MRCET CAMPUS

# MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY

(AUTONOMOUS INSTITUTION – UGC, GOVT. OF INDIA)


## Department of CSE

## (Emerging Technologies)

## (INTERNET OF THINGS)

**B.TECH(R-20 Regulation)**

**(III YEAR – I SEM)**

**(2022-23)**

# WIRELESS SENSOR NETWORKS

## (R20A0465)

# LECTURE NOTES

**Prepared by**

**Mrs. Vijaya Bharathi**

**MALLA REDDY COLLEGE OF ENGINEERING & TECHNOLOGY**

**(Autonomous Institution – UGC, Govt. of India)**

Recognized under 2(f) and 12(B) of UGC ACT 1956

(Affiliated to JNTUH, Hyderabad, Approved by AICTE-Accredited by NBA & NAAC – 'A' Grade - ISO 9001:2015 Certified)

Maisammaguda, Dhulapally (Post Via. Hakimpet), Secunderabad–500100, Telangana State, India

## Department of Computer Science and Engineering

# EMERGING TECHNOLOGIES

## Vision

- "To be at the forefront of Emerging Technologies and to evolve as a Centre of Excellence in Research, Learning and Consultancy to foster the students into globally competent professionals useful to the Society."

## Mission

***The department of CSE (Emerging Technologies) is committed to:***

- To offer highest Professional and Academic Standards in terms of Personal growth and satisfaction.
- Make the society as the hub of emerging technologies and thereby capture opportunities in new age technologies.
- To create a benchmark in the areas of Research, Education and Public Outreach.
- To provide students a platform where independent learning and scientific study are encouraged with emphasis on latest engineering techniques.

## QUALITY POLICY

- To pursue continual improvement of teaching learning process of Undergraduate and Post Graduate programs in Engineering & Management vigorously.
- To provide state of art infrastructure and expertise to impart the quality education and research environment to students for a complete learning experiences.
- Developing students with a disciplined and integrated personality.
- To offer quality relevant and cost effective programmes to produce engineers as per requirements of the industry need.

**For more information: www.mrcet.ac.in**

*MALLAREDDYCOLLEGEOFENGINEERINGAND TECHNOLOGY*

IIIYearB.TechCSE(IOT)

ISemester

WIRELESS SENSOR NETWORKS

(R20A0465)

OBJECTIVES:

1. To understand the characteristics, basic concepts in Wireless sensor networks
2. To understand the architecture framework and goals of WSN
3. To understand the Medium Access control and its protocols
4. To understand the routing metrics and network layer protocols
5. To understand the QoS based routing protocols

UNIT - I:

Introduction: Components of a wireless sensor node, Motivation for a Network of Wireless Sensor Nodes, Classification of sensor networks, Characteristics of wireless sensor networks, Challenges of wireless sensor networks, Comparison between wireless sensor networks and wireless mesh networks, Limitations in wireless sensor networks, Design challenges, Hardware architecture.

UNIT - II:

Basic Architectural Frame-work:-Physical Layer, Basic Components, Source Encoding, Channel Encoding, Modulation. Network Architecture - Sensor Network Scenarios, Optimization Goals and Figures of Merit, Gateway Concepts.

UNIT - III:

Medium Access Control: : Wireless MAC Protocols, Characteristics of MAC Protocols in Sensor Networks, Contention-Free MAC

Protocols, Contention-Based MAC Protocols, and Hybrid MAC Protocols. Location discovery, quality, other issues, S-MAC, IEEE 802.15.4.

UNIT - IV:

Network Layer: Routing Metrics, Flooding and Gossiping, Data-Centric Routing, Proactive Routing, On-Demand Routing, Hierarchical Routing, Location-Based Routing.

UNIT - V:

QoS-Based Routing Protocols:Node and Network Management: Power Management, Local Power Management aspects, Dynamic Power Management, Conceptual Architecture

TEXTBOOKS:

1. WaltenegusDargie, Christian Poellabauer, "Fundamentals of Wireless Sensor Networks: Theory and Practice", Wiley 2010
2. Mohammad S. Obaidat, Sudip Misra, "Principles of Wireless Sensor Networks", Cambridge, 2014
3. LoWPAN: The Wireless Embedded Internet, Zach Shelby, Carsten Bormann, Wiley
4. Internet of Things: Converging Technologies for Smart Environments and Integrated Ecosystems, Dr. Ovidiu Vermesan, Dr. Peter Friess, River Publishers

REFERENCE BOOKS:

1. Ian F. Akyildiz, Mehmet Can Vuran , "Wireless Sensor Networks", Wiley 2010
2. C S Raghavendra, K M Sivalingam, TaiebZnati, "Wireless Sensor Networks", Springer, 2010
3. C. Sivarmmurthy& B.S. Manoj, "Adhoc Wireless Networks", PHI-2004
4. FEI HU., XIAOJUN CAO, "Wireless Sensor Networks", CRC Press, 2013
5. Feng ZHAO, Leonidas GUIBAS, " Wireless Sensor Networks", ELSEVIER , 2004

6. The Internet of Things: From RFID to the Next-Generation Pervasive Networked Lu Yan, Yan Zhang, Laurence T. Yang, Huansheng Ning
7. Internet of Things (A Hands-on-Approach) , Vijay Madisetti , ArshdeepBahga
8. Designing the Internet of Things , Adrian McEwen (Author), HakimCassimally

COURSE OUTCOMES:

1. Understand the challenges, design goals and architecture of Wireless sensor networks
2. Understand the channel encoding and modulation mechanism.
3. Understand the contention free and contention based MAC protocols.
4. Understand the routing metrics and protocols of Network layer
5. Understand the QoS based routing protocols

## UNIT-I
## Introductionto Wireless Sensor Networks

### Introduction

Wireless Sensor Networks (WSNs) can be defined as a self-configured and infrastructure-less wireless networks to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants and to cooperatively pass their data through the network to a main location or sink where the data can be observed and analysed .A sink or base station acts like an interface between users and the network. One can retrieve required information from the network by injecting queries and gathering results from the sink. Typically a wireless sensor network contains hundreds of thousands of sensor nodes. The sensor nodes can communicate among themselves using radio signals. A wireless sensor node is equipped with sensing and computing devices, radio transceivers and power components.

A Wireless Sensor Network (WSN) is a distributed network and it comprises a large number of distributed, self-directed, tiny, low powered devices called sensor nodes alias motes . WSN naturally encompasses a large number of spatially dispersed, petite, battery-operated, embedded devices that are networked to supportively collect, process, and convey data to the users, and it has restricted computing and processing capabilities. Motes are the small computers, which work collectively to form the networks. Motes are energy efficient, multi-functional wireless device . The necessities for motes in industrial applications are widespread. A group of motes collects the information from the environment to accomplish particular application objectives. They make links with each other in different configurations to get the maximum performance. Motes communicate with each other using transceivers. In WSN the number of sensor nodes can be in the order of hundreds or even thousands. In comparison with sensor networks, Ad Hoc networks will have less number of nodes without any infrastructure.

Now a days wireless network is the most popular services utilized in industrial and commercial applications, because of its technical advancement in processor, communication, and usage of low power embedded computing devices. Sensor nodes are used to monitor environmental conditions like temperature, pressure, humidity, sound, vibration, position etc. In many real time applications the sensor nodes are performing different tasks like neighbor node discovery, smart sensing, data storage and processingdata aggregation, target tracking, control and monitoring, node localization synchronization and efficient routing between nodes and base station.

- **FEATURES**

- A Wireless Sensor Network (WSN) is a distributed network
- A network configuration where every participant can communicate with one another without going through a centralized point.
- WSN comprises a large number of distributed, self-directed, tiny, low powered devices called sensor nodes
- WSN has embedded devices that are networked to supportively collect, process, and convey data to the users.
- WSN has restricted computing and processing capabilities.
- WSN can be defined as a self-configured and infrastructure-less wireless networks
- It is used to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants
- It pass their data through the network to a main location or sink where the data can be observed and analysed
- A sink or base station acts like an interface between users and the network.
- One can retrieve required information from the network by injecting queries and gathering results from the sink.
- A wireless sensor network contains hundreds of thousands of sensor nodes.
- The sensor nodes can communicate among themselves using radio signals.
- A wireless sensor node is equipped with sensing and computing devices, radio transceivers and power components.
- The individual nodes in a wireless sensor network (WSN) are inherently resource constrained.
- They have limited
  - processing speed
  - storage capacity
  - communication bandwidth
- After the sensor nodes are deployed, they are responsible for self-organizing an appropriate network infrastructure often with multi-hop communication with them.
- Then the onboard sensors start collecting information of interest
- Wireless sensor devices also respond to queries sent from a “control site” to perform specific instructions or provide sensing samples.
- The working mode of the sensor nodes may be either continuous or event driven.
- Global Positioning System (GPS) and local positioning algorithms can be used to obtain location and positioning information
-

**Why use a WSN?**

- Ease of deployment
  - ❖ Wireless communication means no need for a communication infrastructure setup
  - ❖ Drop and play
  - ❖ Low-cost of deployment
  - ❖ Nodes are built using off-the-shelf cheap components
- Fine grain monitoring
  - ❖ Feasible to deploy nodes densely for fine grain monitoring
  - ❖

**Applications of wireless sensor networ**k

1. Military applications
2. Area monitoring
3. Transportation
4. Health applications
5. Environmental sensing
6. Structural monitoring
7. Industrial monitoring
8. Agricultural sector
9. Neighbor node discovery
10. Smart sensing
11. Data storage and processing
12. Data aggregation
13. Target tracking
14. Control and monitoring
15. Node localization
16. Synchronization
17. Efficient routing between nodes and base stationBattlefield Monitoring
18. Intelligent Guiding
19. Remote Sensing
20. Sniper Detection
21. Environmental
22. Habitat Monitoring Air or Water Quality Monitoring Hazard
23. Monitoring Disaster
24. Monitoring Health
25. Care Behavior Monitoring
26. Medical Monitoring Home
27. Intelligence Smart
28. Home Remote Metering
29. Industrial Process
30. Control Security and Surveillance

**Military applications**

- ❖ military command
- ❖ control
- ❖ communications
- ❖ computing
- ❖ Intelligence
- ❖ battlefield surveillance
- ❖ reconnaissance
- ❖ targeting systems

**Area Monitoring**

- In area monitoring, the sensor nodes are deployed over a region where some phenomenon is to be monitored.
- When the sensors detect the event being monitored (heat, pressure etc), the event is reported to one of the base stations, which then takes appropriate action will be taken.

**Transportation**

- Real-time traffic information is being collected to feed transportation models and alert drivers of congestion and traffic problems

**Health applications**

- Some of the health applications for sensor networks are
- ❖ supporting interfaces for the disabled patients
- ❖ integrated patient monitoring
- ❖ diagnostics,
- ❖ drug administration in hospitals
- ❖ tele-monitoring of human physiological data
- ❖ tracking & monitoring doctors or patients inside a hospital.

**Environmental sensing**

- This includes sensing volcanoes
- ❖ Oceans
- ❖ glaciers
- ❖ forests etc.
- Air pollution monitoring
- Forest fires detection
- Greenhouse monitoring
- Landslide detection

**Structural monitoring:**

- Wireless sensors can be utilized to monitor the movement within buildings
- to monitor infrastructure such as bridges, flyovers, embankments, tunnels etc
- Enabling Engineering practices to monitor assets remotely

**Industrial monitoring**

- Wireless sensor networks have been developed for machinery condition-based maintenance (CBM) as they offer significant cost savings and enable new functionalities.

**Agricultural sector**

- using a wireless network frees the farmer from the maintenance of wiring in a difficult environment.
- Irrigation automation enables more efficient water use and reduces waste

Components of a wireless sensor network :

The components of WSN system are sensor node, rely node, actor node, cluster head, gateway and base station.

**SENSOR NODES**

- Sensor nodes are used to monitor environmental conditions like temperature, pressure, humidity, sound, vibration, position etc.
- Each and every node is capable to perform data gathering, sensing, processing and communicating with other nodes.
- The sensing unit senses the environment
- The processing unit computes the confined permutations of the sensed data
- The communication unit performs exchange of processed information among neighboring sensor nodes.
- Mostly ATMEGA 16, ATMEGA 128L, MSP 430 controllers are used in commercial motes.
- It is Capable of

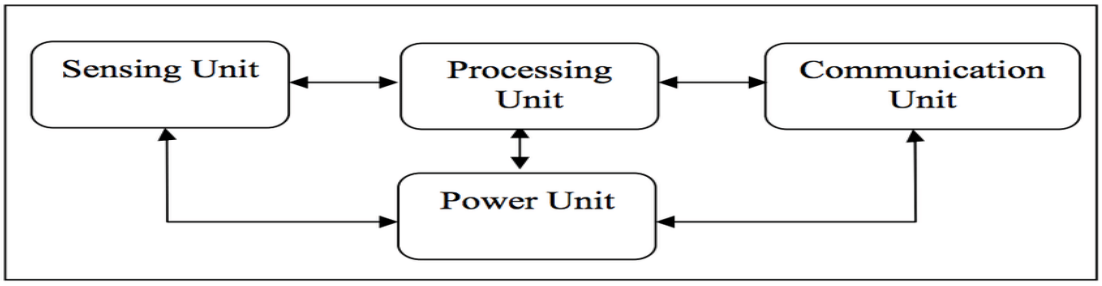
❖ executing data processing
❖ data gathering
❖ communicating with additional associated nodes in the network

- In WSN, based on the sensing range and environment, the sensor nodes are

classified into four groups, namely specialized sensing node, generic sensing node, highbandwidth sensing node and gateway node.

- The radio bandwidth for the sensor nodes are<50 Kbps, <100 Kbps, ≈500 Kbps and >500 Kbps
- Sensing is a technique used to gather information about a physical object or process, including the occurrence of events (i.e., changes in state such as a drop in temperature or pressure.
- An object performing such a sensing task is called a sensor.
- remote sensors, that is, they do not need to touch the monitored object to gather information

- **BLOCK DIAGRAM**




- sensor is a device that translates parameters or events in the physical world into signals that can be measured and analyzed.
- A transducer is often used to describe a device that converts energy from one form into another
- A sensor is a type of transducer that converts energy in the physical world into electrical energy that can be passed to a computing system or controller.
- Phenomena in the physical world (often referred to as process, system, or plant) are observed by a sensor device.
- The resulting electrical signals are often not ready for immediate processing, therefore they pass through a signal conditioning stage.
- A variety of operations can be applied to the sensor signal to prepare it for further use.
- For example, signals often require amplification (or attenuation) to change the signal magnitude to better match the range of the following analog-to-digital conversion
- signal conditioning often applies filters to the signal to remove unwanted noise within certain frequency ranges
- High pass filters can be used to remove 50 or 60Hz noise picked up by surrounding power lines
- After conditioning, the analog signal is transformed into a digital signal using an analog-to-digital converter (ADC).
- The signal is now available in a digital form and ready for further processing, storing, or visualization.
- An increasing number of sensors communicate the collected data wirelessly to a centralized processing station.
- This is important since many network applications require hundreds or thousands of sensor nodes, often deployed in remote and inaccessible areas.
- A wireless sensor has not only a sensing component, but also on-board processing, communication, and storage capabilities.
- The sensors should be utilized in a way that produces the maximum performance with less energy

Sensor node: Capable of executing data processing, data gathering and communicating with additional associated nodes in the network. A distinctive

sensor node capability is about 4-8 MHz, having 4 KB of RAM, 128 KB flash and preferably 916 MHz of radio frequency

Components of a wireless sensor node

- Components of a wireless sensor node
- ❖ sensing unit
- ❖ processing unit
- ❖ communication unit
- ❖ power unit

**sensing unit**

- The sensing unit of sensor nodes integrates different types of sensors like thermal sensors, magnetic sensors, vibration sensors, chemical sensors, bio sensors, and light sensors.
- The measured parameters from the external environment by sensing unit of sensor node are fed into the processing unit.
- The analog signal generated by the sensors are digitized by using Analog to Digital converter (ADC) and sent to controller for further processing.

**processing unit**

- The processing unit is the important core unit of the sensor node.
- The processor executes different tasks and controls the functionality of other components.
- The required services for the processing unit are pre-programmed and loaded into the processor of sensor nodes
- The energy utilization rate of the processor varies depending upon the functionality of the nodes.

**communication unit**

- In communication unit, a common transceiver act as a communication unit
- it is mainly used to transmit and receive the information among the nodes and base station and vice versa.
- There are four states in the communication unit:
- ❖ 1.transmit
- ❖ 2. receive
- ❖ 3. idle
- ❖ 4. sleep

**Sensor Nodes Functions**

- ➢ Effortlessness installation
- ➢ Fault indication,
- ➢ Energy level diagnosis

- Highly reliability
- Easy coordination with other nodes in the network
- Control protocols
- simple network interfaces with other smart devices

- sensor node is often not only responsible for data collection, but also for in-network analysis, correlation
- Sensor nodes communicate not only with each other but also with a base station (BS) using their wireless radios, allowing them to disseminate their sensor data to remote processing, visualization, analysis, and storage systems
- simple sensor nodes may monitor a single physical phenomenon

while more complex devices may combine many different sensing techniques (e.g., acoustic, optical,magnetic).

- They can also differ in their communication capabilities
- Eg. ultrasound, infrared, or radio frequency technologies with varying data rates and latencies.
- While simple sensors may only collect and communicate information about the observed environment.
- Global Positioning System (GPS) receivers, allowing them to accurately determine their position.
- More powerful devices (i.e., devices with large processing, energy, and storage capacities) may also perform extensive processing and aggregation functions .
- Such devices may form communication backbones that can be used by other resource-constrained sensor devices to reach the base station.
- For interconnectivity functions high end smart bandwidth sensing node and gateway nodes are preferred.
- Senor nodes in an open environment regularly sense the physical and environmental changes and transmit the information to the centralized server called a gateway.
- The number of sensor nodes in a sensor network can be in the order of hundreds or even thousands.
- Hence, WSN designed for sensor networks is supposed to be highly scalable.
- Mobility of nodes--In order to increase the communication efficiency, the nodes can move anywhere within the sensor field based on the type of applications.
- The option for reprogramming or reconfiguring should be available for the WSN to become adaptive for any dynamic changes in the network.
- The limited computation and power resources of sensor nodes often make it undesirable to use public key algorithms.
- Sensor node (mote)

- Sensor node topology
- Connection between sensor nodes follows some standard topology.
- The WSN should have the capability to work in the dynamic topology.
- If any node in the WSN fails to exchange data with other nodes, it should be informed without delay to the base station or gateway node
- The WSN should have the capability to work without any central control point

**Sensor Node Special Features**

- Technical advancement in processor
- Communication
- Usage of low power embedded computing devices.
- Sensor nodes are used to monitor environmental conditions like temperature, pressure, humidity, sound, vibration, position etc.

**RELAY NODE**

- It is a midway node used to communicate with the adjacent node. It is used to enhance the network reliability.
- A rely node is a special type of field device that does not have process sensor or control equipment and as such does not interface with the 7 process itself.
- A distinctive rely node processor speed is about 8 MHz, having 8 KB of RAM, 128 KB flash and preferably 916 MHz of radio frequency.

**ACTOR NODE**

- It is a high end node used to perform and construct a decision depending upon the application requirements.

- Typically these nodes are resource rich devices which are outfitted with high quality processing capabilities, greater transmission powers and greater battery life.
- A distinctive actor node processor capability is about 8 MHz, having 16 KB of RAM, 128 KB flash and preferably 916 MHz of radio frequency.

**CLUSTER HEAD**

- It is a high bandwidth sensing node used to perform data fusion and data aggregation functions in WSN.
- Based on the system requirements and applications, there will be more than one cluster head inside the cluster.
- A distinctive cluster head processor is about 4-8 MHz, having 512 KB of RAM, 4 MB flash and preferably 2.4 GHz of radio frequency
- This node assumed to be highly reliable, secure and is trusted by all the nodes in the sensor network.

**GATEWAY**

- Gateway is an interface between sensor networks and outside networks.
- Compared with the sensor node and cluster head the gateway node is most powerful in terms of program and data memory, the processor used, transceiver range and the possibility of expansion through external memory.
- A distinctive gateway processor speed is about 16 MHz, having 512 KB of RAM, 32 MB flash and preferably 2.4 GHz of radio frequency.

**BASE STATION**

- It is an extraordinary type of nodes having high computational energy and processing capability

**Motivation for a Network of Wireless Sensor Nodes**

The recent developments in engineering, communication and networking has led to new sensor designs, information technologies and wireless systems. Such advanced sensors can be used as a bridge between the physical world with the digital world.Sensors are used in numerous devices, industries, machines and environment and help in avoiding infrastructure failures, accidents, conserving natural resources, preserving wildlife, increase productivity, provide security etc .The use of distributed sensor network or system has also been contributed by the technological advances in VLSI, MEMS and Wireless Communication.. This miniaturization of processing, computing and sensing technologies has led to tiny, low-power and cheap sensors, controllers and actuators.

**CLASSIFICATIONS OF WSN**

Classification of sensor networks is given below.

1) Static and Mobile Network
2) Deterministic and Nondeterministic Network
3) Single base station and multi base station network

4) Single - Sink and Multi-sink Network
5) Single - Hop and Multi-hop Network
6) Self - Reconfigurable and Non - Self - Configurable Network
7) Homogeneous and Heterogeneous Network

**Static and Mobile Network**

- In many applications, all the sensor nodes are fixed without movement and these are static networks.
- Especially in biological systems, require mobile sensor nodes. These are known as mobile networks.
- An example of mobile network is animal monitoring.

**Deterministic and Nondeterministic Network**

- In a deterministic WSN, the position of a sensor node is calculated and fixed.
- The pre-planned deployment of sensor nodes is possible in only a limited number of applications.
- Determining the position of sensor nodes is not possible due to several factors like harsh environment or hostile operating conditions. Such networks are nondeterministic and require a complex control system

Single base station and multi base station network

- In a single base station WSN, only a single base station is used which is located close to the sensor node region. All the sensor nodes communicate with this base station, in case of a multi base station WSN, more than base station is used and a sensor node can transfer data to the closest base station

**Static Base Station & Mobile Base Station WSN**

- A static base station has a fixed position usually close to the sensing region.
- A mobile base station moves around the sensing region so that the load of sensor nodes is balanced.

**Single-hop and Multi-hop WSN**

- In a single-hop WSN, the sensor nodes are directly connected to the base station.
- In case of multi-hop WSN, peer nodes and cluster heads are used to relay the data so that energy consumption is reduced

**Self – Reconfigurable & Non – Self – Configurable WSN**

- In most WSNs, the sensor nodes are capable of organizing and maintaining the connection and work collaboratively with other sensor nodes to accomplish the task.
- In a non – Self – Configurable WSN, the sensor networks cannot organize themselves in a network and cannot rely on a control unit to collect information.

**Homogeneous and Heterogeneous WSN**

- In a homogeneous WSN, all the sensor nodes have similar energy consumption, computational power and storage capabilities.
- Homogeneous is a single network architecture & operating system.EG. ETHERNET LAN
- Hetero-genous is a connecting computers with different OS and protocols.
- hetero-genous WSN, some sensor nodes have higher computational power and energy.
- In homogenous sensor networks, all sensor nodes have the same property in terms of computation, communication, memory, energy level and reliability.
- In heterogeneous sensor networks, the nodes are of different capabilities in terms of computation, communication, memory, energy level and reliability.

- If a sensor network is deployed via random distribution, the protocols will not be aware of the communication status between each nodes after deployment.

**Characteristics of wireless sensor networks**

The concept of wireless sensor networks implies a number of WSN characteristics which heavily influence the software architecture

1) Fault tolerance
2) Mobility of nodes
3) Dynamic network topology
4) Communication failures
5) Heterogeneity of nodes
6) Scalability
7) Independency
8) Programmability
9) Utilization of sensors
10) Impracticality of public key cryptosystems
11) Lack of a prior knowledge of post-deployment configuration
12) self organizing
13) perform cooperative processing
14) Energy efficiency
15) modular.
16)

1.**Fault tolerance:**

- Each node in the network is prone to unanticipated failure.

- Faulttolerance is the capability to maintain sensor network functionalities without anybreak due to sensor node failures.
- Fault Tolerance Some sensor nodes may fail or be blocked due to lack of power, or have physical damage or environmental interference.
- The failure of sensor nodes should not affect the overall task of the sensor network.
- Node failure
- ➢ As the nodes are frequently deployed in extreme environmental locations for gathering data
- ➢ causing hardware problems
- ➢ physical damage
- ➢ depletion of the battery
- The reliability is modeled in using the Poisson distribution:

$R_k(t) = \exp(-\lambda_k t)$,
where $\lambda_k$ is the failure rate of sensor node k,
t is the time period

- Deployment of protocols which can detect these failures at the earliest time is demanded for the faster retrieval of data and efficient and normal functioning of the network.
- The protocol should also vary by location and usage.
- The protocols used for fault tolerance are
- ❖ automatic repeat request (ARQ)
- ❖ forward error correction (FEC),
- ❖ hybrid ARQ

2**. Mobility of nodes:**

- In order to increase the communication efficiency, the nodescan move anywhere within the sensor field based on the type of applications.

3**. Dynamic network topology:**

- Connection between sensor nodes follows somestandard topology. The WSN should have the capability to work in the dynamictopology.

4.**Communication failures:**

- If any node in the WSN fails to exchange data with other nodes, it should be informed without delay to the base station or gatewaynode.

5. **Heterogeneity of nodes**:

- The sensor nodes deployed in the WSN may be ofvarious types and need to work in a cooperative fashion.

**6.Scalability:**

- The number of sensor nodes in a sensor network can be in the order of hundreds or eventhousands. Hence, WSN designed for sensor networks is supposed to be highly scalable.
- New schemes must be utilize the high density of the sensor networks.
- The density $\mu$ can be calculated according to as $\mu(R) = (N * \pi R2) / A$, where N is the number of scattered sensor nodes in region A, and R is

the radio transmission range.

- WSNs can consist of hundreds to thousands of sensors based on the area of deployment and the type of deployment.
- For high resolution of data, node density may vary from place to place.
- The type of protocol being used for data retrieval needs to be scalable and such that it maintains an adequate performance.

**7. Independency:**
The WSN should have the capability to work without any centralcontrol point.
**8. Programmability**: The option for reprogramming or reconfiguring should beavailable for the WSN to become adaptive for any dynamic changes in thenetwork.
**9. Utilization of sensors**: The sensors should be utilized in a way that produces themaximum performance with less energy.
**10. Impracticality of public key cryptosystem**s: The limited computation and powerresources of sensor nodes often make it undesirable to use public key algorithms.
**11. Lack of aprior knowledge of post-deployment configuration:**
If a sensornetwork is deployed via random distribution, the protocols will not be aware ofthe communication status between each nodes after deployment.
**12.Self-Organization**
The large number of nodes in a WSN renders direct manipulation by a user for network organization impractical . A user could not go through thousands of nodes directing the network configuration and clustering. Subsequently, the nodes must be capable of organizing the network and partitioning it for efficient operation given the environment and network attributes. Additionally, the nodes of a sensor network must be robust . The aggregate formed by the nodes must have a high up time. The large number of nodes in a network along with unattended operation complicates any attempt at a fault tolerant design . Sensor networks with wired connections do not necessarily rely on other nodes to transmit data. This reduces the need for redundancy and the robustness of individual nodes.

In contrast, wireless sensor network nodes transmit information from node to node with a small amount of processing in between .Consequently individual nodes must be highly robust, while the organization of the network must tolerate individual device failure . Variations in the network topology can affect the degree of network vulnerability to failures, necessitating complex routines to implement fault tolerance .

**13.Concurrency, Cooperative Processing**

1. The nodes in a network primarily direct information flow through the network to various datasinks – the points to which data from the network is fed . Each sensor node may posses a limited amount of memory, so the

buffering of data is im-practical . Additionally, the node performs a number of simultaneous operations: capturing, processing, and transmitting sensor data, while simultaneously forwarding data from other nodes in multi-hop or bridging situations . WSNs also provide a unique opportunity for cooperative processing. Cooperative processing can reduce network traffic through data aggregation and pre processing . For example, theestablishment of a wireless network might involve the triangulation of a new node when it joins a network to establish the node's position.

14.**Energy Efficiency**

Energy Efficiency Wired sensor network have the luxury of external power sources such as power over Ethernet. The nodes of wireless networks have no practical way of utilizing an external energy source, which would in any case be contrary to the point of a WSN. A sensor network may also be distributed in hostile or remote environments . Energy efficiency dictates the minimization of communication between nodes. Therefore the choice of protocols and network configuration are key in terms of network lifespan .

Protocol related energy savings are directly related to the physical, link, and network layers . Additional power savings come from an operation system (OS) for the nodes which supports advanced power management and lower power task scheduling . Power sensitive task scheduling can minimize power use though non linear battery effects . Advanced power management would put any hardware not in use to sleep, minimizing power consumption

**15. Modularity**

Modularity Sensor nodes in a network tend to be specific, and therefore contain only the hardware needed for the application . The range of possible applications dictates a large variance in the hardware required for sensor nodes . Accordingly, the software for the nodes must exhibit a high degree of modularity

## ISSUES AND CHALLENGES IN DESIGNING WSN

1.Sensor networks do not fit into any regular topology, because while deploying the sensor nodes they are scattered
2.Very limited resources

- Limited memory,
- Limited computation
- Limited power

3.It comes under fewer infrastructures and also maintenance is very difficult.
4.Unreliable communication
5. Unreliable data transfer
6.Conflicts and latency
7.Sensor node relies only on battery and it cannot be recharged or replaced.
8.Hardware design for sensor node should also be considered.
9.Unattended operations
10.Exposure to physical attack
11.Remotely managed
12.No central control point
13.Achieving synchronization between nodes is also another issue.
14.Node failure, topology changes and adding of nodes and deletion of nodes isanother challenging issue
15.Because of its transmission nature and hostile environment, security is a challenging issue.
16.Based on the applications, sensor node has to be chosen with respect toComputational rate.
. 17.Security: Security is often a broadly used term encompassing the characteristicsof authentication, integrity, privacy, non repudiation, and anti-playback. Thegreater the dependency on the info supplied by the networks may be increased,the more potential risk of secure transmission of information in the networks hasincreased. To the secure transmission of numerous kinds of information overnetworks, several cryptographic, steganography and other techniques are utilizedthat happen to be renowned. In this section, we discuss the network securityfundamentals you bet the techniques are meant for wireless sensor networks .
14. Cryptography: The encryption-decryption techniques devised for your traditionalwired networks usually are not feasible to be employed directly for the wirelessnetworks in particular for wireless sensor networks. WSNs include things like tinysensors which really suffer from the possible lack of processing, memory andbattery Applying the security mechanisms for instance encryption could alsoincrease delay, jitter and packet loss in wireless sensor networks when applyingencryption schemes to WSNs like, what sort of keys are generated ordisseminated. How a keys are managed, revoked, assigned to your new sensorput into the network or renewed for ensuring robust to protect the

networkAdoption of pre-loaded keys or embedded keys could hardly be an efficientsolution.

15.Steganography: While cryptography aims at hiding necessary of a message,steganography aims at hiding a good the message. Steganography is the art ofcovert communication by embedding a note in to the multimedia data (image,sound, video, etc.) . The leading objective of steganography is to modify thecarrier in a fashion that is just not perceptible and hence, it looks the same asordinary [Physical Layer Secure Access: Physical layer secure access in wireless sensornetworks may very well be offered by using frequency hopping. A dynamicmixture of the parameters like hopping set (available frequencies for hopping),dwell time (interval per hop) and hopping pattern (the sequence in which thefrequencies in the available hopping set is used) could be combined with a littleexpense of memory, processing and resources. Important points in physical layer

secure access will be the efficient design in order that the hopping sequence ismodified in less time than is required to discover it and for employing this bothsender and receiver should maintain a synchronized clock. A scheme asproposed in may be utilized which introduces secure physical layer accessemploying the singular vectors while using channel synthesized modulation.Attacks against wireless sensor networks may very well be broadly consideredfrom two different levels of views. One is the attack from the security mechanismsand this band are brilliant from the basic mechanisms (like routing mechanisms).Ideas signalize the most important attacks in wireless sensor networks

16 Localization: It is amongst the key techniques in wireless sensor network. Theplace estimation method is usually classified into Target / source localization andnode self-localization. In target localization, we mainly introduce the energy-basedmethod. Then we investigate the node self-localization methods. Considering that

the widespread adoption on the wireless sensor network, the localization methodsare wide and varied in several applications.

Problems in localization

1. localization in non-line-of-sight,
2. node selection criteria for localizationin energy-constrained network
3. scheduling the sensor node to optimize thetradeoff between localization performance and energy consumption
4. cooperative node localization
5. localization algorithm in heterogeneous network.

Finally, we introduce the evaluation criteria for localization in wireless sensor network. Theentire process of estimating the unknown node position inside the network isknown as node self-localization. And WSN comprises a large number ofinexpensive nodes which are densely deployed in a very region of interests tomeasure certain phenomenon. The leading objective would be to determine

the location of the target . Localization is significant travelers have an uncertaintywith the exact location of some fixed or mobile devices. One example has been inthe supervision of humidity and temperature in forests and/or fields, wherethousands of sensors are deployed by way of plane, giving the operator minimalpossible ways to influence may location of node. An efficient localization algorithmmight utilize all the free information from the wireless sensor nodes to infer thepositioning of the individual devices. Another application will be the positioning ofan mobile robot determined by received signal strength from your number of radiobeacons placed at known locations around the factory floor. The primary functionof an location estimation method to calculate the geographic coordinates of
network nodes with unknown position in the deployment area. Localization inwireless sensor networks is the process of determining the geographical positionsof sensors. Only a number of the sensors (anchors) inside the networks have priorknowledge about their geographical positions. Localization algorithms utilizelocation information of anchors and estimates of distances between neighboringnodes to discover the positions in the rest of the sensors
17.Power-Consumption: However,because the wireless sensor node is normally positioned in a hard to reachlocation, changing the battery regularly will not be free and inconvenient. Anessential take into account the introduction of a wireless sensor node is makingsure that there's always adequate energy accessible to power the system .The facility consumption rate for sensors in the wireless sensor network variesgreatly good protocols the sensors use for communications. The Gossip-BasedSleep Protocol (GSP) implements routing and many MAC functions in aenergyconserving manner. The effectiveness of GSP has already been demonstrated viasimulation. However, no prototype system has become previously developed.GSP was implemented for the Mica2 platform and measurements were conductedto discover the improvement in network lifetime. Results for energy consumption,transmitted and received power, minimum voltage supply necessary for operation,effect of transmission power on energy consumption, and different methods formeasuring time of a sensor node are presented. The behavior of sensor nodeswhen they're all around their end of lifetime is described and analyzed.
18.Deployment: Sensor networks provide capability to monitor real-world phenomenain more detail and also at large scale by embedding wireless network of sensornodes in the environment. Here, deployment is anxious with establishing anoperational sensor network inside a real-world environment. On many occasions,deployment is often a labor-intensive and cumbersome task as environmentalinfluences trigger bugs or degrade performance in a way that is not observedduring pre-deployment testing within a lab. The real reason for this really is thatthe real life features a strong influence for the function of your sensor network bygoverning the output of sensors, by influencing the existence and excellence ofwireless communication links, and also by putting physical

strain on sensornodes. These influences is only able to be modeled to your very restricted extentin simulators and lab testbeds. Home the typical problems encountered during deployment is rare.You can only speculate for the grounds for this

Main Challenges

- high bandwidth demand
- high energy consumption,
- quality of service (QoS) provisioning,
- data processing
- compressing techniques
- cross-layer design
- physical environment

Designing factors in WSN

- Production cost
- The cost of a single node is very important .
- If the cost is more expensive than deploying traditional sensors, the sensor network is not cost-justified.
- Hardware Constraints --A sensor node is made up of four basic components: sensing unit, processing unit, transceiver unit, and power unit.
- They may also have additional application-dependent components such as a location finding system, power generator, and mobilizer.
- The required all of these subunits may be smaller than even a cubic centimeter.
- The frequent failure of sensor nodes may lead to a heavy cost for the WSN.
- The sensor node cost should be kept in mind before deployment of nodes in WSNs.

- Transmission media
- Hardware Constraints
- Node Unit Costs
- Environment
- Sensor nodes are densely deployed either very close or directly inside the phenomenon to be observed.
- They may be working in the interior of large machinery, at the bottom of an ocean, in a biologically or chemically contaminated field, in a battlefield beyond the enemy lines, and in a home or large building.
- Transmission Media
- In a multi-hop sensor network, communicating nodes are linked by a wireless medium.

- These links can be formed by radio, infrared, or optical media. The chosen transmission medium must be available worldwide.
- Sensor nodes are densely deployed either very close or directly inside the phenomenon to be observed.
- They may be working in the interior of large machinery, at the bottom of an ocean, in a biologically or chemically contaminated field, in a battlefield beyond the enemy lines, and in a home or large building.
- Transmission Media
- In a multi-hop sensor network, communicating nodes are linked by a wireless medium.
- These links can be formed by radio, infrared, or optical media. The chosen transmission medium must be available worldwide.

- Latency
- Receiving exact information within a WSN is one of the primary aspects.
- This can be achieved by minimizing delay by selecting a suitable routing protocol and network topology.

- Transmission Channels
- Connectivity
- Network Topology
- Issues related to topology maintenance and change in three phases:
- Pre-deployment and deployment phasePost-deployment phase
- Re-deployment of additional nodes phase
- Pre-deployment and deployment phase : Sensor nodes can be either thrown in mass or placed one by one in the sensor field.
- Post-deployment phase : Topology changes are due to change nodes' position, reach-ability, available energy, malfunctioning, and task details.
- Re-deployment of additional nodes phase: Additional sensor nodes can be redeployed at any time to replace malfunctioning nodes or due to changes in task dynamics.

- Power Consumption
- The wireless sensor node, being a microelectronic device, can only be equipped with a limited power source.
- The malfunctioning means fail to function normally.
- The malfunctioning of a few nodes can cause significant topological changes and might require rerouting of packets and reorganization of the network.
- Power consumption can hence be divided into three domains:
- WSNs normally rely on non-renewable sources of energy.
- in terms of energy is the size of the batteries needed.
- Energy is also needed for data compression, which is again an issue.
- Limited energy of the nodes is a crucial issue, as it is impossible to replace or recharge batteries.

- Harvesting solar energy is not possible as it cannot be adopted due to the large size of solar panels.
- Power consumption is factor equivalent to lack of bandwidth, memory, and processing capability.
- In order to increase the network lifetime, quality of-service (QoS) support mechanisms need to be developed and should be lightweight with low and balanced energy consumption.

- sensing, communication, and data processing.
-
-

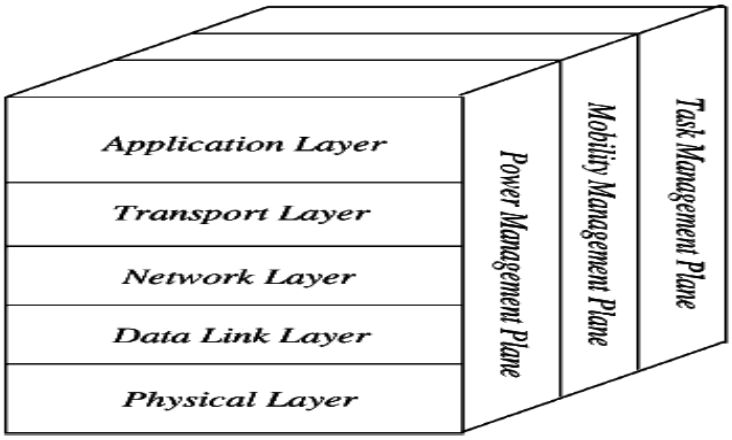
Limitations in wireless sensor networks,

- As it is wireless in nature, it is prone to hacking by hackers.
- It can not be used for high speed communication as it is designed for low speed applications.
- It is expensive to build such network and hence can not be affordable by all.
- limited processing speed
- limited storage capacity
- limited communication bandwidth.
- Overhearing
- Overhearing occurs when a node receives a packet destined to other nodes.
- Overhearing can be a major reason of energy waste mainly with a high node density causing a heavy traffic load.

- 1. Less secure because hackers can enter the access point and obtain all the information.
- 2. Lower speed as compared to a wired network.
- 3. More complicated to configure compared to a wired network.
- 4. Easily troubled by surroundings (walls, microwave, large distances due to signal attenuation, etc).
- 5. It is easy for hackers to hack it we couldn't control propagation of waves.
- 6. Comparatively low speed of communication.
- 7. Gets distracted by various elements like Blue-tooth. 8. Still Costly (most importantly).

Introduction

The physical layer is mostly concerned with modulation and demodulation of digital data; this task is carried out by so-called transceivers. In sensor networks, the challenge is to find modulation schemes and transceiver architectures that are simple, low cost, but still robust enough to provide the desired service.




- The physical (PHY) layer is responsible for the conversion of bit streams into signals that are best suited for communication across the wireless channel
- It describes major PHY layer technologies used for wireless communication.
- The main types of technologies used for RF communication in WSNs can be classified into three
  - narrow-band,
  - spread-spectrum
  - ultra-wide-band (UWB) techniques.
- In addition to RF techniques, optical, acoustic, and magnetic induction communication techniques are used.
- The physical layer is mostly concerned with modulation and demodulation of digital data;
- this task is carried out by so-called transceivers.
- In sensor networks, the challenge is to find modulation schemes
- transceiver architectures that are simple, low cost, but still robust enough to provide the desired service
- The physical layer is responsible for
  - carrier frequency selection
  - carrier frequency generation
  - modulation
  - signal detection/demodulation
  - data encryption
  - Data decryption

- Data transmission
- Data reception
- A communication physical layerprovides mechanical, electrical, functional, and procedural characteristics to establish, maintain, and release physical connections between data link entities.802.11b and Bluetooth also use the same band, so all systems, in this band have to be robust against interference form other systems.

Factors that effect the designing Physical layer

- 1. Low Power Consumption.
  2. Low Transmission and Reception range.
  3. Interference from other systems, working in the same band.
  4. Low complexity.
  5. Low duty cycle, i.e. most of the time sensor nodes are switched off.
  6. Low data rates most of the time and high data rate only for a short period of time.

The most challenging aspect in physical layer design for sensor networks is to find, low cost transceivers which consume less power, simple modulation schemes which are robust enough to provide required service .

Power in PHY layer

- Generally the transceivers used in sensor network are only 10% efficient.
- To radiate a power of 1 mw the transceivers will consume at least 10 mw of power.
- The power consumed for reception is somewhat similar to power consumed for transmission;

sometimes one of them may be more and one less depending upon design of the transceivers.

- For Mica motes 21 mw is consumed in transmit mode and 15 mw is consumed in receive mode.
- The transmission and reception of data are the most power consuming activity in the sensor node.
- So transmission and reception of data should be kept as less possible .
- The power consumed by the transceivers in idle mode will not be significantly less than the power consumed by transceivers in receive or transmit mode.
- So it is always preferable to put the transceiver in sleep mode rather than in idle mode when not required.
- But care has to be taken to see to that, the power consumed during start up and time taken to startup the transceiver does not overcome the advantage of putting the transceiver in sleep mode.
- The most commonly used transceiver is CC2420.

choice of modulation

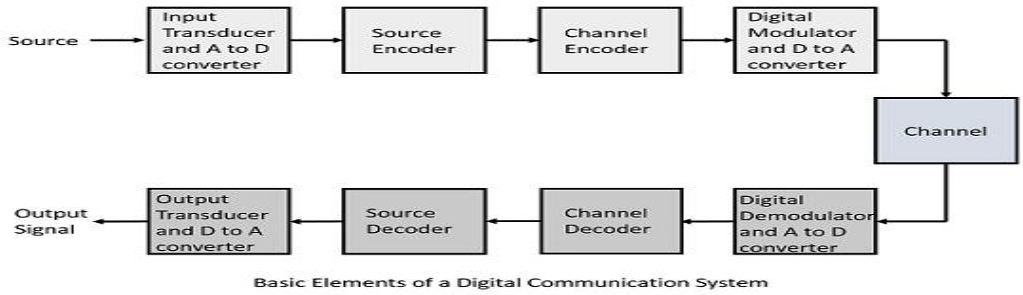
The choice depends upon the complexity that can be supported by the node, as one cannot go for very complex processes in sensor networks

- data rate --The data rate is another factor to be considered, as this tells about what amount of data can be transmitted or received when the node is on.

SAVING --To save more and more energy the sensor node should be in sleep as much as possible and more the data rate more amount of data can be transmitted and received in small active period.
symbol rate --more is the symbol rate more is the power consumption. The BER has also to be considered, because to keep BER low the power radiated has to be more

- IEEE 802.15.4. IEEE 802.15.4 defines wireless personal area network standard for low rate devices, this specifies the standards for Physical and MAC layer.
- 802.15.4 Works at a data rate of less than 250kbps.
- It works up to a range of 75m and it supports up to 254 nodes.
- This protocol is designed specially for networks whose data rate is less, which have energy constraints, and which require good QoS.
- The network using this protocol can survive from 6 months to 2 years with two AA batteries.
- 802.15.4 Works in three frequency bands they are 868 MHz (20 kbps data rate), 915MHz (40 kbps data rate) and 2.4GHz (250 kbps data rate).
- It uses Direct Sequence Spread Spectrum for modulation and uses BPSK for first two bands and 16 bit array QPSK for the last band.
- The first band has 1 channel in it
- the second band has 10 channels in it
- third band has 16 channels in it.

SOURCE ENCODER:



Basic Elements of a Digital Communication System

- Information is the source of a communication system, whether it is analog or digital.
- Information theory is a mathematical approach to the study of coding of information along with the quantification, storage, and communication of information.

**Conditions of Occurrence of Events**

- ❖ If we consider an event, there are three conditions of occurrence.
- ❖ If the event has not occurred, there is a condition of uncertainty.
- ❖ If the event has just occurred, there is a condition of surprise.
- ❖ If the event has occurred, a time back, there is a condition of having some information.
- ❖ These three events occur at different times.
- ❖ The difference in these conditions help us gain knowledge on the probabilities of the occurrence of events

**Source coder**

- The symbols produced by information source are given to encoder.
- These symbols cannot be transmitted directly.
- They are first converted into digital form.
- Every binary'0' and '1' are called bit.
- The group of bits is called codeword.
- Source encoder assigns codewords to symbols.
- For every distinct word there is a unique code word.
- The codeword can be of 4,8,16nor 32 bit length.
- As number of bits increased, symbols are increased.

**source coding Parameters**

- 1.block size
- 2.code word length
- 3.Average date rate
- 4.efficiency of encoder

**block size**

- This gives maximum number of distinct words that can be represented by source encoder.
- It depends on maximum number of bits .
- For 8 bits -256 code words (2 power 8)

**code word length**

- This is number of bits used to represent each codeword.
- Codeword =8 bits

**Average date rate**

- it is output bits from source encoder.
- Source coder assigns multiple number of bits to every input symbol.
- Therefore data rate is higher than symbol rate.
- Data rate=10 symbols/sec
- Codeword=8bits
- Data rate= symbol rate X codeword length
- =10X8 =80 bits/sec
- Information rate is minimum number of bits to convert information from source to destination.
- Average data rate is higher than symbol rate &information rate.

**EFFICIENCY OF ENCODER**

- This is ratio of minimum source information rate to actual data output rate.

**source coding**

- A compression technique that takes into account the nature of the information to be compressed
- The use of variable-length codes in order to reduce the number of symbols in a message to the minimum necessary to represent the information in the message, or at least to go some way toward this, for a given size of alphabet.

**AIM** ::The aim of source coding is to represent information as accurately as possible using as few bits as possible and in order to do so redundancy from the source needs to be removed.

- The Source encoder ( or Source coder) converts the input i.e. **symbol sequence** into a binary sequence of 0's and 1's by assigning code words to the symbols in the input sequence.
- The output of the source encoder is an analog signal.

**EFFICIENCY**

- The maximum entropy of an 8 symbol source is log2 8 = 3 bit/symbol .
- the source efficiency is therefore given by:
- If the symbols are each allocated 3 bits, comprising all the binary patterns between 000 and 111,
- the coding efficiency will remain unchanged at 85%.
- The purpose of source coding is to reduce the number of bits required to convey the information provided by the information source.

- For the file of the source data this means compression
- For the transmission, this is a bit rate reduction
- The source encoder is responsible for compressing the input information sequence to represent it with less redundancy.
- The compressed data is passed to the channel encoder.

**Data compression**

- Minimize the size of information representation.
- Reduce the redundancy of the original representation. • Purposes: •
- ❖ Save storage space.
- ❖ Reduce transmission time.
- categories of data compression
- ❖ Lossless compression, such as zip, gzip, gif
- ❖ Lossy compression, such as mpeg, jpeg
- An exact copy of the original data is obtained after decompression .
- decompression into exactly the original form (typical for text).
- Structured data can be compressed to 40-60% of their original size

**Lossy compression**

- Original information content is lost
- decompression into approximately the original form (typical for signals and images)
- Any data can be compressed significantly (e.g., 100 times**)**

**information theory**

- The area of information theory explores the information content of data.
- We can predict the size of the content by modeling data Within a given model.
- we can obtain lower bounds on the number of bits required to represent data.

**Modeling1**

- Consider the following message of x1, x2,….x12 9,11,11,11,14,13,15,17,16,17,20,21
- Using binary encoding scheme, we can store each of these numbers using 5 bits per symbol

**Modeling information 2**

- Original message 9,11,11,11,14,13,15,17,16,17,20,21
- We can store instead the following 0,2,2,2,5,4,6,8,7,8,11,12
- Each of these values represents the number n+9, thus, we can use 4 bits per number .

**Modeling information3**

- Original: 9,11,11,11,14,13,15,17,16,17,20,21 Or
- we can store instead the following 9,2,0,0,3,-1,2,2,-1,2,2,-1,1,3,1 The first number is stored as is.
- To encode xi, we use xi –x i-1.
- There are only 5 distinct values which can be encoded with 3 bits each

**Probabilistic Modeling1**

- An ensemble 'X' is a random variable x with a set of possible outcomes Ax ={a1,a2,…. aI}, having probabilities {p1,p2,…pI}, with P(x=aI)=Pi, Pi>=0 and ΣxϵAx P(x)=1. A is called Alphabet

**Probabilistic Modeling2**

- Source outputs alphabet {1,2,3,4}.
- Each sample is modeled as a random variable with the probabilities
- P(1)= 8/15
- P(2) = 4/15
- P(3)= 2/15
- P(4) = 1/15

**Probabilistic Modeling3**

- If no compression takes place, then the simple code with the average 2 bit/symbol is as follows
- 1 00
- 2 01
- 3 10
- 4 11
- **PROBLEM**
- **Suppose the channel capacity is 1.75 b/sec**
- Variable length code 1
- Intuition: Less bits for higher probability symbols P(1)= 8/15
- P(2) = 4/15
- P(3)= 2/15
- P(4) = 1/15
- 1 0
- 2 10
- 3 110
- 4 111
-

**Variable length code 2**

- The expected number of bits per code sample (average length l ) is 1*p(1) + 2*p(2) +3*p(3)+3*p(3)= 5/3
- That is 1.67.
- Thus, the channel will transmit the code perfectly.

**Measure of information**

- Shannon defined a quantity: self-information associated with a symbol in a message.
- The higher the probability that a symbol occurs the lower the information content.
- At the extreme if P(a)=1, then there is nothing learned when receiving an a, since that is the only possibility.

## ENTROPY

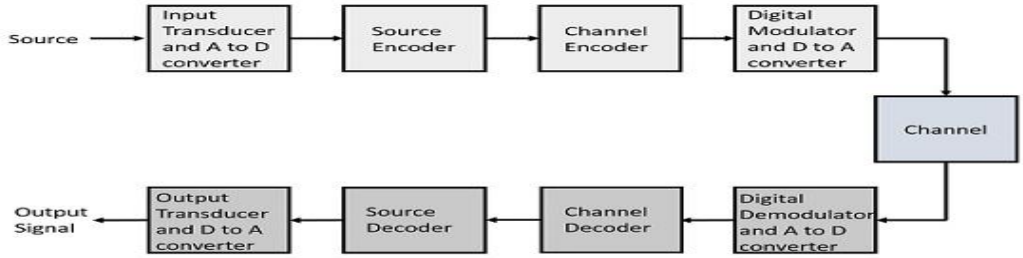
- ENTROPY associated with a message representing the average self information per symbol in the message expressed in radix b.
- Entropy is a measure of information (b/symbol) that gives a low bound for coding

Shannon's first theorem

- The noiseless source coding theorem (also called Shannon's first theorem) states that an instantaneous code can be found that encodes a source of entropy H(x) with an average number of bits per symbol B such that B=H(x) Assumption: all symbols are independent and identically distributed (iid**)**

- **P(1)= 8/15**
- **P(2) = 4/15**
- **P(3)= 2/15**
- **P(4) = 1/15**
- **H(x) = -8/15 *log(8/15)-4/15* log(4/15) – 2/15*log(2/15) –1/15 *log(1/15) = 1.5b/symbol**

## CHANNEL ENCODER




Basic Elements of a Digital Communication System

- Message or information is converted in binary form.
- Communication channel adds noise & interference to signal.
- Therefore errors are introduced in binary sequence at receiver.
- Errors are also introduced in symbols generated from these binary codes.
- To avoid these errors, channel coder is used.
- add redundancy to the transmitted signal
- The channel encoder introduces controlled redundancy, called parity bits.
- The input to the channel encoder has a block length of 260 bits, and the output of the encoder is 456 bits long

WHY Channel Encoder

- Wireless communication systems depend on channel coding (sometimes called forward error correction) to ensure that the data received is the

same as the data sent.

- Wireless links suffer from interference and fading which causes errors, so to overcome this the transmitter adds additional information before the data is sent.
- Then at the receiver end, complex codes requiring sophisticated algorithms decode this information and recover the original data .
- Channel Encoder adds some redundant bits to input sequence.
- These redundant bits are added with some pre-defined logic.
- E.G.
- Code word of source encoder =3bit
- Add 1 redundant bit to make 4 bit long
- Added $4^{th}$ bit may be 1 or 0
- We can add in such a way that there are even number of 1s
- Every codeword at output of channel encoder contains even number of 1s.
- At the receiver, if odd number of 1's are detected, then receiver comes to know that there is an error in received signal.
- EFFECTING PARAMETERS OF CHANNEL ENCODER

1. Method of coding used
2. Coding rate, which depends on redundant bits added by channel encoder
3. Coding efficiency= data rate at input / data rate at output
4. Detecting & correcting errors
5. Complexity of encoder & decoder
6. Time delay involved in decoding

- The primary purpose of channel coding is to increase efficiency
- the better the system can correct the inevitable errors introduced by wireless transmission the more efficient it will be.

benefits of channel encoding

1. Reduced error rates and retransmission
2. Increased capacity
3. Increased throughput
4. Reduced power usage

Reduced error rates and retransmission

- The Block Error Rate or BLER measures errors that cannot be corrected
- un-correctable errors require retransmissions, which introduce latency and lead to poor end user experience.

Increased capacity

- As more data is sent across a link noise and interference increase.
- Better error correction enables the system to tolerate the increased errors

created and therefore allows greater capacity – ultimately resulting in lower per user cost.

Increased throughput

- Higher performance channel coders and decoders can sustain higher data rates leading to increased throughput

Reduced power usage

- Increasing the efficiency of the wireless link reduces the power requirements at both the base station and the user's device, which in turn leads to increased battery life.

Channel decoder

- A channel decoder is used on the receiver side to return the binary information back to its original form by removing the parity bits.
- The channel-coded bits should be mapped into a certain electromagnetic waveform employing amplitude, frequency, and phase by a modulator.
- the channel decoder exploits the redundancy to facilitate the detection and correction of bit errors
- determine the actual transmitted input bit sequence.
- Channel decoder at receiver is thus able to detect error in bit sequence.
- It also reduces effect of channel noise & distortion.

CHANNEL ENCODER & DECODER

- Channel Encoder & Decoder thus serve to increase reliability of received signal.
- Extra bits added by channel encoder does not carry any information, rather they are used by channel decoder to detect & correct if errors any.
- These error correcting bits may be added recurrently after the block of few symbols or added in every symbol.
- Coding & decoding operations at encoder & decoder needs memory(storage) & processing of binary data.
- Because of microcontrollers& computers the complexity of Channel Encoder & Decoder is reduced.
-

Network Architecture

The most common wireless sensor network architecture follows the OSI architecture Model. The architecture of the WSN includes five layers and three cross layers.These layers of the WSN are used to accomplish the n/w and make the sensors work together in order to raise the complete efficiency of the network. Network Architecture is used in Wireless Sensor Network (WSN). It can be used in various places like schools, hospitals, buildings, roads, etc for various applications like disaster management, security management, crisis management, etc.

There are *2 types of architecture used in WSN:* Layered Network Architecture, and Clustered Architecture.

Layered Network Architecture:

Layered Network Architecture makes use of a few hundred sensor nodes and a single powerful base station. Network nodes are organized into concentric Layers.

It consists of 5 layers and three cross layers

- The 5 layers are:
  - Physical layer
  - Data link layer
  - Network layer
  - Transport layer
  - Application layer

The physical layer

The physical layer emphasizes on the transmission media between sending and receiving nodes, the data rate, signal strength, frequency types are also addressed in this layer. Ideally FHSS frequency hopping spread spectrum is used in sensor networks.

Data link layer

- Data link layer does the error detection and correction, and encoding of data. Link layer is vulnerable to jamming and DoS attacks. TinySec has introduced link layer encryption which depends on a key management scheme. However, an attacker having better energy efficiency can still rage an attack. Protocols like LMAC have better anti-jamming properties which are viable countermeasure at this layer.

Network layer

- Network layer is responsible for routing of messages from node to node, node to cluster leader, cluster leaders to cluster leaders, cluster leaders to the base station and vice versa.

Application layer

Data is collected and managed at application layer therefore it is important to ensure the
reliability of data. Wagner has presented a resilient aggregation schemewhich is applicable to a cluster based network where a cluster leader acts as an aggregator in sensor networks. However this technique is applicable if the aggregating node is in the range with all the source nodes and there is no intervening aggregator between the aggregator and source nodes. To prove the validity of the aggregation, cluster leaders use the cryptographic techniques to ensure the data reliability.

The cross layers consist of the following:

- Power Management Plane
- Mobility Management Plane
- Task Management Plane

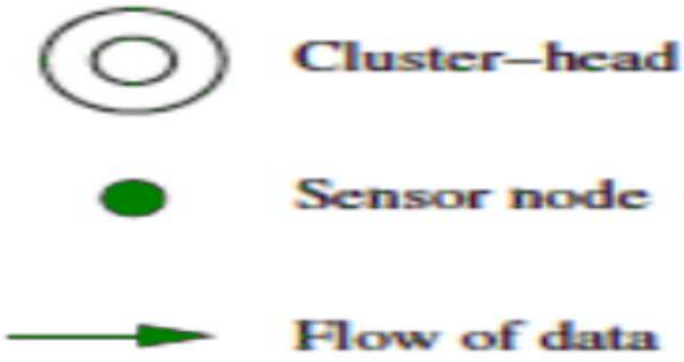
The advantage of using *Layered Network Architecture* is that each node participates only in short-distance, low power transmissions to nodes of the neighbouring nodes because of which power consumption is less as compared to other Sensor Network Architecture. It is scalable and has a higher fault tolerance.

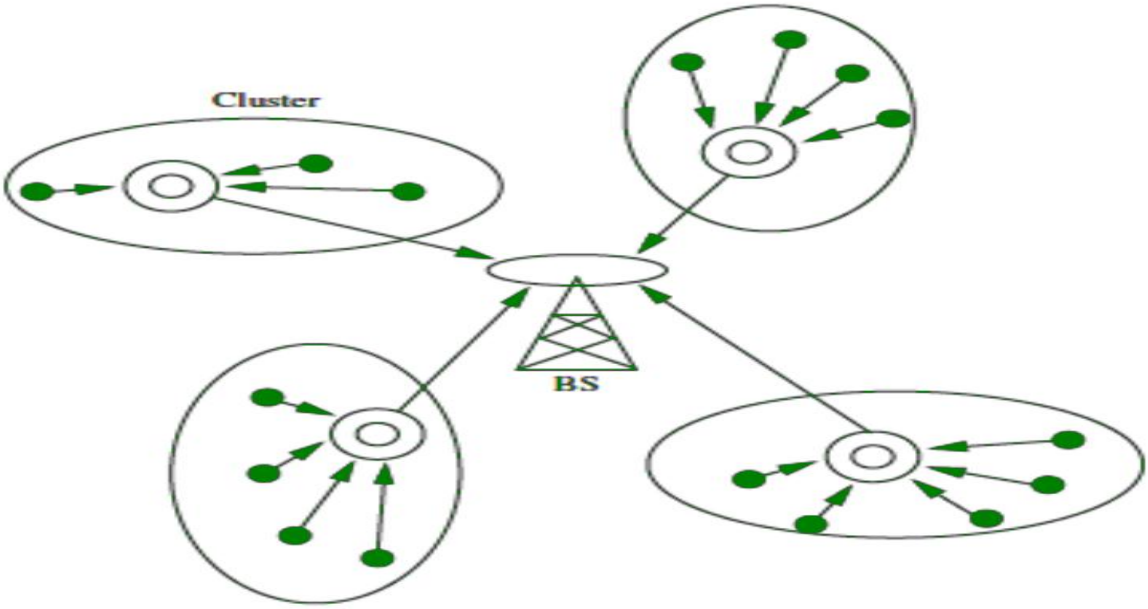
Clustered Network Architecture:

In Clustered Network Architecture, Sensor Nodes autonomously clubs into groups called clusters. It is based on the *Leach Protocol* which makes use of clusters. Leach Protocol stands for Low Energy Adaptive Clustering Hierarchy.

Properties of Leach Protocol:

- It is a 2-tier hierarchy clustering architecture.
- It is a distributed algorithm for organizing the sensor nodes into groups called clusters.
- The cluster head nodes in each of the autonomously formed clusters create the Time-division multiple access (TDMA) schedules.
- It makes use of the concept called *Data Fusion* which makes it energy efficient.






Clustered Network Architecture is a very useful sensor network because of the property of Data Fusion. Inside each cluster, each node communicate with the cluster head to gather the information. All the clusters which are formed share their gathered information to the base station. The cluster formation and selection of cluster head inside each cluster is an independent and autonomous distributed process.

**MODULATION**

- Modulation is the process by which some characteristics of a carrier is varied in accordance with a modulation wave .
- Modulation is the process of putting information onto a high frequency carrier for transmission (frequency translation).
- Such a modification is achieved by mean of a process called modulation .
- In the modulation process, the baseband signal is called “modulating signal”
- another higher frequency signal is called as the “carrier”.
- The carrier signal will carry the modulating signal to the destination.
- Modulation schemes use analog signals to represent digital bits
- Modulation is what takes a signal from low frequencies (the message) and pulls it up to a higher frequency (the carrier)
- Modulation is the process of converting data into electrical signals optimized for transmission.
- Modulation techniques are roughly divided into four types:

- Analog modulation
- Digital modulation,
- Pulse modulation
- Modulation is the process by which information is encoded into electrical signals for transmission over a medium.
- Binary information, as represented by a series of 1s and 0s, must be converted to analog or digital electrical signals for transmission.
-

**Reasons for Modulation**

- Simultaneous transmission of several signals
- Practical Design of Antennas
- Exchange of power and bandwidth

**Why Modulation**

- It increases the range of communication.
- The frequency of baseband signal is low, and the low frequency signals can not travel long distance when they are transmitted .
- They get heavily attenuated .
- The attenuation reduces with increase in frequency of the transmitted signal.So they travel longer distance .
- Modulation allows us to send a signal over a bandpass frequency range.
- If every signal gets its own frequency range, then we can transmit multiple signals simultaneously over a single channel, all using different frequency ranges.
- Another reason to modulate a signal is to allow the use of a smaller antenna

**Modulation Signal**

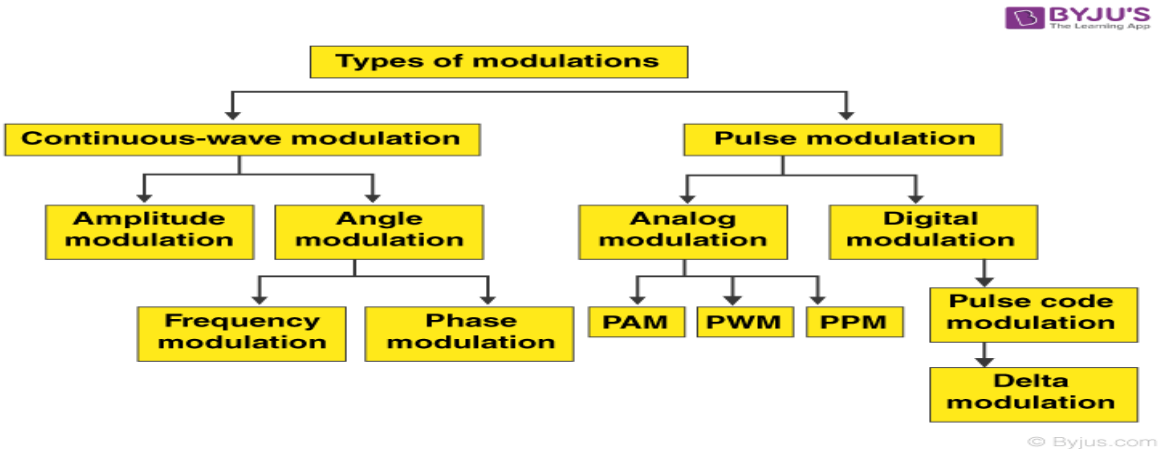
- The modulation signal might be
- an audio signal representing sound from a microphone,
- a video signal representing moving images from a video camera,
- or a digital signal representing a sequence of binary digits,
- a bitstream from a computer.

**The carrier is higher in frequency than the modulation signal**.

## Modulation and its need

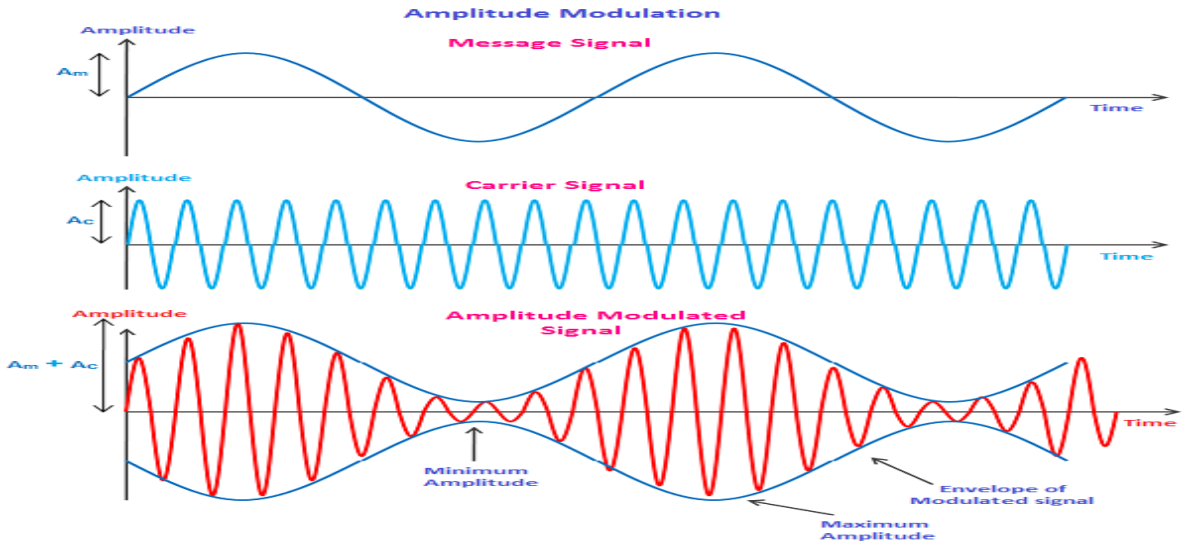
- Modulation is simply a widely used process in communication systems in which a very high-frequency carrier wave is used to transmit the low-frequency message signal
- so that the transmitted signal continues to have all the informatio.

## Classsifcation



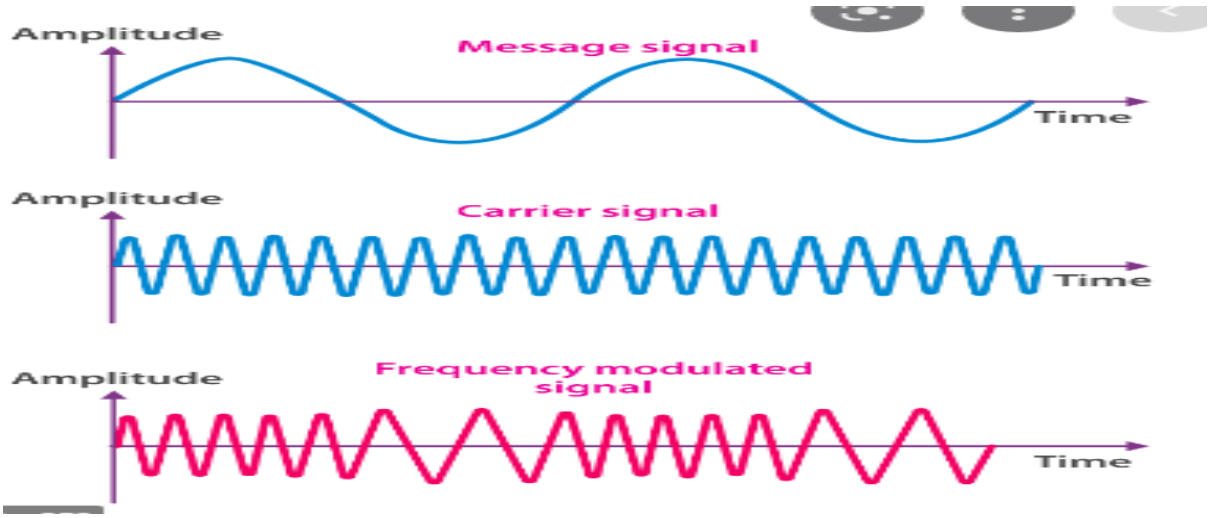

## Where is modulation done?

- The modulation is done at the carrier signal while transmitting that signal.
- Hence the carrier signal properties get modulated.
- The receiver section demodulates the modulated signal to extract the original signal.
- **AMPLITUDE MODULATION**
- **Amplitude modulation** (AM) is a modulation technique used in electronic communication, most commonly for transmitting messages with a radio wave.

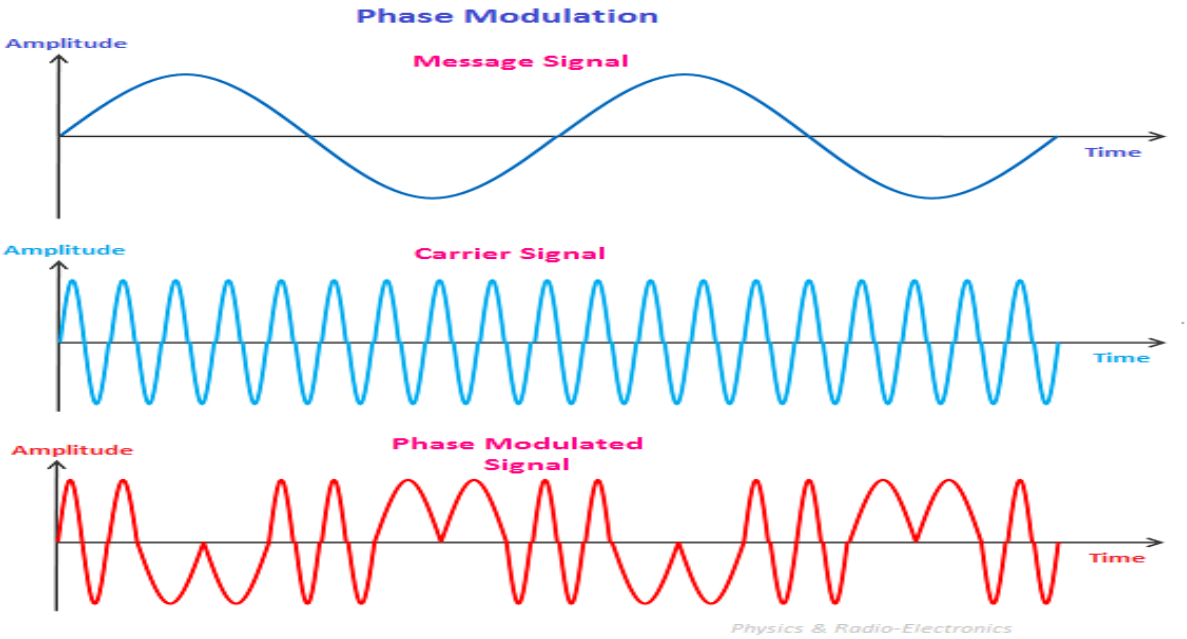


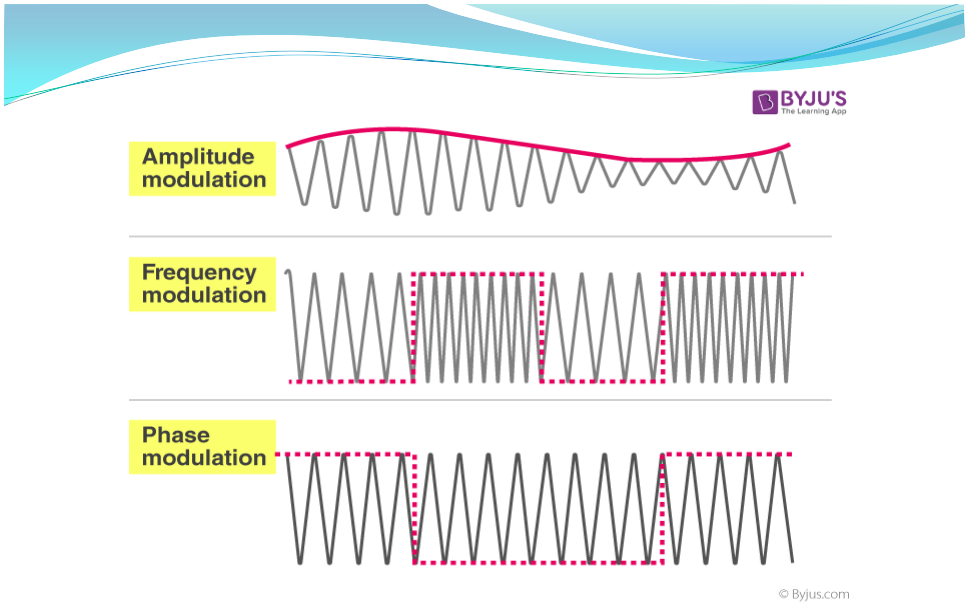

## FREQUENCY MODULATION

- Frequency modulation is a technique or a process of encoding information on a particular signal (analogue or digital) by varying the carrier wave frequency in accordance with the frequency of the modulating signal.




phase modulation is a type of modulation, used in communication systems, in which the phase of a radio carrier wave is varied by an amount proportional to the instantaneous amplitude of the modulating signal.

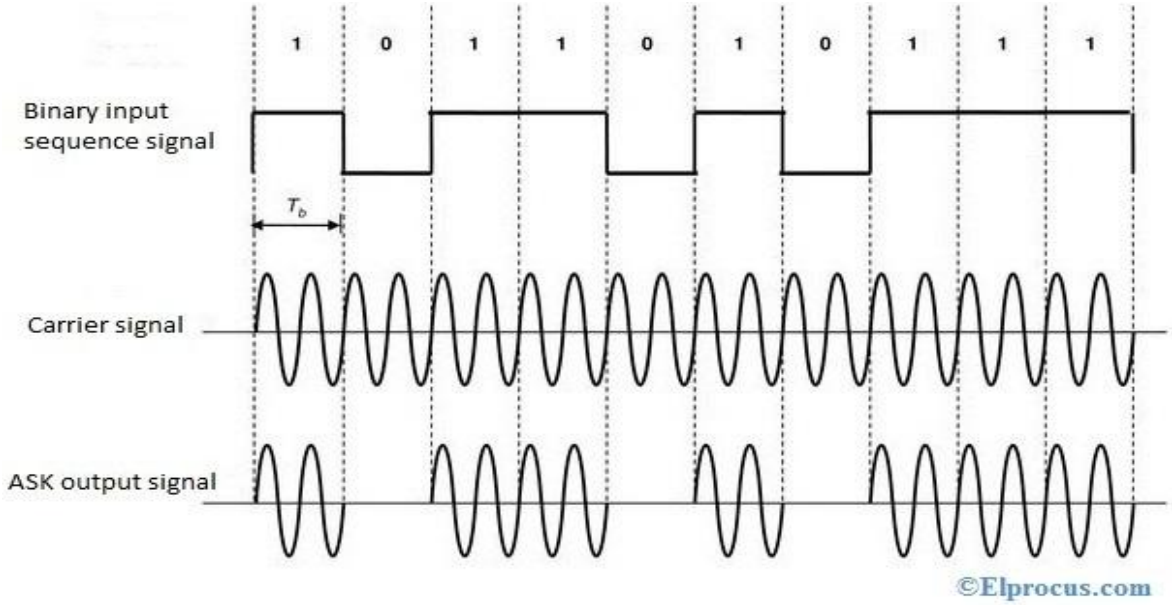



Pulse amplitude modulation (PAM) is the transmission of data by varying the amplitude s ( voltage or power levels) of the individual pulses in a regularly timed sequence of electrical or electromagnetic pulses.

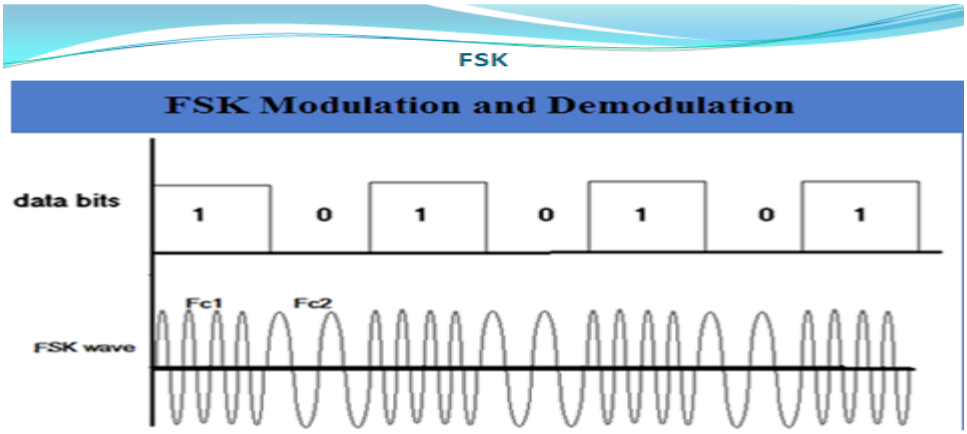
## ASK

- If amplitude is the only parameter of the carrier wave to be altered by the information signal, the modulating method is called amplitude-shift keying (ASK).
- ASK can be considered a digital version of analog amplitude modulation
- a burst of radio frequency is transmitted only when a binary 1 appears and is stopped when a 0 appears. In another variation, the 0 and 1 are represented in the modulated signal by a shift between two preselected amplitudes.




FSK

- If frequency is the parameter chosen to be a function of the information signal, the modulation method is called frequency-shift keying (FSK).
- In the simplest form of FSK signaling, digital data is transmitted using one of two frequencies, whereby one frequency is used to transmit a 1 and the other frequency to transmit a 0



- 

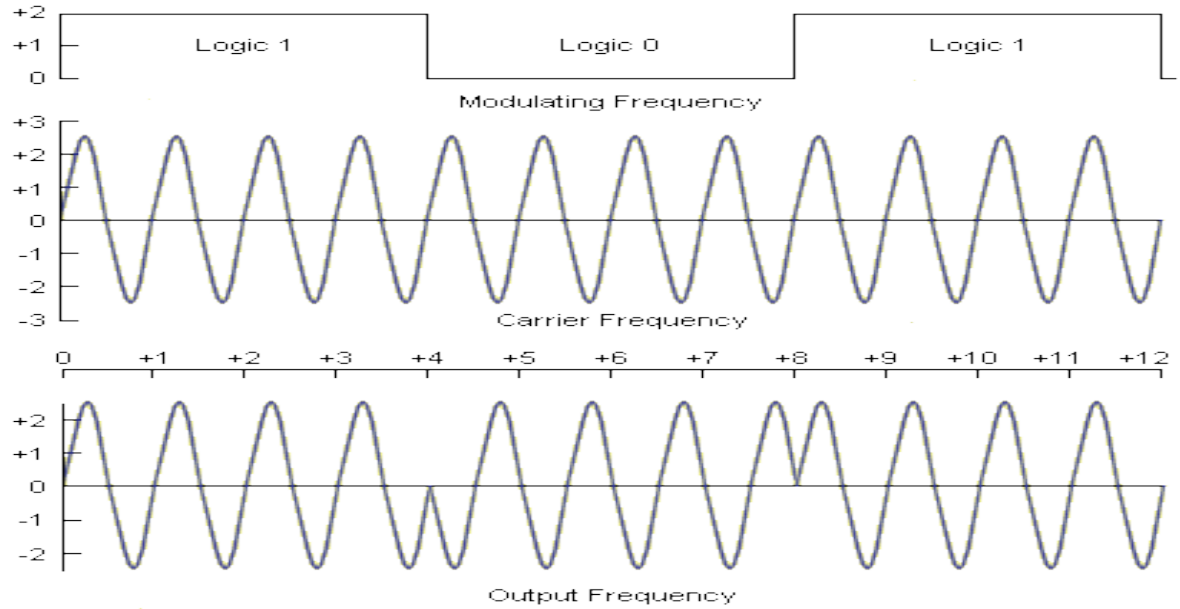
**PSK**

Phase Shift Keying (PSK) refers to a digital modulation technique in which the sine and cosine inputs of a radio wave vary at a specific time, altering the phase of the carrier signal.

When phase is the parameter altered by the information signal, the method is called phase-shift keying (PSK).

In the simplest form of PSK a single radio frequency carrier is sent with a fixed phase to represent a 0 and with a 180° phase shift—that is, with the opposite polarity—to represent a 1.

This kind of method is used to transmit data by modulating otherwise changing the phase of the carrier signal which is known as a reference signal

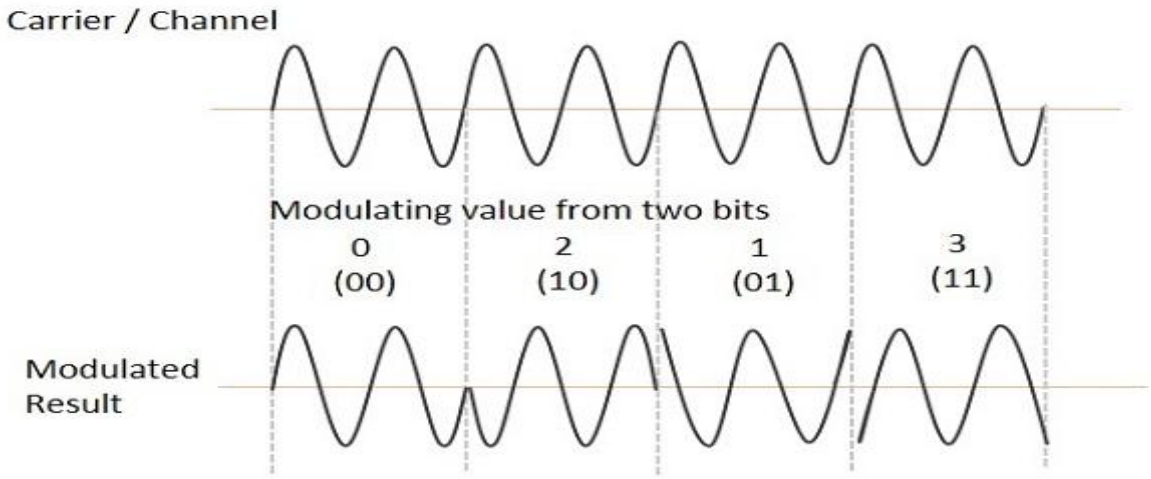



**QPSK**

Quadrature Phase Shift Keying (QPSK) is **a form of Phase Shift Keying in which two bits are modulated at once, selecting one of four possible carrier phase shifts (0, 90, 180, or 270 degrees)**

- In this modulator the carrier assumes one of two phases.
- A logic 1 produces no phase change and a logic 0 produces a 180° phase change.
- The QPSK Modulator uses a bit-splitter, two multipliers with local oscillator, a 2-bit serial to parallel converter, and a summer circuit.

.




**QPSK vs BPSK**

- QPSK devices modulate input signals by 0°, 90°, 180°, and 270° phase shifts.
- BPSK devices modulate input signals by 0° and 180° phase shifts.
- QPSK represents two bits using complex carrier symbol each having 90 degree shift with one another.
- BPSK is easy in the receiver to receiver the original bits.
- Both QPSK modulators and BPSK modulators are used in conjunction with demodulators that extract information from the modulated, transmitted signal.
- **BPSK**
- BPSK modulation has highest efficiency
- BPSK format are often believed to be the most power efficient one
- modulation format requiring the lowest signal-to-noise ratio (SNR) per bit to reach a given bit error rate (BER).
- binary phase shift keying (BPSK), shifts the carrier sine wave 180° for each change in binary state

**Pulse modulation** is a type of modulation in which the signal is transmitted in the form of pulses.

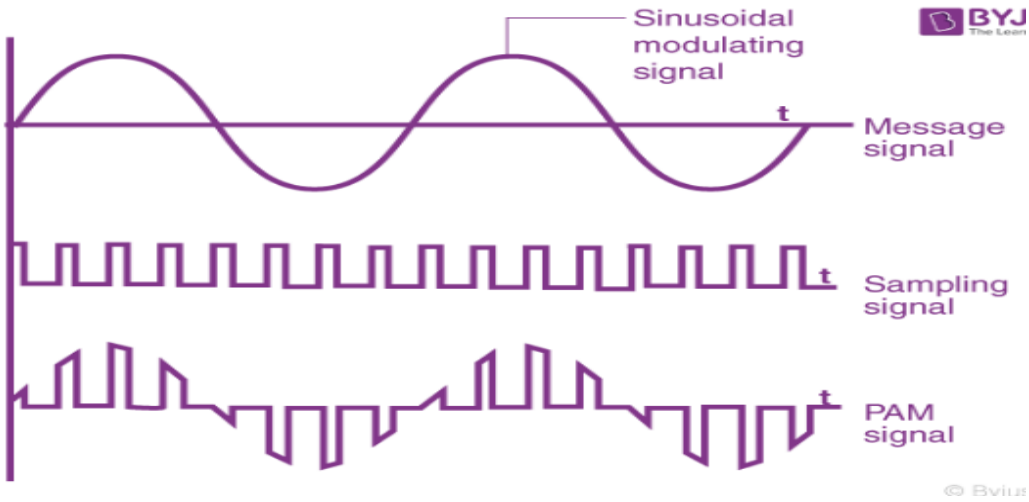
It can be used to transmit analogue information.

In pulse modulation, continuous signals are sampled at regular intervals.

Pulse modulation can be classified into two major types.

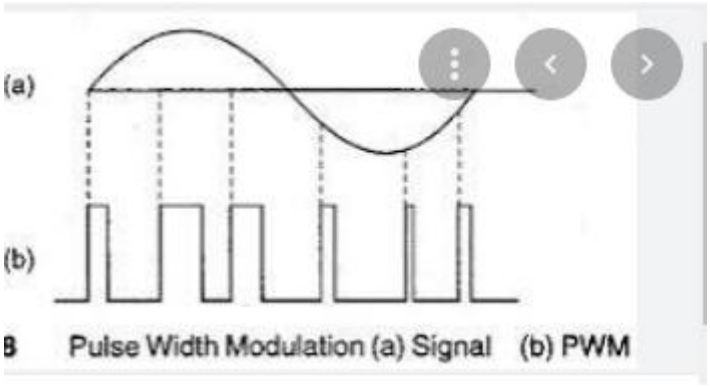
- Analogue: Indication of sample amplitude is infinitely variable
- Digital: Indicates sample amplitude at the nearest predetermined **level.**

**Pulse amplitude modulation** (PAM) is the transmission of data by varying the amplitude s ( voltage or power levels) of the individual pulses in a regularly timed sequence of electrical or electromagnetic pulses.




**PWM**

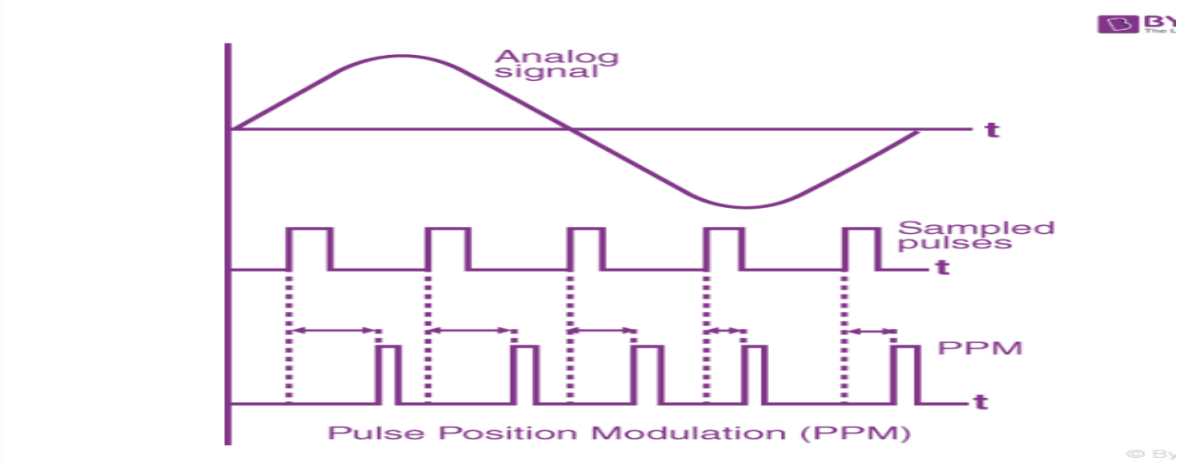
- Pulse Width Modulation is also known as pulse duration modulation (PDM).
- Here, as the name suggests, the width of the pulse is varied in proportional to the amplitude of the signal.
- Pulse width modulation (PWM) is a modulation technique that generates variable-width pulses to represent the amplitude of an analog input signal.



Pulse Width Modulation (a) Signal (b) PWM

**Pulse Position Modulation (PPM)**

- In this type of modulation, both the amplitude and width of the pulse are kept constant.
- We vary the position of each pulse with reference to a particular pulse.
- Pulse position modulation is an analogue modulation scheme
- here the amplitude and width of the pulse are kept constant, while the position of the pulse with respect to the position of a reference pulse is varied according to the instantaneous value of message signal.




## QAM

- QAM is the most bandwidth efficient one.
- QAM (quadrature amplitude modulation) is a method of combining two amplitude modulation (AM) signals into a single channel.
- This approach helps double its effective bandwidth.

QAM is also used with pulse AM (PAM) in digital systems, like wireless applications.

- QAM is a modulation scheme used by network operators when transmitting data.

- QAM relates to a way of changing the amplitude, or power level, of two signals.
- QAM enables an analog signal to efficiently transmit digital information and increases the useable bandwidth.
- Digital microwave relay, dial up modem and etc uses QAM modulation technique.

**Why it is called QAM**

- QAM is a technique used to transmit two digital bit streams or two analog signals by modulating or changing the amplitudes of two carrier waves.
- so that they differ in phase by 90 degrees, a quarter of a cycle, hence the name quadrature

QAM is a digital television standard using quadrature amplitude modulation.

- It is the format by which digital cable channels are encoded and transmitted via cable television providers.
- QAM is used in a variety of communications systems such as Dial-up modems and Wi-Fi.
- QAM can be used to increase the capacity and speed of a wireless network.

Modulation factor is defined as the ratio of the change in the amplitude of the carrier wave to the amplitude of the original carrier wave.

**Conclusion**

- There are many variants of modulation techniques used in transceivers of Wireless Sensor Network nodes based on the application for which the nodes are deployed.
- The option of modulation is based on the channel environment, distance, encoding, etc.
- By varying the modulation technique in a periodic manner with the analysis of those parameters, an appreciable link quality could be produced.
- Nowadays as the number of nodes in sensor network is drastically increased that requires good link quality.
- Considering more parameters in the sensor nodes increases the design complexity of the sensor nodes

**modulation and demodulation of signals**

- In modulation, the original message signal is mixed with a carrier wave whose parameters are required to be changed.
- In demodulation, the combination of carrier and message signal are separated from each other, to have an original information signal.
- demodulation is the recovery of original information at the distant end of the carrier.
- A modem is an equipment that performs both modulation and demodulation.

**sensor networks scenarios**

Wireless Sensor Network (WSN) is an infrastructure-less wireless network that is deployed in a large number of wireless sensors in an ad-hoc manner that is used to monitor the system, physical or environmental conditions.

A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous devices using sensors to cooperatively monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations.

A sensor network is a group of sensors where each sensor monitors data in a different location and sends that data to a central location for storage, viewing, and analysis.

Sensor nodes are used for constant sensing, event ID, event detection & local control of actuators. The applications of wireless sensor networks mainly include health, military, environmental, home, & other commercial areas. Such networks are used in industrial and consumer applications, such as industrial process monitoring and control and machine health monitoring. A WSN is built of "nodes" – from a few to hundreds or thousands, where each node is connected to other sensors.

The sensor network connects to the internet or computer networks to transfer data for analysis and use. Sensor network nodes cooperatively sense and control the environment. They enable interaction between persons or computers and the surrounding environment.

properties of sensor networks

A wireless sensor network consists of many different components of which a sensor node is an important yet small part. The characteristics of a good wireless sensor network include power efficiency, scalability, responsiveness, reliability and mobility.

Benefits or advantages of SN

1) It is scalable

2) It can accommodate any new nodes or devices at any time.
3) It is flexible and hence open to physical partitions.
4) All the WSN nodes can be accessed through centralized monitoring system.
5) As it is wireless in nature, it does not require wires or cable

**Sensor network scenarios**

**Types of sources and sinks**

- patterns found in WSN are
  - ❖ event detection
  - ❖ periodic measurements
  - ❖ function approximation
  - ❖ edge detection
  - ❖ tracking

**Event detection**

Sensor nodes should report to the sink(s) once they have detected the occurrence of a specified event. The simplest events can be detected locally by a single sensor node in isolation (e.g. a temperature threshold is exceeded); more complicated types of events require the collaboration of nearby or even remote sensors to decide whether a (composite) event has occurred (e.g. a temperature gradient becomes too steep). If several different events can occur, event classification might be an additional issue.

**Periodic measurements**Sensors can be tasked with periodically reporting measured values. Often, these reports can be triggered by a detected event; the reporting period is application dependent.

**Function approximation and edge detection**The way a physical value like temperature changes from one place to another can be regarded as a function of location. A WSN can be used to approximate this unknown function (to extract its spatial characteristics), using a limited number of samples taken at each individual sensor node. This approximate mapping should be made available at the sink. How and when to update this mapping depends on the application's needs, as do the approximation accuracy and the inherent trade-off against energy consumption.

**Tracking**The source of an event can be mobile (e.g. an intruder in surveillance scenarios). The WSN can be used to report updates on the event source's position to the sink(s), potentially with estimates about speed and direction as well. To do so, typically sensor nodes have to cooperate before updates can be reported to the sink.

- A source is any entity in the network that can provide information, that is, typically a sensor node; it could also be an actuator node that provides feedback about an operation.A sink is the entity where information is required

**Single-hop versus multihop networks**

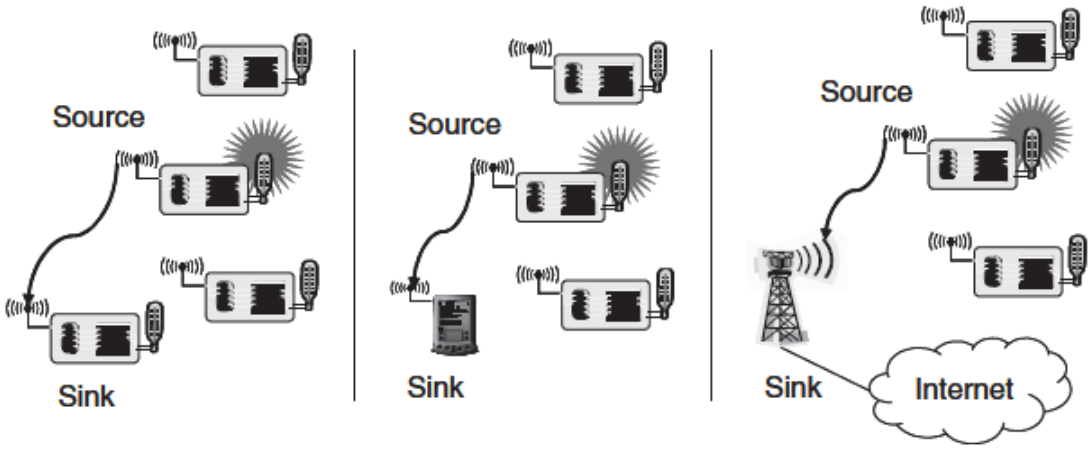
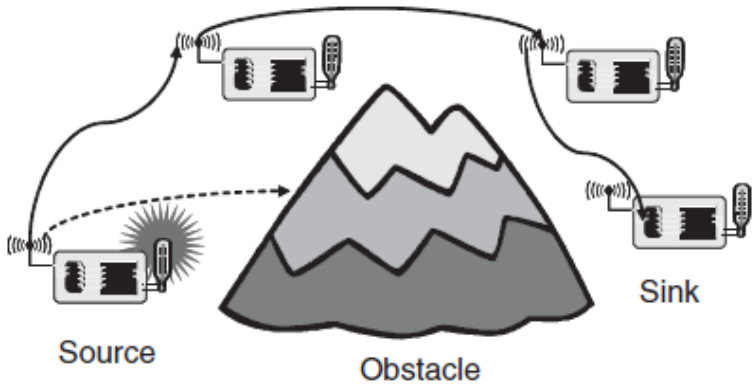



**Figure 3.1** Three types of sinks in a very simple, single-hop sensor network

From the basics of radio communication and the inherent power limitation of radio communication follows a limitation on the feasible distance between a sender and a receiver. Because of this limited distance, the simple, direct communication between source and sink is not always possible, specifically in WSNs, which are intended to cover a lot of ground (e.g. in environmental or agriculture applications) or that operate in difficult radio environments with strong attenuation (e.g. in buildings).
To overcome such limited distances, an obvious way out is to use relay stations, with the data packets taking multi hops from the source to the sink. This concept of multi hop networks is particularly attractive for WSNs as the sensor nodes themselves can act as such relay nodes, foregoing the need for additional equipment. Depending on the particular application, the likelihood of having an intermediate sensor node at the right place can actually be quite high – for example, when a given area has to be uniformly equipped with sensor nodes anyway – but nevertheless, there is not always a guarantee that such multi hop routes from source to sink exist, nor that such a route is particularly short.

While multi-hopping is an evident and working solution to overcome problems with large distances or obstacles, it has also been claimed to improve the energy efficiency of communication. The intuition behind this claim is that, as attenuation of radio signals is at least quadratic in most environments (and usually larger), it consumes less energy to use relays instead of direct communication: When targeting for a constant SNR at all receivers (assuming for simplicity negligible error rates at this SNR), the *radiated* energy required for direct communication over a distance $d$ is $cd\alpha$ ($c$ some constant, $\alpha \geq 2$ the path loss coefficient); using a relay at distance $d/2$ reduces this energy to $2c(d/2)\alpha$.
Even assuming that this relay belongs to the WSN and is willing to cooperate, when computing the total required energy it isnecessary to take into account the

complete power consumption .It is an easy exercise to show that energy is actually wasted if intermediate relays are used for short distances *d*.Only for large *d* does the radiated energy dominate the fixed energy costs consumed in transmitter and receiver electronics – the concrete distance where direct and multihopcommunication are in balance depends on a lot of device-specific and environment-specific parameters. Nonetheless, this relationship is often not considered.




Multihop networks: As direct communication is impossible because of distance and/or obstacles, multihop communication can circumvent the problem Multihopping saves energy as the number one myth about energy consumption in wireless communication. Great care should be taken when applying multihopping with the endof improved energy efficiencyonlymultihop networks operating in a store and forward fashionare considered here. In such a network, a node has to correctly receive a packet before it can forwardit somewhere.

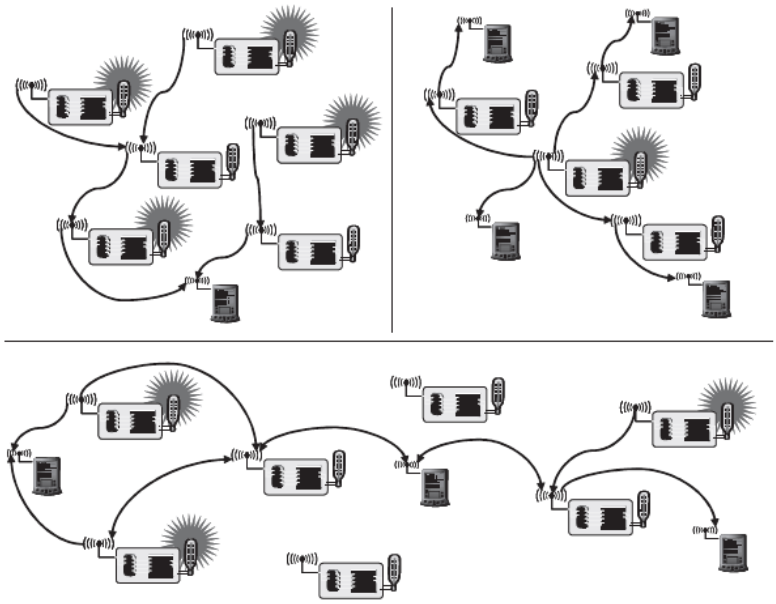


**Figure 3.3** Multiple sources and/or multiple sinks. Note how in the scenario in the lower half, both sinks and active sources are used to forward data to the sinks at the left and right end of the network

Alternative, innovative approaches attempt to exploit even erroneous reception of packets, for example, when multiple nodes send the same packet and each individual transmission could not be received, but collectively, a node can reconstruct the full packet. Such cooperativerelayingtechniques are not considered here.

**Multiple sinks and sources**

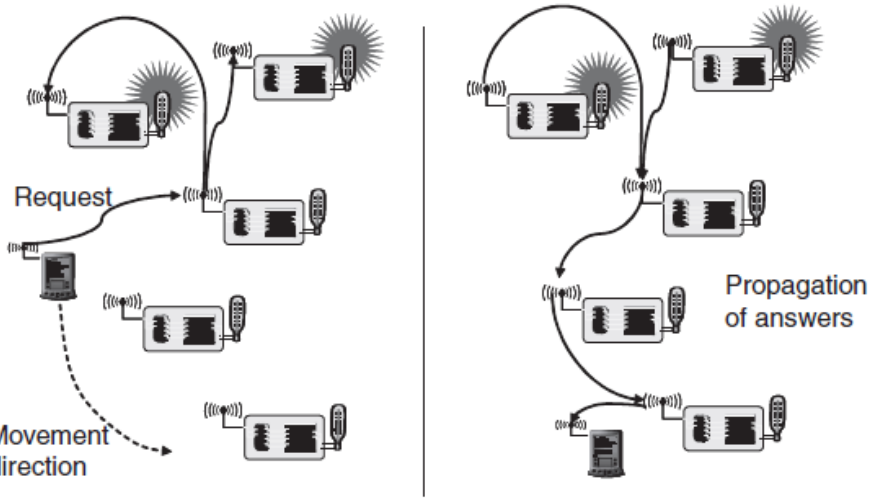
there are multiple sources and/or multiple sinks present. In the most challenging case, multiple sources should send information to multiple sinks, where either all or some of the information has to reach all or some of the sinks

**Three types of mobility**

In wireless sensor networks mobility can appear in three main forms:
Node mobility The wireless sensor nodes themselves can be mobile. The meaning of such mobility is highly application dependent. In examples like environmental control, node mobility should not happen; in livestock surveillance (sensor nodes attached to cattle, for example), it is the common rule.

In the face **of node mobility**, the network has to reorganize itself frequently enough to be able to function correctly. It is clear that there are trade-offs between the frequency and speed of node movement on the one hand and the energy required to maintain a desired level of functionality in the network on the other hand.

**Sink mobility**The information sinks can be mobile (Figure 3.4). While this can be a special case of node mobility, 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. Whether the requester is allowed interactions with any node or only with specific nodes is a design choice for the appropriate protocol layers.




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

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

**Event mobility**

In applications like event detection and in particular in tracking applications, the cause of the events or the objects to be tracked can be mobile. In such scenarios, it is (usually) important that the observed event is covered by a sufficient number of sensors at all time. Hence, sensors will wake up around the object, engaged in higher activity to observe the present object, and then go back to sleep. As the event source moves through the network, it is accompanied by an area of activity within the network – this has been called the *frisbee* model, (which also describes algorithms for handling the "wakeup wavefront"). This notion is described by Figure  where the task is to detect a moving elephant and

to observe it as it moves around. Nodes that do not actively detect anything are intended to switch to lower sleep states unless they are required to convey information from the zone of activity to some remote sink

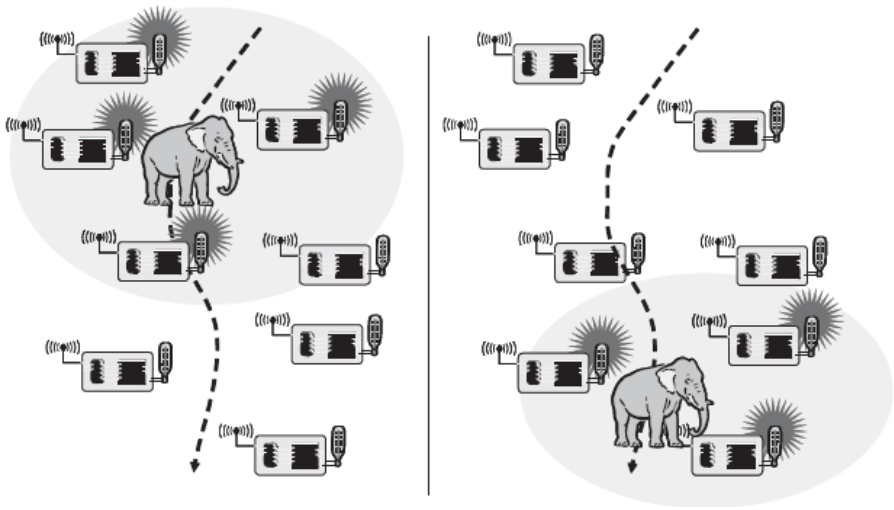


FIG:Area of sensor nodes detecting an event – an elephant – that moves through the network along with the event source (dashed line indicate the elephant's trajectory; shaded ellipse the activity are following or even preceding the elephant)

Communication protocols for WSNs will have to render appropriate support for these forms of mobility. In particular, event mobility is quite uncommon, compared to previous forms of mobile or wireless networks.

**Optimization goals and figures of merit**

optimization goals in wireless sensor networks

- ❖ Objective 1: Optimization of energy consumption.
- ❖ Objective 2: optimization of system lifetime.
- ❖ Objective 3: coverage optimization problem.
- ❖ Objective 4: Optimization of the participating number of satellites.

**FACTORS OF OPTIMIZATION GOALS& FIGURE OF MERIT**

- **1.QOS**
- **2. Energy efficiency**
- **3. Scalability**
- **4.Robustness**

- **figure of merit** is a quantity used to characterize the performance of a device, system or method, relative to its alternatives**.**

**goal of sensor management in WSN---**Configuration management is a high correlation of sensor nodes in wireless sensor network management

system. The goal of WSNManagement is to monitor sensor network collection, processing, and transmission of data, and ultimately control an environment.

**performance metrics that are used for evaluating the performance of WSN**

- **The three performance metrics are:**

**1. coverage**

**2. energy consumption**

**worst case delay.**

**I Quality of service**

WSNs differ from other conventional communication networks mainly in the type of service they offer. These networks essentially only move bits from one place to another. Such QoS can be regarded

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

While the first kind of attributes is applicable to a certain degree to WSNs as well (bandwidth, for example, is quite unimportant), the second one clearly is not, but is really the more important one to consider! Hence, high-level QoS attributes corresponding to the subjective QoS attributes inconventional networks are required. high-level QoS attributes in WSN highly depend on the application.

QOS Depends on

1. Event detection/reporting probability
2. Event classification error
3. Event detection delay
4. Missing reports
5. Approximation accuracy
6. Tracking accuracy

Event detection/reporting probability

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

Event classification error

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

Event detection delay

- is the delay between detecting an event and reporting it to any/all interested sinks?

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

Approximation accuracy

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

Tracking accuracy

- In Tracking accuracy Tracking applications must not miss an object to be tracked, the reported position should be as close to the real position as possible, and the error should be small.
- Other aspects of tracking accuracy are, for example, the sensitivity to sensing gaps

**II Energy efficiency**

- Energy is a precious resource in WSN
- Energy efficiency should therefore make an evident optimization goal.
- with an arbitrary amount of energy, most of the QoS metrics can be increased almost at will (approximation and tracking accuracy are notable exceptions as they also depend on the density of the network).
- Hence, putting the delivered QoS and the energy required to do so into perspective should give a first, reasonable understanding of the term energy efficiency.
- The term "energy efficiency" is, in fact, rather an umbrella term for many different aspects of a system, which should be carefully distinguished to form actual, measurable figures of merit.

considered aspects in energy efficiency

Energy per correctly received bit

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

Energy per reported (unique) event

- In Energy per reported (unique) event what is the average energy spent

to report one event?

- Since the same event is sometimes reported from various sources,
- it is usual to normalize this metric to only the unique events (redundant information about an already known event does not provide additional information).

Network lifetime

- Network lifetime is The time for which the network is operational or, put another way, the time during which it is able to fulfill its tasks (starting from a given amount of stored energy).
- It is not quite clear, however, when this time ends.

Time to loss of coverage

- Time to loss of coverage is Usually, with redundant network deployment and sensors that can observe a region instead of just the very spot where the node is located, each point in the deployment region is observed by multiple sensor nodes.
- A possible figure of merit is thus the time when for the first time any spot in the deployment region is no longer covered by any node's observations.
- If *k redundant observations are necessary (for tracking applications, for example), the* corresponding definition of loss of coverage would be the first time any spot in the deployment region is no longer covered by at least *k different sensor nodes.*
- Time to failure of first event notification

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

**III Scalability**

The ability to maintain performance characteristics irrespective of the size of the network is referred to as scalability. With WSN potentially consisting of thousands of nodes, scalability is an evidently indispensable requirement. Scalability is ill served by any construct that requires globally consistent state, such as addresses or routing table entries that have to be maintained. Hence, the need to restrict such information is enforced by and goes hand in hand with the resource limitations of sensor nodes, especially with respect to memory.

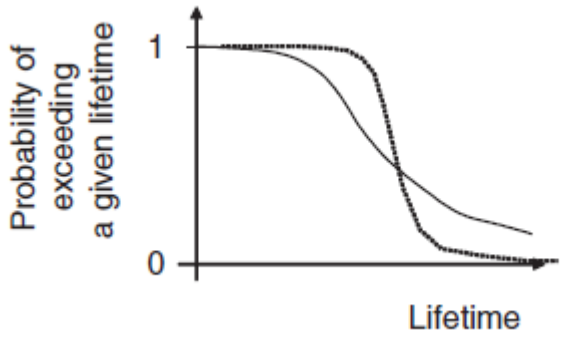



FIG:Two probability curves of a node exceeding a given lifetime – the dotted curve trades off betterminimal lifetime against reduced maximum lifetime

- The need for extreme scalability has direct consequences for the protocol design. Often, a penalty in performance or complexity has to be paid for small networks. Architectures and protocols should implement *appropriate* scalability support rather than trying to be as scalable as possible. Applications with a few dozen nodes might admit more efficient solutions than applications with thousands of nodes; these smaller applications might be more common in the first place.Nonetheless, a considerable amount of research has been invested into highly scalable architectures and protocols.

**IV Robustness**

- 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 and severs existing radio links between two nodes –
- if possible, these failures have to be compensated for, for example, 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

**figure of merit** is a quantity used to characterize the performance of a device, system or method, relative to its alternatives.

**Gateway Concepts.**

- The Gateway acts as **a bridge between the WSN and the other network**.
- This enables data to be stored and processed by devices with more resources,
- A wireless wide area network used primarily for low-power devices is known as a Low-Power Wide-Area Network (LPWAN).
- Gateway is a network node used in telecommunications that connects two networks with different transmission protocols together.

- Gateways serve as an entry and exit point for a network as all data must pass through or communicate with the gateway prior to being routed.

**Gateway types**

- There are two main types of gateways
- ❖ unidirectional gateways
- ❖ bidirectional gateways.
- Unidirectional gateways allow alerts to flow in only one direction.

**Gateway Function**

- The gateway converts information, data or other communications from one protocol or format to another.
- A router may perform some of the functions of a gateway.
- An Internet gateway can transfer communications between an enterprise network and the Internet.
- A gateway is a network node that forms a passage between two networks operating with different transmission protocols
- The most common type of gateways, the network gateway operates at layer 3, i.e. network layer of the OSI (open systems interconnection) model.
- A product or feature that uses proprietary techniques to link heterogeneous systems.
- Because enterprises often use protocols on their local-area networks (LANs) that differ from those of the Internet
- gateway will often act as a protocol converter so that users can send and receive communications over the Internet.

**Network gateway**

- Network gateway known as protocol translation gateways or mapping gateways
- They can perform protocol conversions to connect networks with different network protocol technologies.
- For example, a network gateway connects an office or home intranet to the Internet**.**

**gateway device**

- A gateway device provides communication to a remote network or an autonomous system that is out of bounds for the host network nodes.
- Gateways serve as the entry and exit point of a network
- all data routed inward or outward must first pass through and communicate with the gateway in order to use routing paths.

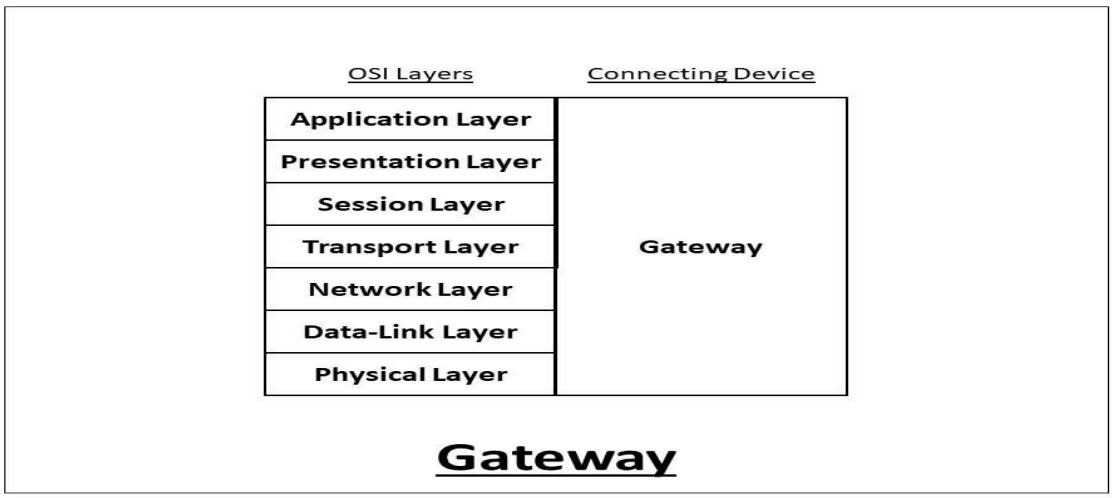
**Default gateway**

- The default gateway is the path used to pass information when the device doesn't know where the destination is.

- More directly, a default gateway is a router that connects your host to remote network segments.
- It's the exit point for all the packets in your network that have destinations outside your network.

**Advantages of using a Gateway:**

1) It can connect the devices of two different networks having dissimilar structures.
2) It is an intelligent device with filtering capabilities.
3) It has control over both collisions as well as a broadcast domain.
4) It uses a full-duplex mode of communication.




**Difference between router and gateway?**

- Both Routers and Gateways are network connecting devices.
- Routers work at the network layer and are responsible to find the shortest path for a packet.
- Routers connect devices across multiple networks.
- Gateways, in contrast, function as a node that serves as a gateway to a network's other nodes.

**need for gateways**

- For practical deployment, a sensor network only concerned with itself is insufficient.
- The network rather has to be able to interact with other information devices
- for example, a user equipped with a PDA moving in the coverage area of the network or with a remote user, trying to interact with the sensor network via the Internet

- The standard example is to read the temperature sensors in one's home while traveling and accessing the Internet via a wireless connection.

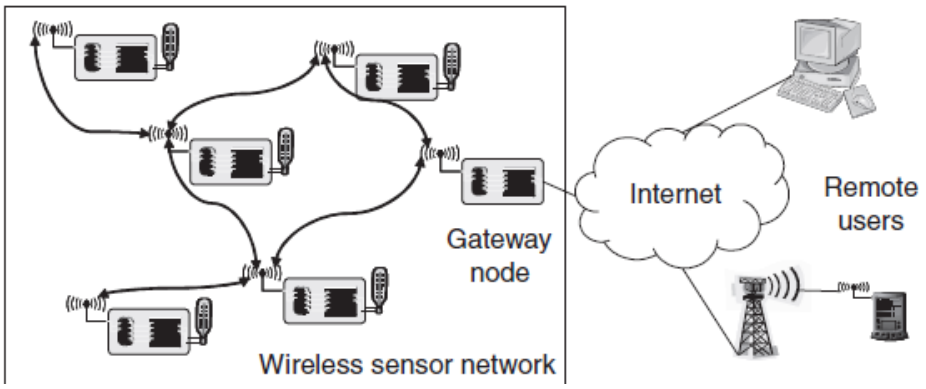



FIG: A wireless sensor network with gateway node, enabling access to remote clients via the Internet

- To this end, the WSN first of all has to be able to exchange data with such a mobile device or with some sort of gateway, which provides the physical connection to the Internet.
- This is relatively straightforward on the physical, MAC, and link layer – either the mobile device/the gateway is equipped with a radio transceiver as used in the WSN, or some (probably not all) nodes in the WSN

support standard wireless communication technologies such as IEEE 802.11.

- Either option can be advantageous, depending on the application and the typical use case.
- Possible trade-offs include the percentage of multi technology sensor nodes that would be required to serve mobile users in comparison with the overhead and inconvenience to fit WSN transceivers to mobile devices like PDAs.
- The design of gateways becomes much more challenging when considering their logical design.
- One option to ponder is to regard a gateway as a simple router between Internet and sensor network.
- This would entail the use of Internet protocols within the sensor network. While this option has been considered as well and should not be disregarded lightly, it is the prevalent consensus that WSNs will require specific, heavily optimized protocols. Thus, a simple router will not suffice as a gateway

**WSN to Internet communication**

- Assume that a sensor node wants to deliver an alarm message to some Internet host.
- The first problem to solve is akin to ad hoc networks, namely,
- how to find the gateway from within the network.

- Basically, a routing problem to a node that offers a specific service has to be solved, integrating routing and service discovery

**how to choose gateway**

- In particular, if not all Internet hosts are reachable via each gateway
- ❖ at least if some gateway should be preferred for a given destination host
- How to handle several gateways, each capable of IP networking, and the communication among them?
- ❖ One option is to build an IP overlay network on top of the sensor network

**How does a sensor node know to which Internet host to address such a message**

- Even if the sensor node does not need to be able to process the IP protocol, it has to include sufficient information (IP address and port number) in its own packets;
- the gateway then has to extract this information and translate it into IP packets.
- which source address to use here – the gateway in a sense has to perform tasks similar to that of a Network Address Translation (NAT) device

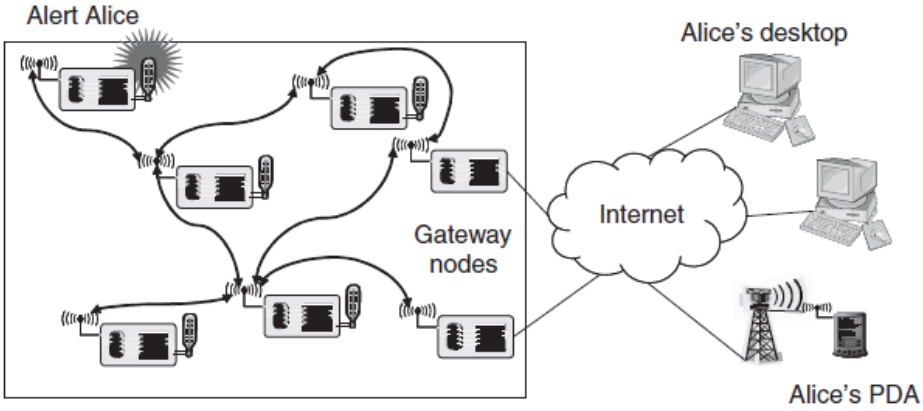



FIG:An event notification to "Alice" needs decisions about, among others, gateway choice, mapping "Alice" to a concrete IP address, and translating an intra-WSN event notification message to an Internet application message

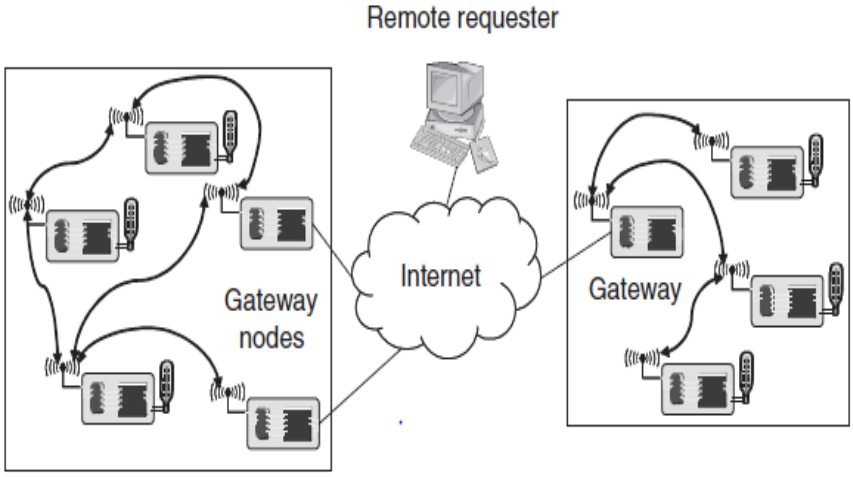



FIG Requesting sensor network information from a remote terminal entails choices about which network to address, which gateway node of a given network, and how and where to adapt application-layer protocol in the Internet to WSN-specific protocols

**Internet to WSN communication**

- Internet-based entity trying to access services of a WSN is even more challenging.
- This is fairly simple if this requesting terminal is able to directly communicate with the WSN.
- example, a mobile requester equipped with a WSN transceiver, and also has all the necessary protocol components at its disposal.
- In this case, the requesting terminal can be a direct
- part of the WSN and no particular treatment is necessary.
- however, a terminal "far away" requesting the service, not immediately able to communicate with any sensor node and thus requiring the assistance of a gateway node.

**how to find out that there actually is a sensor network in the desired location, and how to find out about the existence of a gateway node?Once the requesting terminal has obtained this information, how to access the actual services?**

- Addressing an individual sensor (like addressing a communication peer in a traditional Internet application) both goes against the grain of the sensor network philosophy where an individual sensor node is irrelevant compared to the data that it provides
- It is impossible if a sensor node does not even have an IP address.
- This assumes that there is an application-level protocol that a remote requester and gateway can use
- that is more suitable for communication over the Internet than the actual sensor network protocols &that is more convenient for the remote terminal to use.

- The gateway can then mask, for example, a data-centric data exchange within the network behind an identity-centric exchange used in the Internet
- an application-level protocol exists that represents an actual simplification over just extending the actual sensor network protocols to the remote terminal, but there are some indications in this direction.
- For example, it is not necessary for the remote terminal to be concerned with maintaining multi hop routes in the network nor should it be considered as "just another hop" as the characteristics of the Internet connection are quite different from a wireless hop.

**Web Service Protocols**

- There are some clear parallels for such an application-level protocol with so-called Web Service Protocols(WSP) .
- WSP can explicitly describe services and the way they can be accessed.
- The Web Service Description Language (WSDL) can be a promising starting point for extension with the required attributes for WSN service access –
- for example, required accuracy, energy trade-offs, or data-centric service descriptions.
- We can integrate WSN with general middleware architectures

to make WSN services accessible from a standard Web browser -----which should be an almost automatic by-product of using WSDL and related standards in the gateway

**WSN tunneling**

- Gateways can also act as simple extensions of one WSN to another WSN.
- The idea is to build a larger, "virtual" WSN out of separate parts, transparently "tunneling" all protocol messages between these two networks and simply using the Internet as a transport network .
- This can be attractive, but care has to be taken not to confuse the virtual link between two gateway nodes with a real link; otherwise, protocols that rely on physical properties of a communication link can get quite confused .
- (e.g. time synchronization or localization protocols).

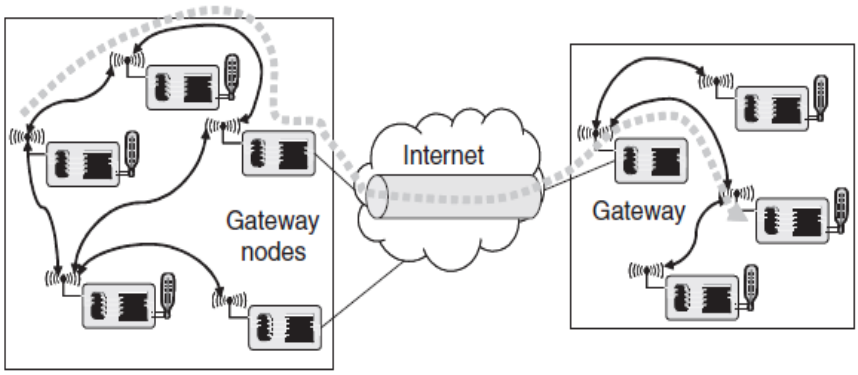



**Figure 3.12** Connecting two WSNs with a tunnel over the Internet

**Conclusion**

- wireless sensor networks and their networking architecture will have many different guises and shapes.
- Multi hop communication is the crucial enabling technology.
- WSN-specific forms of quality of service support, energy efficiency, scalability, and robustness – dominate the requirements for WSNs and have to be carefully arbitrated and balanced against each other.
- Distributed organization of the network, the use of in-network processing, a data-centric view of the network, and the adaptation of result fidelity and accuracy to given circumstances are pivotal techniques to be considered for usage.
- To allow Internet-based hosts a simple access to WSN services, is also still a fairly open problem

# • UNIT 3

# • Medium Access Control

## Wireless MAC Protocols

- MAC protocol is the first protocol layer above the Physical Layer .
- The primary task of any MAC protocol is to control the access of the nodes to shared medium.
- Medium Access Control (MAC) protocols coordinate the times where a number of nodes access a shared communication medium.
- Nodes in a wireless network share a common broadcast radio channel.
- Since the radio spectrum is limited, the bandwidth available for communication in such networks is also limited.
- Access to this shared medium should be controlled in such a manner that all nodes receive a fair share of the available bandwidth, and that the bandwidth is utilized efficiently.
- Characteristics of the wireless medium are completely different from wired medium.
- So a different set of protocols is required for controlling access to the shared medium in such networks.
- This is achieved by using Medium Access Control (MAC) protocol.
- 
- They differ, among others, in the types of media they use and in the performance requirements for which they are optimized.
- MAC protocols are heavily influenced by its properties.
- The fundamental task of any MAC protocol is to regulate the access of a number of nodes to a shared medium in such a way that certain application-dependent performance requirements are satisfied.
- **Why MAC is used in wireless network?**
- The MAC protocol in cellular networks is designed to maximize the utilization of the expensive licensed spectrum

**Media Access Control Address**.

- The MAC address is known as the hardware id number.
- In particular, each computer's NIC (network interface card), including a Bluetooth, Wi-Fi card or Ethernet card, has an unchanged MAC address inserted by the manufacturer at the time of production.

**Issues in Designing MAC Protocol forWireless sensorNetworks:**

1. Bandwidth Efficiency
2. Quality of Service Support
3. Synchronization:
4. Hidden and Exposed Terminal Problems
5. Mobility of Nodes
6. Error-Prone Shared Broadcast Channel
7. Distributed Nature/Lack of Central Coordination

**Bandwidth Efficiency**

- Since the radio spectrum is limited, the bandwidth available for communication is also very limited.
- The MAC protocol must be designed in such a way that to maximize bandwidth efficiency.
- bandwidth efficiency is the ratio of the bandwidth used for actual data transmission to the total available bandwidth
- That is the uncommon bandwidth is utilized in an efficient manner

**Quality of Service Support**

- Providing QoS support to data sessions in Ad-hoc networks is very difficult due to their characteristic nature of nodes mobility.
- Most of the time, Bandwidth reservation made at one point of time may become invalid once the node moves out of the region.
- The MAC protocol for Ad-hoc wireless networks that are to be used in such real-time applications must have resource reservation mechanism & take care of nature of the wireless channel and the mobility of nodes.
- **Synchronization:**
- The MAC protocol must take into consideration of the synchronization between nodes in the network.
- Synchronization is very important for bandwidth (time slot) reservations by nodes
- exchange of control packets may be required for achieving time synchronization among nodes.
- This control packets must not consume too much of network bandwidth.

**Hidden and Exposed Terminal Problems**

- The hidden node terminal problem
  - It refers to the collision of packets at a receiving node due to the simultaneous transmission of those nodes that are not with in the

transmission range of sender, but are with in the transmission range of receiver

- Collision occurs when both nodes transmits packet at same time without about the transmission of each other.
- The exposed terminal problem refers
- to the inability of a node,
- which is blocked due to transmission by a nearby transmitting node,
- to transmit to another node.

- Hidden and Exposed Terminal Problems significantly reduce the throughput of a network when traffic load is high.
- It is desirable that MAC protocol is free from Hidden and Exposed Terminal Problems

**Mobility of Nodes**

- This is a very important factor affecting the performance (throughput) of the protocol. Nodes is wireless network are mobile most of time.
- If node mobility is high,
- Bandwidth reservations mode or control packet exchange may no use.

The MAC protocol obviously has no role to play in influencing the mobility of the nodes.

**Error-Prone Shared Broadcast Channel**

- Due to broadcast nature of the radio channel ,transmissions made by a node are received by all nodes within its direct transmission range .
- When a node is transmitting the data, no other nodes in its neighbour-hood , apart from sender should transmit.
- Since multiple nods may contend for a channel simultaneously
- There is a possibility of packet collisions is quite high in wireless networks.
- A MAC protocol should grant channel access to nodes in such a manner that collisions are minimized.
- Protocol should ensure that all nodes are treated fairly with respect to bandwidth allocations.

**Distributed Nature/Lack of Central Coordination**

- wireless networks do not have centralized coordinators.
- centralized coordinators allocate bandwidth to mobile terminal .
- But it is not possible here because nodes keep moving continuously.
- Therefore, nodes must be scheduled in a distributed fashion for gaining access to the channel.
- The MAC protocol must make sure that the additional overhead, in terms of bandwidth consumption is not very high.

**problem associated with MAC protocol in WSN**

- In the design of MAC protocols for wireless sensor networks (WSN) it is necessary to fulfill some requirements such as low energy consumption, scalability, simplicity, etc.
- These requirements are not easy to fulfill from the viewpoint of implementation on FPGA or ASIC technologies**.**

**The MAC layer main functions are**

1. frame delimiting and recognition,
2. addressing,
3. transfer of data from upper layers,
4. error protection (generally using frame check sequences),
5. arbitration of access to one channel shared by all nodes

- **characteristics of MAC protocols are**

❖ Scalability,
❖ Adaptability
❖ Decentralization

**design issues of MAC protocol of WSN**

❖ node mobility
❖ an error- prone
❖ broadcast and shared channel
❖ time-synchronization
❖ bandwidth efficiency
❖ QoS support.

**performance metrics of MAC protocol**

- **The most common performance metrics are**

➢ delay and throughput,
➢ though in certain cases
➢ like sensor networks
➢ power consumption

**Objectives of MAC protocols**

❖ Collision avoidance
❖ Energy efficiency
❖ Scalability
❖ Latency
❖ Fairness
❖ Throughput
❖ Bandwidth utilization

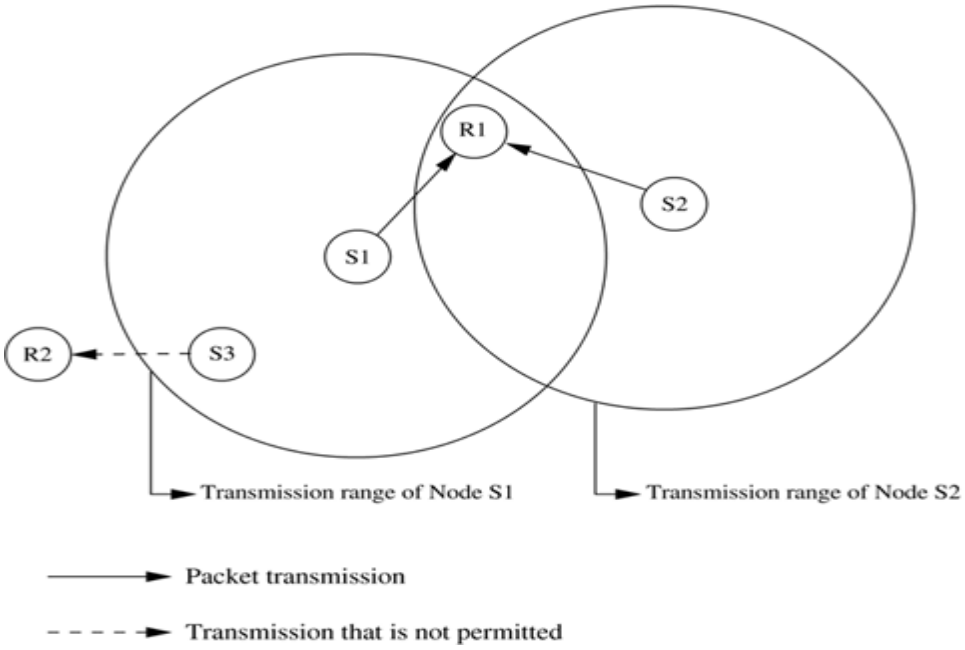
## 3.1ISSUES IN DESIGNING A MAC PROTOCOL FOR AD HOC

- **WIRELESS NETWORKS**
- The following are the main issues that need to be addressed while designing a MAC protocol for wireless networks.
- ***3.1.1 Bandwidth Efficiency***

- As mentioned earlier, since the radio spectrum is limited, the bandwidth available forcommunication is also very limited.
- The MAC protocol must be designed in such a way that thescarce bandwidth is utilized in an efficient manner.
- The control overhead involved must be keptas minimal as possible.
- Bandwidth efficiency can be defined as the ratio of the bandwidth usedfor actual data transmission to the total available bandwidth.
- The MACprotocol must try to maximize this bandwidth efficiency.

### *3.2.2 Quality of Service Support*

- Due to the inherent nature of the ad hoc wireless network, where nodes are usually mobile mostof the time, providing quality of service (QoS) support to data sessions in such networks is verydifficult.
- Bandwidth reservation made at one point of time may become invalid once the nodemoves out of the region where the reservation was made.
- QoS support is essential for supportingtime-critical traffic sessions such as in military communications.
- The MAC protocol for wireless networks that are to be used in such real-time applications must have some kind of aresource reservation mechanism that takes into consideration the nature of the wireless channel and the mobility of nodes.
- ***6.2.3 Synchronization***
- The MAC protocol must take into consideration the synchronization between nodes in thenetwork.
- Synchronization is very important for bandwidth (time slot) reservations by nodes.
- Exchange of control packets may be required for achieving time synchronization among nodes.
- The control packets must not consume too much of network bandwidth.
- ***3 .2.4 Hidden and Exposed Terminal Problems***
- The hidden and exposed terminal problems are unique to wireless networks.
- The hidden terminalproblem refers to the collision of packets at a receiving node due to the simultaneoustransmission of those nodes that are not within the direct transmission range of the sender, butare within the transmission range of the receiver
- . Collision occurs when both nodes transmitpackets at the same time without knowing about the transmission of each other.
- For example,consider Figure 6.1. Here, if both node S1 and node S2 transmit to node R1 at the same time,their packets collide at node R1. This is because both nodes S1 and S2 are hidden from each other as they are not within the direct transmission range of each other and hence do not knowabout the presence of each other.
- **Figure 6.1. Hidden and exposed terminal problems.**




- The exposed terminal problem refers to the inability of a node, which is blocked due totransmission by a nearby transmitting node, to transmit to another node. Consider the example in Figure 6.1.
- Here, if a transmission from node S1 to another node R1 is already in progress,node S3 cannot transmit to node R2, as it concludes that its neighbor node S1 is in transmitting mode and hence it should not interfere with the on-going transmission.
- The hidden and exposed terminal problems significantly reduce the throughput of a networkwhen the traffic load is high.
- It is therefore desirable that the MAC protocol be free from thehidden and exposed terminal problems.

### *3.2.5Error-Prone Shared Broadcast Channel*

- Another important factor in the design of a MAC protocol is the broadcast nature of the radiochannel, that is, transmissions made by a node are received by all nodes within its direct transmission range.
- When a node is receiving data, no other node in its neighborhood, apart from the sender, should transmit.
- A node should get access to the shared medium only when itstransmissions do not affect any ongoing session.
- Since multiple nodes may contend for thechannel simultaneously, the possibility of packet collisions is quite high in wireless networks.
- A MAC protocol should grant channel access to nodes in such a manner that collisions areminimized.
- Also, the protocol should ensure that all nodes are treated fairly with respect tobandwidth allocation.

### *3.2.6 Distributed Nature/Lack of Central Coordination*

- Ad hoc wireless networks do not have centralized coordinators.
- In cellular networks, forexample, the base stations act as central coordinating nodes and allocate bandwidth to the mobileterminals.
- But this is not possible in an ad hoc network, where nodes keep moving continuously.
- Therefore, nodes must be scheduled in a distributed fashion for gaining access to the channel.
- This may require exchange of control information.
- The MAC protocol must make sure that the additional overhead, in terms of bandwidth consumption, incurred due to this control informationexchange is not very high.

### *3.2.7 Mobility of Nodes*

- This is a very important factor affecting the performance (throughput) of the protocol.
- Nodes inan ad hoc wireless network are mobile most of the time.
- The bandwidth reservations made or thecontrol information exchanged may end up being of no use if the node mobility is very high.
- The MAC protocol obviously has no role to play in influencing the mobility of the nodes. Theprotocol design must take this mobility factor into consideration so that the performance of thesystem is not significantly affected due to node mobility.

## DESIGN GOALS OF A MAC PROTOCOL FOR AD HOC WIRELESS NETWORKS

- The following are the important goals to be met while designing a medium access control
- (MAC) protocol for ad hoc wireless networks:
- The operation of the protocol should be distributed.
- The protocol should provide QoS support for real-time traffic.
- The access delay, which refers to the average delay experienced by any packet to gettransmitted, must be kept low.
- The available bandwidth must be utilized efficiently.
- The protocol should ensure fair allocation (either equal allocation or weighted allocation) ofbandwidth to nodes.
- Control overhead must be kept as low as possible.
- The protocol should minimize the effects of hidden and exposed terminal problems.
- The protocol must be scalable to large networks.
- It should have power control mechanisms in order to efficiently manage energy consumption ofthe nodes.
- The protocol should have mechanisms for adaptive data rate control (adaptive rate controlrefers to the ability to control the rate of outgoing

traffic from a node after taking intoconsideration such factors as load in the network and the status of neighbor nodes).

- It should try to use directional antennas which can provide advantages such as reducedinterference, increased spectrum reuse, and reduced power consumption.
- Since synchronization among nodes is very important for bandwidth reservations, the protocolshould provide time synchronization among nodes.

# 3.2 Characteristics of MAC Protocols in WSNs

Most MAC protocols are built for fairness everybody should get an equal amount of resourcesλ no one should receive special treatmentλ In a WSN, all nodes cooperate to achieve a common purpose, thereforeν fairness is less of a concern Instead, wireless nodes are mostly concerned with energy consumptionν Sensing applications may value low latency or high reliability over fairness

**Energy Efficiency**

Sensor nodes must operate using finite energy sources, therefore MAC protocols must consider energy efficiency Common technique: dynamic power management (DPM)ν a resource can be moved between different operational modes such asλ active, idle, and asleep for resources such as the network, the active mode can group togetherλ multiple different modes of activity, e.g., transmitting and receiving Periodic traffic models are very common in WSNsν significant energy savings can be obtained by putting a device into aλ low-power sleep mode fraction of time a sensor nodes spends in active mode is called the dutyλ cycle often very small due to the infrequent and brief data transmissions4 occurring in most sensor networks

Reasons for energy inefficiency idle listeningλ inefficient protocol designs (e.g., large packet headers)λ reliability features (collisions requiring retransmissions or other errorλ control mechanisms) control messages to address the hidden-terminal problemλ choice of modulation schemeλ choice of transmission rateλoveremittingλ

## Scalability73

Many wireless MAC protocols have been designed for use in infrastructurebased networks access points or controller nodes arbitrate access to the channel andλ perform some centralized coordination and management functions Most wireless sensor networks rely on multi-hop and peer-to-peerν communications without centralized coordinators

MAC protocols must be able to allow for efficient use of resources withoutν incurring unacceptable overheads, particularly in very large networks MAC protocols based on CDMA have to cache a large number of codeν (may be impractical for resource-constrained sensor devices) WSNs are not only constrained in their energy resources, but also in theirν processing and memory capacities Therefore, MAC protocols should not impose excessive computationalν burden should not require too much memory to save state information

## Adaptability

A key characteristic of a WSN is its ability to self-manage adapt to changes in the networkλ including changes in topology, network size, density, and trafficλ characteristics A MAC protocol for a WSN should be able to gracefully adapt to suchν changes without significant overheads This requirement generally favors protocols that are dynamic in natureν protocols that make medium access decisions based on currentλ demand and network state Protocols with fixed assignments (e.g., TDMA with fixed-size frames andν slots) may incur large overheads due to adaptations of such assignments that may affect many or all nodes in the network

## Low Latency and Predictability

Many WSN applications have timeliness requirements sensor data must be collected, aggregated, and delivered within certainλ latency constraints or deadlines example: wildfire detection (sensor data must be delivered to monitoringλ stations in a timely fashion to ensure timely responses) MAC protocol designν choice of frame size and slot allocations in TDMA-based protocols mayλ lead to large delays in contention-based protocols, nodes may be able to access theλ wireless medium sooner (than TDMA), but collisions and the resulting retransmissions incur delays choice of MAC protocol can affect how predictable the experiencedλ delay is (expressed as upper latency bounds) some contention-based MAC protocols allow the theoretical possibilityλ of starvation

## Reliability

Common requirement for WSNs MAC protocol design affects reliabilityν responsible for detecting and recovering from transmission errors andλ collisions

## Classification of MAC protocols

MAC is the first protocol layer above the Physical Layer in ad hoc .The primary task of any MAC protocol is to control the access of the nodes to shared medium.

**Classification of MAC protocols :**
These are as following below.

1. **Contention-based protocols without reservation/scheduling –**
   - Bandwidth are not reserved.
   - No guarantees.
2. **Contention-based protocols with reservation mechanisms –**
   - Bandwidth is reserved for transmission.
   - Guarantees can be given.
3. **Contention-based protocols with scheduling mechanisms –**
   - Distributed scheduling is done between nodes.
   - Guarantees can be given.
4. **Other protocols –**
   - Combine multiple features of other protocols.
   - It can also use a completely new approach

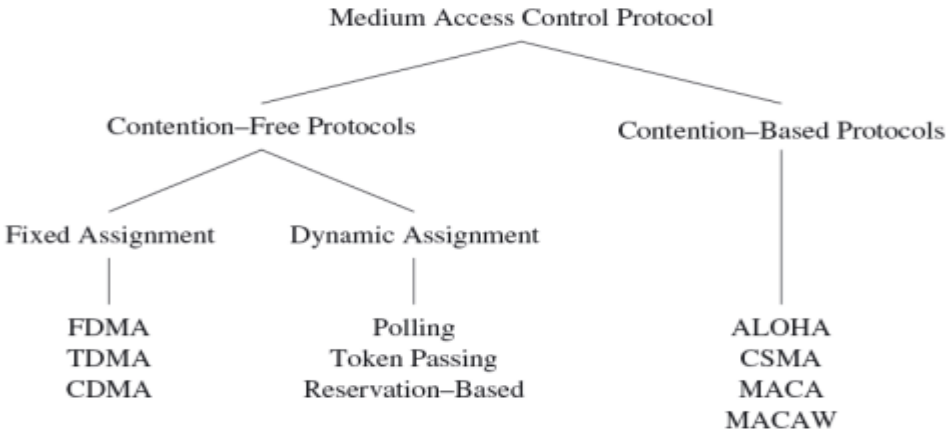
**Contention-based protocols without reservation/scheduling**

- Sender-initiated protocols:
  The transmission of packets are initiated by the sender node.
  - Single-channel sender initiated. For example, MACAW, FAMA.
  - Multiple-channel sender initiated protocols. For example, BTMA, DBTMA, ICSMA.
- Receiver-initiated protocols:
  The connection is initiated by the receiver node. For example, RI-BTMA, MACA-BI, MARCH.

**Contention-based protocols with reservation mechanisms**

- Synchronous protocols:
  All node are kept synchronized. For example, D-PRMA, CATA, HRMA, SRMA/PA, FPRP.
- Asynchronous protocols:
  Relative time information is used to achieve effecting reservations. For example, MACA/PR, RTMAC..

## MAC Protocol Categorization




**Contention-Free MAC Protocols**

Concept:

- allow only one sensor node to access the channel at any given time
- thereby avoiding collisions and message retransmissions
- assuming a perfect medium and environment
- i.e., no other competing networks or misbehaving devices exist that could otherwise cause collisions or even jam a channel
- Contention-free protocols allocate resources to individual nodes to ensureν exclusive resource access by only one node at any given time
- Exposes a number of desirable characteristicsν node knows exactly when it has to turn on its radioduring all other times, radio can be turned off to preserve energy
- fixed slot allocations impose upper bounds on delay
- difficult to design schedules for large networks
- difficult to handle changes in topology, density, traffic load

**Traffic-Adaptive Medium Access**

- TRAMA is an example of a contention-free MAC protocol with the goal to
- increase network throughput and energy efficiency (compared to TDMA)
- It uses a distributed election scheme to determine when nodes are allowedν to transmit based on information about the traffic at each node
- avoids assigning slots to nodes with no traffic to send (increasedλthroughput) allows nodes to determine when they can become idle and do not have

- to listen to the channel (increased energy efficiency)
- TRAMA assumes a time-slotted channel, where time is dived into: periodic random-access intervals (signaling slots)
- scheduled-access intervals (transmission slots)

Random-access intervals

- Neighbor Protocol (NP) is used to propagate one-hop neighbor
- information among neighboring nodesjoin a network by transmitting during a randomly selected slot
- packets transmitted during these slots are used to gather4 neighborhood information by carrying a set of added and deleted neighbors in case no changes have occurred, these packets serve as "keepalive" beacons
- NP allows nodes to obtain consistent two-hop topology information

Schedule Exchange Protocol (SEP) establishes and broadcasts actual schedules (i.e., allocations of slots to a node) each node computes a duration SCHEDULE_INTERVAL4 – represents the number of slots for which the node can announce its schedule to its neighbors – this duration depends on the rate at which the node's applications can produce packets at time t, the node then computes the number of slots within [t, t+4 SCHEDULE_INTERVAL] for which it has the highest priority among its two-hop neighbors the node announces the selected slots and the intended receivers4 using a schedule packet the last slot in this schedule is used to announce the next schedule4 for the next interval

Schedule Exchange Protocol (SEP)

– a node's SCHEDULE_INTERVAL is 100 slots – the current time (slot number) is 1000 – a possible slot selection for interval [1000, 1100] for this node could be 1011, 1021, 1049, 1050, and 1093 – during slot 1093, the node broadcasts its new schedule for interval [1093, 1193] list of intended receivers in the schedule packet is implemented as a4 bitmap – length of a bitmap is equal to the number of one-hop neighbors » each bit in the bitmap corresponds to one particular receiver ordered by its identities » every node can determine the receiver address based on the bitmap and its list of neighbors

- slot selection is based on the node's priority at time t
- – uses a pseudo-random hash of the concatenation of the node's identity i and t: – node can indicate which slots it gives up, allowing other nodes to claim these unused slots a node can determine its state for any given time slot t based on its4
- two-hop neighborhood information and the announced schedules – node i is in the transmit (TX) state if it has the highest priority and if it has data to send – node i is in the receive (RX) state if it is the intended receiver of the transmitter during slot t – otherwise, the node can be switched into the sleep (SL) state

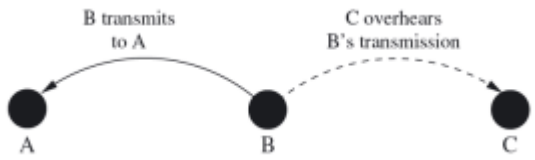
- compared to CSMA-based protocolsλ reduces the probability of collisions
- increases the sleep time and energy savings
- unlike standard TDMA approaches
- TRAMA divides time into random-access and scheduled-access
- intervals during the random-access intervals
- – nodes are awake to either transmit or receive topology information – the length of the random-access interval affects the overall duty cycle and achievable energy savings of a node

**Contention-Based MAC Protocols**

- These protocols do not rely on transmission schedules, instead they requireν other mechanisms to resolve contention when it occurs
- The main advantage of contention-based techniques is their simplicityν
- compared to most schedule-based techniques schedule-based MAC protocols must save and maintain schedules or
- tables indicating the transmission order most contention-based protocols do not require to save, maintain, or
- share state information this also allows contention-based protocols to adapt quickly to changesλ in network topologies or traffic characteristics However, they typically result in higher collision rates and overheads due toν idle listening and overhearing (overheads usually refer to additional bits in a packet or additional packets such as control packets) They may also suffer from fairness issues (i.e., some nodes may be able toν obtain more frequent channel accesses than others)

**Power Aware Multi-Access with Signaling**

The PAMAS protocol attempts to avoid unnecessary energy expenditure caused by overhearing PAMAS uses two separate channels to prevent collisions among datav transmissions: one for data frames and one for control frames (ready-tosend or RTS and clear-to-send or CTS)




The separate signaling channel allows nodes to determine when and howν long to power down their wireless transceivers devices transmit busy tones on the control channel to ensure thatλ devices that did not overhear either RTS or CTS will not access the data channel for transmissions

Initiating a data transfer  a PAMAS device sends an RTS message over the control channel toλ the receiver  receiver responds with CTS if it does not detect activity on the dataλ channel and has not overheard other recent RTS or CTS messages  if the source does not receive a CTS within a specific timeout interval, itλ will attempt to transmit again after a back-off time (exponential back-off algorithm)  otherwise, it begins data transmission and the receiver node issues aλ busy tone over the control channel (whose length is greater than twice the length of a CTS)  the receiver device also issues a busy tone over the control channelλ whenever it receives an RTS message or detects noise on the control channel while it receives a frame done to corrupt possible CTS message replies to the detected RTS,4 thereby blocking any data transmission of the receiver's neighbors

- Every node in a PAMAS network independently decides when to power off
- its transceiver  Specifically, a node decides to turn off its transceiver whenever one of two conditions holds:  a neighbor begins a transmission and the node has no frames to transmit  a neighbor transmits a frame to another neighbor, even if the node has frames to transmit
- A node can easily detect either condition byν  overhearing its neighbor's transmissions (condition 1) or
- overhearing its neighbor's busy tone (condition 2)
- A node can identify how long to power down its transceiver by embeddingν the size or expected transmission duration into messages

However, when a transmission begins while a node is still asleep, it doesν not know how long this transmission will last and how long it should continue to sleep  Therefore, the node issues a probe frame (containing a certain time interval)ν over the control channel to all transmitting nodes in its neighborhood all transmitters that will complete during this interval respond with theirλ predicted completion time  if such a response is received by the awakening node without collision,λ it can return to the sleep mode until the completion time indicated by the transmitting node  if multiple transmitters respond and their responses collide, the nodeλ reissues the probe frame with a shorter time interval  similarly, if the node did not receive a response, it can reissue the probeλ with a different time interval  in effect, the node chooses time intervals to perform a binary search toλ identify when the last ongoing transmission will end

## Contention-Free Medium Access

- Collisions can be avoided by ensuring that each node can use its allocated resources exclusively

Examples of fixed assignment strategies

- FDMA: Frequency Division Multiple Access
- the frequency band is divided into several smaller frequency bands the data transfer between a pair of nodes uses one frequency band all other nodes use a different frequency band
- TDMA: Time Division Multiple Access
  - multiple devices to use the same frequency band4 relies on periodic time windows (frames)4 – frames consist of a fixed number of transmission slots to separate the medium accesses of different devices – a time schedule indicates which node may transmit data during a certain slot

simultaneous accesses of the wireless medium are supported using CDMA: Code Division Multiple Access

Examples of fixed assignment strategies (contd.): ν different codes if these codes are orthogonal, it is possible for multiple4 communications to share the same frequency band forward error correction (FEC) at the receiver is used to recover from4 interferences among these simultaneous communications it is impossible to reallocate slots belonging to one device to otherλ Fixed assignment strategies are inefficient ν these schedules may require modifications every time the network4generating schedules for an entire network can be a taunting task 4devices if not needed in every frame topology or traffic characteristics in the network change

**Dynamic assignment strategies**: allow nodes to access the medium on demand ν asking each station if it has data to send if no data to be sent, the controller polls the next station

stations pass a polling request to each other (round-robin fashion) using a4 token passing special frame called a token a station is allowed to transmit data only when it holds the token

e.g., a node can indicate its desire to transmit data by toggling a reservation4static time slots used to reserve future access to the medium 4 reservation-based protocols λ bit in a fixed location these often very complex protocols then ensure that other potentially4 conflicting nodes take note of such a reservation to avoid collisions

Nodes may initiate transmissions at the same timeν
requires mechanisms to reduce the number of collisions and to recoverλ Nodes may initiate transmissions at the same time ν from collisions uses acknowledgments to confirm the success of a broadcast dataλ Example 1: ALOHA protocol ν addresses collisions with approaches such as exponential back-off4allows nodes to access the medium immediately 4transmission to increase the likelihood of successful transmissions requires that a station may commence transmission only at predefinedλ Example 2: slotted-ALOHA protocol ν points in time (the beginning of a time slot) introduces the need for synchronization among nodesλ increases the efficiency of ALOHA λ introduces the need for synchronization among nodes

Carrier Sense Multiple Access (CSMA) CSMA with Collision Detection (CSMA/CD)λ sender first senses the medium to determine whether it is idle or4 busy – if it is found busy, the sender refrains from transmitting packets – if the medium is idle, the sender can initiate data transmission CSMA with Collision Avoidance (CSMA/CA)λ CSMA/CD requires that sender aware of collisions4 instead, CSMA/CA attempts to avoid collisions in the first place4

Hidden-terminal problem senders A and C are able to reach B, but cannot overhear each other's signals it is possible for A and C to transmit data to B at the same time, causingλ a collision at B, without being able to directly detect this collision Exposed-terminal problemν C wants to transmit data D, but decides to wait because it overhears anλ ongoing transmission from B to A B's transmission could not interfere with data reception at Cλ

**Carrier Sense Multiple Access**
Nodes first sense the medium before they begin a transmission (reduces number of collisions) Non-persistent CSMAν node is allowed to immediately transmit data once medium is idleλ if the medium is busy, the node performs a back-off operationλ wait for a certain amount of time before attempting to transmit again4 1-persistent CSMAν node wishing to transmit data continuously senses the medium forλ activity once the medium is found idle, the node transmits data immediatelyλ if a collision occurs, the node waits for a random period of time beforeλ attempting to transmit again

p-persistent CSMA node continuously senses the mediumλ node transmits data with a probability p once the medium becomes idleλ delays transmission with a probability 1 − pλ random back-off values are either continuous values

in the case of unslotted CSMA or multiples of a fixed slot size in slotted CSMAλ CSMA/CA (CSMA with Collision Avoidance)ν nodes sense the medium, but do not immediately access the channelλ when it is found idle instead, a node waits for a time period called DCF interframe spaceλ (DIFS) plus a multiple of a slot size in case there are multiple nodes attempting to access the medium, theλ one with the shorter back-off period will win

Example: node A waits for DIFS + 4 ∗ s (where s represents the slot size), whileλ node B's back-off is DIFS + 7 ∗s once node A begins with its transmission, node B freezes its own backoff timer and resumes the timer after node A completes its transmissionλ plus another period of DIFS once node B's back-off timer expires, it can also begin its transmissionλ

## MACA and MACAW

Multiple Access with Collision Avoidance (MACA) dynamic reservation mechanismλ sender indicates desire to send with ready-to-send (RTS) packetλ intended receiver responds with clear-to-send (CTS) packetλ if sender does not receive CTS, it will retry at later point in timeλ nodes overhearing RTS or CTS know that reservation has taken placeλ and must wait (e.g., based on the size of data transmission) address hidden terminal problem and reduces number of collisionsλ MACA for Wireless LANs (MACAW)ν receiver responds with acknowledgment (ACK) after data receptionλ other nodes in receiver's range learn that channel is available4 nodes hearing RTS, but not CTS do not know if transmission will occurλ MACAW uses data sending (DS) packet, sent by sender after4 receiving CTS to inform such nodes of successful handshake

In MACA-BI, destination device initiates data transfers by sending a Ready To Receive (RTR) packet to the source source then responds with the data messageλ Compared to MACA, MACA-BI reduces overheadν increases the theoretical maximum throughputλ depends on the destination knowing when to receive dataλ Source nodes can use an optional field within the data message to indicateν the number of queued messages providing the destination with an indication that more RTS packets willλ be required

### *Contention-Based Protocols*

These protocols follow a contention-based channel access policy. A node does not make anyresource reservation *a priori*. Whenever it receives a packet to be transmitted, it contends with itsneighbor nodes for access to the shared channel. Contention-based protocols cannot provide QoSguarantees to sessions since

nodes are not guaranteed regular access to the channel. Randomaccess protocols can be further divided into two types:

• Sender-initiated protocols: Packet transmissions are initiated by the sender node.

• Receiver-initiated protocols: The receiver node initiates the contention resolution protocol.

Sender-initiated protocols can be further divided into two types:

• Single-channel sender-initiated protocols: In these protocols, the total available bandwidth isused as it is, without being divided. A node that wins the contention to the channel can make useof the entire bandwidth.

• Multichannel sender-initiated protocols: In multichannel protocols, the available bandwidth isdivided into multiple channels. This enables several nodes to simultaneously transmit data, eachusing a separate channel. Some protocols dedicate a frequency channel exclusively fortransmitting control information.

***Contention-Based Protocols with Reservation Mechanisms***

Ad hoc wireless networks sometimes may need to support real-time traffic, which requires QoSguarantees to be provided. In contention-based protocols, nodes are not guaranteed periodicaccess to the channel. Hence they cannot support real-time traffic. In order to support suchtraffic, certain protocols have mechanisms for reserving bandwidth *a priori*Such protocols canprovide QoS support to time-sensitive traffic sessions.

These protocols can be further classifiedinto two types:

• Synchronous protocols: Synchronous protocols require time synchronization among all nodesin the network, so that reservations made by a node are known to other nodes in its
neighborhood. Global time synchronization is generally difficult to achieve.

• Asynchronous protocols: They do not require any global synchronization among nodes in thenetwork. These protocols usually use relative time information for effecting reservations.

***Contention-Based Protocols with Scheduling Mechanisms***

these protocols focus on packet scheduling at nodes, and also schedulingnodes for access to the channel. Node scheduling is done in a manner so that all nodes are treatedfairly and no node is starved of bandwidth. Scheduling-based schemes are also used for enforcing priorities among flows whose packets are queued at nodes. Some scheduling schemesalso take into consideration battery characteristics, such as remaining battery power, whilescheduling nodes for access to the channel.

***Other Protocols***

There are several other MAC protocols that do not strictly fall under the above categories.

## CONTENTION-BASED PROTOCOLS

Contention-based protocols do not have any bandwidth reservationmechanisms. All ready nodes contend for the channel simultaneously, and the winning nodegains access to the channel. Since nodes are not guaranteed bandwidth, these protocols cannot beused for transmitting real-time traffic, which requires QoS guarantees from the system. In thissection, several contention-based MAC protocols are described in detail.

### *MACAW: A Media Access Protocol for Wireless LANs*

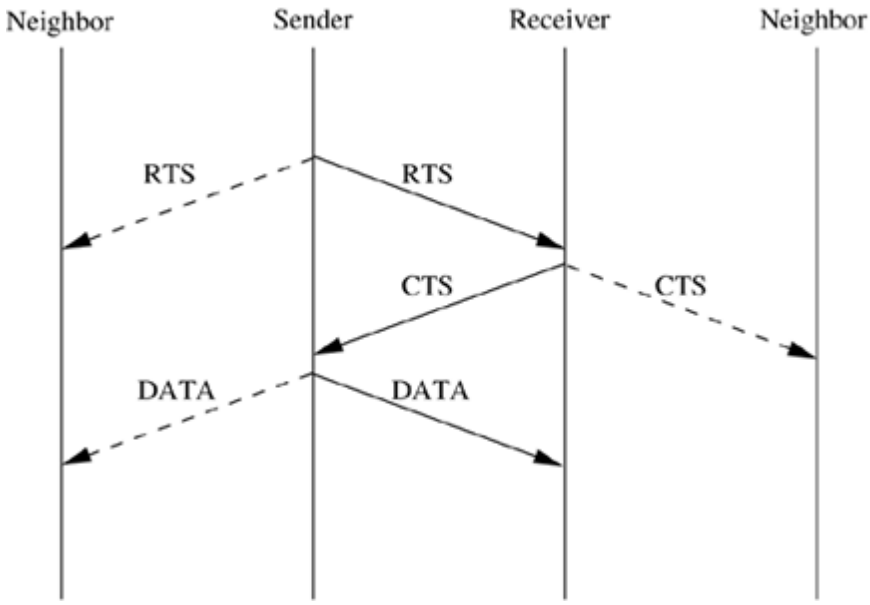
This protocol is based on the multiple access collision avoidance protocol (MACA) proposed byKarn . MACA was proposed due to the shortcomings of CSMA protocols when used forwireless networks. In what follows, a brief description on why CSMA protocols fail in wirelessnetworks is given. This is followed by detailed descriptions of the MACA protocol and the**MACAW protocol.**

**MACA** Protocol

- The MACA protocol was proposed as an alternative to the traditional carrier sense multiple
- access (CSMA) protocols used in wired networks.
- In CSMA protocols, the sender first senses thechannel for the carrier signal. If the carrier is present, it retries after a random period of time.
- Otherwise, it transmits the packet. CSMA senses the state of the channel only at the transmitter.
- This protocol does not overcome the hidden terminal problem.
- In a typical ad hoc wirelessnetwork, the transmitter and receiver may not be near each other at all times.

In such situations,the packets transmitted by a node are prone to collisions at the receiver due to simultaneoustransmissions by the hidden terminals.Also, the bandwidth utilization inCSMA protocols is lessbecause of the exposed terminal problem.MACA does not make use of carrier-sensing for channel access. It uses two additional signalingpackets: the request-to-send (RTS) packet and the clear-to-send (CTS) packet. When a nodewants to transmit a data packet, it first transmits an RTS packet. The receiver node, on receivingthe RTS packet, if it is ready to receive the data packet, transmits a CTSpacket.

Once the senderreceives the CTS packet without any error, it starts transmitting the data packet. This datatransmission mechanism is depicted in below Figure .

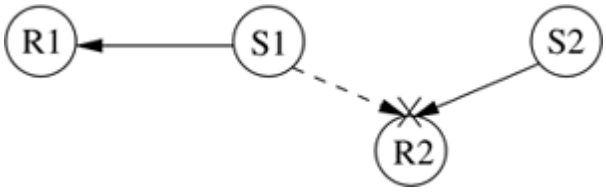



MACAW Protocol

If a packet transmitted by a node is lost, the node uses the binary exponential back-off (BEB) algorithm to back off for a random interval of time before retrying. In the binary exponential backoff mechanism, each time a collision is detected, the node doubles its maximum back-off window. Neighbor nodes near the sender that hear the RTS packet do not transmit for a long enough period of time so that the sender could receive the CTS packet. Both the RTS and theCTS packets carry the expected duration of the data packet transmission. A node near the receiver, upon hearing the CTS packet, defers its transmission till the receiver receives the data packet. Thus, MACA overcomes the hidden node problem. Similarly, a node receiving anRTS defers only for a short period of time till the sendercould receive the CTS. If no CTS is heard by the node during its waiting period, it is free totransmit packets once the waiting interval is over. Thus, a node that hears only the RTS packet isfree to transmitsimultaneously when the sender of the RTS is transmitting data packets. Hence, the exposed terminal problem is also overcome in MACA. But MACA still has certain problems,which was why MACAW, described below, was proposed.

The binary exponential back-off mechanism used in MACA at times starves flows. For example, consider Figure 3.4.

**Figure 3.4. Example topology.**




. Here both nodes S1 and S2 keep generating a high volume of traffic. Thenode that first captures the channel (say, node S1) starts transmitting packets. The packetstransmitted by the other node S2 get collided, and the node keeps incrementing its back-offwindow according to the BEB algorithm. As a result, the probability of node S2 acquiring thechannel keeps decreasing, and over a period of time it gets completely blocked.
To overcomethis problem, the back-off algorithm has been modified in MACAW . The packet header nowhas an additional field carrying the current back-off counter value of the transmitting node. Anode receiving the packet copies this value into its own back-off counter. This mechanismallocates bandwidth in a fair manner. Another problem with BEBalgorithm is that it adjusts theback-off counter value very rapidly, both when a node successfully transmits a packet and whena collision is detected by the node. The back-off counter is reset to the minimum value afterevery successful transmission. In the modified back-off process, this would require a period ofcontention to be repeated after each successful transmission in order to build up the back-offtimer values. To prevent such large variations in the back-off values, amultiplicative increaseand linear decrease (MILD) back-off mechanism is used in MACAW. Here, upon a collision, theback-off is increased by a multiplicative factor (1.5), and upon a successful transmission, it isdecremented by one. This eliminates contention and hence long contention periods after everysuccessful transmission, at the same time providing a reasonably quick escalation in the back-offvalues when the contention is high.

In MACAW another modification related to the back-off mechanism has been made. MACAWimplements per flow fairness as opposed to the per node fairness in MACA. This is done bymaintaining multiple queues at every node, one each for each data stream, and running the backoffalgorithm independently for each queue. A node that is ready to transmit packets firstdetermines how long it needs to wait before it could transmit an RTS packet to each of thedestination nodes corresponding to the top-most packets in the node's queues. It then selects thepacket for which the waiting time is minimal.
In addition to the RTS and CTS control packets used in MACA, MACAW uses another newcontrol packet called acknowledgment (ACK) packet. The need for using this additional packetarises because of the following reason. In MACA, the responsibility of recovering fromtransmission errors lies with the transport layer. As many TCP implementations have a minimumtimeout period of about

0.5 sec, significant delay is involved while recovering from errors. But inMACAW, the error recovery responsibility is given to the data link layer (DLL).

In DLL, theRecoveryrecovery process can be made quicker as the timeout periods can be modified in order to suit thephysical media being employed. In MACAW, after successful reception of each data packet, thereceiver node transmits an ACK packet. If the sender does not receive the ACK packet, itreschedules the same data packet for transmission. The back-off counter is incremented if theACK packet is not received by the sender. If the ACK packet got lost in transmission, the senderwould retry by transmitting an RTS for the same packet. But now the receiver, instead of sendingback a CTS, sends an ACK for the packet received, and the sender moves on to transmit the nextdata packet.In MACA, an exposed node (which received only the RTS and not the CTS packet) is free totransmit simultaneously when the source node is transmitting packets.

**Figure 3.5. Example topology.**


For example, inFigure 3.5,
when a transmission is going on between nodes S1 and R1, node S2 is free totransmit. RTS transmissions by node S2 are of no use, as it can proceed further only if it canreceive a CTS from R2. But this is not possible as CTS packets from R2 get collided at node S2with packets transmitted by node S1. As a result, the back-off counter at node S2 builds up
unnecessarily. So an exposed node should not be allowed to transmit. But an exposed node, sinceit can hear only the RTS sent by the source node and not the CTS sent by the receiver, does notknow for sure whether the RTS-CTS exchange was successful.

To overcome this problem,MACAW uses another small (30-bytes) control packet called the data-sending (DS) packet.Before transmitting the actual data packet, the source node transmits this DS packet.The DS packet carries information such as the duration of the data packet transmission, whichcould be used by the exposed nodes for updating information they hold regarding the duration ofthe data packet transmission. An exposed node, overhearing the DS packet, understands that theprevious RTS-CTS exchange was successful, and so defers its transmissions until the expectedduration of the DATA-ACK exchange. If the DS packet was not used, the exposed node (nodeS2) would retransmit after waiting for random intervals of time, and with a high probability thedata transmission (between nodes S1 and R1) would be still going on when the exposed noderetransmits. This would result in a collision and the back-off period being further incremented,which affects the node even

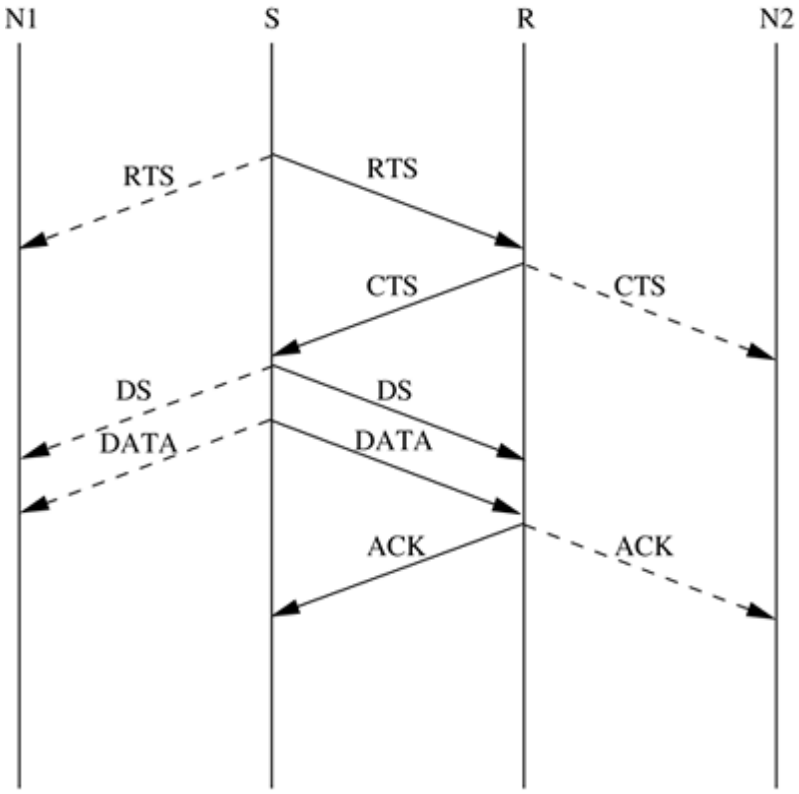
more.The MACAW protocol uses one more control packet called the request-for-request-to-send(RRTS) packet. The following example shows how this RRTS packet proves to be useful.

**Figure 3.6. Example topology.**


Consider Figure 3.6. Here assume transmission is going on between nodes S1 and R1. Now nodeS2 wants to transmit to node R2. But since R2 is a neighbor of R1, it receives CTSpackets fromnode R1, and therefore it defers its own transmissions. Node S2 has no way to learn about thecontention periods during which it can contend for the channel, and so it keeps on trying,incrementing its back-off counter after each failed attempt. Hence the main reason for thisproblem is the lack of synchronization information at source S2. MACAW overcomes thisproblem by using the RRTS packet. In the same example shown in Figure 3.6, receiver node R2contends for the channel on behalf of source S2. If node R2 had received anRTS previously forwhich it was not able to respond immediately because of the on-going transmission betweennodes S1 and R1, then node R2 waits for the next contention period and transmitsthe RRTS packet. Neighbor nodes that hear the RRTS packet (including node R1) are made towait for two successive slots (for the RTS-CTS exchange to take place). The source node S2, onreceiving the RRTS from node R2, transmits the regular RTS packet to node R2, and the normalpacket exchange (RTS-CTS-Data-ACK) continues from here.

Figure 3.7 shows the operation ofthe MACAW protocol. In the figure, S is the source node and R denotes the receiver node. N1and N2 are neighbor nodes. When RTS transmitted by node S is overheard by node N1, itrefrains from transmitting until node S receives the CTS. Similarly, when the CTS transmitted bynode R is heard by neighbor node N2, it defers its transmissions until the data packet is receivedby receiver R. On receiving this CTS packet, node S immediately transmits the DS messagecarrying the expected duration of the data packet transmission. On hearing this packet, node N1backs off until the data packet is transmitted. Finally, after receiving the data packet, node R acknowledges the reception by sending node S an ACK packet.

Figure 6.7. Packet exchange in MACAW




To summarize, the MACAW protocol has been designed based on four main observations. Thefirst is that the relevant congestion occurs at the receiver node and not at the sender. Thisrealization makes CSMA protocols unsuitable for ad hoc wireless networks, and thereforethe RTS-CTS-DATA exchangemechanism of MACA becomes necessary. MACAW furtherimproves upon this scheme using the RTS-CTS-DS-DATA-ACK exchange mechanism. Thesecond observation is that congestion is dependent on the location of the receiver. Therefore,instead of characterizing back-off by a single back-off parameter, separate back-off parametershave been introduced for each flow. The third is that learning about congestion at various nodesmust be a collective enterprise. Therefore, the notion of copying back-off values from overheard packets has been introduced in MACA. And the final observation is that in order that nodescontend effectively for the channel, the synchronization information needs to be propagated tothe concerned nodes at appropriate times. This is done in MACAW throughthe DS and RRTS packets. Because of the various changes described above, the performance ofMACAW is significantly improved when compared to the MACA protocol.

### 3.2.1 *Floor Acquisition Multiple Access Protocols*

The floor acquisition multiple access (FAMA) protocols are based on a channel accessdiscipline which consists of a carrier-sensing operation and a collision-avoidance dialog betweenthe sender and the intended receiver of a packet. Floor acquisition refers to the process of gainingcontrol of the channel. At any given

point of time, the control of the channel is assigned to onlyone node, and this node is guaranteed to transmit one or more data packets to different destinations without suffering from packet collisions. Carrier-sensing by the sender, followed bythe RTS-CTS control packet exchange, enables the protocol to perform as efficientlyas MACA in the presence of hidden terminals, and as efficientlyas CSMA otherwise. FAMA requires a node that wishes to transmit packets to first acquire thefloor (channel) before starting to transmit thepackets. The floor is acquired by means ofexchanging control packets. Though the control packets themselves may collide with othercontrol packets, it is ensured that data packets sent by the node that has acquired the channel are always transmitted without any collisions. Any single-channel MAC protocol that does notrequire a transmitting node to sense the channel can be adapted for performing floor acquisitiontasks. Floor acquisition using the RTS-CTS exchange is advantageous as the mechanism alsotries to provide a solution for the hidden terminal problem.Two FAMA protocol variants are discussed in this section: RTS-CTS exchange with no carriersensing,and RTS-CTS exchange with non-persistent carrier-sensing. The first variant uses theALOHA protocol for transmitting RTS packets, while the second variant uses nonpersistentCSMA for the same purpose.

Multiple Access Collision Avoidance

Multiple access collision avoidance (MACA) belongs to the category of FAMA protocols. In MACA, a ready node transmits an RTS packet. Aneighbor node receiving the RTS defers its transmissions for the period specified in theRTS. Onreceiving the RTS, the receiver node responds by sending back a CTS packet, and waits for along enough period of time in order to receive a data packet. Neighbor nodes of the receiverwhich hear this CTS packet defer their transmissions for the time duration of the impending datatransfer. In MACA, nodes do not sense the channel. A node defers its transmissions only if it receives an RTS or CTS packet. In MACA, data packets are prone to collisions with RTS packets.

According to the FAMA principle, in order for data transmissions to be collision-free, theduration of an RTS must be at least twice the maximum channel propagation delay.Transmission of bursts of packets is not possible in MACA. In FAMA-NTR the MACA protocol is modified to permittransmission of packet bursts by enforcing waitingperiods on nodes, which are proportional to the channel propagation time.

FAMA – Non-Persistent Transmit Request

This variant of FAMA, called FAMA – non-persistent transmit request (FAMA-NTR), combinesnon-persistent carrier-sensing along with the RTS-CTS control packet exchange mechanism.Before sending a packet, the sender node senses the channel. If the channel is found to be busy,then the node backs off for a

random time period and retries later. If the channel is found to befree, it transmits the RTS packet. After transmitting the RTS, the sender listens to the channel forone round-trip time in addition to the time required by the receiver node to transmit a CTS. If itdoes not receive the CTS within this time period or if theCTS received is found to be corrupted,then the node takes a random back-off and retries later. Once the sender node receivesthe CTS packet without any error, it can start transmitting its data packet burst. The burst is limited to a maximum number of data packets, after which the node releases the channel, andcontends with other nodes to again acquire the channel.

In order to allow the sender node to send a burst of packets once it acquires the floor, the receivernode is made to wait for a time duration of $\tau$ seconds after processing each data packet received.

Here, $\tau$ denotes the maximum channel propagation time. A waiting period of $2\tau$ seconds isenforced on a transmitting node after transmitting any control packet. This is done to allowthe RTS-CTS exchange to take place without any error. A node transmitting anRTS is requiredto wait for $2\tau$ seconds after transmitting the RTS in order to enable the receiver node to receivethe RTS and transmit the corresponding CTS packet. After sending the final data packet, asender node is made to wait for $\tau$ seconds in order to allow the destination to receive the datapacket and to account for the enforced waiting time at the destination node

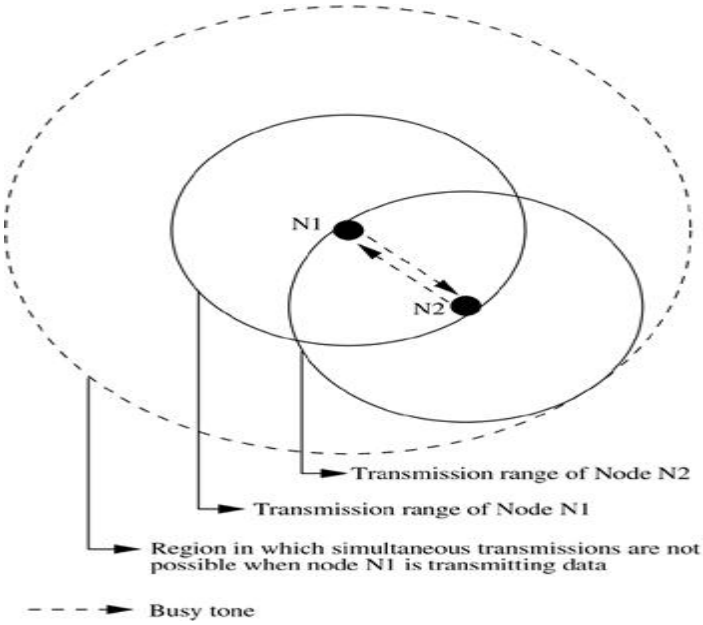
### *3.5.3 Busy Tone Multiple Access Protocols*

Busy Tone Multiple Access

The busy tone multiple access (BTMA) protocol is one of the earliest protocols proposed forovercoming the hidden terminal problem faced in wireless environments. The transmissionchannel is split into two: a data channel and a control channel. The data channel is used for datapacket transmissions, while the control channel is used to transmit the busy tone signal .When a node is ready for transmission, it senses the channel to check whether the busy tone is active. If not, it turns on the busy tone signal and starts data transmission; otherwise, itreschedules the packet for transmission after some random rescheduling delay. Any other nodewhich senses the carrier on the incoming data channel also transmits the busy tone signal on thecontrol channel. Thus, when a node is transmitting, no other node in the two-hop neighborhood of the transmitting node is permitted to simultaneously transmit. Though the probability of collisions is very low in BTMA, the bandwidth utilization is very poor.

Figure 3.8 shows theworst-case scenario where the node density is very high; the dotted circle shows the region inwhich nodes are blocked from simultaneously transmitting when node N1 is transmittingpackets.

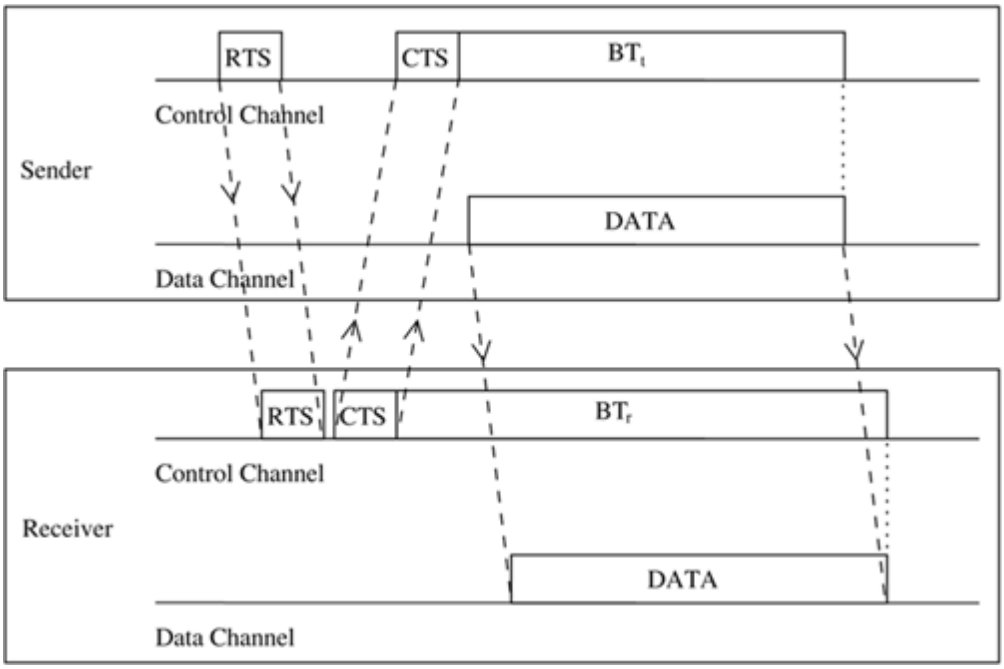
**Figure 6.8. Transmission in BTMA.**




## Dual Busy Tone Multiple Access Protocol

The dual busy tone multiple access protocol (DBTMA) is an extension of the BTMAscheme.Here again, the transmission channel is divided into two: the data channel and the controlchannel.As in BTMA, the data channel is used for data packet transmissions. The control channel is usedfor control packet transmissions (RTS and CTS packets) and also for transmitting the busytones. DBTMA uses two busy tones on the control channel, *BTt*and *BTr*. The *BTt*tone is used bythe node to indicate that it is transmitting on the data channel. The *BTr*tone is turned on by anode when it is receiving data on the data channel. The two busy tone signals are two sine wavesat different well-separated frequencies. When a node is ready to transmit a data packet, it first senses the channel to determine whetherthe *BTr*signal is active. An active *BTr*signal indicates that a node in the neighborhood of theready node is currently receiving packets. If the ready node finds that there is no *BTr*signal, ittransmits the RTS packet on the control channel. On receiving theRTS packets, the node towhich the RTS was destined checks whether the *BTt*tone is active in its neighborhood. An Active*BTt*implies that some other node in its neighborhood is transmitting packets and so itcannot receive packets for the moment. If the node finds no*BTt*signal, it responds by sendinga CTS packet and then turns on the *BTr*signal (which informs other nodes in its neighborhoodthat it is receiving). The sender node, on receiving this CTS packet, turns on the *BTt*signal (toinform nodes in its neighborhood that it is transmitting) and starts transmitting data packets.After completing transmission, the sender node turns off the *BTt*signal. The receiver node, afterreceiving all data packets, turns off the*BTr*signal. The above process is depicted in Figure 3.9.

**Figure 3.9. Packet transmission in DBTMA.**




When compared to other RTS/CTS-based medium access control schemes (such as MACA and MACAW ), DBTMA exhibits better network utilization. This is because the otherschemes block both the forward and reverse transmissions on the data channel when they reservethe channel through their RTS or CTS packets. But in DBTMA, when a node is transmitting orreceiving, only the reverse (receive) or forward (transmit) channels, respectively, are blocked. Hence the bandwidth utilization of DBTMA is nearly twice that of other RTS/CTS-basedschemes.

Receiver-Initiated Busy Tone Multiple Access Protocol

In the receiver-initiated busy tone multiple access protocol (RI-BTMA) , similar toBTMA, theavailable bandwidth is divided into two channels: a data channel for transmitting data packetsand a control channel. The control channel is used by a node to transmit the busy tone signal. Anode can transmit on the data channel only if it finds the busy tone to be absent on the controlchannel.

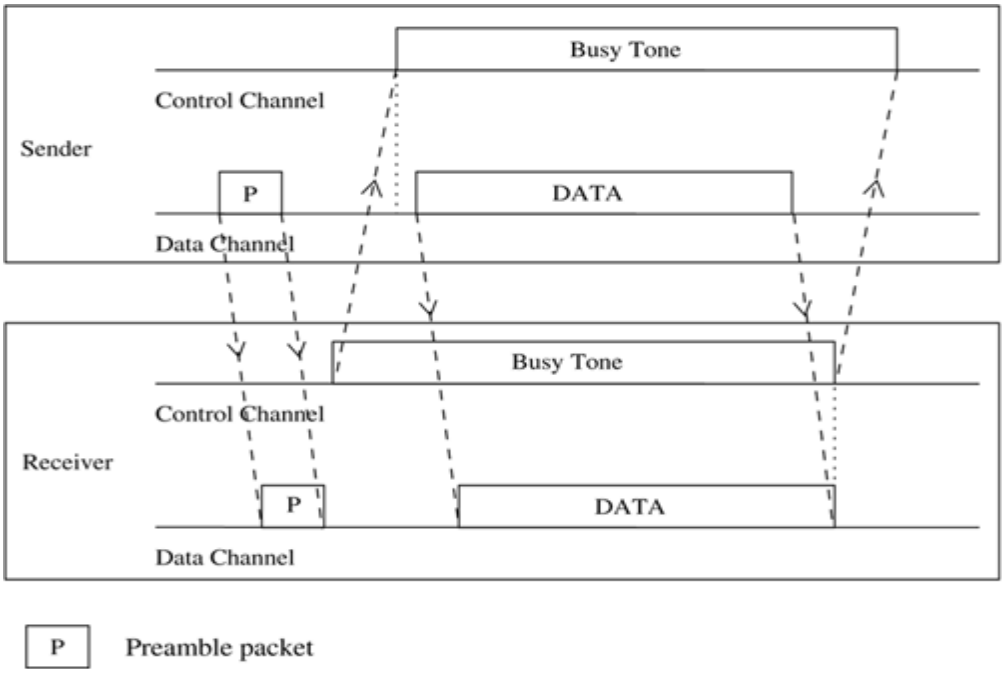
The data packet is divided into two portions: a preamble and the actual data packet. Thepreamble carries the identification of the intended destination node. Both the data channel andthe control channel are slotted, with each slot equal to the length of the preamble.

Data transmission consists of two steps. First, the preamble needs to be transmitted by the sender.Once the receiver node acknowledges the reception of this preamble by transmitting the busytone signal on the control channel, the actual data packet is transmitted. A sender node that needsto transmit a data packet first waits for a free slot, that is, a slot in which the busy tone signal isabsent on the control channel. Once it finds such a slot, it transmits the preamble packet on thedata channel. If the destination node receives this

preamble packet correctly without any error, ittransmits the busy tone on the control channel. It continues transmitting the busy tone signal aslong as it is receiving data from the sender. If preamble transmission fails, the receiver does notacknowledge with the busy tone, and the sender node waits for the next free slot and tries again.
The operation of the RI-BTMA protocol is shown in Figure 3.10. The busy tone serves twopurposes. First, it acknowledges the sender about the successful reception of the preamble.Second, it informs the nearby hidden nodes about the impending transmission so that they do nottransmit at the same time

**Figure 3.10. Packet transmission in RI-BTMA.**

There are two types of RI-BTMA protocols: the basic protocol and the controlled protocol. Thebasic packet transmission mechanism is the same in both protocols. In the basic protocol, nodesdo not have backlog buffers to store data packets. Hence packets that suffer collisions cannot beretransmitted. Also, when the network load increases, packets cannot be queued at the nodes.This protocol would work only when the network load is not high; when network load startsincreasing, the protocol becomes unstable.



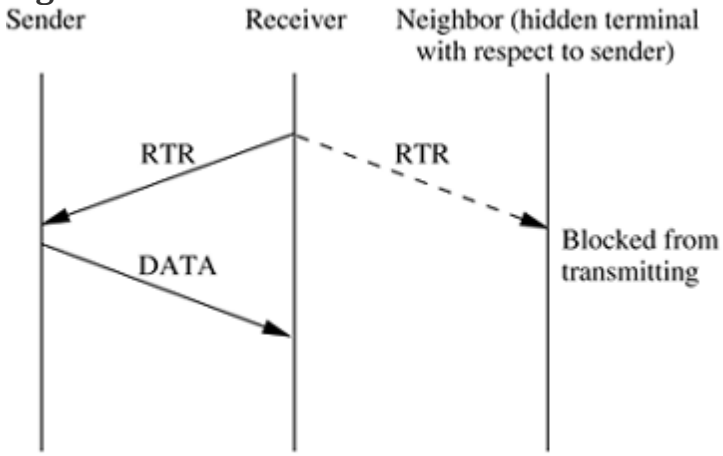

.
The controlled protocol overcomes this problem. This protocol is the same as the basic protocol,the only difference being the availability of backlog buffers at nodes. Therefore, packets thatsuffer collisions, and those that are generated during busy slots, can be queued at nodes. A nodeis said to be in the backlogged mode if its backlog buffer is non-empty. When a node in the

Backlogged mode receives a packet from its higher layers, the packet is put into the buffer andtransmitted later. Suppose the packet arrives at a node when it is not in the backlogged mode,then if the current slot is free, the preamble for the packet is transmitted with probability *p* in thecurrent slot itself (not transmitted in the same slot with probability (1 - *p*)). If the packet wasreceived during a busy slot, the packet is just put into the backlog buffer, where it waits until thenext free slot. A backlogged node transmits a backlogged packet in the next idle slot with aprobability *q*. All other packets in the backlog buffer just keep waiting until this transmissionsucceeds.This protocol can work for multi-hop radio networks as well as for single-hop fully connectednetworks.

### *3.5.4MACA-By Invitation*

MACA-by invitation (MACA-BI) is a receiver-initiated MAC protocol. It reduces thenumber of control packets used in the MACA [1] protocol. MACA, which is a sender-initiatedprotocol, uses the three-way handshake mechanism (which was shown in Figure 3.3), where firstthe RTS and CTS control packets are exchanged, followed by the actual DATA packettransmission. MACA-BI eliminates the need for the RTS packet.In MACA-BI the receiver node initiates data transmission by transmitting a ready to receive(RTR) control packet to the sender (Figure 6.11). If it is ready to transmit, the sender noderesponds by sending a DATA packet. Thus data transmission in MACA-BI occurs through atwo-way handshake mechanism.

**Figure 3.11. Packet transmission in MACA-BI.**




The receiver node may not have an exact knowledge about the traffic arrival rates at itsneighboring sender nodes. It needs to estimate the average arrival rate of packets. For providingnecessary information to the receiver node for this estimation, the DATA packets are modified tocarry control information regarding the backlogged flows at the transmitter node, number ofpackets queued, and packet lengths. Once this information is available at the receiver node, theaverage rate of the flows can be easily estimated.

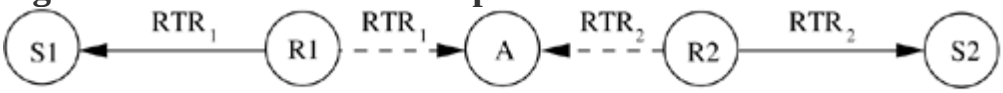
Suppose the estimation is incorrect or is notpossible (when the first data packet of the session is to be transmitted), theMACA-BI protocolcan be extended by allowing the sender node to declare its backlog through an RTS control packet, if an RTR packet is not received within a given timeout period.

In MACA, the CTS packet was used to inform the hidden terminals (nodes) about the impendingDATA packet transmission, so that they do not transmit at the same time and disrupt the session.This role is played in MACA-BI by the RTR packets.

An RTR packet carries information about the time interval during which the DATA packet would be transmitted. When a nodehears RTR packets transmitted by its neighbors, it can obtain information about the duration ofDATA packet transmissions by nodes that may be either its direct one-hop neighbors or its twohopneighbors, that is, hidden terminals. Since it has information about transmissions by thehidden terminals, it refrains from transmitting during those periods (Figure 3.11). Hence thehidden terminal problem is overcome inMACA-BI. Collision among DATA packets is impossible.

However, the hidden terminal problem still affects the control packet transmissions. This leads toprotocol failure, as in certain cases the RTR packets can collide with DATA packets. One suchscenario is depicted in Figure 3.12. Here, RTR packets transmitted by receiver nodes R1 and R2collide at node A. So node A is not aware of the transmissions from nodes S1 and S2. When node A transmits RTR packets, they collide with DATA packets at receiver nodes R1 and R2.

**Figure 3.12. Hidden terminal problem in MACA-BI.**




S1, S2 – Sender nodes
R1, R2 – Receiver nodes
A – Neighbor node

The efficiency of the MACA-BI scheme is mainly dependent on the ability of the receiver nodeto predict accurately the arrival rates of traffic at the sender nodes.

### *3.5.5 Media Access with Reduced Handshake*

The media access with reduced handshake protocol (MARCH) [8] is a receiver-initiatedprotocol. MARCH, unlike MACA-BI [7], does not require any traffic prediction mechanism. Theprotocol exploits the broadcast nature of traffic from omnidirectional antennas to reduce thenumber of handshakes involved in data transmission. In MACA , the RTS-CTScontrolpackets exchange takes place before the transmission of every data packet. But in MARCH,the RTS packet is used only for the first packet of the stream. From the second packet onward,

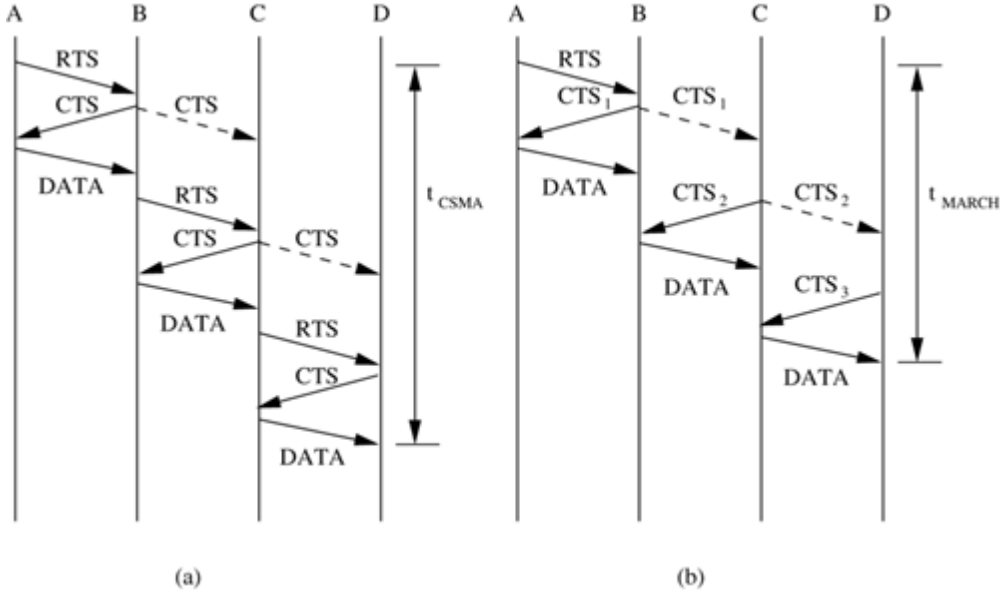
only the CTS packet is used.A node obtains information about data packet arrivals at its neighboring nodes by overhearingthe CTS packets transmitted by them. It then sends a CTS packet to the concerned neighbor nodefor relaying data from that node. This mechanism is illustrated inFigure 3.13.

- Figure 3.13(a) depicts the packet exchange mechanism of MACA

Here two controlpackets RTS and CTS need to be exchanged before each data packet is transmitted. It can beseen from this figure that node C, for example, can hear both the CTS and the RTSpacketstransmitted by node B. MARCH uses this property of the broadcast channel to reduce the twowayhandshake into a single CTS-only handshake.

- Figure 3.13 (b) shows the handshakemechanism of MARCH. Here, when node B transmits the *CTS*1 packet, this packet is also heardby node C. A CTS packet carries information regarding the duration of the next data packet.Node C therefore determines the time at which the next data packet would be available at nodeB. It sends the *CTS*2 packet at that point of time. On receiving the *CTS*2packet, node B sends thedata packet directly to node C. It can be observed from the figure that the time taken for a packettransmitted by node A to reach node D in MARCH, that is,*tMARCH*, is less compared to the timetaken in MACA, *tMACA*.

**Figure 3.13. Handshake mechanism in (a) MACA and (b) MARCH.**

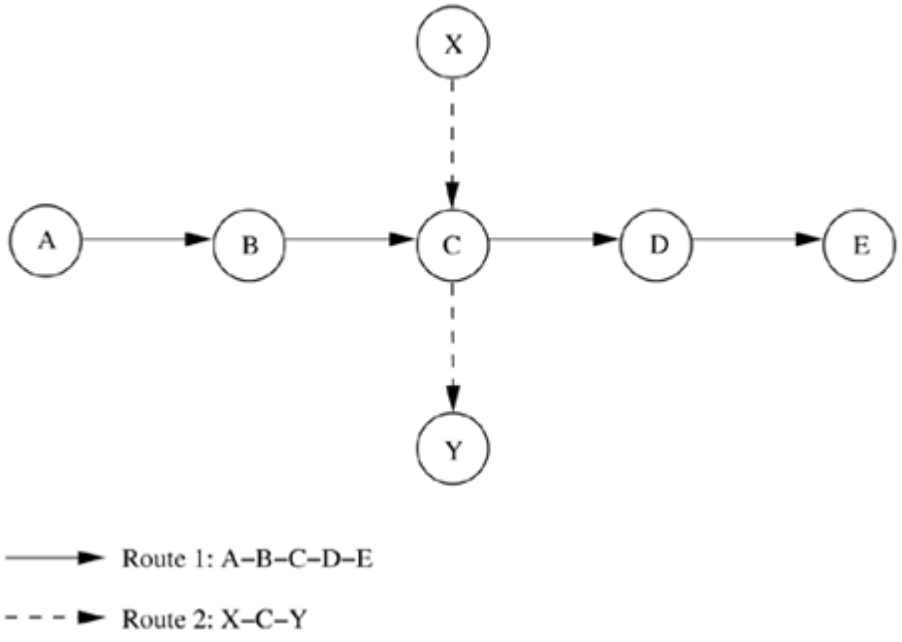



The CTS packet carries the MAC addresses of the sender and the receiver node, and the route identification number (RTid) for that flow. The RTid is used by nodes in order to avoid misinterpretation of CTS packets and initiation of false CTS-only handshakes.

ConsiderFigure 3.14. Here there are two routes – Route 1: A-B-C-D-E and Route 2: X-C-Y. When node C hears a CTS packet transmitted by node B, by means of the RTid field on the packet, it understands that the CTS was transmitted by its upstream node (upstream node refers to the next hop neighbor node on the path from the current node to the source node of the data session) on Route 1. It invokes a timer T which is set to expire after a certain period of time, long enough for node B to receive a packet from node A. A CTS packet is transmitted by node C once the timer expires. This CTS is overheard by node Y also, but since the RTid carried on the CTS is different from the RTid corresponding to Route 2, node Y does not respond. In MARCH, the MAC layer has access to tables that maintain routing information (such asRTid), but the protocol as such does not get involved in routing.

**Figure 3.14.Example topology.**

The throughput of MARCH is significantly high when compared to MACA, while the controloverhead is much less. When the network is heavily loaded, the average end-to-end delay inpacket delivery for MARCH is very low compared to that of MACA. All the above advantagesare mainly due to the fact that MARCH has a lower number of control packet handshakescompared to MACA. The lower number of control packets transmitted reduces the control overhead while improving the throughput, since less bandwidth is being consumed for controltraffic.




## 3.6 CONTENTION-BASED PROTOCOLS WITH RESERVATION MECHANISMS

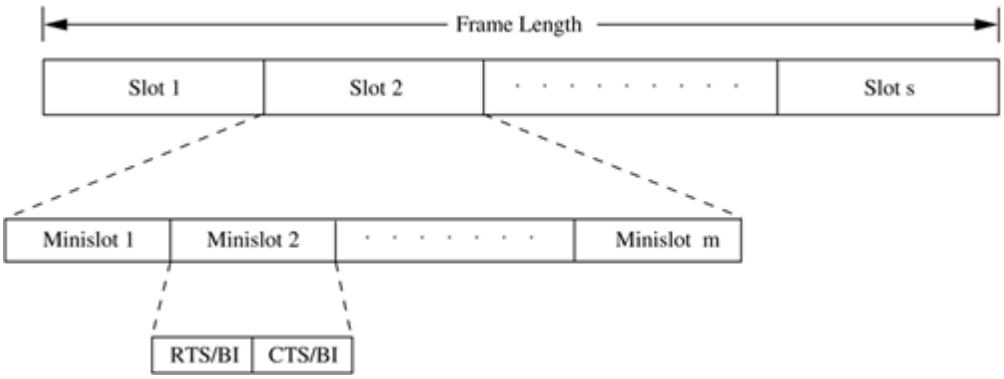
Protocols described in this section have certain mechanisms that aid the nodes in effectingbandwidth reservations. Though these protocols are contention-based, contention occurs onlyduring the resource (bandwidth) reservation phase. Once the bandwidth is reserved, the node gets exclusive access to the reserved bandwidth. Hence, QoS support can be provided for real-timetraffic.

### *3.6.1 Distributed Packet Reservation Multiple Access Protocol*

The distributed packet reservation multiple access protocol (D-PRMA) extends the earliercentralized packet reservation multiple access (PRMA) scheme into a distributed schemethat can be used in ad hoc wireless networks. PRMA was proposed for voice support in awireless LAN with a base station, where the base station serves as the fixed entity fortheMAC operation. D-PRMA extends this protocol for providing voice support in ad hocwireless networks.

D-PRMA is a TDMA-based scheme. The channel is divided into fixed- and equal-sized framesalong the time axis (Figure 3.15). Each frame is composed of *s* slots, and each slot consistsof*m* minislots. Each minislot can be further divided into two controlfields, RTS/BI and CTS/BI(BI stands for busy indication), as shown in the figure. These controlfields are used for slot reservation and for overcoming the hidden terminal problem. All nodes having packets ready fortransmission contend for the first minislot of each slot. The remaining (*m* - 1) minislots aregranted to the node that wins the contention. Also, the same slot in each subsequent frame can bereserved for this winning terminal until it completes its packet transmission session. If no nodewins the first minislot, then the remaining minislots are continuously used for contention, until acontending node wins any minislot. Within a reserved slot, communication between the sourceand receiver nodes takes place by means of either time division duplexing (TDD) or frequencydivision duplexing (FDD). Any node that wants to transmit packets has to first reserve slots, ifthey have not been reserved already. A certain period at the beginning of each minislot isreserved for carrier-sensing. If a sender node detects the channel to be idle at the beginning of aslot (minislot 1), it transmits an RTS packet (slot reservation request) to the intended destinationthrough the RTS/BI part of the current minislot. On successfully receiving this RTS packet, the receiver node responds by sending a CTS packet through theCTS/BI of the same minislot. If thesender node receives this CTS successfully, then it gets the reservation for the current slot andcan use the remaining minislots, that is, minislots 2 to*m*. Otherwise, it continues the contentionprocess through the subsequent minislots of the same slot.

**Figure 3.15. Frame structure in D-PRMA.**




In order to prioritize nodes transmitting voice traffic (voice nodes) over nodes transmittingnormal data traffic (data nodes), two rules are followed in D-PRMA. According to the first rule,the voice nodes are allowed to start contending from minislot 1 with probability $p = 1$; datanodes can start contending only with probability $p < 1$. For the remaining $(m - 1)$ minislots, boththe voice nodes and the data nodes are allowed to contend with probability $p < 1$. This is becausethe reservation process for a voice node is triggered only after the arrival of voice traffic at thenode; this avoids unnecessary reservation of slots. According to the second rule, only if the nodewinning the minislot contention is a voice node, is it permitted to reserve the same slot in each subsequent frame until the end of the session. If a data node wins thecontention, then it isallowed to use only one slot, that is, the current slot, and it has to make fresh reservations foreach subsequent slot.

Nodes that are located within the radio coverage of the receiver should not be permitted totransmit simultaneously when the receiver is receiving packets. If permitted, packets transmittedby them may collide with the packets of the on-going traffic being received at the receiver.Though a node which is located outside the range of a receiver is able to hear packets transmittedby the sender, it should still be allowed to transmit simultaneously. The above requirements, in essence, mean that the protocol must be free of the hidden terminal and exposed terminalproblems.

In D-PRMA, when a node wins the contention in minislot 1, other terminals must beprevented from using any of the remaining $(m - 1)$ minislots in the same slot for contention (*requirement 1*). Also, when a slot is reserved in subsequent frames, other nodes should beprevented from contending for those reserved slots (*requirement 2*). The RTS-CTS exchangemechanism taking place in the reservation process helps in trying to satisfy *requirement 1.* A node that wins the contention in minislot 1 starts transmitting immediately from minislot 2. Anyother node that wants to transmit will find the channel to be busy from minislot 2. Sincean RTS can be sent only when the channel is idle, other neighboring nodes would not contend forthe channel until the on-going transmission gets completed. A node sends an RTS inthe RTS/BI part of a minislot. Only a node that receives an RTS destined to it is allowed to use

the CTS/BI part of the slot for transmitting the CTS. So the CTS packet does not suffer anycollision due to simultaneous RTS packet transmissions. This improves the probability for asuccessful reservation. In order to avoid the hidden terminal problem, all nodes hearingthe CTS sent by the receiver are not allowed to transmit during the remaining period of that sameslot. In order to avoid the exposed terminal problem, a node hearing the RTS but notthe CTS (sender node's neighbor) is still allowed to transmit. But, if the communication is duplexin nature, where a node may transmit and receive simultaneously, even such exposed nodes (thathear RTS alone) should not be allowed to transmit. Therefore, D-PRMA makes such a nodedefer its transmissions for the remaining time period of the same slot. If an RTS or CTS packetcollides, and a successful reservation cannot be made in the first minislot, then the subsequent (*m* - 1) minislots of the same slot are used for contention.

For satisfying *requirement 2,* the following is done. The receiver of the reserved slot transmits abusy indication (BI) signal through the RTS/BI part of minislot 1 of the same slot in each of thesubsequent frames, without performing a carrier-sense. The sender also performs a similarfunction, transmitting the BI through the CTS/BI part of minslot 1 of the same slot in each subsequent frame. When any node hears a BI signal, it does not further contend for that slot inthe current frame. Because of this, the reserved slot in each subsequent frame is made free ofcontention. Also, making the receiver transmit the BI signal helps in eliminating the hiddenterminal problem, since not all neighbors of the receiver can hear from the sender. Finally, after anode that had made the reservation completes its data transmission and does not anymore require a reserved slot, it just stops transmitting the BIsignal. D-PRMA is more suited for voice traffic than for data traffic applications.

### *3.6.2 Collision Avoidance Time Allocation Protocol*

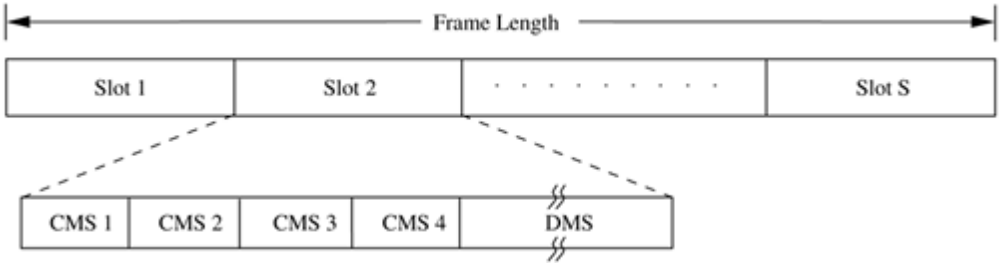
The collision avoidance time allocation protocol (CATA) is based on dynamic topologydependenttransmission scheduling. Nodes contend for and reserve time slots by means of adistributed reservation and handshake mechanism. CATA supports broadcast, unicast, andmulticast transmissions simultaneously. The operation of CATA is based on two basicprinciples:

- The receiver(s) of a flow must inform the potential source nodes about the reserved slot onwhich it is currently receiving packets. Similarly, the source node must inform the potentialdestination node(s) about interferences in the slot.
- Usage of negative acknowledgments for reservation requests, and control packet transmissionsat the beginning of each slot, for distributing slot reservation information to senders of broadcastor multicast sessions.

Time is divided into equal-sized frames, and each frame consists of *S* slots (Figure 3.16). Eachslot is further divided into five minislots. The first four minislots are used for transmitting controlpackets and are called control minislots (CMS1, CMS2, CMS3, and CMS4). The fifth and lastMinislot, called

data minislot (DMS), is meant for data transmission. The data minislot is muchlonger than the control minislots as the control packets are much smaller in size compared to datapackets.

**Figure 3.16. Frame format in CATA.**




Each node that receives data during the DMS of the current slot transmits a slot reservation (SR)packet during the CMS1 of the slot. This serves to inform other neighboring potential sendernodes about the currently active reservation. The SR packet is either received without error at theneighbor nodes or causes noise at those nodes, in both cases preventing such neighbor nodesfrom attempting to reserve the current slot. Every node that transmits data during the DMS of thecurrent slot transmits a request-to-send (RTS) packet during CMS2 of the slot. This RTS packet,when received by other neighbor nodes or when it collides with other RTS packets at theneighbor nodes, causes the neighbor nodes to understand that the source node is scheduled totransmit during the DMS of the current slot. Hence they defer their transmissions during the current slot.The control minislots CMS3 and CMS4 are used as follows. The sender of an intendedreservation, if it senses the channel to be idle during CMS1, transmits an RTS packet duringCMS2. The receiver node of a uni-cast session transmits a clear-to-send (CTS) packet duringCMS3. On receiving this packet, the source node understands that the reservation was successfuland transmits data during the DMS of that slot, and during the same slot in subsequent frames,until the unicast flow gets terminated. Once the reservation has been made successfully in a slot,from the next slot onward, both the sender and receiver do not transmit anything during CMS3,and during CMS4 the sender node alone transmits a not-to-send (NTS) packet. The purpose ofthe NTS packet is explained below. If a node receives anRTS packet for broadcast or multicastduring CMS2, or if it finds the channel to be free during CMS2, it remains idle and does nottransmit anything during CMS3 and CMS4. Otherwise, it sends a not-to-send (NTS) packetduring CMS4. The NTS packet serves as a negative acknowledgment; a potential multicast orbroadcast source node that receives theNTS packet during CMS4, or that detects noise during CMS4, understands that its reservation request had failed, and it does not transmit duringthe DMS of the current slot. If it finds the channel to be free during CMS4, which implies that itsreservation request was successful, it starts transmitting the multicast or broadcast packets duringthe DMS of the slot.

The length of the frame is very important in CATA. For any node (say, node A) to broadcastsuccessfully, there must be no other node (say, node B) in its two-hop neighborhood thattransmits simultaneously. If such a node B exists, then if node B is within node A's one-hopneighborhood, node A and node B cannot hear the packets transmitted by each other. If node B iswithin the two-hop neighborhood of node A, then the packets transmitted by nodes A and B would collide at their common neighbor nodes. Therefore, for any node to transmit successfullyduring one slot in every frame, the number of slots in each frame must be larger than the numberof two-hop neighbor nodes of the transmitting node. The worst-case value of the frame length, that is, the number of slots in the frame, would be $Min(d2 + 1, N)$, where$d$ is the maximum degree(degree of a node refers to the count of one-hop neighbors of the node) of a node in the network,and $N$ is the total number of nodes in the network.CATA works well with simple single-channel half-duplex radios. It is simple and providessupport for collision-free broadcast and multicast traffic.

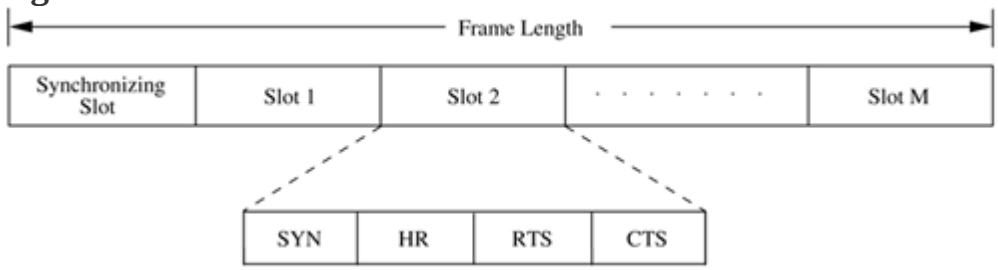
### *3.6.3 Hop Reservation Multiple Access Protocol*

The hop reservation multiple access protocol (HRMA) is a multichannel MAC protocolwhich is based on simple half-duplex, very slow frequency-hopping spread spectrum (FHSS)radios. It uses a reservation and handshake mechanism to enable a pair of communicating nodesto reserve a frequency hop, thereby guaranteeing collision-free data transmission even in thepresence of hidden terminals. HRMA can be viewed as a time slot reservation protocol where each time slot is assigned a separate frequency channel.Out of the available $L$ frequency channels, HRMA uses one frequency channel denoted by $f0$ , as a dedicated synchronizing channel. The nodes exchange synchronization information on$f0$ . Theremaining $L$ - 1 frequencies are divided into frequency pairs (denoted by ($fi$,), $i$= 1, 2, 3, .., $M$), thereby restricting the length of the hopping sequence to $M$. $fi$ is used fortransmitting and receiving hop-reservation (HR) packets, request-to-send (RTS) packets, clearto-send (CTS) packets, and data packets. is used for sending and receiving acknowledgment(ACK) packets for the data packets received or transmitted on frequency $fi$. In HRMA, time is slotted, and each slot is assigned a separate frequency hop, which is one among the $M$ frequencyhops in the hopping sequence. Each time slot is divided into four periods, namely, synchronizingperiod, HR period, RTS period, and CTS period, each period meant for transmitting or receivingthe synchronizing packet, FR packet, RTS packet, andCTS packet, respectively. All idle nodes, that is, nodes that do not transmit or receive packets currently, hop together. During thesynchronizing period of each slot, all idle nodes hop to the synchronizing frequency $f0$ andexchange synchronization information. During the HR,RTS, and CTS periods, they just stay idle,dwelling on the common frequency hop assigned to each slot. In addition to the synchronization period used for synchronization purposes, an exclusive synchronization slot is also defined at thebeginning of each HRMA frame (Figure 6.17). This slot is of

the same size as that of the othernormal slots. All idle nodes dwell on the synchronizing frequency *f*0 during the synchronizingslot and exchange synchronization information that may be used to identify the beginning of a frequency hop in the common hopping sequence, and also the frequency to be used in theimmediately following hop. Thus the HRMA frame, as depicted in

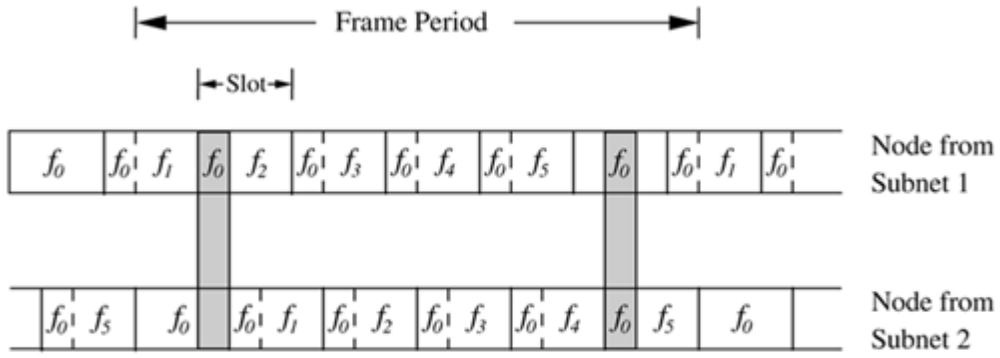
Figure 3.17, is composed ofthe single synchronizing slot, followed by *M* consecutive normal slots.

**Figure 6.17. Frame format in HRMA.**




When a new node enters the network, it remains on the synchronizing frequency *f*0 for a longenough period of time so as to gather synchronization information such as the hopping patternand the timing of the system. If it receives no synchronization information, it assumes that it isthe only node in the network, broadcasts its own synchronization information, and forms a onenodesystem. Since synchronization information is exchanged during every synchronization slot,new nodes entering the system can easily join the network. If $\mu$ is the length of each slotand $\mu s$the length of the synchronization period on each slot, then the dwell time of *fo*at thebeginning of each frame would be $\mu + \mu s$. Consider the case where nodes from two differentdisconnected network partitions come nearby. Figure 3.18depicts the worst-case frequencyoverlap scenario. In the figure, the maximum number of frequency hops $M = 5$. It is evident fromthe figure that within any time period equal to the duration of a HRMA frame, any two nodesfrom the two disconnected partitions always have at least two overlapping time periods oflength $\mu s$on the synchronizing frequency *f*0. Therefore, nodes belonging to disconnected networkcomponents can easily merge into a single network.

**Figure 3.18. Merging of subnets.**




$f_0$ synchronizing frequency

M = 5

When a node receives data to be transmitted, it first listens to the HR period of the immediatelyfollowing slot. If it hears an HR packet, it backs off for a randomly chosen period (which is amultiple of slot time). If it finds the channel to be free during the SR period, it transmitsan RTS packet to the destination during the RTS period of the slot and waits for theCTS packet.On receiving the RTS, the destination node transmits the CTS packet during theCTS period ofthe same slot, stays on the same frequency currently being used, and waits for the data packet. Ifthe source node receives the CTS packet correctly, it implies that the source and receiver nodeshave successfully reserved the current hop. In case the source node does not receiveany CTS packet, it backs off for a random number of time slots and repeats the entire processagain. The source and receiver nodes dwell on the same reserved frequency throughout the data transmission process, which starts immediately after the CTS period. As mentioned earlier, aseparate frequency ($f_i^*, i = 1, 2, ..., M$) is used for transmitting acknowledgments. Aftertransmitting each data packet, the source node hops onto this acknowledgment frequency. Thereceiver sends an acknowledgment (ACK) packet back to the source on this acknowledgment frequency. Once the ACK packet transmission/reception is over, both the source and receiverhop back to the reserved frequency to continue with the data transmission.After the CTS period of a slot, the idle nodes that do not transmit or receive packets hop onto thesynchronization frequency $f0$ and exchange synchronization information. They dwell on $f0$for atime period of $\mu s$and then hop onto the next frequency hop in the common hopping sequence.

The data packets transmitted can be of any size. Data transmission can take place through asingle packet or through a train of packets. A maximum dwell period has been defined in order toprevent nodes from hogging onto a particular frequency channel. Therefore, the transmissiontime for the data packet, or the train of data packets, should not exceed this maximum dwell time. Suppose the sender needs to transmit data packets across multiple frames, then it informsthe receiver node through the header of the data packet it transmits. On reading this information,the receiver node transmits an HR packet

during the HR period of the same slot in the nextframe. The neighbor nodes of the receiver on hearing this HRpacket refrain from using the frequency hop reserved. On receiving the HR packet, the source node of the session sendsan RTS packet during the RTS period and jams other RTS packets (if any) destined to itsneighbors, so that the neighbor nodes do not interfere on the reserved frequency hop. Both thesender and the receiver remain silent during the CTS period, and data transmission resumes oncethis CTS period gets over.

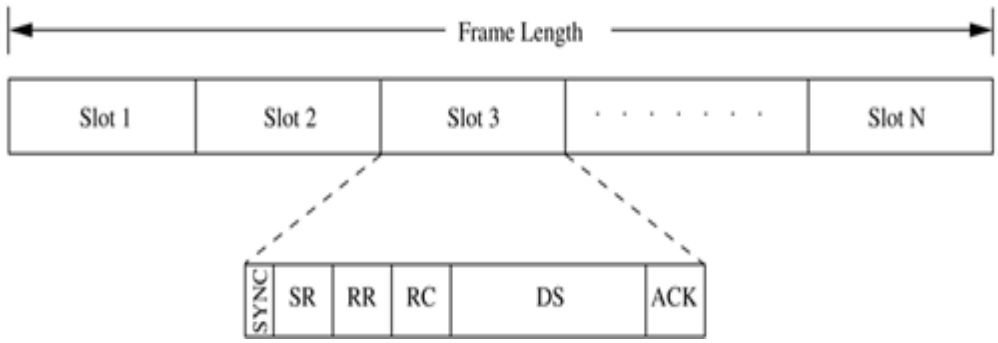
### *3.6.4 Soft Reservation Multiple Access with Priority Assignment*

Soft reservation multiple access protocol with priority assignment (SRMA/PA) wasdeveloped with the main objective of supporting integrated services of real-time and non-realtimeapplications in ad hoc wireless networks, at the same time maximizing the statisticalmultiplexing gain. Nodes use a *collision avoidance* handshake mechanism and a*softreservation* mechanism in order to contend for and effect reservation of time slots. The softreservation mechanism allows any urgent node, transmitting packets generated by a real-timeapplication, to take over the radio resource from another node of a non-real-time application onan on-demand basis.

SRMA/PA is a TDMA-based protocol in which nodes are allocated different time slots so that the transmissions are collision-free. The main featuresofSRMA/PA are a unique frame structure and soft reservation capability for distributed anddynamic slot scheduling, dynamic and distributed access priority assignment and update policies,and a time-constrained back-off algorithm.

Time is divided into frames, with each frame consisting of a fixed number (*N*) of time slots. Theframe structure is shown in Figure 3.19. Each slot is further divided into six different fields,SYNC, soft reservation (SR), reservation request (RR), reservation confirm (RC), data sending(DS), and acknowledgment (ACK). The SYNC field is used for synchronization purposes. The SR, RR, RC, and ACK fields are used for transmitting and receiving the correspondingcontrol packets. The DS field is used for data transmission. The SR packet serves as a busy tone.It informs the nodes in the neighborhood of the transmitting node about the reservation of theslot. The SR packet also carries the access priority value assigned to the node that has reserved the slot. When an idle node receives a data packet for transmission, the node waits for a free slotand transmits the RR packet in the RR field of that slot. A node determines whether or not a slotis free through the SR field of that slot. In case of a voice terminal node, the node tries to takecontrol of the slot already reserved by a data terminal if it finds its priority level to be higher thanthat of the data terminal. This process is called *soft reservation.* This makesthe SRMA/PA different from other protocols where even if a node has lower access prioritycompared to other ready nodes, it proceeds to complete the transmission of the entire data burstonce it has reserved the channel.

**Figure 3.19. Frame structure in SRMA/PA.**




Priority levels are initially assigned to nodes based on the service classes (real-time or non-realtime)in a static manner. Once the node acquires the channel, the corresponding slot staysreserved for the node until the node completes transmitting the entire data burst. The node isassigned a prespecified priority $p_v^{(R)}$ or $p_d^{(R)}$, respectively, for voice and data terminals. Rdenotes that the node is a reserved node, that is, a node that has successfully reserved the slot. It is required that $p_v^{(R)} > p_d^{(R)}$, such that delay-sensitive voice applications get preference overnormal data applications. Whenever the reservation attempt fails due to collision, the accesspriority of the node is updated based on the urgency of its packets.

A node that is currently transmitting is said to be in the active state. A node is said to be in theidle state if it does not have any packet to be transmitted. In the active state itself, nodes can be inone of the two states: access state and reserved state. Access state is one in which the node isbacklogged and is trying to reserve a slot for transmission. The node is said to be in the reserved state if it has already reserved the slot for transmission. Whenever the access priority level of avoice terminal in the access state becomes greater than that of a data terminal in the reservedstate, which could be known from the SR field, the corresponding slot is taken over by theprioritized voice terminal. In order to effect this mechanism, the values of priority levels must be such that $p_v^{(R)} > p_v(n) > p_d^{(R)} > p_d(n)$, where $p_v(n)$ and $p_d(n)$ are the access priority values of a voice terminal and data terminal, respectively, after its *nth* reservation attempt resultsin a collision. This soft reservation feature of SRMA/PA, where a voice terminal can take overthe slots reserved by a data terminal whenever, due to the urgent nature of its traffic, its access priority becomes higher than that of the data terminal, helps in maximizing the statisticalmultiplexing gain for voice-data integrated services.

The RR-RC-DS-ACK exchange mechanism of SRMA/PA is similar to the RTS-CTS-DATAACKexchange mechanism of MACAW . The RR and RC packets help in eliminating thehidden terminal problem. The major difference between SRMA/PA and CATA protocol isthat, while in CATA the slot reservation (SR) packet is transmitted by the receiver of the session,in SRMA/PA it is sent by the source node. Also, the soft reservation feature ofSRMA/PA isabsent in CATA.

The access priorities are assigned to nodes and updated in a distributed and dynamic manner.This allows dynamic sharing of the shared channel. On receiving a new packet for transmission,an idle node becomes active. Now, transition to the access state is made with the initial access priority assigned a value $p_v^{(0)}$ or $p_d^{(0)}$, depending on whether it is a voice or data terminal. If the random access attempt to effect the reservation by transmitting an RRpacket ends up in acollision, then the access priorities of the node concerned are increased as follows:

$$p_v(n+1) = p_v(n) + \Delta p_v, \qquad p_v(0) = p_v^{(0)}$$
$$p_d(n+1) = p_d(n) + \Delta p_d, \qquad p_d(0) = p_d^{(0)}$$

where $\Delta pv$and $\Delta pd$are the incremental access priorities for voice and data services, respectively.

They reflect the urgency of the traffic queued at the two types of terminals, and are given asbelow:

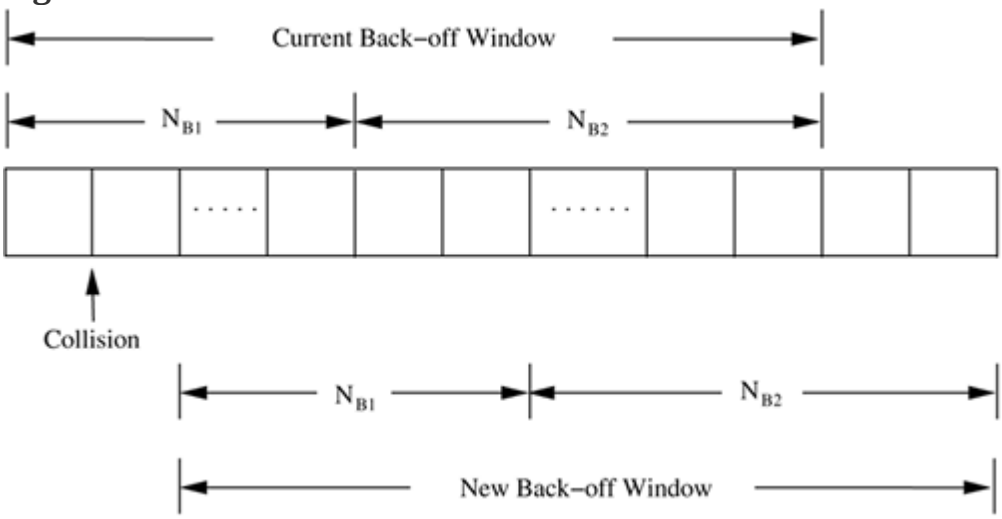
$$p_d^{(0)} < p_d(n) < p_d^{(R)} = p_d^{(max)} < p_v^{(0)} < p_v(n) < p_v^{(R)} = p_v^{(max)}$$

Though dynamic assignment and update of access priority values are followed in SRMA/PA,collisions among nodes with the same priority and carrying traffic of the same service typescannot be avoided completely. Collisions occur during the RR field of the slot. In order to avoidcollisions, a binary exponential back-off algorithm is used for non-real-time connections, and amodified binary exponential back-off algorithm is used for real-time connections. The modified algorithm implements a priority access policy in order to meet the delay requirements of realtimesessions. Here the back-off window is divided into two different regions, each regionhaving a length of *NB* 1 and *NB* 2 ,respectively, for real-time and non-real-time traffic. Each nodechecks the laxity of its head-of-line packet (laxity is the difference between the maximum access delay allowed and the residual lifetime of the packet). If the laxity exceeds the threshold *Tlimit* slots, one slot out of the *NB* 1 slots is selected randomly. Otherwise, one out ofthe *NB* 2 slots is chosen randomly, that is, if the node is unable to make a reservation within thegiven time (*Tlimit* slots), it is given a higher priority compared to other non-real-time nodes by choosing a slot from *NB* 1; otherwise, the node selects a slot from *NB* 2 . The RR packet istransmitted on this chosen slot. Here again, if more than one node selects the

same random slotand their RR packets collide again, a new back-off window starts immediately after the currentslot. This is shown in Figure 3.20.

The above back-off mechanism, which gives high preferenceto nodes transmitting delay-sensitive traffic, helps in guaranteeing the QoS requirements of realtimeservices in the network. The parameters *NB* 1 ,*NB* 2 , and *Tlimit* significantly affect theperformance of the protocol, and must be chosen carefully based on the traffic load expected onthe network.

**Figure 3.20. Back-off windows.**




### *6.6.5 Five-Phase Reservation Protocol*

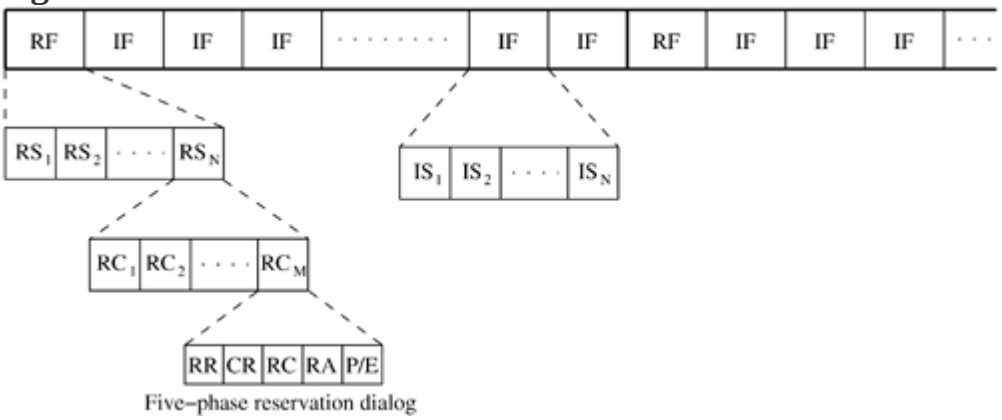
The five-phase reservation protocol (FPRP) [14] is a single-channel time division multiple access(TDMA)-based broadcast scheduling protocol. Nodes use a contention mechanism in order toacquire time slots. The protocol is fully distributed, that is, multiple reservations can be simultaneously made throughout the network. No ordering among nodes is followed; nodes need not wait for making time slot reservations. The slot reservations are made using a five-phasereservation process. The reservation process is localized; it involves only the nodes locatedwithin the two-hop radius of the node concerned. Because of this, the protocol is insensitive tothe network size, that is, it is scalable. FPRP also ensures that no collisions occur due to the hidden terminal problem.

Time is divided into frames. There are two types of frames: reservation frame (RF) andinformation frame (IF). Each RF is followed by a sequence of IFs. Each RF has *N* reservationslots (RS), and each IF has *N* information slots (IS). In order to reserve an IS, a node needs tocontend during the corresponding RS. Based on these contentions, a TDMA schedule isgenerated in the RF and is used in the subsequent IFs until the next RF. The structure of theframes is shown in Figure 3.21.

Each RS is composed of *M* reservation cycles (RC). Withineach RC, a five-phase dialog takes place, using which a node reserves slots. If a node wins the

contention in an RC, it is said to have reserved the IS corresponding to the current RS in thesubsequent IFs of the current frame. Otherwise, the node contends during the subsequent RCs ofthe current RS until itself or any other node (1-hop or 2-hop neighbor) succeeds. During thecorresponding IS, a node would be in one of the following three states: transmit (T), receive (R),or blocked (B). The five-phase dialog ensures that the protocol is free from the hidden nodeproblem, and also ensures that once a reservation is made by a node with a high probability, itgets sole access to the slot within its neighborhood.

**Figure 3.21. Frame structure in FPRP.**




The protocol assumes the availability of global time at all nodes. Each node therefore knowswhen a five-phase cycle would start. The five phases of the reservation process are as follows:

1. Reservation request phase: Nodes that need to transmit packets send reservation request (RR)packets to their destination nodes.
2. Collision report phase: If a collision is detected by any node during the reservation request phase,then that node broadcasts a collision report (CR) packet. The corresponding source nodes, uponreceiving the CR packet, take necessary action.
3. Reservation confirmation phase: A source node is said to have won the contention for a slot if itdoes not receive any CR messages in the previous phase. In order to confirm the reservationrequest made in the reservation request phase, it sends a reservation confirmation (RC) messageto the destination node in this phase.
4. Reservation acknowledgment phase: In this phase, the destination node acknowledges receptionof the RC by sending back a reservation acknowledgment (RA) message to the source. Thehidden nodes that receive this message defer their transmissions during the reserved slot.
5. Packing and elimination (P/E) phase: Two types of packets are transmitted during this phase:packing packet and elimination packet. The details regarding the use of these packets will bedescribed later in this section.

Each of the above five phases is described below.

*Reservation request phase:*

In this phase, each node that needs to transmit packets sends an RR packet to the intendeddestination node with a contention probability *p*, in order to reserve an IS. Such nodes thatsend RR packets are called requesting nodes (RN). Other nodes just keep listening during thisphase.

*Collision report phase:*

If any of the listening nodes detects collision of RR packets transmitted in the previous phase, itbroadcasts a collision report (CR) packet. By listening for CR packets in this phase, an RNcomesto know about collision of the RR packet it had sent. If no CR is heard by the RN in this phase,then it assumes that the RR packet did not collide in its neighborhood. It then becomes atransmitting node (TN). Once it becomes a transmitting node, the node proceeds to the next phase, the reservation confirmation phase. On the other hand, if it hears a CRpacket in this phase,it waits until the next reservation request phase, and then tries again. Thus, if two RNs are hiddenfrom each other, their RR packets collide, both receive CRpackets, and no reservation is made, thereby eliminating the hidden terminal problem.

*Reservation confirmation phase:*

An RN that does not receive any CR packet in the previous phase, that is, a TN, sendsan RCpacket to the destination node. Each neighbor node that receives this packet understandsthat the slot has been reserved, and defers its transmission during the corresponding informationslots in the subsequent information frames until the next reservation frame.

*Reservation acknowledgment phase:*

On receiving the RC packet, the intended receiver node responds by sending an RA packet backto the TN. This is used to inform the TN that the reservation has been established. In case the TN is isolated and is not connected to any other node in the network, then it would notreceive the RA packet, and thus becomes aware of the fact that it is isolated. Thus the RCpacketprevents such isolated nodes from transmitting further. The reservation acknowledgment phasealso serves another purpose. Other two-hop neighbor nodes that receive this RA packet getblocked from transmitting. Therefore, they do not disturb the transmission that is to take place inthe reserved slots.When more than two TNs are located nearby, it results in a deadlock condition. Such situationsmay occur when there is no common neighbor node present when the RNs transmit RR packets.

Collisions are not reported in the next phase, and so each node claims success and becomesa TN. Deadlocks are of two types: isolated and non-isolated. An isolated deadlock is a conditionwhere none of the deadlocked nodes is connected to any non-deadlocked node. In the nonisolateddeadlock situation, at least one deadlocked node is connected to a non-deadlockedneighbor node. The RA phase can resolve isolated deadlocks. None of the nodes transmits RA, and hence the TNs abort their transmissions.

***Packing/elimination (P/E) phase:***

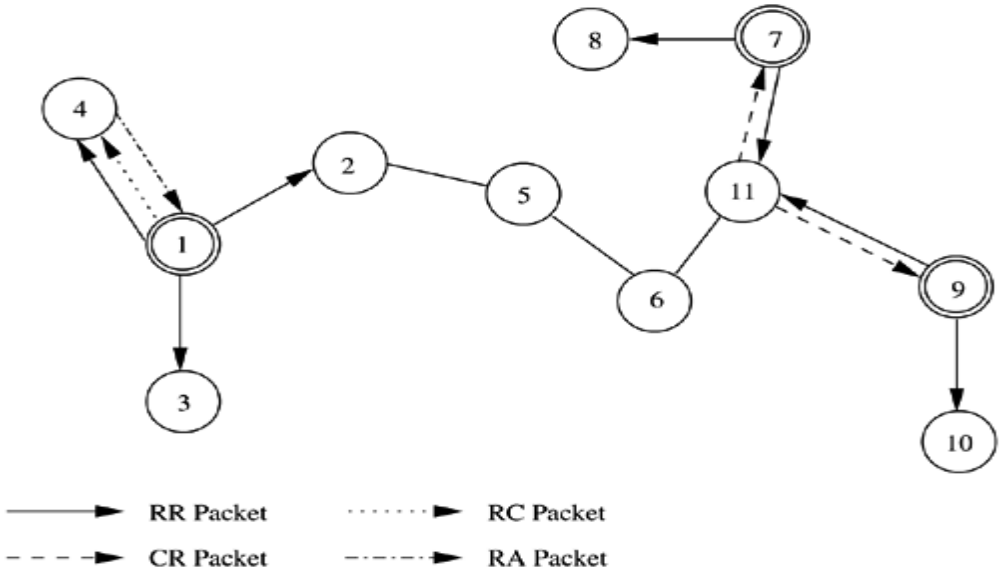
In this phase, a packing packet (PP) is sent by each node that is located within two hops froma TN, and that had made a reservation since the previous P/E phase. A node receivinga PPunderstands that there has been a recent success in slot reservation three hops away from it,and because of this some of its neighbors would have been blocked during this slot. The nodecan take advantage of this and adjust its contention probability *p*, so that convergence is faster.

In an attempt to resolve a non-isolated deadlock, each TN is required to transmit an eliminationpacket (EP) in this phase, with a probability 0.5. A deadlocked TN, on receiving an EP beforetransmitting its own EP, gets to know about the deadlock. It backs off by marking the slot asreserved and does not transmit further during the slot.

Consider Figure 3.22. Here nodes 1, 7, and 9 have packets ready to be transmitted to nodes 4, 8,and 10, respectively. During the reservation request phase, all three nodes transmit RRpackets.Since no other node in the two-hop neighborhood of node 1 transmits simultaneously, node 1does not receive any CR message in the collision report phase. So node 1 transmitsan RC message in the next phase, for which node 4 sends back an RAmessage, and thereservation is established. Node 7 and node 9 both transmit the RR packet in the reservation request phase. Here node 9 is within two hops from node 7. So if both nodes 7 and 9 transmitsimultaneously, their RR packets collide at common neighbor node 11. Node 11 sendsa CR packet which is heard by nodes 7 and 9. On receiving the CR packet, nodes 7 and 9 stopcontending for the current slot.

**Figure 3.22. FPRP - Example.**

The reservation process in FPRP is simple. No information needs to be distributed to nodes otherthan the one-hop neighbor nodes before the reservation becomes successful



### *3.6.6MACA with Piggy-Backed Reservation*

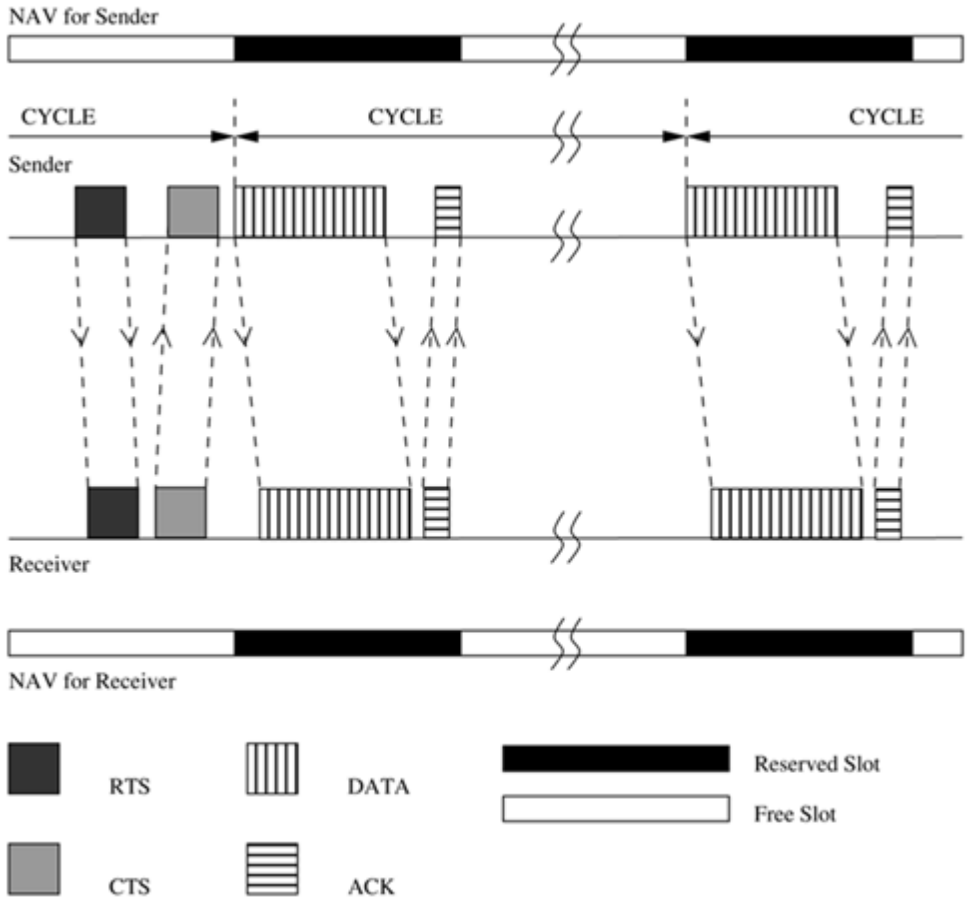
MACA with piggy-backed reservation (MACA/PR) is a protocol used to provide real-time traffic support in multi-hop wireless networks. The MAC protocol used is based on the MACAW protocol , with the provisioning of non-persistent CSMA (as in FAMA ). The main components of MACA/PR are: a MAC protocol, a reservation protocol, and a QoS routing protocol. MACA/PR differentiates real-time packets from the best-effort packets. While providing guaranteed bandwidth support for real-time packets, at the same time it provides reliable transmission of best-effort packets. Time is divided into slots. The slots are defined by the reservations made at nodes, and hence are asynchronous in nature with varying lengths. Each node in the network maintains a reservation table (RT) that records all the reserved transmit and receive slots/windows of all nodes within its transmission range.

In order to transmit a non-real-time packet, a MACAW -based MAC protocol is used. The ready node (a node which has packets ready for transmission) first waits for a free slot in the RT. Once it finds a free slot, it again waits for an additional random time of the order of a single-hop round-trip delay time, after which it senses the channel. If the channel is found to be still free, the node transmits an RTS packet, for which the receiver, if it is ready to receive packets, responds with a CTS packet. On receiving the CTS packet, the source node sends a DATA packet, and the receiver, on receiving the packet without any error, finally sends an ACK packet back to the source. The RTS and CTS control packets contain, in them, the time duration in which the DATA packet is to be transmitted. A nearby node that hears these packets avoids transmission during that time. If, after the random waiting time, the channel is found to be busy, the node waits for the channel to become idle again, and then repeats the same procedure.

For real-time traffic, the reservation protocol of MACA/PR functions as follows. The sender isassumed to transmit real-time packets at certain regular intervals, say, every CYCLE time period. The first data packet of the session is transmitted in the usual manner just as a best-effort packet would be transmitted. The source node first sends an RTS packet, for which the receiver node responds with a CTS packet. Now the source node sends the first DATA packet of the real-time session. Reservation information for the next DATA packet to be transmitted (which is scheduled to be transmitted after CYCLE time period) is piggy-backed on this current DATA packet. On receiving this DATA packet, the receiver node updates its reservation table with the piggy-backed reservation information. It then sends an ACK packet back to the source. Thereceiver node uses the ACK packet to confirm the reservation request that was piggy-backed on the previous DATA packet. It piggy-backs the reservation confirmation information on the ACK packet. Neighbor nodes that hear the DATA and ACK packets update their reservation tables with the reservation information carried by them, and refrain from transmitting when the

next packet is to be transmitted. Unlike MACAW , MACA/PR does not make use of RTS/CTS packets for transmission of the subsequent DATA packets. After receiving the ACK, the source node directly transmits the next DATA packet at its scheduled transmission time in the next CYCLE. This DATA packet in turn would carry reservation information for the next DATA packet. Real-time data transmission, hence, occurs as a series of DATA-ACK packet exchanges. The real-time packets (except for the first packet of the session that is used to initiate the reservation process) are transmitted only once. If an ACK packet is not received for a DATA packet, the source node just drops the packet. The ACK packet therefore serves the purpose of renewing the reservation, in addition to recovering from packet loss. If the source node fails to receive ACK packets for a certain number of consecutive DATA packets, it then assumes the reservation to have been lost. It restarts the real-time session again with an RTS-CTS control packet exchange, either on a different slot on the same link, or on a different link in case of a path break. In order to transmit an RTS to the receiver node, the source needs to find a slot that is free at both the nodes. For maintaining consistent information regarding free slots at all nodes, MACA/PRuses periodic exchange of reservation tables. This periodic table exchange automatically overcomes the hidden terminal problem. When a hidden terminal receives a reservation table from a node, it refrains from transmitting in the reserved slots of that node. Slot reservation information maintained in the reservation tables is refreshed every cycle. If the reservation is not refreshed for a certain number of consecutive cycles, it is then dropped.
Thetransmission of packets in MACA/PR is depicted in Figure 3.23. It can be seen from the figure that theRTS-CTS exchange is used only for transmitting the first packet of the session. Since each DATA packet carries reservation information for the next DATA packet that would be transmitted after CYCLE time period, RTS-CTS exchange is not required for the subsequentDATA packets. Neighbor nodes that receive DATA packets update their reservation tables accordingly and do not contend for the channel during the reserved slots. The network allocation vector (NAV) at each node reflects the current and future state of the channel as perceived by the node.

**Figure 3.23. Packet transmission in MACA/PR.**




Best-effort packet transmissions and real-time packet transmissions can be interleaved at nodes,with higher priority being given to real-time packets. For real-timepackets, MACA/PReffectively works as a TDM system, with a superframe time of CYCLE. Thebest-effort packets are transmitted in the empty slots (which have not been reserved) of the cycle.

When a new node joins the network, it initially remains in the listening mode during which itreceives reservation tables from each of its neighbors and learns about the reservations made inthe network. After this initial period, the node shifts to its normal mode of operation.

The QoS routing protocol used with MACA/PR is the destination sequenced distance vector(DSDV) routing protocol .Bandwidth constraint has beenintroduced in the routing process. Each node periodically broadcasts to its neighbors the(bandwidth, hop distance) pairs for each preferred path, that is, for each bandwidth value, to eachdestination. The number of preferred paths is equal to the maximum number of slots in a cycle.After this is done, if a node receives a real-time packet with a certain bandwidth requirement thatCannot be satisfied using the current available paths, the packet is dropped and no ACK packet issent. The sender node would eventually reroute the packet.

Thus, MACA/PR is an efficient bandwidth reservation protocol that can support real-time trafficsessions. One of the important advantages of MACA/PR is that

it does not require globalsynchronization among nodes. A drawback of MACA/PR is that a free slot can be reserved onlyif it can fit in the entire RTS-CTS-DATA-ACK exchange. Therefore, there is a possibility ofmany fragmented free slots not being used at all, reducing the bandwidth efficiency of theprotocol.

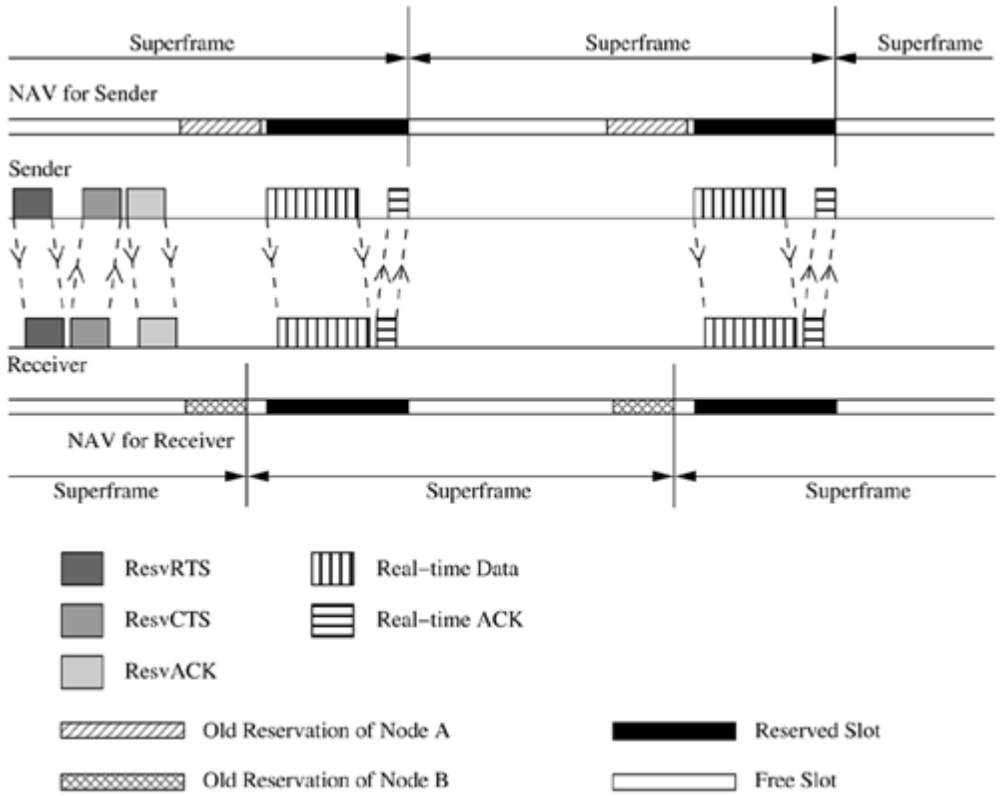
### *3.6.7 Real-Time Medium Access Control Protocol*

The real-time medium access control protocol (RTMAC) provides a bandwidth reservationmechanism for supporting real-time traffic in ad hoc wireless networks.

RTMACconsists of twocomponents, a MAC layer protocol and a QoS routing protocol. The MAClayer protocol is areal-time extension of the IEEE 802.11 DCF. The QoS routing protocol is responsible for end-toendreservation and release of bandwidth resources. The MAC layer protocol has two parts: a medium-access protocol for best-effort traffic and a reservation protocol for real-time traffic.A separate set of control packets, consisting of *ResvRTS*, *ResvRTSResvCTS*, and *ResvACK*, isused for effecting bandwidth reservation for real-time packets. RTS, CTS, and ACK controlpackets are used for transmitting best-effort packets. In order to give higher priority for real-time packets, the wait time for transmitting a *ResvRTS*packet is reduced to half of DCF inter-framespace (DIFS), which is the wait time used for best-effort packets.

Time is divided into superframes. As can be seen from Figure 6.24, the superframe for each nodemay not strictly align with the other nodes. Bandwidth reservations can be made by a node byreserving variable-length time slots on superframes, which are sufficient enough to carry thetraffic generated by the node. The core concept of RTMAC is the flexibility of slot placement in the superframe. Each superframe consists of a number of reservation-slots (resv-slots). The timeduration of each resv-slot is twice the maximum propagation delay. Data transmission normallyrequires a block of resv-slots. A node that needs to transmit real-time packets first reserves a setof resv-slots. The set of resv-slots reserved by a node for a connection on a superframe is called aconnection-slot. A node that has made reservations on the current superframe makes use of thesame connection-slot in the successive superframes for transmitting packets. Each nodemaintains a reservation table containing information such as the sender id, receiver id, andstarting and ending times of reservations that are currently active within its direct transmission range.

**Figure 3.24. Reservation mechanism in RTMAC.**




In RTMAC, no time synchronization is assumed. The protocol uses relative time for allreservation purposes. When a node receives this relative-time-based information, it converts therelative time to absolute time by adding its current time maintained in its clock. A three-wayhandshake protocol is used for effecting the reservation. For example, node A, which wants toreserve a slot with node B, sends a *ResvRTS*packet which contains the relative time informationof starting and ending of the connection-slot (a number of resv-slots) to be reserved. Node B, onreceiving this packet, first checks its reservation table to see whether it can receive on those resvslots.

If it can, it replies by sending a *ResvCTS*packet containing the relative time information of the same resv-slots to be reserved. Neighbor nodes of the receiver, on receivingthe *ResvCTS,* update their reservation tables accordingly. Source node A, on receivingthe *ResvCTS*packet, responds by sending a *ResvACK*packet. This packet also carries relativetime information regarding the reserved slots. The *ResvACK*packet is meant for the neighbornodes of the source node (node A) which are not aware of the reservation as they may notreceive the *ResvCTS*packet. Such nodes update their reservation tables on receivingthe *ResvACK*packet. Transmission of the *ResvACK*packet completes the reservation process.Once the reservation is made, real-time packets are transmitted in these reserved slots.Transmission of each real-time packet is followed by the transmission of a real-time ACK(RTACK) packet by the receiver.
The bandwidth reservation process is illustrated in Figure 3.24.

In the figure, NAV indicates thenetwork allocation vector maintained at each node. The NAV at a node reflects the current and future state of the channel as perceived by the node.The sender node first transmits a *ResvRTS*packet indicating the connection-slot (represented bythe offset time from the current time for the beginning and end of the connection-slot) to bereserved. The receiver node on receiving this packet checks its NAVand finds that the requestedconnection-slot is free. So it responds by sending a *ResvCTS*packet carrying the sameconnection-slot information. The sender node, on receiving this packet, completes the reservationprocess by sending a *ResvACK*packet. The corresponding connection-slot is marked as reservedat both the sender and the receiver. This is indicated inFigure 3.24 by the dark-shaded regions in the NAVs of the sender and receiver. Once the reservation is made, the real-time session getsstarted, and packets are transmitted in the reserved connection-slot by means of *Real-timeData –Real-timeACK* exchanges.

If the receiver node receives the *ResvRTS*packet on a slot which has already been reserved byone of its neighbor nodes, it does not respond with a *ResvCTS*packet. It just discards thereceived *ResvRTS*packet. This is because, if the node responds with a negative or positive ACK, the ACK packet may cause collisions with the reservations made by its neighbor. The sendernode times out and retries later. In case the *ResvRTS*is received on a free slot, but the requested connection-slot is not free at the receiver node, the receiver sends a negative CTS(*ResvNCTS*)back to the sender node. On receiving this, the sender node reattempts following the sameprocedure but with another free connection-slot. If the real-time session gets finished, or a route break is detected by the sender node, the nodereleases the resources reserved for that session by sending a reservationrelease RTS(*ResvRelRTS*) packet. The *ResvRelRTS*packet is a broadcast packet. Nodes hearingthis packet update their reservation tables in order to free the corresponding connection slots. Incase the receiver node receives this (*ResvRelRTS*) packet, it responds by broadcasting a(*ResvRelCTS*) packet. The receiver's neighbor nodes, on receiving this (*ResvRelCTS*) packet, freeup the corresponding reservation slots. In case the downstream node of the broken link does notreceive the *ResvRelRTS*packet, since it also does not receive any DATA packet belonging to thecorresponding connection, it times out and releases the reservations made.
A QoS routing protocol is used with RTMAC to find an end-to-end path that matches the QoSrequirements (bandwidth requirements) of the user. The QoS routing protocol used here is anextension of the destination sequenced distance vector (DSDV) routing protocol. When a nodereceives a data packet for a new connection, the node reserves bandwidth on the forward link andforwards the packet to the next node on the path to the destination. In order to maintain a

consistent view of reservation tables of neighboring nodes at each node, each node transmits itsreservation information along with the route update packet, which is defined as part ofthe DSDV protocol. The routing protocol can specify a specific connection-slot to be reservedfor a particular connection; this gives flexibility for the routing protocol to decide on the positioning of the connection-slot. But generally, the first available connection-slot is used.
One of the main advantages of RTMAC is its bandwidth efficiency. Since nodes operate in theasynchronous mode, successive reservation slots may not strictly align with each other. Hencesmall fragments of free slots may occur in between reservation slots. If the free slot is justenough to accommodate a DATA and ACK packet, then RTMAC can make use of the free slotby transmitting *ResvRTS-ResvCTS-ResvACK*in some other free slot. Such small free slots cannotbe made use of in other protocols such as MACA/PR, which require the free slot to accommodate the entire RTS-CTS-DATA-ACK exchange. Another advantage of RTMAC is itsasynchronous mode of operation where nodes do not require any global time synchronization.

## 3.7 CONTENTION-BASED MAC PROTOCOLS WITH SCHEDULING MECHANISMS

Protocols that fall under this category focus on packet scheduling at the nodes and transmissionscheduling of the nodes. Scheduling decisions may take into consideration various factors suchas delay targets of packets, laxities of packets, traffic load at nodes, and remaining battery powerat nodes. In this section, some of the scheduling-based MACprotocols are described.

### *3.7.1 Distributed Priority Scheduling and Medium Access in Ad Hoc Networks*

This work, proposed in , presents two mechanisms for providing quality of service (QoS)support for connections in ad hoc wireless networks. The first technique, called distributedpriority scheduling (DPS), piggy-backs the priority tag of a node's current and head-of-linepackets on the control and data packets. By retrieving information from such packets transmittedin its neighborhood, a node builds a scheduling table from which it determines its rank(information regarding its position as per the priority of the packet to be transmitted next) compared to other nodes in its neighborhood. This rank is incorporated into the back-offcalculation mechanism in order to provide an approximate schedule based on the ranks of thenodes. The second scheme, called multi-hop coordination, extends the DPS scheme to carry outscheduling over multi-hop paths. The downstream nodes in the path to the destination increasethe relative priority of a packet in order to compensate for the excessive delays incurred by thepacket at the upstream nodes.

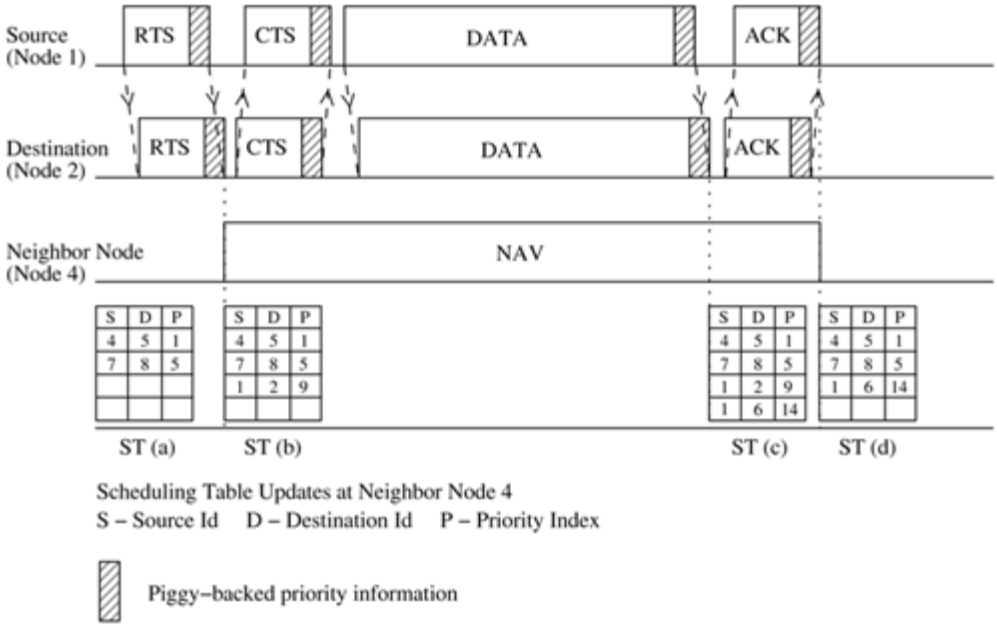
Distributed Priority Scheduling

The distributed priority scheduling scheme (DPS) is based on the IEEE 802.11 distributedcoordination function. DPS uses the same basic RTS-CTS-DATA-ACK packet exchangemechanism. The RTS packet transmitted by a ready node carries the priority tag/priority indexfor the current DATA packet to be transmitted. The priority tag can be the delay target for theDATA packet. On receiving the RTS packet, the intended receiver node responds witha CTS packet. The receiver node copies the priority tag from the received RTS packet and piggybacksit along with the source node id, on the CTS packet. Neighbor nodes receivingtheRTS or CTS packets (including the hidden nodes) retrieve the piggy-backed priority taginformation and make a corresponding entry for the packet to be transmitted, in their schedulingtables (STs). Each node maintains an ST holding information about packets, which wereoriginally piggy-backed on control and data packets. The entries in the ST are ordered accordingto their priority tag values. When the source node transmits a DATA packet, its head-of-linepacket information (consisting of the destination and source ids along with the priority tag) ispiggy-backed on the DATA packet (head-of-line packet of a node refers to the packet to betransmitted next by the node). This information is copied by the receiver onto the ACK packet it sends in response to the received DATA packet. Neighbor nodes receiving the DATA or ACKpackets retrieve the piggy-backed information and update their STs accordingly. When a nodehears an ACK packet, it removes from its ST any entry made earlier for the correspondingDATA packet.

Figure 3.25 illustrates the piggy-backing and table update mechanism.

Node 1 needs to transmita DATA packet (with priority index value 9) to node 2. It first transmits an RTSpacket carryingpiggy-backed information about this DATA packet. The initial state of the STof node 4 which isa neighbor of nodes 1 and 2 is shown in ST (a). Node 4, on hearing thisRTS packet, retrieves the piggybacked priority information and makes a corresponding entry in its ST, as shown in ST (b).The destination node 2 responds by sending a CTS packet. The actual DATA packet is sent bythe source node once it receives the CTS packet. This DATA packet carries piggy-backedpriority information regarding the head-of-line packet at node 1. On hearing this DATA packet,neighbor node 4 makes a corresponding entry for the head-of-line packet of node 1, in its ST. ST(c) shows the new updated status of the ST at node 4. Finally, the receiver node sends an ACKpacket to node 1. When this packet is heard by node 4, it removes the entry made for thecorresponding DATA packet from its ST. The state of the scheduling table at the end of this datatransfer session is depicted in ST (d).

**Figure 6.25. Piggy-backing and scheduling table update mechanism in DPS.**




In essence, each node's scheduling table gives the rank of the node with respect to other nodes inits neighborhood. This rank information is used to determine the back-off period to be taken bythe node. The back-off distribution is given by

$$\text{back} - \text{off} = \begin{cases} Uniform[0, (2^t CW_{min}) - 1] & r = 1,\ n < nmax \\ \alpha \times CW_{min} + Uniform[0, \gamma CW_{min} - 1] & r > 1,\ n = 0 \\ Uniform[0, (2^n \gamma CW_{min}) - 1] & r > 1, n \geq 1 \end{cases}$$

where *CWmin*is the minimum size of the contention window. *r* is the rank in the scheduling tableof the node's highest priority packet; *n* is the current number of transmission attempts made bythe node; *nmax*is the maximum number of retransmissions permitted; $\alpha$ is a constant; and $\gamma$ is aconstant that is used to control the congestion in the second attempt for the highest ranked nodes.

Multi-Hop Coordination

By means of the multi-hop coordination mechanism, the excess delay incurred by a packet at theupstream nodes is compensated for at the downstream nodes. When a node receives a packet, itwould have already received the priority index of the packet piggy-backed on theprevious RTS packet. In case the node is an intermediate node which has to further forward thepacket, the node calculates the new priority index of the DATA packet in a recursive fashion,

based on the received value of the priority index. If ${d_{i,j}}^k$ is the priority index assigned tothe *k*th packet of flow *i*with size ${l_i}^k$ at its *j*thhop, and if is the time at which the*k*th packet offlow *i*arrives at its first hop (the next hop node to the source node on the path to the destination),then the new priority index assigned to the received packet at intermediate node *j* is given as

$$d_{i,j}^{k} = \begin{cases} t_i^k + \delta_{i,1}^k, & j = 1 \\ d_{i,j-1}^k + \delta_{i,j}^k, & j > 1 \end{cases}$$

where the increment of the priority index $\delta_{i,j}^{k}$ is a non-negative function of *i, j,* $l_i^k$, and $t_i^k$

Because of this mechanism, if a packet suffers due to excess delay at the upstream nodes, thedownstream nodes increase the priority of the packet so that the packet is able to meet its end-toenddelay target. Similarly, if a packet arrives very early due to lack of contention at theupstream nodes, then the priority of that packet would be reduced at the downstream nodes. Any suitable scheme can be used for obtaining the values for $\delta_{i,j}^{k}$. One simple scheme, calleduniform delay budget allocation, works as follows. For a flow *i*with an end-to-end delay targetof *D,* the increment in priority index value for a packet belonging to that flow, at hop *j*, is given as $\delta_{i,j}^{k} = \frac{D}{K}$, where *K* is the length of the flow's path.

The distributed priority scheduling and multi-hop coordination schemes described above are fully distributed schemes. They can be utilized for carrying time-sensitive traffic on ad hoc wireless networks.

### *6.7.2 Distributed Wireless Ordering Protocol*

The distributed wireless ordering protocol (DWOP) [19] consists of a media access schemealong with a scheduling mechanism. It is based on the distributed priority scheduling schemeproposed in . DWOP ensures that packets access the medium according to the orderspecified by an ideal reference scheduler such as first-in-first-out (FIFO), virtual clock, orearliest deadline first. In this FIFO is chosen as the reference scheduler. In FIFO,packet priority indices are set to the arrival times of packets. Similar to DPS , control packetsare used in DWOP to piggy-back priority information regarding head-of-line packets of nodes.

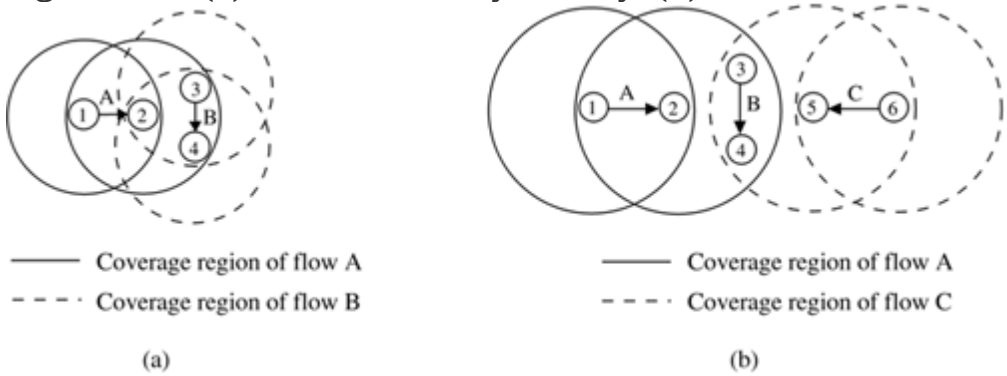
As the targeted FIFO schedule would transmit packets in order of the arrival times, each nodebuilds up a scheduling table (ST) ordered according to the overheard arrival times.

The key concept in DWOP is that a node is made eligible to contend for the channel only if itslocally queued packet has a smaller arrival time compared to all other arrival times in its ST(allother packets queued at its neighbor nodes), that is, only if the node finds that it holds the nextregion-wise packet in the hypothetical FIFO schedule. Two additional table managementtechniques, receiver participation and stale entry elimination, are used in order to keep the

actualschedule close to the reference FIFO schedule. DWOP may not suffer due to informationasymmetry. Since in most networks all nodes are not within the radio range of each other, atransmitting node might not be aware of the arrival times of packets queued atanother nodewhich is not within its direct transmission range. This information asymmetry might affect thefair sharing of bandwidth.

For example, in Figure 3.26 (a), the sender of flow B would be awareof the packets to be transmitted by the sender of flow A, and so it defers its transmissionwhenever a higher priority packet is queued at the sender of flow A. But the sender of flow A isnot aware of the arrival times of packets queued at the sender of flow B and hence it concludesthat it has the highest priority packet in its neighborhood. Therefore, node 1 unsuccessfully triesto gain access to the channel continuously. This would result in flow B receiving an unfair highershare of the available bandwidth. In order to overcome this information asymmetry problem,the *receiver participation* mechanism is used.

**Figure 3.26. (a) Information asymmetry. (b) Perceived collisions.**




In the receiver participation mechanism, a receiver node, when using its ST information, findsthat the sender is transmitting out of order, that is, the reference FIFO schedule is being violated,an *out-of-order notification* is piggy-backed by the receiver on the control packets (CTS/ACK) itsends to the sender. In essence, information regarding the transmissions taking place in the twohop neighborhood of the sender is propagated by the receiver node whenever it detectsa FIFO schedule violation. Since the notification is sent only when a FIFOviolation is detected,the actual transmission may not strictly follow the FIFO schedule; rather, it approximatesthe FIFO schedule. On receiving an out-of-order packet from a sender node, the receiver nodetransmits a notification to the sender node carrying the actual rank *R*of the sender with respect tothe receiver's local ST. On receiving this out-of-order notification, the sender node goes into aback-off state after completing the transmission of its current packet. The back-offperiod *Tback − off* is given by

$$T_{back-off} = R \times (EIFS + DIFS + T_{success} + CW_{min})$$

where *Tsuccess*is the longest possible time required to transmit a packet successfully, includingthe RTS-CTS-DATA-ACK handshake. Thus the node backs off, allowing higher priority packetsin the neighborhood of the receiver to

get transmitted first. In order to obtain aperfect FIFO schedule, the receiver can very well be made not to reply to the out-of-orderrequests (RTS) of the sender. This would cause the sender to time out and back off, therebyavoiding any out-of-order transmission. But since the sender has already expended its resourcesin transmitting the RTS successfully, it is allowed to complete the transmission of its currentpacket. This is a trade-off between achieving perfect FIFO scheduling and high systemutilization. Since in DWOP a node's access to the medium is dependent on its rank in thereceiver node's ST (the rank of a node denotes the position of the node's entry in the receivernode's ST as per the priority of the corresponding packet), information maintained inthe ST must be consistent with the actual network scenario. The stale entry elimination mechanism makes sure that the STs are free of stale entries. An entry is deleted from the STonlyafter an ACK packet for the corresponding entry is heard by the node. In case the ACK packetcollides at the node, the corresponding entry in the ST will never be removed. This may cause alarge deviation from the ideal FIFO schedule.

Figure 3.26 (b) shows an example-perceivedcollisions scenario. The sender and receiver of flow B might have stale entries because ofcollisions caused by packets belonging to flow A and flow C at the sender and receiver of flow B. It can be observed that, in case there is a stale entry in the ST of a node, the node's own headof-line packet position remains fixed, while other entries below the head-of-line entry keepchanging. The above observation is used as a stale entry detection method. Thus, when a nodeobserves that its rank remains fixed while packets whose priorities are below the priority of itshead-of-line packet are being transmitted, it concludes that it may have one or more stale entriesin its ST. The node simply deletes the oldest entry from its ST, assuming it to be the stale entry.This mechanism thus eliminates stale entries from the STs of nodes.

In summary, DWOP tries to ensure that packets get access to the channel according to the orderdefined by a reference scheduler. The above discussion was with respect to the FIFOscheduler.Though the actual schedule deviates from the ideal FIFO schedule due to information asymmetryand stale information in STs, the receiver participation and the stale entry elimination mechanisms try to keep the actual schedule as close as possible to the ideal schedule.

### *3.7.3 Distributed Laxity-Based Priority Scheduling Scheme*

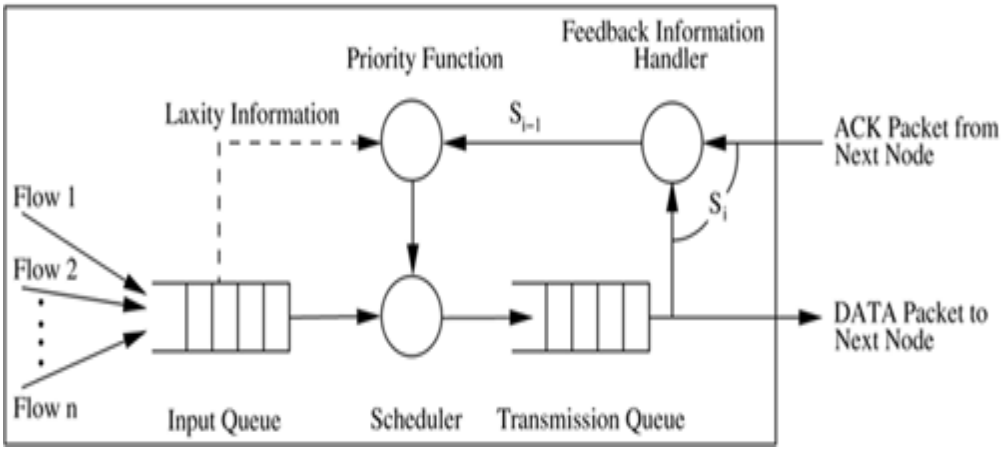
The distributed laxity-based priority scheduling (DLPS) scheme is a packet schedulingscheme, where scheduling decisions are made taking into consideration the states of neighboringnodes and the feedback from destination nodes regarding packet losses. Packets are reorderedbased on their uniform laxity budgets (ULBs) and the packet delivery ratios of the flows to which they belong.Each node maintains two tables: scheduling table (ST) and packet delivery ratio table (PDT).

The ST contains information about packets to be transmitted by the node and packets overheardby the node, sorted according to their *priority index* values. Priority index expresses the priorityof a packet. The lower the priority index, the higher the packet's priority. The PDT contains thecount of packets transmitted and the count of acknowledgment (ACK) packets received for every flow passing through the node. This information is used for calculating current packet deliveryratio of flows.A node keeps track of packet delivery ratios (used for calculating priority index of packets) of allflows it is aware of by means of a feedback mechanism.

Figure 3.27 depicts the overallfunctioning of the feedback mechanism. Incoming packets to a node are queued in the node'sinput queue according to their arrival times. The scheduler sorts them according to their priority values and inserts them into the transmission queue. The highest priority packet from this queueis selected for transmission. The node, after transmitting a packet, updates the count of packetstransmitted so far in its PDT. The destination node of a flow, on receiving data packets, initiatesa feedback by means of which the count of DATA packets received by it is conveyed to the source through ACK packets traversing the reverse path. These two pieces of information,together denoted by *Si* in Figure 3.27, are received by the feedback information handler (FIH).

The FIH, in parallel, also sends the previous state information $Si-1$ to the priority function module(PFM). The ULB of each packet in ST is available at the node .This information is also sent to PFM, which uses the information fed to it to calculate the priority indices of packets in the ST.

**Figure 3.27. Feedback mechanism.** ***Reproduced with permission from [20], © Elsevier, 2004.***




Using the count of DATA packets transmitted (*pktsSent*) and count information carried by ACKpackets (*acksRcvd*), available in PDT, packet delivery ratio (*PDR*) of the flow at any given timeis computed as

$$PDR = \frac{acksRcvd}{pktsSent}$$

**(6.7.1)**

Priority index of a packet (*PI*) is defined as

**(6.7.2)**

$$PI = \frac{PDR}{M} \times ULB$$

Here, $ULB = \frac{deadline - currentTime}{remHops}$ is the uniform laxity budget of the packet, and *M* is a userdefinedparameter representing the desired packet delivery ratio for the flow.*deadline* is the endto-end deadline target of the packet and is equal to (*packet creation time + end-to-end delaytarget*). *CurrentTime* denotes the current time according to the node's local clock.

When greater numbers of packets belonging to a flow meet their delay targets,

the term $\frac{PDR}{M}$ would have a high value. Hence priority index would be high for packets of that flow, andtherefore the actual priority of the packets would be low. When very few packets of a flow meettheir delay targets, the value of

$\frac{PDR}{M}$ would be much less, thereby lowering the priority indexand increasing the priority of packets of that flow. *ULB* also plays an equally important role. Since *remHops*, the number of hops remaining to be traversed, is in the denominator of theexpression for *ULB*, when a packet is near its source and needs to traverse several hops to reachits destination, its priority index value will be lowered, thereby increasing its priority. When itnears its destination, the fewer number of hops to be traversed tends to increase the priorityindex, thereby lowering its priority.

RTS and CTS packets transmitted by a node are modified to carry piggy-backed informationregarding the *highest priority packet queued at the node.* Similarly, DATA and ACK packetstransmitted by a node carry piggy-backed information corresponding to the *highest prioritypacket entry in the ST of the node.* A node hearing any packet retrieves the piggy-backed priorityinformation, calculates the priority index of the corresponding packet, and adds a correspondingentry in its ST.

A DATA packet also carries information about itself. The end-to-end delay target, remainingnumber of hops, actual source ID, and the flow ID constitute this information. A node, onreceiving a DATA packet, using the above information and the information maintained inthePDT, can obtain the priority index of the packet (*PI*), as given in Equation 3.7.2. Since the

priority index of a packet keeps changing with time, it needs to be updated constantly. Eachnode, before calculating its backoff period and before inserting a new entry into its ST,recalculates and updates the priority index of each entry in its ST.

When a node hears a DATA packet, if an entry for the corresponding packet exists in its ST, then that entry is deleted. The sender node deletes its entry from the ST only when an ACK for the transmitted DATA packet is received. It may happen that a DATA packet transmitted is notheard by a node which had previously been located within the transmission range of a sendernode holding the highest priority packet in its locality. This might be because of reasons such asnode mobility and channel errors. In such cases, the stale entries might affect the desiredscheduling of packets. Another reason for stale entries in the ST is that, when the network load ishigh, some of the packets would miss their deadline targets while waiting in the node's queueitself. Such packets will never be transmitted. In order to remove stale entries, whenever tableupdates are performed, entries whose deadline targets have been missed already are deleted fromthe ST.

The objective of the back-off mechanism used in DLPS is to reflect the priority of the node'shighest priority packet on the back-off period to be taken by the node. If *r* is the rank (rank of anentry is the position of that entry in the scheduling table of the node), in ST of the node, of thecurrent packet to be sent, *n* is the number of retransmission attempts made for the packet,and *nmax*is the maximum number of retransmission attempts permitted, then the back-off interval is given by

**(6.7.3) (6.7.4) (6.7.5)**

$$\text{back}-\text{off} = \begin{cases} Uniform[0, (2^n \times CW_{min}) - 1] \\ \quad \text{if } r = 1 \text{ and } n \leq nmax \\ \\ \frac{PDR}{M} \times CW_{min} + \quad Uniform[0, CW_{min} - 1] \\ \quad \text{if } r > 1 \text{ and } n = 0 \\ \\ ULB \times CW_{min} + \quad Uniform[0, (2^n \times CW_{min}) - 1] \\ \quad \text{otherwise} \end{cases}$$

where *CWmin*is the minimum size of the contention window, and *M* is the desired packet deliveryratio.

This means that if the packet has the highest rank in the broadcast region of the node, then it hasthe lowest back-off period according to Equation 3.7.3 and faces much less contention. Else, if itis the first time the packet is being transmitted, then the back-off distribution follows the secondscheme as in Equation 3.7.4, where the back-off is more than that for the first case. Here the current *PDR* of the flow affects the back-off period. If *PDR* is considerably less, then the firstterm would be less, and if it is high, then the first term would be high and the node would have towait for a longer time. Finally, if the packet does not fit into these two categories, then the backoffvalue is as per the third

scheme in Equation 3.7.5, and is the longest of the three cases. Thehigher the value of *ULB*, the longer the back-off period.DLPS delivers a higher percentage of packets within their delay targets and has lower averageend-to-end delay in packet delivery when compared to the 802.11 DCF and the DPS schemes.

## 3.9 OTHER MAC PROTOCOLS

### *3.9.1 Multichannel MAC Protocol*

The multichannel MAC protocol (MMAC) uses multiple channels for data transmission.There is no dedicated control channel. *N* channels that have enough spectral separation betweeneach other are available for data transmission.Each node maintains a data structure called *PreferableChannel List* (PCL). The usage of thechannels within the transmission range of the node is maintained in the PCL. Based on theirusage, channels can be classified into three types.

• High preference channel (HIGH): The channel has been selected by the current node and isbeing used by the node in the current beacon interval Since a node has only one transceiver, there can be only one HIGH channel at a time.

• Medium preference channel (MID): A channel which is free and is not being currently used inthe transmission range of the node is said to be a medium preference channel. If there is noHIGH channel available, a MID channel would get the next preference.

• Low preference channel (LOW): Such a channel is already being used in the transmission rangeof the node by other neighboring nodes. A counter is associated with each LOW state channel.

For each LOW state channel, the count of source-destination pairs which have chosen thechannel for data transmission in the current beacon interval is maintained.

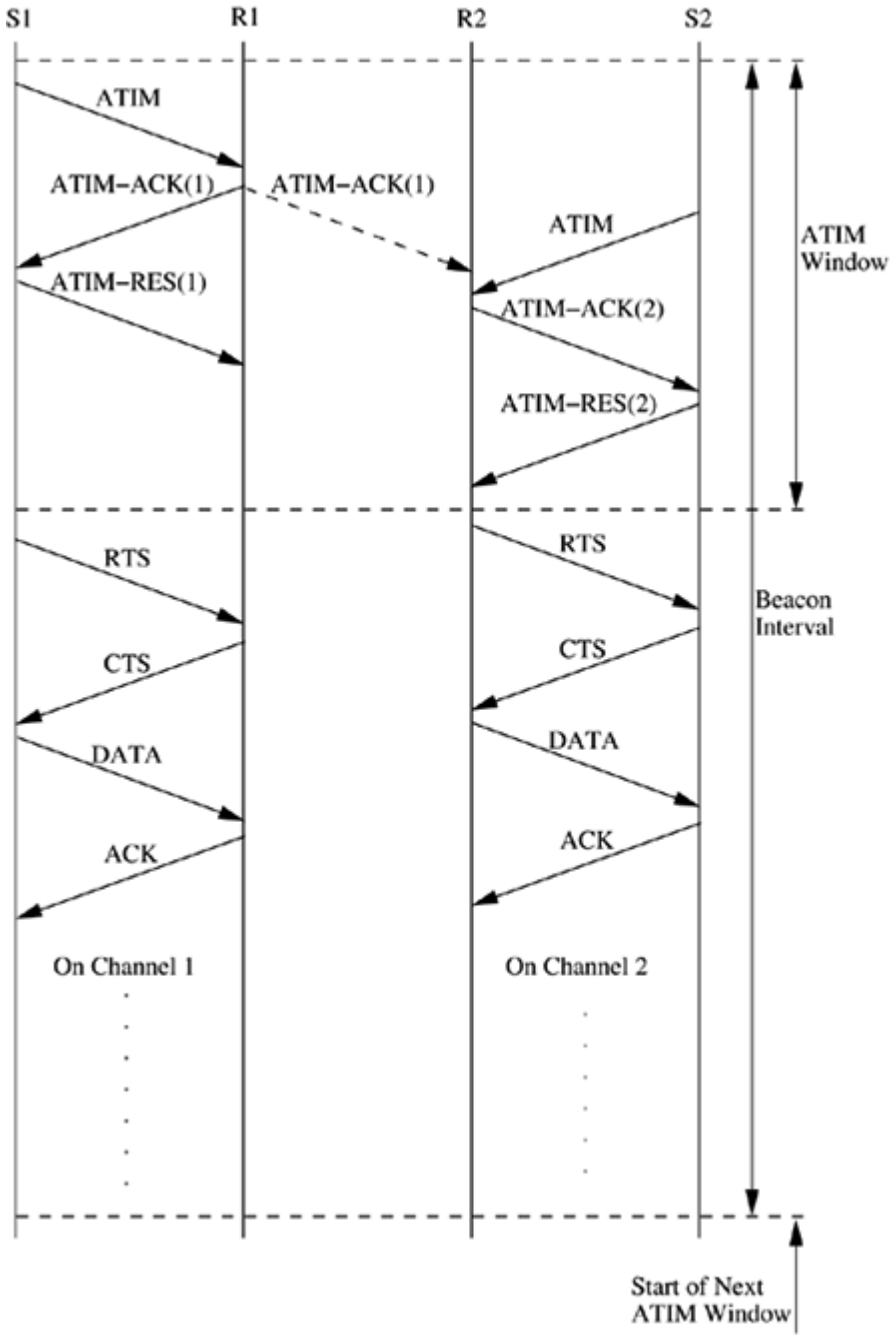
Time is divided into beacon intervals and every node is synchronized by periodic beacon transmissions. So for every node, the beacon interval starts and ends almost at the same time. Atthe start of every beacon interval, there exists a time interval called the ad hoc traffic indicationmessages (ATIM) window. This window is used by the nodes to negotiate for channels fortransmission during the current beacon interval. ATIM messages such as ATIM,ATIM-ACK(ATIM-acknowledgment), and ATIM-RES (ATIM-reservation) are used for this negotiation.The exchange of ATIM messages takes place on a particular channel called the*default channel.* The default channel is one of the multiple available channels. This channel is used forsending DATA packets outside the ATIM window, like any other channel. A node that wants totransmit in the current beacon interval sends an ATIM packet to the intended destination node.

The ATIM message carries the PCL of the transmitting node. The destination node, uponreceiving the packet, uses the PCL carried on the packet and its ownPCL to select a channel. Itincludes this channel information in the ATIM-ACK packet it sends to the source node. Thesource node, on receiving the ATIM-ACK packet, determines whether it can transmit on thechannel indicated in the ATIM-ACK message. If so, it responds by sending the destination node an ATIM-RES packet. The ATIM-ACK and ATIM-RES packets are also used to notify theneighbor nodes of the receiver and sender nodes, respectively, about the channel that is going tobe used for transmission in the current beacon interval. The nodes that hear these packets updatetheir PCLs accordingly. At the end of theATIM window, the source and destination nodes switch to the agreed-upon channel and start communicating by exchanging RTS/CTS control packets. Ifthe source node is not able to use the channel selected by the destination, it cannot transmitpackets to that destination in the current beacon interval. It has to wait for the next beaconinterval for again negotiating channels. The ATIM packets themselves may be lost due tocollisions; in order to prevent this, each node waits for a randomly chosen back-off period (between 0 and *CWmin*) before transmitting the ATIM packet.

Operation of the MMAC protocol is illustrated in Figure 3.34.

At the beginning of the beaconinterval, source node S1 sends an ATIM message to receiver R1. Receiver R1 responds bysending an ATIM-ACK packet (ATIM-ACK(1)) carrying the ID 1 of the channel it prefers(inFigure 3.34 the number within parentheses indicates the ID of the preferred channel). NodeS1, on receiving this packet, confirms the reservation by sending an ATIM-RES packet (ATIMRES(1)) for channel 1. The ATIM-ACK(1) packet sent by receiver R1 is also overheard by nodeR2. When node R2 receives an ATIM packet from source S2, it chooses a different channel withID 2, and sends the channel information to source S2 through the ATIM-ACK packet (ATIMACK(2)). Since channel 2 is agreeable to node S2, it responds by sending theATIM-RES(2)packet, and the reservation gets established. Once the ATIM window finishes, the datatransmission (through RTS-CTS-DATA-ACK packet exchange) between node pairs S1-R1 andS2-R2 takes place on the corresponding reserved channels, channel 1 and channel 2,respectively.

**Figure 3.34. Operation of MMAC protocol.**




In this protocol, it is the receiver node that plays a dominant role in channel selection. In case allchannels are in use at the receiver, even then the receiver selects one of the channels. Since theactual data packet transmissions are protected by the RTS/CTS control packet exchange, thenodes transmitting packets on the same channel need to contend for the channel, asin IEEE 802.11 for transmitting packets. The protocol also employs a power-saving mode. In case a node realizes after the ATIM window that it is neither going to transmit packets nor goingto receive packets, then the node goes into a power-saving *doze*mode.Channel selection is done at the receiver in the following manner. The receiver node usesthePCL on the received ATIM packet and its own PCL

for selecting the best possible channel forcommunication with the source node. The channel selection procedure tries to balance thenetwork load on the channels. If a receiver node R receives an ATIM packet from a source node S, it selects a channel as below.

• If there exists a HIGH state channel in node R's PCL, then that channel is selected.

• Else if there exists a HIGH state channel in the PCL of node S, then this channel is selected.

• Else if there exists a common MID state channel in the PCLs of both node S and node R, thenthat channel is selected. If many such channels exists, one of them is selected randomly.

• Else if there exists a channel which is in the MID state at only one of the two nodes, then thatchannel is chosen. If many such channels exist, one of them is selected randomly.

• If all channels in both PCLs are in the LOW state, the counters of the corresponding channels atnodes S and R are added, and the channel with the least count is selected. Ties are brokenarbitrarily. MMAC uses simple hardware. It requires only a single transceiver. It does not have anydedicated control channel. The throughput of MMAC is higher than that of IEEE 802.11 whenthe network load is high. This higher throughput is in spite of the fact that in MMAC only asingle transceiver is used at each node. Unlike other protocols, the packet size in MMAC neednot be increased in order to take advantage of the presence of an increased number of channels.

### *3.9.2 Multichannel CSMAMAC Protocol*

In the multichannel CSMA MAC protocol (MCSMA) , the available bandwidth is dividedinto several channels. A node with a packet to be transmitted selects an idle channel randomly.The protocol also employs the notion of *soft* channel reservation, where preference is given tothe channel that was used for the previous successful transmission. Though the principle usedin MCSMA is similar to the frequency division multiple access (FDMA) schemes used in cellular networks, the major difference here is that there is no centralized infrastructure available,and channel assignment is done in a distributed fashion using carrier-sensing. The operation ofthe protocol is discussed below.

The total available bandwidth is divided into $N$ non-overlapping channels ($N$ is independent ofthe number of hosts in the network), each having a bandwidth of ($W/N$), where $W$ is the totalbandwidth available for communication. The channels may be created in the frequency domain(FDMA) or in the code domain (CDMA). Since global synchronization between nodes is notavailable in ad hoc wireless networks, channel division in the time domain (TDMA) is not used.An idle node (which is not transmitting packets) continuously monitors all the $N$ channels. Achannel whose total received signal strength1 (TRSS) is below the sensing threshold (ST) of thenode is marked IDLE by the node. The

time at which TRSS drops below ST is also noted foreach IDLE channel. Such channels are put in the *free-channels* list.
The total received signal strength of a signal is calculated by the sum of contributions arisingfrom the various individual multipath components of the signal.When an idle node receives a packet to be transmitted, it does the following. If the free-channelslist is empty, it waits for any channel to become IDLE. It then waits for an additional long interframespace (LIFS) time, and for another random access back-off period. If the channel remainsidle for this entire wait period, then the node starts transmitting its packets on this channel. Incase the free-channels list is non-empty, the node first checks whether the channel it used for itsmost recent successful transmission is included in the list. If so, the node uses this channel for itsnew transmission. Otherwise, one among the IDLE channels available in the free-channels list israndomly chosen (using a uniform random number generator).
Before the actual packet transmission, the node checks the TRSS of the chosen channel. If it hadremained below ST for at least LIFS period of time, then the node immediately initiates packettransmission. Otherwise, the node initiates back-off delay after the LIFS time period. During theback-off period, if the TRSS of the chosen channel goes above ST, then the back-off isimmediately canceled. A new back-off delay is scheduled when the TRSS again goes below the ST. After successfully transmitting a packet (indicated by an acknowledgment from thereceiver), the sender node notes the ID of the channel used. This channel would be givenpreference when a new channel is to be selected for its next transmission.
When the number of channels *N* is sufficiently large, each node tends to *reserve* a channel foritself. This is because a node prefers the channel used in its last successful transmission for itsnext transmission also. This reduces probability of two contending nodes choosing the samechannel for transmission. Nodes are expected to dynamically select channels for transmissions ina mutually exclusive manner, so as to enable parallel interference-free transmissions. Even athigh traffic loads, due to the tendency of every node to choose a*reserved* channel for itself, thechances of collisions are greatly reduced.The number of channels into which the available bandwidth is split is a very important factoraffecting the performance of the protocol. If the number of channels is very large, then theprotocol results in very high packet transmission times.

### *3.9.3 Power Control MAC Protocol for Ad Hoc Networks*

The power control MAC protocol (PCM) [26] allows nodes to vary their transmission powerlevels on a per-packet basis. The PCM protocol is based on the power control protocol used in, which is referred to as the *BASIC* protocol in this section. In what follows, the working ofthe BASIC power control protocol is briefly described. This is followed by a discussion of
the PCM protocol.

In the BASIC scheme, the RTS and CTS packets are transmitted with maximum power *pmax*.The RTS-CTS handshake is used for deciding upon the transmission power for thesubsequent DATA and ACK packet transmissions. This can be done using two methods. In the first method, source node A transmits the RTS with maximum power *pmax*. This RTS is received at the receiver with signal level *pr*. The receiver node B can calculate the minimum requiredtransmission power level *pdesired*for the DATA packet, based on the received power level *pr* , thetransmitted power level *pmax*, and the noise level at receiver B. Node B then specifies this *pdesired*inthe CTS packet it transmits to node A. Node A transmits the DATA packet using power level *pdesired*. In the second method, when the receiver node B receives anRTS packet, it respondswith a CTS packet at the usual maximum power level *pmax*. When the source node receivesthis CTS packet, it calculates *pdesired*based on the received power level*pr*and transmitted powerlevel *pmax*as

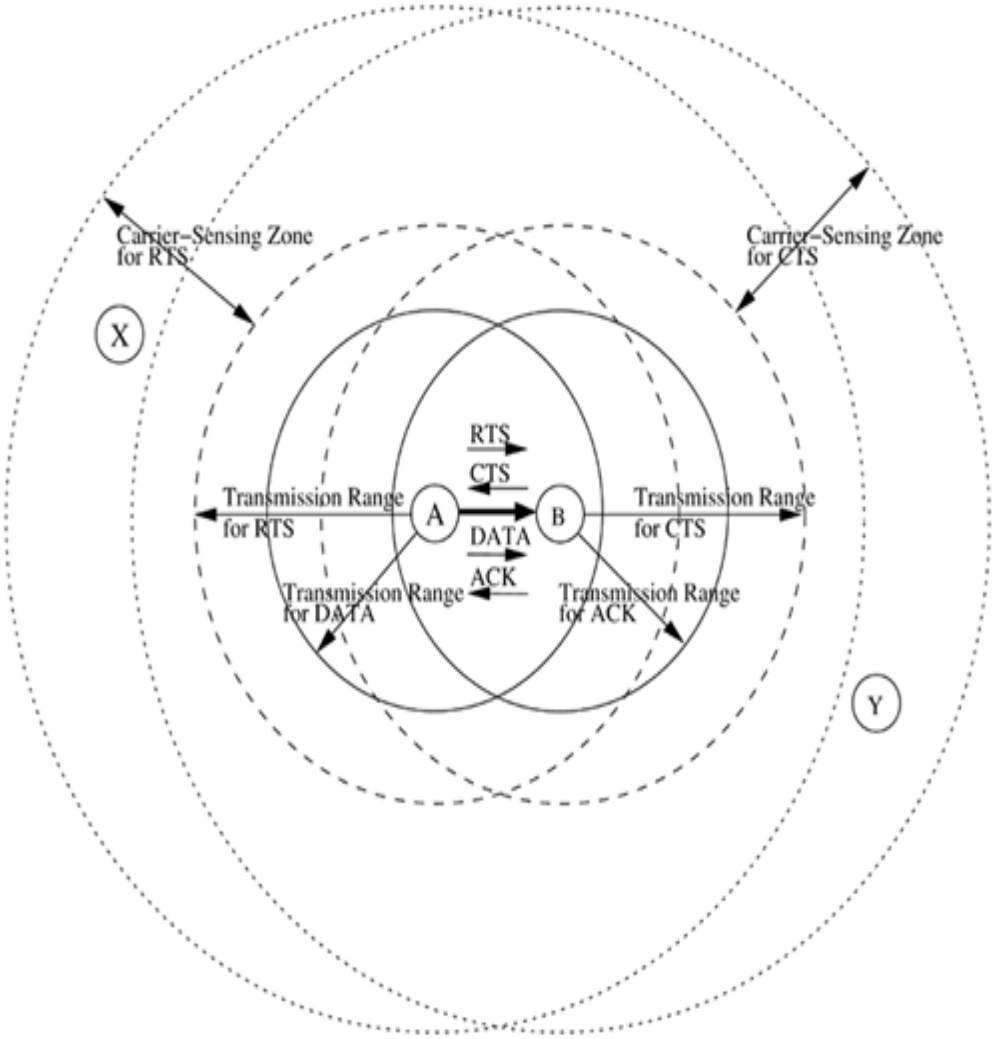
$$p_{desired} = \frac{p_{max}}{p_r} \times Rx_{thresh} \times c$$

where *Rx thresh* is the minimum necessary received signal strength and *c* is a constant. The sourcenode uses power level *pdesired*to transmit the DATA packet. Similarly, the receiver uses the signalpower of the received RTS packet to determine the power level to be used,*pdesired*, for the ACKpacket. This method assumes the attenuation between the sender and receiver nodes to be the same in both directions. It also assumes the noise level at the nodes to be below a certainpredefined threshold value.

Thus, the BASIC scheme uses maximum transmit power for RTS and CTS packets, and onlynecessary power levels for the DATA and ACK packets. But this scheme has a drawback.

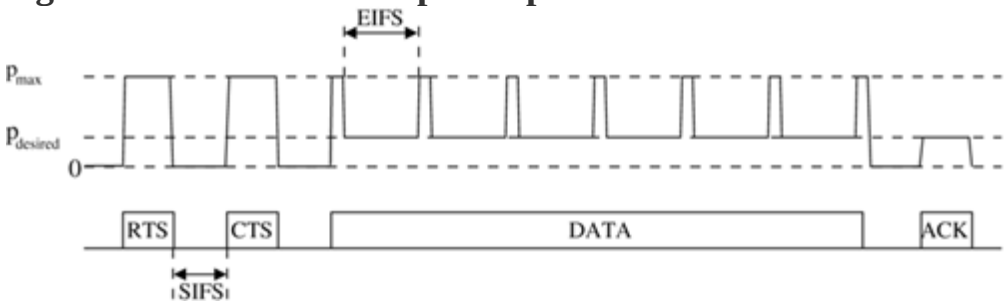
Consider Figure 3.35. Node A sends an RTS to node B, for which node B sends a CTS packet.Since these packets are sent at maximum power, nodes X and Y that are located in the carriersensingzones of nodes A and B, respectively (when a node N1 is in the carrier-sensing zone ofnode N2, node N1 can sense the signal from node N2, but the received signal strength is not highenough to decode it correctly), defer their transmissions for a sufficient enough period of time[extended inter-frame space (EIFS) period of time] so as to not interfere with the RTSCTSexchange. But since the DATA and ACK transmissions use only the minimum necessarypower, the DATA transmitted by node A cannot be sensed by node X, and the ACK packettransmitted by node B cannot be sensed by node Y. So if nodes X and Y transmit afterthe EIFS period (which is set in their NAVs on sensing the RTS or CTS packets), the packettransmitted by node X would collide at node A with the ACK packet from node B, and the packet transmitted by node Y would collide with the DATA packet at node B.

**Figure 3.35. Packet transmission in BASIC scheme.**




PCM modifies this scheme so as to minimize the probability of collisions. The source andreceiver nodes transmit the RTS and CTS packets, as usual, with maximum power *pmax*. Nodes inthe carrier-sensing zones of the source and receiver nodes set their NAVs for EIFSduration whenthey sense the signal but are not able to decode it. The source node generally transmits withminimum necessary power, as in the BASIC scheme. But, in order to avoid collisions with packets transmitted by the nodes in its carrier-sensing zone, the source node transmits the DATApacket at maximum power level *pmax*periodically. The duration of each such transmission mustbe larger than the time required for physical carrier-sensing. Since the nodes in the carriersensingzone defer their transmissions for EIFS duration if they are not able to decode thereceived signal, the transmit power for the DATA packet is increased (and brought down back tooriginal level) every EIFS duration. The power level changes forRTS-CTS-DATA-ACKtransmissions is depicted in Figure 6.36. Thus this protocol prevents collisions of ACK packetsat the sender node.

**Figure 3.36.Transmission power pattern in PCM.**




Hence with the above simple modification, the PCM protocol overcomes the problems faced inthe BASIC scheme. PCM achieves throughput very close to that of the 802.11 protocol whileusing much less energy.

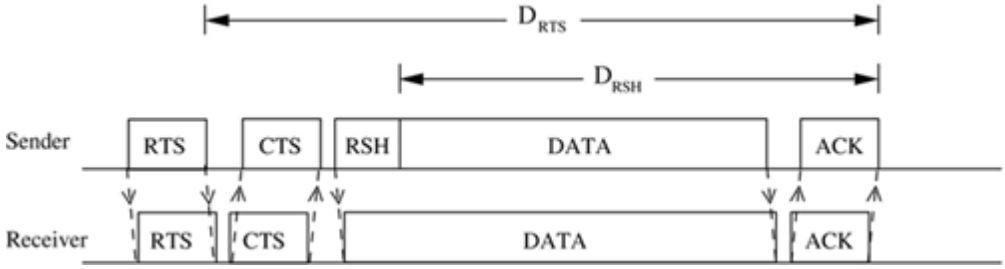
### *3.9.4 Receiver-Based Autorate Protocol*

The receiver-based autorate protocol (RBAR) uses a novel rate adaptation approach. Therate adaptation mechanism is at the receiver node instead of being located at the sender. Rateadaptation is the process of dynamically switching data rates in order to match the channelconditions so that optimum throughput for the given channel conditions is achieved. Rateadaptation consists of two processes, namely, channel quality estimation and rate selection. Theaccuracy of the channel quality estimates significantly influences the effectiveness of the rate adaptation process. Therefore, it is important that the best available channel quality estimates areused for rate selection. Since it is the channel quality at the receiver node which determineswhether a packet can be received or not, it can be concluded that the best channel quality
estimates are available at the receiver. The estimates must be used as early as possible beforethey get stale. If the sender is to implement the rate adaptation process, significant delay wouldbe involved in communicating the channel quality estimates from the receiver to the sender,which may result in the estimates becoming stale before being used. Therefore,the RBAR protocol advocates for rate adaptation at the receiver node rather than at the sender.
Rate selection is done at the receiver on a per-packet basis during the RTS-CTS packetexchange. Since rate selection is done *during* the RTS-CTS exchange, the channel qualityestimates are very close to the actual transmission times of the data packets. This improves theeffectiveness of the rate selection process. The RTS and CTS packets carry the chosenmodulation rate and the size of the data packet, instead of carrying the duration of thereservation.

The packet transmission process is depicted in Figure 3.37. The sender node choosesa data rate based on some heuristic and inserts the chosen data rate and the size of the data packetinto the RTS. When a neighbor node receives this RTS, it calculates the duration of thereservation *DRTS* using the data rate and packet size carried on the RTS. The neighbor node thenupdates its NAV accordingly to reflect the reservation. While receiving the data packet, the receiver node generates an estimate of the channel conditions for the impending data transfer.Based on this estimate, it chooses an appropriate data rate. It stores

the chosen data rate and thesize of the packet on the CTS packet and transmits the CTS to the sender. Neighbor nodesreceiving the CTS calculate the expected duration of the transmission and update their NAVsaccordingly. The source node, on receiving the CTSpacket, responds by transmitting the datapacket at the rate chosen by the receiver node.

**Figure 3.37. Packet transmission in RBAR.**




If the rates chosen by the sender and receiver are different, then the reservation Duration*DRTS* calculated by the neighbor nodes of the sender would not be valid. *DRTS* time period,which is calculated based on the information carried initially by the RTS packet, is referred toas *tentative reservation.* In order to overcome this problem, the sender node sends the datapacket with a special MAC header containing a *reservation subheader* (RSH). The RSHcontains a subset of header fields already present in the IEEE 802.11 data frame, along with a checksequence for protecting the subheader. The fields in the RSH contain control information fordetermining the duration of the transmission. A neighbor node with tentative reservation entriesin its NAV, on hearing the data packet, calculates *DRSH* , the new reservation period, and updatesits NAV to account for the difference between *DRTS* and*DRSH*.

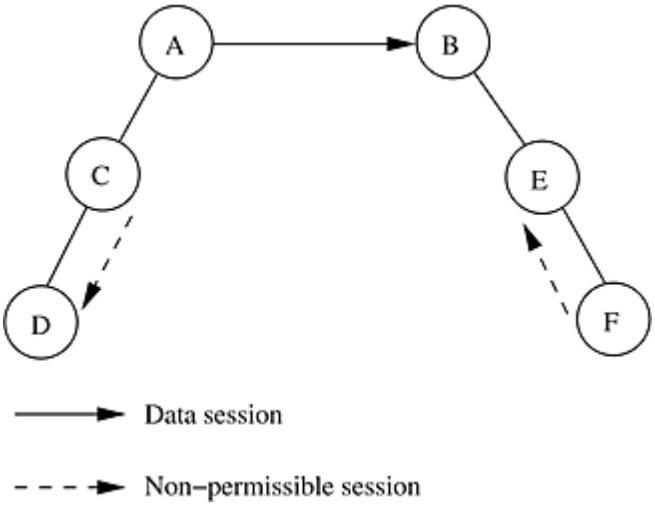
For the channel quality estimation and prediction algorithm, the receiver node uses a sample ofthe instantaneous received signal strength at the end of RTS reception. For the rate selectionalgorithm, a simple threshold-based technique is used. Here the rate is chosen by comparing thechannel quality estimate [*e.g.*, signal to noise ratio (SNR)] against a series of thresholdsrepresenting the desired performance bounds (*e.g.*, a series of SNR thresholds). The modulation scheme with the highest data rate, satisfying the performance objective for the channel qualityestimate, is chosen.RBAR employs an efficient quality estimation mechanism, which leads to a high overall systemthroughput. RBAR can be easily incorporated into many existing medium access controlprotocols.

### *3.9.5 Interleaved Carrier-Sense Multiple Access Protocol*

The interleaved carrier-sense multiple access protocol (ICSMA) [29] efficiently overcomes theexposed terminal problem faced in ad hoc wireless networks. The inability of a source node totransmit, even though its transmission may not affect other ongoing transmissions, is referred toas the exposed terminal problem. For example, consider the topology shown inFigure 3.38. Here,when a transmission is going from node A to node B, nodes C and F would not be permitted totransmit to nodes D and E, respectively. Node C is

called a sender-exposed node, and node E iscalled a receiver-exposed node. The exposed terminal problem reduces the bandwidth efficiency
of the system.

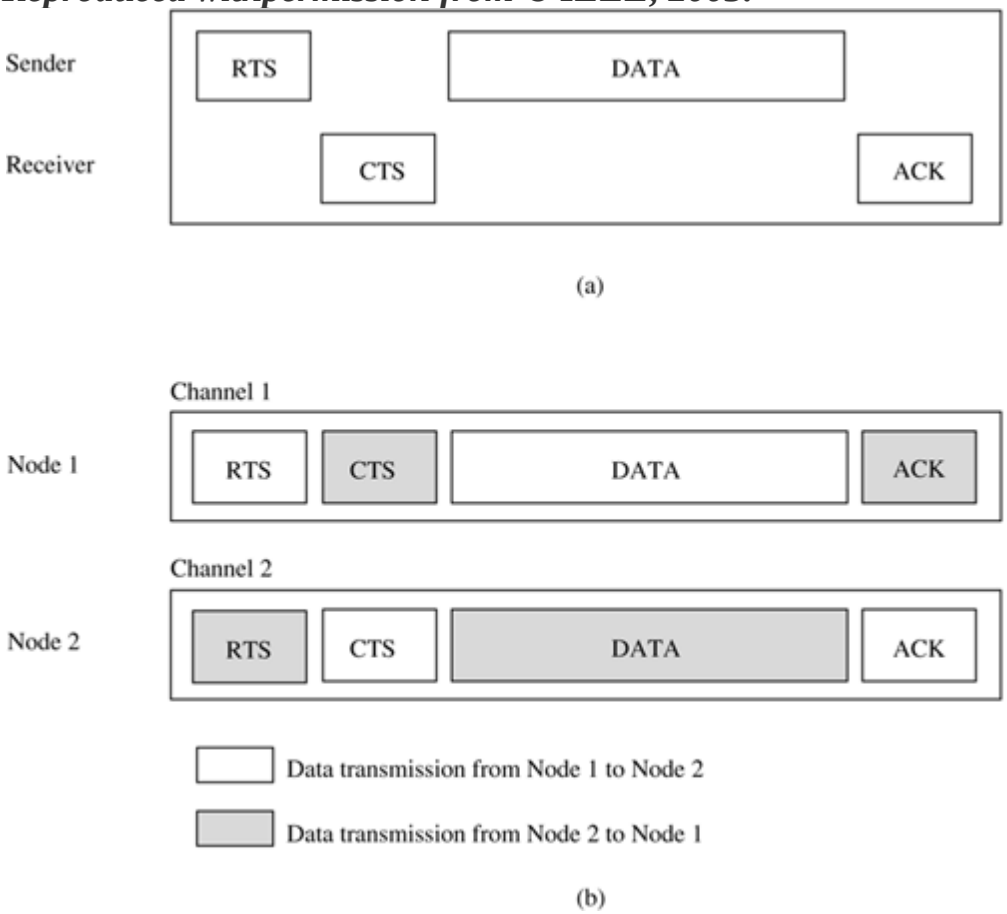
**Figure 3.38. Exposed terminal problem.**
***Reproduced with permission from [29], © IEEE,2003.***




In ICSMA, the total available bandwidth is split into two equal channels (say, channel 1 andchannel 2). The handshaking process is interleaved between the two channels, hence the nameinterleaved carrier-sense multiple access. The working of ICSMA is very simple. It uses thebasic RTS-CTS-DATA-ACK exchange mechanism used in IEEE 802.11 DCF. If the sourcenode transmits the RTS packet on channel 1, the receiver node, if it is ready to accept packets from the sender, responds by transmitting the CTS packet on channel 2. Each node maintains adata structure called extended network allocation vector (E-NAV), which is analogous to thenetwork allocation vector (NAV) used in IEEE 802.11 DCF. On receiving an RTS packet, thereceiver node checks its E-NAV and finds out whether free time slots are available. It sendsthe CTS only if free slots are available. The source node, on receiving this CTS, transmits the DATA packet on channel 1. The receiver acknowledges the reception of the DATA packet bytransmitting the ACK on channel 2.
The ICSMA channel access mechanism is illustratedinFigure 6.39.
Figure 6.39 (a) shows the RTS-CTS-DATA-ACK exchange of802.11 DCF.
Figure 6.39 (b) shows simultaneous data packet transmissions between two nodesin ICSMA. After transmitting an RTS or a DATA packet on a channel, the sender node waits onthe other channel for the CTS or ACK packet. If it does not receive any packet on the otherchannel, it assumes that the RTS or DATA packet it transmitted was lost, and retries again.Similarly at the receiver, after transmitting a CTS frame on one of the channels, the receiver

node waits on the other channel for the DATA packet. If the DATA packet is not received withinthe timeout period, it retransmits the CTS packet.

**Figure 3.39. Packet transmissions in (a) 802.11 DCF and (b) ICSMA.** ***Reproduced withpermission from © IEEE, 2003.***




The performance improvement of ICSMA is attributed to the following facts:
• Nodes that hear RTS in a particular channel (say, channel 1) and do not hear thecorresponding CTS on the other channel (channel 2) conclude that they are only sender-exposedin channel 1. Therefore, if they have packets to send, they can use channel 1 to transmit RTS toother nodes. This would not have been possible in 802.11 DCF, where transmissions by a senderexposednode would have collided with the corresponding currently active sender node.
• Nodes that hear only the CTS in a particular channel (say, channel 1) and had not heard thecorresponding RTS on the other complementary channel (channel 2) realize that they are onlyreceiver-exposed on channel 1 to the on-going transmission. If they receive any RTS on channel2, they would not refrain from sending a CTS on channel 1 for the received RTS. This wouldalso not have been possible in 802.11 DCF, where there would have been collision at the receiverof the on-going session between the CTS packet transmitted by this node and the DATA packetsbelonging to the on-going session. Also, if this CTS

transmission is successful, then there mighthave been collisions between DATA packets belonging to the two sessions at both the receivernodes.
The E-NAV used in ICSMA is implemented as two linked lists of blocks, namely, the *SEL*ist andthe *REL*ist. Each block in each linked list has a start time and an end time. A typical list lookslike *s*1, *f*1; *s*2, *f*2; ...; *sk*, *fk*where *Si* denotes the start time of the *ith*block in the list and *fi*denotes thefinish time of the *ith*block in the list. The *SEL*ist is used to determine if the node would be sender-exposed at any given instant of time in the future. A node is predicted to be senderexposedat any time *t* if there is a block *sj, fj* in the *SEL*ist such that *sj*< *t* < *fj*. Similarly,the *REL*ist tells if the node would be receiver-exposed at any time in the future. A node ispredicted to be receiver-exposed at any time *t* if there exists a block *sj*, *fj* in the *REL*ist suchthat *sj*< *t* < *fj*. The *SEL*ist and the *REL*ist are updated whenever the RTS and CTSpackets arereceived by the node. The *SEL*ist is modified when an RTS packet is received, and the *REL*ist is modified when a CTS packet is received by the node. The modification in the list might beadding a new block, modifying an existing block, or merging two or more existing blocks andmodifying the resulting block.
ICSMA is a simple two-channel MAC protocol for ad hoc wireless networks that reduces thenumber of exposed terminals and tries to maximize the number of simultaneoussessions.ICSMA was found to perform better than the 802.11 DCF protocol is terms of metricssuch as throughput and channel access delay.

# Hybrid MAC Protocols

Display characteristics of both categories  attempt to reduce the number of collisions by relying on featuresλ present in contention-free medium access protocols  take advantage of the flexibility and low complexity of contention-basedλ protocols

**Mobility Adaptive Hybrid MAC**

The slots of a frame belong to one of the two categories:  static slotsλ  mobile slotsλ  Each node uses a mobility estimation algorithm to determine its mobility andν which type of slots the node should use  mobility estimation is based on periodic hello messages and receivedλ signal strength hello messages are always transmitted with the same transmit4 power receiving nodes compare consecutive message signal strengths to4 estimate the relative position displacement between themselves and each of their neighbors  a mobility beacon interval is provided at the beginning of a frame toλ distribute mobility information to neighbors
Static slots  use an approach similar to the LMAC approach and have two portions:λ a control section4 – indicate the slot assignment information in a neighborhood – all static nodes must listen to this part of the static slot a data section4 – only the transmitter and receiver stay awake – all other nodes can

turn their radios off  Mobile slotsν  nodes contend for the medium in a two-phase contention periodλ a wakeup tone is sent during the first phase4 the data is sent during the second phase4  priority ordering among mobile nodes to reduce the effective contentionλ

MH-MAC provides a mechanism to dynamically adjust the ratio between static and mobile slots  based on the observed mobilityλ  necessary since the ratio of static versus mobile nodes can varyλ  each node estimates its own mobility and sends this information in theλ previously mentioned beacon time slot at the beginning of a frame  using this mobility information, each node calculates a mobilityλ parameter for the network, which determines the ratio of static and mobile slots

# S-MAC

The S-MAC (Sensor-MAC) protocol  provides mechanisms to circumvent idle listening, collisions, and overhearing. As opposed to STEM, it does not require two different channels. S-MAC adopts a periodic wakeup scheme, that is, each node alternates between a fixed-length listen period and a fixed-length sleep period according to its **schedule**, compare Figure 5.6. However, as opposed to STEM, 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:

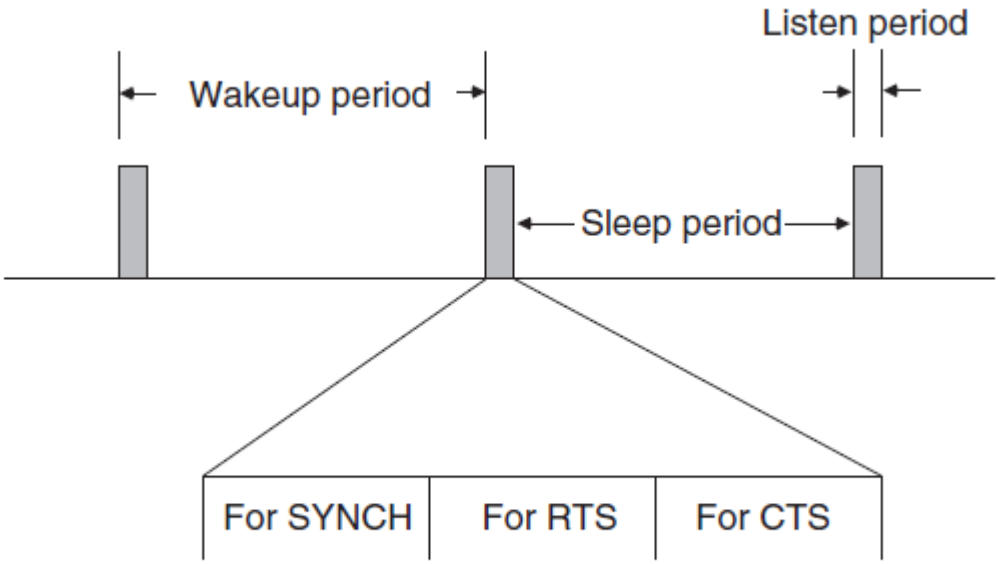



**Figure 5.6** S-MAC principle

\

In the first phase (**SYNCH phase**), node *x* accepts SYNCH packets from its neighbors. Inthese packets, the neighbors describe their own schedule and *x* stores their schedule in a table(the **schedule table**). Node *x*'s SYNCH phase is subdivided into time slots and *x*'s neighborscontend according to a CSMA scheme with additional backoff, that is, each neighbor *y* wishingto 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 modeand waits for *x*'s next wakeup. In the other direction, since *x* knows a neighbor *y*'s schedule, *x*can wake at appropriate times and send its own SYNCH packet to *y*(in broadcast mode). It isnot required that *x* broadcasts its schedule in every of *y*'s wakeup periods. However, for reasonsof time synchronization and to allow new nodes to learn their local network topology, *x* shouldsend SYNCH packets periodically. The according period is called **synchronization period**.

· In the second phase (**RTS phase**), *x* listens for RTS packets from neighboring nodes. In S-MAC,the RTS/CTS handshake described in Section 5.1.2 is used to reduce collisions of data packetsdue to hidden-terminal situations. Again, interested neighbors contend in this phase according to a CSMA scheme with additional backoff.

· In the third phase (**CTS phase**), node *x* transmits a CTS packet if an RTS packet was received in theprevious 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, including thevirtual carrier-sense mechanism, whereby a node maintains a NAV variable. The NAV mechanismcan be readily used to switch off the node during ongoing transmissions to avoid overhearing.When transmitting in a broadcast mode (for example SYNCH packets), the RTS and CTS packetsare dropped and the nodes use CSMA with backoff.

If we can arrange that the schedules of node *x* and its neighbors are synchronized, node *x* andall its neighbors wake up at the same time and *x* can reach all of them with a single SYNCHpacket. The S-MAC protocol allows neighboring nodes to agree on the same schedule and to create**virtual clusters**. The clustering structure refers solely to the exchange of schedules; the transfer ofdata packets is not influenced by virtual clustering.

The S-MAC protocol proceeds as follows to form the virtual clusters: A node *x*, newly switchedon, listens for a time of at least the (globally known) synchronization period. If *x* receives anySYNCH packet from a neighbor, it adopts the announced schedule and broadcasts it in one ofthe neighbors' next listen periods. In the other case, node *x* picks a schedule and broadcasts it.

If *x* receives another node's schedule during the broadcast packet's contention period, it drops itsown schedule and follows the other one. It might also happen

that a node *x* receives a differentschedule after it already has chosen one, for example, because bit errors destroyed previous SYNCHpackets. If node *x* already knows about the existence of neighbors who adopted its own schedule, it keeps its schedule and in the future has to transmit its SYNCH and data packets according toboth schedules. On the other hand, if *x* has no neighbor sharing its schedule, it drops its own andadopts the other one. Since there is always a chance to receive SYNCH packets in error node *x*periodically listens for a whole synchronization period to relearn its neighborhood. This makes the virtual cluster formation fairly robust.

By this approach, a large multihop network is partitioned into "islands of schedule synchronization".Border nodes have to follow two or more different schedules for broadcasting their SYNCHpackets and for forwarding data packets. Thus, they expend more energy than nodes only havingneighbors of the same "schedule regime".

The periodic wakeup scheme adopted by S-MAC allows nodes to spend much time in the sleepmode, but there is also a price to pay in terms of latency. Without further modifications, the per-hoplatency of S-MAC will be approximately equal to the sleep period on average when all nodes follow the same schedule. Ye et al. describe the **adaptive-listening** scheme, which roughly halvesthe per-hop latency. Consider the following situation: Node *x*receives during its listen period an RTS or CTS packet belonging to a packet exchange from neighbor node *y* to node *z*. From theduration field of these packets, *x* can infer the time *t*0 when the packet exchange ends. Since it might happen that *x* is the next hop for *z*'s packet, node *x* schedules an extra listen period around time *t*0 and *z* tries to send an extra RTS at time *t*0, ignoring *x*'s normal wakeup cycle. Under ideal circumstances, *x* is awake when *z* sends the RTS and the packet can take the next hop quickly.

S-MAC also adopts a message-passing approach (illustrated in Figure 5.7), where a message is a larger data item meaningful to the application. In-network processing usually requires the aggregating node to receive a message completely. On the other hand, on wireless media, it is advisable to break a longer packet into several shorter ones (fragmentation, see also Chapter 6). 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 (data, ack, RTS,CTS) have a duration field and a neighboring node *C* is required to set its NAV field accordingly.

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

plus ack packet, and the medium is reserved for this prolonged time. However, thereis the problem of how a nonparticipating node shall learn about the elongation of the transaction when he has only heard the initial RTS or CTS packets.

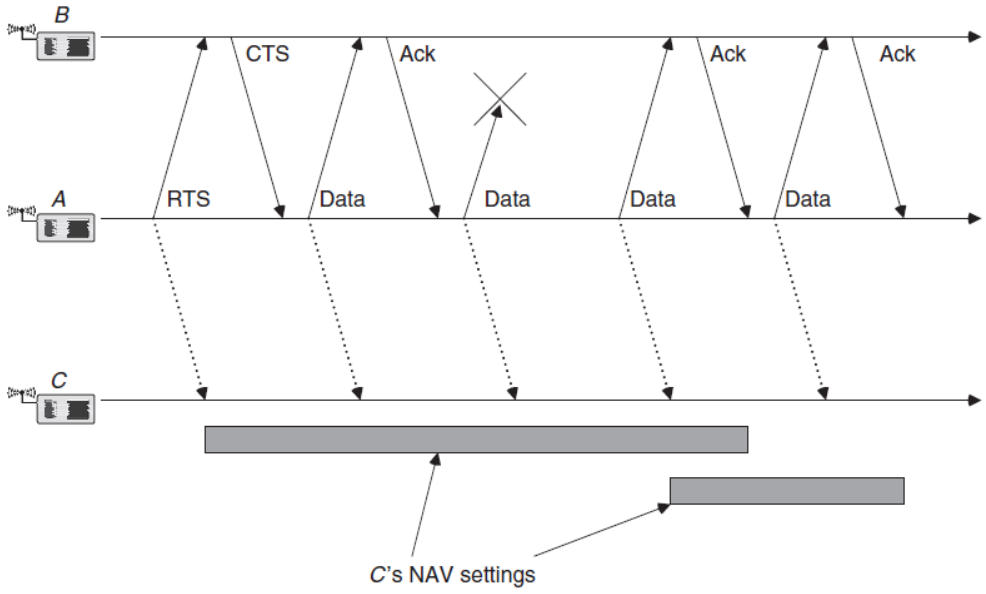



**Figure 5.7** S-MAC fragmentation and NAV setting

This scheme has some similarities to the fragmentation scheme used in IEEE 802.11 but there are important differences. In IEEE 802.11, the RTS and CTS frame reserve the medium only for the time of the first fragment, and any fragment reserves only for the next fragment. If one packet needs to be retransmitted, the initiating node has to give up the channel and recontend for it in the same way as for a new packet. The approach taken by S-MAC reduces the latency of complete messages by suppressing intertwined transmissions of other packets. Therefore, in a sense, thisprotocol is unfair because single nodes can block the medium for long time. However, as explained in Section 5.1.3, the fairness requirement has a different weight in a wireless sensor network than it has in a data network where users want to have fair medium access.

S-MAC has one major drawback: it is hard to adapt the length of the wakeup period to changing load situations, since this length is essentially fixed, as is the length of the listen period.
The T-MAC protocol presented by van Dam and Langendoenis similar to S-MAC but adaptively shortens the listen period. If a node *x* senses no activity on the medium for a specified duration, it is allowed to go back into sleep mode

prematurely. Therefore, if no node wants totransmit to *x*, the listen period can be ended quickly, whereas in S-MAC, the listen period has afixed length.

# SCHEDULE BASED PROTOCOLS-

A second fundamental advantage of schedule-based protocols is that transmission schedules can be computed such that nocollisions occur at receivers and hence no special mechanisms are needed to avoid hidden-terminal situations. However, these schemes also have downsides. First, the setup and maintenance of schedulesinvolves signaling traffic, especially when faced to variable topologies. Second, if a TDMA variant is employed, time is divided into comparably small slots, and both transmitter and receiver have to agree to slot boundaries to actually meet each other and to avoid overlaps with other slots, which would lead to collisions. However, maintaining time synchronization involves some extra signaling traffic. For cheap sensor nodes with cheap oscillators, one can expect the clocks of different nodes to drift comparably quickly and resynchronization is required frequently
.

A third drawback is that such schedules are not easily adapted to different load situations on smalltimescales. Specifically, in TDMA, it is difficult for a node to give up unused time slots to its neighbors. A further disadvantage is that the schedule of a node ( possibly those of its neighbors) may require a significant amount of memory, which is a scarce resource in several sensor node designs

# LEACH

The LEACH protocol (Low-energy Adaptive Clustering Hierarchy) presented by Heinzelmanet al. assumes a dense sensor network of homogeneous, energy-constrained nodes, whichshall report their data to a sink node. In LEACH, a TDMA-based MAC protocol is integrated withclustering and a simple “routing” protocol.

LEACH partitions the nodes into **clusters** and in each cluster a dedicated node, the **clusterhead**, is responsible for creating and maintaining a TDMA schedule; all the other nodes of a cluster are**member nodes**. To all member nodes, TDMA slots are assigned, which can be used to exchange data between the member and the clusterhead; there is no peer-to-peer communication. With the exception of their time slots, the members can spend their time in sleep state. The cluster head aggregates the data of its members and transmits it to the sink node or to other nodes for further relaying. Since the sink is often far away, the clusterhead must spend significant energy for this transmission. For a member, it is typically much cheaper to reach the clusterhead than to transmitdirectly to the sink. The clusterheads role is energy consuming since it is always switched on and is responsible for the long-range transmissions. If a fixed node has this

role, it would burnits energy quickly, and after it died, all its members would be "headless" and therefore useless.
Therefore, this burden is rotated among the nodes. Specifically, each node decides independent of other nodes whether it becomes a clusterhead, and therefore there is no signaling traffic related to clusterhead election (although signaling traffic is needed for subsequent association of nodes to some clusterhead). This decision takes into account when the node served as cluster head the last time, such that a node that has not been a clusterhead for a long time is more likely to elect itself than a node serving just recently. The protocol is round based, that is, all nodes make their decisions whether to become a clusterhead at the same time and the noncluster head nodes have to associate to a clusterhead subsequently. The nonclusterheads choose their cluster head based on received signal strengths. The network partitioning into clusters is time variable and the protocol assumes global time synchronization.
After the clusters have been formed, each clusterhead picks a random CDMA code for its cluster, which it broadcasts and which its member nodes have to use subsequently. This avoids a situation where a border node belonging to clusterhead *A* distorts transmissions directed to clusterhead *B*,
shown in Figure 5.10.

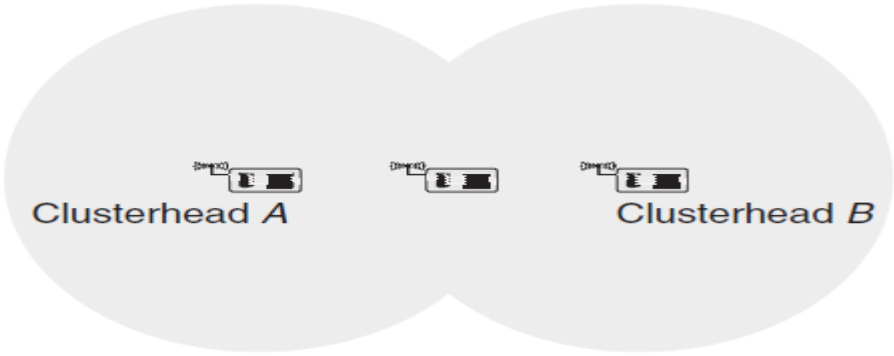



**Figure 5.10** Intercluster interference

A critical network parameter is the percentage of nodes that are cluster heads. If there are only a few clusterheads, the expected distance between a member node and its cluster head becomes longer and therefore the member has to spend more energy to reach its cluster head while maintaining a given BER target. On the other hand, if there are many clusterheads, there will be more energy expensive transmissions from clusterheads to the sink and less aggregation.

Therefore, there exists an optimum percentage of clusterheads, which for the scenario investigated in is≈5%. Ifthis optimum is chosen, LEACH can achieve a seven to eight times lower overall energy dissipationcompared to the case where each node transmits its data directly to the sink, and between four and

eight times lower energy than in a scenario where packets are relayed in a multi hop fashion. In addition, since LEACH distributes the clusterhead role fairly to all nodes, they tend to die at about the same time.

The protocol is organized in **rounds** and each round is subdivided into a setup phase and asteady-state phase (Figure 5.11). The **setup phase** starts with the self-election of nodes to cluster heads. In the following **advertisement phase**, the clusterheads inform their neighborhood with an advertisement packet. The clusterheads contend for the medium using a CSMA protocol with no further provision against the hidden-terminal problem. The nonclusterhead nodes pick the advertisement packet with the strongest received signal strength. In the following cluster-setup phase, the members inform their clusterhead ("join"), again using a CSMA protocol. After the cluster setup-phase, the clusterhead knows the number of members and their identifiers.

It constructs a TDMA schedule, picks a CDMA code randomly, and broadcasts this information in the broadcast schedule subphase. After this, the TDMA steady-state phase begins.

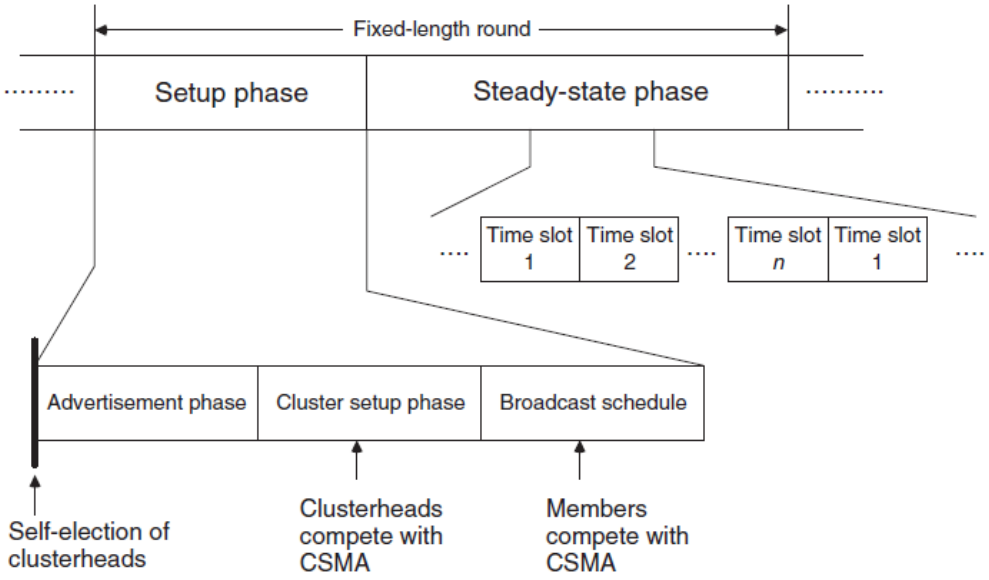



**Figure 5.11** Organization of LEACH rounds

Because of collisions of advertisement or join packets, the protocol cannot guarantee that each nonclusterhead node belongs to a cluster. However, it can guarantee that nodes belong to at most one cluster.The clusterhead is switched on during the whole round and the member nodes have to be switchedon during the setup phase and occasionally in the steady-state phase, according to their position in the cluster's TDMA schedule.With the protocol described so far, LEACH would not be able to cover large geographical areasof some square miles or more, because a clusterhead two miles away from the sink likely does not have enough energy to reach the sink at all, not to mention achieving a low

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

# SMACS

The Self-Organizing Medium Access Control for Sensor Networks (SMACS) protocol described bySohrabiet al., Sohrabiand Pottie is part of a wireless sensor network protocol suite that addresses MAC, neighbor discovery, attachment of mobile nodes, a multihop routing protocol, and a local routing protocol for cooperative signal processing purposes.
SMACS essentially combines neighborhood discovery and assignment of TDMA schedules to nodes. SMACS is based on the following assumptions:
· The available spectrum is subdivided into many channels and each node can tune its transceiver to an arbitrary one; alternatively, it is assumed that many CDMA codes are available.
· Most of the nodes in the sensor network are stationary and such an assignment is valid for fairly long times.Each node divides its time locally into fixed-length superframes (of duration $T_{frame}$ seconds),which do not necessarily have the same phase as the neighbor's superframes. However, all nodes have the same superframe length and this requires time synchronization. Superframes are also subdivided into time slots but this is only loose since transmissions are not confined to occur*only within* a single time slot.
The goal of SMACS is to detect neighboring nodes and to set up exclusive **links** or **channels**to these. A link is directional, that is on a given link all packets are transmitted in one direction.Furthermore, a link occupies a TDMA slot in either endpoint. When two nodes want bidirectionaloperation, two such links are needed; from the perspective of one node, there is a **receive slot** anda **transmit slot** to the other node. The assignment of links shall be such that no collisions occur atreceivers.
To achieve this, SMACS takes care that for a single node the time slots of different linksdo not overlap (using a simple greedy algorithm) and furthermore for each link randomly one outof a large number of frequency channels/CDMA codes is picked and allocated to the link. It is notrequired that a node and its neighbors transmit at entirely different times. In this case, however,they must transmit to different receivers and have to use different frequencies/codes. After linksetup, the nodes wake up periodically (once per superframe) in the respective receive time slotswith the receiver tuned to the corresponding frequency or with the correct CDMA code at hand;the transmit time slots are only used when required.

By using a local scheme instead of a global assignment, the task of transmitting the neighborhood information to a central node and the computation results back is avoided..

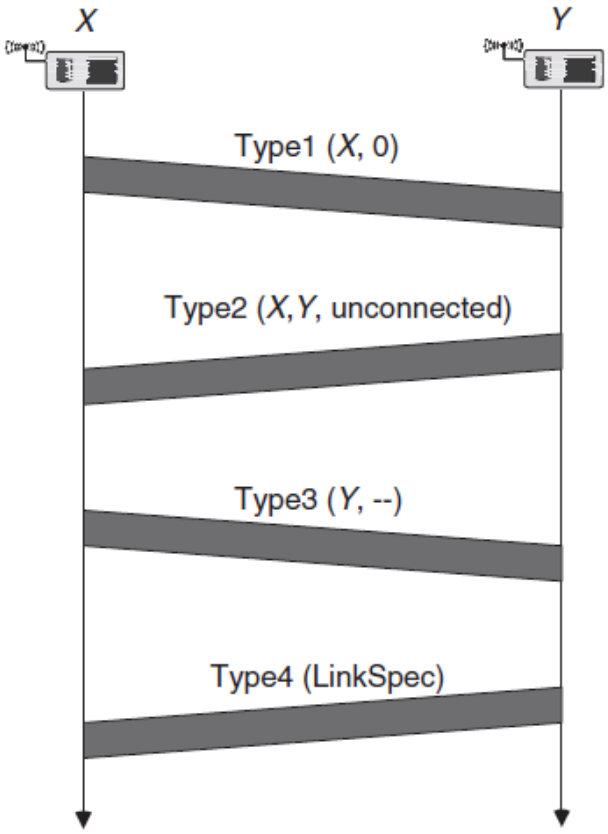



**Figure 5.12** SMACS: link setup for two lonesome nodes

Regarding the neighbor discovery and link setup, we consider four different cases. Suppose that nodes *x* and *y* want to set up a link and *x* is switched on first. First, we assume that neither *x* nor*y* has any neighbor so far, as illustrated by Figure 5.12. Node *x* listens on a fixed frequency band for a random amount of time. If nothing is received during this time, node *x* sends an invitation message, more specifically a `TYPE1`(*x*, unattached) message, indicating its own node identification and the number of attached neighbors, which so far is zero. When any neighbor *z* of node *x* receives this message, it waits for a random but bounded amount of time and answers with a `TYPE2`(*x*, *z*,*n*) message, indicating its own address, *x*'s address and its number of neighbors *n*.

Now, suppose that the so-far unconnected node *y* answers first – with `TYPE2`(*x*, *y*, unattached) – and *x* receives this message properly. Since *y* sent the first answer, *x* invites *y* to construct a link by sending a`TYPE3`(*y*, –) message, carrying the identification of the "winning" node *y* and no further parameters. This message is sent when the contention period for the `TYPE2`(・, ・, ・) answer message ends. Now, node *y* knows that (i) it has been selected, and (ii) it

can pick any time slot it wants since neither *x* nor *y* has any link allocated so far. Node *y* answers to node *x* with a link specification, that is,two time slot specifications and a frequency/code, using a TYPE2(*x*, *y*, Link Spec) message. The time slot specifications have a common time base since node *y* adopts *x*'s super frame phase upon receiving the TYPE3(*y*, –) message. By this rule, neighboring nodes that discover each other first, have a common phase (and a common period).Any other node *z* loosing against *y* goes back into sleep mode and tries again at some later time. The nodes repeat their invitations periodically using TYPE1(・, ・) messages.

The second case is where node *x* already has some neighbors but the winning node *y* has noneso far. Therefore, *x* sends a TYPE1(*x*, attached) message and *y* manages to answer first with itsTYPE2(*x*, *y*, unattached) message. After this, node *x* knows that it can schedule the connection to*y* freely, since *y* has no obligations so far. Node *x* picks two convenient time slots and a frequency and sends a TYPE3(*y*, LinkSpec) message to *y*. Again, since *y* has no neighbors so far, *y* adopts the superframe phase of *x*. Finally, node *y* answers with TYPE2(*x*, *y*, –) message, carrying an empty link specification (meaning that *x*'s link specification is adopted).

In the third case, node *x* does not have any neighbor yet, but *y* has. Therefore, *y* answers to*x*'s TYPE1(*x*, unattached) with a TYPE2(*x*, *y*, attached) message. Node *x* proceeds with sending a TYPE3(*y*, –) message without link specification to *y*, and it is *y*'s turn to pick the time slots and frequency. Accordingly, *y* sends back a TYPE2(*x*, *y*, LinkSpec) to *x*.

In the final case, both *x* and *y* are already attached to other nodes and their super frames are typically not aligned. Accordingly, *x* sends a TYPE1(*x*, attached) message and *y* answers with a TYPE2(*x*, *y*, attached) message. Node *x* answers with a TYPE3(*y*, Schedule) message, which contains its entire schedule as well as timing information allowing *y* to determine the phase shift between *x* and *y*'s superframes. After receiving this information, node *y* determines time slots that are free in both schedules, and which are not necessarily aligned with any time slot boundaries in either schedule.

This protocol allows to set up static connections between stationary nodes. Since the neighborhood discovery process is repeated from time to time, the protocol can adapt to changes in topology. In reference, an extension ("eavesdrop and register" algorithm) is described that allows a mobile node to set up, maintain, and tear down connections to stationary nodes as it moves through the network. A critical issue with this protocol is the choice of the superframe length. It should be large enough to accommodate the highest node degree in the network, which is a random variable for random deployments. If the superframe length is too short, some of a node's neighbors may simply not

be visible to it. A second problem occurs in a densely populated sensor network with low traffic load, where schedules are highly populated and nodes wake up quite often just to notice that there is nopacket destined to them. The number of wakeup slots depends on the node density for this protocol.

## Traffic-adaptive medium access protocol (TRAMA)

The Traffic-Adaptive Medium Access (TRAMA) protocol presented by Rajendran et al. creates schedules allowing nodes to access a single channel in a collision-free manner. The schedulesare constructed in a distributed manner and on an on-demand basis. The protocol assumes that all nodes are time synchronized and divides time into **random access periods** and **scheduled-accessperiods**. A random access period followed by a scheduled-access period is called a **cycle**. The nodes broadcast their neighborhood information and, by capturing the respective packets from their neighbors, can learn about their two-hop neighborhood. Furthermore, they broadcast their schedule information, that is, they periodically provide their neighbors with an updated list of receivers for the packets currently in a nodes queue. On the basis of this information, the nodes execute a distributed scheduling algorithm to determine for each time slot of the scheduled-access period the transmitting and receiving nodes and the nodes that can go into sleep mode.

The protocol itself consists of three different components: the **neighborhood protocol**, the**schedule exchange protocol** and the **adaptive election algorithm**. The **neighborhood protocol**is executed solely in the random access phase, which is subdivided into small time slots. A node picks randomly a number of time slots and transmits small control packets in these without doing any carrier sensing. These packets indicate the node's identification and contain incremental neighborhood information, that is only those neighbor identifications are included that belong to new neighbors or neighbors that were missing during the last cycle. When a node does not transmit, it listens to pick up its neighbors' control packets. The length of the random access phase should be chosen such that a node receives its neighbors packets with sufficiently high probability to ensure consistent topology information. It depends thus on the node degree. All nodes' transceivers must be active during the random access period.

By the **schedule exchange protocol**, a node transmits its current transmission schedule (indicating in which time slots it transmits to which neighbor) and also picks up its neighbors' schedules. This information is used to actually allocate slots to transmitters and receivers. How does a node know which slots it can use? All nodes possess a global hash function $h$, and a node with identification $x$ computes for time slot occurring at $t$ the following priority value $p$:

$$p = h(x \oplus t)$$

where $x \oplus t$ is the concatenation of $x$'s node identification with the current time $t$ . To compute itsschedule, a node looks ahead for a certain number of time slots, called its **schedule interval** (say: 100 slots) and for each of these slots computes its own priority and the priority of all its two-hop neighbors. For higher node densities, this incurs significant computation costs. The slots for which$x$ has the highest priority value can be used by $x$ to transmit its packets. These are called *winningslots*; for the sake of example, let us say these are slots 17, 34, 90, and 94. By looking at its packet queue, $x$ can determine whether it needs all of these slots or can leave some of them to other nodes.

Node $x$ assigns to each of its winning slots a receiving node or a set of receivers, and sends this assignment as its schedule packet. The last of the future winning slots (slot 94 in our example) is always used for broadcasting $x$'s next schedule, that is the whole schedule computation has to be repeated immediately before slot 94, spanning again over a full schedule interval. By using the last winning slot, the schedule can be transmitted without risk of collision.

The neighbors of $x$ should wake up at slot 94 to receive $x$'s next schedule (they should also have woken up to receive the current schedule!) and to determine when they have to leave sleep mode to receive a packet from $x$. In turn, $x$ should also wake up when its neighbors have announced to transmit their next schedule.

With what we have described so far, node $x$ can determine its winning slots and thus its transmit opportunities. The other question is: when must $x$ prepare for receptions and when can $x$ go into sleep mode during the scheduled-access phase? Fix one specific slot. There are two easy cases:

Suppose that a one-hop neighbor $y$ of $x$ has the highest priority in $x$'s two-hop neighborhood and that $y$ has announced a packet for this slot. Either $x$ is the receiver or $x$ can go to sleep. A more complicated situation is depicted in Figure 5.13. Here, node $D$ has the highest priority in $B$'s two hop neighborhood, but, on the other hand node, $A$ has highest priority in *its* two-hop neighborhood.

The **adaptive election algorithm** of TRAMA provides approaches for resolving this situation and also for allowing nodes to reuse their neighbors' unused winning slots.

Rajendran et al. compare the performance of TRAMA with S-MAC (having 10 and 50% duty cycle, respectively), the IEEE 802.11 protocol, a CSMA protocol according to reference and NAMA , a precursor of TRAMA. The investigated performance measures areaverage packet delivery ratio, the achievable percentage of sleep time, the average sleep interval (which should be long since switching on and off transceivers costs energy), and the averagequeuing delay of a packet waiting for transmission. Two scenarios are simulated: a single-hop

scenario and a multihop scenario with one sink node and the sensors transmitting periodically tothe sink. The underlying physical layer resembles the RF Monolithics TR1000 transceiver . As opposed to S-MAC, the energy savings of TRAMA depend on theload situation, while in S-MAC it depends on the duty cycle. The investigations confirmed also a well-known property of TDMA protocols stating that these have higher delays but also higher maximum throughput than contention-based protocols.
The TRAMA protocol needs significant computation and memory in dense sensor networks since the two-hop neighborhood of a node tends to be large in this case. Therefore, TRAMA is a feasible solution only if the sensor nodes have sufficient resources.

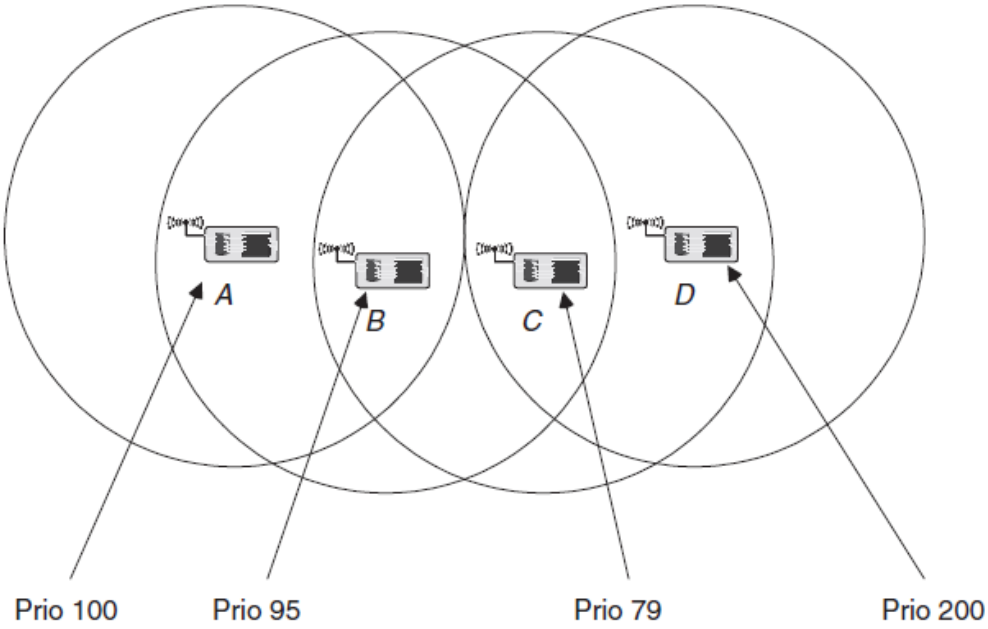



**Figure 5.13** TRAMA: conflict situation

# IEEE 802.15.4

IEEE 802.15.4 created for low-power devices in the 868 MHz, 915 MHz, and 2.45 GHzλ frequency bands supports two transmission modes:λ UWB PHY4 –

bit rates: 110 kbps, 851 kbps (nominal), 6.81 kbps, and 27.24 Mbps CSS PHY4– bit rates: 1 Mbps (nominal) and 250 kbps

The Institute of Electrical and Electronics Engineers (IEEE) finalized the IEEE 802.15.4 standard in October 2003 . The standard covers the physical layer and the MAC layer of a low-rate Wireless Personal Area Network (WPAN). Sometimes, people confuse IEEE 802.15.4 with ZigBee5, an emerging standard from the ZigBee alliance. ZigBee uses the services offered by IEEE 802.15.4 and adds network construction (star networks, peer-to-peer/ mesh networks, cluster-tree networks), security, application services, and more.
The targeted applications for IEEE 802.15.4 are in the area of wireless sensor networks, home automation, home networking, connecting devices to a PC, home security, and so on. Most of these applications require only low-to-medium bitrates (up to some few hundreds of kbps), moderate average delays without too stringent delay guarantees, and for certain nodes it is highly desirable to reduce the energy consumption to a minimum. The physical layer offers bitrates of 20 kbps (a single channel in the frequency range 868–868.6 MHz), 40 kbps (ten channels in the range between 905 and 928 MHz) and 250 kbps (16 channels in the 2.4 GHz ISM band between 2.4 and 2.485 GHz with 5-MHz spacing between the center frequencies). There are a total of 27 channels available, but the MAC protocol uses only one of these channels at a time; it is not a multichannel protocol. The MAC protocol combines both schedule-based as well as contention-based schemes. The protocol is asymmetric in that different types of nodes with different roles are used.

### 5.5.1 Network architecture and types/roles of nodes

The standard distinguishes on the MAC layer two types of nodes:
· A Full Function Device (FFD) can operate in three different roles: it can be a **PAN coordinator**
(PAN = Personal Area Network), a simple **coordinator** or a **device**.
· A Reduced Function Device (RFD) can operate only as a device.
A device must be associated to a coordinator node (which must be a FFD) and communicates only
with this, this way 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 among others the following tasks:

- It manages a list of associated devices.
- Devices are required to explicitly associate and disassociate with a coordinator using certain signaling packets.

- It allocates short addresses to its devices. All IEEE 802.15.4 nodes have a 64-bit device address.
- When a device associates with a coordinator, it may request assignment of a 16-bit short address to be used subsequently in all communications between device and coordinator. The assigned address is indicated in the association response packet issued by the coordinator.
- In the beaconed mode of IEEE 802.15.4, it transmits regularly **frame beacon** packets announcing the PAN identifier, a list of outstanding frames, and other parameters. Furthermore, the coordinator can accept and process requests to reserve fixed time slots to nodes and the allocations are indicated in the beacon.
- It exchanges data packets with devices and with peer coordinators.

In the remainder of this section, we focus on the data exchange between coordinator and devices
in a star network; a possible protocol for data exchange between coordinators is described in

- Section 5.2.3. We start with the beaconed mode of IEEE 802.15.4.

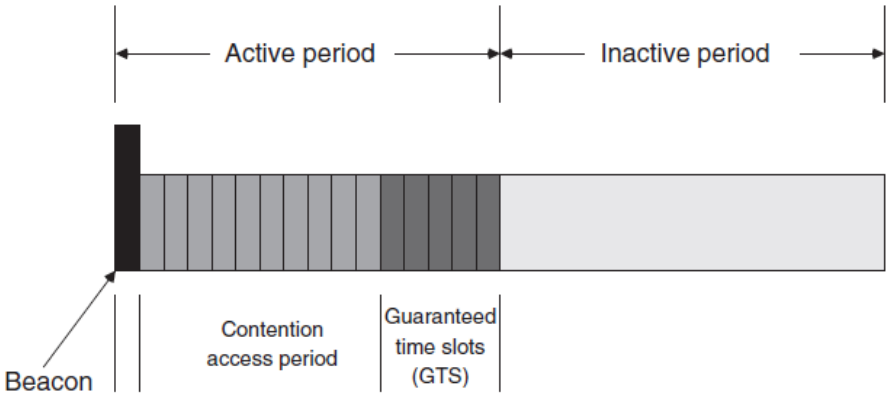



**Figure 5.14** Superframe structure of IEEE 802.15.4

- 
- **5.5.2 Superframe structure**
- The coordinator of a star network operating in the **beaconed mode** organizes channel access and data transmission with the help of a superframe structure displayed in Figure 5.14.
- All superframes have the same length. The coordinator starts each superframe by sending a frame beacon packet.
- The frame beacon includes a **superframe specification** describing the length of the various components of the following superframe:
- The superframe is subdivided into an **active period** and an **inactive period**. During the inactiveperiod, all nodes including the coordinator can switch off their transceivers and go into sleepstate. The nodes have to wake up immediately before the inactive period ends to receive the nextbeacon. The inactive period may be void.

- The active period is subdivided into 16 time slots. The first time slot is occupied by the beaconframe and the remaining time slots are partitioned into a **Contention Access Period (CAP**followed by a number (maximal seven) of contiguous **Guaranteed Time Slots (GTSs)**.

The lengthof the active and inactive period as well as the length of a single time slot and the usageof GTS slots are configurable.The coordinator is active during the entire active period. The associated devices are active in the GTS phase only in time slots allocated to them; in all other GTS slots they can enter sleep mode.In the CAP, a device can shut down its transceiver if it has neither any own data to transmit norany data to fetch from the coordinator.
It can be noted already from this description that coordinators do much more work than devicesand the protocol is inherently asymmetric. The protocol is optimized for cases where energyconstrainedSensors are to be attached to energy-unconstrained nodes.

**5.5.3 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 the requested time slot is a **transmitslot** 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. Another field in the request specifies the desired number of contiguous time slots in the GTS phase. The coordinator answers the request packet in two steps: An immediate acknowledgment packet confirms that the coordinator has received the request packet properly but contains no information about success or failure of the request.
After receiving the acknowledgment packet, the device is required to track the coordinator's beacons for some specified time (called*aGTSDescPersistenceTime*). When the coordinator has sufficient resources to allocate a GTS to the node, it inserts an appropriate **GTS descriptor** into one of the next beacon frames. This GTS descriptor specifies the short address of the requesting node and the number and position of the time slots within the GTS phase of the super frame. A device can use its allocated slots each time they are announced by the coordinator in the GTS descriptor. If the coordinator has insufficient resources, it generates a GTS descriptor for (invalid) time slot zero, indicating the available resources in the descriptors length field. Upon receiving such a descriptor, the device may consider renegotiation. If the device receives no GTS descriptor within *aGTSDescPersistenceTime* time after sending the request, it concludes that the allocation request has failed.
A GTS is allocated to a device on a regular basis until it is explicitly deallocated. The deallocationcan be requested by the device by means of a special control frame. After sending this frame, the device shall not use the allocated slots any further. The coordinator can also trigger deallocation

based on certain criteria. Specifically, the coordinator monitors the usage of the time slot: If the slot is not used at least once within a certain number of superframes, the slot is de allocated. The coordinator signals deallocation to the device by generating a GTS descriptor with start slot zero.

### 5.5.4 Data transfer procedures

Let us first assume that a device wants to transmit a data packet to the coordinator. If the devicehas an allocated transmit GTS, it wakes up just before the time slot starts and sends its packetimmediately without running any carrier-sense or other collision-avoiding operations. However, thedevice can do so only when the full transaction consisting of the data packet and an immediate acknowledgment sent by the coordinator as well as appropriate InterFrame Spaces (IFSs) fit intothe allocated time slots. If this is not the case or when the device does not have any allocatedslots, it sends its data packet during the CAP using a slotted CSMA protocol, described below. Thecoordinator sends an immediate acknowledgment for the data packet.

The other case is a data transfer from the coordinator to a device. If the device has allocated areceive GTS and when the packet/acknowledgment/IFS cycle fits into these, the coordinator simplytransmits the packet in the allocated time slot without further coordination. The device has toacknowledge the data packet.

The more interesting case is when the coordinator is not able to use a receive GTS. The handshakebetween device and coordinator is sketched in Figure 5.15. The coordinator announces a bufferedpacket to a device by including the devices address into the **pending address field** of the beaconframe. In fact, the device's address is included as long as the device has not retrieved the packet or a certain timer has expired. When the device finds its address in the pending address field, itsends a special **data request** packet during the CAP. The coordinator answers this packet withan acknowledgment packet and continues with sending the data packet. The device knows uponreceiving the acknowledgment packet that it shall leave its transceiver on and prepares for the incoming data packet, which in turn is acknowledged. Otherwise, the device tries again to sendthe data request packet during one of the following superframes and optionally switches off itstransceiver until the next beacon.

### 5.5.5 Slotted CSMA-CA protocol

When nodes have to send data or management/control packets during the CAP, they use a slottedCSMA protocol. The protocol contains no provisions against hidden-terminal situations, for examplethere is no RTS/CTS handshake. To reduce the probability of collisions, the protocol uses random delays; it is thus a CSMA-CA protocol (CSMA with Collision Avoidance). Using such random delays is also part of the protocols described in Section 5.3.1. We describe the protocol operation in some more detail; please refer to Figure 5.16 also. The time slots making up the CAP are subdivided into smaller time slots, called backoff periods. One backoff period has a length corresponding to 20 channel symbol times and the slots considered by the slotted CSMA-CA protocol are

just these backoff periods. The device maintains three variables NB, CW, and BE.

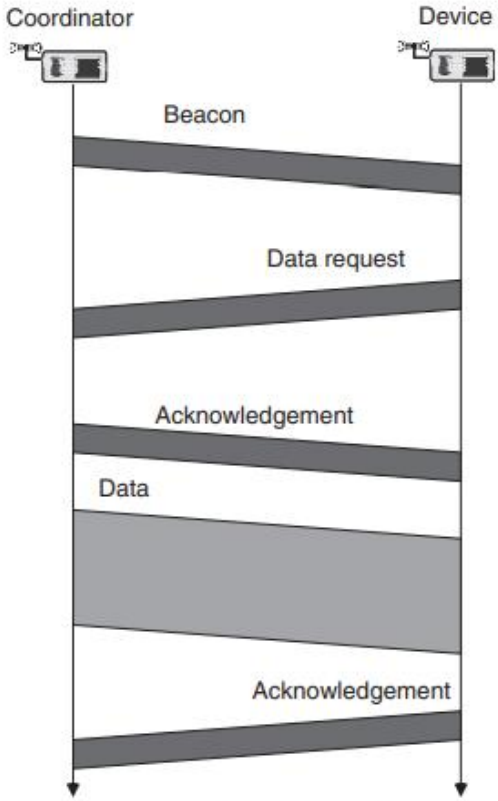



**Figure 5.15** Handshake between coordinator and device when the device retrieves a packet [468, Fig. 8]

. The variable NB counts the number of backoffs, CW indicates the size of the current congestion window, and BE is the current backoff exponent. Upon arrival of a new packet to transmit, these variables are initialized with NB = 0, CW = 2, and BE = macMinBE (with macMinBE being a protocol parameter), respectively. The device awaits the next backoff period boundary and draws an integer random number r from the interval [0, 2BE − 1]. The device waits for r backoff periods and performs a carrier-sense operation (denoted as Clear Channel Assessment (CCA) in the standard). If the medium is idle, the device decrements CW, waits for the next backoff period boundary, and senses the channel again. If the channel is still idle, the device assumes that it has won contention and starts transmission of its data packet. If either of the CCA operations shows a busy medium, the number of backoffs NB and the backoff exponent BE are incremented and CW is set back to CW = 2. If NB exceeds a threshold, the device drops the frame and declares a failure. Otherwise, the device again draws an integer r from [0, 2BE − 1] and waits for the indicated number of backoff slots. All subsequent steps are repeated

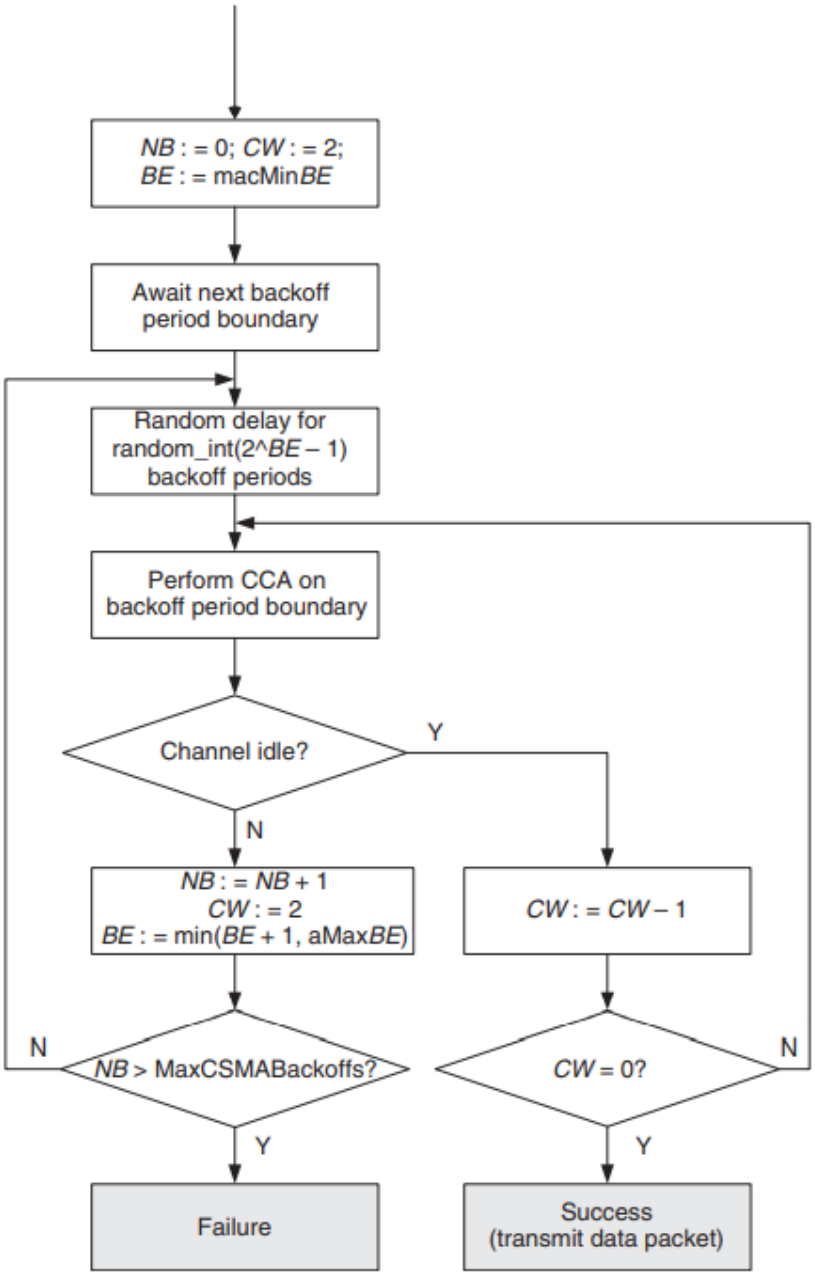



**Figure 5.16** Schematic of the slotted CSMA-CA algorithm (simplified version of [468, Fig. 61])

5.5.6 Nonbeaconed mode The IEEE 802.15.4 protocol offers a nonbeaconed mode besides the beaconed mode. Some important differences between these modes are the following: • In the nonbeaconed mode, the coordinator does not send beacon frames nor is there any GTS mechanism. The lack of beacon packets takes away a good opportunity for devices to acquire time synchronization with the coordinator. • All packets from devices are transmitted using an unslotted (because of the lack of time synchronization) CSMA-CA protocol. As opposed to the slotted CSMA-CA protocol, there is no synchronization to backoff period boundaries and, in addition, the device performs only a single CCA operation. If this indicates an idle

channel, the device infers success. • Coordinators must be switched on constantly but devices can follow their own sleep schedule. Devices wake up for two reasons: (i) to send a data/control packet to the coordinators, or (ii) to fetch a packet destined to itself from the coordinator by using the data request/acknowledgment/ data/acknowledgment handshake (fetch cycle) discussed above. The data request packet is sent through the unslotted CSMA-CA mechanism and the following acknowledgment is sent without any further ado. When the coordinator has a data packet for the device, it transmits it using the unslotted CSMA-CA access method and the device sends an immediate acknowledgment for the data. Therefore, the device must stay awake for a certain time after sending the data request packet. The rate by which the device initiates the fetch cycle is application dependent

# UNIT 4
# Network Layer: Routing Metrics

Objectives

In a multihop network, intermediate nodes have to relay packets from the source to the destination node. Such an intermediate node has to decide to which neighbor to forward an incoming packet not destined for itself. Typically, routing tables that list the most appropriate neighbor for any given packet destination are used. The construction and maintenance of these routing tables is the crucial task of a distributed routing protocol.

4.1 The many faces of forwarding and routing Whenever a source node cannot send its packets directly to its destination node but has to rely on the assistance of intermediate nodes to forward these packets on its behalf, a multihop networkresults – an example is shown in Figure 11.1.

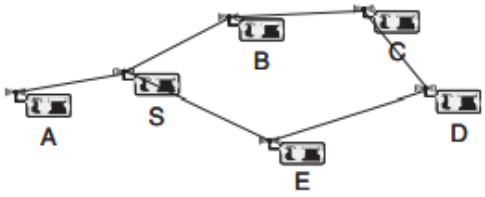



**Figure 11.1** A simple example of routing in a multihop network – node S sends packets to node D

In such a network, an intermediate node (as well as the source node) has to decide to which neighboring node an incoming packet should be passed on so that it eventually reaches the destination – for example, node S sending to node A would not do. This act of passing on is called forwarding, and several different options exist how to organize this forwarding process. The simplest forwarding rule is to flood the network: Send an incoming packet to all neighbors. As long as source and destination node are in the same connected component of the network, the packet is sure to arrive at the destination. To avoid packets circulating endlessly, a node should only forward packets it has not yet seen (necessitating, for example, unique source identifier and sequence numbers in the packet). Also, packets usually carry some form of expiration date (time to live, maximum number of hops) to avoid needless propagation of the packet (e.g. if the destination node is not reachable at all). An alternative to forwarding the packet to all neighbors is to forward it to an arbitrary one. Such gossiping results in the packet randomly traversing the network in the hope of eventually finding the destination node. Clearly, the packet delay can be substantially larger.

Flooding and gossiping are two extreme ends of a design spectrum; alternatively, the source could send out more than a single packet on a random walk or each node could forward an incoming packet to a subset of its neighbors – for example, as determined by a topology-control algorithm, equivalent to flooding on a reduced topology. This last option is sometimes called controlled flooding. While these forwarding rules are simple, their performance in terms of number of sent packets or delay, . for example, is likely poor. These shortcomings are due to ignoring the network's topology. In the example of Figure 11.1, without knowing that node A is even further away from the destination node D, the source node S has no means of avoiding it when forwarding its own packet. Hence, some information about the suitability of a neighbor in the forwarding process would be required. A neighbor's suitability is captured by the cost it incurs to send a packet to its destination via this particular neighbor. These costs can be measured in various metrics, for example, the minimal number of hops or the minimal total energy it requires to reach the destination via the given neighbor. Each node collects these costs in routing tables; Table 11.1 shows two examples

**Table 11.1** Routing tables for some nodes from Figure 11.1, using hop count as cost metric

| Destination | Next-hop neighbor | Cost |
|---|---|---|
| A | A | 1 |
| D | A | 3 |
| D | B | 3 |
| D | E | 2 |
| E | E | 2 |

(a) Node S's routing table

| Destination | Next-hop neighbor | Cost |
|---|---|---|
| A | S | 2 |
| D | C | 2 |
| D | S | 3 |
| E | A | 2 |
| E | C | 3 |

(b) Node B's routing table

Determining these routing tables is the task of the routing algorithm with the help of the routing protocol. In wired networks, these protocols are usually based on link state or distance vector algorithms (Dijkstra's or Bellman–Ford). In a wireless, possibly mobile, multihop network, different approaches are required. Routing protocols here should be distributed, have low overhead, be self-configuring, and be able to cope with frequently changing network topologies. This question of ad hoc routing has received a considerable amount of attention in the research literature and a large number of ad hoc routing protocols have been developed. A commonly used taxonomy [707] classifies these protocols as either (i) table-driven or proactive protocols, which are "conservative" protocols in that they do try to keep accurate information in their routing tables, or (ii) on-demand protocols, which do not attempt to maintain routing tables at all times but only construct them when a packet is to be sent to

a destination for which no routing information is available. As usual, the borders are not sharp between these classes and there are some ideas for hybrid solutions. Examples for table-driven protocols are Destination-Sequenced Distance Vector (DSDV) [633], Clusterhead Gateway Switch Routing (CGSR) [149], and Wireless Routing Protocol (WRP) [673]. Popular on-demand protocols are, among others, Dynamic Source Routing (DSR) [536], AODV [634], Temporally Ordered Routing Algorithm (TORA) [619], Associativity-Based Routing (ABR) [825],and Signal Stability Routing (SSR) [212]. Overviews of this topic can be found in various books on ad hoc networks [373, 635, 827] and survey papers [356, 680, 707]. A common problem for many of these ad hoc routing protocols is that they require flooding of control messages to explore the network topology and to find destination nodes. The full range of ad hoc networking is too broad to be covered here in full detail and not all the research in this context is relevant to the case of wireless sensor networks (e.g. routing of multimedia traffic in ad hoc networks). Rather, the exposition in this chapter will concentrate on the most crucial aspect: energy efficiency. This pertains both to the selection of energy-efficient routes as well as to the overhead imposed by the construction of the routing tables themselves. Secondary aspects that are briefly touched upon are stability and dependability of the routes as well as routing table size (nodes with limited memory cannot store large routing tables).

, when nodes rely on energy scavenging for their operation, they might have to power off at unforeseeable points in time until enough energy has been harvested again. Consequently, it may be desirable to use not only a single path between a sender and receiver but to at least explore multiple paths. Such multiple paths provide not only redundancy in the path selection but can also be used for load balancing, for example, to evenly spread the energy consumption required for forwarding. Multipath routing schemes have been considered in the ad hoc literature as well [585, 618, 641, 837]; they are treated in Sections 11.2 and 11.3 where appropriate. Apart from the unicast case, where one node sends packets to another, uniquely identified node, both broadcasting (sending to all nodes in a network)1 and multicasting (sending to a specified group of nodes) are important tasks in WSNs. Both these tasks are treated in Section 11.4. One special way to define such a group is by specifying a geographic region such that all nodes in the region should receive the packet. This requires nodes to know about their positions, and once such knowledge is available, it can be used both to assist conventional routing and as a definition for target groups in a multicast sense. Section 11.5 describes such geographic routing approaches
All these options discussed so far are in a sense node-centric in that certain nodes are addressed by source nodes and packets should be delivered to these nodes. An alternative view on routing is enabled by data-centric network where the set of target nodes is only implicitly described by providing certain

attributes that these nodes have to fulfill (geographic routing can indeed be conceived of as data-centric routing in this sense). These routing approaches are very important in WSNs as they reflect natural usage cases – in particular, collection of data and dissemination of events to interested nodes. These approaches are treated separately in Chapter 12 along with in-network processing schemes for aggregating information in the network

## 4.2 Gossiping and agent-based unicast forwarding

### 4.2.1 Basic idea

This section deals with forwarding schemes that attempt to work without routing tables, either because the overhead to create these tables is deemed prohibitive (when a node only issues a command, for example, and does not expect any answers) or because these tables are to be constructed in the first place. The simplest option is flooding – forwarding each new, incoming message – but more efficient schemes are desirable. The topology-control discussion in Chapter 10 has already shown that reducing the forwarding set can considerably improve efficiency. The approaches taken here try to find a forwarding set without recurring to topology-control mechanisms but try to solve it strictly locally. The perhaps earliest paper [201] along these lines draws a parallel between the distribution of data in a replicated database system and epidemics occurring in human populations. Various options are described; one is “rumor mongering”: Once a site receives an update, it periodically, randomly chooses another site to propagate this update to; it stops doing so after the update has already been received by a sufficient number of sites (supposedly similar to the way rumors or epidemics are propagating in a population). The goal is to spread updates to all nodes as fast as possible while minimizing the message overhead. The question is to select neighbors for gossiping the rumor at hand (how often, which neighbors, etc.).

This same idea of randomly choosing forwarding nodes can also be applied to wireless sensor networks. There is, in fact, one advantage of wireless communication over wired communication that comes to bear in this context: a single transmission can be received by all neighboring nodes in radio ranges, thus incurring transmission costs only once for many neighbors. This property has been called the wireless multicast advantage [874]. Evidently, whether this advantage is actually relevant heavily depends on the deployed MAC protocol and on the relative costs of sending and receiving.

### 4.2.2 Randomized forwarding

On the basis of this consideration, Haas et al. [320] look at the question how information spreads in a wireless network by such a gossiping mechanism. The key parameter of their mechanism is the probability with which a node retransmits a newly incoming message. In the simplest case, this probability is constant. They show that there is a critical probability value below which the gossip – typically – dies out quickly and reaches only a small number of nodes. If, on the other hand, nodes use a probability larger than the critical threshold to retransmit messages, then most of the gossips reach (almost) all of the nodes in the network. Typical value for the critical threshold are about 65 to 75 %. The existence of this threshold shows that gossiping exhibits a typical phase transition behavior, in accordance with what can be expected from a percolation-theoretical treatment of the problem.

Haas et al. [320] also point out that nodes near the boundary of the sensor network's deployment region are critical as they have, on average, a smaller number of neighbors than nodes in the center of the region. They discuss various possible remedies, for example, (i) to have the neighbors of a node with few neighbors retransmit with higher probability, (ii) to prevent a gossip from dying out too fast by retransmitting messages over the first few hops with probability 1, or (iii) to retransmit a message (despite having decided not to do so) if the node does not overhear the message repeated from at least one of its neighbors (the actual minimum number is an optimization problem). Using such enhancements, the ratio of nodes that receive a gossip is considerably increased. Ni et al. [595, 596], Tseng et al. [833] also look at a similar problem, under the perspective of implementing broadcasting. They propose a couple of heuristics that let a node decide when to repeat a received or overheard packet. They look at rules that are based on counters (do not retransmit when a message has been overheard a certain number of times), distance based (do not retransmit if the distance to the sender is small), or location based (determine the additional coverage that could be obtained by retransmitting, based on the location of the nodes that have already sent the message).

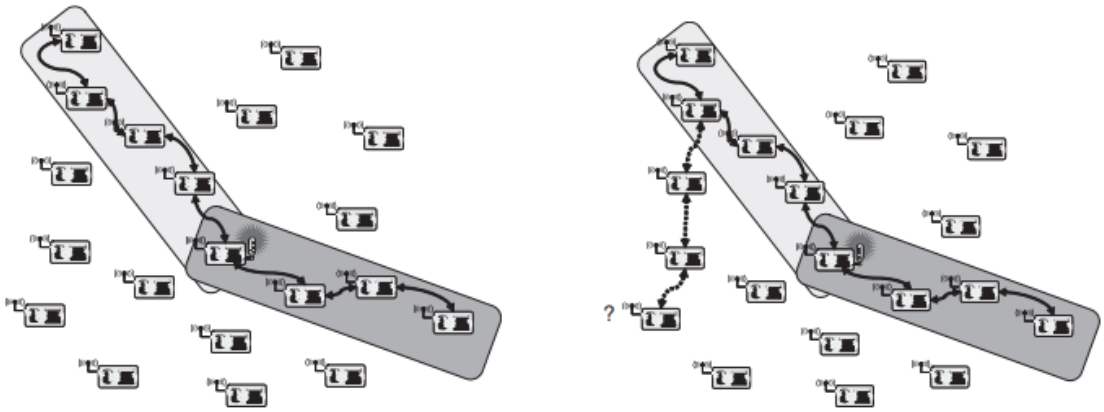
### 4.2.3 Random walks

Limiting flooding by only probabilistically forwarding a packet is only one option. Another approach is to think of a data packet as an "agent" that wanders through the network in search of its destination. In the simplest form, this is a purely random walk, where a packet is randomly forwarded to an arbitrary neighbor. Hence, the agents are sent via unicast, not via local broadcast, to their next hop. Instead of a single "agent", several of them can be injected into the network by the source to shorten the time to arrival by parallelism.2 The probabilistic properties of random walks have been extensively studied, but without any additional measures, a purely random

walk is too inefficient to be useful for WSNs. Two examples of such extensions to random walks shall be briefly discussed.

Rumor routing

Braginsky and Estrin [99] consider this approach in the context of event notification: Assume some sensors are interested in certain events (e.g. temperature exceeding a given value) and a sensor can observe it. Classical options are to flood either the query for the event or the notifications that an event has occurred through the entire network. The "rumor routing" approach proposed here does not flood the network with information about an event occurrence but only installs a few paths in the network by sending out one or several agents. Each of these agents propagates from node to node and installs routing information about the event in each node that it visited. This is illustrated in Figure 11.2(a) where the node in the middle detects an event and installs two event paths in the network (shaded areas)




(a) Setting up two event paths from the event source in the middle

(b) Node starts a query (in the lower left corner, marked with "?"), propagating until it meets a pre-installed event path

**Figure 11.2** Rumor routing example

Once a node tries to query an event (or to detect whether an event actually exists), it also sends out one or more agents. Such a search agent is forwarded through the network until it intersectswith a preinstalled event path and then knows how to find an event. In Figure 11.2(b), the node in the lower left corner sends out such a search, which happens to propagate upward until it intersects with one event path. All these agent propagations are limited to avoid endless circling of data. The rationale behind this technique is the relatively high probability that two random lines in a square intersect each other; Braginsky and Estrin [99] state a probability of about 69 %. While neither the event paths nor the search paths will in reality be straight lines, the approximation is claimed to be good enough. Using five instead of one event paths increases this probability to about99.7 %. In

effect, rumor routing allows to trade off effort in path creation and/or search against probability of detecting an event.
There are a few more functionalities included in rumor routing. For example, agents spread information about more than one event if they have crossed an event path for another event. Also, an agent uses opportunities to shorten existing event paths if they know about shorter paths.

### Random walks with known destination

A different perspective on random walks is taken by Servetto and Barrenechea [750]. They consider the problem of a WSN where lots of nodes are redundantly deployed but some of these nodes are randomly turned off and later on again (e.g. due to energy scavenging), giving rise to a dynamic graph. The idea is to use random walks to ensure that all possible paths in the network are used with equal probability, spreading the forwarding burden over all nodes. To do so, only local computations should be required for each node. The concrete scenario under investigation in reference [750] is a rectangular grid of nodes where the source is in the upper left corner and the destination in the lower right corner; nodes in between are randomly active. For such a situation, formulas are developed to compute the probability of passing an incoming packet either down or to the right, based on a distributed computation of the number of paths from the source to an intermediate node and from the intermediate node to the destination. Compared to assigning both the lower node and the node to the right a probability of 50 % each, the random walks based on these formulas indeed result in a much more uniform traffic density in the network

## 4.3 Energy-efficient unicast

### 4.3.1 Overview

At a first glance, energy-efficient unicast routing appears to be a simple problem: take the network graph, assign to each link a cost value that reflects the energy consumption across this link, and pick any algorithm that computes least-cost paths in a graph. An early paper along these lines is reference [747], which modified Dijkstra's shortest path algorithm to obtain routes with minimal total transmission power. What qualifies as a good cost metric in general is, however, anything but clear and depends on the precise intention of energy-efficient unicast routing. In fact, there are various aspects how energy or power efficiency can be conceived of in a routing context. The presentation here mostly follows reference [768]; acronyms are taken from reference [826]. Figure 11.3 shows an example scenario for a communication between nodes A and H including link energy costs and available battery capacity per
node.

**Minimize energy per packet (or per bit)**

The most straightforward formulation is to look at the total energy required to transport a packet over a multihop path from source to destination (including all overheads). The goal is then to minimize, for each packet, this total amount of energy by selecting a good route. Minimizing the hop count will typically not achieve this goal as routes with few hops might include hops with large transmission power to cover large distances – but be aware of distance-independent, constant offsets in the energy-consumption model. Nonetheless, this cost metric can be easily included in standard routing algorithms. It can lead to widely differing energy consumption on different nodes

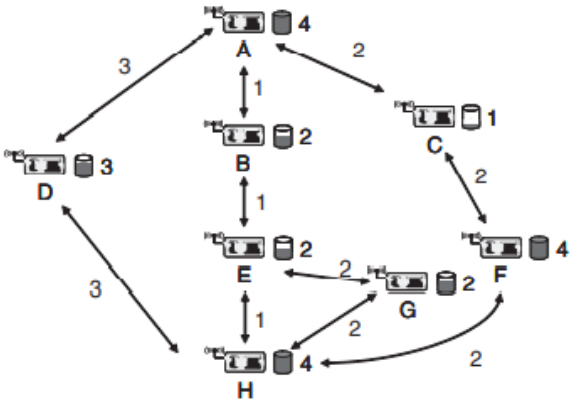



**Figure 11.3** Various example routes for communication between nodes A and H, showing energy costs per packet for each link and available battery capacity for each node (adapted from reference [17])

In the example of Figure 11.3, the minimum energy route is A-B-E-H, requiring 3 units of energy. The minimum hop count route would be A-D-H, requiring 6 units of energy.

**Maximize network lifetime** A WSN's task is not to transport data, but to observe (and possibly control). Hence, energy-efficient transmission is at best a means to an end and the actual end should be the optimization goal: the network should be able to fulfill its duty for as long as possible. Which event to use to demarcate the end of a network's lifetime is, however, not clear either. Several options exist (already discussed at length in Section 3.2.2): • Time until the first node fails. • Time until there is a spot that is not covered by the network (loss of coverage, a useful metric only for redundantly deployed networks). • Time until network partition (when there are two nodes that can no longer communicate with each other) [136, 768]. While these aspects are related, they require different solutions. For the network partition, for example, nodes in the graph's minimal cut set should have equal energy consumption (or rather, supplies) to ensure maximum time to partition. Also, their solutions can be infeasible – for example, maximizing the time to network partition is reported as NP-complete [768]. Moreover, Li et al. [489] state that maximizing the time until the first node

runs out of energy does not have a constant competitive ratio with the optimal off-line algorithm that knows the arrivals of future packets (when optimizing the number of messages the network can successfully carry, a competitive ratio logarithmic in the number of nodes can be shown [402]). Because of these theoretical limitations, only approximative solutions are practically relevant.

**Routing considering available battery**energy While maximizing the network lifetime is clearly a useful goal, it is not immediately obvious how to reach this goal using observable parameters of an actual network. As the finite energy supply in nodes' batteries is the limiting factor to network lifetime, it stands to reason to use information about battery status in routing decisions. Some of the possibilities are:

**Maximum Total Available Battery Capacity** Choose that route where the sum of the available battery capacity is maximized, without taking needless detours (called, slightly incorrectly, "maximum available power" in reference [17]). Looking only at the intermediate nodes in Figure 11.3, route A-B-E-G-H has a total available capacity of 6 units, but that is only because of the extra node G that is not really needed – such detours can of course arbitrarily increase this metric. Hence, AB-E-G-H should be discarded as it contains A-B-E-H as a proper subset. Eventually, route A-C-F-H is selected.

Minimum Battery Cost Routing (MBCR) Instead of looking directly at the sum of available battery capacities along a given path, MBCR instead looks at the "reluctance" of a node to route traffic [768, 826]. This reluctance increases as its battery is drained; for example, reluctance or routing cost can be measured as the reciprocal of the battery capacity. Then, the cost of a path is the sum of this reciprocals and the rule is to pick that path with the smallest cost. Since the reciprocal function assigns high costs to nodes with low battery capacity, this will automatically shift traffic away from routes with nodes about to run out of energy.

In the example of Figure 11.3, route A-C-F-H is assigned a cost of 1/1 + 1/4 = 1.25, but route A-D-H only has cost 1/3. Consequently, this route is chosen, protecting node C from needless effort.

**Min–Max Battery Cost Routing (MMBCR)** This scheme [768, 826] follows a similar intention, to protect nodes with low energy battery resources. Instead of using the sum of reciprocal battery levels, simply the largest reciprocal level of all nodes along a path is used as the cost for this path. Then, again the path with the smallest cost is used. In this sense, the optimal path is chosen by minimizing over a maximum. The same effect is achieved by using the smallest battery level along a path and then maximizing over these path values [17]. This is then a maximum/minimum formulation of the problem. In the example of Figure 11.3, route A-D-H will be selected.

**Conditional Max–Min Battery Capacity Routing (CMMBCR)** Another

option is to conditionalize upon the actual battery power levels available [826]. If there are routes along which all nodes have a battery level exceeding a given threshold, then select the route that requires the lowest energy per bit. If there is no such route, then pick that route which maximizes the minimum battery level.

**Minimize variance in power levels**To ensure a long network lifetime, one strategy is to use up all the batteries uniformly to avoid some nodes prematurely running out of energy and disrupting the network.3 Hence, routes should be chosen such that the variance in battery levels between different routes is reduced.

**Minimum Total Transmission Power Routing (MTPR**) Without actually considering routing as such, Bambos [47] looked at the situation of several nodes transmitting directly to their destination, mutually causing interference with each other. A given transmission is successful if its SINR exceeds a given threshold. The goal is to find an assignment of transmission power values for each transmitter (given the channel attenuation metric) such that all transmissions are successful and that the sum of all power values is minimized. MTPR is of course also applicable to multihop networks. A direct performance comparison between these concepts is difficult as they are trying to fulfill different objectives. Moreover, while these objectives are fairly easy to formulate, it is not trivial to implement them in a distributed protocol that judiciously balances the overhead necessary to collect routing information with the performance gained by clever routing choices. The following section describes some concrete protocols that tackle this challenge; a good overview is also included in reference [916]. Care has to be taken about the details here – Safwat et al. [715] show that a non-power-aware protocol can actually have (in many circumstances) a better energy-consumption behavior than some straightforward power-aware solutions (although it is not clear to what degree this conclusion is owing to the use of an IEEE 802.11-type MAC protocol in the paper).

## 4.3.2Some example unicast protocols

**Attracting routes**

by redirecting An early proposal by Gomez et al. [300] uses the idea that nodes can overhear packet exchanges between other nodes. If, in these packets, information about the energy required to communicatebetween two adjacent nodes X and Z is included, a third node Y can decide whether it can offer a more energy-efficient route by breaking the direct communication X-Z into a two-hop communication X–Y–Z. If so, Y can “attract” this route by sending route redirect messages to X, Z, or both. The advantage of this scheme is that its administrative overhead regarding explicit message exchanges is small. The need to overhear traffic is, however, not quite as appealing and makes this scheme not particularly suitable for WSNs

**Distance vector routing on top of topology control**

The relay regions concept described in Section 10.2.3 also lends itself to a

formulation of an energy-efficient routing problem. In reference [697], a Bellmann–Ford–type algorithm is used to find paths with minimal power consumption in the enclosure graph.

**Maximizing time to first node outage as a flow problem**

Chang and Tassiulas [137] attempt to maximize the time until the first node runs out of energy. To do so, they use a centralized, flow-based modeling approach. Given is a directed graph to represent the network annotated with the initial battery capacity of each node and, for each link, the energy costs to transmit a fixed-size packet. Moreover, the rates of data flows coming from certain nodes in the network, and their destination nodes are known. The goal is to find assignments of flows to forwarding nodes such the time until the first node runs out of energy is maximized. This problem can be formulated as a linear programming problem with certain conditions on flow conservation. As the forwarding energy differs at each node, the normal maximum flow algorithms are actually not applicable to solve this problem. Therefore, two approximation algorithms are proposed. The core idea of the first algorithm is to find a generalized description of the "costs" of a link. The observation is that both the actual energy cost eij of a link from node i to j as well as the initial and residual battery capacity Ei and Ei of node i should be taken into account. Hence, a generalized link cost cij = eα ijEβ i /Ei γ is used, where α, β, and γ are nonnegative weighting factors. Setting some of these factors to 0 and computing the "lowest cost" paths results in the routing strategies described above; it also allows to generalize these diverse approaches into a single metric. The second algorithm is a flow redirection algorithm. Both of them are distributed and base on locally available information. The core result is that system lifetime can be extended up to 60 % in the scenarios investigated here in comparison to simple minimum energy routing when battery capacity also is taken into account. Further information on this approach can be found in references [136, 138]; Zussman and Segall [948] discuss an extension of the same basic techniques to an anycast routing problem.

**Maximizing time to first node outage by a max–min optimization**

Li et al. [489] approach the network lifetime maximization problem as a max–min optimization problem. They propose two algorithms. One of them has to know the battery power level of each node in the network and the other can work without this information at only slightly reduced performance

**The max min zP min approximation**

This heuristic starts out from the intuition to use paths that have a large residual energy, that is, that path where the minimal remaining power in all nodes is the largest. This heuristic, however, can result in arbitrarily bad performance (compare Fig. 3 in reference [489]). Moreover, this approachdoes not take into account the total power consumption of a given

path, possibly giving preference to very expensive paths (in absolute terms). Hence, a proper compromise must be found. The idea is to use a max–min path but limit its maximum power consumption. This limit cannot be chosen in absolute terms but is best defined as a ratio to the path with the smallest possible power consumption. Thus, the path to be chosen should have at most a power consumption of a factor z times the power consumption along the most efficient path, Pmin; among the paths that fulfill these constraints, the max–min path with respect to battery power will be selected as the actual path to be used. Evidently, proper choosing of $1 < z < \infty$ will determine the efficiency of this approximation. In fact, z should adapt itself to the residual power levels in the network. This adaptation can be based on estimates of the shortest remaining lifetime of all nodes in the network, estimated over some period T . Parameter z will be additively increased or decreased after each period and the sign of the change depends on whether the estimated shortest lifetime has increased or decreased compared to the previous period. The magnitude of the adaption is reduced over time. This heuristic is shown to have good performance by both simulations and analysis

**The zone routing approximation**

The disadvantage of max min zPmin is that knowledge of battery power levels is required. The "zone routing" heuristic removes this need. This is done by partitioning the network in geographical zones where nodes within the zones are responsible for routing in the zone. Routing among zones is organized hierarchically. The reader is referred to reference [489] for a full description. The main point is that the resulting performance loss is relatively small, showing that the approximative maximum lifetime routing can be implemented on the basis of locally available information (assuming location information is available).

**Maximizing number of messages**

A slightly different optimization goal is to maximize the number of messages that can be sent over a network before it runs out of energy. In practice, this can be more important than maximizing the time until the first node runs out of energy, depending on what can be assumed about the actual data sensing process and energy consumption. Kar et al. [402] consider this problem. Interestingly enough, they can prove a competitive ratio that is logarithmic in the network size if admission control (the routing algorithm is allowed to reject messages although there would be a path to carry the message) is assumed. What is more, the constructed "CMAX" routing algorithm does not depend in practice on admission control but performs well nonetheless (even when using network lifetime as figure of merit, for which the algorithm is not actually designed). The crucial insight for this property is the choice of the link weights. Given an edge between nodes i and j with energy costs eij to transmit a message of unit size, the weight wij is chosen as $w_{ij} = e_{ij}(\lambda\alpha_i - 1)$, where $\lambda$ is a constant and $\alpha_i$ is the fraction of

battery capacity that node i has already used up when the present routing decision is to be made. Admission control is then formulated as disregarding paths the total weight of which exceeds a given threshold. In practice, this path weight threshold can be ignored. Using any standard algorithm, the lowest cost path (based on these link weights) is then selected to transmit the packet. The information needed for this heuristic and the max min zPmin heuristic are identical. In fact, the same zone-based variant should apply to this heuristic as well. An obvious improvement over the max min zPmin heuristic is that only a single shortest path computation is necessary. Reference [402] discusses performance properties of the CMAX routing algorithm in detail. In short, it performs well with respect to maximum number of messages and, with respect to network lifetime, outperforms even specialized algorithms like the max min zPmin heuristic.
Note that both this and the previous scheme – and similarly all other schemes that depend on battery capacity – have to recompute routing tables relatively frequently to mirror the change in available capacity. This can be a considerable burden.

**Bounding the difference between routing protocols**The previous sections have discussed several possible approaches to energy-efficient unicast routing. It seems like the choice of the routing has a considerable impact on the chosen energy efficiency metric. But the remaining question is: What is the maximal improvement that can be gained from an improved routing protocol? What is the biggest possible difference between a stupid and a perfect routing protocol? Alonso et al. [19] answer this question (Figure 11.4). They consider a class of networks where a all nodes transmit with identical power and all nodes continuously have data to deliver to a base station (possibly over multiple hops). Apart from these assumptions, their results apply to a wide range of real networks, irrespective, for example, of the concrete topology; data aggregation is not considered. The key question is the energy consumption of nodes during the data exchange, not the overhead of the routing protocol itself. To approach this problem, the graph is partitioned into “spheres” Si that include all the nodes that are reachable from the base station in at most i hops. The interesting case is then networks where “most” nodes are more than a single hop away from the base station (otherwise, the network is not particularly interesting anyway). Then, all traffic has to go through the nodes of sphere S1, and because there are relatively few of these nodes, they limit the lifetime of the network. For such networks, the authors show that no routing will have an energy efficiency worse than a factor of $2|S1| - 1$ than the best possible one – and $|S1|$ is a small constant compared to the network size. Put the other way around, unless the number of direct neighbors of the base station is large, the possible impact of routing is limited. For example, for a base station with four neighbors, even the worst possible routing protocol will only reduce the

energy efficiency of data deliveryby a factor of seven compared to the optimal routing. Nonetheless, while such factors may not be impressive theoretically, they obviously have a large practical relevance.4

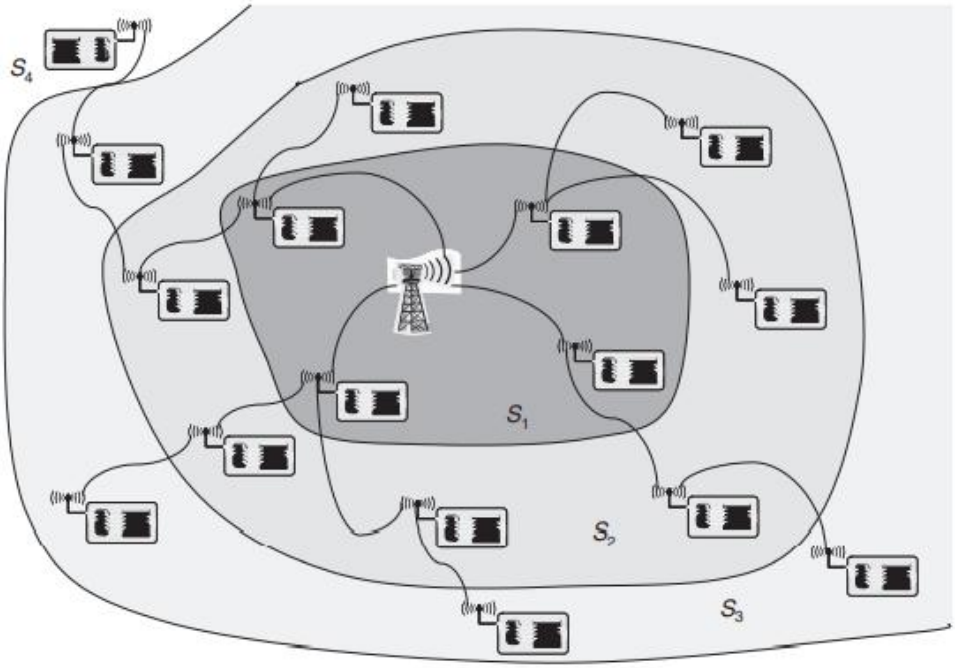



**Figure 11.4** Spheres and balls as used by ALONSO et al. [19]

## 4.3.3Multipath unicast routing

OverviewThe unicast routing protocols discussed so far tried to construct a single energy-efficient path (with whatever interpretation of this term) between a sink and a receiver, typically by giving a clever meaning to the "cost" of a link. These costs try to balance, for example, energy required for communication across this link against the battery capacity of the nodes involved. Focusing on choosing the best possible path, however, limits the opportunities for making such trade-offs. Extending the focus to multiple paths and trying to balance, for example, energy consumption across multiple path is therefore an option worthwhile exploring. Moreover, multiple paths provide redundancy in that they can serve as "hot standbys" to quickly switch to when a node or a link on a primary path fails. Such multipath routing protocols construct several paths between a given sender and receiver. The basic goal is to find k paths that do not have either links or nodes in common (apart from source and destination node, of course). Some basic references on finding multiple paths in general networks are [475, 604, 765]. Once the paths have been established by the routing protocol, the forwarding phase can then dynamically decide which path (or even paths) to choose to transmit a packet. This can increase the robustness of the forwarding process toward link or node failures. Applying multipath routing

to wireless networks, both general ad hoc and sensor networks, is a well-studied problem; some example references include [32, 50, 473, 584, 585, 612, 627, 929]. Some of the more WSN-relevant papers are briefly described here. But even some routing schemes discussed in other chapters, for example, directed diffusion, have multipath characteristics, even though it might not be their most prominent feature.

**Sequential Assignment Routing (SAR)**

As a basic rule of thumb, computing such k-disjoint paths requires about k times more overhead than a single-path routing protocol [778]. Sohrabi et al. [778] try to reduce the multipath-induced overhead by focusing the disjointness requirements to that part of a network where they truly matter – near the data sink, as the nodes close to the sink are (often) those that likely are going to fail first because of depleted battery resources. Hence, they only require paths to use different neighbors of the sink. The Sequential Assignment Routing (SAR) algorithm achieves this objectiveby constructing trees outward from each sink neighbor; in the end, most nodes will then be part of several such trees. A packet's actual path is then selected by the source on the basis of information about the available battery resources along the path and the performance metrics (e.g. delay) of a given path.

**Constructing energy-efficient secondary paths**

When using multiple paths as standby paths to quickly switch to when the primary path fails, an obvious concern is that of the energy efficiency of these secondary paths compared to the (hopefully) optimal primary path. Ganesan et al. [276] consider the question how to construct the secondary paths from this perspective, without worrying about battery capacity or similar metrics along the various paths. Their first observation is that strictly requiring node disjointness between the various paths tends to produce rather inefficient secondary paths as large detours can be necessary.

To overcome this problem and yet retain the robustness advantages of multiple paths, they suggest the construction of so-called "braided" paths (sometimes also called "meshed" multipaths [195]). These braided paths are only required to leave out some (even only one) node(s) of the primary path but are free to use other nodes on the primary path. This relaxed disjointness requirement results in paths that can "stay close" to the primary path and are therefore likely to have a similar, close to optimal energy efficiency as the primary path. Figure 11.5 illustrates these two redundant paths' concepts.

Constructing these two different types of redundant paths is simple in a centralized fashion; a distributed construction is described in reference [276] as a modification to the reinforcement mechanism (popularized by directed diffusion, Section 12.2.2). For disjoint paths, the data sink not only reinforces the primary path via its best neighbor toward the data source but also sends out an "alternate path" reinforcement to its second-best neighbor

(or several such neighbors, for multiple standby paths). This alternate path reinforcement is then forwarded toward the best neighbor that is not already on the primary path. For braided paths, each node on the primary path (including the sink) sends out such an alternate path reinforcement, which only has to avoid the next upstream node on the primary path but is then free to use nodes on the primary path.

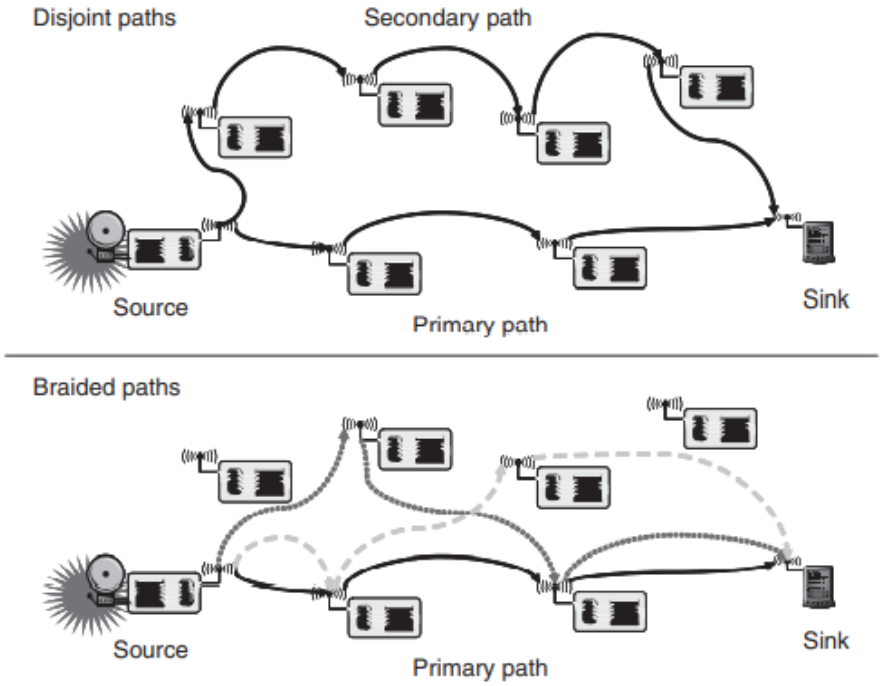



**Figure 11.5** Disjoint and braided paths around a primary path

Which of these two schemes is advantageous clearly depends on the node failure patterns. The authors look at both independent node failures and so-called "patterned" failures (all nodes within a circle of known radius around randomly selected points fail; the appearance of points follows a Poisson distribution). The main figure of merit is the "resilience", the percentage of cases where the failure of the primary path is compensated for by an alternative path. Reference [276] discusses the relative performance in detail; in summary, braided paths tend to have a better overall resilience.

**Simultaneous transmissions over multiple paths**

When using multiple paths as a standby for the primary path, failover times might be improved compared to strictly single-path solutions. Nevertheless, there is some delay in detecting the need to use a secondary path. Depending on which node makes this decision – only the source node or any node on the primary path – there can be more or less overhead involved. To further shorten the time to delivery and to increase the delivery ratio of a given packet, it is also conceivable to use all or several of the multiple paths simultaneously. The simplest idea is to assume node-disjoint paths and to

send several copies of a given packet over these different paths to the destination. Clearly, this trades off resource consumption against packet error rates. De et al. [195] provide a performance comparison of such a packet replication scheme with other multipath schemes, for example, one that uses additional FEC to protect against packet errors. Dulmann et al. [214] combine the basic idea of sending packet replicas with FEC by proposing to split a packet and its error correction redundancy over several paths, to be recombined at the receiver. The degree of redundancy and the number of paths can be tuned to the expected error behavior, trading off overhead against residual packet error rate.

**Randomly choosing one of several paths** When maintaining multiple paths, it actually makes sense also to use paths that are less energy efficient than the optimal one. One reason to do so is to share the load among all nodes in order to use the available battery capacity in the network better. A relatively straightforward way of doing so is described by Shah and Rabaey [752]. Each node maintains an energy cost estimate for each of its neighbors (toward the destination, packets are not routed "away" from their destination). When forwarding a packet, the next hop is randomly chosen proportional to the energy consumption of the path over this neighbor. To the upstream node, the appropriately weighted average of these costs (i.e., the harmonic mean of the costs) is reported. More formally: suppose node v has neighbors v1 to vn that advertise cost c1,...,cn, respectively. Node v will advertise c = n/n i=1 ci as its own cost and will forward an incoming packet to neighbor i with probability (1/ci)/(1/ n i=1 ci). This routing approach is extended by Willig et al. [877] by introducing the notion of altruists. An altruistic node is one that is willing to do more work on behalf of its neighbors, for example, because it has a tethered power supply. Willig et al. show that such asymmetric nodes can be efficiently exploited by the routing protocol, simply by occasionally broadcasting "altruistic announcements" into the network.

**Trade-off analysis**

Clearly, supporting such multiple paths in a network implies a trade-off between robustness (the probability that paths are available even after node failure) and energy efficiency (as both the 304 Routing protocols management of these paths and the nonoptimal choices made for packet forwarding decisions imply increased energy expenditure) – irrespective of the concrete routing protocol in use. This tradeoff is analyzed by Krishnamachari et al. [440], who compare the robustness gained by multiple paths with those owing to simply increasing transmission power. Their basic observation, made in a simplified scenario of five nodes, is that it is not possible to simultaneously optimize both robustness and energy efficiency of a given set of paths, but rather that only the notion of Pareto optimality can be applied.5 They do observe, however, that singlepath solutions that require

a larger transmission power tend to dominate multipath solutions with low transmission power. To test these basic observations, the authors conducted a set of simulation experiments, comparing various degrees of redundancy via braided multipaths. As one might expect, for low failure rates, the robustness of even two paths is perfectly sufficient. The two controlled parameters are the degree of redundancy via additional paths and the maximum transmission power, enabling the system to bridge across failed nodes if necessary. Using these two factors to influence Pareto optimality with respect to the robustness and energy efficiency objectives shows, interestingly, that the single-path schemes actually perform "best". Overall, the results of this paper highlight the need to carefully choose between various sources of redundancy.

**Routing and topology control**

In a clustered network or in a network where the topology is based on a dominating set, the routing problem has to be solved as well. Superficially, the problem is simple owing to a reduced network topology. However, it is not clear how, for example, information about battery capacity can be taken into account. A few references dealing with this topic are [150, 150, 356, 488, 628, 771]; LEACH [344, 346], for example, also belongs to this class of approaches.

**Maximizing data flow for multiple source/destination pairs**

In reference [781], the authors look at a situation where several sources of data are distributed in the network, each one trying to send as much energy to a dedicated sink, possibly using multiple routes, each one equipped with a utility function. The optimization is to select routes such that the total utility of the network is maximized before the first node runs out of energy. The authors derive a flow control algorithm; the techniques used here are interesting and should be applicable to similar problems as well

**Consider all costs** Some of the previously discussed papers have tried to find minimum (energy) cost paths but did not necessarily take into account all possible sources of energy consumption. Banerjee and Misra [51] argue, for example, that costs for retransmissions have to be taken into account as well (normally, residual error rates over wireless links cannot be neglected) and compare paths resulting from local retransmission schemes with end-to-end retransmission schemes.

**Integrate scheduling and power control**To obtain optimal solutions, Cruz and Santhanam [185] describe a scheme that takes into account link scheduling and power control jointly with the computation of routes. The result is a relatively complicated optimization problem that minimizes total average power consumption, based on traffic requirements for all links. The resulting scheme has some strong assumptions, though. A similar approach is taken by Bergamo et al. [65], who use the results of a power control

algorithm as an input to classical routing algorithms.

**Routing and link quality**

A related issue is the quality of the underlying links. While most routing protocols are formulated in a graph-theoretical manner, it is often by no means clear which nodes are connected by a link. Links fluctuate in reliability and can have relatively high packet error rates. Using flooding-based protocols over such links can result in rather convoluted routing tables where nodes are considered to be neighbors only because a flooding packet happened to go through despite actually bad link quality. To overcome these problems, Wang et al. [859] advocate a careful selection of actual neighbors (in their case, parents for a routing tree toward a data sink), using information that the link layer can provide.

**Routing and lifetime guarantees**

The determined routes evidently influence the lifetime of the network and several of the discussed protocols address this issue. References [716, 717] go a step further in that they attempt to provide guarantees on the lifetime of the network. Routing for one-shot queries In WSN, a typical routing problem is that of a query that is to be routed to the place where it can be evaluated and then the routing of the answer back to the place where the query originated. Depending on the dynamics of the network, such a query can be regarded as a "one-shot" event if the structure of the network has changed sufficiently by the next query so that any topology information that the first query might have acquired is already outdated. In such a situation it is not clear how to route a query and the answer. A typical assumption is that the location of the target is known and that geographical routing can be used or some form of data-centric routing comes into play. Helmy [348] proposes a scheme that handles such one-shot queries without recurring to location knowledge. The intuition behind it is that nodes know about their R-hop neighborhood; for queries outside this immediate neighborhood, so-called contact nodes are involved. These nodes are selected by nodes at the border of a given vicinity when the query has not been satisfied locally

## 14.4 Broadcast and multicast

### 4.4.1 Overview

The protocols were trying to find efficient means to transmit data from one node to another one, possibly over multiple hops. In doing so, some of them had to collect or distribute information to all nodes in the network; they had to perform a broadcast operation. In fact, broadcasting can be a common operation in many wireless network applications.Similarly, it is often necessary to distribute some data to a given, typically known, subset of all nodes in the network; this is called multicast.

Multicast bases on the same principal ideas but both these tasks deserve some specific treatment from the routing perspective. The multicast problem

in a graph G = (V , E) can be described by a set of sources S = (s1,...,sn) and, for each source, a set of destinations Di = (di1,...,dimi), for each i = 1,...,n. In general, Di ⊂V . A frequent simplification is to assume that all destination sets are identical. Also, the edges of the graph are annotated with communication costs.

**Source-based tree**The first idea is to construct, for each source, a tree, rooted at the given source, that contains all the destinations for this source and, if necessary, additional nodes of V to ensure that the tree can be constructed. Which tree to select (out of, in general, many possible ones) is determined by the optimization goal, which reflects the link costs:

**For each source, minimize the total cost** Try to find a tree for which the sum of all link costs is minimal (over all possible trees rooted at the source). This is the Steiner tree problem6 and matches best the intuitive expectation how a multicast routing structure should look like. It is, however, NP-complete (yet approximable within 1 + (ln 3)/2 ≈ 1.55 [182]). Note that the minimal cost broadcast problem is equivalent to the minimal cost spanning tree problem (unless "wireless advantage" is assumed, see below), which is in fact solvable in polynomial time.

**For each source, minimize the maximum cost to each destination** Owing to the complexity of the previous optimization problem, a different optimization goal can be considered: Instead of trying to minimize the total cost of the tree, one can minimize the costs to each individual destination separately. In effect, this maps the multicast problem to repeated unicast shortest path problems, which can be solved by any routing algorithm, for example, Dijkstra's [147, 874]

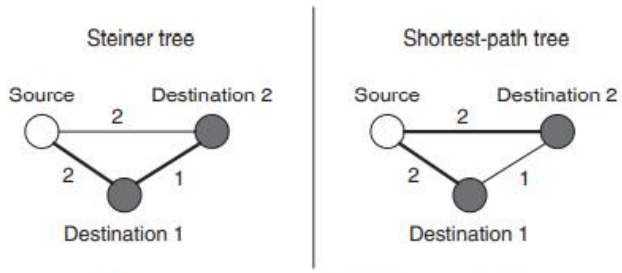



**Figure 11.6** Difference between Steiner tree and shortest-path tree (thick lines indicate links that are part of the tree)

Figure 11.6 illustrates the difference between these two optimization problems and the resulting trees

**Shared, core-based tree** Constructing and maintaining a dedicated tree for each source incurs considerable overhead. This overhead can be reduced if only a single tree is maintained – typically, this is promising when the

destination sets for all sources are identical. The obvious downside is that for a given source the paths to its destinations can, in general, no longer be as short as with a dedicated, source-based tree. To share a tree among several sources, a representative node in the network (not necessarily a source or a destination node) is selected and, from this node, a tree is constructed to contain all destination nodes; this tree is shared among all the sources. Selecting this core node is again NP-complete and optimization goals similar to the source-based trees can be considered. In such a shared tree concept, the core node evidently becomes a single point of failure. To overcome this shortcoming, multicore shared trees are also considered in the literature.

**Mesh** While trees represent the overhead-optimal routing structures, they are not redundant – failure of even a single link will disconnect the tree. Adding additional links to the tree to obtain redundancy, however, will alter its essential properties, in particular the absence of cycles. The resulting routing structure will be a mesh [283] and requires more complicated forwarding structures than does a simple tree

Orthogonal to the resulting structure of the routing tables is the assumption about the forwarding behavior of a node. Does a node use local unicast or can it actually, by virtue of the broadcast nature of the wireless channel, reach several or all of its neighbors with a single transmission? The first option directly maps to the standard graph model (separate transmissions are required to reach each neighbor). The second option is attractive since only a single transmission can spread information to many neighbors, but whether this is realistic depends on the assumptions made about hardware, MAC protocol, and, in particular, sleeping cycles of nodes. Ultimately, the ratio between transmission power and idle power consumption comes into play here. When this wireless multicast advantage [874] is assumed, the design of protocols (and of the resulting trees or meshes) changes considerably as now not the sum of the costs of outgoing links has to be considered but only the cost of the most expensive link that has to be invested in to transmit data to all neighbors. In practice, when a link is activated in the course of a routing protocol, all the cheaper links of that node are implicitly contained in the routing graph as well. Clementi et al. [171] and also Cagalj et al. [113] show that even under this assumption the problem of computing the optimal multicast graph is still NP-complete; some approximation results are proved in references [852, 853] – the best known, provable approximation factor is 12, making this a harder problem than the Steiner tree construction, for example. It is interesting to note that other broadcast optimization problems are NP-hard as well. Gandhi et al. [270], for example, show that minimizing the broadcast latency (finding a broadcast tree such that the time until the latest node has received the message is minimized) is NP-hard, as is the problem of minimizing the number of retransmissions (they also provide

heuristics with a constant approximation factor). Ferreira and Jarry [256] consider the problem in the context of evolving graphs (graphs where the set of available links changes over time) and show that computing minimum spanning trees in a planar, mobile network is NP-complete. Figure 11.7 summarizes these principal options how to organize multicasting in a wireless network. The following sections will discuss typical representatives of these categories in some more detail. For completeness, it should be pointed out that flooding the network is also a means to implement broadcast and/or multicast. Efficient flooding structures (minimum connected dominating sets) have been discussed in Chapter 10. References [287, 610] are some examples where this concept is investigated further. Also, some of the gossiping techniques from Section 11.2 can also be amenable to implement a (probabilistic) broadcast

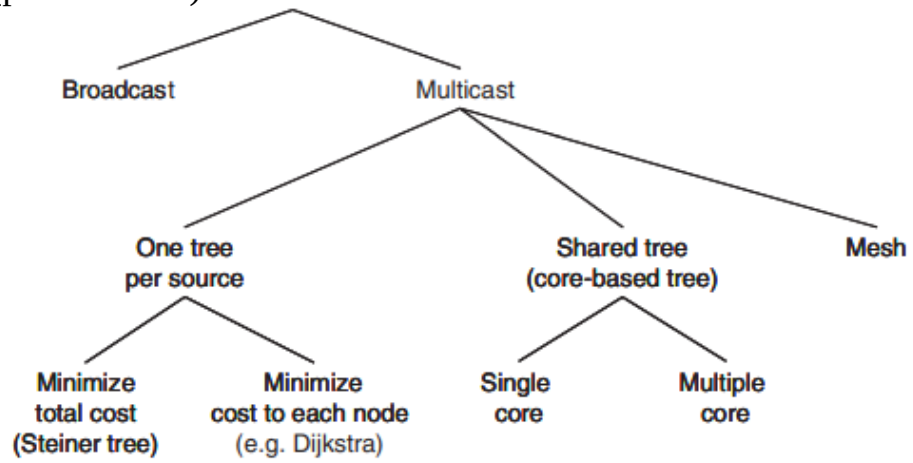



**Figure 11.7** Overview of possible multicast approaches

## 4.4.2 Source-based tree protocol

Source-based trees can be constructed in a number of different ways. First, a simple heuristic is explained that ("simplifyingly") maps the tree construction onto the problem of finding shortest paths. Then, an essential algorithm for broadcast trees is introduced and the multicast problem is solved by various Steiner tree approximations. These approaches work on a basic graph interpretation of the network. Solutions that exploit the wireless multicast advantage – in particular, BIP – are described later

### A greedy heuristic – Shortest Path Tree

A trivial heuristic for broadcasting and/or multicasting is to compute, to each destination, the shortest (or rather, cheapest) path and overlay all these paths onto a tree (described, for example, as Shortest Path Tree (SPT) in reference [874]). Wan et al. [852] show that this greedy heuristic does not have a good approximation ratio. Consider a network with the source node at the center, m nodes p1,...,pm distributed on the unit circle, and another m nodes q1,...,qm each placed on the line connecting the origin with node pi at

distance ε from the origin. SPT will send one broadcast packet (using the wireless multicast advantage) to the inner ring at cost ε2 (assuming quadratic path loss), and each node qi forwards at cost (1 − ε)2. The total cost is thus ε2 + m(1 − ε)2. Having the source node send at power 1 will, on the other hand, distributed the information to all nodes. Thus, the approximation ratio is (ε2 + m(1 − ε)2)/1, which converges to m as ε → 0. Wan et al. [852] also describe another greedy heuristic, broadcast average incremental power, which has an unsatisfying approximation ratio. The heuristics described below actually have constant ratios, outperforming greedy heuristics.

**Broadcasting using minimum cost spanning tree– Prim's algorithm**

A simple broadcasting algorithm can be based on a minimum cost spanning tree. One possible algorithm to compute it is due to Prim [657].7 It starts with a tree consisting of the source node, and in |V | − 1 steps adds one node per step to the tree. The node that is added is the one that has the lowest-cost link to any node already in the tree (evidently, this is not an immediately distributed algorithm)

More formally, given a graph G = (V , E) with edge weights W : E → R (for example, energy cost for the edge), the algorithm maintains the tree (VT , ET ) itself and a set of candidate nodes VC along with one candidate edge per candidate node. Listing 11.1 shows this algorithm in more detail

Listing 11.1: Prim's minimum-cost spanning tree algorithm

```
V_T = E_T = ∅
V_C = source node
while (V_T ≠ V) {
  Select v ∈ V_C with smallest candidate edge weight
  Add v to V_T
  foreach neighbor u of v in V \ {V_T ∪ V_C} {
    // new candidate u found
    add u to V_C
    add (v,u) as candidate edge
  }
  foreach neighbor u of v in V_C {
    if (W(v,u) < weight of u's existing candidate edge) {
      replace u's candidate edge with edge (v,u)
    }
  }
}
```

An important newer algorithm is a randomized one with linear-time complexity [403].

**Some Steiner tree approximations for multicasting**

**A simple approximation**

As computing the optimal Steiner tree is an NP-complete problem, quick approximations are required (references [371, 607] provide overviews). One simple approximation arbitrarily orders all the destination nodes as well as the source and then successively adds these nodes to the tree: For the first two nodes, a shortest path between these two nodes is constructed. Then, for

every node, construct the shortest path to any node already on the Steiner tree (where it is not a trivial task to determine what the best connection point on the tree is). The result will be a tree including all the required nodes; the quality of the approximation depends on the order in which nodes are added to the tree. In practice, this approximation tends to perform reasonably well.

**Takahashi Matsuyama**

Instead of fixing the order in which nodes are to be added to the tree a priori, it stands to reason to let the algorithm find the next best node to be added. Thus, start with the source node. In each step, determine the next so-far-unconnected destination that has the shortest distance to the already existing tree; add this node via a shortest path to the tree. Repeat for all destination nodes. This is the Takahashi–Matsuyama heuristic [806].

**KMB heuristic**

The KMB heuristic (after Kou et al. [434]), on the other hand, maps the Steiner tree construction in the original graph G to the finding of a minimum spanning tree in another graph. Let D be the set of all nodes to be connected by the Steiner tree (including the source node). Construct the complete graph K (all nodes are directly connected) with D as the node set; assign edge weights in K as the cost of the shortest path between the respective nodes in the original graph. Constructa minimum spanning tree in K. Then, transform back this spanning tree to the original graph, that is, replace the artificial links in K with the (in general) multihop paths in G, obtaining another graph T . This graph T need not necessarily be a tree, so compute another minimum spanning tree on T , obtaining T . This is almost the sought-after Steiner tree; it remains now only to remove any possibly unnecessary leaves. KMS is attractive since its competitive ratio is at most 2 and in practice KMS often comes within 5 % of the best possible solution

**A heuristic for multiple rates**

The Steiner tree is well suited for a situation where a data source periodically sends data updates to multiple sinks. The problem becomes slightly different if these sinks require data more or less frequently. In such a situation, branching points in the tree only forward data with the necessary maximum rate per child to cover all the sinks in the respective subtree. Adding a source to such a tree can then require an increase in the sending rate of a branching point and will thus have an impact on the energy consumption. Kim et al. [416] suggest a heuristic to handle this case, exploiting knowledge about location of nodes. When adding a new sink, they perform a recursive computation of the required additional power when using a given node as a branching point, taking into account possible needs to increase rates in a branch of the tree. As a result, paths might be used that are not strictly the most energy efficient ones but are those paths that already carry a high rate anyway. In addition, the tree can be locally modified to merge existing branches that run in parallel, pushing the branching point deeper toward the

sinks. Another heuristic that uses node locations is a modification of the Takahashi–Matsuyama heuristic and is described in reference.

**Broadcasting/multicasting with a finite set of power levels** The Steiner tree is also amenable to solve the broadcast or multicast problem when each node only has a finite set of k different power levels at its disposal. A simple helper graph construction [498] can solve this problem: Replace each node v in the original graph G = (V , E) with a small subgraph, consisting of v itself and k nodes uv1,...,uvk . Add an edge (v, uvi) and annotate it with a weight representing the transmission costs at power level i, i = 1,...,k. The helper graph is then the union of all these replacement graphs for each v, with the additional edges (uvi, w) (with w ∈ V an original node) if and only if node v can communicate with node w using transmission power level i (for each v and each i = 1,...,k). On this helper graph, solve a Steiner tree problem, with all or some of the original nodes as destinations. Further optimizations are possible if all nodes use the same transmission power levels.

**Exploiting wireless multicast advantage for broadcast: Broadcast incremental power**

In all the algorithms described so far, a node that wants to transmit to multiple neighbors (because it has multiple children in the tree, for example) experiences a proportional cost. The Broadcast Incremental Power (BIP) algorithm [874] differs here in that it exploits the wireless multicast advantage to compute a heuristic for a broadcast tree. The core idea is that a node that is already transmitting to some other node would only have to raise its transmission power in order provide data also to further nodes, without incurring cost for another transmission. Hence, the additional cost for a node to supply a further node with data is only the difference between the current and the needed (higher) transmission power.

On the basis of this idea, a modification of Prim's algorithm is possible. Like in Prim's algorithm, one node is added to the tree per round and each as-yet-not-added node maintains a "candidateedge" that represents its current best option to be added to the tree (updated each round). In each round, the node with the lowest-cost candidate edge is chosen. The difference lies mainly in the computation of the cost assigned to the candidate edge of a node and, since each node is reachable from each other node with sufficiently high power, there is no notion of a set of "candidate" nodes representing the fringe of the growing tree (alternatively, all nodes are in the fringe since the underlying graph is complete). An obvious modification of the algorithm assumes maximum transmission power per node, thus reintroducing possible candidates for transmission and speeding up the computing; this and other improvements are left out here for simplicity. The interesting part is the candidate edge weight computation: Unlike in Prim's algorithm, the currently used transmission power of a node is subtracted from the actual

edge weight to reflect the fact that only additional costs are incurred if the wireless multicast advantage can be exploited.

Listing 11.2: The broadcast incremental power algorithm for exploiting the wireless multicast advantage

```
// Initialize
V_T = {source node}
P(source node) = 0 // transmission power assigned to a node
foreach (v in V \ V_T) {
  Set candidate edge to (source node, v)
  Set candidate edge weight to transmission power to
    reach v from source node
}
// Compute tree
while (V_T ≠ V) {
  Select v ∈ V \ V_T with smallest candidate edge weight
  Add v to V_T using its candidate edge (u, v)
  Increase P(u) to smallest power that reaches v
  // Recompute candidate edges and their weights
  foreach (v in V \ V_T) {
    Select u which minimizes P'(u) − P(u)
    // where P'(u) ≥ P(u) is smallest power to reach v from u
    Set candidate edge to (u, v)
    Set candidate edge weight to P'(u) − P(u)
  }
}
```

This algorithm tends to produce good broadcast graphs. In some situations, however, it can unnecessarily assign high transmission power levels to nodes to cover neighbors that are already covered by some third node. Figure 11.9 shows such an example where nodes are placed at coordinates A = (0, 0), B = (3, 0), S = (4, 0), and C = (6, 0), the node at (4, 0) is the source, and the transmission costs between two nodes are assumed to be proportional to the square of the distance (path loss coefficient of 2). Since B has to use transmission power 9 anyway to cover node A, italso implicitly supplies node C, avoiding the need for S to send at more than transmission power 1. Detecting such opportunities to reduce transmission power is called a sweep operation [874]. Such sweeping is of course applicable to other broadcast/multicast algorithms as well.

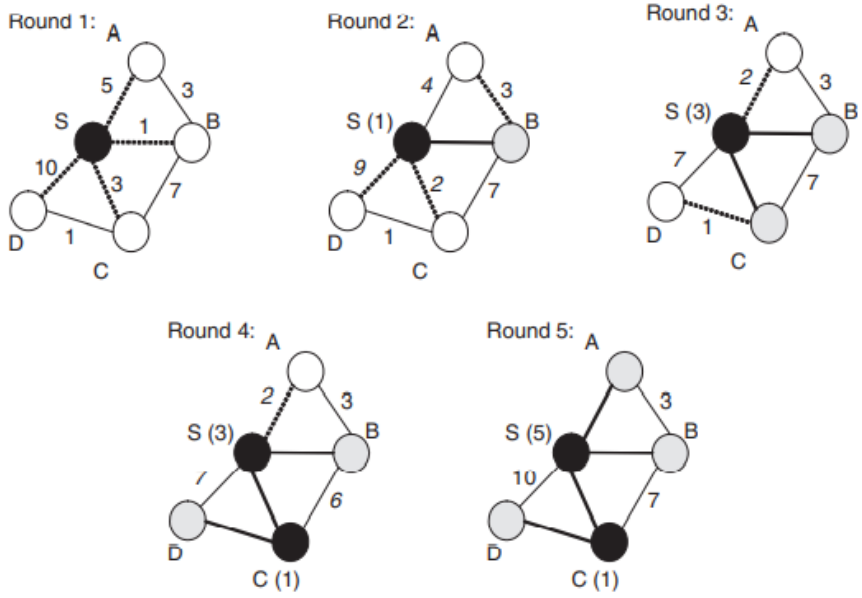



**Figure 11.8** Illustration of the BIP algorithms's operation (in parentheses, the currently used transmission power is shown)

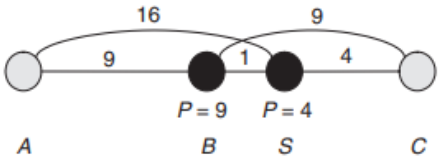



**Figure 11.9** Example of opportunity for "sweeping" in the BIP algorithm

**Exploiting wireless multicast advantage for multicast: Pruning broadcast trees by Multicast Incremental Power (MIP)**

On the basis of the computation of a broadcast tree, one simple heuristic for the multicast case is evident. Given a broadcast tree, prune it by removing all nodes from the tree that have no members of the destination set as "downstream" nodes; in addition, the sweep operation can be applied to reduce transmission power further if high power is only needed to supply subtrees without any destination nodes. Wieselthier et al. [874] study this pruning heuristic on the basis of several broadcast tree heuristics, in particular their BIP algorithm

**Embedded wireless multicast advantage – Transforming existing graphs**

A different approach to leverage the wireless multicast advantage is followed by Cagalj et al. [113]. They start from a "traditional", link-oriented broadcast tree, for example, the minimum-costspanning tree. From such a starting point, they look for opportunities to increase transmission power levels of certain trees such that the additionally covered nodes can stop transmitting and the resulting, modified tree consumes less energy than the original one that did not take into account the wireless multicast advantage. The algorithm proceeds in rounds, building up the final broadcast tree along

the lines of the preliminary, MST-based tree. The first round starts from the source node. In each round, nodes are added as covered nodes to the final tree and one node is selected as a transmitting tree; some nodes that were transmitting in the preliminary tree can be demoted to normal, nontransmitting nodes (they are "excluded"). The decision of which node to select as a transmitting node and at which power to operate it depends on the possible gain that such a decision incurs. The gain is computed with respect to the possibility to exclude certain nodes, thus saving energy: For a potential transmission candidate u and a to-be-excluded node v, the transmission power for u is computed as that power required to cover both v itself and all of its children. This requires more energy on behalf of u but saves energy in v and also in potentially other transmitting nodes (which will then also be excluded). Eventually, the node that results in the biggest gain is chosen as transmission node. This algorithm is repeated until all nodes are covered. Cagalj et al. [113] also describe a distributed version of this heuristic. It is also based on a preliminary, distributed determination of a minimum spanning tree. The final tree's computation is again divided into rounds, which are moreover structured into three phases. In these phases, information about neighboring nodes and the structure of the transmission chain from the source to a given node is used to modify the original tree – the details are somewhat involved and the reader is referred to reference [113]. Marks et al. [541] follow similar lines as Cagalj et al. [113] in that they also are interested in transforming existing trees to obtain ones with better energy consumption. Marks et al., however, start from stochastically generated trees (rather than MIPs) and their search operations are different as well. They base their heuristics on the viability lemma.

## A distributed, position-based approach to the wireless multicast advantage

Another construction [120] for energy-efficient broadcasts does not start from the minimum spanning tree but starts from the Relative Neighborhood Graph (RNG) (already introduced in Section 10.2.3). The RNG is defined as that subgraph of the original graph where two nodes u and v are only connected if there is no third node w that would be closer to each of u and v. While the RNG typically has a higher degree than the MST, it is easily constructed using locally available information about positions of neighboring nodes or distances between neighbors' neighbors. The RNG as such is a unicast graph as it does not take into account the wireless multicast advantage. To correct this, the proposed heuristic first determines, for each node v, the transmission range as the smallest range that connects v to all of its neighbors in the RNG. Then, the RNG$*$ is defined as the subgraph induced by these (local broadcast) transmission ranges. Neighbors in this graph can know about joint neighbors in the underlying RNG. A broadcast is

then performed on the RNG∗, using a “neighbor elimination” procedure. The source node s broadcasts a message using its assigned transmission range. A receiving neighbor v of the source can then decide which of its neighbors in the RNG have already been covered by the first broadcast, eliminates them from its list, and, if this list is nonempty, rebroadcasts the received message. Nodes that redundantly receive a message can still extract information about which neighboring nodes have already been supplied with the message and can eliminate them from their neighbor list. To take advantage of this information from other nodes’ forwarding, the rebroadcasting of a received message is randomly delayed for some time. Reference [120] describes the procedure in more detail. It also compares this schemes with others, in particular, with BIP. The authors show that this RNG-based scheme is advantageous, especially in dense networks.

### 4.4.3 Shared, core-based tree protocol

The challenge in core-based tree multicast protocols lies in finding a good core node. Once this node is determined, essentially the problem can be reformulated as a source-based tree protocol with the core node as the source (although better optimization is possible). An overview of such protocols and a performance comparison can be found in reference [474]; further examples are contained in references [147, 917]. To give an idea how such a core search could work, the “merge point formation” from reference [564] is briefly described here. Assume there are a few sinks in a network to which data shall be distributed via a core-based multicast tree. A “merge point” for this tree is to be found. To do so, each sink broadcasts advertisement messages indicating its presence; each node in the network collects these advertisements along with sink identifier and number of hops that the advertisement took. After a certain time, each node that has received more than one sink advertisement broadcasts merge advertisement messages. These messages are only forwarded by nodes that have heard from fewer sinks or whose cumulative distance to all sinks is larger. Eventually, only one node (depending on network topology and timer values) will not have heard other merge advertisements overruling its own and will declare itself the merge point. This leader election result is then spread through the network. When dropping the assumption that the broadcast or multicast tree is actually rooted in and constructed from the actual source of the broadcast, optimality is evidently sacrificed. On the other hand, having to maintain only a single tree for all sources instead of one tree per source is attractive as well. The question is thus how big the performance penalty is that has to be paid by using only a single (e.g. core-based) tree for all multicast operations. This question is answered by Papadimitriou and Georgiadis [613]. They show that, given a tree and exploiting the wireless multicast advantage, the difference in total power consumption between any two sources on this tree

is at most a factor of two. Moreover, they give a construction for a single broadcasting tree (SBT) such that the total power using this tree from any arbitrary source s is at most 2H (|V | − 1) times that of the optimal tree for the given source (where H (·) is the harmonic function

## 4.4.4 Mesh-based protocols

To overcome scalability (in number of sources) and robustness issues of tree-based protocols, a structure with higher connectivity is necessary that can connect multiple sources to their destinations. The first proposal in this sense was the Core-Assisted Mesh Protocol (CAMP) [283]. The mesh, a subgraph of the original graph, has to contain all sources and destinations and provide at least one path from each source to each destination. The redundancy of the mesh can actually enable shorter paths in the mesh than would be possible in a core-based tree; it is, however, up to the forwarding procedure to actually be able to exploit these shortcuts without resorting to flooding the entire mesh with data. Reference [917] summarizes several mesh-based protocols, mostly from an ad hoc background, and compares them against tree-based protocols. The energy consumption of mesh-based protocols is, on the average, (f + 1)/2 times larger than that of tree-based multicast protocols (where $f \leq 2$ is the node connectivity, analyzed for nodes laid out in a grid pattern). Hence, their use in sensor networks requires careful deliberation.

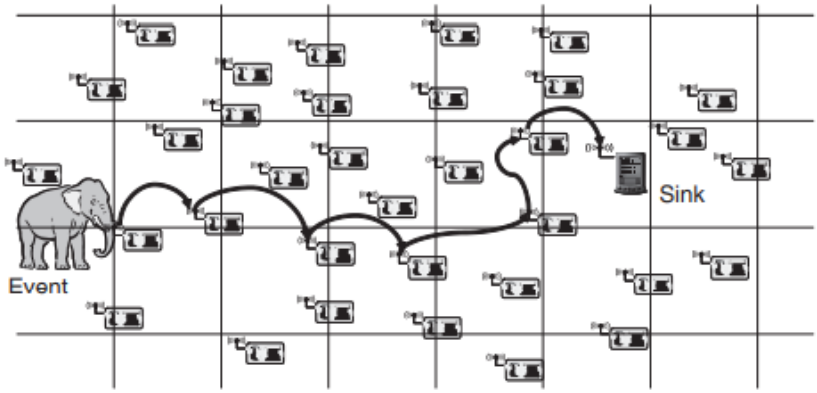



**Figure 11.10** Two-tier data dissemination as an example of a mesh-based protocol

As an example of a (in a sense) mesh-based protocol, let us consider the Two-Tier Data Dissemination (TTDD) described by Ye et al. [911] (Figure 11.10). Here, source nodes detect certain events that have to be forwarded to several mobile sinks. To do so, each source node, after detecting an event, starts to build a regularly spaced mesh of "dissemination points", resulting in rectangular cells of known size l (geographical unicast routing is assumed to

be available). Sinkson the other hand, flood their queries in their local vicinity with a bounded radius to live (significantly more efficient than unrestricted flooding). This radius is selected large enough so that at least one dissemination point must be found within it. In this sense, two tiers of forwarding functionality are deployed – global multicasting on a mesh and local query broadcasting. Once a dissemination node receives a query, it forwards it upstream to the source of the data via the multicast mesh; it does not repeat that for multiple queries, reducing the query overhead. Dissemination nodes keep on forwarding (aggregated) queries toward the source and store information from where the queries arrived. This information is used to then send the actual data back to the sinks, implicitly setting up a multicast tree on top of the multicast mesh

## 4.4.5 Further reading on broadcast and multicast

**Gossiping for multicast** Gossiping can be used for multicast and broadcast as well. Chandra et al. [131], for example, describe a scheme where gossiping is used to improve the reliability of multicasting.

**Directed antennas for multicasting** When directed antennas are available, robustness and trafficcarrying capacity of a network should improve considerably (given a fixed amount of available power). Wieselthier et al. [875], for example, extend their BIP/MIP protocols to take advantage of such antennas, assuming that directed antennas can be used to concentrate power to the neighbors in the multicast tree and thus reduce power consumption. This results in considerable improvements in network lifetime and total delivered traffic.

**Relationship to topology control**

There is clearly a tight relationship to topology-control protocols. A difference is often the source-based nature of multicast/broadcast protocols, whereas topology-control protocols try to optimize the network as a whole. There are also some papers that explicitly combine these aspects

**Optimal solutions by linear programming**

An optimization technique that has not been discussed so far is linear programming where the problem is cast as a linear equation that is to be maximized (or minimized) subject to certain constraints. While this modeling approach obviously does not change the complexity of the problem (integer linear programming is NP-hard), it can often lead to good approximations of the optimal solutions by standard relaxation techniques. Das et al. [191] describe three different integer linear programming problems to capture the minimum power broadcast problem. The solution to these problems can act as benchmarks for practical algorithms.

**Optimal solution for tree networks** How to optimally collect and distribute data in a tree network

**Time to complete a multicast** So far, mostly the energy required for a multicast or broadcast has been considered. But the time necessary to do so can also be important. Reference [261] gives an example of how this problem can be approached

**Data placement** A further variant of Steiner tree approximations is investigated by Bhattarcharya et al. [77]. They are interested in placing data caches in a network, which boils down to choosing good Steiner points.

**Cooperative multihop broadcast**In a broadcast flood, a node can overhear the same packet transmitted by several different senders. Using advanced signal processing, a node might be able to reconstruct the correct packet even if each individual reception is erroneous. Such cooperative broadcasting is described in reference [540].

# UNIT 5
# NETWORK LAYER SOLUTIONS

## 5.1 QoS Routing Protocol

QoS routing protocols search for routes with sufficient resources in order to satisfy the QoS requirements of a flow. The information regarding the availability of resources is managed by a resource management module which assists the QoS routing protocol in its search for QoS feasible paths. The QoS routing protocol should find paths that consume minimum resources. The QOSmetrics can be classified as additive metrics, concave metrics, and multiplicative metrics

An additive metric Am is defined as , where Li(m) is the value of metric m over link Li and Li × P. The hop length of path P is h. A concave metric represents the minimum value over a path P and is formally defined as C m = min(Li(m)), Li(m) ε P. A multiplicative metric represents the product of QoS metric values and is defined as , Li(m) ε P. To find a QoS feasible path for a concave metric, the available resource on each link should be at least equal to the required value of the metric. Bandwidth is a concave metric, while cost, delay, and delay jitter are additive metrics. The reliability or availability of a link, based on some criteria such as link-break-probability, is a multiplicative metric. Finding an optimal path with multiple constraints may be an NP-complete problem if it involves two or more additive metrics. For example, finding a delay-constrained least-cost path is an NP-complete problem.

To assist QoS routing, the topology information can be maintained at the nodes of ad hoc wireless networks. The topology information needs to be refreshed frequently by sending link state update messages, which consume precious network resources such as bandwidth and battery power. Otherwise, the dynamically varying network topology may cause the topology information to become imprecise. This trade-off affects the performance of the QoS routing protocol. As path breaks occur frequently in ad hoc wireless networks, compared to wired networks where a link goes down very rarely, the path satisfying the QoS requirements needs to be recomputed every time the current path gets broken. The QoS routing protocol should respond quickly in case of path breaks and recompute the broken path or bypass the broken link without degrading the level of QoS. In the literature, numerous routing protocols have been proposed for finding QoS paths. In the following sections, some of these QoS routing protocols are described

## NODE AND NETWORK MANAGEMENT

# Power Management

The power consumption of a wireless sensor network (WSN) is of crucial concern because of the scarcity of energy. Whereas energy is a scarce resource in every wireless device, the problem in WSNs is amplified for the following reasons:

1. Compared to the complexity of the task they carry out – namely, sensing, processing, self-managing, and communication – the nodes are very small in size to accommodate high-capacity power supplies.
2. Ideally, a WSN consists of a large number of nodes. This makes manually changing, replacing or recharging batteries almost impossible.
3. While the research community is investigating the contribution of renewable energy and self-recharging mechanisms, the size of nodes is still a constraining factor.
4. The failure of a few nodes may cause the entire network to fragment prematurely.

The problem of power consumption can be approached from two angles. One is to develop energy-efficient communication protocols (self-organization, medium access, and routing protocols) that take the peculiarities ofWSNs into account. The other is to identify activities in the networks that are both wasteful and unnecessary and mitigate their impact. Wasteful and unnecessary activities can be described as local (limited to a node) or global (having a scope network-wide). In either case, these activities can be further considered as accidental side-effects or results of non optimal software and hardware implementations (configurations). For example, observations based on field deployment reveal that some nodes exhausted their batteries prematurely because of unexpected

overhearing of traffic that caused the communication subsystem to become operational for a longer time than originally intended (Jiang *et al.* 2007). Similarly, some nodes exhausted their batteries prematurely because they aimlessly attempted to establish links with a network that had become no longer accessible to them.

Most inefficient activities are, however, results of non optimal configurations in hardware and software components. For example, a considerable amount of energy is wasted by an idle processing or a communication subsystem. A radio that aimlessly senses the media or overhears while neighboring nodes communicate with each other consumes a significant amount of power.

\A dynamic power management (DPM) strategy ensures that power is consumed economically.

The strategy can have a local or global scope, or both. A local DPM strategy aims to minimize the power consumption of individual nodes by providing each subsystem with the amount of power that is sufficient to carry out a task at hand. When there is no task to be processed, the DPM strategy forces some of the subsystems to operate at the most economical power mode or puts them into a sleeping mode. A global DPM strategy attempts to minimize the power consumption of the overall network by defining a network-wide sleeping state. There are different ways to achieve this goal. One way is to let individual nodes define their own sleeping schedules and share these schedules with their neighbors to enable a coordinated sensing and an efficient inter node communication. This is called synchronous sleeping. The problem with this approach is that neighbors need to synchronize time as well as schedules and the process is energy intensive. Another way is to let individual nodes keep their sleeping schedules to themselves; and a node that initiates a communication should send a preamble until it receives an acknowledgment from its receiving partner. This approach is known as asynchronous sleeping schedule and avoids the needs to synchronize schedules. But it can have a latency side-effect on data transmission. In both approaches, individual nodes wake up periodically to determine whether there is a node that wishes to communicate with them and to process tasks waiting in a queue.

## Local Power Management Aspects

The first step toward developing a local power management strategy is the understanding of how power is consumed by the different subsystems of a wireless sensor node. This knowledge enables wasteful activities to be avoided and to frugally budget power. Furthermore, it enables one to estimate the overall power dissipation rate in a node and how this rate affects the lifetime of the entire network.

### *5.1.1 Processor Subsystem*

Most existing processing subsystems employ microcontrollers, notably Intel's Strong ARM and Atmel's AVR. These microcontrollers can be configured to

operate at various power modes. For example, the ATmega128L microcontroller has six different power modes: idle, ADC noise reduction, power save, power down, standby, and extended standby. The *idle* mode stops the CPU while allowing the SRAM, Timer/Counters, SPI port, and interrupt system to continue functioning. The *power down* mode saves the registers' content while freezing the oscillator and disabling all other chip functions until the next interrupt or Hardware Reset. In the *power save mode*, the asynchronous timer continues to run, allowing the user to maintain a timer base while the remaining components of the device enter into a sleep mode. The *ADC noise reduction* mode stops the CPU and all I/O modules, except the asynchronous timer and the ADC. The aim is to minimize switching noise during ADC conversions. In *standby* mode, a crystal/resonator oscillator runs while the remaining hardware components enter into a sleep mode. This allows very fast start-up combined with low power consumption. In *extended standby* mode, both the main oscillator and the asynchronous timer continue to operate. Additional to the above configurations, the processing subsystem can operate with different supply voltages and clock frequencies.

While operating the processor subsystem at various power modes is energy-efficient, transiting from one power mode to another also has its own power and latency cost. This cost has to be considered before a decision for a particular operation of power mode is made.

### *5.1.2 Communication Subsystem*

The power consumption of the communication subsystem can be influenced by several aspects, including the modulation type and index, the transmitter's power amplifier and antenna efficiency, the transmission range and rate, and the sensitivity of the receiver. Some of these aspects can be dynamically reconfigured. Moreover, the communication subsystem itself can activate or turn off the transmitter and the receiver, or both. Because of the presence of a large number of active components in the communication subsystem (amplifiers and oscillators), a significant amount of quiescent current flows even if the device is idle. Determining the most efficient active state operational mode is not a simple decision. For example, the power consumption of a transmitter may not necessarily be reduced by simply reducing the transmission rate or the transmission power. The reason is that there is a tradeoff between the useful power required for data transmission and the power dissipated in the form of heat at the power amplifier. Usually, the dissipation power (heat energy) increases as the transmission power decreases. In fact most commercially available transmitters operate efficiently at one or two transmission power levels. Below a certain level, the efficiency of the power amplifier falls drastically. In some cheap transceivers, even when at the maximum transmission power mode, more than 60% of the supply DC power is dissipated in the form of useless heat.

The amplifier efficiency is defined as the ratio of the transmission power to the DC input power consumed by the amplifier. An additional challenge to the power issue is the time needed by the communication subsystem to transit from an idle, or standby mode into an active mode. The transmission introduces latency and consumes power. For example, the Chipcon's transceiver frequency synthesizer's phase locked loop (PLL) requires 192 μs to lock up..

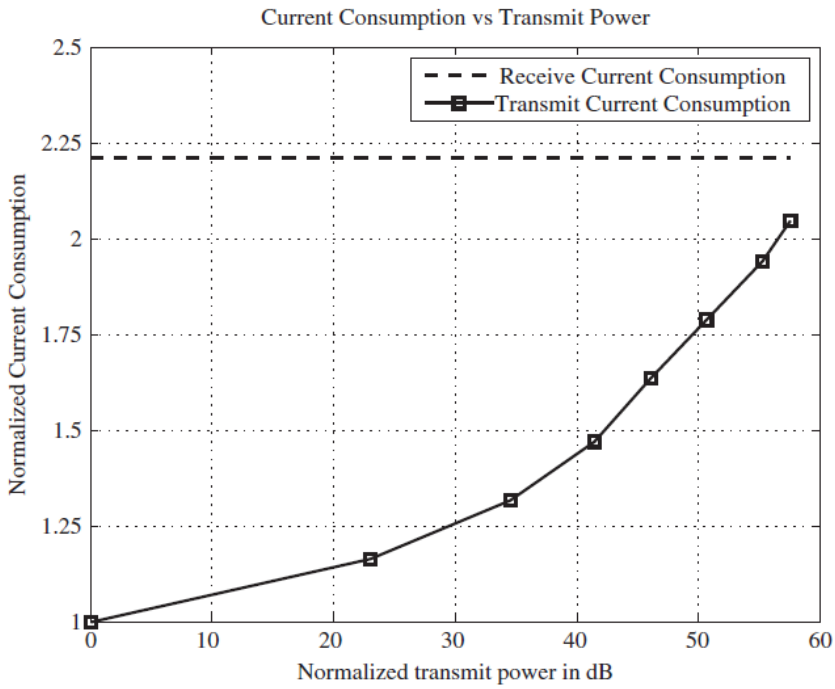



**Figure 8.1** Relation between transmit power and current consumption in Chipcon CC2420 transceiver.

### *5.1.3 Bus Frequency and RAM Timing*

The processor subsystem consumes power when it interacts with the other subsystems via the internal high-speed buses. The specific amount depends on the frequency and bandwidth of the communication. These two parameters can be optimally configured depending on the interaction type, but bus protocol timings are usually optimized for particular bus frequencies.
Moreover, bus controller drivers require to be notified when bus frequencies change to ensure optimal performance.

### *5.1.4 Active Memory*

The active memory is made up of electrical cells which are arranged in rows and columns, each row being a single memory bank. The cells have to be recharged periodically in order to store data. The refresh rate or refresh interval is a measure of the number of rows that must be refreshed. A low refresh

interval corresponds to a low clock frequency that must elapse before a refreshing operation takes place. On the contrary, a higher refresh interval corresponds to a high clock frequency that must elapse before a refresh operation takes place. Consider two typical values: 2K and 4K. The lower refresh interval refreshes more cells at a low interval and completes the process faster, thus it consumes more power than the 4K refresh rate. The 4K refresh rate refreshes less cells at a slower pace, but it consumes less power.

A memory unit can be configured to operate in one of the following power modes: temperature-compensated self-refresh mode, partial array self-refresh mode, or power down mode. The standard refresh rate of a memory unit can be adjusted according to its ambient temperature. For this reason, some commercially available dynamic RAMs (DRAMs) already integrate temperature sensors. Apart from this, the self-refresh rate can be increased if the entire memory array is not needed to store data. The refresh operation can be limited to the portion of the memory array in which data will be stored. This approach is known as the partial array self-refresh mode. If no actual data storage is required, the supply voltage of most or the entire on-board memory array can be switched off.

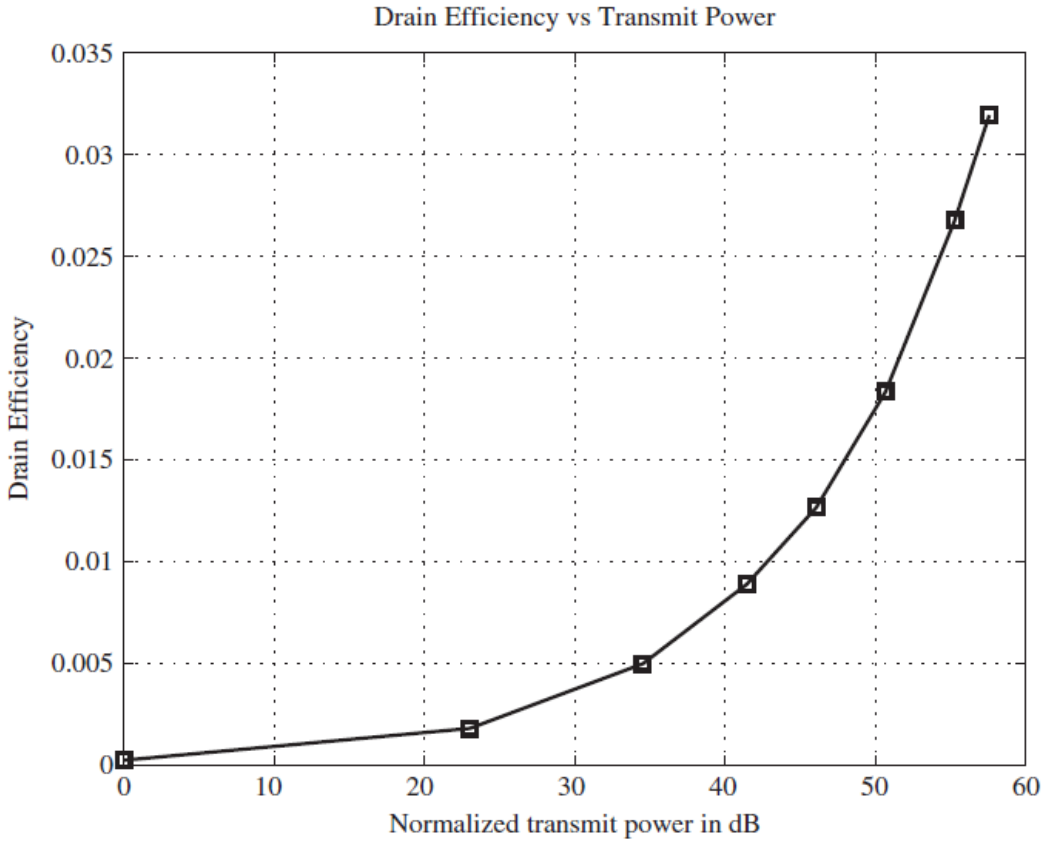



**Figure 8.2** Amplifier efficiency in the Chipcon CC2420 transceiver.

The RAM timing is another parameter that affects the power consumption of the

memory unit. It refers to the latency associated with accessing the memory unit. Before a processor subsystem accesses a particular cell in a memory, it should first determine the particular row or bank and then activate it with a row access strobe (RAS) signal. Once a row is activated, it can be accessed until the data is exhausted. The time required to activate a row in a memory is *t*RAS, which is relatively small but could impact the system's stability if set incorrectly.

**Table 8.2** Parameters of RAM timing

| Parameter | Description |
|---|---|
| RAS | Row Address Strobe or Row Address Select |
| CAS | Column Address Strobe or Column Address Select |
| $t_{RAS}$ | A time delay between the precharge and activation of a row |
| $t_{RCD}$ | The time required between RAS and CAS access |
| $t_{CL}$ | CAS latency |
| $t_{RP}$ | The time required to switch from one row to the next row |
| $t_{CLK}$ | The duration of a clock cycle |
| Command rate | The delay between Chip Select (CS) |
| Latency | The total time required before data can be written to or read from memory |

A memory cell is activated through a column access strobe (CAS). The delays between the activation of a row as well as a cell and the writing of data into or reading of data from the cell is given as *t*RCD. This time can be short or long, depending on how the memory cell is accessed. If it is accessed sequentially, it is insignificant. If, on the other hand, the memory is accessed in a random fashion, the current active row must first be deactivated before a new row is activated, in which case, *t*RCD can cause significant latency.The delay between the CAS signal and the availability of valid data on the data pins is called *CAS latency*. Low CAS latency means high performance but also high power consumption. The time required to terminate one row access and begin the next row access is *t*RP. In conjunction with *t*RCD, the time (or clock cycles) required to switch banks (rows) and select the next cell for reading, writing, or refreshing is expressed as *t*RP + *t*RCD. The duration of time required between the active and precharge commands is called *t*RAS. It is a measure of how long the processor must wait before the next memory access can begin.

Table 8.2 summarizes the quantities that express RAM timing.

When a RAM is accessed by clocked logic, the times are generally rounded up to the nearest clock cycle. For example, when accessed by a 100-MHz processor (with 10 ns clock duration), a 50-ns SDRAM can perform the first read in 5 clock cycles and additional reads within the same page every 2 clock cycles. This is generally described as "5 – 2 – 2 – 2" timing.

### *5.1.5 Power Subsystem*

The power subsystem supplies power to all the other subsystems. It consists of the battery and the DC–DC converter. In some cases, it may consist of additional components such as a voltage regulator. The DC–DC converter is responsible for providing the right amount of supply voltage to each individual hardware component by transforming the main DC supply voltage into a suitable level. The transformation can be a step-down (buck), a step-up (boost), or an inversion (flyback) process, depending on the requirements of the individual subsystem. Unfortunately, a transformation process has its own power consumption and may be inefficient.

#### 5.1.5.1 Battery

A wireless sensor node is powered by exhaustible batteries. Several factors affect the quality of these batteries, but the main factor is cost. In a large-scale deployment, the cost of hundreds and thousands of batteries is a serious deployment constraint. Batteries are specified by a rated current capacity, $C$, expressed in *ampere-hour*. This quantity describes the rate at which a battery discharges without significantly affecting the prescribed supply voltage (or potential difference). Practically, as the discharge rate increases, the rated capacity decreases.

Most portable batteries are rated at 1$C$. Thismeans a 1000 mAh battery provides 1000mA for 1 hour, if it is discharged at a rate of 1$C$. Ideally, the same battery can discharge at a rate of 0.5$C$, providing 500mA for 2 hours; and at 2$C$, 2000mA for 30 minutes and so on. 1$C$ is often referred to as a 1-hour discharge. Likewise, a 0.5$C$ would be a 2-hour and a 0.1$C$ a 10-hour discharge.

In reality, batteries perform at less than the prescribed rate. Often, the Peukert Equation is applied to quantifying the capacity offset (i.e., how long a battery lasts in reality):

$$t = \frac{C}{I^n} \qquad (8.1)$$

where $C$ is the theoretical capacity of the battery expressed in ampere-hours; $I$ is the current drawn in Ampere (A); $T$ is the time of discharge in seconds, and $n$ is the Peukert number, a constant that directly relates to the internal resistance of the battery. The value of the Peukert number indicates how well a battery performs under continuous heavy current.A value close to 1 indicates that the battery performs well; the higher the number, the more capacity is lost when the battery is discharged at high current. The Peukert number of a battery is determined empirically.

For example, for lead acid batteries, the number is typically between 1.3 and1.4. Drawing current at a rate greater than the discharge rate results in a current consumption rate higher than the rate of diffusion of the active elements in the electrolyte. If this process continues for a long time, the electrodes run out of active material even though the electrolyte has not yet exhausted its active materials. This situation can be overcome by intermittently drawing current from the battery.

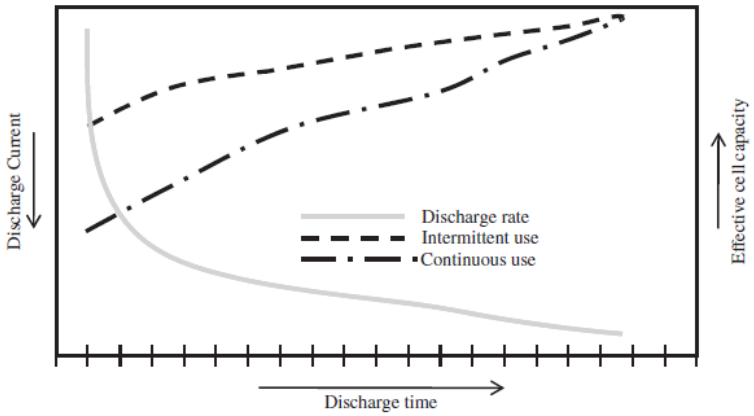



**Figure 8.3** The Peukert curve displaying the relationship between the discharging rate and the effective voltage. The $x$-axis is a time axis.

Figure 8.3 shows how the effective battery capacity can be reduced at high and continuous discharge rates. By intermittently using the battery, it is possible during quiescent periods to increase the diffusion and transport rates of active ingredients and to match up the depletion created by excessive discharge. Because of this potential for recovery, the capacity reduction can be undermined and the operating efficiency can be enhanced. This is illustrated in the figure by the middle, dash-dotted line.

### 5.1.5.2 DC–DC Converter

The DC–DC converter transforms one voltage level into another. It is the equivalent of a transformer which performs AC–AC voltage transformation. The main problem with a DC–DC converter is its conversion efficiency. A typical DC–DC converter consists of a power supply, a switching circuit, a filter circuit, and a load resistor.

Figure 8.4 illustrates the basic circuit structure of a DC–DC converter. As can be seen in the figure, the circuit consists of a single-pole, double-throw (SPDT) switch that is connected to a DC supply voltage, *Vg*. Considering the inductor, *L*, as a short circuit and the capacitor, *C*, as an open circuit for the DC supply voltage, the switch's output voltage, *Vs (t)* equals to *Vg* when the switch is in position 1 and 0 when it is in position 2. Varying the position of the switch at a frequency *fs* yields a periodically varying square wave, *vs (t)*, that has a period $Ts = 1/fs$ .

*vs (t)* can be expressed by a duty cycle *D*, which describes the fraction of time that the switch is in position 1, such that $0 \leq D \leq 1$. The output voltage of the switching circuit is displayed in Figure 8.5.
A DC–DC converter is realized by employing active switching components, such as diodes and power MOSFETs. Typically, the switching frequencies range from 1 kHz to 1MHz, depending on the speed of the semiconductor devices. Using the inverse Fourier transformation, the DC component of *vs (t)* (*Vs* ) is described as:

$$V_s = \frac{1}{T_s}\int_0^{T_s} v_s(t)\mathrm{d}t = DV_g \qquad (8.2)$$

which is the average value of *vs (t)*.

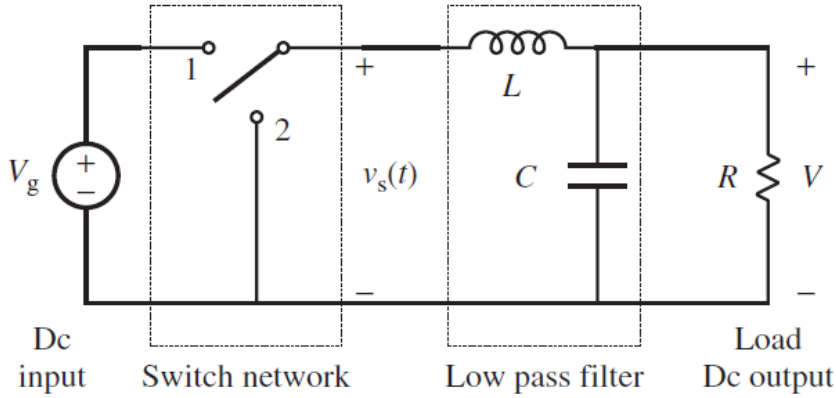



**Figure 8.4** A DC–DC converter consisting of a supply voltage, a switching circuit, a filter circuit, and a load resistance.

In other words, the integral value represents the area under the waveform of Figure 8.5 for a single period, or the height of *Vg* multiplied by the time *Ts* . It can be seen that the switching circuit reduces the DC component of the supply voltage by a factor that equals to the duty cycle, *D*. Since $0 \leq D \leq 1$ holds, $Vs \leq Vg$ holds as well. Ideally the switching circuit does not consume power. In practice, however, due to the existence of a resistive component in the switching circuit, there is power dissipation. As a result, the efficiency of a typical switching circuit is between 70 and 90%.

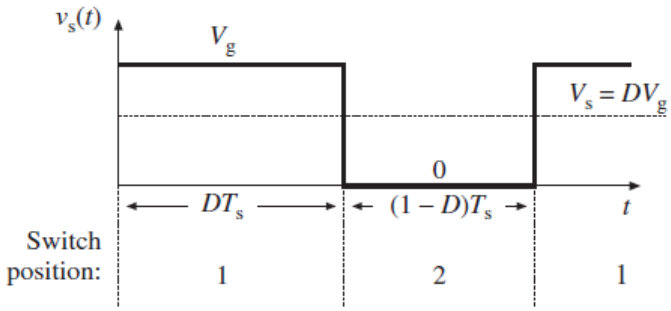



**Figure 8.5** The output voltage of a switching circuit of a DC–DC converter.

In addition to the desired DC voltage, *vs (t)* also contains undesired harmonics of the switching frequency, *fs* . These harmonics must be removed so that the converter's output voltage *v(t)* is essentially equal to the DC component *V = Vs* . For this purpose, a DC–DC converter employs a low pass filter. In Figure 8.4, a first-order *LC* low pass filter is connected to the switching circuit. The filter's cutoff frequency is given by:

$$f_c = \frac{1}{2\pi\sqrt{LC}} \qquad (8.3)$$

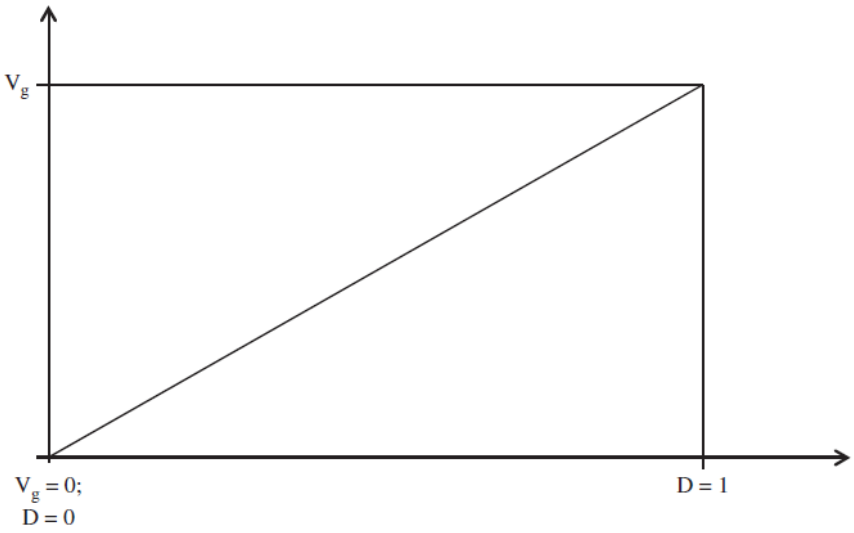



**Figure 8.6** A linear relationship between a DC supply voltage and the duty cycle of a switching circuit.

The cutoff frequency, *fc*, should be sufficiently less than the switching frequency, *fs* , so that the lowpass filter allows only the DC component of *vs (t)* to pass while effectively attenuating all the harmonic components. Once again, in an ideal filter, there is no power dissipation because the passive components (inductors and capacitors) are energy storage components. Subsequently, the DC–DC converter produces a DC output voltage whose magnitude is controlled by the duty cycle, *D*, using circuit elements that (ideally) do not dissipate power. The conversion ratio, *M(D)*, is defined as the ratio of the DC output voltage, *V* , to the DC input voltage, *Vg*, under a steady-state condition:

$$M(D) = \frac{V}{V_g} \qquad (8.4)$$

For the buck converter shown in Figure 8.4, *M(D) = D*. Figure 8.6 illustrates the linear relationship between the input DC voltage, *Vg* and the switching circuit's duty cycle, *D*.

## 5.2 Dynamic Power Management

Wireless sensor nodes can be developed by taking the aspects discussed so far into account at design time. Once the design time parameters are fixed, a dynamic power management (DPM) strategy attempts to minimize the power consumption of the system by dynamically defining the most economical operation conditions. This condition takes the requirements of the application, the topology of the network, and the task arrival rate of the different subsystems into account. Whereas there are different approaches to a DPM strategy, they can be categorized in one of the following three approaches:
1. Dynamic operation modes.
2. Dynamic scaling.
3. Energy harvesting.

### *5.2.1 Dynamic Operation Modes*

The subsystems of a wireless sensor node can be configured to operate in different power modes, depending on their present and anticipated activity. This has already been explained in the previous subsections. In general, a subcomponent can have $n$ different power modes.
If there are $x$ hardware components that can have $n$ distinct power consumption levels, a DPM strategy can define $x \times n$ different power mode configurations, $Pn$. Obviously, not all these configurations are plausible because of various constraints and system stability preconditions. Hence, the task of the DPM strategy is to select the optimal configuration that matches the activity of a wireless sensor node. There are, however, two associated challenges in selecting a specific power configuration.
1. Transition between the different power configurations costs extra power.
2. A transition has an associated delay and the potential of missing the occurrence of an interesting event.
Table 8.3 demonstrates an example DPM strategy with six different power modes: $\{P0, P1, P2, P3, P4, P5\}$. Figure 8.7 shows corresponding potential transitions between five arbitrary power modes.

**Table 8.3** Power saving configurations

| Configuration | Processor | Memory | Sensing subsystem | Communication subsystem |
|---|---|---|---|---|
| $P_0$ | Active | Active | On | Transmitting/receiving |
| $P_1$ | Active | On | On | On (transmitting) |
| $P_2$ | Idle | On | On | Receiving |
| $P_3$ | Sleep | On | On | Receiving |
| $P_4$ | Sleep | Off | On | Off |
| $P_5$ | Sleep | Off | Off | Off |

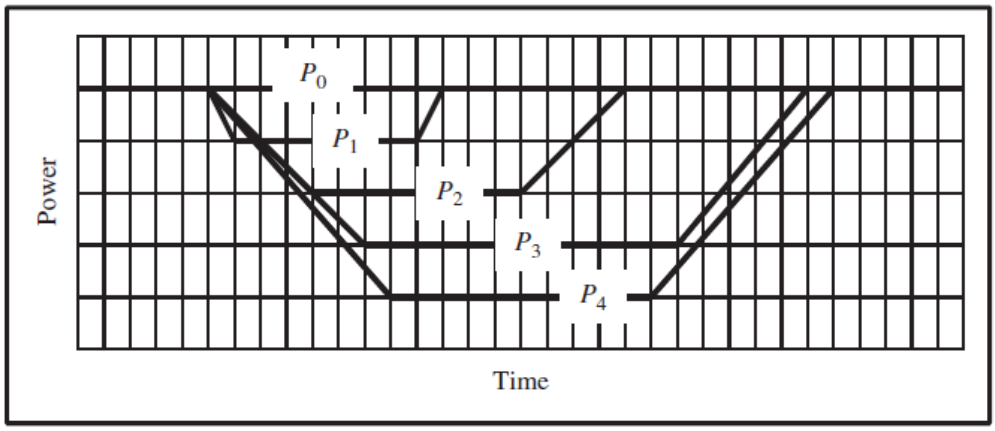



**Figure 8.7** Transition between different power modes and the associated transition costs.

The decision for a particular power mode depends on the present as well as the anticipated task in the queues of the different hardware components. A realistic estimation of future tasks enables a node to determine the time it needs to put the required components in the right power mode, so that they can process the tasks with minimum latency. By the same token, failure to realistically estimate future tasks can cause a node to miss interesting events or to delay in response. In a WSN, the events outside of the network (for example, a leak in a pipeline; a fracture in a structure; a pestilence in a farm; etc.) cannot be modeled as deterministic phenomena. Otherwise there is no need for setting up a monitoring system. Therefore, estimation of the arrival of events should be probabilistic. Knowledge of the sensing task can be useful to establish a realistic probabilistic model for estimating the arrival rate as well as the duration of events. An accurate event arrival model enables a DPM strategy to decide for the right configuration that has a long duration and minimal power consumption.

### 5.2.1.1 Transition Costs

Suppose each subsystem of a wireless sensor node operates in just two different power modes only, namely, it can be either *on* or *off* . Moreover, assume that the transition from *on* to *off* does not have an associated power cost, but the reverse transition (from *off* to *on*) has a cost in terms of both power and a time delay. These costs are justified if the power it saves in the *off* state is large enough. In other words, the amount of the *off* state power is considerably large and the duration of the *off* state is long. It is useful to quantify these costs and to set up a transition threshold.

Suppose the minimum time that a subsystem stays in an *off* state is *t*off; the power consumed during this time is *P*off; the transition time is *t*off, on; the

power consumed during the transition is *p*off, on; and the power consumed in an *on* state is *P*on. Hence:

$$P_{\text{off}} \cdot t_{\text{off}} + P_{\text{off,on}} \cdot t_{\text{off,on}} \geq P_{\text{on}} \cdot \left(t_{\text{off}} + t_{\text{off,on}}\right) \tag{8.5}$$

Therefore, $t_{\text{off}}$ is justified if (Chiasserini and Rao 2003):

$$t_{\text{off}} \geq \max\left(0, \frac{\left(P_{\text{on}} - P_{\text{off,on}}\right) \cdot t_{\text{off,on}}}{P_{\text{on}} - P_{\text{off}}}\right) \tag{8.6}$$

Equations (8.5) and (8.6) can easily be generalized to describe a subsystem with $n$ distinct operational power modes, in which case a transition from any state $i$ into $j$ is described as $t_{i,j}$. Hence, the transition is justified if Equation (8.7) is satisfied.

$$t_j \geq \max\left(0, \frac{\left(P_i - P_{j,k}\right) \cdot t_{i,j}}{P_i - P_j}\right) \tag{8.7}$$

where *tj* is the duration of the subsystem in state *j* .
The equations above assume that the transition cost from a higher power mode (on) to a lower power mode (off) is negligible. If this is not the case, the energy that can be saved through a power transition (from state *i* to state *j* , *E*saved,*j* ) is expressed as:

$$E_{\text{saved},j} = P_i \cdot \left(t_j + t_{i,j} + t_{j,i}\right) - \left(P_{i,j} \cdot t_{i,j} + p_{j,i} \cdot t_{j,i} + p_j \cdot t_j\right) \tag{8.8}$$

If the transition from state $i$ to state $j$ costs the same amount of power and time delay as the transition from state $j$ to state $i$ (a symmetric transition cost), Equation (8.8) can be expressed as:

$$E_{\text{saved},j} = P_i \cdot \left(t_j + t_{i,j} + t_{j,i}\right) - \left(\frac{P_i + P_j}{2}\right)\left(t_{i,j} + t_{j,i}\right) - \left(P_i - P_j\right) \cdot t_j \tag{8.9}$$

Obviously, the transition is justified if *E*saved,*j* >0. This can be achieved in three different ways, namely, by:
1. increasing the gap between *Pi* and *Pj* ;
2. increasing the duration of state *j*, (*tj* ); and
3. decreasing the transition times, particularly, *tj,i* .

### *5.2.2 Dynamic Scaling*

Dynamic voltage scaling (DVS) and dynamic frequency scaling (DFS) are complementary to the approach discussed in Section 5.2.1. These two approaches aim to adapt the performance of the processor core (as well as the memory unit and the communication buses) when it is in the active state. In most cases, the tasks scheduled to be carried out by the processor core do not require its peak performance. Rather, some tasks are completed ahead

of their deadline and the processor enters into a low-leakage idle mode for the remaining time. Figure 8.8 shows a subsystem processing at peak performance. Even though the two tasks are completed ahead of their schedule, the processor still runs at peak frequency and supply voltage, which is wasteful.
Figure 8.9 displays the application of dynamic frequency and voltage scaling in which the performance of the processing subsystem is adapted (reduced) according to the criticality of the tasks it processes. As can be seen, each task is stretched to its planned schedule while the supply voltage and the frequency of operation are reduced.

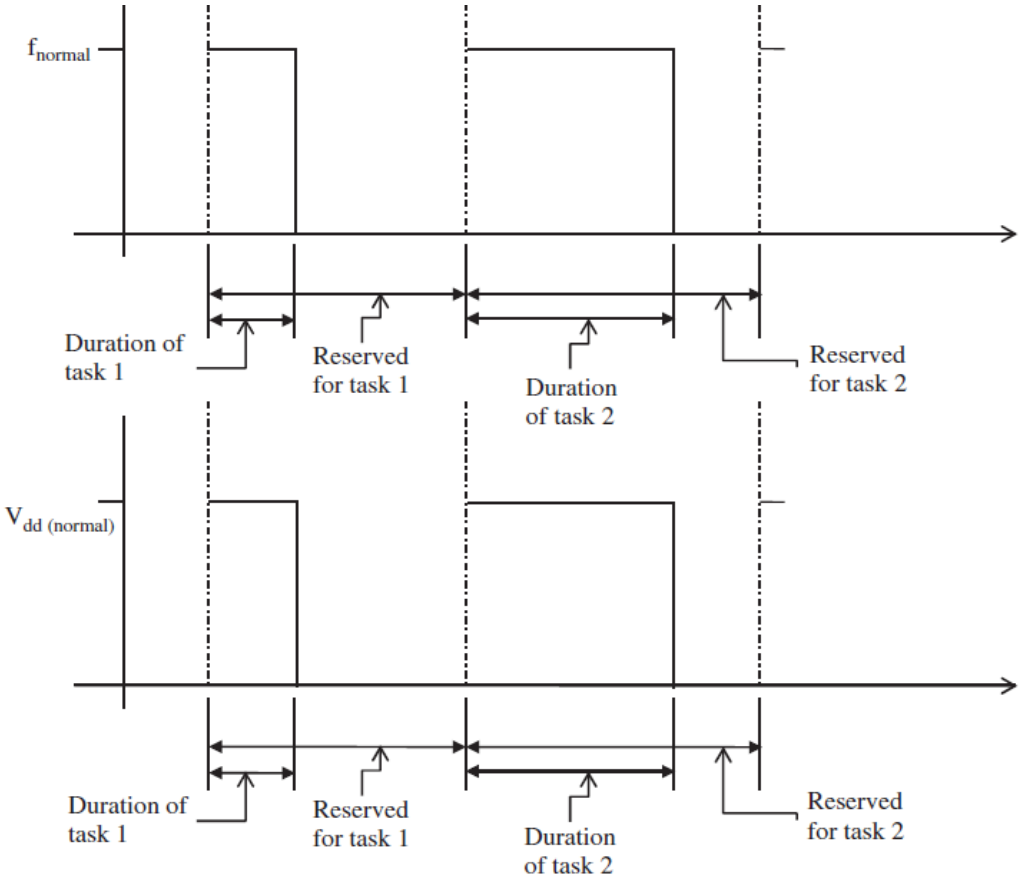



**Figure 8.8** A processor subsystem operating at its peak performance.

The basic building blocks of the processor subsystem are transistors. Depending on their operation regions (namely, cut-off, linear, and saturation), transistors are classified into analog and digital (switching) transistors. An analog transistor (amplifier) operates in the linear amplification region and there is a

linear relationship between the input and output of the transistor. This is expressed as:

$$v_{out} = \frac{A}{1 - AB} v_{in} \qquad (8.10)$$

where $A$ is the open loop gain of the amplifier and $B$ is a term that determines the portion of the output that should be fed back to the input in order to stabilize the amplifier

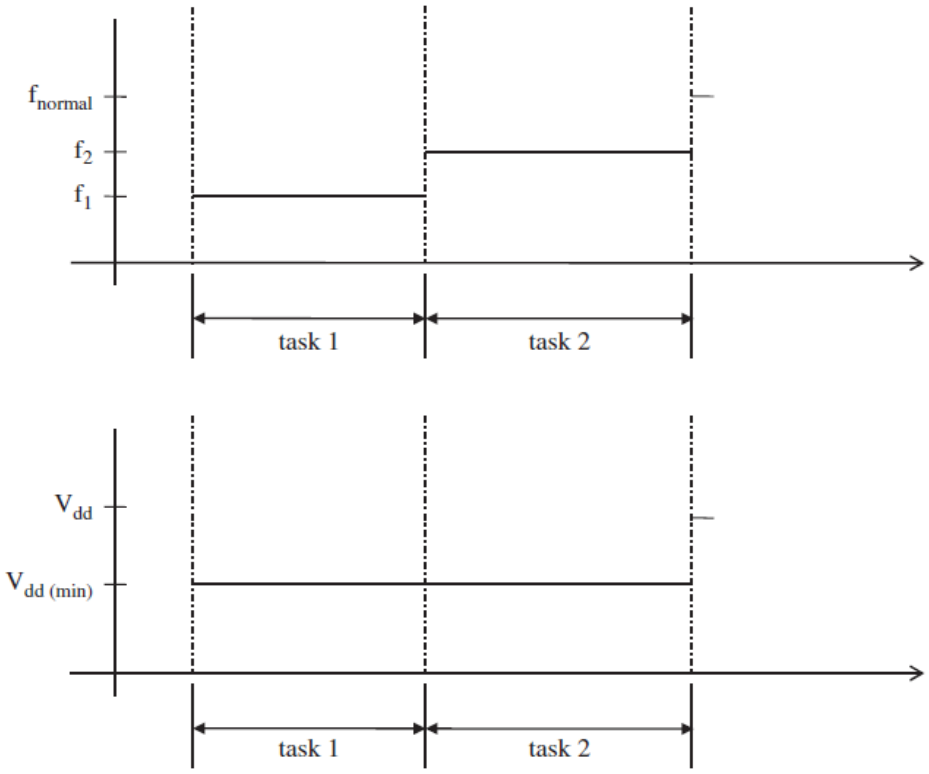



**Figure 8.9** Application of dynamic voltage and frequency scaling.

A switching transistor, on the contrary, operates in either the cutoff or the saturation region, making the relationship between the input and the output voltage nonlinear. That is how the *zeros* and *ones* of a digital system are generated, represented or processed. The transition duration from the cutoff to the saturation region determines how good a transistor is as a switching element. In an ideal switching transistor, the transition takes place in no time. In practical transistors, however, the duration is greater than zero. The quality of the processor depends on, among other things, the switching time. The switching in turn depends on many factors, one of them being the cumulative capacitance effect created in turn between the three joints of the transistors.

Figure 8.10 displays a typical NAND gate made up of CMOS transistors.

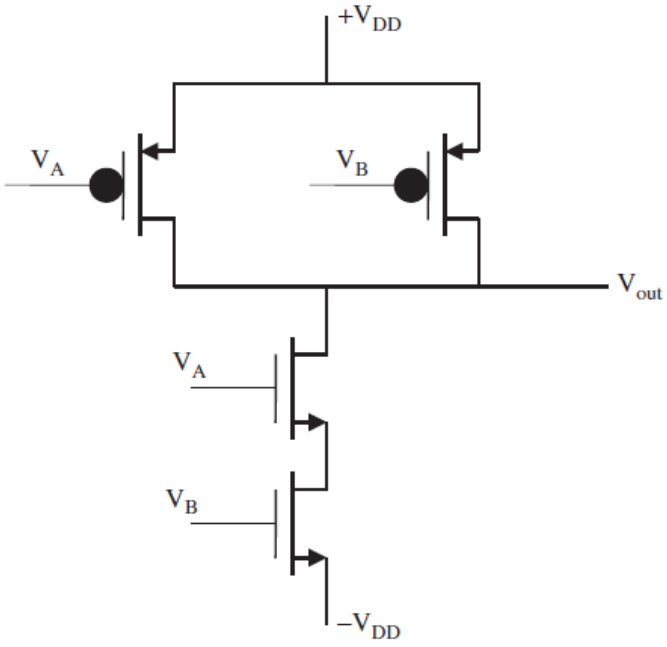



**Figure 8.10** A schematic diagram of a NAND gate based on CMOS technology.

Recall that a capacitor is created by two conductors that are separated by a dielectric material and there is a potential difference between the two conductors. The capacitance of a capacitor is proportional to the cross-sectional area of the conductors and inversely proportional to the separating distance..
In a switching transistor, at a very high operating frequency, a capacitance is created at the contact points of the source, gate, and drain, affecting the transistor's switching response.
The switching time can be approximated by the following equation:

$$t_{\text{delay}} = \frac{C_s \cdot V_{dd}}{I_{d_{\text{sat}}}} \qquad (8.11)$$

where *Cs* is the source capacitance, *Vdd* is the biasing voltage of the drain, and *Id*sat is the saturation drain current. Switching costs energy and the magnitude of the energy depends on many factors, among which are the operating frequency and the biasing voltage. Sinha and Chandrakasan (2001) provide a first-order approximation that can be expressed as:

$$E(r) = CV_0 2T_s f_{\text{ref}} r \left[ \frac{V_t}{V_0} + \frac{r}{2} + \sqrt{r\frac{V_t}{V_0} + \left(\frac{r}{2}\right)^2} \right] \qquad (8.12)$$

where, *C* is the average switching capacitance per cycle; *Ts* is the sampling period; *f*ref is the operating frequency at *V*ref; *r* is the normalized processing rate ($r = f/f\text{ref}$); and $V0 = (V\text{ref} - Vt)2/V\text{ref}$ with *Vt* being the threshold voltage. From Equation (8.12), it can be deduced that reducing the operating frequency linearly reduces the energy cost, whereas reducing the biasing voltage reduces

the energy cost quadratically. However, these two quantities cannot be reduced beyond a certain limit. For example, the minimum operating voltage for a CMOS logic to function properly was first derived by Swanson and Meindl (1972) and is expressed as:

$$V_{dd,\text{limit}} = 2 \cdot \frac{kT}{q} \cdot \left[1 + \frac{C_{fs}}{C_{ox} + C_d}\right] \cdot \ln\left(1 + \frac{C_d}{C_{ox}}\right) \qquad (8.13)$$

where *Cf s* is the surface state capacitance per unit area; *Cox* is the gate-oxide capacitance per unit area; and *Cd* is the channel depletion region capacitance per unit area. For a CMOS logic such as shown in Figure 8.10, Equation (8.13) yields, *Vdd,*limit = 48mV at 300K. Finding the optimal voltage limit requires a tradeoff between the switching energy cost and the associated delay.

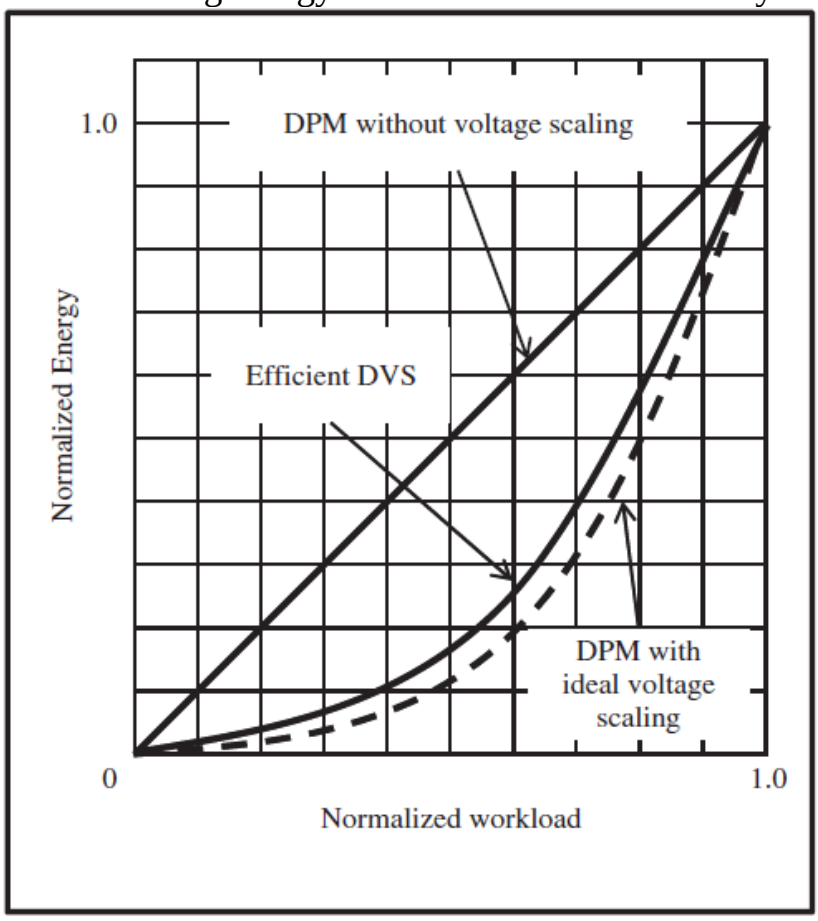



**Figure 8.11** Application of dynamic voltage scaling based on workload estimation (Sinha and Chandrakasan 2001).

### *5.2.3 Task Scheduling*

In a dynamic voltage and frequency scaling, the DPM strategy aims to autonomously determine the magnitude of the biasing voltage (*Vdd*) and the clock frequency of the processing subsystem.

The decision for a particular voltage or frequency is based on several factors,

including the application latency requirement and the task arrival rate. Ideally, these two parameters are adjusted so that a task is completed “just in time”. This way, the processor does not remain idle and consume power unnecessarily. Practically, however, since the processor’s workload cannot be known a priori, the estimation contains error and, as a result, idle cycles cannot be completely avoided. Comparison between an ideal and real dynamic voltage scaling strategies is shown in Figure 8.11.

## 5.3 Conceptual Architecture

A conceptual architecture for enabling a DPM strategy in a wireless sensor node should address three essential concerns:
1. In attempting to optimize power consumption, how much is the extra workload that should be produced by the DPM itself?
2. Should the DPM be a centralized or a distributed strategy?
3. If it is a centralized approach, which of the subcomponents should be responsible for the task?

A typical DPM strategy monitors the activities of each subsystem and makes decisions concerning the most suitable power configuration that optimizes the overall power consumption. This decision should reflect the application requirements, nevertheless. Since this process consumes a certain amount of power, it can be justified if the power that is saved as a result is significantly large. An accurate DPM strategy requires bench marking to estimate the task arrival and processing rate.

The decision whether a DPM strategy should be central or distributed depends on several factors. One advantage of a centralized approach is that it is easier to achieve a global view of the power consumption of a node and to implement a comprehensible adaptation strategy. On the other hand, a global strategy can add a computational overhead on the subsystem that does the management. A distributed approach scales well by authorizing individual subsystems to carry out local power management strategies. The problem with this approach is that local strategies may contradict with global goals. Given the relative simplicity of a wireless sensor node and the quantifiable tasks that should be processed, most existing power management strategies advocate a centralized solution.

In case of a centralized approach, the main question is which of the subsystems is responsible for handling the task – the processor subsystem or the power subsystem. Intuitively, the power subsystem should handle the management task, since it has complete information about the energy reserve of the node and

the power budget of each subsystem. However, it requires vital information, such as the task arrival rate and priority of individual tasks, from the processing subsystems. Moreover, it needs to have some computational capability. Presently available power subsystems do not have these characteristics.
Most existing architectures for a wireless sensor node place the processor subsystem at the center and all the other subsystems communicate with each other through it. Moreover, the operating system (runtime environment) runs on the processing subsystem, managing, prioritizing, and scheduling tasks. Subsequently, the processing subsystem can have a more comprehensive knowledge about the activities of all the other subsystems, and these characteristics make the processing subsystem the appropriate place for executing a DPM.

### *5.3.1 Architectural Overview*

Though the aim of a DPM strategy is to optimize the power consumption of a node, it should not affect the system's stability. Furthermore, the application requirements in terms of the quality of sensed data and latency should be satisfied. Fortunately, in most realistic situations, a WSN is deployed for a specific task. That task does not change, or changes only gradually. Therefore, the designer of a DPM has at his or her disposal the architecture of the wireless sensor node, the application requirements, and the network topology to devise a suitable strategy. The design space is illustrated in Figure 8.12.
The system's hardware architecture is the basis for defining multiple operational power modes and the possible transitions between them. A local power management strategy then defines rules to describe the behavior of the power mode transition according to a change in the activity of the node or based on a request from a global (network-wide) power management scheme, or from the application.
.

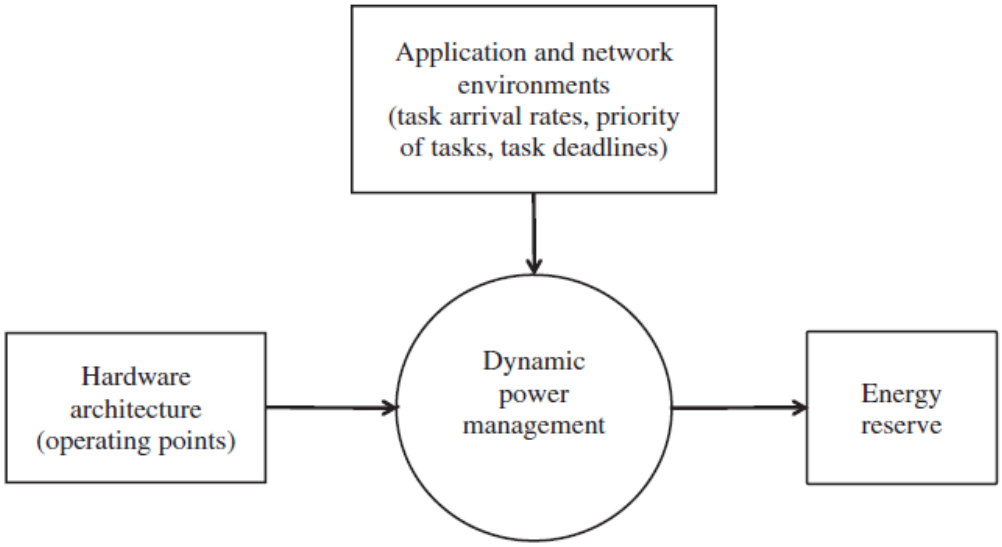



**Figure 8.12** Factors affecting a dynamic power management strategy.

This (see Figure 8.13) can be described as a circular process consisting of three basic operations – energy monitoring, power mode estimation, and task scheduling.

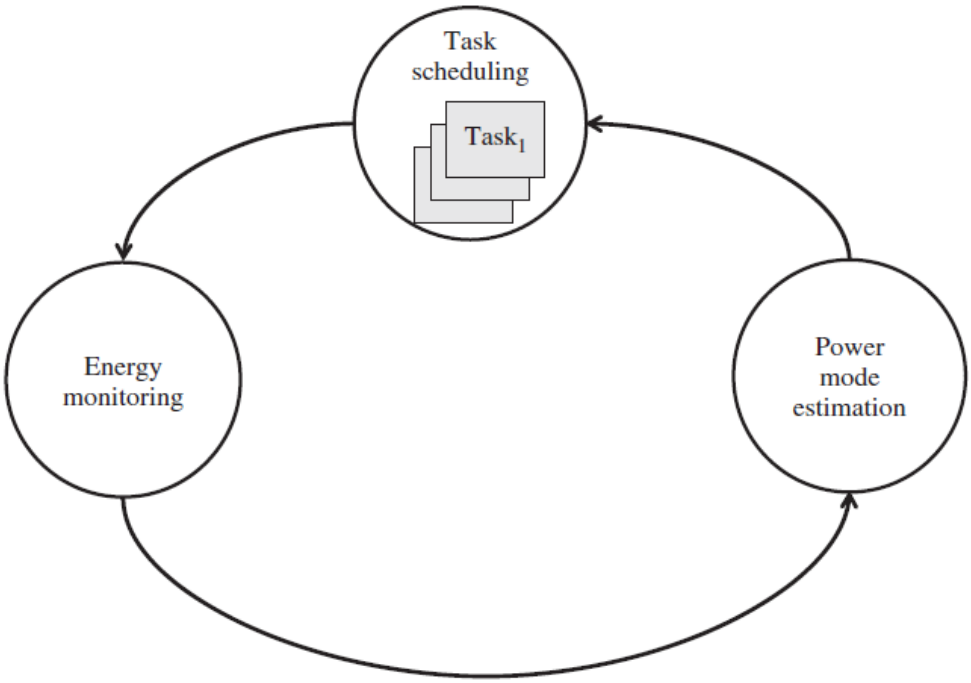



**Figure 8.13** An abstract architecture for a dynamic power management strategy.

Figure 8.13 illustrates how dynamic power management can be thought of as a machine that moves through different states in response to different types of events – tasks are scheduled in a task queue, and the execution time and energy consumption of the system are monitored. Depending on how fast the tasks are completed, a new power budget is estimated and transitions in power modes take place. In case of a deviation in the estimated power budget from the power mode that can be supported by the system, the DPM strategy decides the higher level of operating power mode.

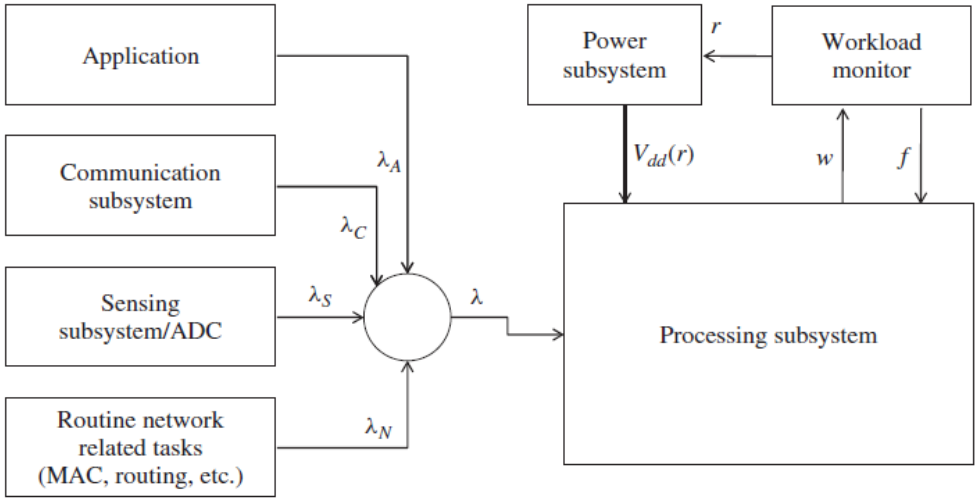



**Figure 8.14** A conceptual architecture of a dynamic voltage scaling. (This architecture is the modified version of the one proposed by Sinha and Chandrakasan in (Sinha and Chandrakasan 2001)).

Figure 8.14 shows an implementation of the abstract architecture of Figure 8.13 to support dynamic voltage scaling. The processing subsystem receives tasks from the application, the communication subsystem, and the sensing subsystem. Additionally, it handles internal tasks pertaining to network management, such as managing a routing table and sleeping schedules. Each of these sources produces a task at a rate of $\lambda i$ . The overall task arrival rate, $\lambda$, is the summation of the individual tasks arrival rates, $\lambda = \_ \lambda i$. The *workload monitor* observes $\lambda$ for a duration of $\tau$ seconds and predicts the task arrival rate for the next $\beta$ seconds. The estimated task arrival rate is represented by $r$ in the figure. Based on the newly computed task arrival rate $r$, the processing subsystem estimates the supply voltage and the clock frequency it requires to process upcoming tasks.