Wireless Sensor Networks (AEC526)

Professional Elective

IARE-R16

B.TECH-ECE-VII Sem

ELECTRONICS AND COMMUNICATION ENGINEERING

INSTITUTE OF AERONAUTICAL ENGINEERING

(AUTONOMOUS)

UNIT-I
OVERVIEW OF
WIRELESS SENSOR NETWORKS

Outline

- Challenges in wireless sensor networks.
- Characteristic requirements of sensor networks.
- Enabling technologies for wireless sensor networks.
- Advantages of sensor networks.
- Sensor network applications.

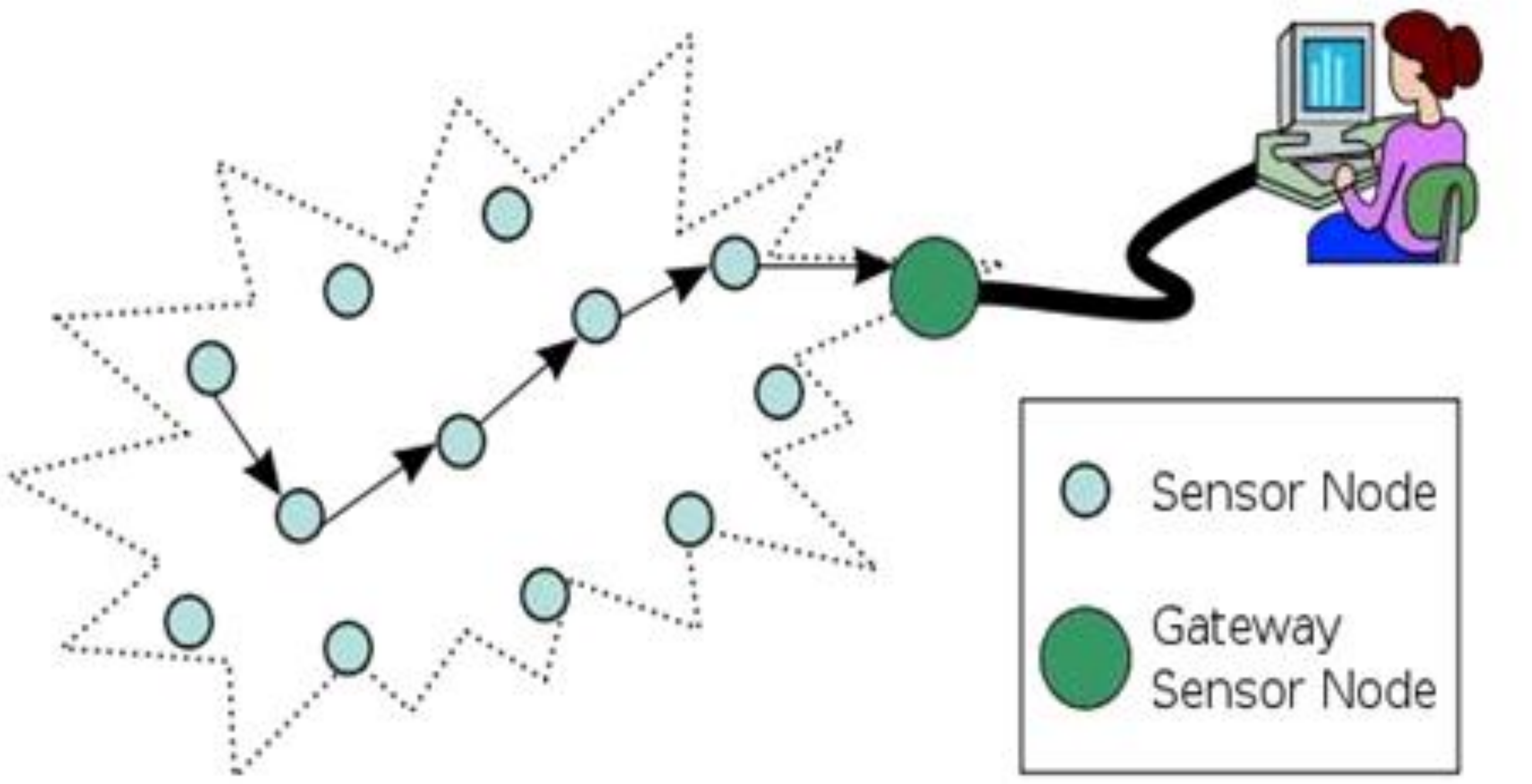
Introduction

- 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 collection of sensing devices that can communicate wirelessly.

Introduction

- 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 collection of sensing devices that can communicate wirelessly.

Wireless Sensor Network Architecture



Architecture for a WSN

Special addressing requirement

- Local unique addresses
- Data-centric
- Example: Each node has an unique number.

Attribute-based naming architecture

- Data is named by one or more attributes.
- Example: Each node is distinguished by an attribute – GPS sensors are practical for this.

Wireless Sensor Node

- sensor
  - A transducer
  - converts physical phenomenon e.g. heat, light, motion, vibration, and sound into electrical signals
- sensor node
  - basic unit in sensor network
  - contains on-board sensors, processor, memory, transceiver, and power supply
- sensor network
  - consists of a large number of sensor nodes
  - nodes deployed either inside or very close to the sensed phenomenon

Challenges of WSN

Design Challenges:

- **Heterogeneity**
  - The devices deployed may be of various types and need to collaborate with each other.
- **Distributed Processing**
  - The algorithms need to be centralized as the processing is carried out on different nodes.
- **Low Bandwidth Communication**
  - The data should be transferred efficiently between sensors

Challenges of WSN

- **Large Scale Coordination**
  - The sensors need to coordinate with each other to produce required results.
- **Utilization of Sensors**
  - The sensors should be utilized in a ways that produce the maximum performance and use less energy.
- **Real Time Computation**
  - The computation should be done quickly as new data is always being generated.

Challenges of WSN

Operational Challenges of Wireless Sensor Networks:

- Energy Efficiency
- Limited storage and computation
- Low bandwidth and high error rates
- Errors are common
  - Wireless communication
  - Noisy measurements
  - Node failure are expected
- Scalability to a large number of sensor nodes
- Survivability in harsh environments

Design issues of WSN

➢**Fault Tolerance**

Sensor nodes are vulnerable and frequently deployed in dangerous environment. Nodes can fail due to hardware problems or physical damage or by exhausting their energy supply. We expect the node failures to be much higher than the one normally considered in wired or infrastructure-based wireless network.

➢**Scalabilty:**

Sensor network vary in scale from several nodes to potentially several hundred thousand. In addition, the deployment density is also variable.

➢**Production cost:**

Because many deployment models consider the sensor nodes to be disposable devices, sensor network can compete with traditional information gathering approaches only if the individual sensor nodes can be produced very cheaply.

Design issues of WSN

- **Hardware Constraints:**

At minimum, every sensor node needs to have a sensing unit, a processing unit, a transmission unit, and a power supply. Optionally, the nodes may have several built-in sensors or additional devices such as a localization system to enable location-aware routing.

- **Sensor Network Topology**

Although WSNs have evolved in many aspects, they continue to be network with constrained resources in terms of energy, computing power, memory, and communications capabilities.

Design issues of WSN

- **Transmission Media**

The communication between the nodes is normally implemented using radio communication over the popular ISM bands. However, some sensor network use optical or infrared communication, with the latter having the advantage of being robust and virtually interference free.

- **Power Consumption:**

The size of the nodes limits the size of the battery. The software and hardware design needs to carefully consider the issues of efficient energy use.

Enabling Technology for WSN

- Cost reduction
  - For wireless communication, simple microcontroller, sensing, batteries
- Miniaturization
  - Some applications demand small size
  - "Smart dust" as the most extreme vision
- Energy scavenging
  - Recharge batteries from ambient energy (light, vibration, …)

Enabling Technology for WSN

- ***Operating systems***
  - **eCos** (*embedded configurable operating system*)
  - **uC/OS** (Micro-Controller Operating Systems)
  - **LiteOS, Contiki, TinyOS**

- ***Hardware standards***
  - Radio, Battery, Microcontroller, Analog circuit, and Sensor Interface

Enabling Technology for WSN

Some technologies have a direct impact in the IoT implementation

- Wireless sensor networks;
- *Machine-to-Machine* protocols;
- Network Technologies (*wireless*);
- Devices and micro-systems featuring low energy consumption;
- Micro and nano-*Energy Harvesting technologies*;
- Sensors technologies;
- Localization Systems (*indoor*);
- ICT Infrastructure of Internet (Big Data)

Applications of Wireless Sensor Network

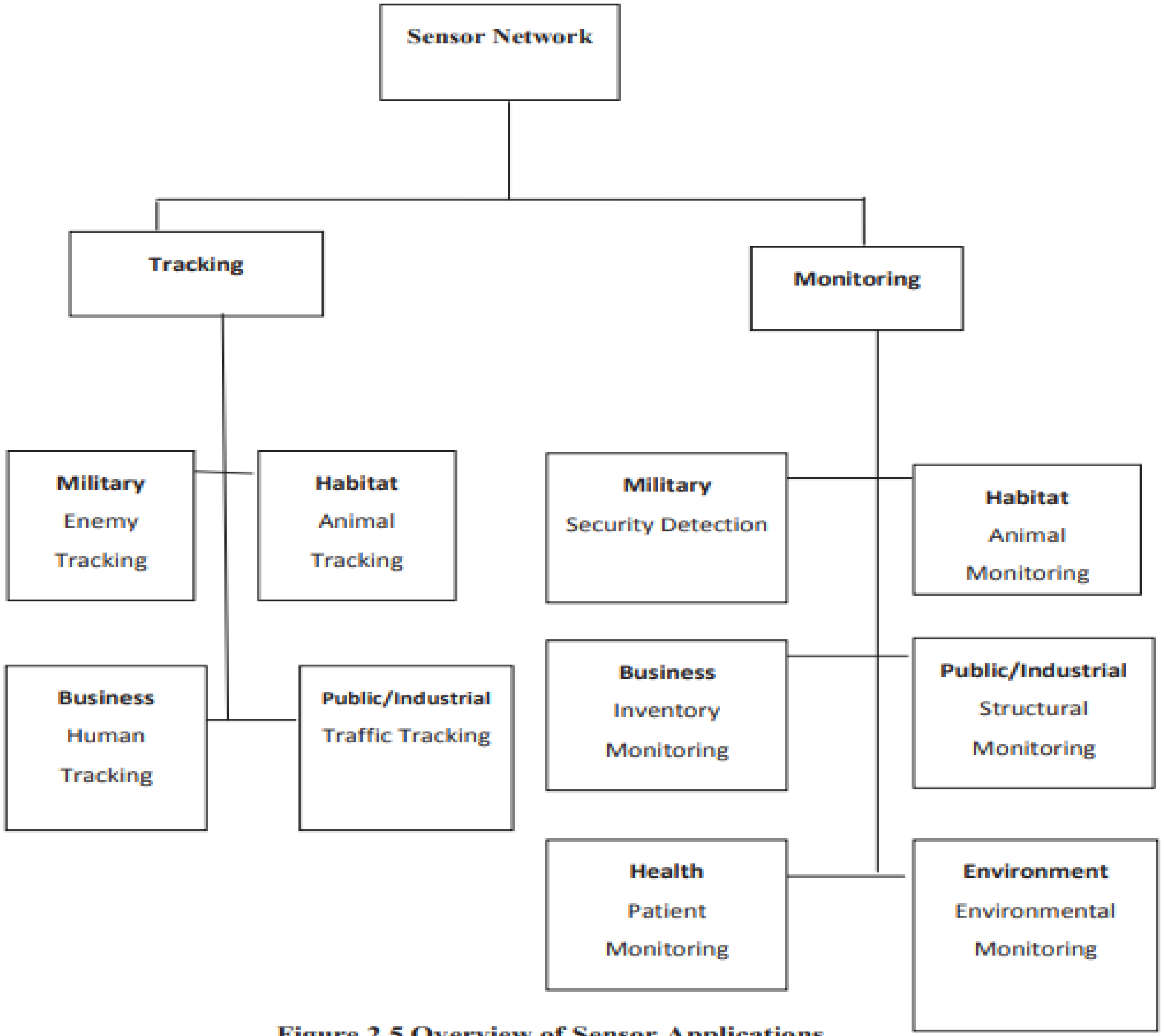

Sensor Network
Tracking
Monitoring
Military Enemy Tracking
Habitat Animal Tracking
Business Human Tracking
Public/Industrial Traffic Tracking
Military Security Detection
Habitat Animal Monitoring
Business Inventory Monitoring
Public/Industrial Structural Monitoring
Health Patient Monitoring
Environment Environmental Monitoring

Figure 2.5 Overview of Sensor Applications

Advantages of Wireless Sensor Network

- It avoids a lot of wiring .
- It can accommodate new devices at any time .
- It's flexible to go through physical partitions .
- It can be accessed through a centralized monitor
- **Ease of Deployment**

A sensor network contains hundreds or even thousands of nodes and can be deployed in remote or dangerous environments. Since these nodes are small and economical, throwing of hundreds or thousands of micro-sensors from a plane flying over a remote or dangerous area allows extracting information in ways that could not have been possible otherwise.

Advantages of Wireless Sensor Network

➢**Extended Range of Sensing**

Single macro-sensor nodes can only extract data about events in a limited physical range. In contrast, a micro-sensor network uses large numbers of nodes enabling them to cover a wide area.

➢**Improved Lifetime**

The nodes located close to each other will have correlated data therefore they can be grouped together. Only one of the nodes in a round robin fashion from the group therefore needs to be in active state at any instance of time keeping other nodes in sleep state. It will enhance the network life time.

Advantages of Wireless Sensor Network

- **Fault Tolerance**

In WSN several sensor nodes are close to each other and have correlated data, it makes these systems much more fault tolerant than single macro-sensor system.

- **Improved Accuracy:**

While an individual micro-sensor’s data might be less accurate than a macro-sensor’s data.

- **Lower Cost**

Even though, to replace each macro-sensor node several micro-sensor nodes are required they will still be collectively much cheaper than their macro-sensor counterpart due to their reduced size, simple as well as cheap circuitry and lesser accuracy constraints.

Disadvantages

- Lower speed compared to wired network.
- Less secure because hacker's laptop can act as Access Point. If you connected to their laptop, they'll read all your information (username, password.. etc).
- More complex to configure than wired network.
- Gets distracted by various elements like Blue-tooth .
- Still Costly at large.
- It does not make sensing quantities in buildings easier.
- It does not reduce costs for installation of sensors.
- It does not allow us to do more than can be done with a wired system

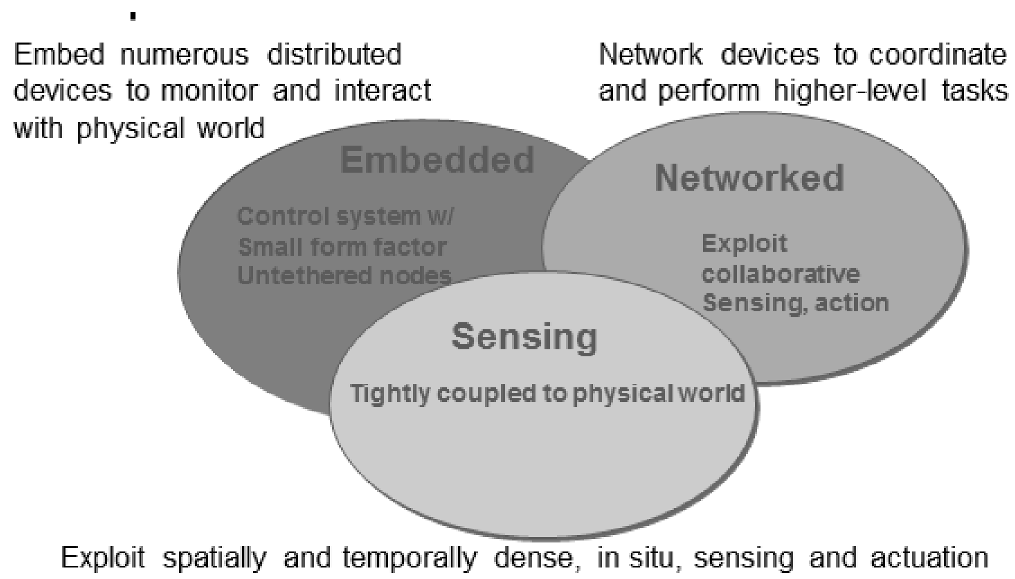
UNIT-II

ARCHITECTURES

Single-node architecture, hardware components, energy consumption of sensor nodes, operating systems and execution environments, network architecture, sensor network scenarios, optimization goals and figures of merit, gateway concepts.

INTRODUCTION

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


Embed numerous distributed devices to monitor and interact with physical world
Network devices to coordinate and perform higher-level tasks
Embedded
Control system w/ Small form factor Untethered nodes
Networked
Exploit collaborative Sensing, action
Sensing
Tightly coupled to physical world
Exploit spatially and temporally dense, in situ, sensing and actuation

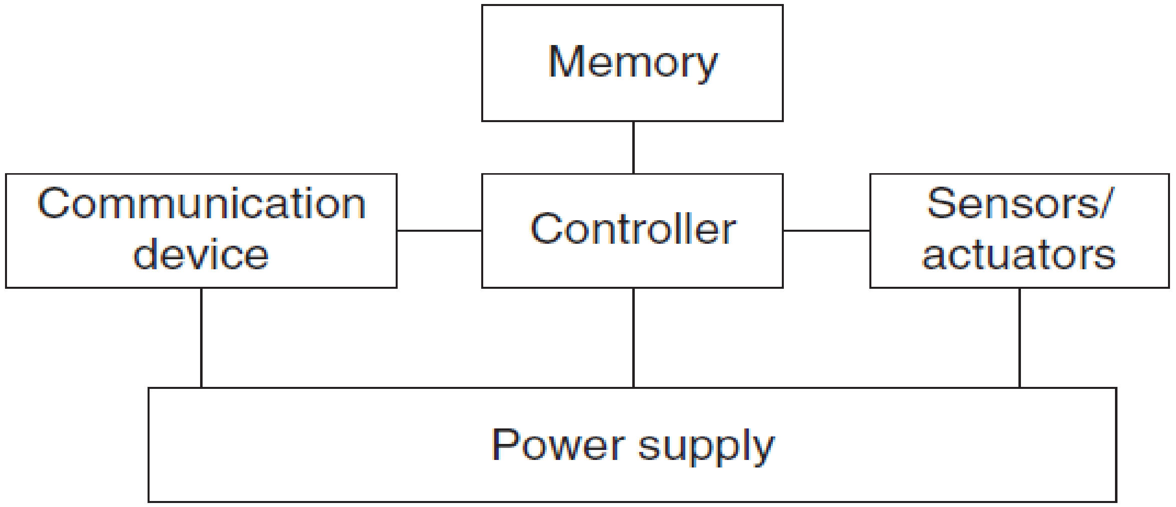
SINGLE-NODE ARCHITECTURE:

HARDWARE COMPONENTS:

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

SINGLE-NODE ARCHITECTURE:

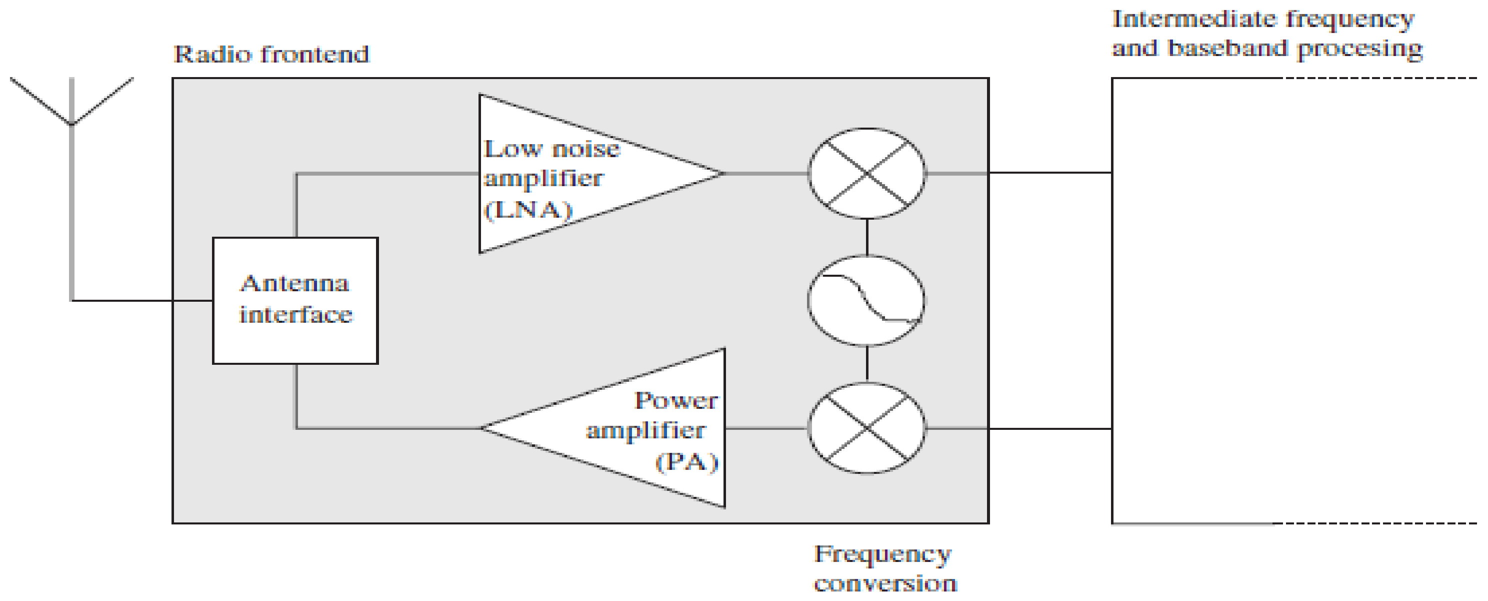
HARDWARE COMPONENTS:



Sensor node Hardware components

SINGLE-NODE ARCHITECTURE:

Communication Device:



RF front end

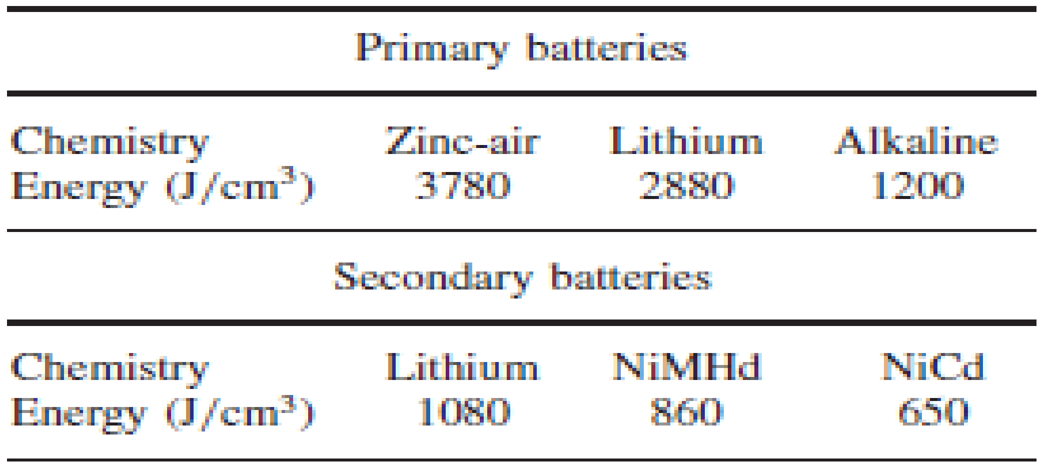
SINGLE-NODE ARCHITECTURE:

Power supply:

As usually no tethered power supply is available, some form of batteries are necessary to provide energy. Sometimes, some form of recharging by obtaining energy from the environment is available as well (e.g. solar cells). There are essentially two aspects: Storing energy and Energy scavenging.

SINGLE-NODE ARCHITECTURE:

Power supply:

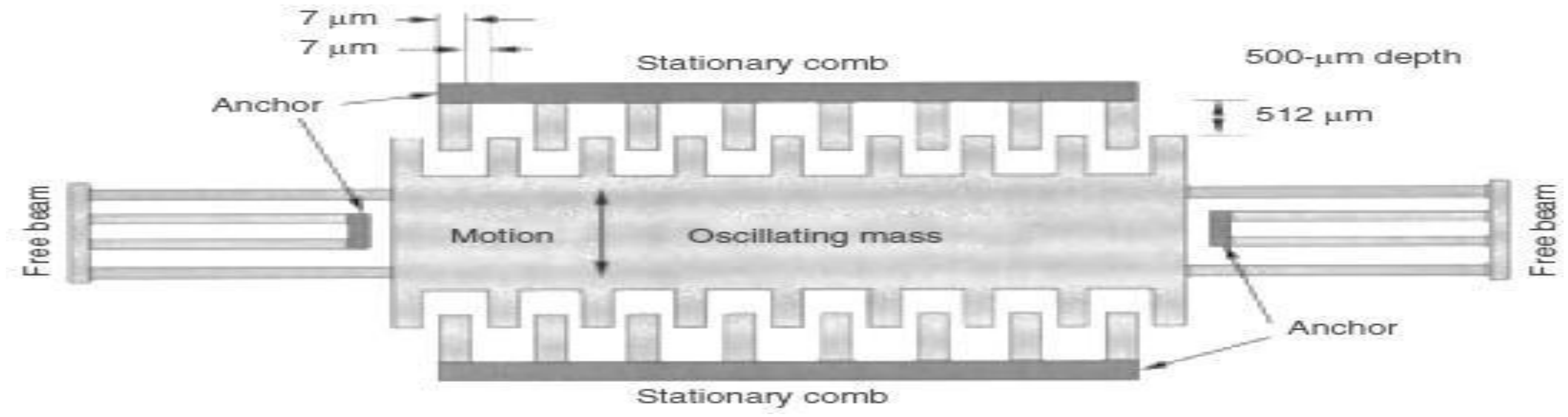


Primary batteries			
Chemistry	Zinc-air	Lithium	Alkaline
Energy (J/cm^3)	3780	2880	1200
Secondary batteries			
Chemistry	Lithium	NiMHd	NiCd
Energy (J/cm^3)	1080	860	650

Energy densities for various primary and secondary battery types

SINGLE-NODE ARCHITECTURE:

Power supply:

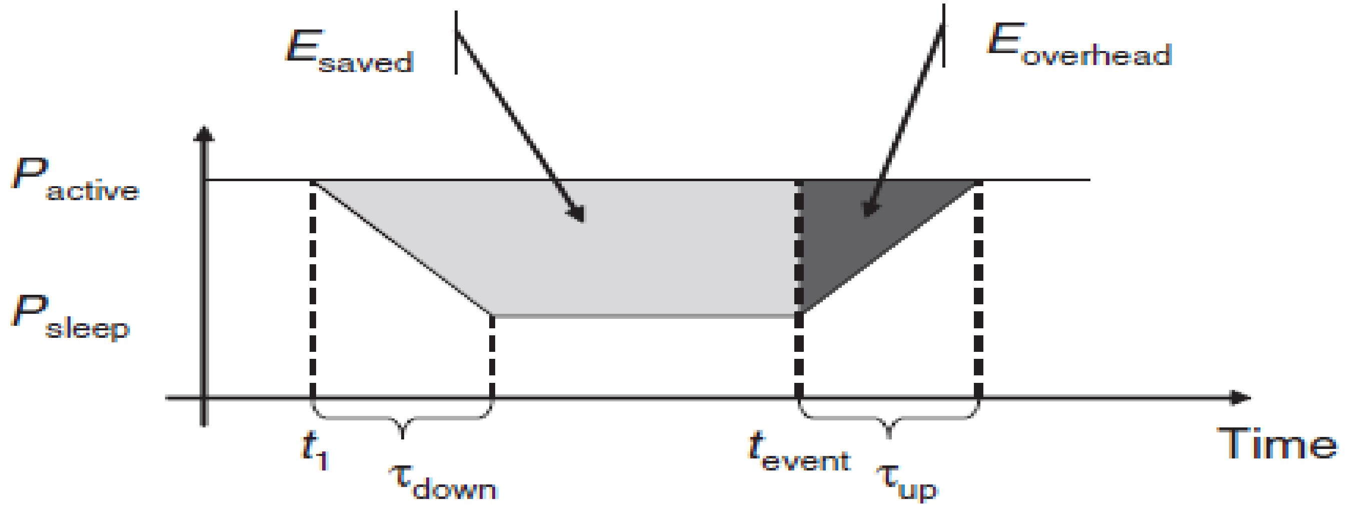


A MEMS device for converting vibrations to electrical energy, based on a variable capacitor

ENERGY CONSUMPTION OF SENSOR NODES:

The main consumers of energy are the controller, the radio front ends, the memory, and type of the sensors. One method to reduce power consumption of these components is designing low- power chips, it is the best starting point for an energy-efficient sensor node.

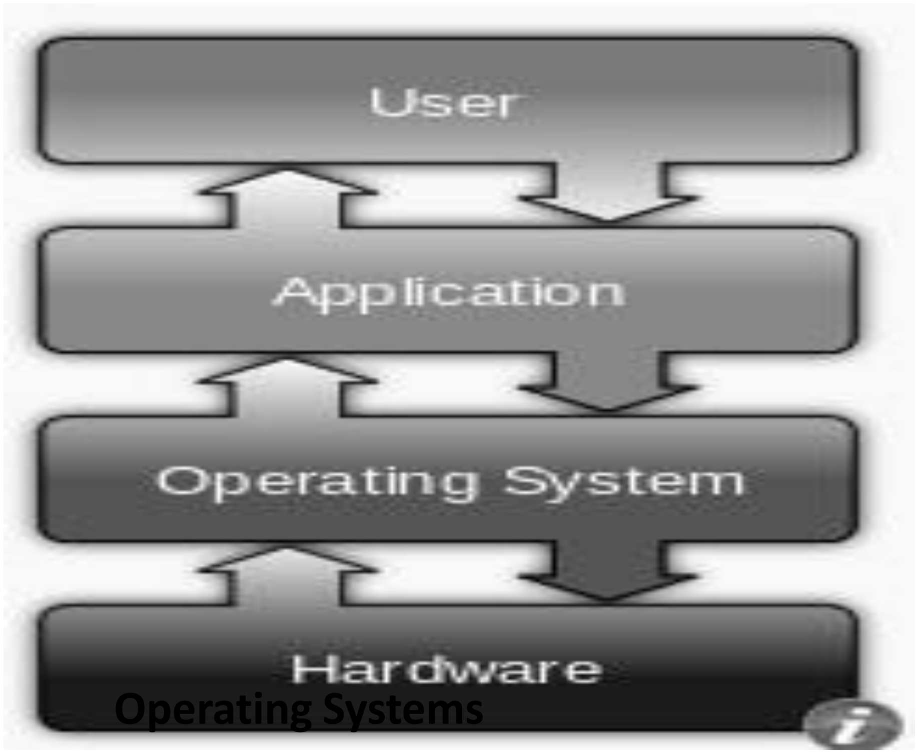
ENERGY CONSUMPTION OF SENSOR NODES:



Energy savings and overheads for sleep modes

OPERATING SYSTEMS AND EXECUTION ENVIRONMENTS:

Embedded operating systems:



Operating Systems

OPERATING SYSTEMS AND EXECUTION ENVIRONMENTS:

Programming paradigms and application programming interfaces

Concurrent Programming:

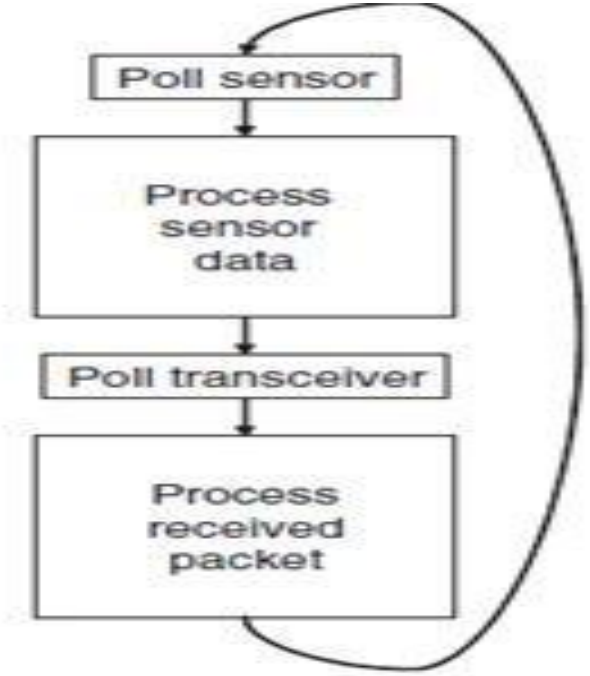


Figure 1.8 Sequential programming model

OPERATING SYSTEMS AND EXECUTION ENVIRONMENTS:

Programming paradigms and application programming interfaces
Process-based concurrency

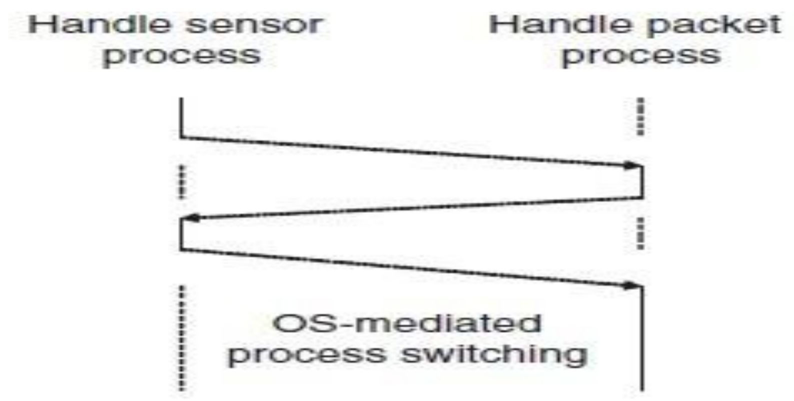


Figure 1.9 Process-based programming model

OPERATING SYSTEMS AND EXECUTION ENVIRONMENTS:

Programming paradigms and application programming interfaces
Event-based programming:

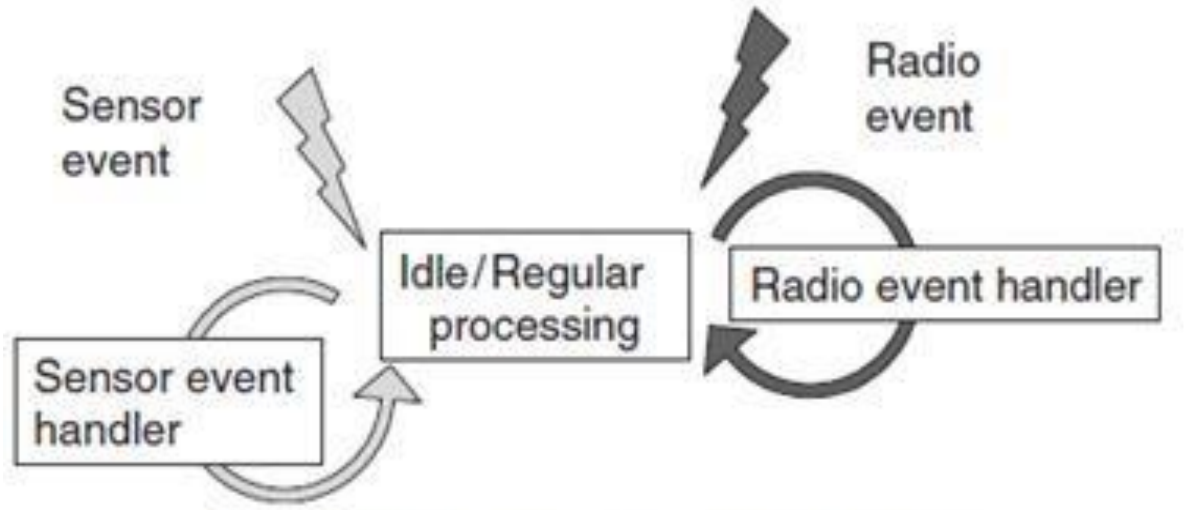


Figure 1.10 Event-based programming model

NETWORK ARCHITECTURE:

The basic principles of turning individual sensor nodes into a wireless sensor network. In this optimization goals of how a network should function are discussed as

- Sensor network scenarios
- Optimization goals and figures of merit
- Gateway concepts

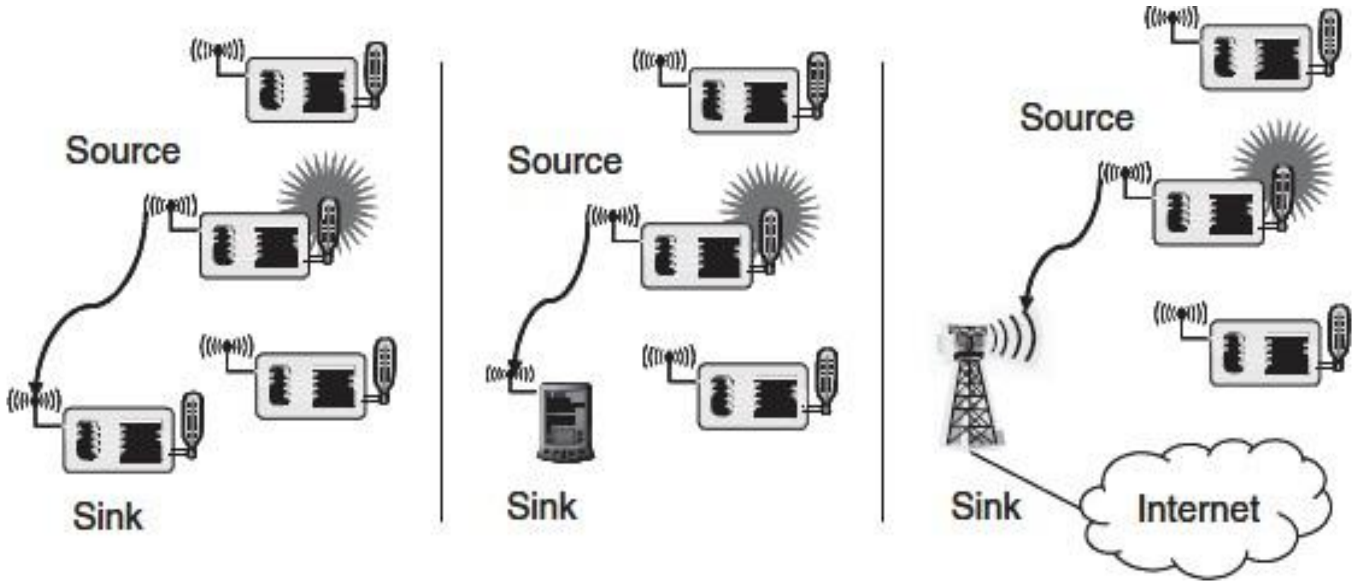
NETWORK ARCHITECTURE:

•**<u>SENSOR NETWORK SCENARIOS:</u>**

Source is any unit in the network that can provide information (sensor node). A sink is the unit where information is required, it could belong to the sensor network or outside this network to interact with another network or a gateway to another larger Internet**.** Sinks are illustrated by Figure 1.11, showing sources and sinks in direct communication.

NETWORK ARCHITECTURE:

SENSOR NETWORK SCENARIOS:



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

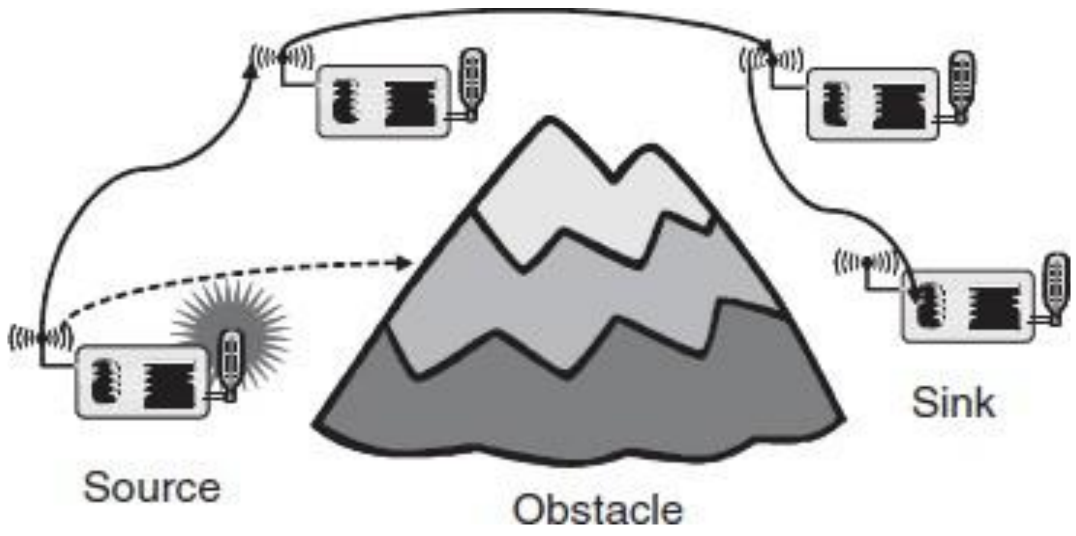
NETWORK ARCHITECTURE:

Single-hop versus multi-hop networks:

Because of limited distance the direct communication between source and sink is not always possible. In WSNs, to cover a lot of environment the data packets taking multi hops from source to the sink. To overcome such limited distances it better to use relay stations, The data packets taking multi hops from source to the sink

NETWORK ARCHITECTURE:

Single-hop versus multi-hop networks:

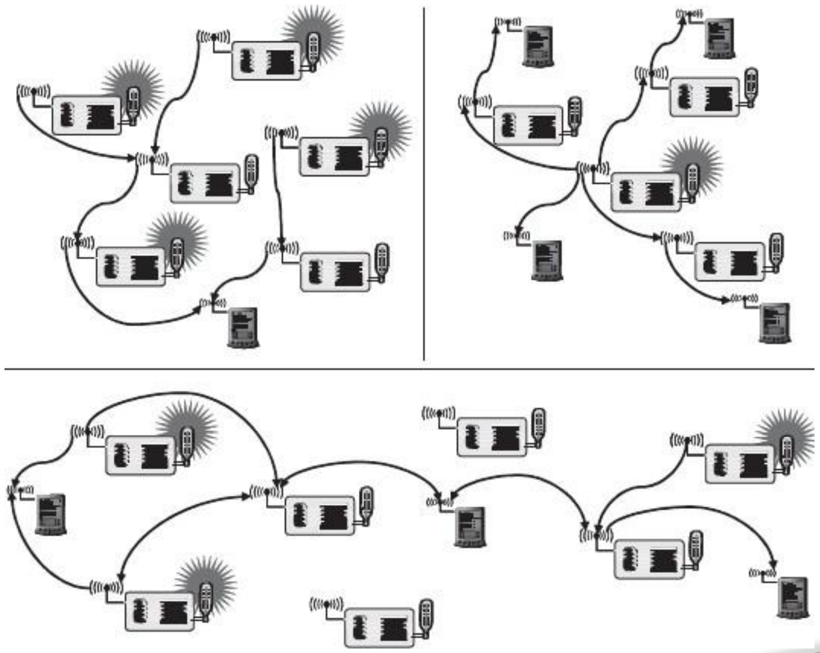


Multi-hop networks: As direct communication is impossible because of distance and/or obstacles

NETWORK ARCHITECTURE:

Multiple sinks and sources:

In many cases, multiple sources and multiple sinks present. Multiple.sources should send information to multiple sinks. Either all or some of the information has to reach all or some of the sinks.



NETWORK ARCHITECTURE:

Three types of mobility:

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

- Node mobility
- Sink mobility
- Event mobility

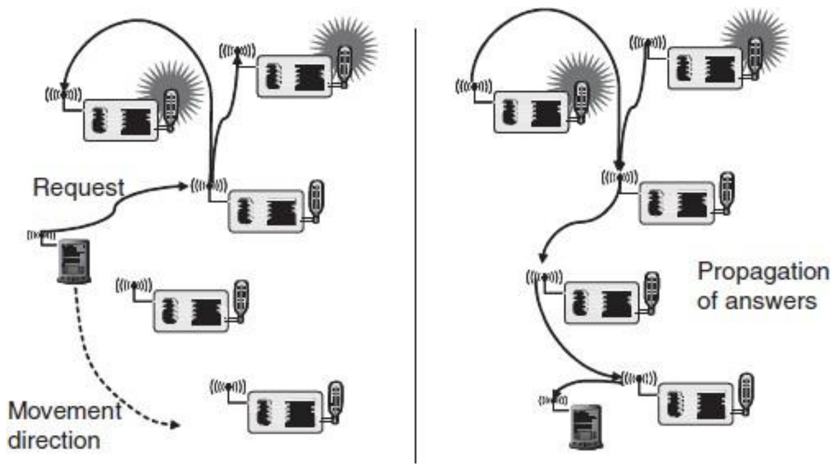
NETWORK ARCHITECTURE:

Node Mobility:

The wireless sensor nodes themselves can be mobile. The meaning of such mobility is highly application dependent. In examples like environmental control, node mobility should not happen; in livestock surveillance (sensor nodes attached to cattle, for example), it is the common rule. In the face of node mobility, the network has to reorganize to function correctly.

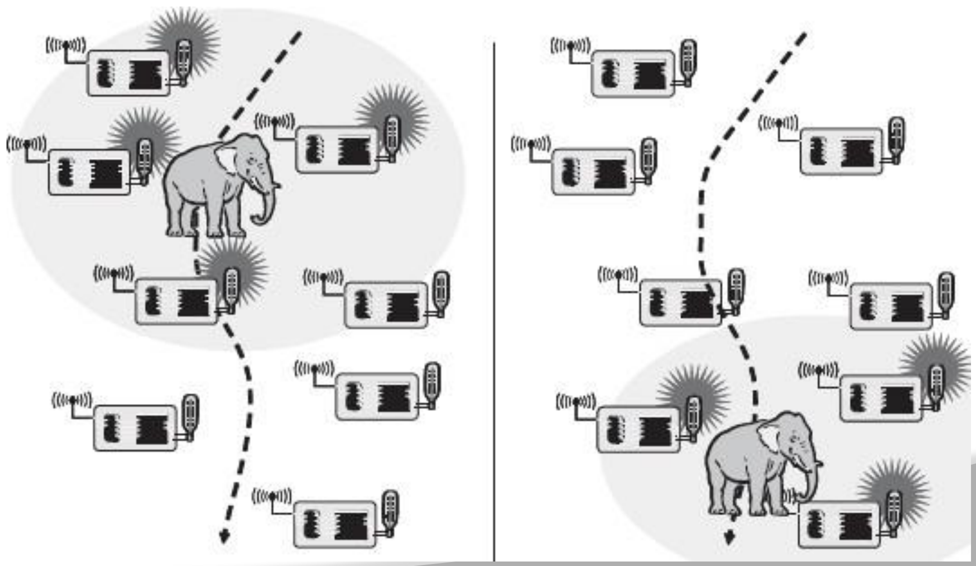
NETWORK ARCHITECTURE:

Sink Mobility: The information sinks can be mobile. For example, a human user requested information via a PDA while walking in an intelligent building. In a simple case, such a requester can interact with the WSN at one point and complete its interactions before moving on, In many cases, consecutive interactions can be treated as separate, unrelated requests.



NETWORK ARCHITECTURE:

Event Mobility: In tracking applications, the cause of the events or the objects to be tracked can be mobile. In such scenarios, it is (usually) important that the observed event is covered by a sufficient number of sensors at all time. As the event source moves through the network, it is accompanied by an area of activity within the network – this has been called the frisbee model.



OPTIMIZATION GOALS AND FIGURES OF MERIT:

For all WSN scenarios and application types have to face the challenges such as,

.

How to optimize a network and How to compare these solutions?

•How to decide which approach is better?

•How to turn relatively inaccurate optimization goals into measurable figures of merit? For all the above questions the general answer is obtained from

•Quality of service

•Energy efficiency

•Scalability

•Robustness

OPTIMIZATION GOALS AND FIGURES OF MERIT:

Quality of service:

WSNs differ from other conventional communication networks in the type of service they offer. These networks essentially only move bits from one place to another Some generic possibilities are

- Event detection/reporting probability
- Event classification error
- Event detection delay
- Missing reports
- Approximation accuracy
- Tracking accuracy

OPTIMIZATION GOALS AND FIGURES OF MERIT:

Energy efficiency:

Energy efficiency should be optimization goal. The
· most commonly considered aspects are:

- Energy per correctly received Bit
- Energy per reported (unique) event
- Delay/energy trade-offs
- Network lifetime
- Time to first node death
- Network half-life
- Time to partition
- Time to loss of coverage
- Time to failure of first event notification

OPTIMIZATION GOALS AND FIGURES OF MERIT:

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 obviously essential requirement.

OPTIMIZATION GOALS AND FIGURES OF MERIT:

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 by finding other routes.

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 by finding other routes.

GATE WAY CONCEPTS:

Need for gateways:

•For practical deployment, a sensor network only concerned with itself is insufficient.

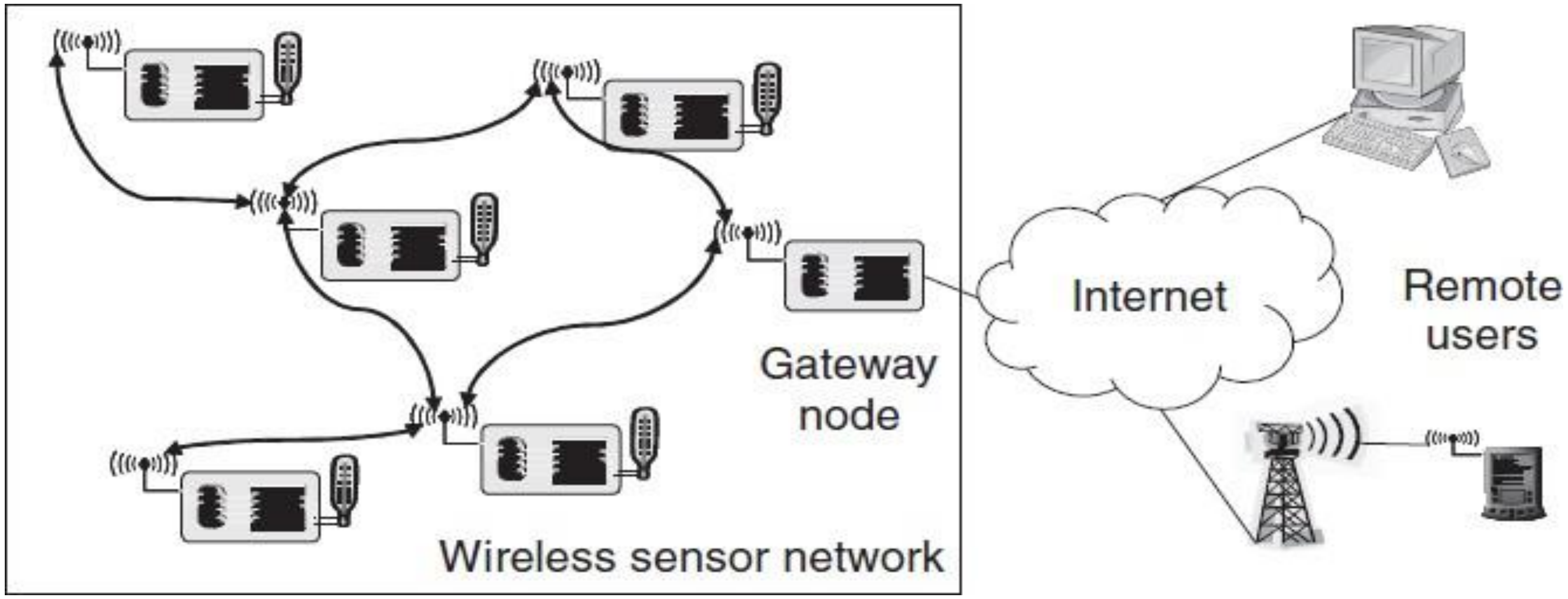
•The network rather has to be able to interact with other information devices for example to read the temperature sensors in one’s home while traveling and accessing the Internet via a wireless.

•Wireless sensor networks should also exhibit an appropriate robustness

•They should not fail just because of a limited number of nodes run out of energy or because of their environment changes and breaks existing radio links between two nodes.

•If possible, these failures have to be compensated by finding other routes.

GATE WAY CONCEPTS:

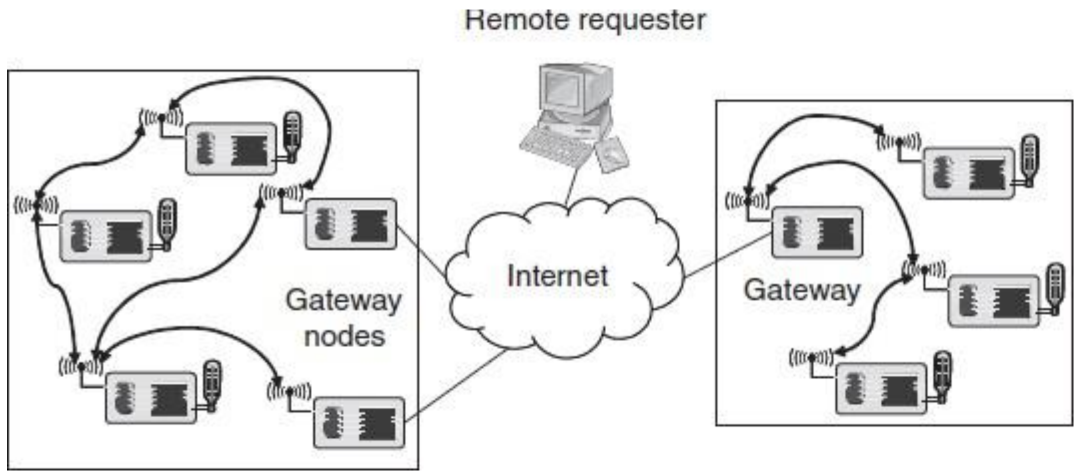


A wireless sensor network with gateway node, enabling access to remote clients via the Internet

GATE WAY CONCEPTS:

WSN - Internet communication:

Assume that the initiator of a WSN – Internet communication resides in the WSN.

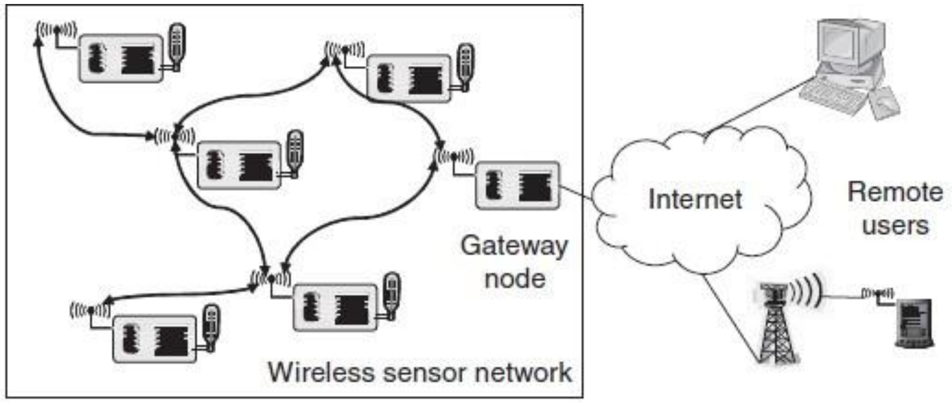


wireless Sensor Network with gateway node, enabling access to remote clients via the WSN

GATE WAY CONCEPTS:

Internet to WSN communication: The case of an Internet-based entity trying to access services of a WSN is even more challenging.

.

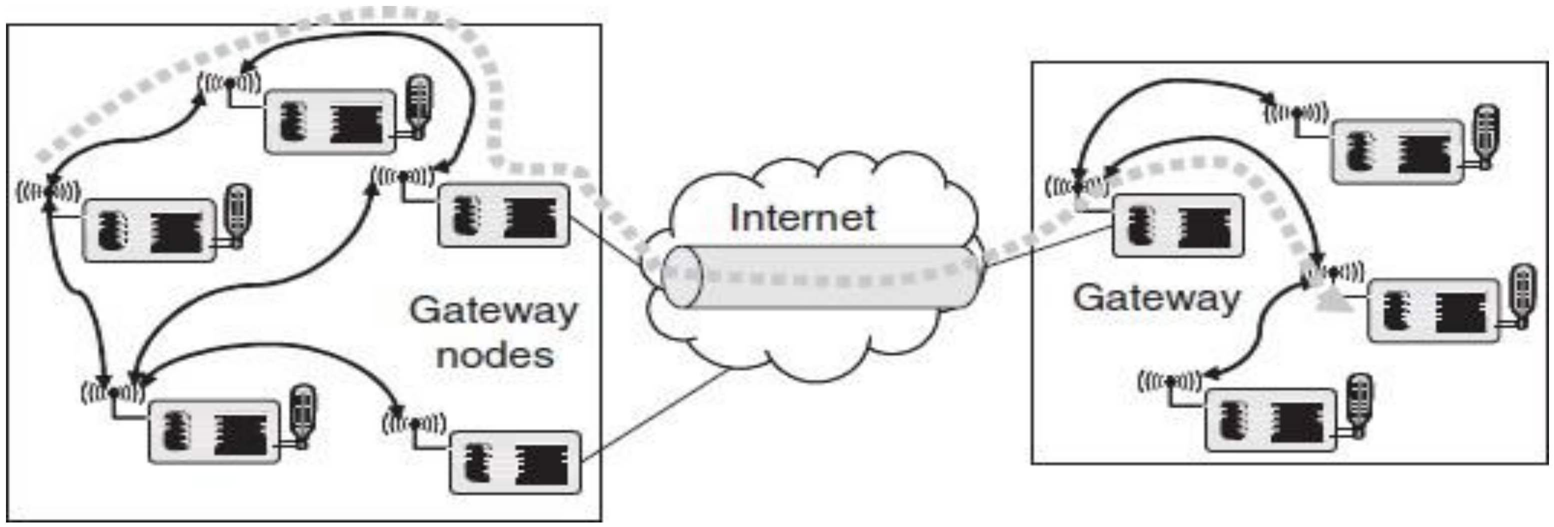


A wireless sensor network with gateway node, enabling access to remote clients via the Internet

wireless Sensor Network with gateway node, enabling access to remote clients via the internet

GATE WAY CONCEPTS:

WSN tunnelling:



Connecting two WSNs with a tunnel over the Internet

UNIT III
NETWORKING SENSORS

• Goals of this chapter

- Controlling when to send a packet and when to listen for a packet are perhaps the two most important operations in a wireless network
- Especially, idly waiting wastes huge amounts of energy
- This chapter discusses schemes for this medium access control that are
- Suitable to mobile and wireless networks
- Emphasize energy-efficient operation

Overview

- Principal options and difficulties
- Contention-based protocols
- Schedule-based protocols
- IEEE 802.15.4

Principal options and difficulties

- Medium access in wireless networks is difficult mainly because of
  - Impossible (or very difficult) to sende and receive at the same time
  - Interference situation at receiver is what counts for transmission success, but can be very different from what sender can observe
  - High error rates (for signaling packets) compound the issues

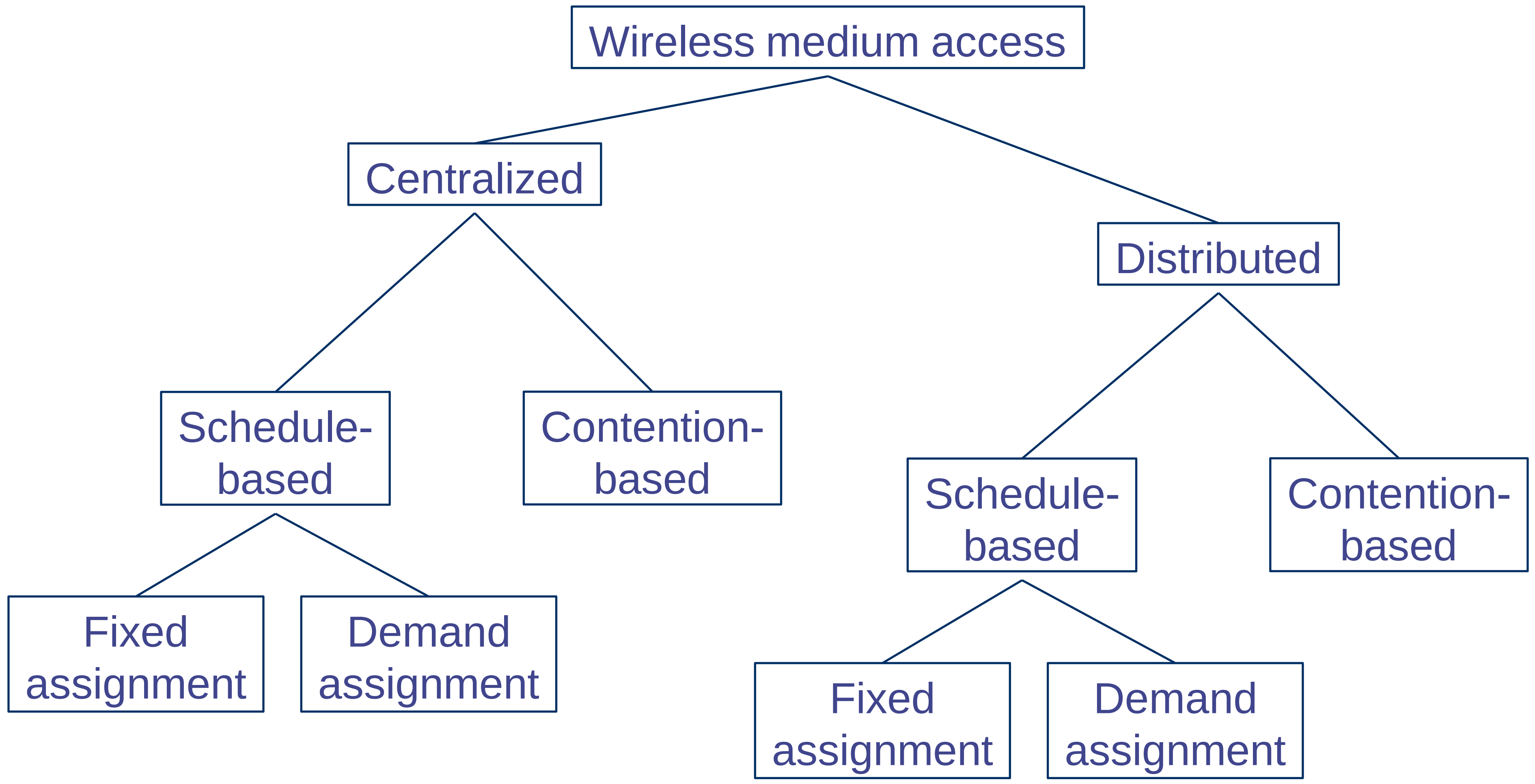
- Requirement
  - As usual: high throughput, low overhead, low error rates, …
  - Additionally: energy-efficient, handle switched off devices!

Requirements for energy-efficient MAC protocols

- Recall
- Transmissions are costly
- Receiving about as expensive as transmitting
- Idling can be cheaper but is still expensive
- Energy problems
- Collisions – wasted effort when two packets collide
- Overhearing – waste effort in receiving a packet destined for another node
- Idle listening – sitting idly and trying to receive when nobody is sending
- Protocol overhead

- Always nice: Low complexity solution

Main options



Centralized medium access

- Idea: Have a central station control when a node may access the medium
  - Example: Polling, centralized computation of TDMA schedules
  - Advantage: Simple, quite efficient (e.g., no collisions), burdens the central station

- Not directly feasible for non-trivial wireless network sizes
- But: Can be quite useful when network is somehow divided into smaller groups
  - Clusters, in each cluster medium access can be controlled centrally – compare Bluetooth piconets, for example

Schedule- vs. contention-based MACs

- Schedule-based MAC
  - A schedule exists, regulating which participant may use which resource at which time (TDMA component)
  - Typical resource: frequency band in a given physical space (with a given code, CDMA)
  - Schedule can be fixed or computed on demand
    - Usually: mixed – difference fixed/on demand is one of time scales
  - Usually, collisions, overhearing, idle listening no issues
  - Needed: time synchronization!

- Risk of colliding packets is deliberately taken
- Hope: coordination overhead can be saved, resulting in overall improved efficiency
- Mechanisms to handle/reduce probability/impact of collisions required
- Usually, randomization used somehow

Overview

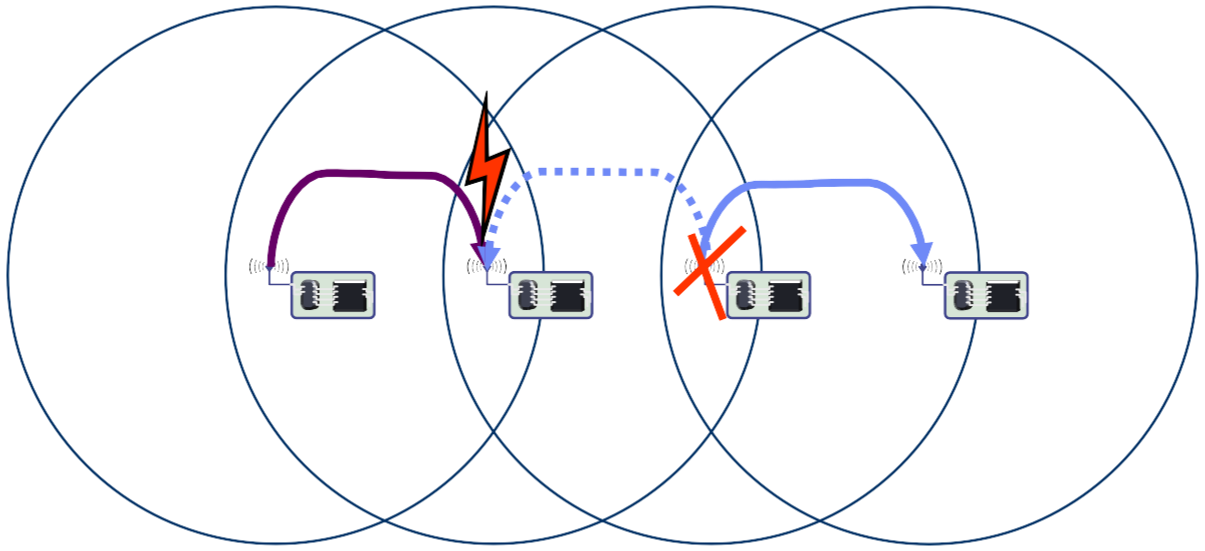
- Principal options and difficulties
- Contention-based protocols
  - MACA
  - S-MAC, T-MAC
  - Preamble sampling, B-MAC
  - PAMAS
- Schedule-based protocols
- IEEE 802.15.4

Distributed, contention-based MAC

- Basic ideas for a distributed MAC
  - ALOHA – no good in most cases
  - Listen before talk (Carrier Sense Multiple Access, CSMA) – better, but suffers from sender not knowing what is going on at receiver, might destroy packets despite first listening for a

! Receiver additionally needs some possibility to inform possible senders in its vicinity about impending transmission (to "shut them up" for this duration)

Overview



Main options to shut up senders

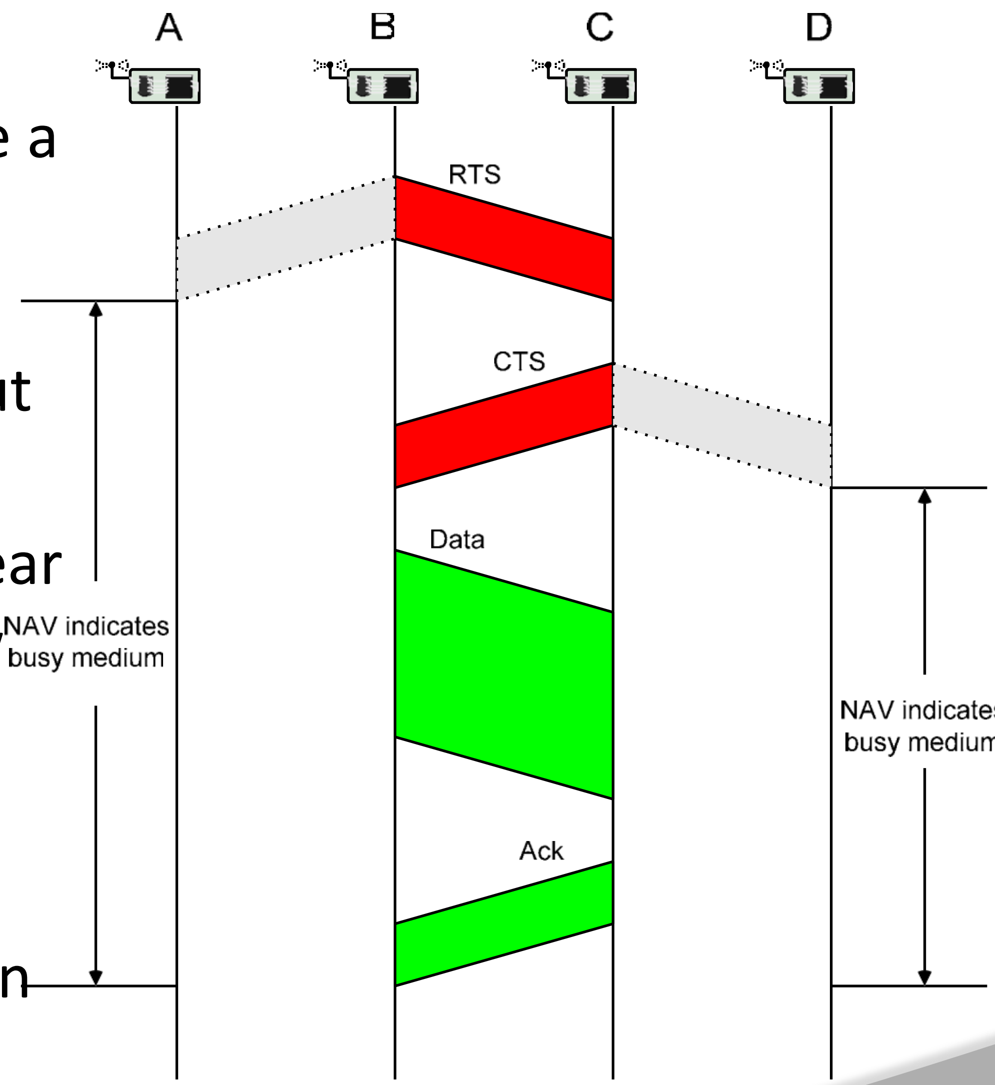
- Receiver informs potential interferers while a reception is on-going
  - By sending out a signal indicating just that
  - Problem: Cannot use same channel on which actual reception takes place

  ! Use separate channel for signaling
  - Busy tone protocol
- Receiver informs potential interferers before a reception is on-going
  - Can use same channel
  - Receiver itself needs to be informed, by sender, about impending transmission
  - Potential interferers need to be aware of such information, need to store it

Receiver informs interferers before transmission – MACA

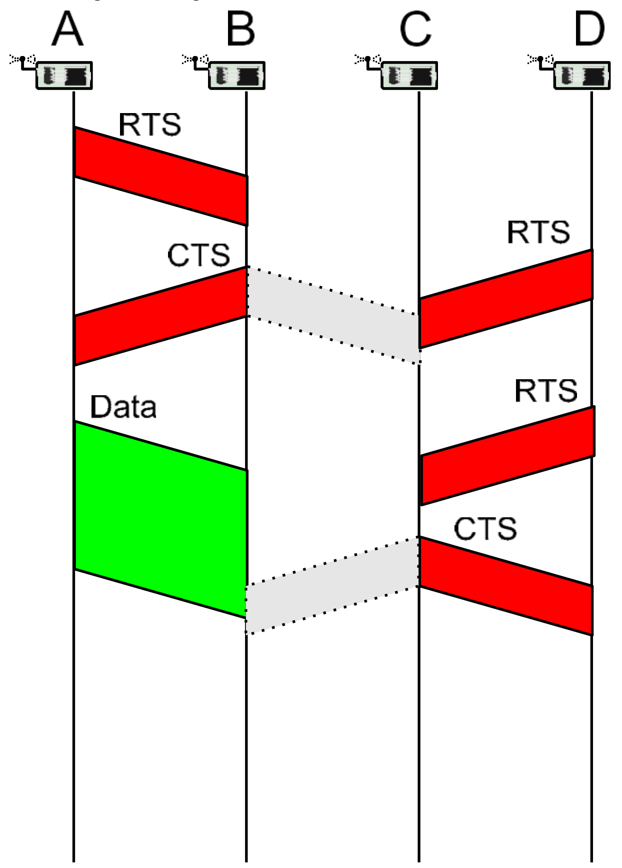
- Sender B asks receiver C whether C is able to receive a transmission
  Request to Send (RTS)
- Receiver C agrees, sends out a Clear to Send (CTS)
- Potential interferers overhear either RTS or CTS and know about impending transmission and for how long it will last

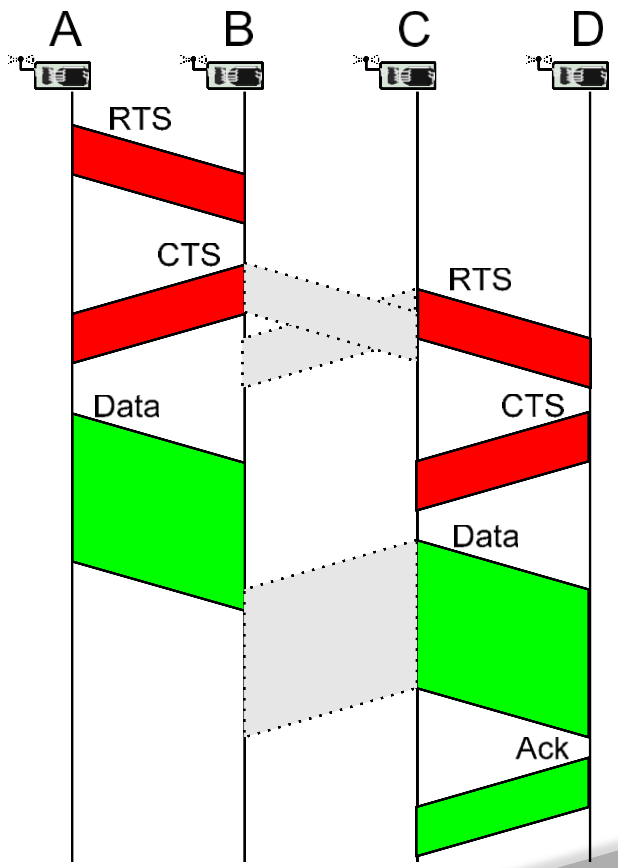
MACA protocol (used e.g. in IEEE 802.11)



RTS/CTS

- RTS/CTS ameliorate, but do not solve hidden/exposed terminal problems
- Example problem cases:



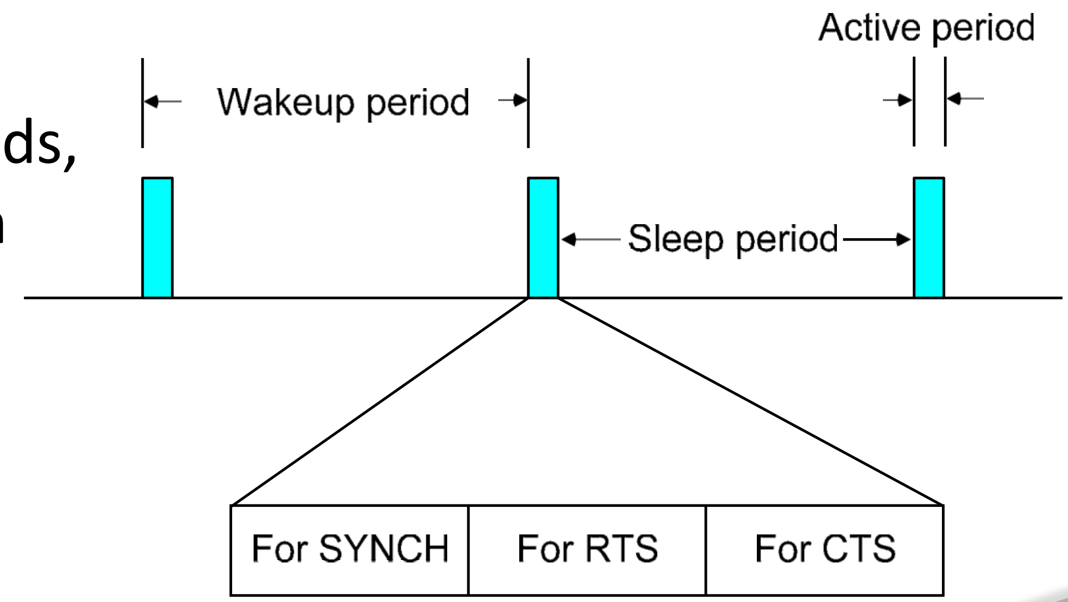


MACA Problem: Idle listening

- Need to sense carrier for RTS or CTS packets
- In some form shared by many CSMA variants; but e.g. not by busy tones
- Simple sleeping will break the protocol
- IEEE 802.11 solution: ATIM windows & sleeping
- Basic idea: Nodes that have data buffered for receivers send
- traffic indicators at pre-arranged points in time
- Receivers need to wake up at these points, but can sleep otherwise
- Parameters to adjust in MACA
- Random delays – how long to wait between listen/transmission attempts?
- Number of RTS/CTS/ACK re-trials

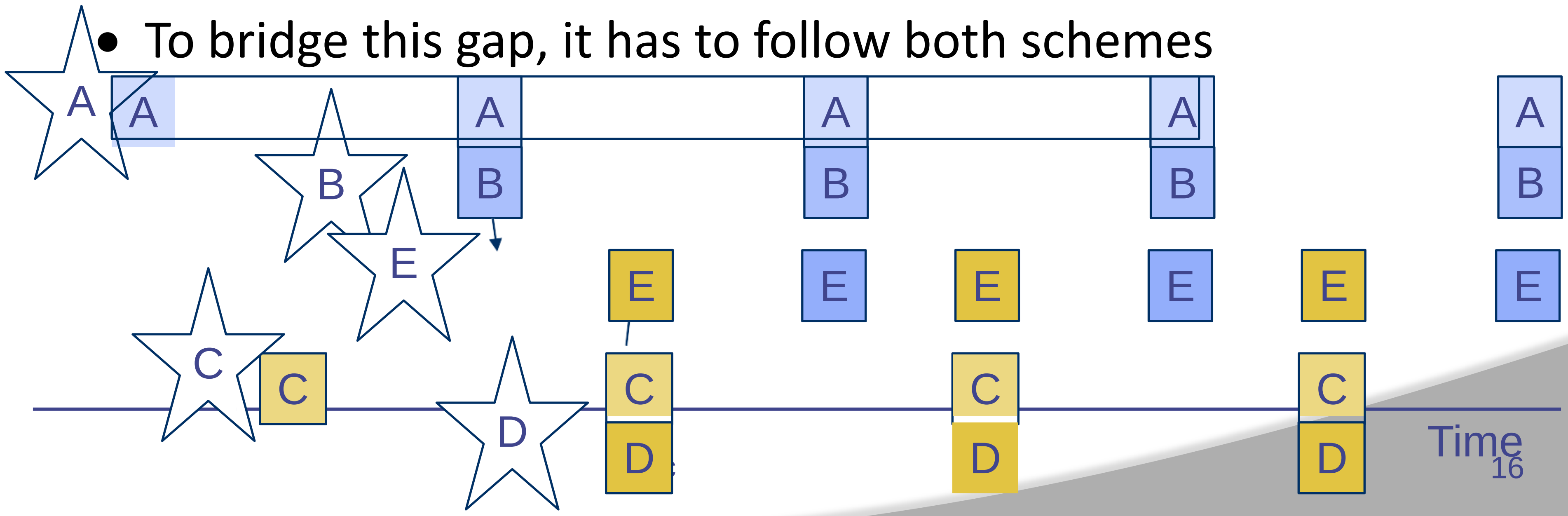
Sensor-MAC (S-MAC)

- MACA's idle listening is particularly unsuitable if average data rate is low
- Most of the time, nothing happens
- Idea: Switch nodes off, ensure that neighboring nodes turn on simultaneously to allow packet exchange (rendez-vous)
- Only in these active periods, packet exchanges happen
- Need to also exchange wakeup schedule between neighbors
- When awake, essentially perform RTS/CTS
- Use SYNCH, RTS, CTS phases



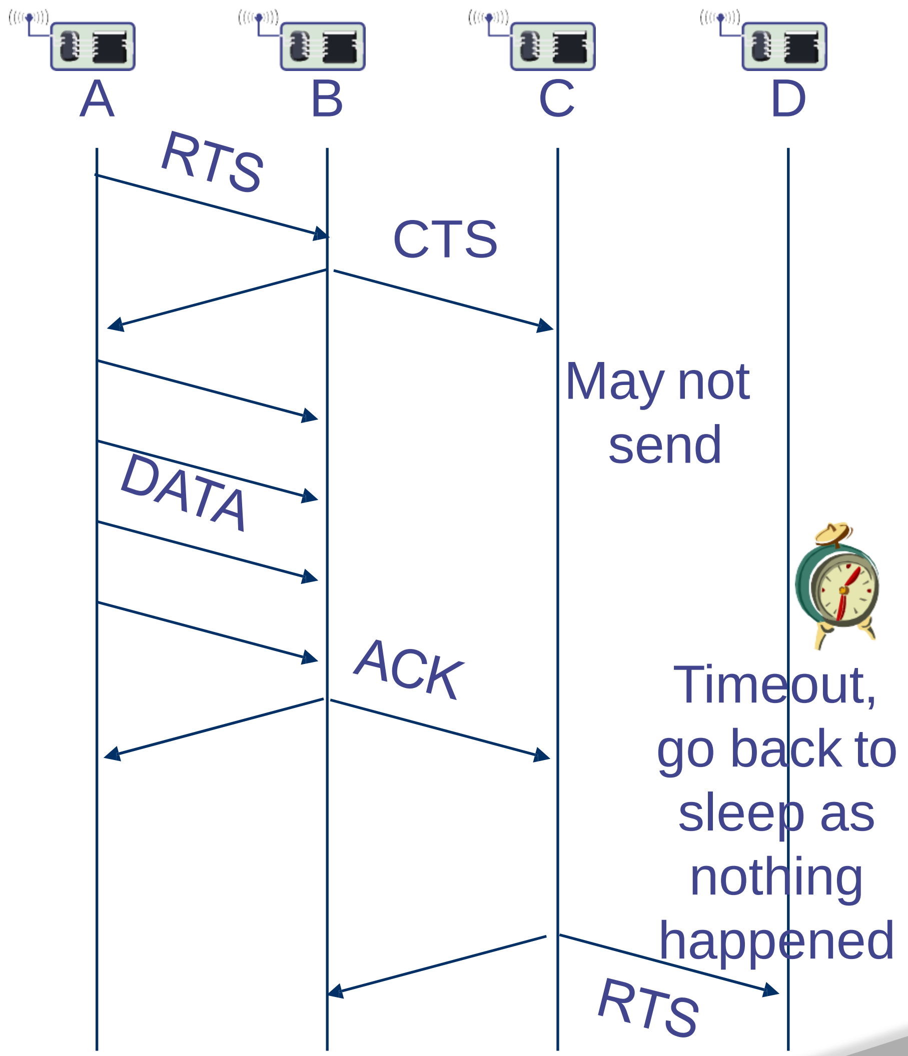
S-MAC synchronized islands

- Nodes try to pick up schedule synchronization from neighboring nodes
- If no neighbor found, nodes pick some schedule to start with
- If additional nodes join, some node might learn about two different schedules from different nodes
  - “Synchronized islands”
- To bridge this gap, it has to follow both schemes



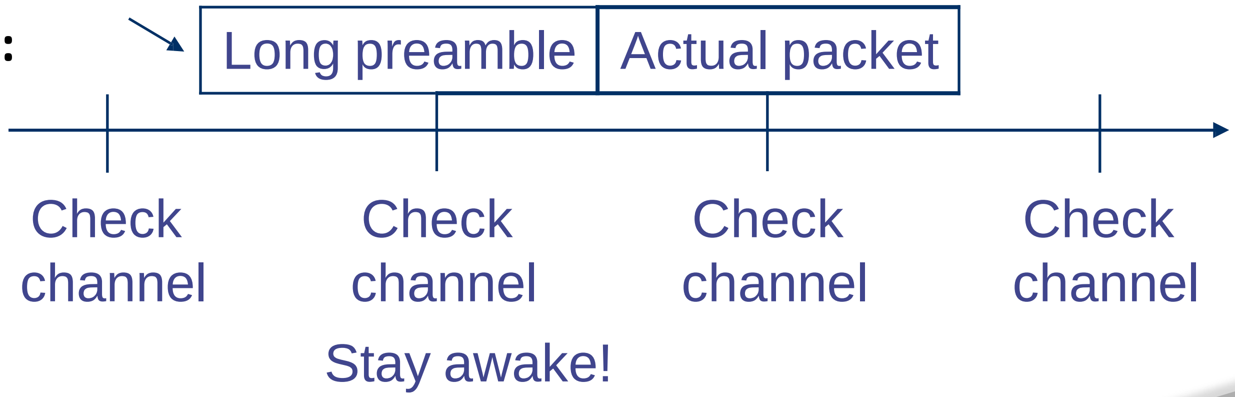
Timeout-MAC (T-MAC)

- In S-MAC, active period is of constant length
- Nodes stay awake needlessly long
- Idea: Prematurely go back to sleep mode when no traffic has happened for a certain time (=timeout) ! T-MACAdaptive duty cycle
- One ensuing problem: Early sleeping



Preamble Sampling

- So far: Periodic sleeping supported by some means to synchronize wake up of nodes to ensure rendez-vous between sender and receiver
- Alternative option: Don't try to explicitly synchronize nodes
- Have receiver sleep and only periodically sample the channel
- Use long preambles to ensure that receiver stays awake to catch actual packet
- Example: WiseMAC
- Start transmission:



B-MAC

- Combines several of the above discussed ideas
- Takes care to provide practically relevant solutions

- Clear Channel Assessment
- Adapts to noise floor by sampling channel when it is assumed to be free
- Samples are exponentially averaged, result used in gain control
- For actual assessment when sending a packet, look at five channel samples – channel is free if even a single one of them is significantly below noise
- Optional: random backoff if channel is found busy

- Optional: Immediate link layer acknowledgements for received packets

B-MAC II

- Low Power Listening (= preamble sampling)
- Uses the clear channel assessment techniques to decide whether there is a packet arriving when node wakes up
- Timeout puts node back to sleep if no packet arrived

- B-MAC does not have
- Synchronization
- RTS/CTS
- Results in simpler, leaner implementation
- Clean and simple interface

- Currently: Often considered as the default WSN MAC
- protocol

Power Aware Multiaccess with Signaling – PAMAS

- Idea: combine busy tone with RTS/CTS
- Results in detailed overhearing avoidance, does not address idle listening
- Uses separate data and control channels
- Procedure
- Node A transmits RTS on control channel, does not sense channel
- Node B receives RTS, sends CTS on control channel if it can receive and does not know about ongoing transmissions
- B sends busy tone as it starts to receive data
- Control channel
- Time
- Data channel

Overview

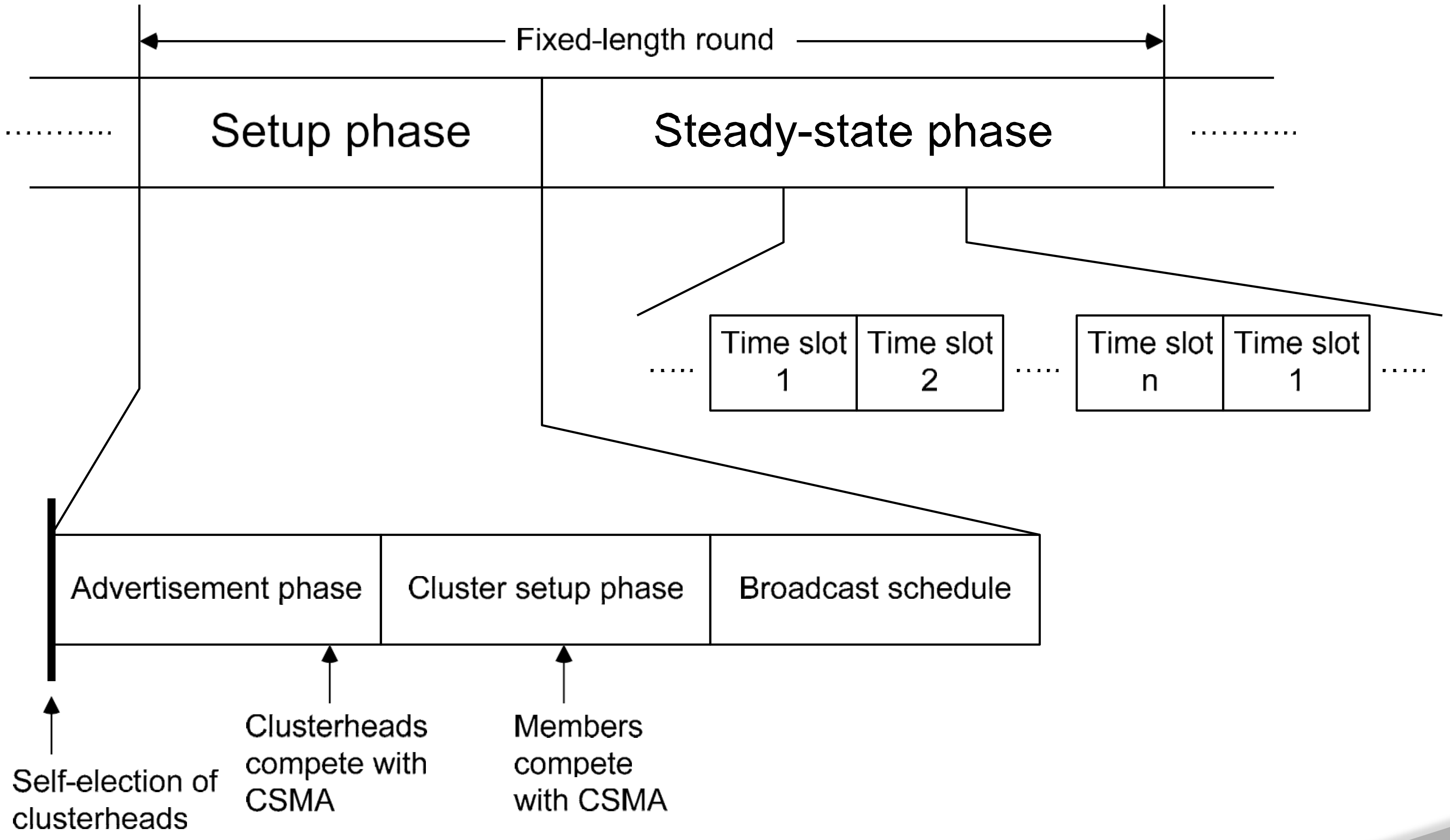
- Principal options and

  difficulties
- Contention-based protocols
- Schedule-based protocols
- LEACH
- SMACS
- TRAMA
- IEEE 802.15.4

Low-Energy Adaptive Clustering Hierarchy (LEACH)

- Given: dense network of nodes, reporting to a central sink, each node can reach sink directly
- Idea: Group nodes into "clusters", controlled by clusterhead
  - Setup phase; details: later
  - About 5% of nodes become clusterhead (depends on scenario)
  - Role of clusterhead is rotated to share the burden
  - Clusterheads advertise themselves, ordinary nodes join CH with strongest signal
  - Clusterheads organize
    - CDMA code for all member transmissions
    - TDMA schedule to be used within a cluster
- In steady state operation
  - CHs collect & aggregate data from all cluster members
  - Report aggregated data to sink using CDMA

LEACH rounds

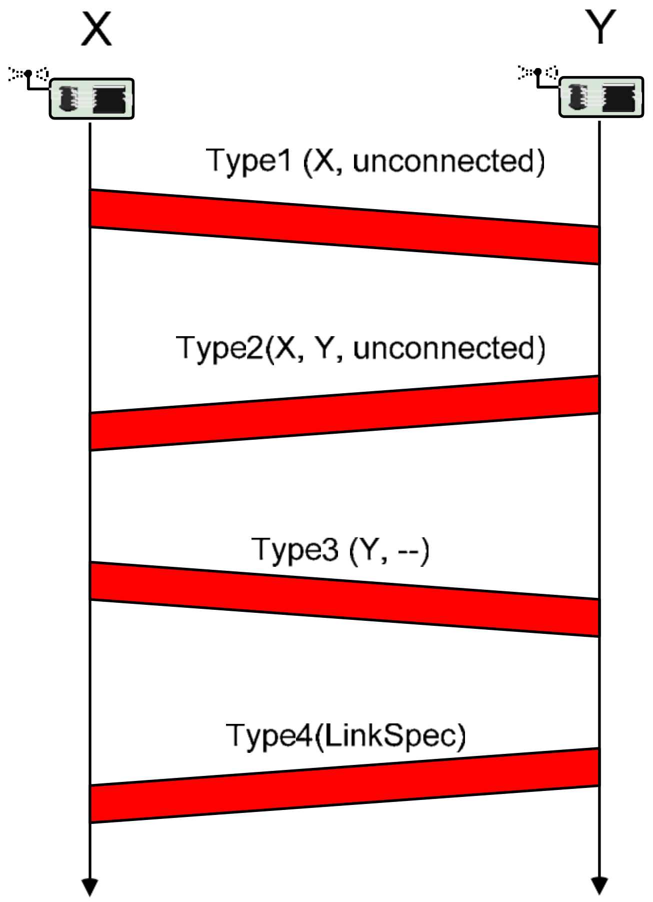


SMACS

- Given: many radio channels, superframes of known length (not necessarily in phase, but still time synchronization required!)
- Goal: set up directional links between neighboring nodes
  - Link: radio channel + time slot at both sender and receiver
  - Free of collisions at receiver
  - Channel picked randomly, slot is searched greedily until a collision- free slot is found
- Receivers sleep and only wake up in their assigned time slots, once per superframe

SMACS link setup

- Case 1: Node X, Y both so far unconnected
  - Node X sends invitation message
  - Node Y answers, telling X that is unconnected to any other node
  - Node X tells Y to pick slot/frequency for the link
  - Node Y sends back the link specification
- Case 2: X has some neighbors, Y not
  - Node X will construct link specification and instruct Y to use it (since Y is unattached)
- Case 3: X no neighbors, Y has some
  - Y picks link specification



Message exchanges protected by randomized backoff

TRAMA

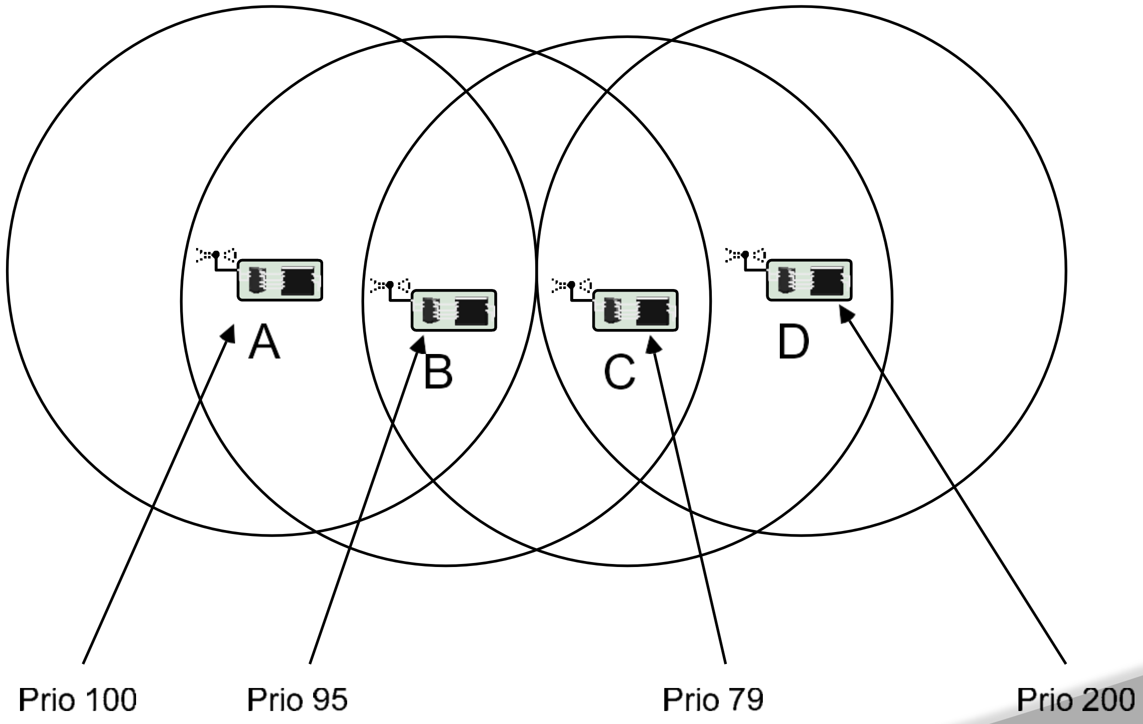
- Nodes are synchronized
- Time divided into cycles, divided into
  - Random access periods
  - Scheduled access periods
- Nodes exchange neighborhood information
  - Learning about their two-hop neighborhood
  - Using neighborhood exchange protocol: In random access period, send small, incremental neighborhood update information in randomly selected time slots
- Nodes exchange schedules
  - Using schedule exchange protocol
  - Similar to neighborhood exchange

TRAMA – adaptive election

- Given: Each node knows its two-hop neighborhood and their current schedules
- How to decide which slot (in scheduled access period) a node can use?
  - Use node identifier x and globally known hash function h
  - For time slot t, compute priority p = h (x © t)
  - Compute this priority for next k time slots for node itself and all two- hop neighbors
  - Node uses those time slots for which it has the highest priority

TRAMA – possible conflicts

- When does a node have to receive?
  - Easy case: one-hop neighbor has won a time slot and announced a packet for it
  - But complications exist – compare example
- What does B believe?
  - A thinks it can send
  - B knows that D has higher priority in its 2-hop neighborhood!
- Rules for resolving such conflicts are part of TRAMA



Comparison: TRAMA, S-MAC

- Comparison between TRAMA & S-MAC
  - Energy savings in TRAMA depend on load situation
  - Energy savings in S-MAC depend on duty cycle
  - TRAMA (as typical for a TDMA scheme) has higher delay but higher maximum throughput than contention-based S-MAC

- TRAMA disadvantage: substantial memory/CPU requirements for schedule computation

Overview

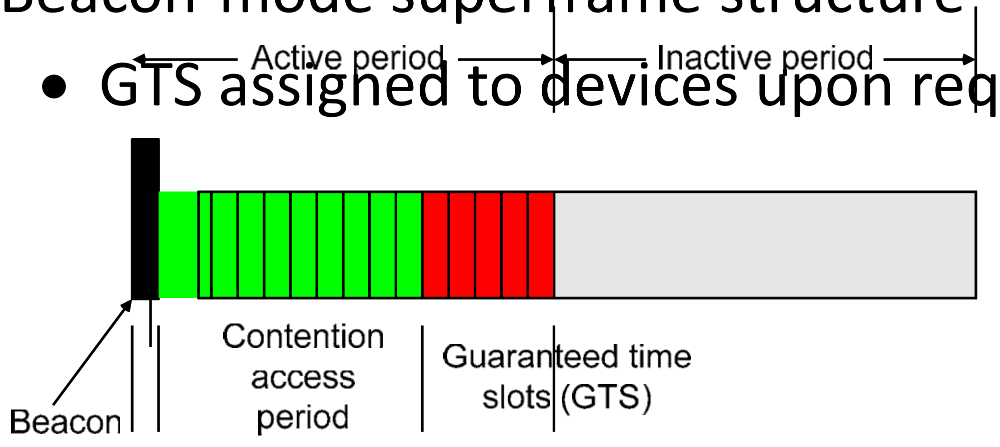
- Principal options and difficulties
- Contention-based protocols
- Schedule-based protocols
- IEEE 802.15.4

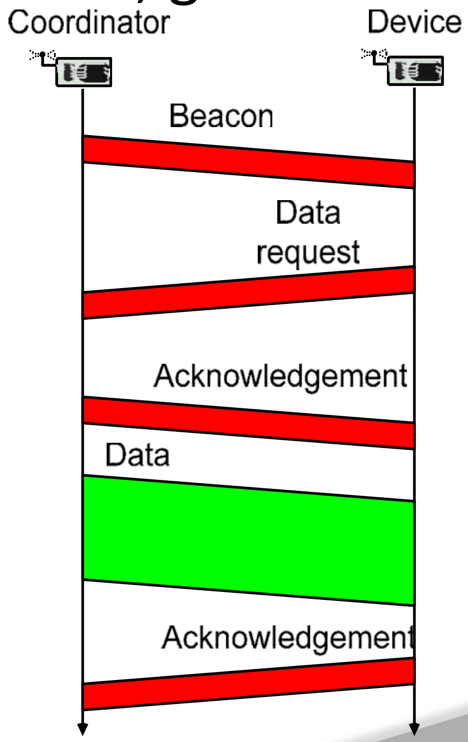
IEEE 802.15.4

- IEEE standard for low-rate WPAN applications
- Goals: low-to-medium bit rates, moderate delays without too stringent guarantee requirements, low energy consumption
- Physical layer
  - 20 kbps over 1 channel @ 868-868.6 MHz
  - 40 kbps over 10 channels @ 905 – 928 MHz
  - 250 kbps over 16 channels @ 2.4 GHz
- MAC protocol
  - Single channel at any one time
  - Combines contention-based and schedule-based schemes
  - Asymmetric: nodes can assume different roles

IEEE 802.15.4 MAC overview

- Star networks: devices are associated with coordinators
  - Forming a PAN, identified by a PAN identifier
- Coordinator
  - Bookkeeping of devices, address assignment, generate beacons
  - Talks to devices and peer coordinators
- Beacon-mode superframe structure
  - GTS assigned to devices upon request





Wakeup radio MAC protocols

- Simplest scheme: Send a wakeup "burst", waking up all neighbors Significant overhearing
- Not quite so simple scheme: Send a wakeup burst including the receiver address
- Additionally: Send information about a (randomly chosen) data channel, CDMA code, … in the wakeup burst
- Various variations on these schemes in the literature, various further problems
  - One problem: 2-hop neighborhood on wakeup channel might be different from 2-hop neighborhood on data channel
  - Not trivial to guarantee unique addresses on both channels

Further protocols

- MAC protocols for ad hoc/sensor networks is one the most active research fields
  - Tons of additional protocols in the literature
  - Examples: STEM, mediation device protocol, many CSMA variants with different timing optimizations, protocols for multi-hop reservations (QoS for MANET), protocols for multiple radio channels, …
  - Additional problems, e.g., reliable multicast

- This chapter has barely scratched the surface…

Geographic Routing

Make use of location information in routing

Assumptions

- Each node knows of its own location.
  - outdoor positioning device:
    - GPS: global positioning system
    - accuracy: in about 5 to 50 meters
  - indoor positioning device:
    - Infrared
    - short-distance radio
- The destination's location is also known.
  - How? (via a location service)

LAR: Location-Aided Routing

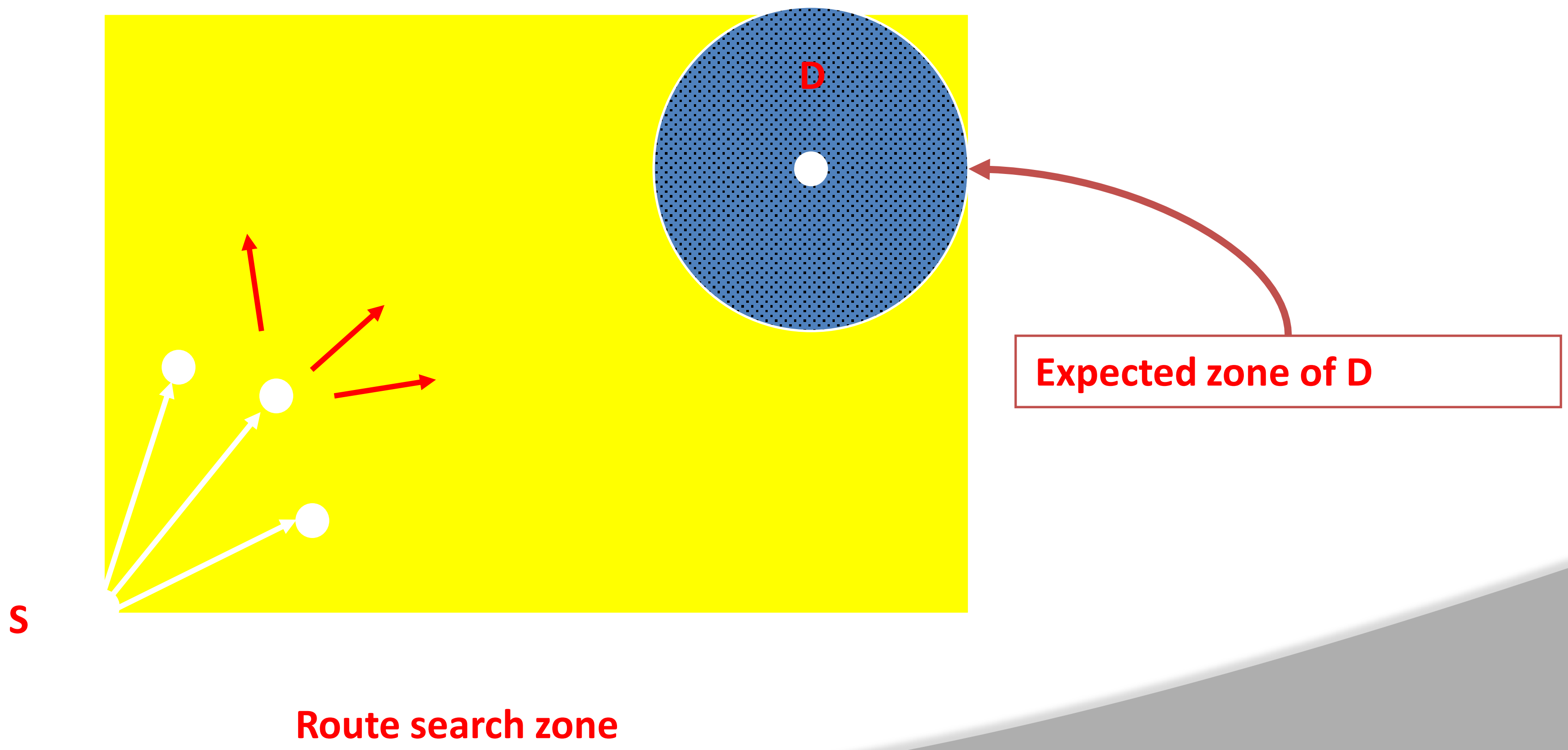
- Location-Aided Routing (LAR) in mobile ad hoc networks
- Young-Bae Ko and Nitin H. Vaidya
- Texas A&M University
- Wireless Networks 6 (2000) 307–321

Basic Idea of LAR

- All packets carry sender’s current location.
- This info enables nodes to learn of each other’s location.

Basic Idea of LAR (cont.)

- Same as DSR, except that if the destination's location is known, the ROUTE_REQ is only flooded over the "route search zone."



DREAM

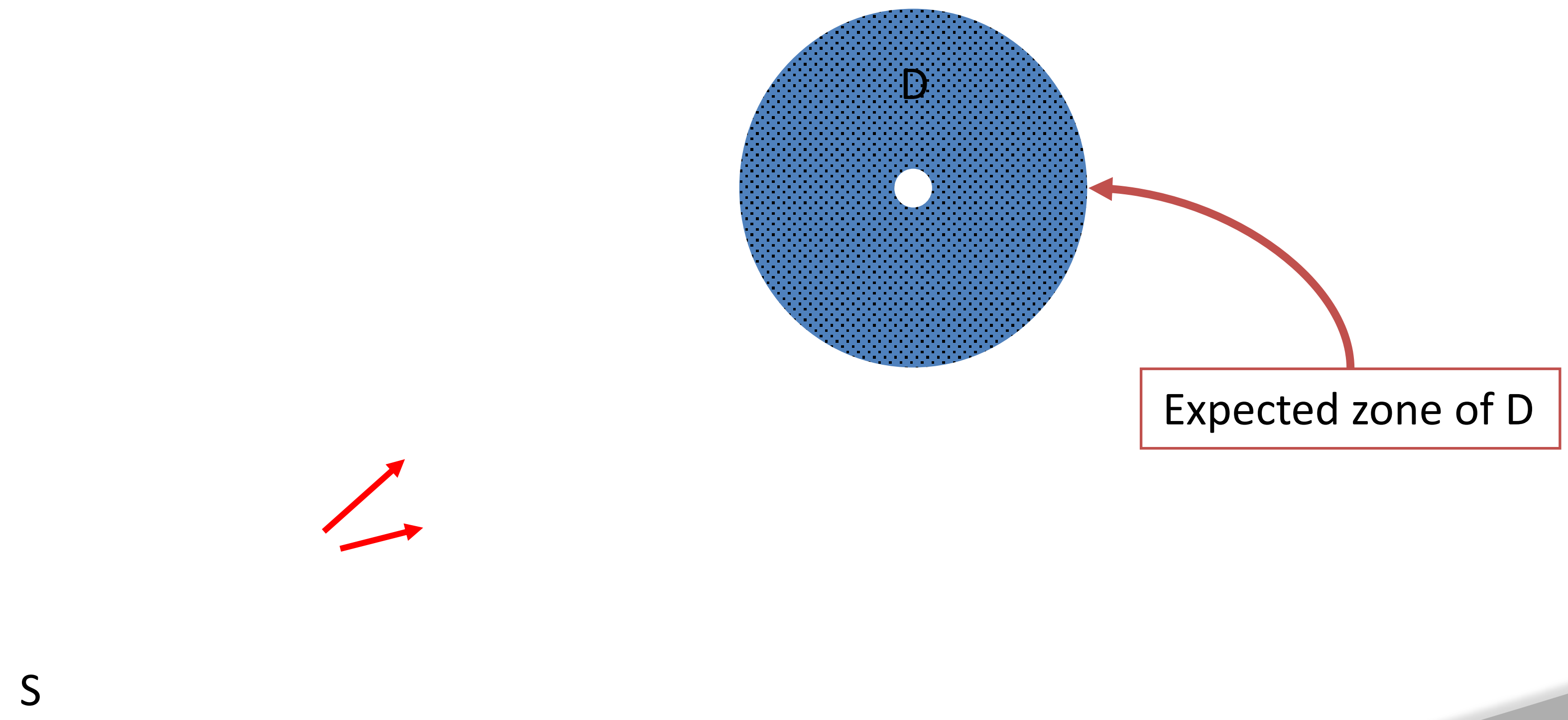
- A Distance Routing Effect Algorithm for Mobility (DREAM)
- S. Basagni, I. Chlamtac, V.R. Syrotiuk, B.A. Woodward
- The University of Texas at Dallas
- Mobicom’98

Basic Idea of DREAM

- Dissemination of location information:
  - Each node periodically advertises its location (and movement information) by flooding.
  - This way, nodes have knowledge of one another’s location.

Basic Idea of DREAM

- **Data Packet carries D’s and S’s locations.**
- **Forwarded toward only a certain direction.**

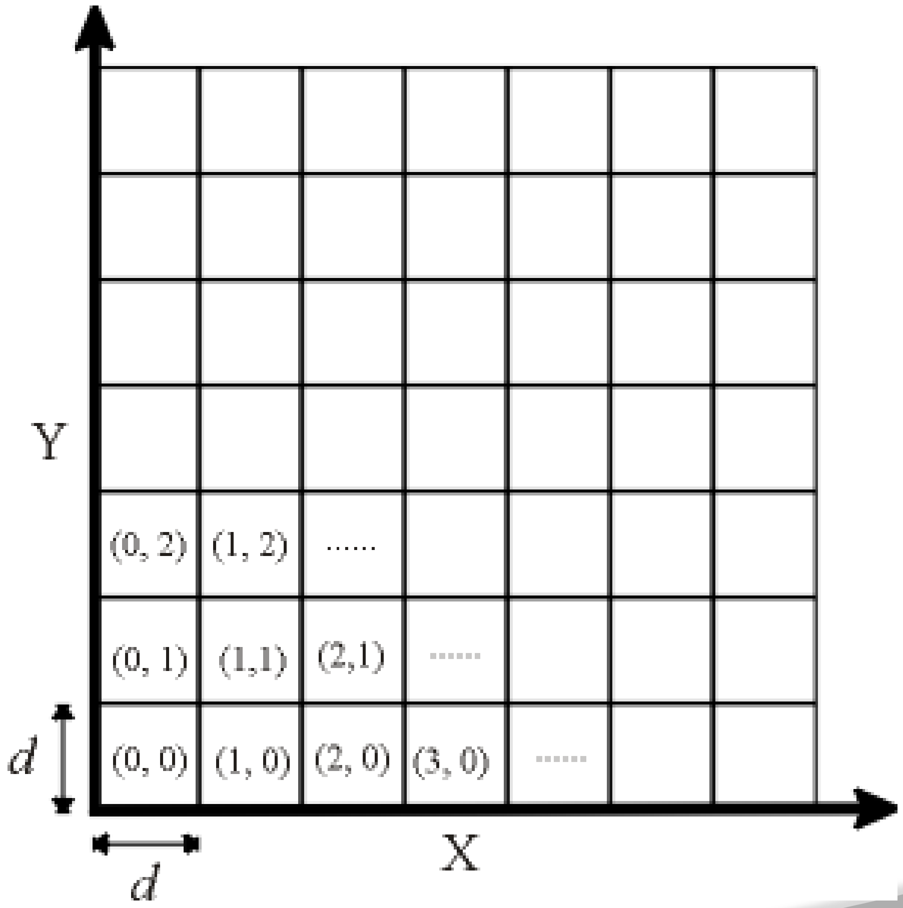


GRID Routing

- “GRID: A Fully Location-Aware Routing Protocol for Mobile Ad Hoc Networks”
- Wen-Hwa Liao, Yu-Chee Tseng, Jang-Ping Sheu
- NCTU
- Telecommunication Systems, 2001.

Basic Idea of GRID Routing

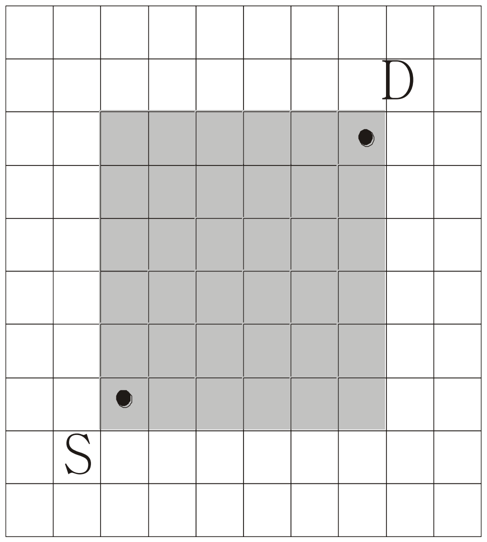
- Partition the physical area into *d* x *d* squares called ***grids***.



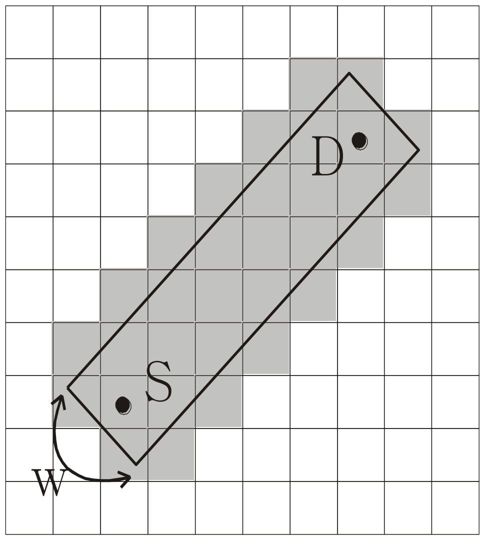
Protocol Overview

- In each grid, a leader is elected, called gateway.
- Responsibility of gateways:
  - forward route discovery packets
  - propagate data packets to neighbor grids
  - maintain routes which passes the grid
- Routing is performed in a grid-by-grid manner.

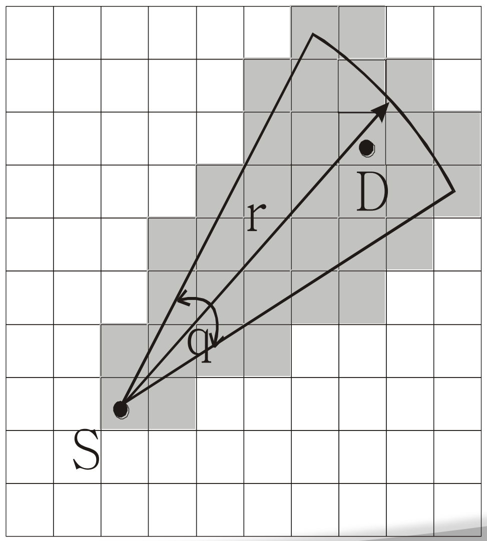
Route Search Range Options



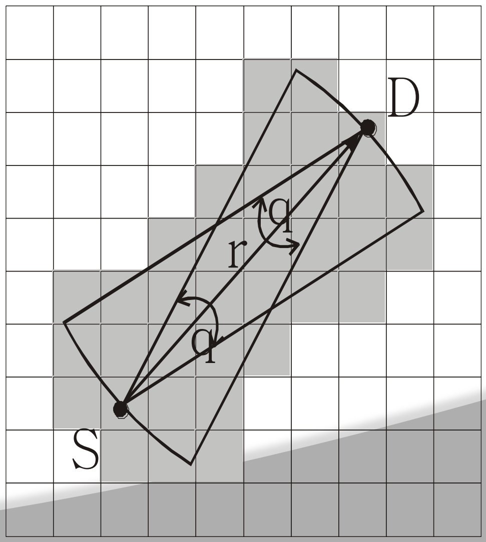
(a) Rectangle



(b) Bar(w)

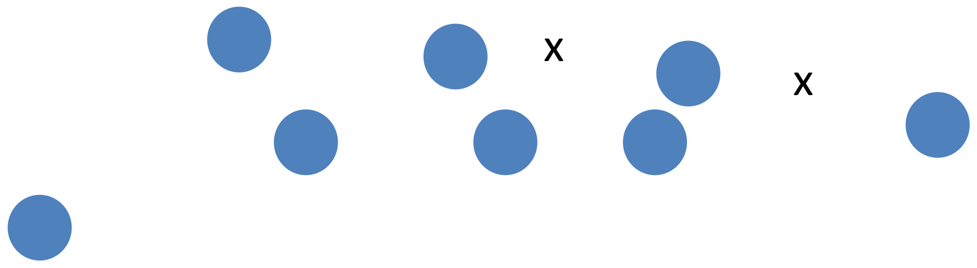


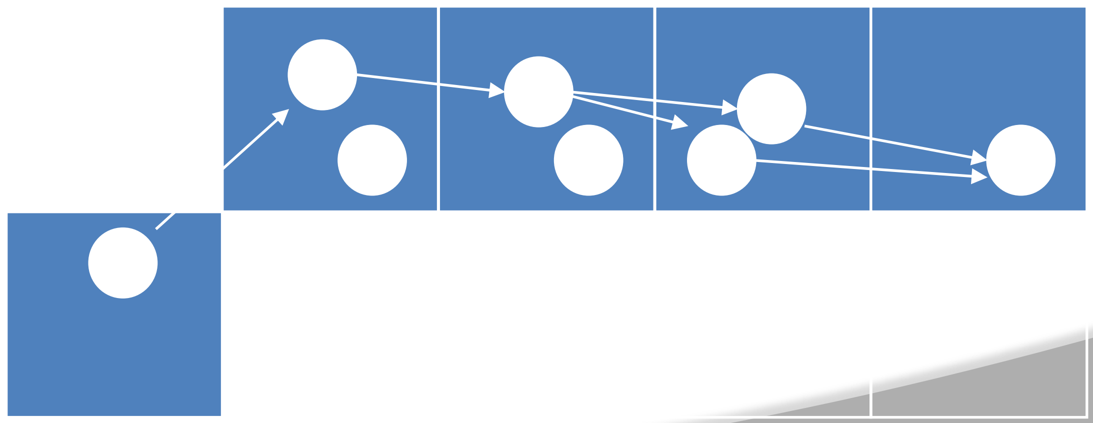
(c) Fan(q, r)



(d) Two_Fan(q, r)

Strength of Grid Routing





Gateway Election in a Grid

- Any “leader election” protocol in distributed computing can be used.
- Multiple leaders in a grid are acceptable.
- Preference in electing a gateway:
  - near the physical center of the grid
    - likely to remain in the grid for longer time
  - once elected, a gateway remains so until leaving the grid

Taxonomy of Geographic Routing Algorithms

- Also called position-based routing
- Three major components of geographic routing:
  - Location services (dissemination of location information)
    - Next topic
  - Forwarding strategies
  - Recovery schemes

Unit-IV
INFRASTRUCTURE ESTABLISHMENT

TOPOLOGY CONTROL

How to set the radio range for each node to use minimize energy ,while still ensuring that the communication graph of the nodes remains connected and satisfies other desirable communication properties.

- Simple case :all nodes must use the same transmission range

The critical transmitting range (CTR) problem define:

- Simple case : All nodes ignore all effects of interference or multi-path and use the same transmission range „ This homogeneous topology
- control setting: how to compute the minimum common transmitting range r such that the network is connected

CLUSTERING

- Clustering allows hierarchical structures to be built on the nodes and enables more efficient use of scare resources ,such as frequency spectrum ,bandwidth and power.

Advantages of clustering :

- Frequency division multiplexing can be reused across non-overlapping cluster
- Clustering allows the health of the network to be monitored and misbehaving node to be identified (watchdog roles in some nodes)
- Network can be comprised of mixtures nodes ,including more powerful or have special capability.

CLUSTERING

Cluster Head: A node declares itself a cluster-head if it has a higher ID than all its uncovered neighbors-neighbors that have not been already claimed by another cluster-head

➢Each node nominates as a cluster-head the highest ID node it can communicate with (including itself) Nominated nodes then form clusters with their nominators.

Time Synchronization: Since the nodes in a sensor network operate independently ,their clock may not be, or stay synchronization with one another. This can cause difficulties when trying to integrate and interpret information sensed at different nodes.

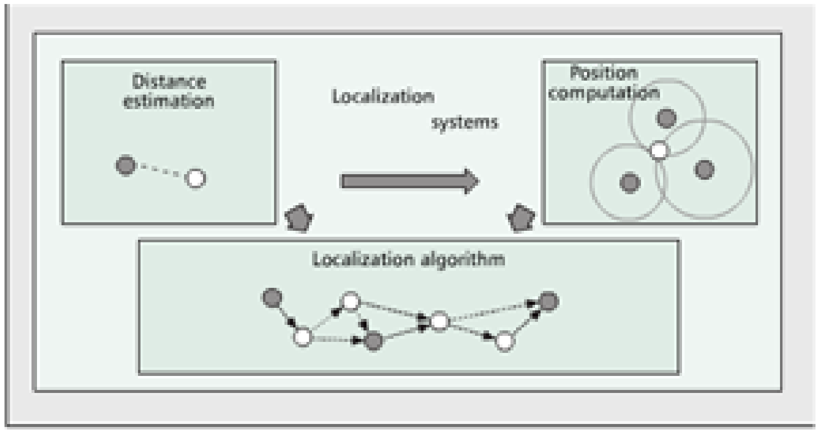
TIME SYNCHRONIZATION

Need for time synchronization:

- Configuring a beam-forming array Setting a TDMA radio schedule
- Synchronization is need for time-of-flight measurements that are then transformed into distances by multiplying with the medium propagation speed for the type of signal used (radio frequency or ultrasonic)
- Time Synchronization is difficult in sensor network , No special master clocks are available, connections are ephemeral, communication delays are inconsistent and unpredictable.

LOCALIZATION AND LOCALIZATION SERVICES

- ***Localization*** is a process to compute the locations of wireless devices in a network
- WSN Composed of a large number of inexpensive nodes that are densely deployed in a region of interests to measure certain phenomenon.
- The primary objective is to determine the location of the target



LOCALIZATION AND LOCALIZATION SERVICES

Distance /Angle Estimation: The distance estimation phase involves measurement techniques to estimate the relative distance between the nodes.

Position Computation: It consists of algorithms to calculate the coordinates of the unknown node with respect to the known anchor node or other neighboring nodes.

Localization Algorithm: Manipulating Available information in order to localize other nodes in wsn.

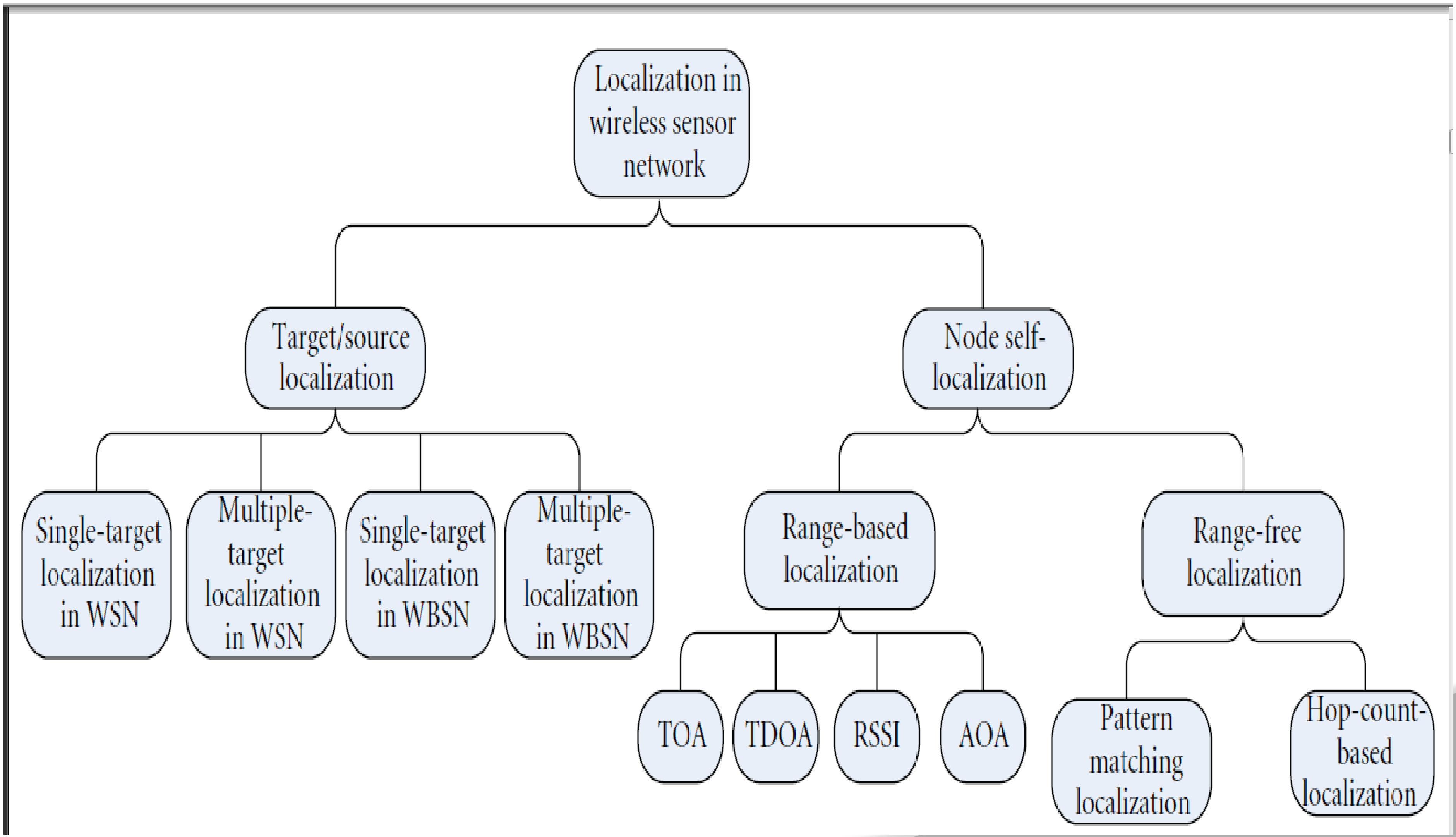
GPS: We need to determine the physical coordinates of a group of sensor nodes in a wireless sensor network (WSN).

Due to application context and massive scale, use of GPS is unrealistic, therefore, sensors need to self-organize a coordinate system.

GPS CONS

- Expensive
- GPS satellite signals are weak (when compared to, say, cellular phone signals), so it doesn't work as well indoors, underwater, under trees, etc.
- The highest accuracy requires line-of-sight from the receiver to the satellite, this is why GPS doesn't work very well in an urban environment
- The US DoD (dept of defense) can, at any given time, deny users use of the system (i.e. they degrade/shut down the satellites)

LOCALIZATION TAXONOMY

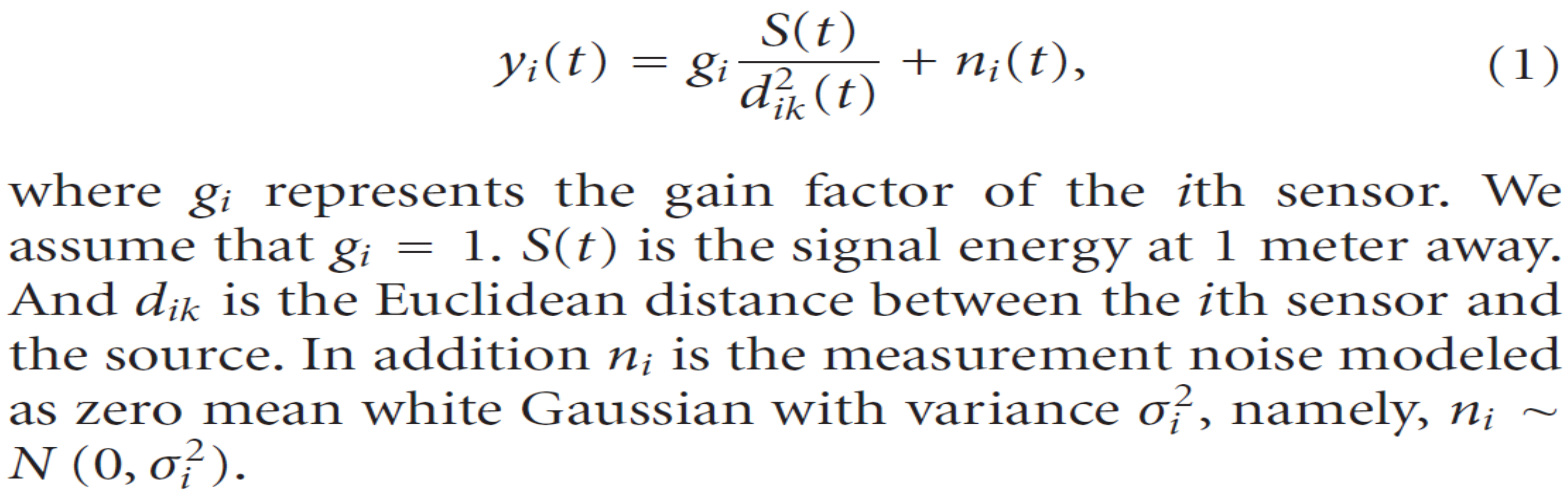

Localization in wireless sensor network
Target/source localization
Node self-localization
Single-target localization in WSN
Multiple-target localization in WSN
Single-target localization in WBSN
Multiple-target localization in WBSN
Range-based localization
Range-free localization
TOA
TDOA
RSSI
AOA
Pattern matching localization
Hop-count-based localization

Target/Source Localization

1- Target/Source Localization: Most of the source localization methods are focused on the measured signal strength.

➢To obtain the measurements, the node needs complex calculating process.

➢The received signal strength of single target/source localization in WSN during time interval t:



$$y_i(t) = g_i \frac{S(t)}{d_{ik}^2(t)} + n_i(t), \tag{1}$$

where g_i represents the gain factor of the ith sensor. We assume that $g_i = 1$. $S(t)$ is the signal energy at 1 meter away. And d_{ik} is the Euclidean distance between the ith sensor and the source. In addition n_i is the measurement noise modeled as zero mean white Gaussian with variance σ_i^2, namely, $n_i \sim N\left(0, \sigma_i^2\right)$.

Target/Source Localization

➢The received signal strength of multiple target/source localization in WSN during time interval t:

$$y_i(t) = g_i \sum_{k=1}^{K} \frac{S_k(t)}{d_{ik}^{\alpha}(t)} + \varepsilon_i(t), \qquad (2)$$

where $d_{ik}(t)$ is the distance between the ith sensor and the kth source. K is the number of the sources. g_i is the gain of ith sensor. $\varepsilon_i(t)$ is random variable with mean μ_i and variance σ_i^2. $S_k(t)$ is the signal energy at 1 meter away for kth source. α is the attenuation exponent.

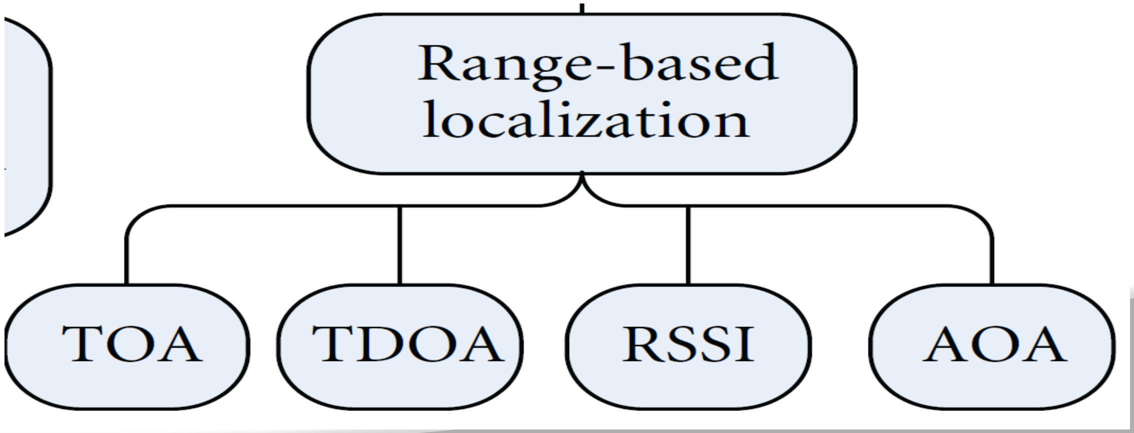
Target/Source Localization

- The Above methods require transmission of a large amount of data from sensors which may not be feasible under communication constraints.
- The binary sensors sense signals (infrared, acoustic, light, etc.) from their vicinity, and they only become active by transmitting a signal if the strength of the sensed signal is above a certain threshold.
- The binary sensor only makes a binary decision (detection or non-detection) regarding the measurement.
- Consequently, only its ID needs to be sent to the fusion center when it detects the target. Otherwise, it remains silent.
- So, the binary sensor is a low-power and bandwidth-efficient solution for WSN.

NODE SELF-LOCALIZATION

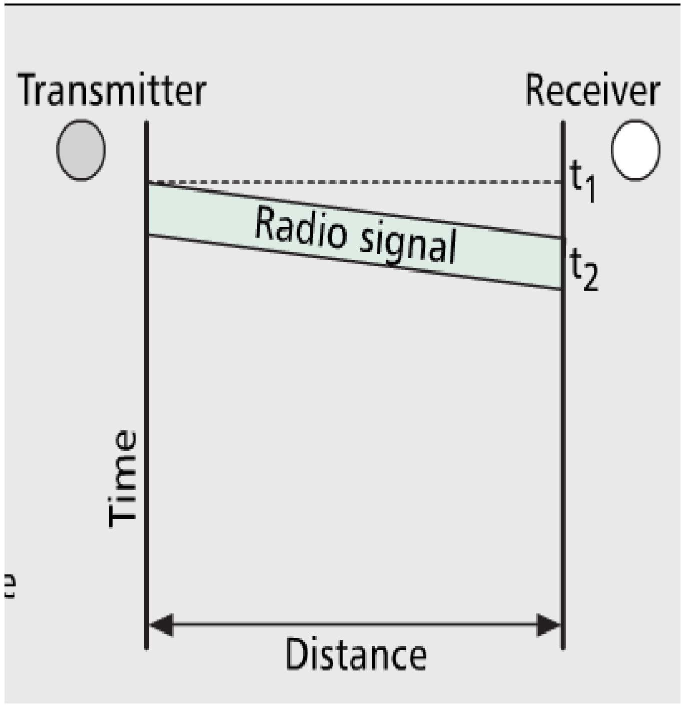
Range-based Localization: uses the measured distance/angle to estimate the indoor location using geometric principles.

Range-free Localization: uses the connectivity or pattern matching method to estimate the location. Distances are not measured directly but hop counts are used. Once hop counts are determined, distances between nodes are estimated using an average distance per hop and then geometric principles are used to compute location.



RANGE BASED LOCALIZATION

- Time of arrival: (TOA): It’s a method that tries to estimate distance between 2 nodes using time based measures.
- Accurate but needs synchronization

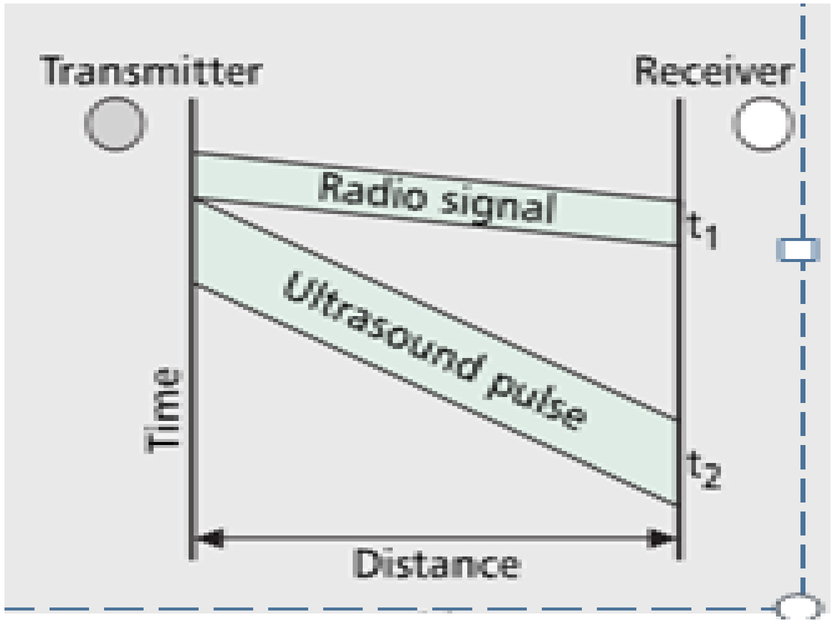


RANGE BASED LOCALIZATION

Time Difference Of Arrival: (TDOA)

It’s a method for determining the distance between a mobile station and a nearby synchronized base station. (Like AT&T)

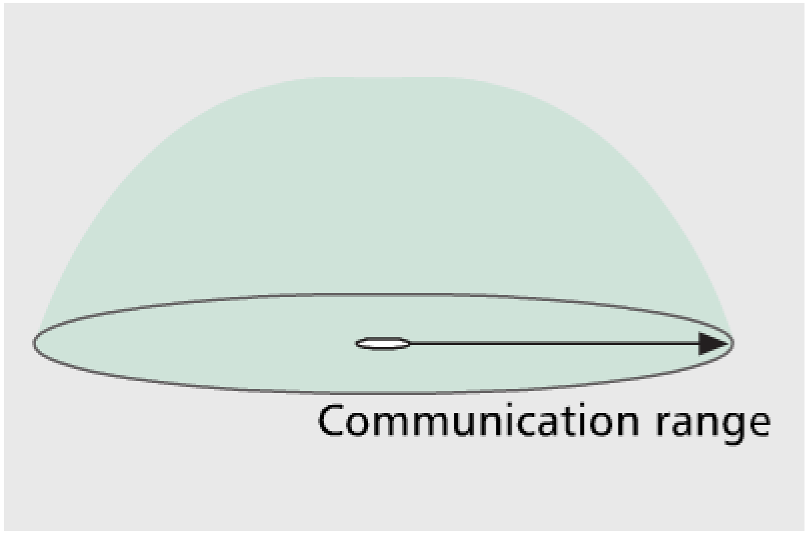
No synchronization needed but costly.



RANGE BASED LOCALIZATION

Received Signal Strength Indicator: (RSSI) Techniques to translate signal strength into distance

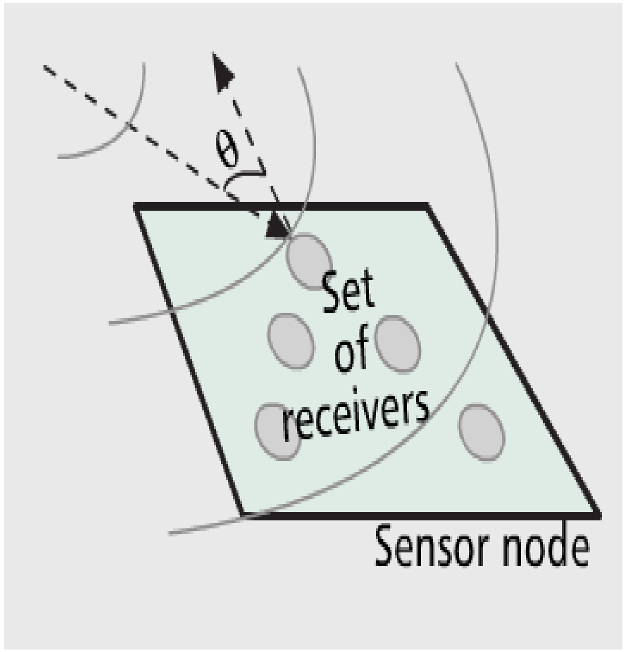
➢Low cost but very sensitive to noise



RANGE BASED LOCALIZATION

Angle Of Arrival: (AOA) It's a method that allows each sensor to evaluate the relative angles between received radio signals.

➢Costly and needs extensive signal processing.



RANGING TECHNIQUES

- Ranging methods aim at estimating the distance of a receiver to a transmitter The first way Use RSS (received signal strength) along with for signal strength as a function of distance to estimate its distance from sender to receiver .
- Localization to within a few meters is the best that can currently be attained with RSS method

RSS method con's:

- The distance to estimate not very accurate
- fading
- Shadowing
- Multi-path effect
- Not well-RF component

RANGE-FREE LOCALIZATION

- DV-Hop is the typical representation
- It doesn’t need to measure the absolute distance between the beacon node and unknown node. It uses the average hop distance to approximate the actual distances and reduces the hardware requirements.

Adv: Easy to implement and applicable to large network.

Disadv: The positioning error is correspondingly increased.

DV-HOP

It is divided into 3 stages:

1. Information broadcast
2. Distance calculation
3. Position estimation

Information broadcast : It includes hop count and is initialized to zero for their neighbors.

➢The receiver records the minimal hop of each beacon nodes and ignores the larger hop for the same beacon nodes.

➢The receiver increases the hop count by 1 and transmits it to neighbor nodes.

➢All the nodes in a network can record the minimal hop counts of each beacon nodes.

DV-HOP

Distance calculation:

➢According to the position of the beacon node and hop count, each beacon node uses the following equation to estimate the actual distance of every hop

$$\mathrm{HopSize}_i = \frac{\sum_{j \neq i} \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2}}{\sum_{j \neq i} h_j}, \qquad (10)$$

where (x_i, y_i) and (x_j, y_j) are the coordinates of beacon nodes i and j, respectively. h_j is the hop count between the beacon nodes. Then, beacon nodes will calculate the

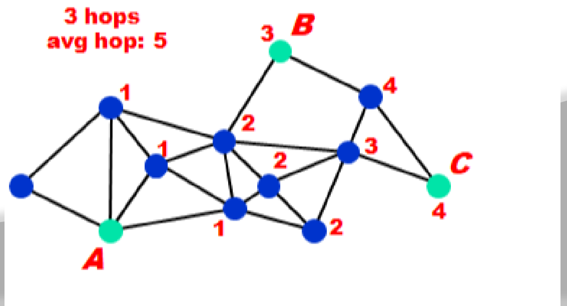
D0V-HOP

Position estimation: The beacon node will calculate the average distance and broadcast the information to network.

- The unknown nodes only record the first average distance and then transmit it to neighbor nodes.
- The unknown node calculates its location through.

- Anchors flood network with own position flood network with avg hop distance.
- Nodes count number of hops to anchors multiply with avg hop distance

- A-B: 15



TASK-DRIVEN SENSING

- To efficiently and optimally utilize scarce resources(e.g., limited on-board battery and limited communication bandwidth) in a sensor network, sensor nodes must carefully tasked and controlled to carry out the required set of tasks.
- A utility-cost-based approach to distributed sensor network management is to address the balance between utilityand resource costs.
- Utility –the total utility of the data
- Cost –power supply, communication bandwidth

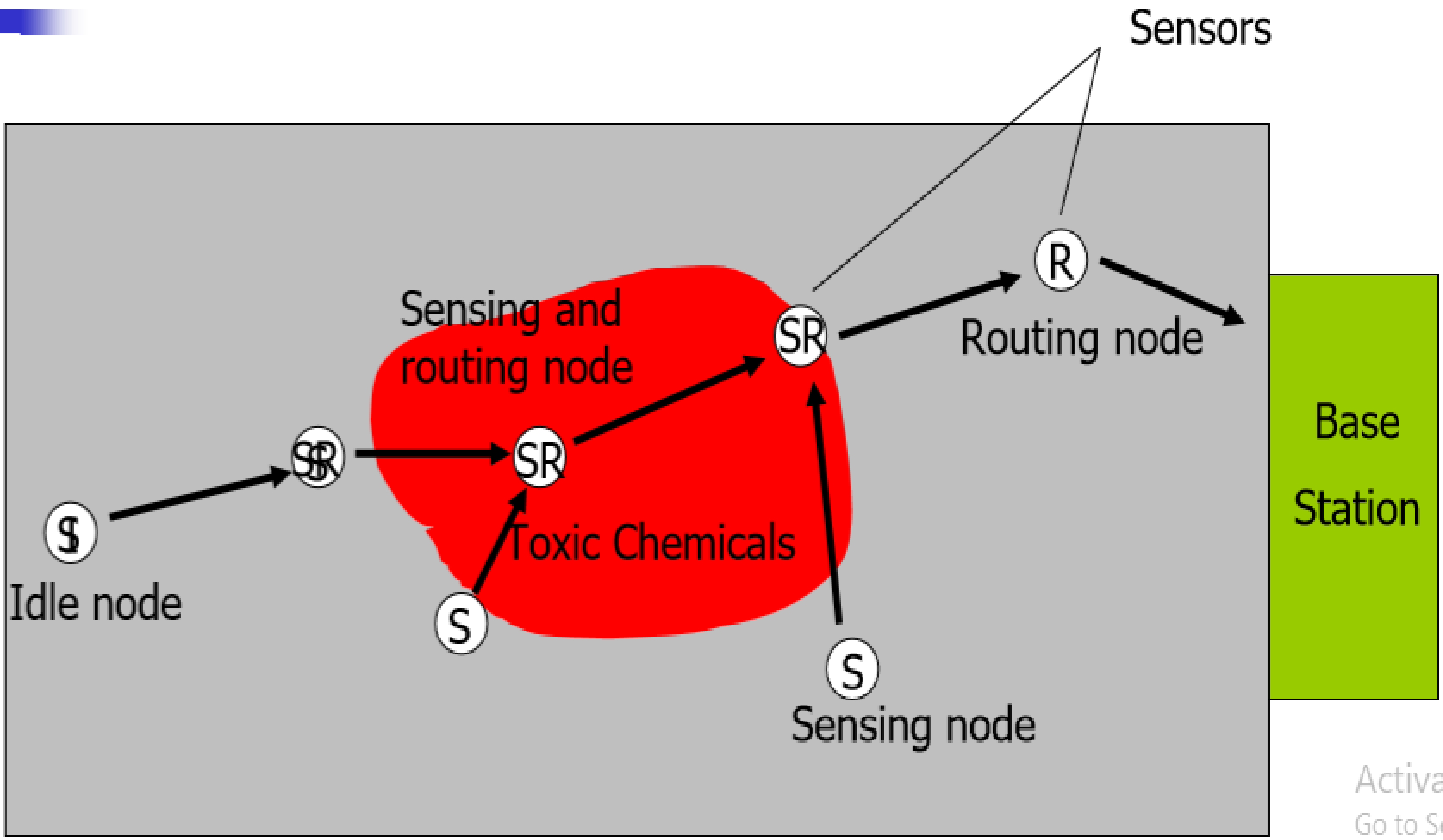
TASK-DRIVEN SENSING

- A sensor may take on a particular role depending on the application task requirement and resource availability such as node power levels.

Example:

- Nodes, denoted by SR, may participate in both sensing and routing.
- Nodes, denoted by S, may perform sensing only and transmit their data to other nodes.
- Nodes, denoted by R, may decide to act only as routing nodes, especially if their energy reserved is limited.
- Nodes, denoted by I, may be in idle or sleep mode, to preserve energy.

TASK-DRIVEN SENSING



T0 =T0+∆T

GENERIC MODEL OF UTILITY AND COST

Utility: We can define a utility function that assigns a scalar value, or utility, to each data reading of a sensing node. The maximum utility over a period of time is

$$\text{Max} \sum_{t} \sum_{i \in V_s(t)} \in U(i,t)$$

where i is sensor index and the set of nodes performing a sensing operation at time t as Vs(t).

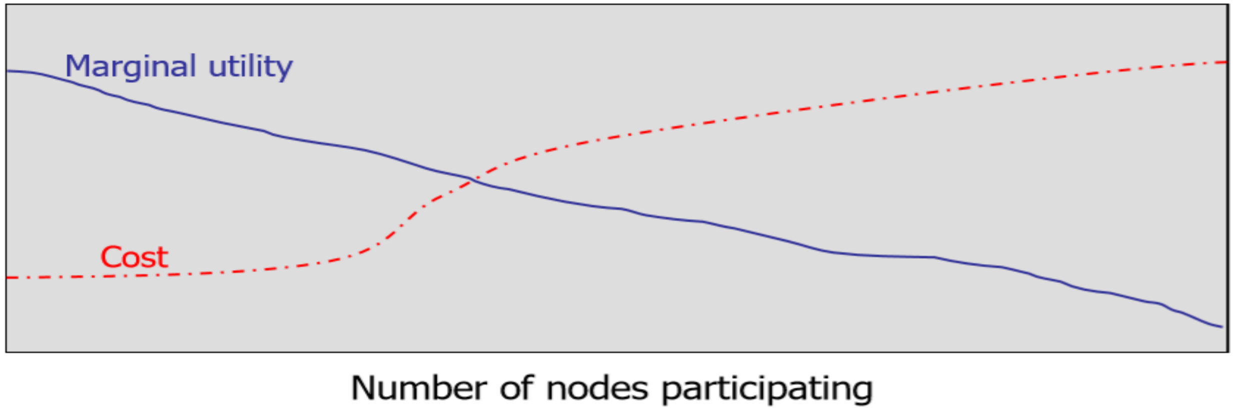
GENERIC MODEL OF UTILITY AND COST

The constraint is defined as

$$\sum_{t} \sum_{V_s(t)} C_s + \sum_{t} \sum_{V_t(t)} (C_t + C_r) + \sum_{t} \sum_{V_a(t)} C_a \leq C_{total}$$

receiving nodes as Vr(t).

➢More nodes are added, the benefit often becomes less and less significant



GENERIC MODEL OF UTILITY AND COST

Cost:

We can assigned a cost to each sensor operation.

Example:

Cs: the cost of a sensing operation

Ca: the cost of data aggregation

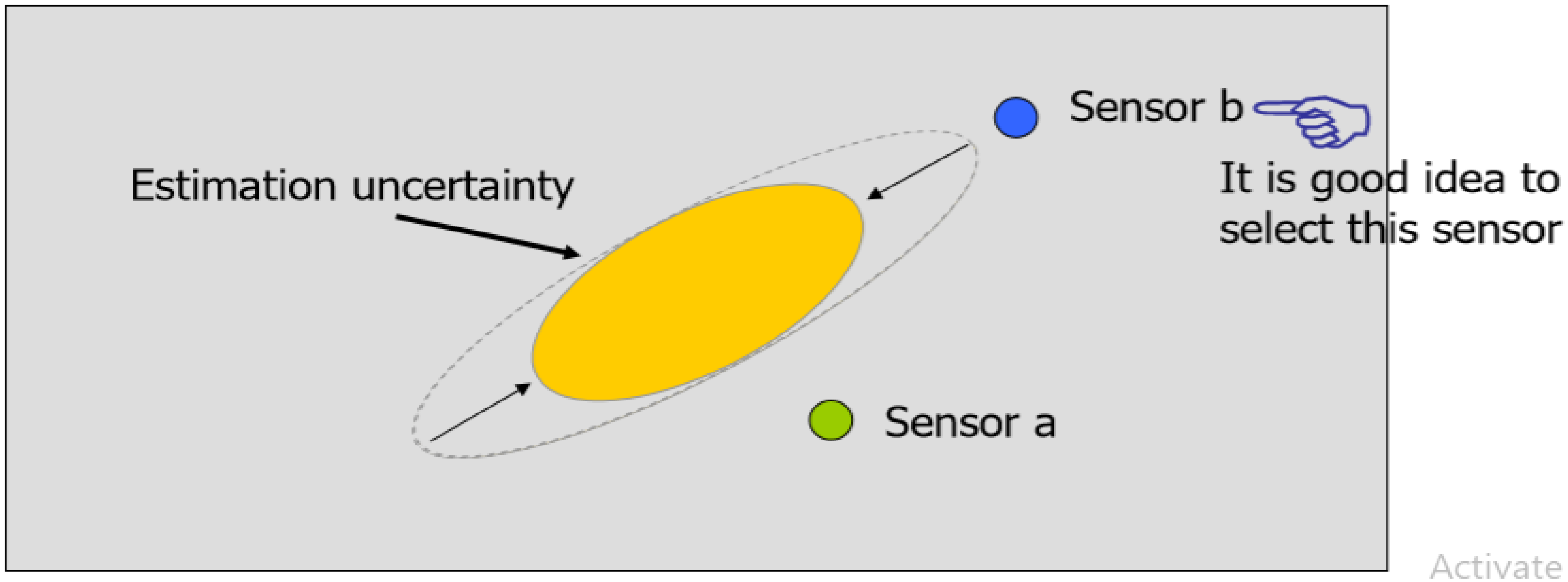
Ct: the cost of data transmission

Cr: the cost of data reception

INFORMATION-BASED SENSOR TASKING

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

SENSOR SELECTION



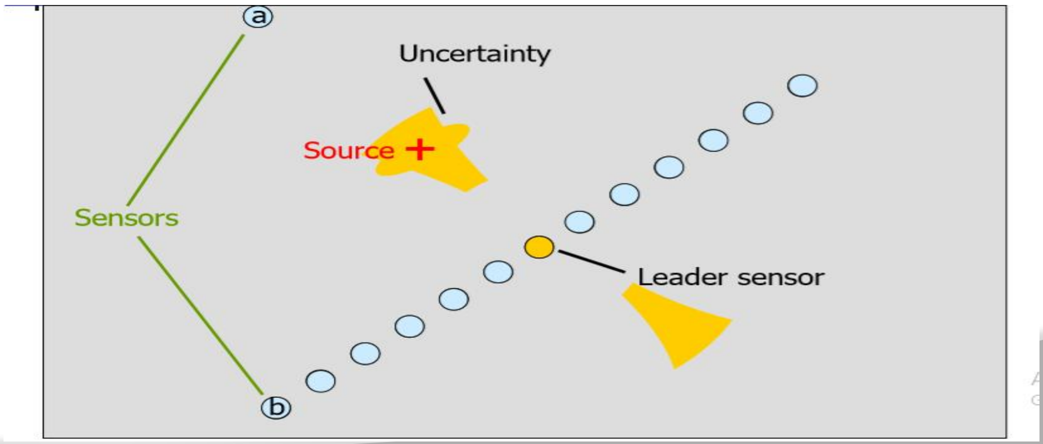
Sensor selection based on information gain of individual sensor contributions

SENSOR SELECTION

The estimation uncertainty can be effectively approximated by a Gaussian distribution, illustrated by uncertainty ellipsoids in the state space.

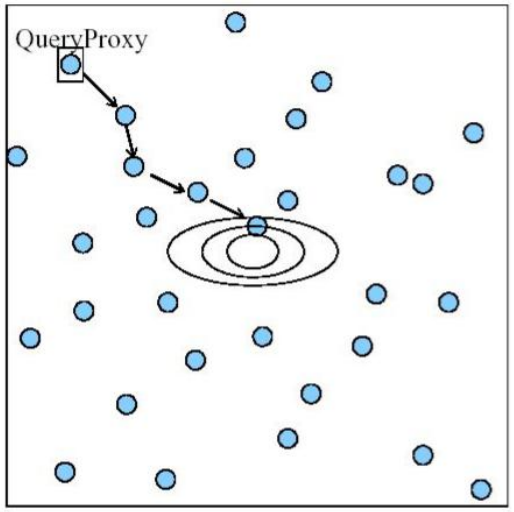
Sensor b would provide better information than a because sensor b lies close to the longer axis of the uncertainty ellipsoid and its range constraint would intersect this longer axis transversely.

Scenario: Localizing a Stationary Source



Joint Routing and Information Aggregation

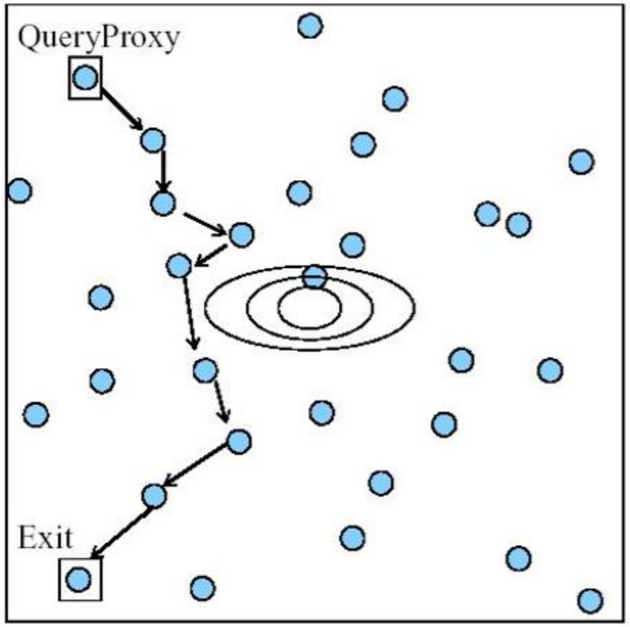
Our primary purpose is to collect and aggregate information. IDSQ just only provide us with a method to obtain max. incremental information gain. This section outlines some techniques to dynamically determine the optimal routing path.



Routing from a query proxy to the high activity region and back.

Joint Routing and Information Aggregation

The ellipses represent iso-contours of an information field. The goal of routing is to maximally aggregate information. This differs from routing in communication networks where the destination is often known a priori to the sender.



Routing from a query proxy to an exit node

Joint Routing and Information Aggregation

The routing has to maximize information gain along the path.

A path toward the high information region may be more preferable than the shortest path

Individual sensors direct and guide the query by maximizing the objective function J.

The local decisions can be based on 4 different criteria.

$$J\left(p\left(x \middle| \{z_i\}_{i\in U} \cup \{z_j\}\right)\right)$$
$$= \gamma \cdot \phi\left(p\left(x \middle| \{z_i\}_{i\in U} \cup \{z_j\}\right)\right) - (1-\gamma)\cdot \psi(z_j)$$

Joint Routing and Information Aggregation

➢each current sensor k evaluate the objective function J, and pick the sensor j that maximizes the objective function.

➢ζjis the position of the node j.

$$\hat{j} = \arg\max_{j} \left(J\left(\zeta_j\right) \right), \forall j \neq k$$

➢Choose the next routing sensor in the direction of the gradient of the objective function, ∇J .

➢ζkis the position of the current routing node.

$$\hat{j} = \arg\max_{j} \left(\frac{(\nabla J)^T \bullet (\zeta_j - \zeta_k)}{\|\nabla J\| \; \|\zeta_j - \zeta_k\|} \right)$$

UNIT-V

SENSOR NETWORK PLATFORM AND TOOLS

Sensor node hardware

- Sensor node hardware can be grouped into three categories
  - Augmented general-purpose computers
  - Dedicated embedded sensor nodes
  - System-on-chip (SoC)

Augmented general-purpose computers

- Off-the-shelf operating systems such as WinCE, Linux and with standard wireless communication protocols such as 802.11 or Bluetooth.
- Relatively higher processing capability
- More power hungry
- Fully supported popular programming languages
- Ex: PDAs

Dedicated embedded sensor nodes

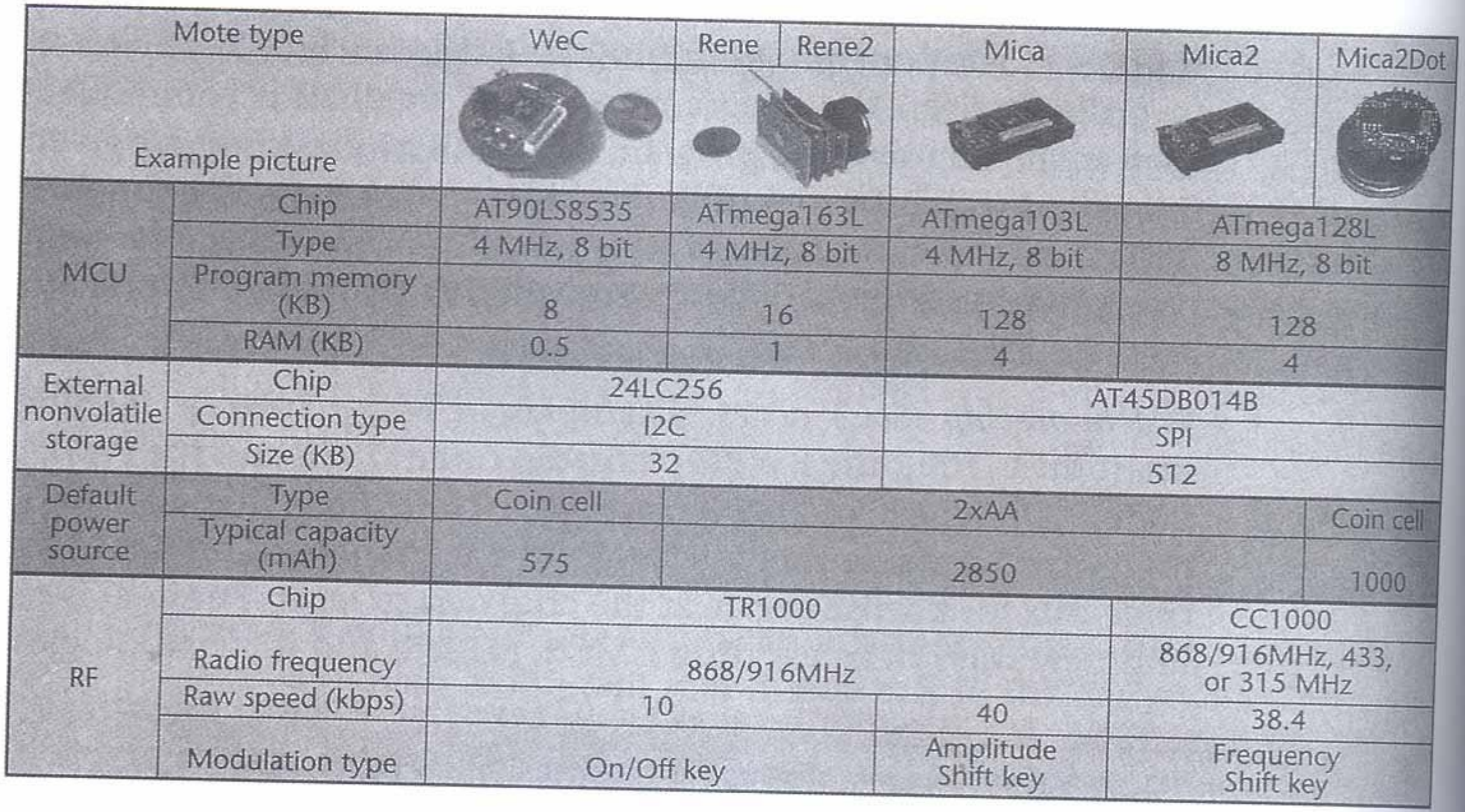
- In order to keep the program footprint small to accommodate their small memory size, programmers of these platforms are given full access to hardware but barely any operating system support.
- Typically support at least one programming language, such as C.
- Ex: mica, TinyOS, nesC

System-on-chip (SoC)

- Build extremely low power and small footprint sensor nodes that still provide certain sensing, computation, and communication capabilities.

- Currently in the research pipeline with no predefined instruction set, there is no software platform support available.

Berkley motes



Mote type		WeC	Rene	Rene2	Mica	Mica2	Mica2Dot
Example picture							
MCU	Chip	AT90LS8535	ATmega163L		ATmega103L	ATmega128L	
	Type	4 MHz, 8 bit	4 MHz, 8 bit		4 MHz, 8 bit	8 MHz, 8 bit	
	Program memory (KB)	8	16		128	128	
	RAM (KB)	0.5	1		4	4	
External nonvolatile storage	Chip	24LC256			AT45DB014B		
	Connection type	I2C			SPI		
	Size (KB)	32			512		
Default power source	Type	Coin cell	2xAA				Coin cell
	Typical capacity (mAh)	575	2850				1000
RF	Chip	TR1000				CC1000	
	Radio frequency	868/916MHz				868/916MHz, 433, or 315 MHz	
	Raw speed (kbps)	10			40	38.4	
	Modulation type	On/Off key			Amplitude Shift key	Frequency Shift key	

Figure 7.1 A comparison of Berkeley motes.

Sensor Network Programming Challenges

- Event-driven execution allows the system to fall into low-power sleep mode when no interesting events need to be processed.

- At the extreme, embedded operating systems tend to expose more hardware controls to the programmers, who now have to directly face device drivers and scheduling algorithms, and optimize code at the assembly level.

Node-level software platforms

- Node-centric design methodologies: Programmers think in terms of how a node should behave in the environment.

- A node-level platform can be a node-centric OS, which provides hardware and networking abstractions of a sensor node to programmers.

Node-level software platforms

TinyOS

- No file system
- Static memory allocation: analyzable, reduce memory management overhead
- Only parts of OS are compiled with the application

Node-level software platforms

TinyOS

- A program executed in TinyOS has two contexts, tasks and events.
- Tasks are posted by components to a task scheduler. Without preempting or being preempted by other tasks
- Triggered events can be preempted by other events and preempt tasks

Node-level software platforms

- Split-phase operation
- Command send() €   eventsendDone()
- Avoid blocking the entire system
- Not accepting another packet Until sendDone() is called, avoid race condition

Node-level software platforms

Imperative Language: nesC

- nesC is an extension of C to support the design of TinyOS.
- A component has an interface specification and an implementation.
- A component specification is independent of the component implementation.
- A provides interface is a set of method calls exposed to the upper layers.
- A uses interface is a set of method calls hiding the lower layer components.

Node-level software platforms

nesC

- An event call is a method call from a lower layer component to a higher layer component. (signal)
- A command is the opposite. (call)
- A component may use or provide the same interface multiple times. Give each interface instance a separate name using as notation.

Node-level software platforms

nesC– component implementation

- There are two types of components in nesC, depending on how they are implemented: modules and configurations.
- Modules are implemented by application code.
- Configurations are implemented by connecting interfaces of existing components.
  - A.a=B.a, the interface a of A is the interface a of B
  - A.a->B.a, interface is hidden from upper layers

Node-level software platforms

nesC

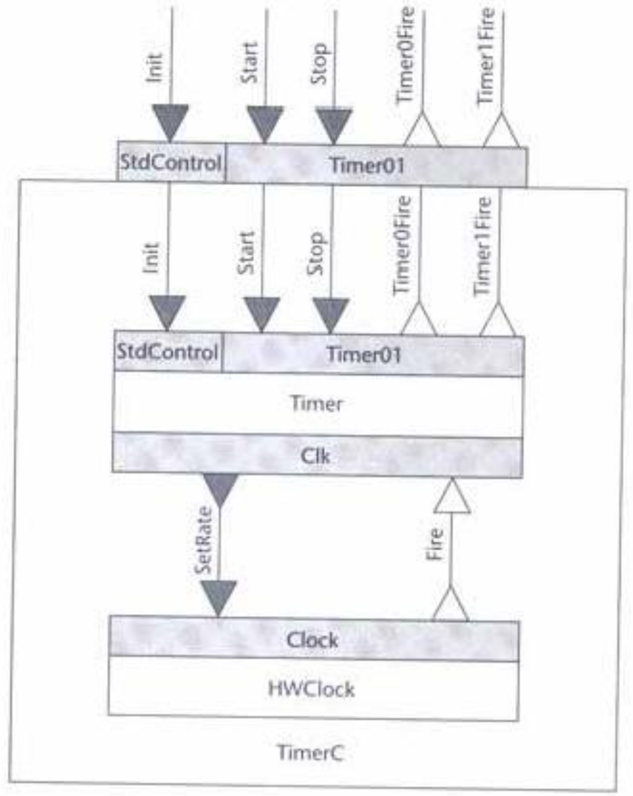


Figure 7.9 The TimerC configuration implemented by connecting Timer with HWClock.

Node-level software platforms

nesC

- An application must contain the Main module which links the code to the scheduler at run time.
- The Main has a single StdControl interface, which is the ultimate source of initialization of all components.

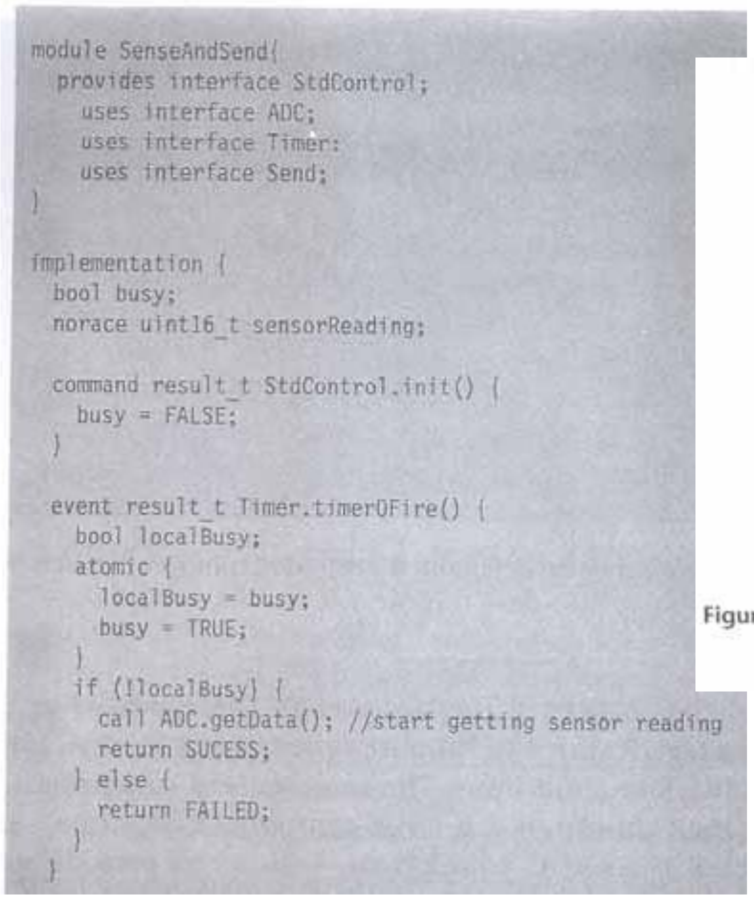
Node-level software platforms

nesC—concurrency and atomicity

- A keyword atomic to indicate that the execution of a block of statements should not be preempted.
- Method calls are not allowed in atomic block.
- A shared variable x is outside of an atomic statement is a compile-time error.
- A norace declaration of the variable can prevent the compiler from checking the race condition on that variable.

Node-level software platforms

nesC—concurrency and atomicity

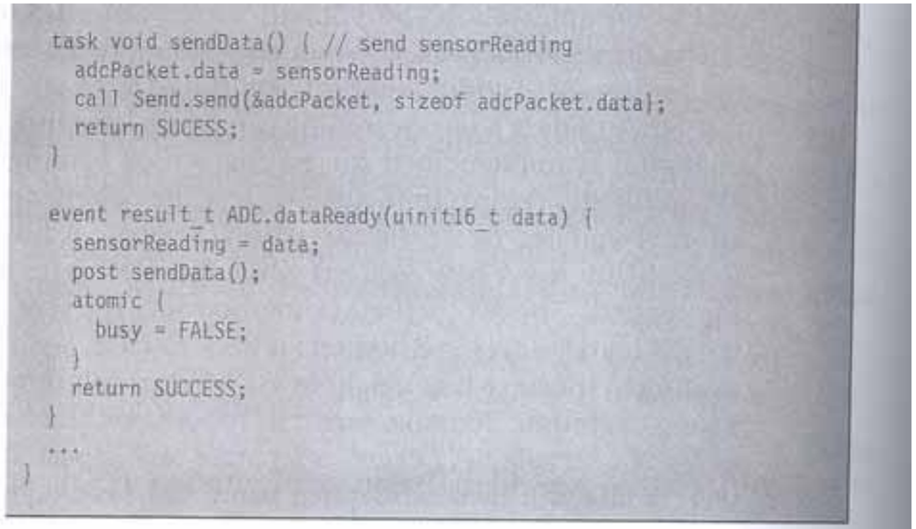


```
module SenseAndSend{
  provides interface StdControl;
    uses interface ADC;
    uses interface Timer;
    uses interface Send;
}

implementation {
  bool busy;
  norace uint16_t sensorReading;

  command result_t StdControl.init() {
    busy = FALSE;
  }

  event result_t Timer.timer0Fire() {
    bool localBusy;
    atomic {
      localBusy = busy;
      busy = TRUE;
    }
    if (!localBusy) {
      call ADC.getData(); //start getting sensor reading
      return SUCESS;
    } else {
      return FAILED;
    }
  }
```



```
  task void sendData() { // send sensorReading
    adcPacket.data = sensorReading;
    call Send.send(&adcPacket, sizeof adcPacket.data);
    return SUCESS;
  }

  event result_t ADC.dataReady(uint16_t data) {
    sensorReading = data;
    post sendData();
    atomic {
      busy = FALSE;
    }
    return SUCCESS;
  }
  ...
}
```

Figure 7.11 A section of the implementation of SenseAndSend, illustrating the handling of concurrency in nesC.

Node-level software platforms

Dataflow-style language: TinyGALS

- Dataflow languages are intuitive for expressing computation on interrelated data units by specifying data dependencies among them.
- A data flow program has a set of processing units called actors.
- Actors have ports to receive and produce data.

Node-level software platforms

TinyGALS

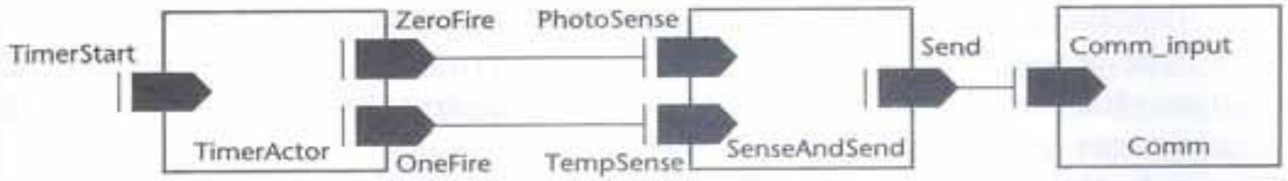


Figure 7.14 Triggering, sensing, and sending actors of the FieldMonitor in TinyGALS.

```
Application FieldMonitor {
  include actors {
    TimerActor;
    SenseAndSend;
    Comm;
  }
  implementation {
    zeroFire => photoSense 5;
    oneFire => tempSense 5;
    send => comm_input 10;
  }
  START@ timerStart;
}
```

Figure 7.15 Implementation of the FieldMonitor in TinyGALS.

Node-Level Simulators

Node-Level Simulators

- For engineer to perform performance study, which in terms of
  - Power
  - Bandwidth
  - Etc

Node-Level Simulators

Node-Level Simulators

- Simulators are consisted by the following models
    - Sensor node model
    - Communication model
    - Physical environment model
    - Statistics and visualization

Node-Level Simulators

Time concept

- A sensor network simulator simulates the behavior of sensor network with respect to time
- In which, time may advance in differ ways: cycle-driven or discrete-event.

Node-Level Simulators

Cycle-driven simulation

- A cycle-driven (CD) discretize the continuous real time into ticks
- Simulator computes phenomenon at each tick. Like: physical environment, sensing data, communication data, etc.
- Communication by RF is assumed to be finished in a tick.

Node-Level Simulators

- CD simulators are easy to implement and use
- Most CD simulators issue are detecting and dealing cycle dependencies among nodes (ex: RF) or algorithms (ex: Thread).

Node-Level Simulators

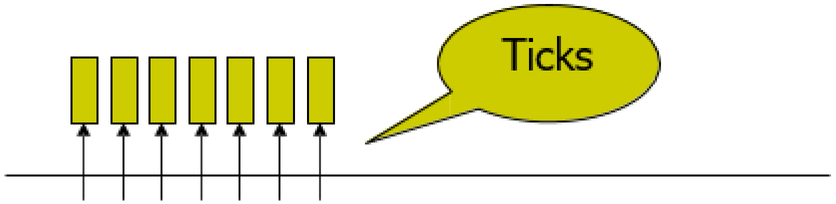
Discrete-event simulation

- Discrete-event (DE) simulator assumes the time is continuous.
- Usually use a Global event queue to store events.
- All events are stored chronologically in the Global event queue.

Node-Level Simulators

Example figure

- Sending a big file(1MB), 0.1MB/s max.
- CD



- DE

Node-Level Simulators

Comparison

- DE simulators are considered as better than CD simulators, because they are more actual. But they’re more complex to design and implement.
- Most popular sensor network simulators are DE simulators, like TOSSIM and NS2.

Node-Level Simulators

Ns2 + Sensor network

- Ns2 was meant to be wired network simulator, so extensions are being made for wireless (802.11,TDMA) and sensor networks.

Node-Level Simulators

Protocol supported:

- 802.3
- 802.11
- TDMA
- Ad hoc routing
- Sensor network routing

State-Centric Programming

- Applications that isn’t just simply generic distributed programs over an ad hoc network. We have to centralize data into nodes.

- EX: target tracking.

State-Centric Programming

Def:

- X: state of a system
- U: inputs
- Y: outputs
- K: update index
- F: state update function
- G: output observation function

State-Centric Programming

- $X_{k+1} = F(X_k, U_k)$
- $Y_k = G(X_k, U_k)$
- In state-centric programming, X and K come from many nodes. So many issue are discussed.

State-Centric Programming

- Where are the state vars stored?
- Where do the inputs com from?
- Where do the outputs go?
- Where are the functions f and g evaluated?
- How long does the acquisition of inputs take?

State-Centric Programming

Collaboration Group

- Which is a set of entities to update data.
- Protocol example:
    - Geographically constrained group
    - N-hop neighborhood group
    - Publish/Subscribe group
    - Acquaintance group
    - Mixing

State-Centric Programming

Geographically constrained group

- Since some phenomenon will be sensed in a area, GCG is useful.
- By broadcasting from one specific sensor, those have heard the packet will become the same group.

State-Centric Programming

N-hop neighbourhood group

- An anchor sets the hop limit and broadcasting it. Those who heard and is under the limit will become the same group.
- 0-hop: itself
- 1-hop: neighbors one hop away

State-Centric Programming

Publish/Subscribe group

- Dynamically defined by the requirement
- Only those have interested data will become the same group.

State-Centric Programming

Acquaintance group

- More dynamically, nodes will be invited to join a group. They can also quit.
- Group leader is selected beforehand, uses a ad hoc routing method to retrieve data from other nodes, then decide which one to invite.