-lateration, estimates of distances to anchor nodes are required.
- This ranging process uses the facilities already present on a wireless node, in particular, the radio communication device.

- The characteristics of wireless communication are determined by the distance between sender and receiver at the receiver, they can serve as an estimator of distance. The most important characteristics are –
  1. Received Signal Strength Indicator (RSSI)
  2. Time of Arrival (ToA)
  3. Time Difference of Arrival (TDoA)

Fig 10 / Triangulation by intersecting 3 circles



1. Received Signal Strength Indicator (RSSI)

- If the transmission power PT, path loss model, and the path loss coefficient ‘α’ are known, the receiver can use the received signal strength ‘P’ to solve for the distance ‘d’ in a path loss equation given by –

$$P_{\text{rcvd}} = c\frac{P_{\text{tx}}}{d^{\alpha}} \Leftrightarrow d = \sqrt[\alpha]{\frac{c\,P_{\text{tx}}}{P_{\text{rcvd}}}}.$$

- The distance estimates can be derived without additional overhead from communication.
- The disadvantage is that RSSI values are not constant but heavily oscillate, even when sender and receiver are fixed.
- This is caused by effects like fast fading and mobility of the environment

2. Time of Arrival (Time of Flight)

- If both sender and receiver know the time of transmission, this parameter at the receiver can be used to compute propagation time and distance.
- Depending on the transmission medium used, time of arrival requires very high resolution clocks to produce results of good accuracy.
- For sound waves, these requirements are modest and very hard for radio wave propagation.

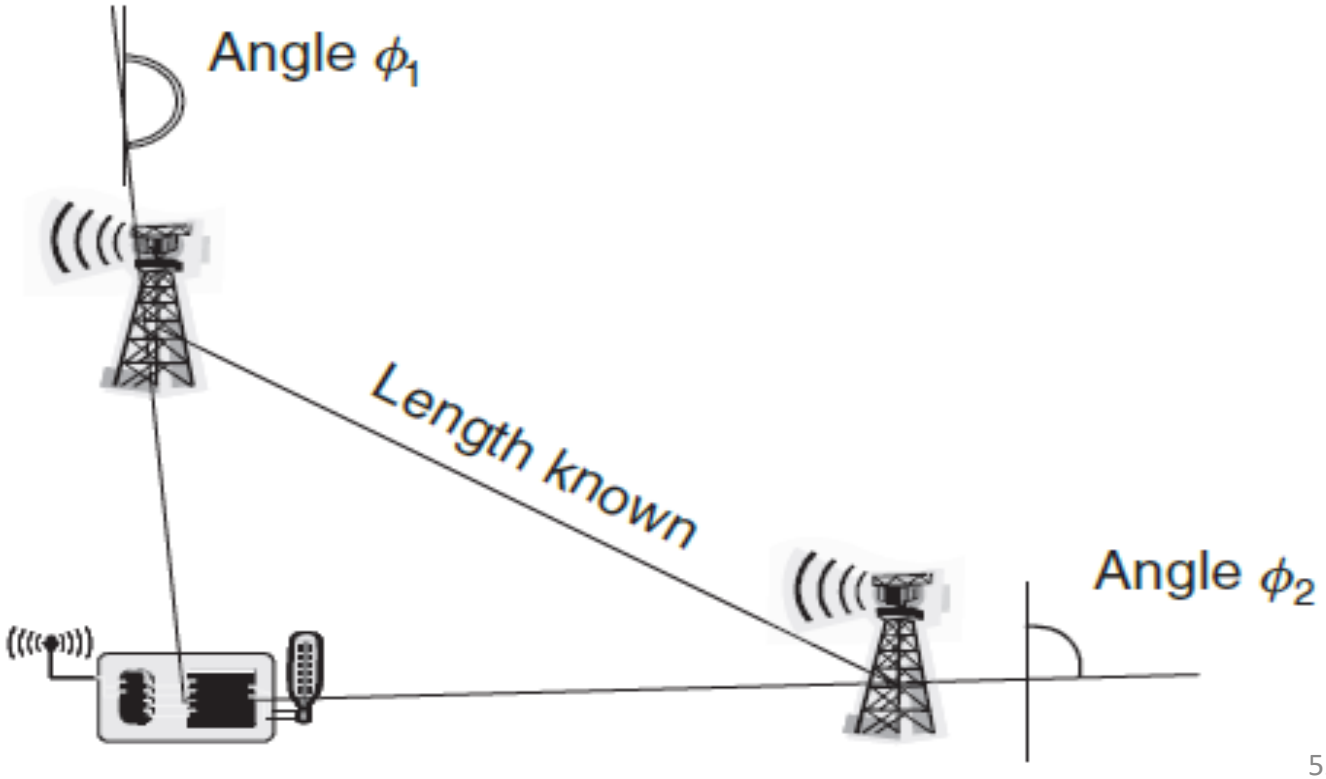
3. Time Difference of Arrival:

- The Time Difference of Arrival (TDoA) method directly provides the start of transmission information to the receiver.
- This can be done if two transmission mediums of very different propagation speeds are used – for example, radio waves propagating at the speed of light and ultrasound, with a difference in speed.
- The disadvantage of this approach is the need for two types of senders and receivers on each node.
- The advantage is better accuracy when compared to RSSI-based approaches.

Triangulation (Determining the Angles)

- Measuring angle can either be angle of a connecting line between an anchor and a position-unaware node to a given reference direction.
- It can also be angle between two such connecting lines if no reference direction is commonly known to all nodes as shown in Figure 4.11.
- A traditional approach to measuring angles is to use directional antennas rotating on their axis, similar to a radar station or a conventional lighthouse.
- This makes angle measurements simple, but inappropriate for sensors nodes.

- In another technique, multiple antennas are mounted on a device at known separation and measuring the time difference between a signal's arrival at the different antennas, the direction from which a wave front arrived at the device can be computed. Fig 11 shows angulation between two anchors.



Scene Analysis

- This method will analyze pictures taken by a camera and derive the position from this picture.
- This requires computational effort and hardly appropriate for sensor nodes.
- Apart from pictures, other measurable characteristic “fingerprints” of a given location can be used for scene analysis, for example, radio wave propagation patterns.
- One option is to use signal strength measurements of one or more anchors transmitting known signal strength and compare the actually measured values with those that of stored in a database.

Single Hop Localization

- Using the above basic building blocks of distance or angle measurements a number of positioning or locationing systems have been developed.
- In single-hop systems, a node with unknown position can directly communicate with anchors.
- These single-hop systems predate wireless sensor networks but provide the basic technology upon which multi-hop systems are built.
- They are –

1. Active Badge

- The first system designed and built for locating simple, portable devices – badges – within a building. Which uses infrared as the transmission medium.

2. Active office

- This system targets the positioning of indoor devices. Here ultrasound is used with receivers placed at well-known position mounted in array at the ceiling of a room.

3. RADAR

- This system is geared toward indoor computation of position estimates. It employs the scene analysis techniques, comparing the received signal characteristics from multiple anchors with premeasured and stored characteristic values.

4. Cricket

- Cricket is an example for systems which can compute their own locations when privacy issues become relevant. It is also based on anchors spread in a building, which provide combined radio wave and ultrasound pulses to allow measuring of signal strength information.

5. Overlapping Connectivity

- This is an example for an outdoor positioning system that operates without any range measurements. It uses only the observation of connectivity to a set of anchors to determine a node's position.

6. Approximate point in a Triangle

- The idea is to decide whether a node is within or outside of a triangle formed by any three anchors. Using this information, a node can intersect the triangles and estimate its own position, similar to the intersection of circles.

7. Using Angle of Arrival Information

- This method use anchors nodes that use narrow, rotating beams where the rotation speed is constant and known to all nodes. Nodes can measure the time of arrival of each such beam, compute the differences between two consecutive signals, and determine the respective angles using geometric relationships

Topic 5

Sensor Tasking & Control

Introduction

- To efficiently utilize resources such as limited on-board battery and limited bandwidth in a sensor network, sensor nodes must carefully tasked and controlled to carry out the required set of tasks.
- A utility-cost-based approach to sensor network management is to address the balance between utility and resource costs.
- The definitions for utility and cost are given below -
  - Utility: Total utility of the data
  - Cost: power supply and the communication bandwidth
- Among the total number of nodes, which sensor nodes are to be activated and what information to transmit to the network is a critical issue.
- This is because the sense values are not known and the cost of sensing may vary with the data.

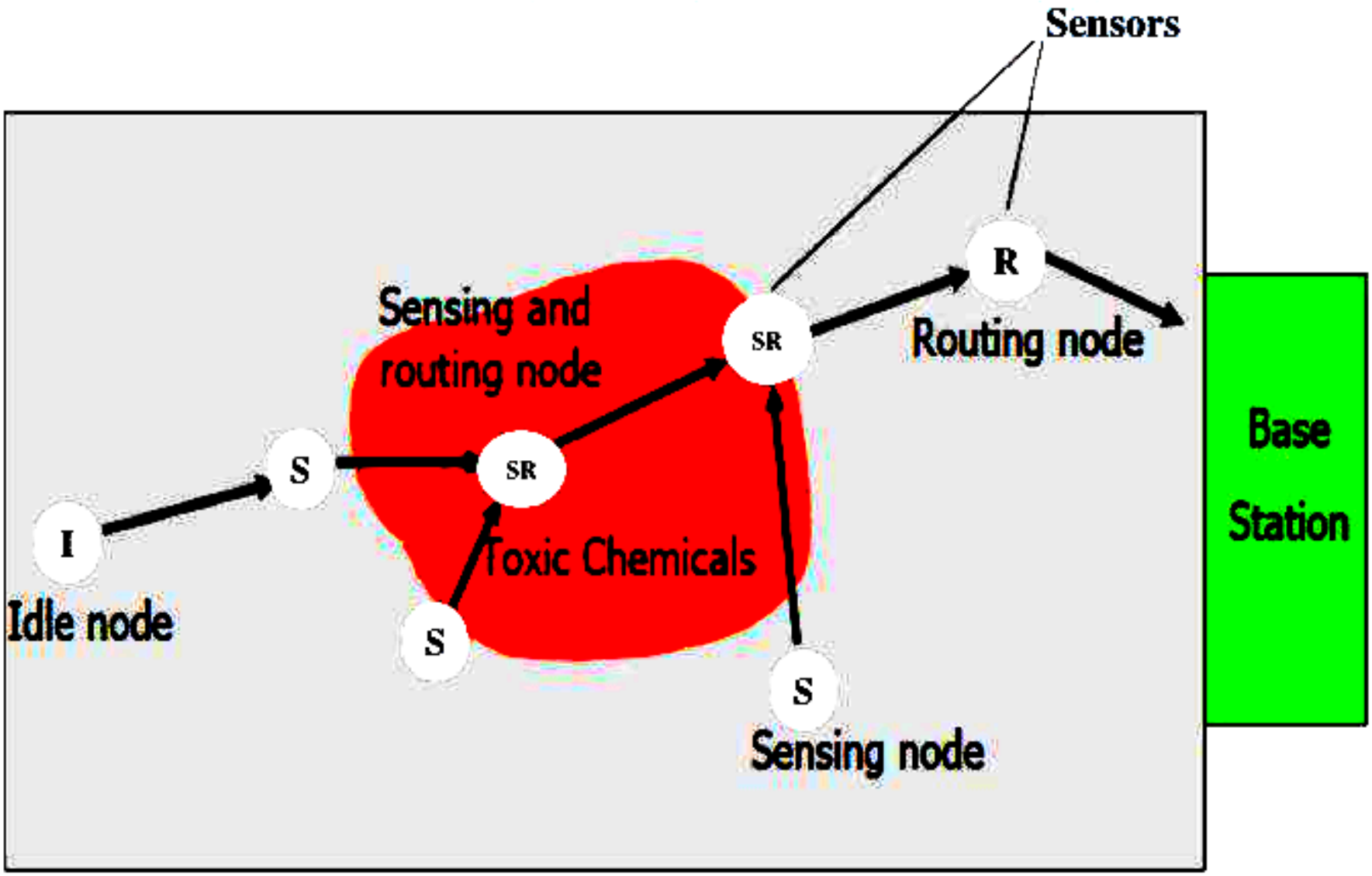
Design Strategy for Sensor Tasking & Control

- The following are the various steps connected with design strategy for sensor tasking and control -
  - The important objects in the environment to be sensed
  - The relevant parameters of these objects
  - The relations among these objects critical to high level information to be known
  - The best sensor to acquire a particular parameter
  - The sensing and communication operations needed to accomplish the task
  - The co-ordination given by the models of different sensors
  - The level of communicate information in a spectrum from a signal to symbol

Role of Sensor Nodes & Utilities

- A sensor may take on a particular role depending on the application task requirement and resource availability such as node power levels as shown in Figure 4.12. For example -
  - Nodes, denoted by SR, participate in both sensing and routing.
  - Nodes, denoted by S, perform sensing only and transmit their data to other nodes.
  - Nodes, denoted by R, decide to act only as routing nodes, especially if their energy reserved is limited.
  - Nodes, denoted by I, be in idle or sleep mode, to preserve energy.

Fig 12 / Role of Sensor Nodes



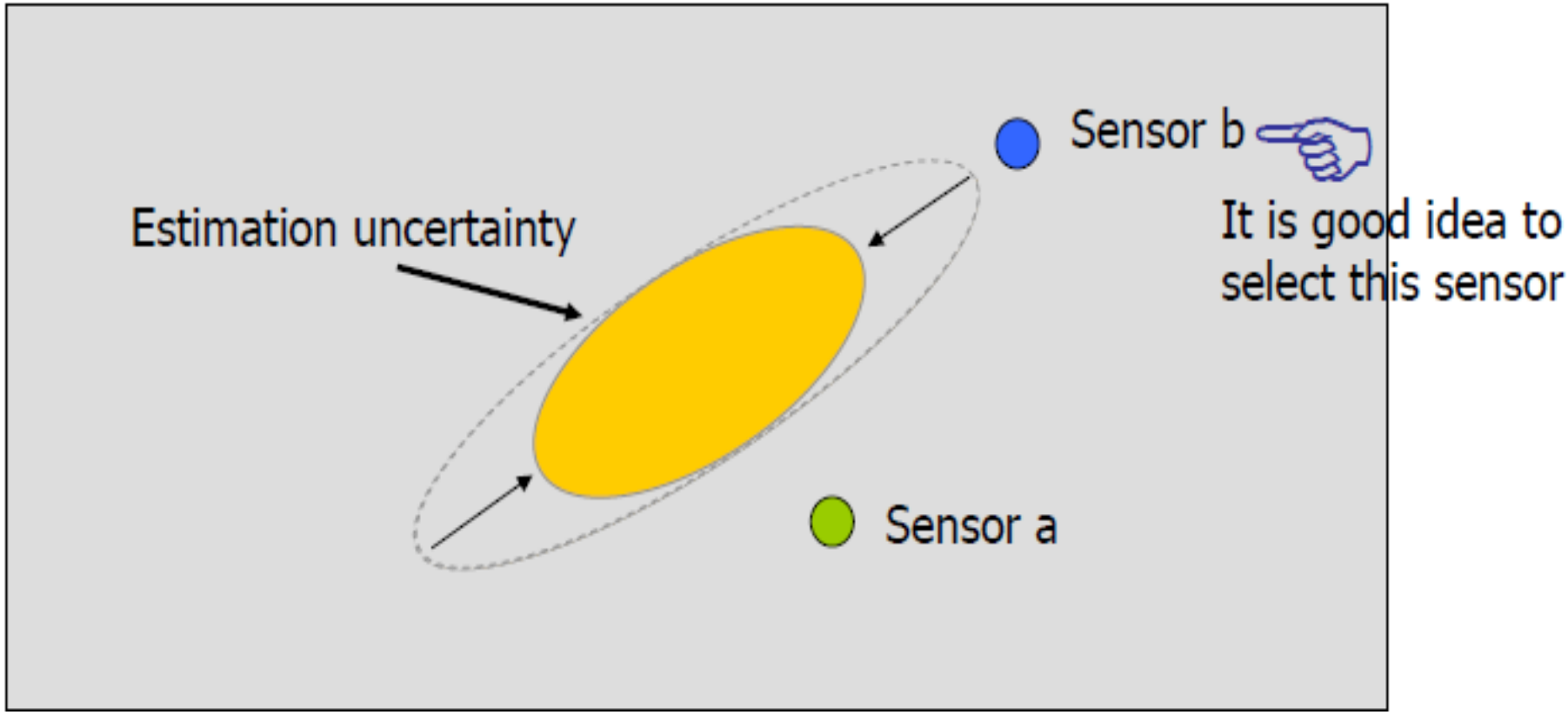
Information Based Sensor Tasking

- Information-based sensor tasking is to query sensors such that information utility is maximized while minimizing communication and resource usage.
- For localization or tracking problem, knowledge about the target state such as position and velocity is required. This requirement is represented as a probability distribution over the state space in the probabilistic framework.

1. Sensor Selection:

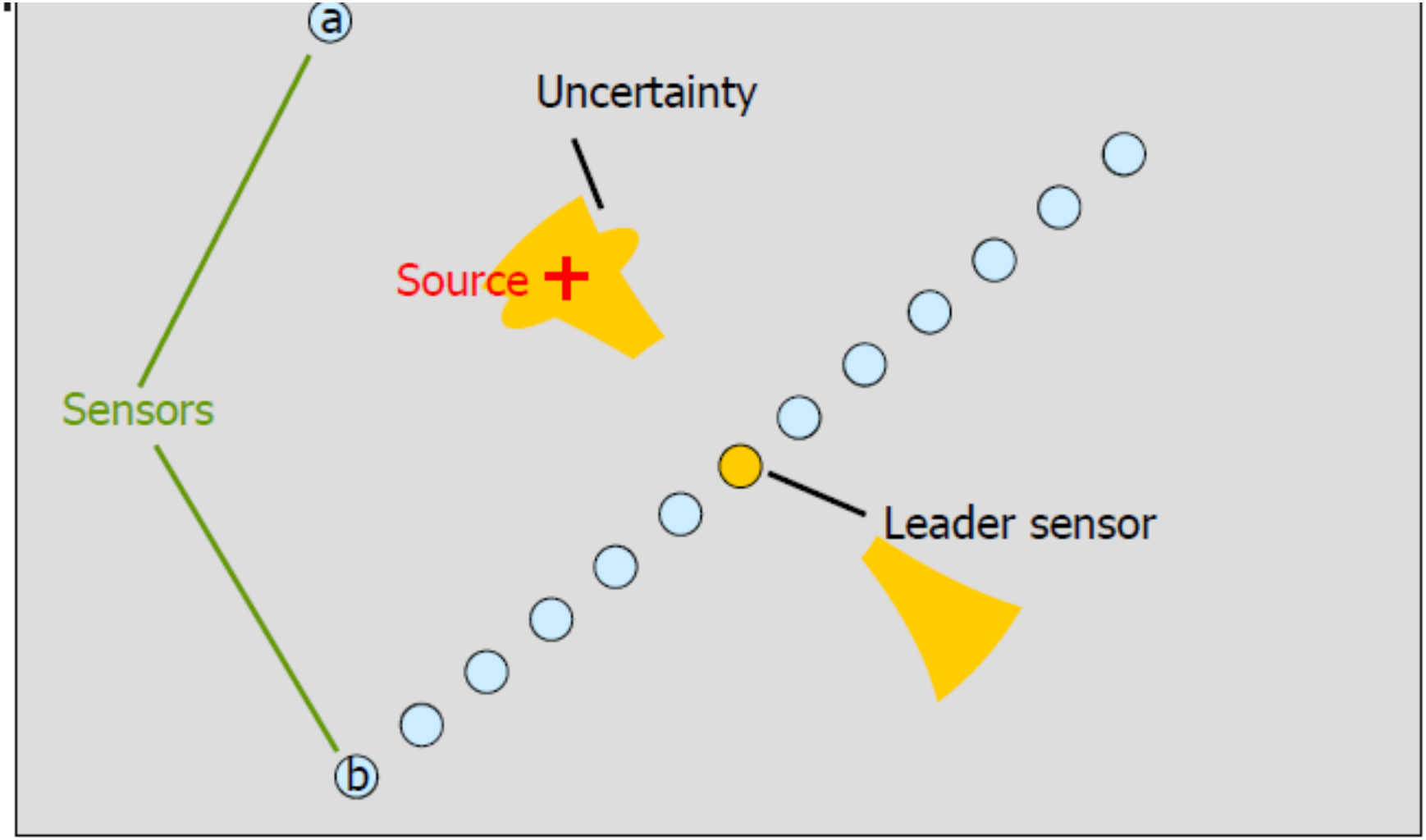
- The estimation uncertainty can be approximated by a Gaussian distribution, illustrated by uncertainty ellipsoids in the state space.
- Sensor ‘b’ would provide better information than because sensor ‘b’ lies close to the longer axis of the uncertainty ellipsoid and its range constraint will intersect this longer axis transversely. Figure 4.13 shows the sensor selection.

Fig 13 / Sensor Selection based on Information Gain



- The following conditions are assumed. Figure 4.14 shows localizing a stationary source.
  - All sensor nodes can communicate with each others.
  - Sensor ‘a’ is farther from the leader node than the sensor ‘b’
  - There are four different criteria for choosing the next sensor.
    - Nearest Neighbor Data Diffusion
    - Mahalanobis distance
    - Maximum likelihood
    - Best Feasible Region

Fig 14 / Localizing a Stationary Source



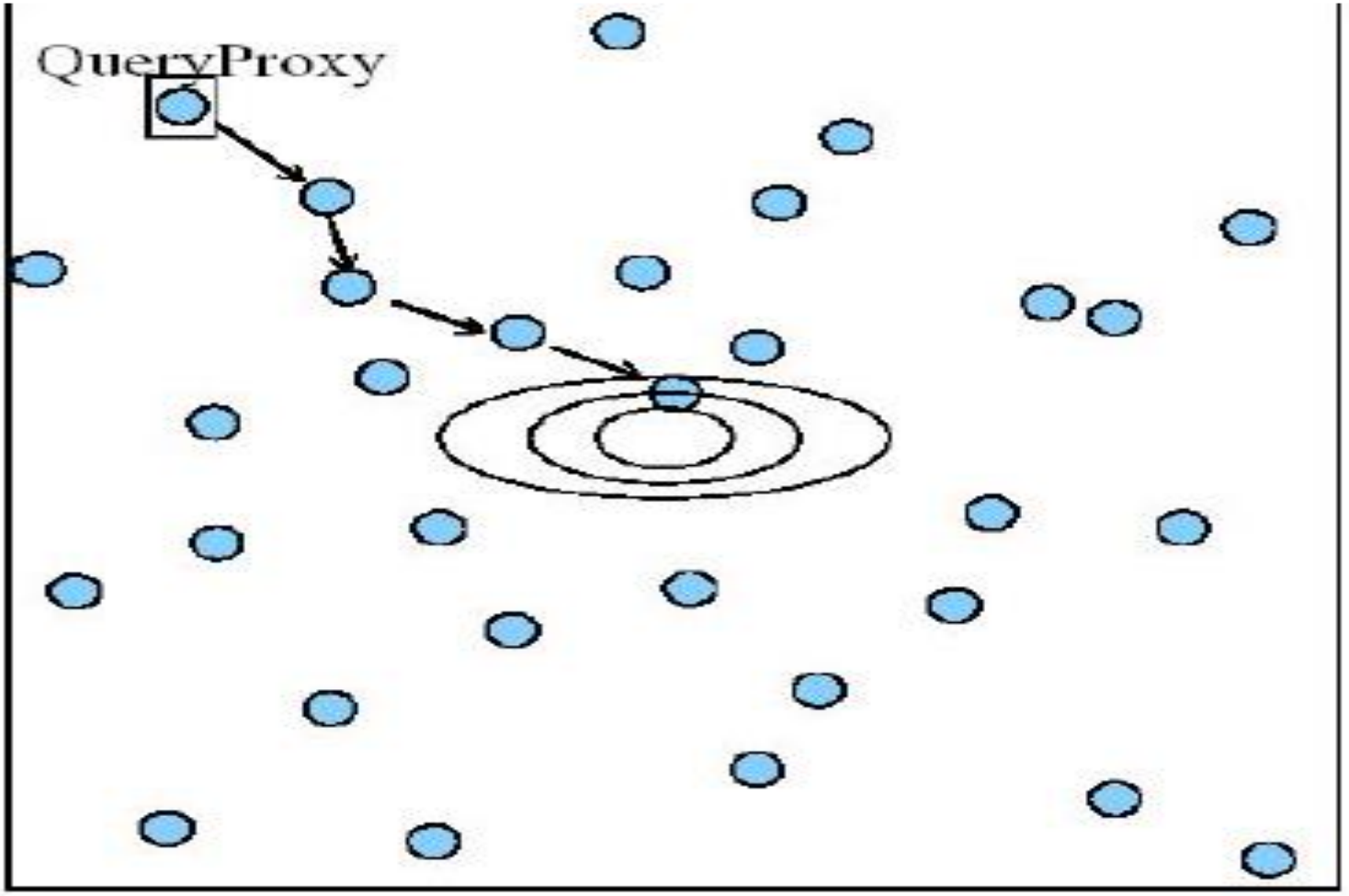
2. Algorithm for IDSQ

- A cluster leader selects optimal sensors to request data from using the information utility measures.
- Using the Mahalanobis distance measure, the cluster leader can determine which node can provide the most useful information while balancing the energy cost, without the need to have sensor data first.
- This algorithm is a single belief carrier node active at a time.

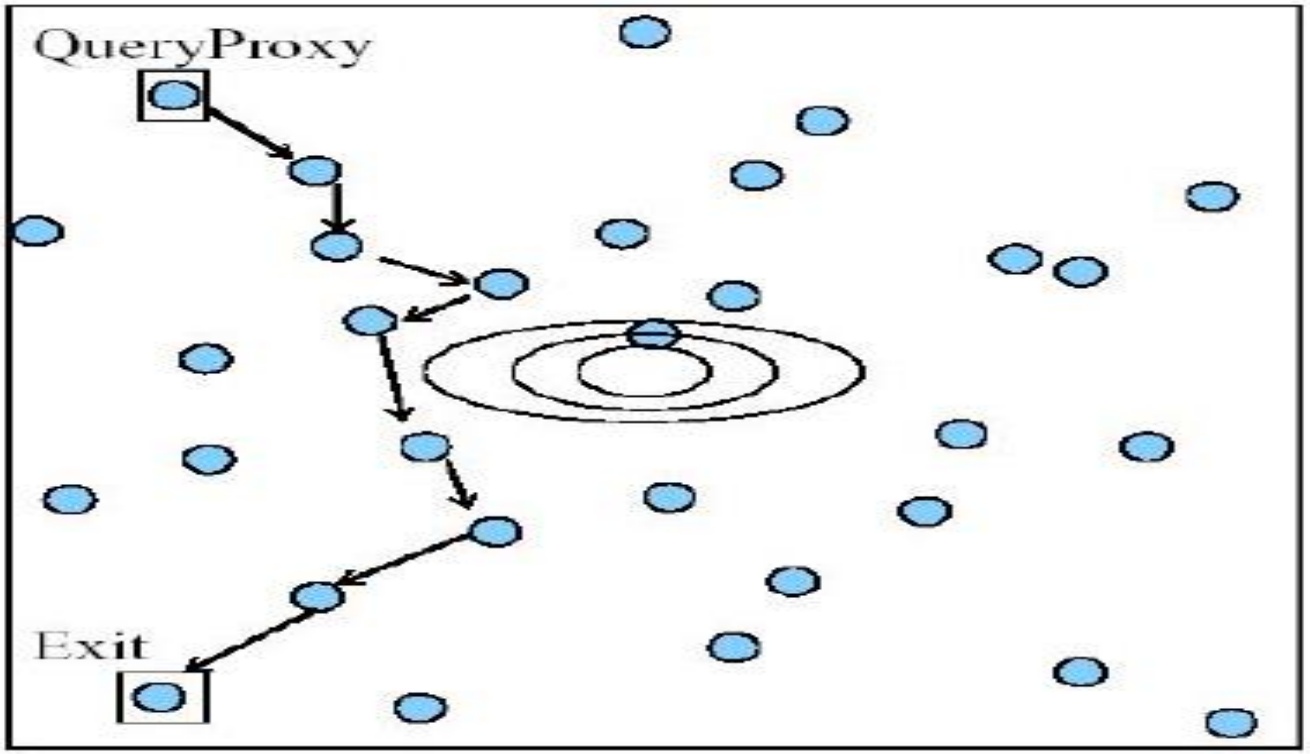
Joint Routing & Information Aggregation

- The primary purpose is to collect and aggregate information.
- Information Driven Sensor Querying (IDSQ) just only provides us with a method to obtain maximum incremental information gain.
- There are some techniques to dynamically determine the optimal routing path.
- The ellipses represent iso-contours of an information field. The goal of routing is to maximally aggregate information.
- This differs from routing in communication networks where the destination is often known a priori to the sender.

Fig 15 / Routing from a Query Proxy to high activity region



- The routing has to maximize information gain along the path. A path toward the high information region may be more preferable than the shortest path.
- Fig 16 Routing from a Query proxy to exit node.



Model Question Bank

PART A

1. What is localization?
2. Give the advantages of localization.
3. What is topology?
4. Give the various aspects of topology.
5. Mention the various types of topology.
6. What is clustering?
7. List various services offered by localization.
8. Why is topology control necessary for WSN?
9. What are advantages of clustering?
10. What is positioning?
11. What is time synchronization?

12. List out the various synchronization protocols.
13. What are task driven in sensor nodes?
14. What is information based tasking?
15. What is sensor tasking?
16. What is sensor control?
17. What is synchronization?
18. What is FTSP protocol?
19. What is triangulation?
20. What is proximity?
21. Mention the important characteristics of tri-lateration.
22. What is called active badge?

PART B

1. Describe the various aspects and options for Topology control in WSN with relevant example protocols.
2. Explain the concept of Localization and Positioning in detail.
3. Discuss in detail the various algorithms of Time Synchronization.
4. Explain the concept of Clustering, its advantages and algorithm for determining independent sets.
5. Write a short note on Sensor Tasking and Control.

Wireless Sensor Networks

Unit 5 / Sensor Network Platforms & Tools

Prepared By

Dr. S.Omkumar

Syllabus / Unit 5

SENSOR NETWORK PLATFORMS & TOOLS:

- Sensor Node Hardware – Berkeley Motes, Programming Challenges, Node-level software platforms, Node level Simulators, State-centric programming

Topic 1

Sensor Node Hardware – Berkeley Notes

Introduction

- A real-world sensor network application has to incorporate all elements subject to energy, bandwidth, computation, storage, and real-time constraints.
- This makes sensor network application development quite different from traditional distributed system development or database programming.
- A sensor network application can assume an always-on infrastructure that provides reliable services such as optimal routing, global directories, or service discovery.
- There are two types of programming for sensor networks, those carried out by end users and those performed by application developers.

- An end user may view a sensor network as a pool of data and interact with the network via queries.
- An application developer must provide end users of a sensor network with the capabilities of data acquisition, processing, and storage.

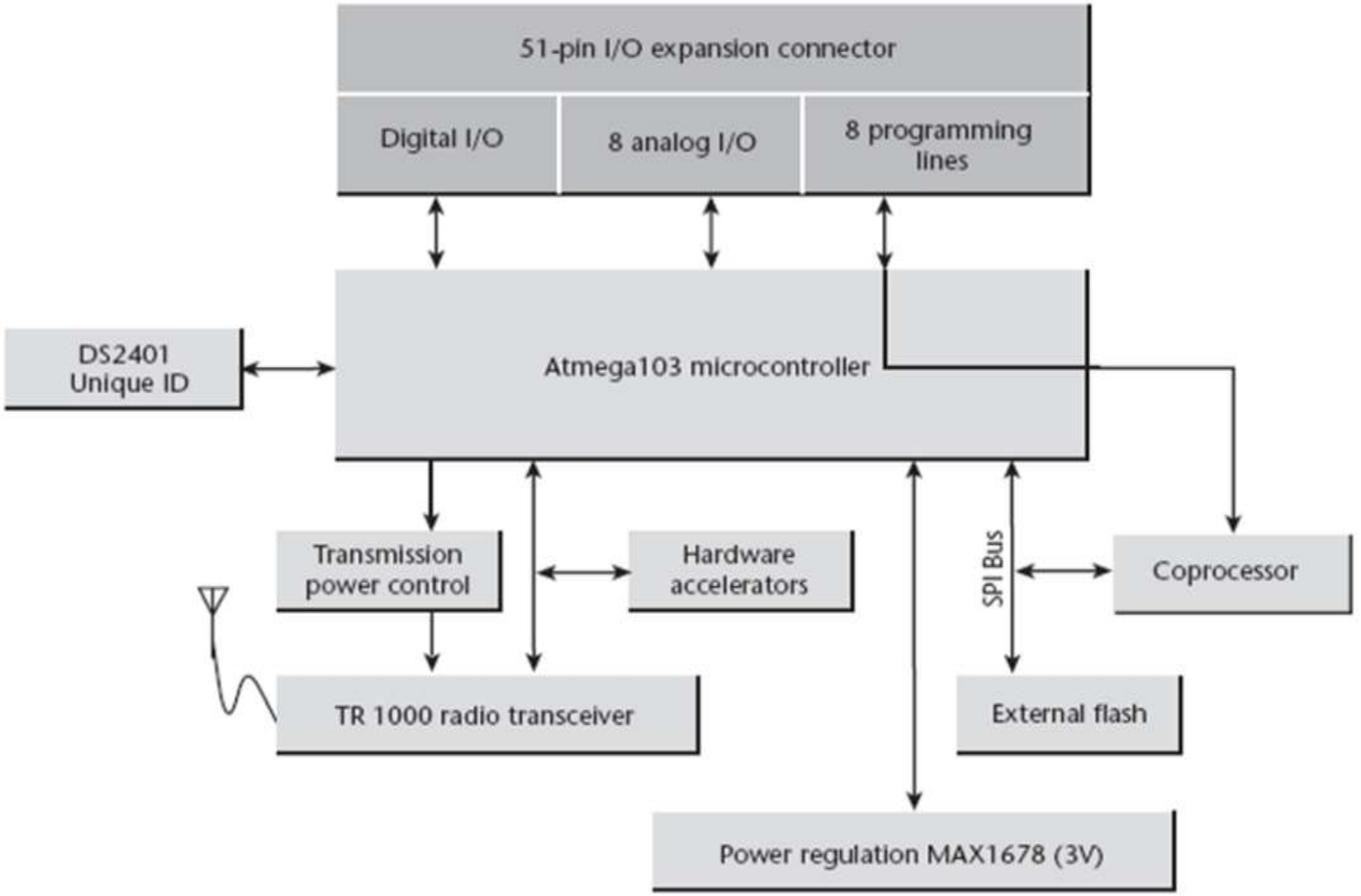
Sensor Node Hardware

- A typical sensor node hardware contains a general-purpose CPU and working memory, long-term stable storage such as flash memory or disk and I/O capabilities to support sensors.
- Sensor nodes have evolved into two broad categories:
  - Small devices with 8-bit microcontrollers as CPUs, 10–100 KB of working memory and 100–1000 KB of flash secondary storage
  - Larger devices with 32-bit CPUs, megabytes each of working memory and secondary storage

Small Devices (Berkeley Notes)

- The Berkeley motes are a family of embedded sensor nodes sharing roughly the same architecture. Let us take the MICA mote as an example.
- The MICA motes have a two-CPU design. The main microcontroller (MCU), an Atmel ATmega103L takes care of regular processing.
- A separate and less capable coprocessor is only active when the MCU is being reprogrammed.
- The ATmega103L MCU has integrated 512 KB flash memory and 4 KB of data memory.
- In addition to that, a MICA mote also has a separate 512 KB flash memory unit for holding data.

Fig 1 / MICA Mote Architecture



- Since the connection between the MCU and this external memory is via a low-speed serial peripheral interface (SPI) protocol, the external memory is more suited for storing data than for storing programs.
- The RF communication on MICA motes uses the TR1000 chip set operating at 916 MHz band.
- It can achieve a maximum of 50 kbps raw data rate. MICA motes implement a 40 kbps transmission rate.
- The maximum transmission range is about 300 feet in open space.
- MICA motes support a 51 pin I/O extension connector. Sensors, actuators, serial I/O boards, or parallel I/O boards can be connected via the connector

- A sensor/actuator board can host a temperature sensor, a light sensor, an accelerometer etc.
- The serial I/O connection allows the mote to communicate with a PC in real time.
- The parallel connection is primarily for downloading programs to the mote.
- A radio transmission bears the maximum power consumption.
- However, each radio packet (e.g., 30 bytes) only takes 4 ms to send, while listening to incoming packets turns the radio receiver ON all the time.
- There are huge differences among the power consumption levels in active mode, idle mode and suspend mode of the MCU.

Large Devices

- The larger class of devices is exemplified by products such as Stargate designed by Intel and Cerfcube from Intrinsyc.
- These devices are used in a variety of embedded applications.
- In the sensor network, they are used as gateways to a collection of motes, or for applications that require heavier-duty signal processing.
- Each such device employs an X-Scale or ARM-based processor which has 64MB working memory and 1GB flash-based secondary storage.
- They support many connectivity options including USB, IEEE 802.11 wireless and a 51-pin mote connector allowing use of a mote and its radio.

Topic 2

Programming Challenges

Introduction

- A sensor network differs from traditional computing environments in various aspects, thereby insisting programming frameworks and tools required for a sensor network.
- Specifically, the following characteristics affect the design of sensor network programming tools:
  - Reliability
  - Resource Utilization
  - Scalability
  - Data Centric Networks

Reliability

- Wireless sensor networks are more unreliable than other distributed systems.
- Therefore, sensor networks are built to adapt to changing dynamics and node & link errors such that network continues to serve until the network fails.
- While many faults in a network will never be noticed by application, resilience to failures and topology changes should be supported by a programming environment.

Resource Constraints

- Wireless sensor networks are very resource-constrained which affects the programming approach, maximum code size and other aspects of application development.
- Energy efficiency is particularly critical in WSNs and penetrates every aspect of sensor network design from duty cycles to routing protocols.
- Therefore, programming tools and models should be developed to exploit energy-saving techniques and approaches.

Scalability

- Sensor networks can scale up to hundreds and thousands of sensor nodes.
- Therefore programming models should support developers in designing applications and software for large-scale and heterogeneous networks.
- Manual configuration, maintenance and repair of individual sensor nodes will be infeasible due to the large number of devices, therefore insisting support for self-management and self-configuration.
- The scale of a network can also be addressed by using programming models that consider entire network as a single entity instead of focusing on each individual device.

Data Centric Networks

- In many wireless sensor networks, not only are the individual sensor nodes of interest, but also the data they generate and disseminate.
- Sensor network applications obtain useful information in a timely fashion. Many applications are only concerned with the collection of data at a central point.
- For example, a server that stores, analyzes, or visualizes the sensor data.
- Other applications require immediate processing and analysis of data within the network to eliminate redundant information, to aggregate data from multiple sensors etc.
- Each category will require different programming models so that programming a network results in generating distributed algorithms to work across many nodes in an efficient manner.

Topic 3

Node Level Software Platforms

Introduction

- The following are the node level software platforms used in wireless sensor networks –
  - TinyOS
  - NesC
  - TinyGALS
  - Sensor Network Application Construction Kit (Snack)
  - Thread Based Model

1. TinyOS

Introduction

- TinyOS is an embedded component-based operating system and platform for low-power wireless devices in wireless sensor networks (WSNs), 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 was collaboration between the University of California, Berkeley, Intel Research and Crossbow Technology.
- It was released as open-source software under BSD license and has grown into an international consortium, the TinyOS Alliance.
- TinyOS has been used in space, being implemented in ESTCube-1.

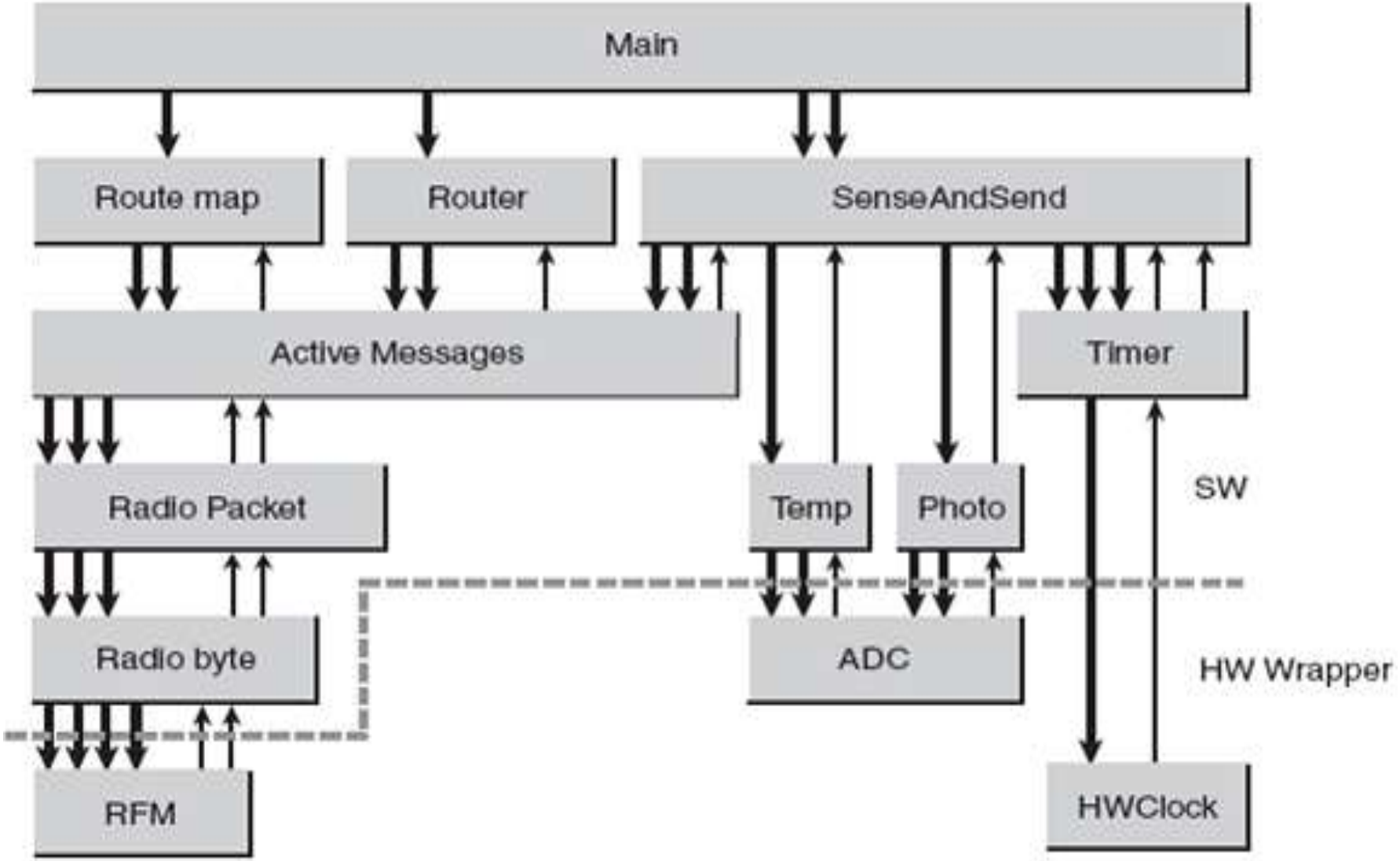
Features of TinyOS

- TinyOS applications are written in the programming language nesC a dialect of C language optimized for the memory limits of sensor networks.
- Its supplementary tools are mainly in the form of Java and shell script front-ends.
- The associated libraries and tools are mostly written in C.
- TinyOS programs are built of software components and they present hardware abstractions.
- Components are connected to each other using interfaces.
- TinyOS provides interfaces and components for common abstractions such as packet communication, routing, sensing, actuation and storage.

Characteristics of TinyOS

- TinyOS is fully non-blocking and it has one call stack.
- All I/O operations are asynchronous and have a callback.
- TinyOS uses nesC's features to link these callbacks called events to enable the native compiler for better optimization,
- TinyOS forces programmers to write complex logic by combining together many small event handlers.
- TinyOS provides tasks similar to a Deferred Procedure Call in order to support larger computations,.
- Tasks are non-preemptive and run in first in first out order.
- This simple concurrency model is sufficient for I/O centric applications but difficult with CPU-heavy applications.

Fig 2 / Field Monitor Example



Application Example

- Let us consider a Tiny OS application example—Field Monitor, where all nodes in a sensor field periodically send their temperature and photo sensor readings to a base station via an ad hoc routing mechanism.
- A diagram of the Field Monitor application is shown in Figure 5.2, where blocks represent TinyOS components and arrows represent function calls among them.
- The directions of the arrows are from callers.

2. NesC

Introduction

- The name nesC is an abbreviation of "network embedded systems C".
- NesC is a component-based event-driven programming language used to build applications for TinyOS platform.
- TinyOS is an operating environment designed to run on embedded devices used in distributed wireless sensor networks.
- NesC is built as an extension to the C programming language with components "wired" together to run applications on TinyOS.

Components

- NesC programs are built out of components assembled to form whole programs.
- Components have internal concurrency in the form of tasks.
- Threads of control may pass into a component through its interfaces. These threads are rooted either in a task or a hardware interrupt.
- Interfaces may be provided or used by components.
- The provided interfaces are intended to represent the functionality that the component provides to its users.
- The used interfaces represent the functionality the component needs to perform its job.

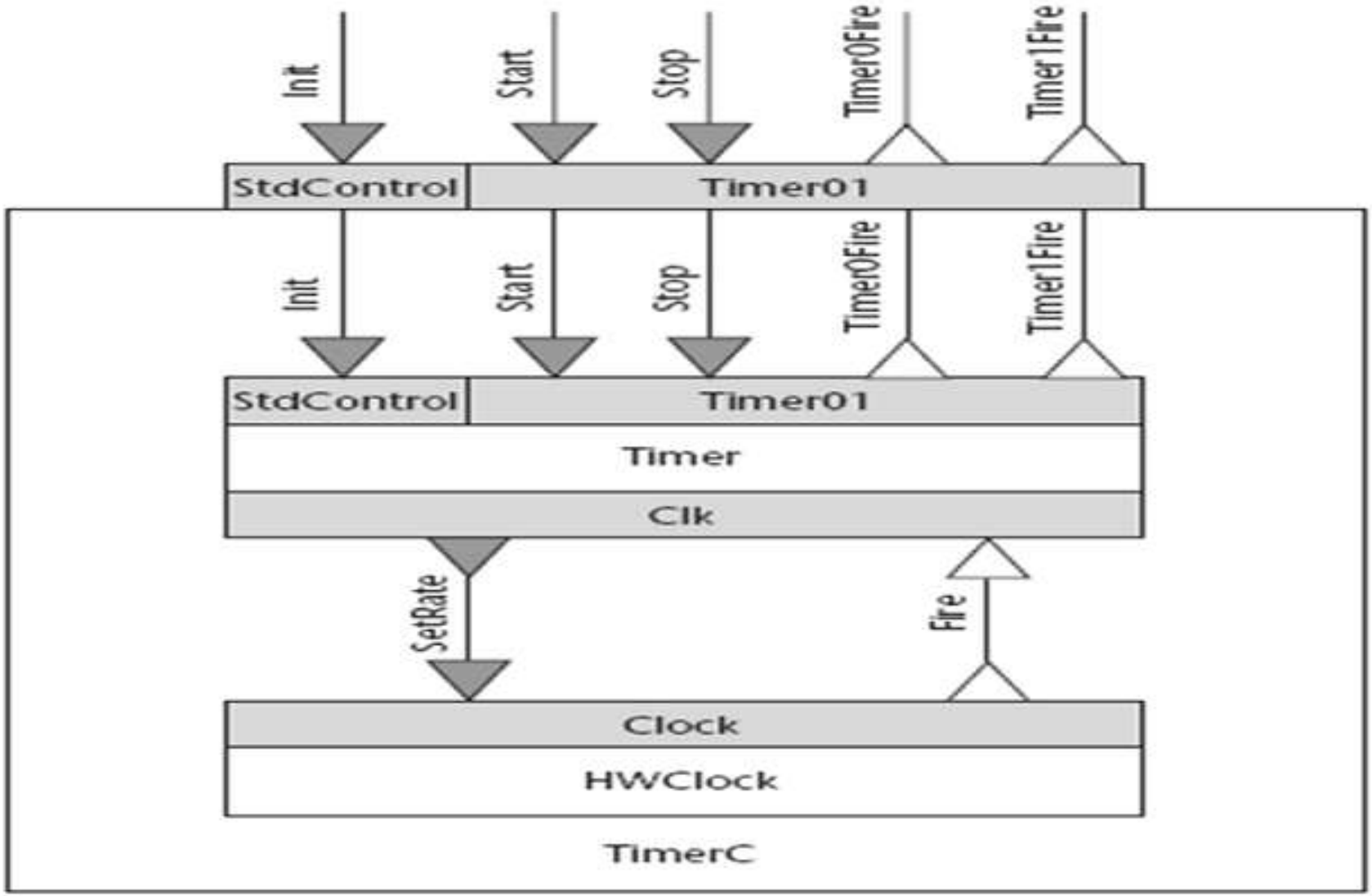
Interfaces

- In nesC, interfaces are bidirectional.
- They specify a set of functions to be implemented by the interface's provider (commands) and a set to be implemented by the interface's user (events).
- This allows a single interface to represent a complex interaction between components.
- This is critical because all lengthy commands in TinyOS are non-blocking and their completion is signaled through an event.
- Typically commands call downwards, i.e., from application components to those closer to the hardware, while events call upwards.
- Components are linked to each other via their interfaces.
- This increases runtime efficiency, encourages robust design, and allows for better static analysis of programs.

Timer Service

- In order to connect the Timer component and a hardware clock wrapper called HWC lock, a timer service called Timer C is provided as shown in Figure 5.3.
- The Timer interface defines two types of *commands*: start and stop.
- The Timer interface also defines an *event*, which is also a function.
- While commands are implemented by the providers of an interface, events are implemented by the users.
- Similarly, all other interfaces in this example define both commands and events.

Fig 3 / Timer C Configuration



Components Implementation

- *Modules* are components implemented by application code, while *configurations* are components implemented by connecting interfaces of existing components.
- Every nesC application has a *top-level configuration* that describes how components are wired together.
- Functions in nesC are described as (*f. i)* where *f* is a function in an interface *i*.
- Functions are invoked using the *call* operation (for commands) and the *signal* operation (for events).

Classification of Codes

- In TinyOS, code executes either asynchronously or synchronously.
- Race conditions can occur when concurrent updates to shared state are performed.
  - **Asynchronous code (AC):** Code reachable from at least one interrupt handler
  - **Synchronous code (SC)** : Code only reachable from tasks
- Synchronous code is always atomic to other synchronous codes because tasks are always executed sequentially without preemption.
- Race conditions are possible when shared state is modified from AC or when shared state from SC is also modified from AC.

- Therefore, nesC provides programmers with two options to ensure atomicity.
- The first option is to convert all of the sharing code to tasks.
- The second option is to use *atomic sections* to modify shared state.

3. TinyGALS

Introduction

- TinyGALS is a globally asynchronous and locally synchronous (GALS) approach for programming event-driven embedded systems.
- A TinyGALS program consists of modules, which are composed of components.
- A component C has a set of internal variables V_C, a set of external variables X_C and a set of methods I_C that operate on these variables.
- Methods are further divided into calls in the $ACCEPTS_C$ set (which can be called by other components) and calls in the $USES_C$ set (which are needed by C and may belong to other components).
- Similar to nesC and TinyOS, TinyGALS defines components using an interface definition and an implementation.

Composition of Modules

- TinyGALS modules consist of one or more components. A module 'M' is a 6-tuple given by –
- $M=(COMPONENTS_M, INIT_M, INPORTS_M, OUTPORTS_M, PARAMETERS_M, LINKS_M)$

  where -
  - $COMPONENTS_M$ is the set of components of M
  - $INIT_M$ - list of methods of M's components,
  - $INPORTS_M$ - inputs of module
  - $OUTPORTS_M$ - outputs of the module,
  - $PARAMETERS_M$ - a set of variables external to the components
  - $LINKS_M$ - relationships between the method call interfaces and I/O of the module.

- Modules are further connected to each other to form a complete TinyGALS system with a 5-tuple ‘S ‘given by –
- $S=(MODULES_S, GLOBALS_S, VAR\_MAPS_S, CONNECTIONS_S, START_S)$

  where -
  - $MODULES_S$ – a set of modules
  - GLOBALS – a set of global variables
  - VAR_MAPS_S – a set of mappings
  - $CONNECTIONS_S$ - a list of the connections between module output ports and input ports
  - $START_S$ - name of an input port of one module used as a starting point for the execution of the system.

TinyGUYS

- The architecture of TinyGALS can be used to automate the generation of scheduling and event handling code.
- By this, error-prone concurrency control code can be avoided.
- Code generation tools can automatically produce all necessary code for component links, module connections, system initialization etc.
- The use of message passing, modules in TinyGALS become decoupled from each other, therefore facilitating their independent development.
- TinyGALS provides another mechanism, called TinyGUYS (Guarded Yet Synchronous) variables, where modules may read global variables synchronously without delay.

4. Sensor Network Application Construction Kit (SNACK)

SNACK

- The Sensor Network Application Construction Kit (SNACK) is a configuration language component and service library and compiler for the development of sensor network applications.
- SNACK's goal is to provide *smart libraries* that can be combined to form sensor network applications.
- This will simplify the development process without losing control over efficiency.
- The SNACK library of components and services contains a variety of components for sensing, aggregation, transmission, routing, and data processing

5. Thread Based Model

Thread Based Model

- The thread-based paradigm is popular in many computing systems and now sensor networks.
- In traditional event-based systems, event handlers are executed in response to events and these handlers (tasks) run to completion without interruption from other tasks.
- The main advantage of the thread-based approach is multiple tasks can make progress in their execution indefinitely.
- For example, a task scheduler can execute a task for a certain amount of time; preempt this task in order to execute another task.
- This *time-slicing* approach simplifies the programming of sensor systems at the cost of increased operating system complexity

Topic 4

Node Level Simulators

Introduction

- Node-level design methodologies are usually associated with simulators that simulate the behavior of a sensor network on a per-node basis.
- Using simulation, designers can quickly study the performance in terms of timing, power, bandwidth, and scalability of potential algorithms without implementing them on actual hardware.
  - NS2
  - GloMoSim and QualNet
  - JiST/SWANS
  - OMNeT++
  - TOSSIM
  - EmStar
  - Avrora

NS2

- It was written in a combination of C++ and an object-oriented dialect of Tcl called OTcl.
- Many enhancements and extensions were developed to provide support for wireless networks and mobile adhoc networks.
- Similarly, a variety of extensions for sensor networks have been created. For example, one such extension adds the concept of a phenomenon to a sensor network simulation.
- The model uses broadcast packets transmitted through a designated channel to represent a phenomenon.
- Broadcast packets are generated using the PHENOM routing protocol, which emits packets with a certain configurable pulse rate

GloMosim & QualNet

- GloMoSim is a simulation tool based on the PARSEC simulation environment.
- Parallel Simulation Environment for Complex (PARSEC) systems is a C-based simulation language used to represent a set of objects in the physical system as logical processes and interactions among these objects as time-stamped message exchanges.
- GloMoSim supports a variety of models at different protocol layers such as CSMA & MACAW (MAC layer), flooding & DSR (network layer) and TCP & UDP (transport layer).

- In addition, it supports different node mobility models of the following -
- **Random waypoint model**: A node chooses a random destination within simulated area and moves toward this destination with a specified speed
- **Random drunken model:** A node periodically moves to a position chosen randomly from its immediate neighboring positions

Jist/SWANS

- JiST stands for Java in Simulation Time.
- The motivation behind JiST is to create discrete event simulations that execute efficiently and transparently.
- Efficiency refers to the ability to execute a given simulation program in parallel, while optimizing the configuration of the simulation across the available computational resources.
- Transparency refers to the ability to transform simulation programs automatically to run with simulation time semantics.
- The primary motivation for JiST was to support simulations of ad hoc networks and Scalable Wireless Ad hoc Network Simulator (SWANS).
- SWANS is a collection of independent software components aggregated to form complete wireless simulations.

OMNeT++

- The Objective Modular Network Testbed (OMNeT++) discrete event simulation environment is a tool used for the simulations of communication networks, Multiprocessors and various distributed systems.
- It is an open-source simulator based on C++ designed for the simulation of large systems and networks.
- A model in OMNeT++ consists of modules that communicate with each other using message passing.
- *Simple modules* can be grouped together to form more complex *compound modules*.
- A user defines the structure of a module using OMNeT++'s topology description language NED.
- The OMNeT++ framework includes a graphical editor used to edit network topologies either graphically or in NED source view.

TOSSIM

- A simulator for TinyOS-based wireless sensor networks is TOSSIM.
- It generates discrete event simulations directly from TinyOS components, running the same code on sensor nodes.
- TOSSIM replaces low-level components for translating hardware interrupts into events in the simulation that drive the TinyOS application.
- TOSSIM works at the bit level. This allows for experimentations with low-level protocols in addition to higher-level protocols or applications.
- Similar to most other tools, TOSSIM comes with a visualization tool called TinyViz.
- TOSSIM scales to thousands of sensor nodes and its advantages include scalability and extensibility.

EmStar

- EmStar is targeted at high capability nodes called *microservers.*
- These nodes in a hierarchical sensor network structure run more complex software than ordinary sensing devices.
- EmStar consists of a Linux microkernel extension, libraries, services, and several tools.
- EmSim operates many virtual nodes in parallel in a simulation that models radio and sensor channels.
- EmCee runs the EmSim core and is an interface to real low-power radios instead of using a modeled channel.
- EmView is a graphical visualizer for EmStar systems.

Avrora

- Avrora is a flexible simulator framework implemented in Java.
- Each node is implemented as its own thread and code is executed in an instruction-by instruction fashion.
- The key component is its implementation of an event queue.
- Many energy-conscious nodes tend to sleep for large periods of time where no instructions are executed and the energy consumption is reduced.
- The event queue in Avrora takes this advantage to boost the performance of the simulator.

Topic 5

State Centric Programming

Introduction

- Applications such as target tracking are not generic distributed programs.
- The vital parameters are the states of physical phenomena and models of their evolution over space and time.
- A distinct property of physical states such as location, shape and motion of objects is continuity in space and time.
- The sensing and control of these states can be handled through sequential state updates.

- The System theories provide the following state centric abstraction for state updating:
  - $x_{k+1} = f(x_k, u_k)$ (1)
  - $y_k = g(x_k, u_k)$ (2)
- where ‘x’ is the system state, ‘k’ is an integer update index over time, ‘u’ is input, ‘y’ is output, ‘f’ is the state update function and ‘g’ is the output or observation function
- This formalization is broad enough to capture a wide variety of algorithms in sensor fusion, signal processing and control.
- State-centric programming abstractions have been successfully applied to synchronous VLSI circuit designs and control system designs.

Domain Specific Run Time Systems

- Synchronous languages such as Signal & Esterel and mixed-signal visual languages such as Matlab's Simulink & Ptolemy are all examples of state-centric programming models.
- However in a distributed real-time embedded system the formulation is not cleanly represented.
- The relationship among subsystems can be highly dynamic. The following concerns must be addressed –
  - The storage of state variables
  - Origin of inputs
  - Destination of outputs
  - Evaluation of functions 'g' and 'f'
  - Time taken to acquire the set of inputs
  - The correct order of arrival of inputs
  - The choice of update interval

- System designers cannot be shielded from these issues without compromising system correctness and efficiency.
- These concerns must be addressed to perform sensing, computation, actuation and play a central role in achieving the overall system performance.
- However, traditional programming models and languages don't support these "nonfunctional" aspects of computation. The novel design methodologies and frameworks are required to provide meaningful abstractions for these issues.
- Domain-specific runtime systems are used to support this design methodology to ensure efficient execution and allow transparent features such as security and reliable communication.

Collaborative Groups

- A group is a set of entities that contribute to a state update.
- These entities can be physical or logical sensor nodes or more abstract system components such as software agents.
- A group encapsulates two properties – its scope and structure. A group's scope defines its members.
- The scope can be specified by a membership function.
- Grouping nodes according to physical attributes rather than node addresses is an important abstraction in sensor network programming.
- The scope can be evaluated locally or dynamically, as long as communication among the group members is maintained.

Examples for Groups

1. Geographically Constrained Group (GCG)

- GCG consists of members within a pre-specified geographical extent.
- Since physical signals from point targets, may propagate only to a limited extent in an environment, this kind of group represents all the sensor nodes that can possibly "sense" a phenomenon.
- There are many ways to specify the geographic shape, such as circles, polygons, and their unions and intersections.
- A GCG can be easily established by geographically constrained flooding.

2. N-hop Neighborhood Group (n-HNG)

- When the communication topology is more important than the geographical extent, hop counts are useful to constrain group membership.
- An n-hop neighborhood group has an anchor node and defines that all nodes within n communication hops are members of the group.
- Since it uses hop counts rather than Euclidean distances, local broadcasting can be used to determine the scope.
- The anchor node is the leader of the group and the group may have a tree structure with the leader as the root to optimize for communication.

3. Publish/Subscribe Group (PSG)

- A publish/subscribe group comprises consumers expressing interest in specific types of data or services and producers that provide those data or services.
- Communication among members of a PSG may be established via rendezvous points, directory servers, or network protocols such as directed diffusion.

4. Acquaintance Group

- In this group, a member belongs to the group because it was “invited” by another member in the group.
- The relationships among the members may not depend on any physical properties at the current time.
- A member may also quit the group without requiring permission from any other member.
- An AG may have a leader, serving as the rendezvous point.
- When the leader is also fixed on a node, GPSR, ad hoc routing trees may facilitate the communication between the leader and the other members.
- The important use of this group is to monitor and control mobile agents from a base station

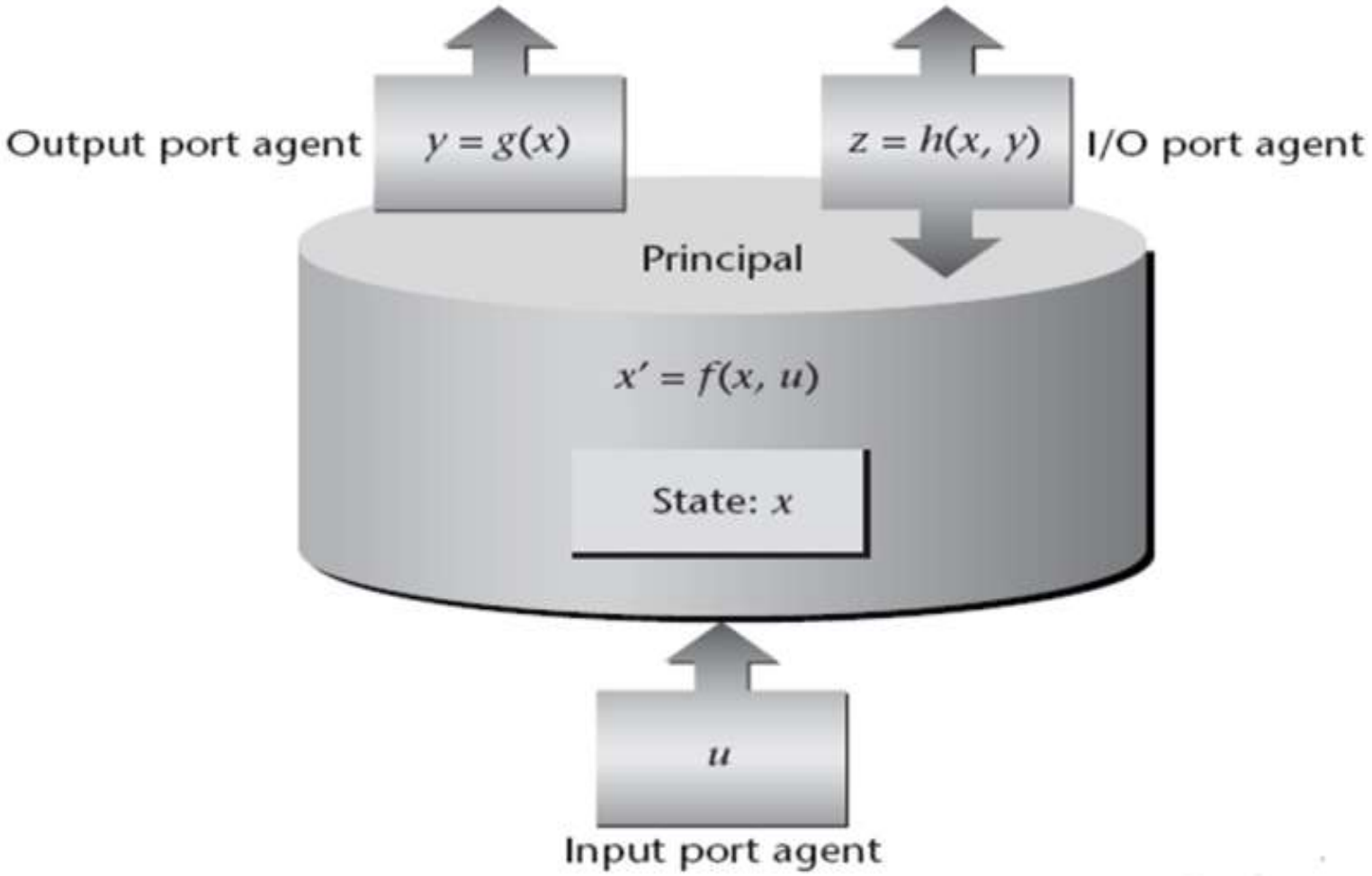
5. Using Multiple types of Groups

- Mixing and matching groups is a powerful technique for tackling system complexity by making algorithms much more scalable and resource efficient without sacrificing conceptual clarity.
- The highly tuned communication protocols can be used for specific groups to reduce latency and energy costs.
- There are various ways to compose groups.
- They can be composed in parallel to provide different types of input for a single computational entity.

State Centric Design Framework (PIECES)

- Programming and Interaction Environment for Collaborative Embedded Systems (PIECES) is a software framework that implements the methodology of state-centric programming over collaboration groups to support the modeling, simulation and design of sensor network applications.
- It is implemented in a mixed Java-Matlab environment. PIECES comprise principals and port agents.
- Figure 5.4 shows the basic relations among principals and port agents.

Fig 4 / Principals & Port Agents



- The features of a Principal are as follows-
  - The key component for maintaining a piece of state.
  - To update its state from time to time, a computation corresponding to evaluating function ‘f’.
  - Accepts other principals’ queries of certain views on its own state, a computation corresponding to evaluating function ‘g’.
  - Gather information from other principals for updating
  - Creates port agents and attaches them onto it and onto the other principals.

- The features of a port are as follows-
  - May be an input, an output, or both.
  - Also called an observer, computes outputs based on the host principal's state and sends them to other agents.
  - May be active or passive.
  - Active observer pushes data autonomously to its destination, while a passive observer sends data only when a consumer requests it.
  - Capture communication patterns among principals.
- The execution of principals and port agents can be either time driven or event-driven where events may include physical events that are pushed to them or query events from other principals or agents.

Multi-target Tracking System

- Tracking of two crossing targets can be decomposed into three phases:
  - When the targets are far apart, the tracking problem can be treated as a set of single-target tracking sub-problems.
  - When the targets are in proximity of each other, they are tracked jointly due to signal mixing.
  - After the targets move apart, the tracking problem becomes two single-target tracking sub-problems again.

Model Question Bank

PART A

1. What is sensor node hardware?
2. What is TinyOS?
3. Where is TinyOS used?
4. Classify the sensor node hardware.
5. Define Berkley notes.
6. What do you mean by node level simulation?
7. What are the programming challenges for sensor networks?
8. What are the different platforms available for sensor networks?
9. What is node level software platform?
10. What is a node level simulator?

11. What is centric programming?
12. List the major concern of sensor node hardware.
13. What is TINYGALS?
14. Highlight the salient feature of component-based operating system.
15. What is PIECES?
16. Give the features of a principal.
17. Give the features of a port.
18. Give examples for collaborative groups.
19. Differentiate between configurations and modules in TinyOS.
20. Mention the classification of code in TinyOS.

PART B

1. Write short notes on (i) Berkley Notes (ii) Programming Challenges.
2. Explain the various Node level software platforms available for sensor networks.
3. Explain the various Node level simulators available for sensor networks.
4. Discuss the concept of centric programming and its collaborative groups with relevant examples.