



WELCOME

NETWORK ARCHITECTURE

Dr.P.Venkatesan
Associate Professor/ECE
Sri Chandrasekharendra Saraswathi Viswa
Mahavidyalaya University (SCSVMV)
Kanchipuram, Tamil Nadu, India.

Outline

1. Sensor network scenarios
2. Design principles for WSNs
3. Physical layer and transceiver design considerations in WSNs
4. Optimization goals and figures of merit
5. Gateway concepts

Sensor network scenarios

Types of sources and sinks:

- ❖ It has introduced several typical interaction patterns found in WSNs – event detection, periodic measurements, function approximation and edge detection, or tracking.
- ❖ The definition of “sources” and “sinks”. A source is any entity in the network that can provide information,
- ❖ A sink, on the other hand, is the entity where information is required. There are essentially three options for a sink:
- ❖ it could belong to the sensor network as such and be just another sensor/actuator node or it could be an entity outside this network.

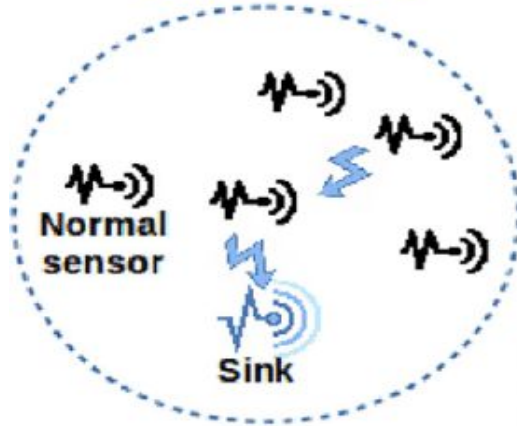
Sensor network scenarios

Types of sources and sinks:

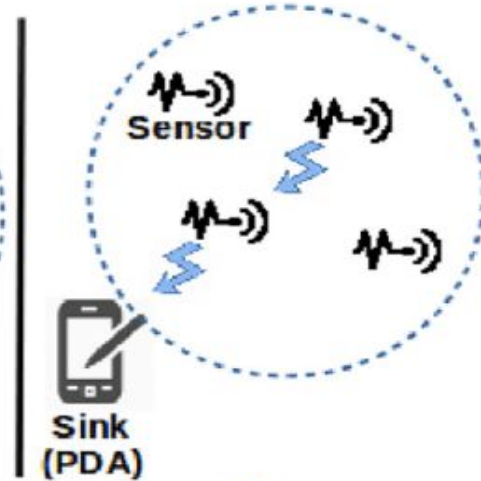
- ❖ second case, the sink could be an actual device, for example, a handheld or PDA used to interact with the sensor network;
- ❖ it could also be merely a gateway to another larger network such as the Internet.
- ❖ where the actual request for the information comes from some node “far away” and only indirectly connected to such a sensor network.
- ❖ whether sources or sinks move, but what they do with the information is not a primary concern of the networking architecture.

Sensor network scenarios

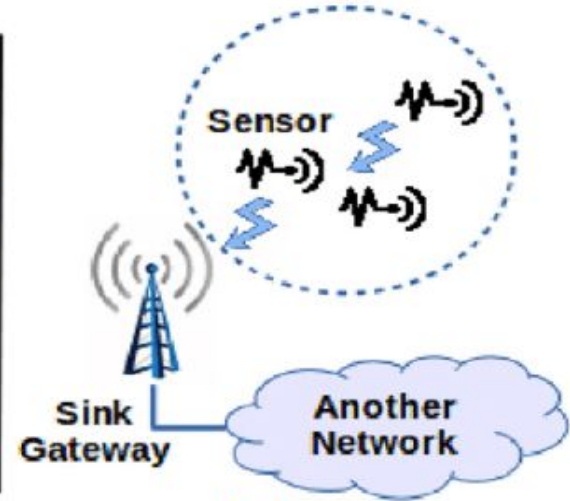
Same network



(a)



(b)



(c)

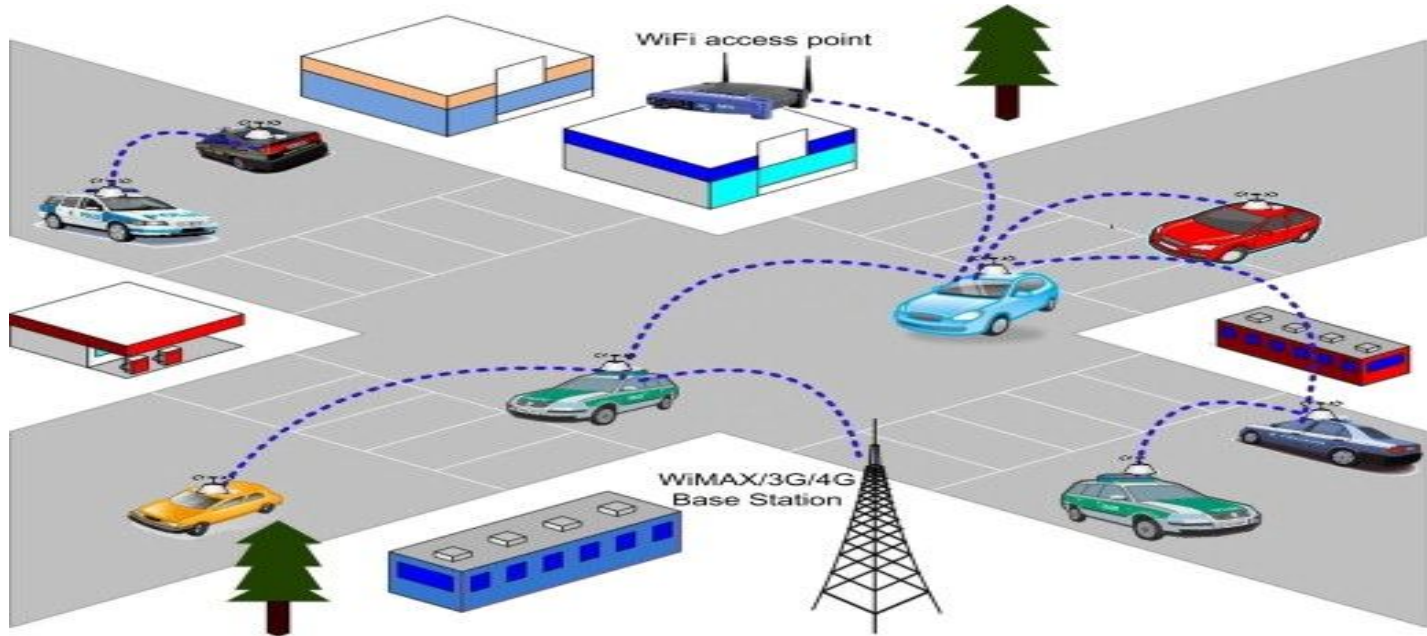
Different sink types: (a) a node belonging to the network, (b) an entity outside the network and (c) a gateway to another network.

Sensor network scenarios

- ❖ 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 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.

Sensor network scenarios

2. Multihop networks



Sensor network scenarios

- ❖ This concept of multihop 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 multihop routes from source to sink exist, nor that such a route is particularly short.

Sensor network scenarios

- ❖ While multihopping 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.

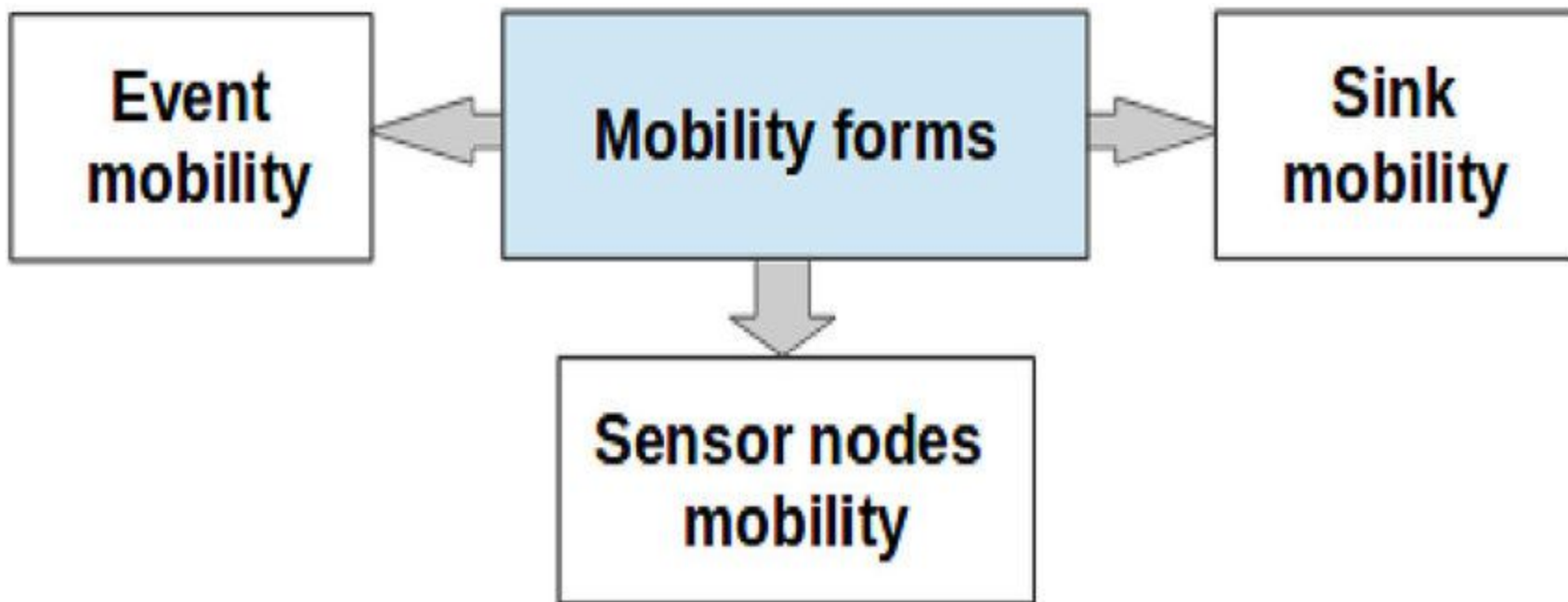
Sensor network scenarios

2. Multiple sinks and sources:

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

Sensor network scenarios

Three types of mobility:

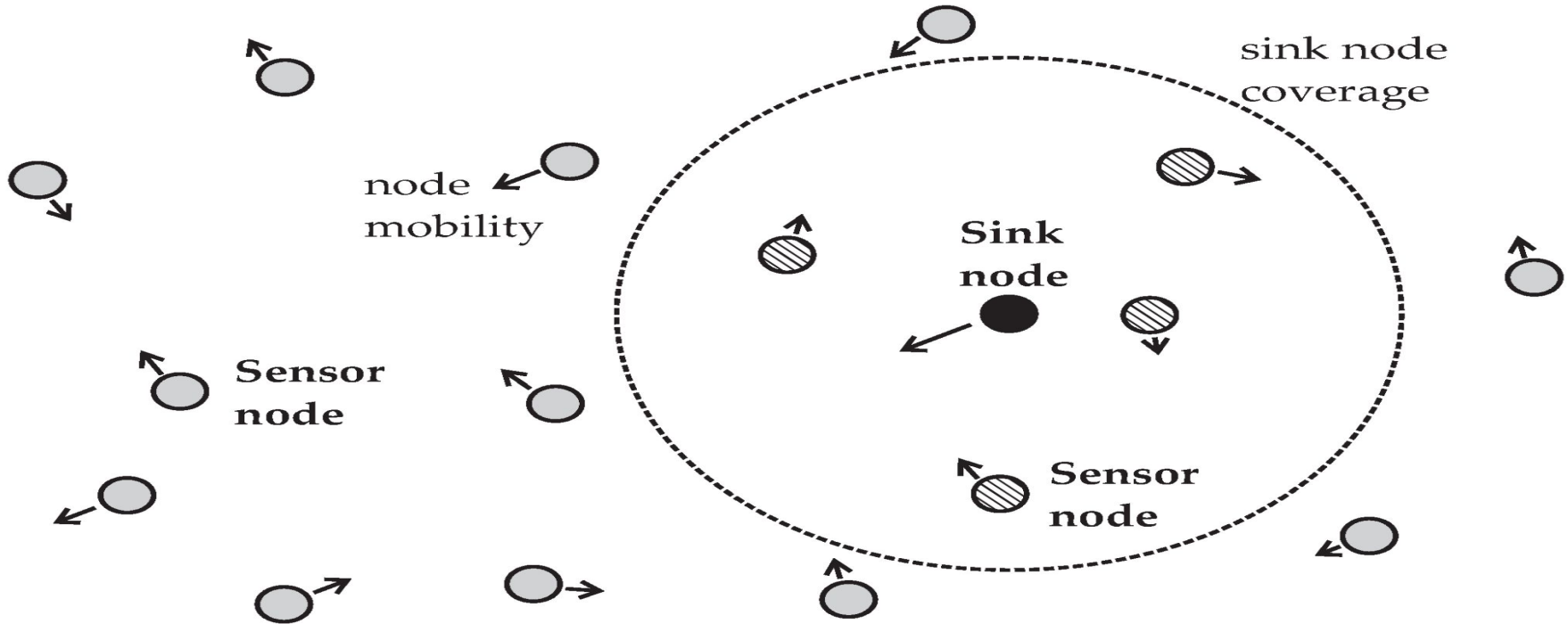


Sensor network scenarios

1. 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.

Node mobility



Sensor network scenarios

2. Sink mobility.

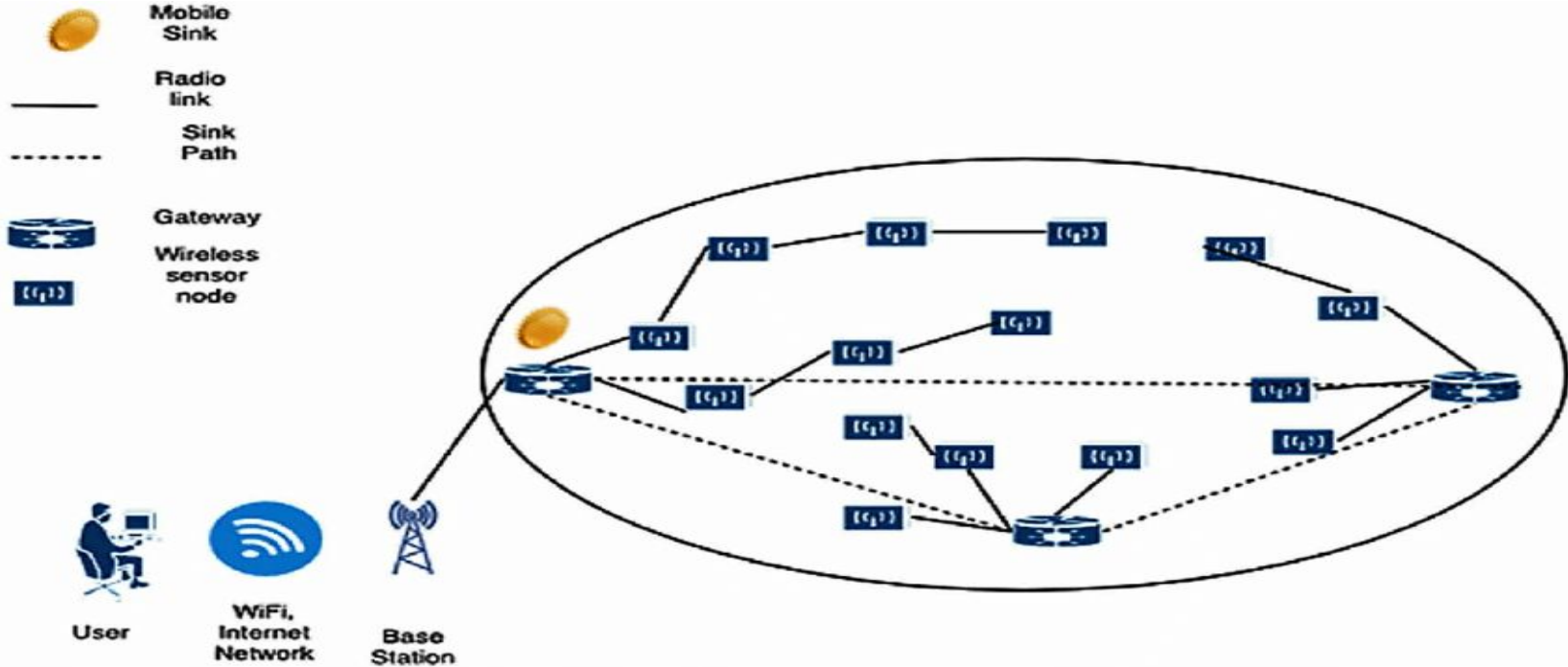
- ❖ The information sinks can be mobile While this can be a special case of node mobility, the important aspect is the mobility of an information sink that is not part of the sensor network, for example, a human user requested information via a PDA while walking in an intelligent building.
- ❖ In a simple 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.

Sensor network scenarios

Sink mobility.

- ❖ 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

Sink mobility.



Sensor network scenarios

3.Event mobility

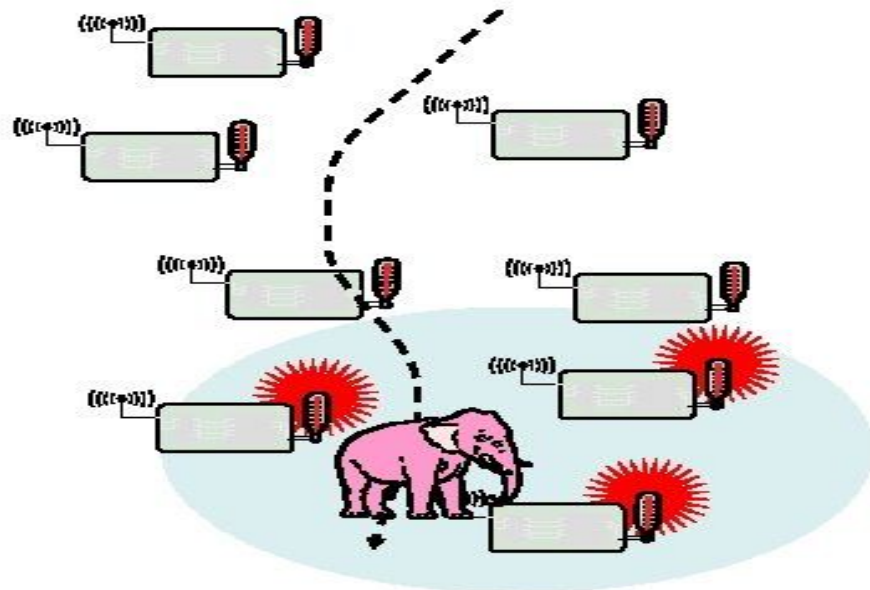
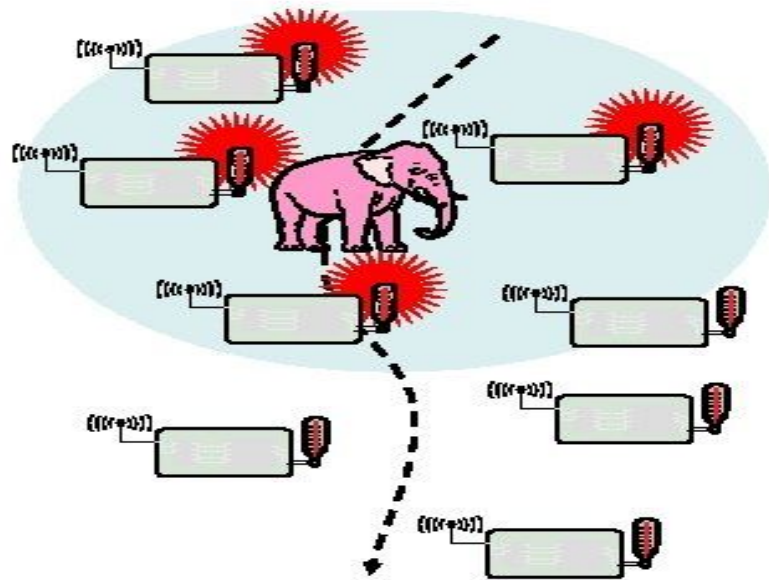
- ❖ In applications like event detection and in particular in tracking applications, the cause of the events or the objects to be tracked can be mobile.
- ❖ In such scenarios, it is (usually) important that the observed event is covered by a sufficient number of sensors at all time.
- ❖ Hence, sensors 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,

Sensor network scenarios

3.Event mobility

- ❖ 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 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

WSN Event Mobility



Different sources of mobility

- **Node mobility**
 - **Deliberately, self-propelled or by external force; targeted or at random**
 - **Happens in both WSN and MANET**
- **Sink mobility**
 - **Sinks may be located outside WSN**
 - **E.g., mobile requester**
- **Event mobility**
 - **In WSN, event that is to be observed moves around (or extends, shrinks)**

Section break



2.Design principles for WSNs

- 1 Distributed organization.
- 2.In-network processing.
- 3 Adaptive fidelity and accuracy.
- 4 Data centrality.
- 5 Exploit location information.
- 6 Exploit activity patterns.
- 7 Exploit heterogeneity.
- 8 Component-based protocol stacks and cross-layer optimization.

Design principles for WSNs

1 Distributed organization:

- ❖ The WSNs nodes should cooperatively organize the network, using distributed algorithms and protocols. Self-organization is a commonly used term for this principle.
- ❖ When organizing a network in a distributed fashion, it is necessary to be aware of potential shortcomings of this approach. In many circumstances, a centralized approach can produce solutions that perform better or require less resources (in particular, energy).

Design principles for WSNs

1 Distributed organization:

- ❖ To combine the advantages, one possibility is to use centralized principles in a localized fashion by dynamically electing, out of the set of equal nodes, specific nodes that assume the responsibilities of a centralized agent, for example, to organize medium access.
- ❖ Such elections result in a hierarchy, which has to be dynamic: The election process should be repeated continuously lest the resources of the elected nodes be overtaxed, the elected node runs out of energy, and the robustness disadvantages of such – even only localized – hierarchies manifest themselves.

Design principles for WSNs

2 In-network processing

- ❖ When organizing a network in a distributed fashion, the nodes in the network are not only passing on packets or executing application programs, they are also actively involved in taking decisions about how to operate the network.
- ❖ This is a specific form of information processing that happens in the network, but is limited to information about the network itself.
- ❖ It is possible to extend this concept by also taking the concrete data that is to be transported by the network into account in this information processing, making in-network processing a first-rank design principle.

Design principles for WSNs

2 In-network processing (Aggregation)

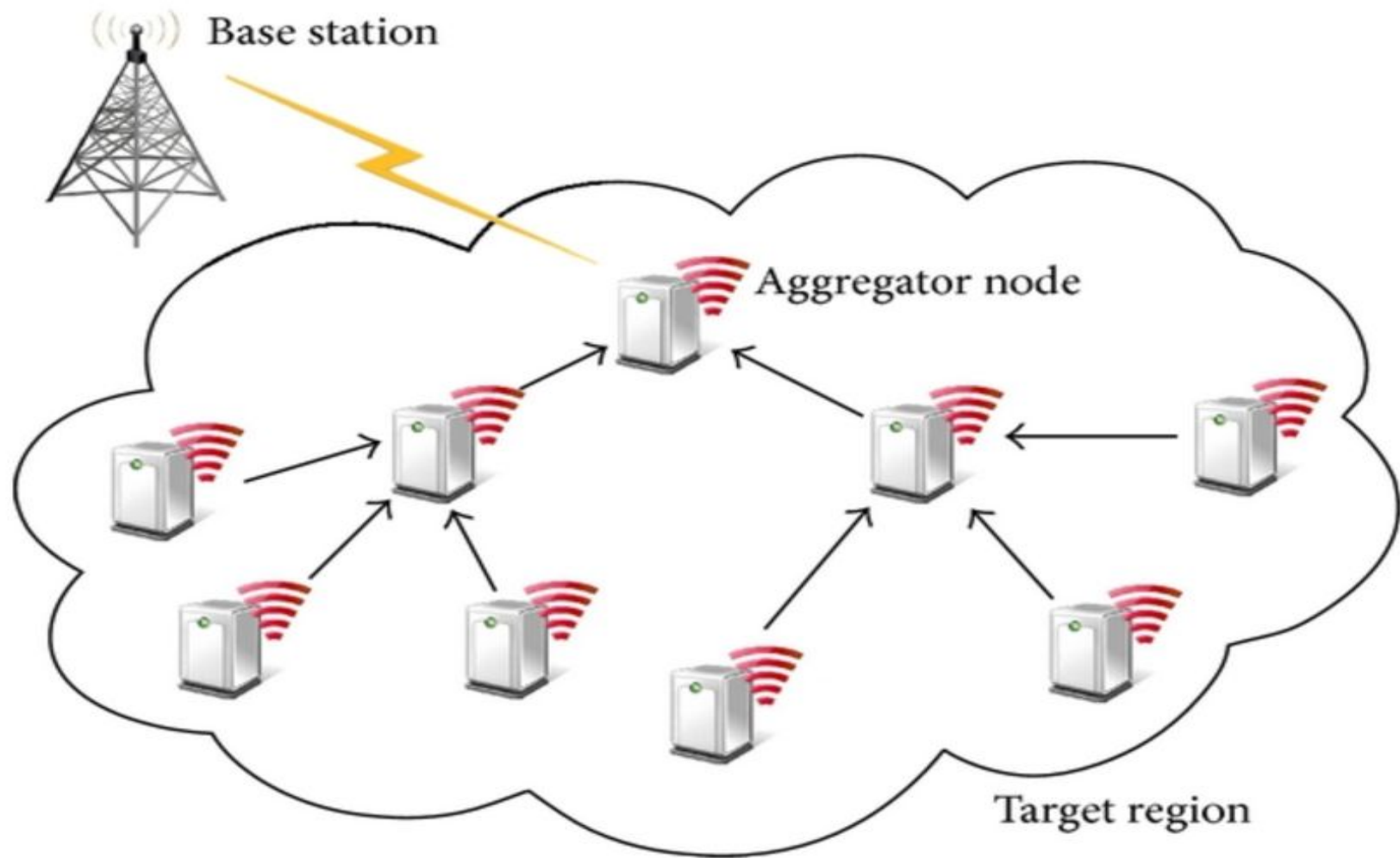
- ❖ The simplest in-network processing technique is aggregation.
- ❖ Suppose a sink is interested in obtaining periodic measurements from all sensors, but it is only relevant to check whether the average value has changed, or whether the difference between minimum and maximum value is too big.
- ❖ The name aggregation stems from the fact that in nodes intermediate between sources and sinks, information is aggregated into a condensed form out of information provided by nodes further away from the sink (and potentially, the aggregator's own readings)

Design principles for WSNs

2 In-network processing(Aggregation)

- ❖ The aggregation function to be applied in the intermediate nodes must satisfy some conditions for the result to be meaningful; most importantly, this function should be composable.
- ❖ A further classification of aggregate functions distinguishes duplicate-sensitive versus insensitive, summary versus exemplary, monotone versus nonmonotone, and algebraic versus holistic.
- ❖ Functions like average, counting, or minimum can profit a lot from aggregation; holistic functions like the median are not amenable to aggregation at all.

2



Design principles for WSNs

2 In-network processing (Distributed source coding and distributed compression)

- ❖ Aggregation condenses and sacrifices information about the measured values in order not to have to transmit all bits of data from all sources to the sink.
- ❖ Is it possible to reduce the number of transmitted bits (compared to simply transmitting all bits) but still obtain the full information about all sensor readings at the sink?
- ❖ It is related to the coding and compression problems known from conventional networks, where a lot of effort is invested to encode, for example, a video sequence, to reduce the required bandwidth.

Design principles for WSNs

2 In-network processing (Distributed source coding and distributed compression)

- ❖ The problem here is slightly different, in that we are interested to encode the information provided by several sensors, not just by a single camera; moreover, traditional coding schemes tend to put effort into the encoding, which might be too computationally complex for simple sensor nodes.
- ❖ The fact that information is provided by multiple sensors be exploited to help in coding? If the sensors were connected and could exchange their data, this would be conceivable (using relatively standard compression algorithms), but of course pointless.

Design principles for WSNs

2 In-network processing (Distributed source coding and distributed compression)

- ❖ It is quite likely that the readings of adjacent sensors are going to be quite similar; they are correlated. Such **correlation** can indeed be exploited such that not simply the sum of the data must be transmitted but that overhead can be saved here.
- ❖ Slepian-Wolf theorem–based work is an example of exploiting spatial correlation that is commonly present in sensor readings, as long as the network is sufficiently dense, compared to the derivative of the observed function and the degree of correlation between readings at two places. Similarly, **temporal correlation** can be exploited in sensor network protocols.

Design principles for WSNs

2 In-network processing (Distributed and collaborative signal processing)

- ❖ The in-networking processing approaches discussed so far have not really used the ability for processing in the sensor nodes, or have only used this for trivial operations like averaging or finding the maximum.
- ❖ An example for this concept is the distributed computation of a Fast Fourier Transform (FFT) .
- ❖ Depending on where the input data is located, there are different algorithms available to compute an FFT in a distributed fashion, with different trade-offs between local computation complexity and the need for communication. In principle, this is similar to algorithm design for parallel computers.

Design principles for WSNs

2 In-network processing (Distributed and collaborative signal processing)

- ❖ the latency of communication but also the energy consumption of communication and computation are relevant parameters to decide between various algorithms.
- ❖ Such distributed computations are mostly applicable to signal processing type algorithms; typical examples are beamforming and target tracking applications.

Design principles for WSNs

2 In-network processing (Mobile code/Agent-based networking)

- ❖ With the possibility of executing programs in the network, other programming paradigms or computational models are feasible.
- ❖ One such model is the idea of mobile code or agent-based networking. The idea is to have a small, compact representation of program code that is small enough to be sent from node to node.
- ❖ This code is then executed locally, for example, collecting measurements and then decides where to be sent next.

Design principles for WSNs

2 In-network processing(Mobile code/Agent-based networking)

- ❖ This idea has been used in various environments; a classic example is that of a software agent that is sent out to collect the best possible travel itinerary by hopping from one travel agent's computer to another and eventually returning to the user who has posted this inquiry.
- ❖ There is a vast amount of literature available on **mobile code/software agents** in general, see, for example,
- ❖ A newer take on this approach is to consider biologically inspired systems, in particular, the **swarm intelligence** of groups of simple entities, working together to reach a common goal

Design principles for WSNs

3 Adaptive fidelity and accuracy:

- ❖ In the context of a single node, the notion of making the fidelity of computation results contingent upon the amount of energy available for that particular computation.
- ❖ This notion can and should be extended from a single node to an entire network .
- ❖ As an example, consider a function approximation application. Clearly, when more sensors participate in the approximation, the function is sampled at more points and the approximation is better.
- ❖ Similar examples hold for event detection and tracking applications and in general for WSNs.

Design principles for WSNs

3 Adaptive fidelity and accuracy:

- ❖ The application should be able to adapt its requirements to the current status of the network – how many nodes have already failed, how much energy could be scavenged from the environment, what are the operational conditions (have critical events happened recently),
- ❖ The context of WSN-specific QoS metrics, the large variety of WSN applications makes it quite challenging to come up with a uniform interface for expressing such requirements,

Design principles for WSNs

4 Data centrality:

- ❖ In traditional communication networks, the focus of a communication relationship is usually the pair of communicating peers – the sender and the receiver of data.
- ❖ In a wireless sensor network, on the other hand, the interest of an application is not so much in the **identity** of a particular sensor node, it is much rather in the actual information reported about the physical environment.
- ❖ This is especially the case when a WSN is redundantly deployed such that any given event could be reported by **multiple nodes** – it is of no concern to the application precisely which of these nodes is providing data.

Design principles for WSNs

4 Data centrality:

- ❖ This fact that not the identity of nodes but the data are at the center of attention is called data-centric networking.
- ❖ Data-centric networking allows very different networking architectures compared to traditional, identity-centric networks. For one, it is the ultimate justification for some in-network processing techniques like data fusion and aggregation.
- ❖ Data-centric addressing also enables simple expressions of communication relationships

Design principles for WSNs

4 Data centrality: Implementation options for data-centric networking

- ❖ There are several possible ways to make this abstract notion of data-centric networks more concrete.

Overlay networks and distributed hash tables:

- ❖ In peer-to-peer networking, the solution for an efficient lookup of retrieval of data from an unknown source is usually to form an overlay network, implementing a Distributed Hash Table (DHT).

Design principles for WSNs

4 Data centrality:

- ❖ The crucial point is that this data source lookup can be performed efficiently, requiring $O(\log n)$ steps where n is the number of nodes, even with only distributed, localized information about where information is stored in the peer-to-peer network.
- ❖ Second, and more importantly, DHTs, coming from an IP-networking background, tend to ignore the distance/the hop count between two nodes and consider nodes as adjacent only on the basis of semantic information about their stored keys.

Design principles for WSNs

4 Data centrality: Publish/Subscribe

- ❖ The required separation in both time and identity of a sink node asking for information and the act of providing this information is not well matched with the synchronous characteristics of a request/reply protocol.
- ❖ Any node interested in a given kind of data can subscribe to it, and any node can publish data, along with information about its kind as well. Upon a publication, all subscribers to this kind of data are notified of the new data.

Design principles for WSNs

4 Data centrality: Publish/Subscribe

- ❖ Implementing this abstract concept of publishing and subscribing to information can be done in various ways.
- ❖ One possibility is to use a central entity where subscriptions and publications are matched to each other, but this is evidently inappropriate for WSNs. A distributed solution is preferable but considerably more complicated.

Design principles for WSNs

4 Data centrality: Databases.

- ❖ This view matches very well with the idea of using a data-centric organization of the networking protocols.
- ❖ Being interested in certain aspects of the physical environment that is surveyed by a WSN is equivalent to formulating queries for a database.
- ❖ To cast the sensor networks into the framework of relational databases, it is useful to regard the sensors as a virtual table to which relational operators can be applied.
- ❖ In SQL-based querying of a WSN can be extended to an easy-to-grasp interface to wireless sensor networks, being capable of expressing most salient interaction patterns with a WSN.

Design principles for WSNs

5 Exploit location information:

- ❖ Another useful technique is to exploit location information in the communication protocols whenever such information is present.
- ❖ Since the location of an event is a crucial information for many applications, there have to be mechanisms that determine the location of sensor nodes.
- ❖ it can simplify the design and operation of communication protocols and can improve their energy efficiency considerably.

Design principles for WSNs

6 Exploit activity patterns:

- ❖ Activity patterns in a wireless sensor network tend to be quite different from traditional networks.
- ❖ While it is true that the data rate averaged over a long time can be very small when there is only very rarely an event to report, this can change dramatically when something does happen.
- ❖ Once an event has happened, it can be detected by a larger number of sensors, breaking into a frenzy of activity, causing a well-known event shower effect.
- ❖ Hence, the protocol design should be able to handle such bursts of traffic by being able to switch between modes of quiescence and of high activity.

Design principles for WSNs

7. Exploit heterogeneity:

- ❖ The exploitation of activity patterns is the exploitation of heterogeneity in the network. Sensor nodes can be heterogeneous by construction, that is, some nodes have larger batteries, farther-reaching communication devices, or more processing power.
- ❖ They can also be heterogeneous by evolution, that is, all nodes started from an equal state, but because some nodes had to perform more tasks during the operation of the network, they have depleted their energy resources or other nodes had better opportunities to scavenge energy from the environment (e.g. nodes in shade are at a disadvantage when solar cells are used).

Design principles for WSNs

7. Exploit heterogeneity:

- ❖ Whether by construction or by evolution, heterogeneity in the network is both a burden and an opportunity.
- ❖ The opportunity is in an asymmetric assignment of tasks, giving nodes with more resources or more capabilities the more demanding tasks.
- ❖ For example, nodes with more memory or faster processors can be better suited for aggregation, nodes with more energy reserves for hierarchical coordination,

Design principles for WSNs

.8 Component-based protocol stacks and cross-layer optimization:

- ❖ All wireless sensor networks will require some – even if only simple – form of physical, MAC and link layer2 protocols;
- ❖ there will be wireless sensor networks that require routing and transport layer functionalities. Moreover, “helper modules” like time synchronization, topology control, or localization can be useful.
- ❖ On top of these “basic” components, more abstract functionalities can then be built. As a consequence, the set of components that is active on a sensor node can be complex, and will change from application to application

Design principles for WSNs

.8 Component-based protocol stacks and cross-layer optimization:

- ❖ Protocol components will also interact with each other in essentially two different ways.
- ❖ One is the simple exchange of data packets as they are passed from one component to another as it is processed by different protocols. The other interaction type is the exchange of cross-layer information.
- ❖ This possibility for cross-layer information exchange holds great promise for protocol optimization, but is also not without danger.



Thank you

3.Physical layer and transceiver design considerations In WSNs

The most crucial points influencing PHY design in wireless sensor networks are:

- ❖ Low power consumption.
- ❖ As one consequence: small transmit power and thus a small transmission range.
- ❖ As a further consequence: low duty cycle. Most hardware should be switched off or operated in a low-power standby mode most of the time.
- ❖ Comparably low data rates, on the order of tens to hundreds kilobits per second, required.
- ❖ Low implementation complexity and costs.
- ❖ Low degree of mobility.
- ❖ A small form factor for the overall node.

3.Physical layer and transceiver design considerations In WSNs

The most crucial points influencing PHY design in wireless sensor networks are:

- ❖ In general, 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.

3.Physical layer and transceiver design considerations In WSNs

- 1 Energy usage profile
- 2 Choice of modulation scheme
- 3 Dynamic modulation scaling
- 4 Antenna considerations

3. Physical layer and transceiver design considerations in WSNs

1 Energy usage profile.

- ❖ The choice of a small transmit power leads to an energy consumption profile different from other wireless devices like cell phones.
- ❖ The radiated energy is small, typically on the order of 0 dBm (corresponding to 1 mW). On the other hand, the overall transceiver (RF front end and baseband part) consumes much more energy than is actually radiated.
- ❖ Estimate that a transceiver working at frequencies beyond 1 GHz takes 10 to 100 mW of power to radiate 1 mW.

3. Physical layer and transceiver design considerations in WSNs

1 Energy usage profile.

- ❖ similar numbers are given for 2.4-GHz CMOS transceivers: : For a radiated power of 0 dBm, the transmitter uses actually 32 mW, whereas the receiver uses even more, 38 mW. For the Mica motes, 21 mW are consumed in transmit mode and 15 mW in receive mode.
- ❖ These numbers coincide well with the observation that many practical transmitter designs have efficiencies below 10 % at low radiated power.

3. Physical layer and transceiver design considerations in WSNs

1 Energy usage profile.

- ❖ A second key observation is that for small transmit powers the transmit and receive modes consume more or less the same power; it is even possible that reception requires more power than transmission depending on the transceiver architecture, the idle mode's power consumption can be less or in the same range as the receive power.
- ❖ To reduce average power consumption in a low-traffic wireless sensor network, keeping the transceiver in idle mode all the time would consume significant amounts of energy.

3. Physical layer and transceiver design considerations In WSNs

1 Energy usage profile.

- ❖ Therefore, it is important to put the transceiver into sleep state instead of just idling. It is also important to explicitly include the received power into energy dissipation models, since the traditional assumption that receive energy is negligible is no longer true.
- ❖ There is the problem of the startup energy/startup time, which a transceiver has to spend upon waking up from sleep mode,

3. Physical layer and transceiver design considerations In WSNs

1 Energy usage profile.

- ❖ A third key observation is the relative costs of communications versus computation in a sensor node. Clearly, a comparison of these costs depends for the communication part on the BER requirements, range, transceiver type, and so forth, and for the computation part on the processor type, the instruction mix, and so on.

3. Physical layer and transceiver design considerations In WSNs

2 Choice of modulation scheme:

- ❖ A crucial point is the choice of modulation scheme. Several factors have to be balanced here: the required and desirable data rate and symbol rate, the implementation complexity, the relationship between radiated power and target BER, and the expected channel characteristics.
- ❖ To maximize the time a transceiver can spend in sleep mode, the transmit times should be minimized. The higher the data rate offered by a transceiver/modulation, the smaller the time needed to transmit a given amount of data and, consequently, the smaller the energy consumption.

3. Physical layer and transceiver design considerations in WSNs

2 Choice of modulation scheme

- ❖ A second important observation is that the power consumption of a modulation scheme depends much more on the symbol rate than on the data rate.
- ❖ For example, power consumption measurements of an IEEE 802.11b Wireless Local Area Network (WLAN) card showed that the power consumption depends on the modulation scheme, with the faster Complementary Code Keying (CCK) modes consuming more energy than DBPSK and DQPSK

3. Physical layer and transceiver design considerations In WSNs

2 Choice of modulation scheme

- ❖ m-ary modulation requires more complex digital and analog circuitry than 2-ary modulation, for example, to parallelize user bits into m-ary symbols.
- ❖ Many m-ary modulation schemes require for increasing m an increased E_b/N_0 ratio and consequently an increased radiated power to achieve the same target BER; others become less and less bandwidth efficient.
- ❖ However, in wireless sensor network applications with only low to moderate bandwidth requirements, a loss in bandwidth efficiency can be more tolerable than an increased radiated power to compensate E_b/N_0 losses.

3. Physical layer and transceiver design considerations In WSNs

2 Choice of modulation scheme:

- ❖ It is expected that in many wireless sensor network applications most packets will be short, on the order of tens to hundreds of bits. For such packets, the startup time easily dominates overall energy consumption, rendering any efforts in reducing the transmission time by choosing m-ary modulation schemes irrelevant.
- ❖ The optimal decision would have to properly balance the modulation scheme and other measures to increase transmission robustness, since these also have energy costs:

3. Physical layer and transceiver design considerations In WSNs

2 Choice of modulation scheme

- ❖ With retransmissions, entire packets have to be transmitted again.
- ❖ With FEC coding, more bits have to be sent and there is additional energy consumption for coding and decoding. While coding energy can be neglected, and the receiver needs significant energy for the decoding process.
- ❖ This is especially cumbersome if the receiver is a power-constrained node.

3. Physical layer and transceiver design considerations In WSNs

2 Choice of modulation scheme:

- ❖ The cost of increasing the radiated power depends on the efficiency of the power amplifier, but the radiated power is often small compared to the overall power dissipated by the transceiver, and additionally this drives the PA into a more efficient regime.

3. Physical layer and transceiver design considerations in WSNs

2 Choice of modulation scheme

- ❖ Specifically, the energy-per-bit consumption (defined as the overall energy consumption for transmitting a packet of n bits divided by n) of different m -ary QAM modulation schemes has been investigated for different packet sizes, taking startup energy and the energy costs of power amplifiers as well as PHY and MAC packet overheads explicitly into account.



3. Physical layer and transceiver design considerations In WSNs

3 Dynamic modulation scaling

- ❖ Even if it is possible to determine the optimal scheme for a given combination of BER target, range, packet sizes and so forth, such an optimum is only valid for short time;
- ❖ as soon as one of the constraints changes, the optimum can change, too. In addition, other constraints like delay or the desire to achieve high throughput can dictate to choose higher modulation schemes.

3. Physical layer and transceiver design considerations In WSNs

3 Dynamic modulation scaling:

- ❖ Therefore, it is interesting to consider methods to adapt the modulation scheme to the current situation. Such an approach, called **dynamic modulation scaling**,
- ❖ In particular, for the case of **m-ary** QAM and a target BER of 10^{-5} , a model has been developed that uses the symbol rate B and the number of levels per symbol m as parameters.

3. Physical layer and transceiver design considerations In WSNs

3 Dynamic modulation scaling:

- ❖ This model expresses the energy required per bit and also the achieved delay per bit (the inverse of the data rate), taking into account that higher modulation levels need higher radiated energy.
- ❖ The energy per bit depends much more on m than on B . In fact, for the particular parameters chosen, it is shown that both energy per bit and delay per bit are minimized for the maximum symbol rate.

3. Physical layer and transceiver design considerations In WSNs

3 Dynamic modulation scaling:

- ❖ The modulation scaling, a packet is equipped with a delay constraint, from which directly a minimal required data rate can be derived.
- ❖ Since the symbol rate is kept fixed, the approach is to choose the smallest m that satisfies the required data rate and which thus minimizes the required energy per bit. Such delay constraints can be assigned either explicitly or implicitly.

3. Physical layer and transceiver design considerations In WSNs

3 Dynamic modulation scaling:

- ❖ When there are no packets present, a small value for m can be used, having low energy consumption. As backlog increases, m is increased as well to reduce the backlog quickly and switch back to lower values of m . This modulation scaling approach has some similarities to the concept of **dynamic voltage scaling**.

3. Physical layer and transceiver design considerations In WSNs

4 Antenna considerations:

- ❖ In small form factor of the overall sensor nodes restricts the size and the number of antennas. As explained above, if the antenna is much smaller than the carrier's wavelength,
- ❖ it is hard to achieve good antenna efficiency, that is, with ill-sized antennas one must spend more transmit energy to obtain the same radiated energy.

3. Physical layer and transceiver design considerations In WSNs

4 Antenna considerations:

- ❖ Secondly, with small sensor node cases, it will be hard to place two antennas with suitable distance to achieve receive diversity.
- ❖ The antennas should be spaced apart at least 40–50 % of the wavelength used to achieve good effects from diversity. For 2.4 GHz, this corresponds to a spacing of between 5 and 6 cm between the antennas, which is hard to achieve with smaller cases.

3. Physical layer and transceiver design considerations In WSNs

4 Antenna considerations:

- ❖ The radio waves emitted from an antenna close to the ground – typical in some applications – are faced with higher path-loss coefficients than the common value $\alpha = 2$ for free-space communication.
- ❖ Typical attenuation values in such environments, which are also normally characterized by obstacles.

3. Physical layer and transceiver design considerations In WSNs

4 Antenna considerations:

- ❖ Depending on the application, antennas must not protrude from the casing of a node, to avoid possible damage to it. These restrictions, in general, limit the achievable quality and characteristics of an antenna for wireless sensor nodes.
- ❖ Nodes randomly scattered on the ground, for example, deployed from an aircraft, will land in random orientations, with the antennas facing the ground or being otherwise obstructed.

3. Physical layer and transceiver design considerations In WSNs

4 Antenna considerations:

- ❖ This can lead to non isotropic propagation of the radio wave, with considerable differences in the strength of the emitted signal in different directions.
- ❖ This effect can also be caused by the design of an antenna, which often results in considerable differences in the spatial propagation characteristics (so-called lobes of an antenna)

3. Physical layer and transceiver design considerations In WSNs

5 Further reading: Jointly optimizing coding and modulation

- ❖ consider coding and modulation from an information-theoretic perspective for different channel models, including the AWGN, flat fading channels and block fading channels.
- ❖ One particularly interesting result is that the capacity of a Rayleigh fading channel with power control can be higher than the capacity of an AWGN channel with the same average radiated power

3.Physical layer and transceiver design considerations In WSNs

5 Further reading:

- ❖ DSSS in WSN Some efforts toward the construction of DSSS transceivers for wireless sensor networks with their space and power constraints, and low-power spread-spectrum transceivers for IEEE 802.11.
- ❖ Energy efficiency in GSM: Reducing energy consumption is an issue not only in wireless sensor networks but also in other types of systems, for example, cellular systems. For the interested: advanced signal processing algorithms for reducing power consumption of GSM transceivers

3. Physical layer and transceiver design considerations In WSNs

5 Further reading:

- ❖ Specifically, the influence of symbol-by symbol power control at the transmitter in the presence of channel-state information such that deep fades are answered with higher output powers (“channel inversion”), of receiver diversity and interleaving and of coding schemes with unequal protection (i.e., user bits of different importance are encoded differently) on the channel capacity.

T A K E

A

B R E A K



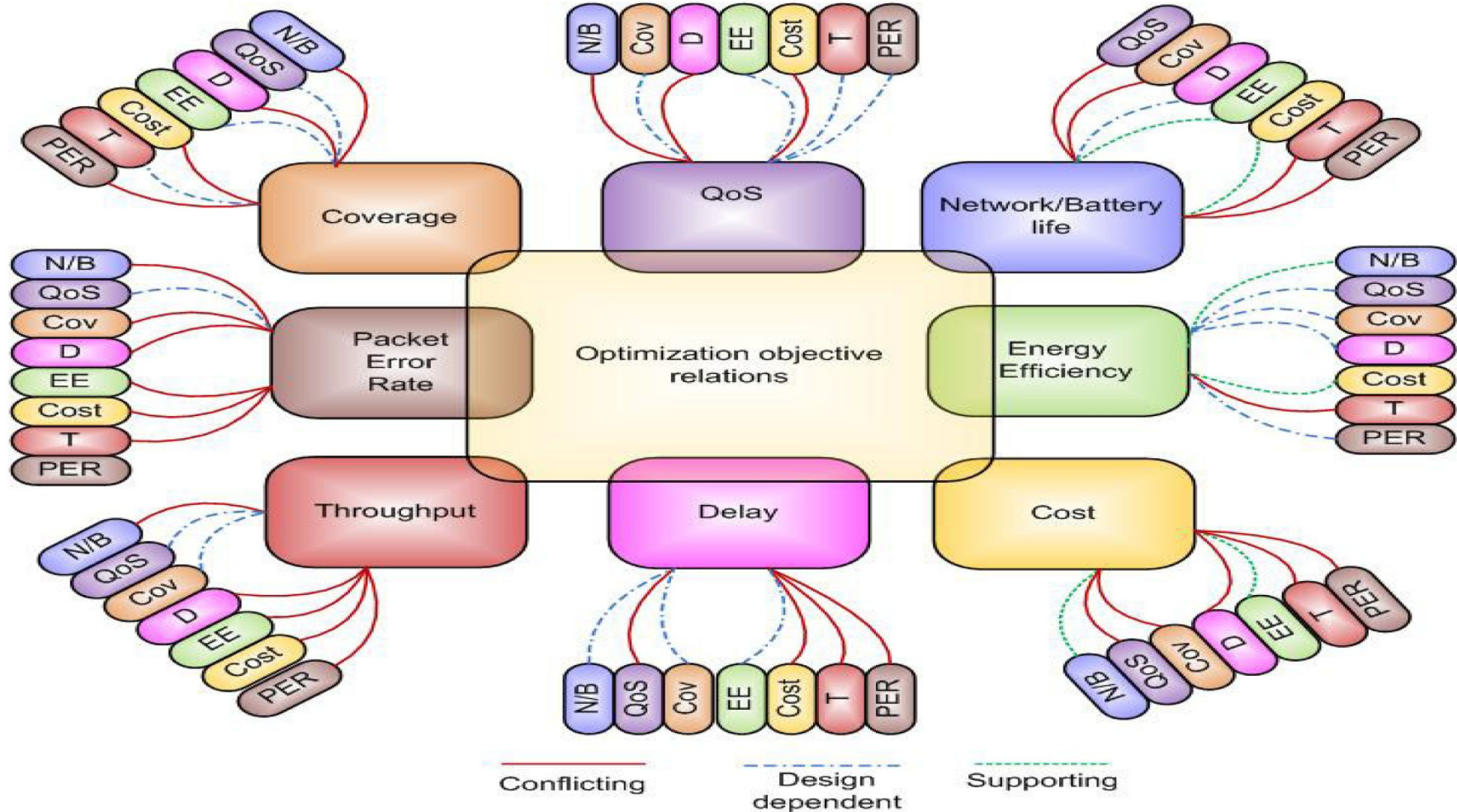
Download from
Dreamstime.com

This watermarked comp image is for previewing purposes only.



4. Optimization goals and figures of merit

- 1 Quality of service.
- 2 Energy efficiency.
- 3 Scalability.
- 4 Robustness



ervice in Wireless Sensor Networks: Issues and Challenges



4. Optimization goals and figures of merit

1 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.
- ❖ Possibly, additional requirements about the offered Quality of Service (QoS) are made, especially in the context of multimedia applications. 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.

4. Optimization goals and figures of merit

1 Quality of service.

- ❖ Hence, high-level QoS attributes corresponding to the subjective QoS attributes in conventional networks are required.
- ❖ But just like in traditional networks, high-level QoS attributes in WSN highly depend on the application. Some generic possibilities are:
 - 1 Event detection/reporting probability
 - 2 Event classification error
 - 3 Event detection delay
 - 4 Missing reports
 - 5 Approximation accuracy
 - 6 Tracking accuracy

4. Optimization goals and figures of merit

1 Quality of service.

Event detection/reporting probability:

- ❖ What 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).

4. Optimization goals and figures of merit

1 Quality of service.

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: What 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.

4. Optimization goals and figures of merit

1 Quality of service.

- ❖ **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:** 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..

4. Optimization goals and figures of merit

2 Energy efficiency

- ❖ It is clear that with an arbitrary amount of energy, most of the QoS metrics defined above 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.

4. Optimization goals and figures of merit

2 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. The most commonly considered aspects are:

4. Optimization goals and figures of merit

2 Energy efficiency.

Energy per correctly received bit: 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 : Similarly, 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).

4. Optimization goals and figures of merit

2 Energy efficiency.

Delay/energy trade-offs: Some applications have a notion of “urgent” events, which can justify an increased energy investment for a speedy reporting of such events. Here, the trade-off between delay and energy overhead is interesting.

Network lifetime 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. Possible definitions are:

4. Optimization goals and figures of merit

2 Energy efficiency.

Time to first node death: When does the first node in the network run out of energy or fail and stop operating?

Network half-life: When have 50 % of the nodes run out of energy and stopped operating? Any other fixed percentile is applicable as well.

Time to partition : When does the first partition of the network in two (or more) disconnected parts occur? This can be as early as the death of the first node (if that was in a pivotal position) or occur very late if the network topology is robust.

4. Optimization goals and figures of merit

2 Energy efficiency.

Time to loss of coverage: 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.

4. Optimization goals and figures of merit

2 Energy efficiency.

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.

4. Optimization goals and figures of merit

2 Energy efficiency.

- ❖ Obviously, the longer these times are, the better does a network perform. More generally, it is also possible to look at the (complementary) distribution of node lifetimes (with what probability does a node survive a given amount of time?) or at the relative survival times of a network (at what time are how many percent of the nodes still operational).
- ❖ This latter function allows an intuition about many WSN-specific protocols in that they tend to sacrifice long lifetimes in return for an improvement in short lifetimes – they “sharpen the drop”

4. Optimization goals and figures of merit

2 Energy efficiency.

- ❖ All these metrics can of course only be evaluated under a clear set of assumptions about the energy consumption characteristics of a given node, about the actual “load” that the network has to deal with (e.g. when and where do events happen), and also about the behavior of the radio channel.

4. Optimization goals and figures of merit

3 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.

4. Optimization goals and figures of merit

3 Scalability.

- ❖ 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.

4. Optimization goals and figures of merit

4 Robustness:

- ❖ Related to QoS and somewhat also to scalability requirements, wireless sensor networks should also exhibit an appropriate robustness. They should not fail just because a limited number of nodes run out of energy, or because their environment changes 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.

5. Gateway concepts:

- 1 The need for gateways.
- 2 WSN to Internet communication.
- 3 Internet to WSN communication.
- 4 WSN tunneling.

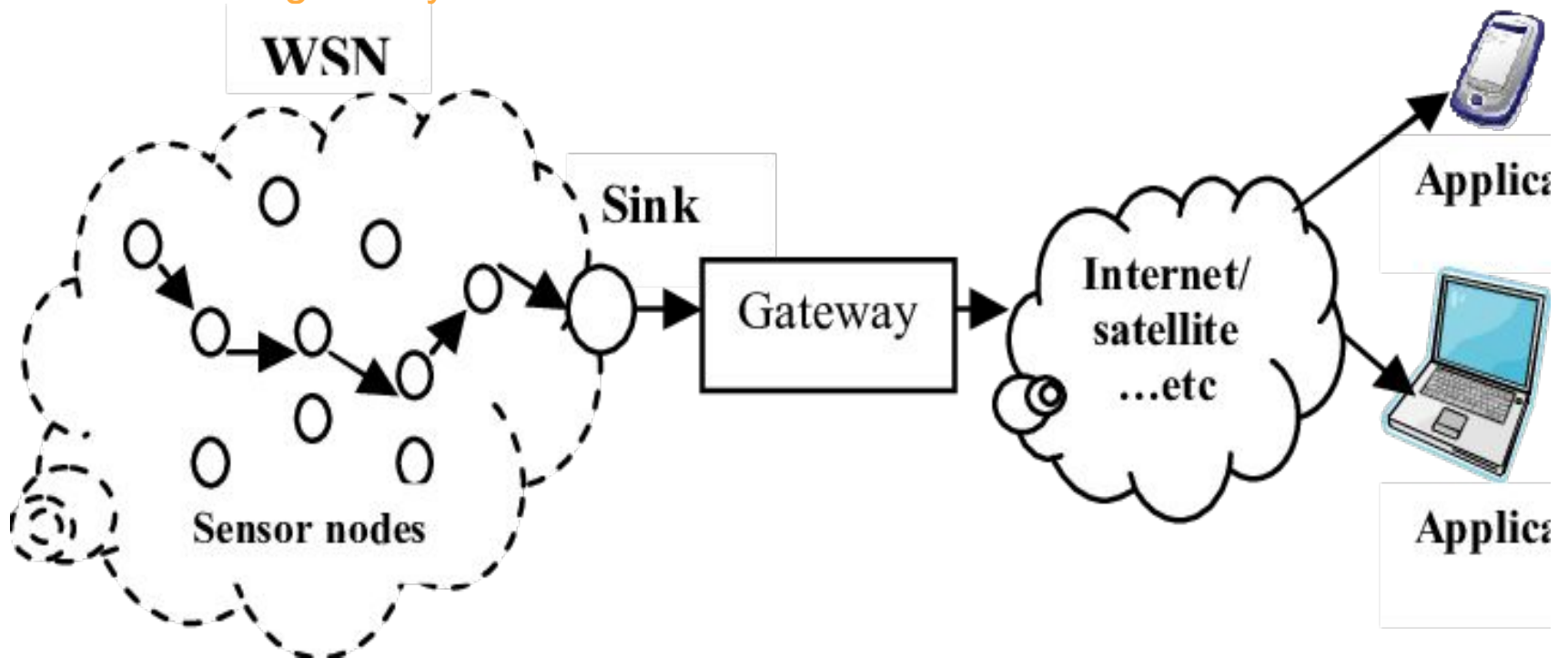
5. Gateway concepts:

1 The 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).

5. Gateway concepts:

1 The need for gateways:



5. Gateway concepts:

1 The need for gateways:

- ❖ 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.

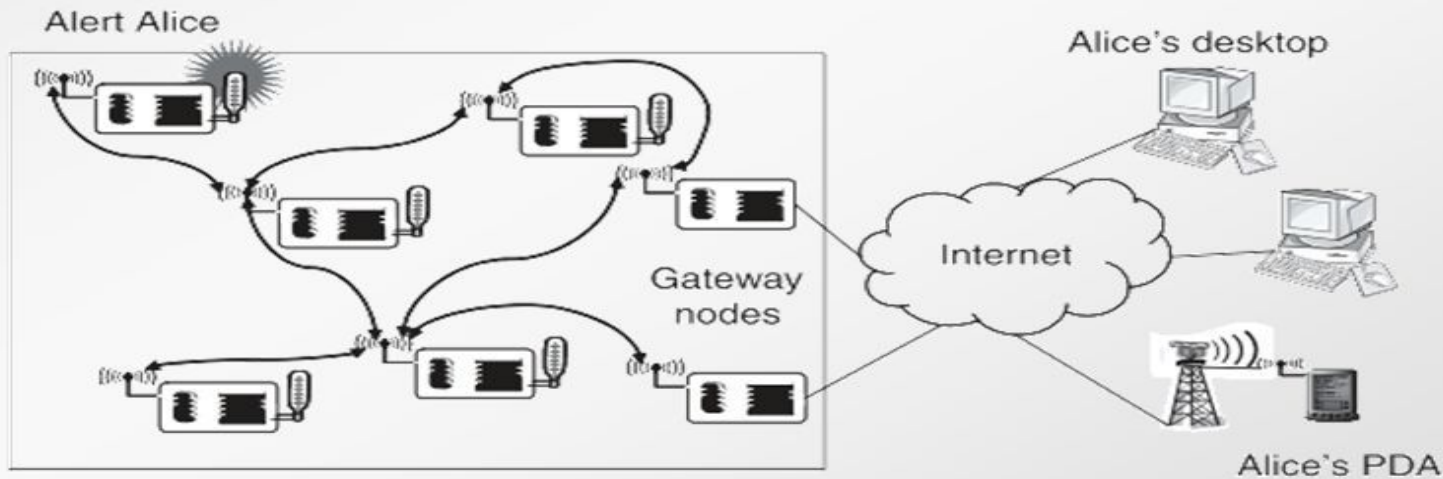
5. Gateway concepts:

1 The need for gateways:

- ❖ 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.
- ❖ The remaining possibility is therefore to design the gateway as an actual application-level gateway: on the basis of the application-level information, the gateway will have to decide its action.

WSN to Internet Communication

- E.g., deliver an alarm message to an Internet host
- Issues
 - Need to find a gateway (integrates routing & service discovery)
 - Choose “best” gateway if several are available
 - How to find Alice or Alice’s IP?



5. Gateway concepts:

2 WSN to Internet communication:

- ❖ Assume that the initiator of a WSN–Internet communication resides in the WSN - for example, 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.

5. Gateway concepts:

2 WSN to Internet communication:

- ❖ If several such gateways are available, how to choose between them? In particular, if not all Internet hosts are reachable via each gateway or 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.

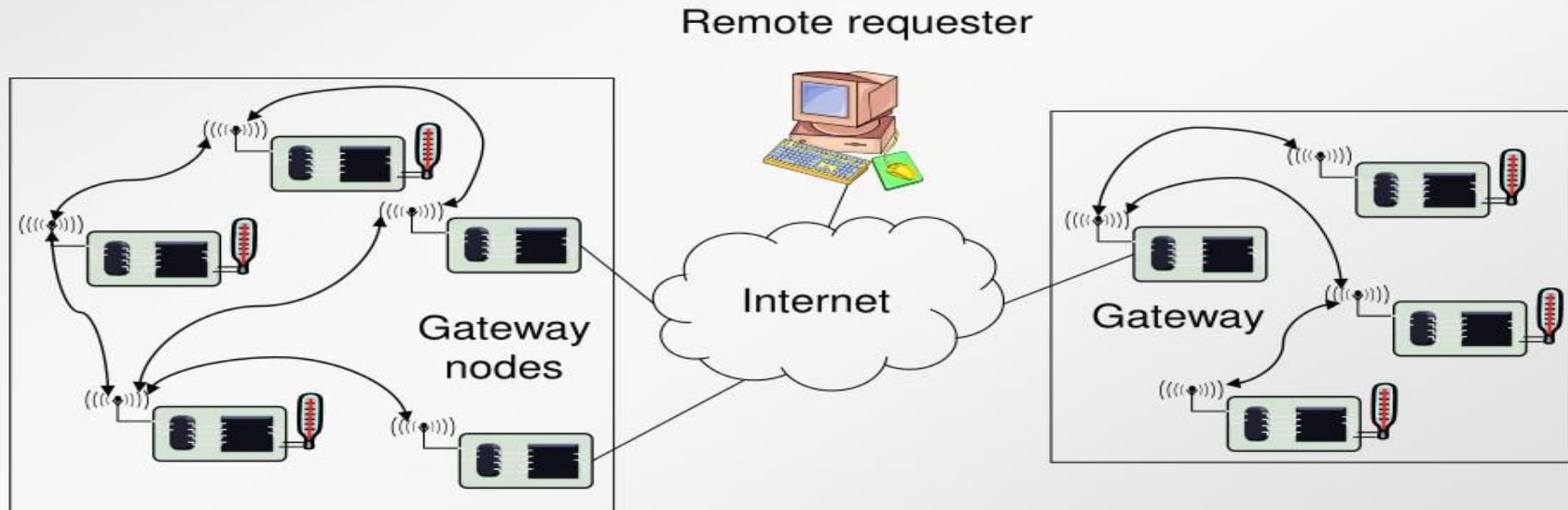
5. Gateway concepts:

2 WSN to Internet communication:

- ❖ How does a sensor node know to which Internet host to address such a message? Or even worse, how to map a semantic notion (“Alert Alice”) to a concrete IP address?
- ❖ 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, for example) in its own packets; the gateway then has to extract this information and translate it into IP packets. An ensuing question is 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.

Internet to WSN communication

- How to find the right WSN to answer a need?
- How to translate from IP protocols to WSN protocols, semantics?



5. Gateway concepts:

3 Internet to WSN communication:

- ❖ The case of an 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,
- ❖ for 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

5. Gateway concepts:

3 Internet to WSN communication:

- ❖ The more general case is, 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.
- ❖ First of all, again the question of service discovery presents itself – 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?

5. Gateway concepts:

3 Internet to WSN communication:

- ❖ Once the requesting terminal has obtained this information, how to access the actual services? Clearly, 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 and is impossible if a sensor node does not even have an IP address.

5. Gateway concepts:

3 Internet to WSN communication:

- ❖ The requesting terminal can instead send a properly formatted request to this gateway, which acts as an application-level gateway or a proxy for the individual/set of sensor nodes that can answer this request; the gateway translates this request into the proper intra sensor network protocol interactions.
- ❖ This assumes that there is an application-level protocol that a remote requester and gateway can use and that is more suitable for communication over the Internet than the actual sensor network protocols and that is more convenient for the remote terminal to use.

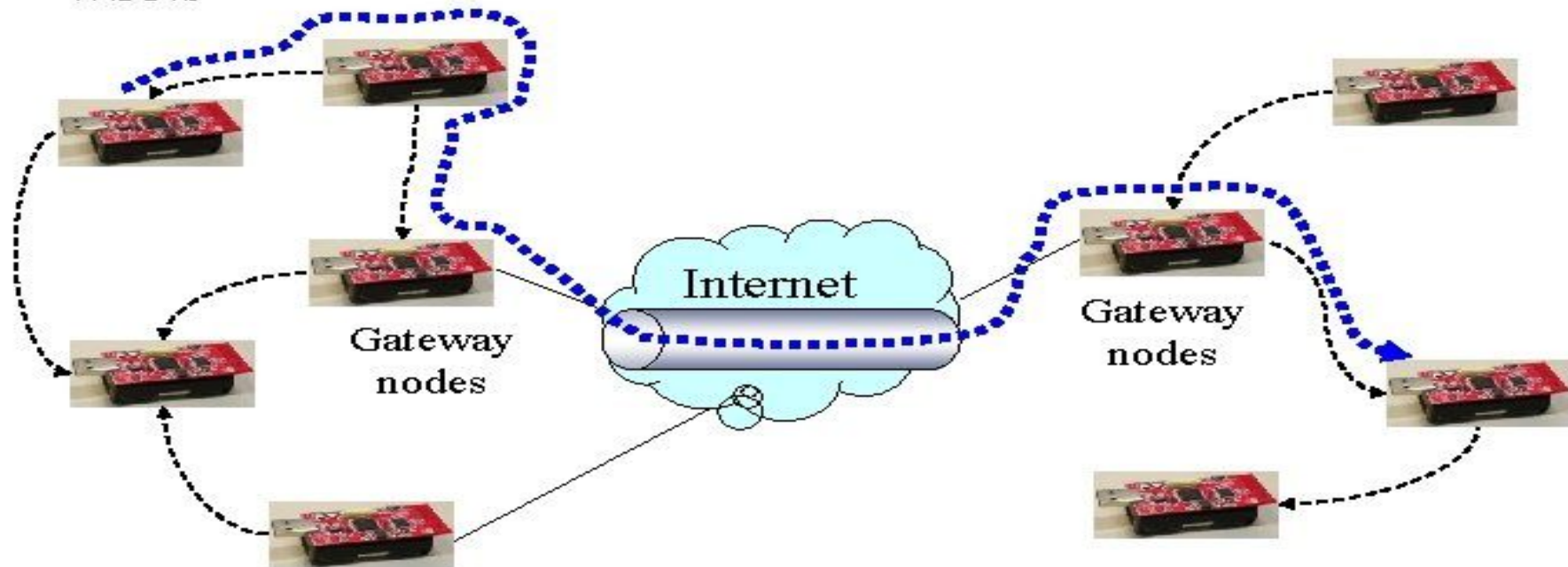
5. Gateway concepts:

3 Internet to WSN communication:

- ❖ There are some clear parallels for such an application-level protocol with so-called Web Service Protocols, which can explicitly describe services and the way they can be accessed.
- ❖ The Web Service Description Language (WSDL), in particular, 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.

WSN Tunneling

- “ The idea is to build a larger, “Virtual” WSN
- “ Use the Internet to “tunnel” WSN packets between two remote WSNs



5. Gateway concepts:

4 WSN tunneling:

- ❖ In addition to these scenarios describing actual interactions between a WSN and Internet terminals, the 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.

5. Gateway concepts:

4 WSN tunneling:

- ❖ 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).
- ❖ Such tunnels need not necessarily be in the form of fixed network connections. even mobile nodes carried by people can be considered as means for intermediate interconnection of WSNs



Thank you
to all



WELCOME

we're glad you're here

NETWORKING SENSORS

Dr.P.Venkatesan

Associate Professor/ECE

**Sri Chandrasekharendra Saraswathi Viswa
Mahavidyalaya University (SCSVMV)
Kanchipuram, Tamil Nadu, India.**

Outline

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

MAC PROTOCOLS

Introduction to MAC Protocols

- ❖ Medium Access Control (MAC) protocols solve a seemingly simple task:
- ❖ They coordinate the times where a number of nodes access a shared communication medium.
- ❖ They differ, among others, in the types of media they use and in the performance requirements for which they are optimized.
- ❖ The fundamentals of MAC protocols and explains the specific requirements and problems these protocols have to face in wireless sensor networks.

MAC PROTOCOLS

Introduction to MAC Protocols

- ❖ The single most important requirement is energy efficiency and there are different MAC-specific sources of energy waste to consider: overhearing, collisions, overhead, and idle listening.
- ❖ One important approach is to switch the wireless transceiver into a sleep mode.
- ❖ Therefore, there are trade-offs between a sensor network's energy expenditure and traditional performance measures like delay and throughput.

MAC PROTOCOLS

Introduction to MAC Protocols

- ❖ Medium Access Control (MAC) protocols is the first protocol layer above the Physical Layer (PHY) and consequently 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.
- ❖ Some of the traditional performance criteria are delay, throughput, and fairness, whereas in WSNs, the issue of energy conservation becomes important.

MAC PROTOCOLS

Introduction to MAC Protocols

- ❖ The MAC protocol determines for a node the points in time when it accesses the medium to try to transmit a data, control, or management packet to another node (unicast) or to a set of nodes (multicast, broadcast).
- ❖ The MAC is considered as a part of the Data Link Layer (DLL), but there is a clear division of work between the MAC and the remaining parts of the DLL.
- ❖ Two important responsibilities of the remaining parts of the DLL are error control and flow control.

MAC PROTOCOLS

Introduction to MAC Protocols

- ❖ Error control is used to ensure correctness of transmission and to take appropriate actions in case of transmission errors and flow control regulates the rate of transmission to protect a slow receiver from being overwhelmed with data.
- ❖ The issue of energy efficiency is the prime consideration in WSN MAC protocols, and therefore, we concentrate on schemes that explicitly try to reduce overall energy consumption.
- ❖ One of the main approaches to conserve energy is to put nodes into sleep state whenever possible.

Fundamentals of (wireless) MAC Protocols

- ❖ Requirements and design constraints for wireless MAC protocols
- ❖ Important classes of MAC protocols
- ❖ MAC protocols for wireless sensor networks

Fundamentals of (wireless) MAC Protocols

Requirements and design constraints for wireless MAC protocols:

- ❖ The most important performance requirements for MAC protocols are throughput efficiency, stability, fairness, low access delay (time between packet arrival and first attempt to transmit it), and low transmission delay (time between packet arrival and successful delivery), as well as a low overhead.
- ❖ The overhead in MAC protocols can result from per-packet overhead (MAC headers and trailers), collisions, or from exchange of extra control packets.

Fundamentals of (wireless) MAC Protocols

Requirements and design constraints for wireless MAC protocols:

- ❖ Collisions can happen if the MAC protocol allows two or more nodes to send packets at the same time.
- ❖ Collisions can result in the inability of the receiver to decode a packet correctly, causing the upper layers to perform a retransmission.
- ❖ For time-critical applications, it is important to provide deterministic or stochastic guarantees on delivery time or minimal available data rate.

Fundamentals of (wireless) MAC Protocols

Requirements and design constraints for wireless MAC protocols:

- ❖ The operation and performance of MAC protocols is heavily influenced by the properties of the underlying physical layer. Since WSNs use a wireless medium, they inherit all the well-known problems of wireless transmission.
- ❖ One problem is time-variable, and sometimes quite high, error rates, which is caused by physical phenomena like slow and fast fading, path loss, attenuation, and man-made or thermal noise.

Fundamentals of (wireless) MAC Protocols

Requirements and design constraints for wireless MAC protocols:

- ❖ Depending on modulation schemes, frequencies, distance between transmitter and receiver, and the propagation environment, instantaneous bit error rates in the range of 10^{-3} ... 10^{-2} can easily be observed.
- ❖ The received power P_{rcvd} decreases with the distance between transmitting and receiving node. This path loss combined with the fact that any transceiver needs a minimum signal strength to demodulate signals successfully leads to a maximum range that a sensor node can reach with a given transmit power.

Fundamentals of (wireless) MAC Protocols

Requirements and design constraints for wireless MAC protocols:

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

Fundamentals of (wireless) MAC Protocols

Requirements and design constraints for wireless MAC protocols:

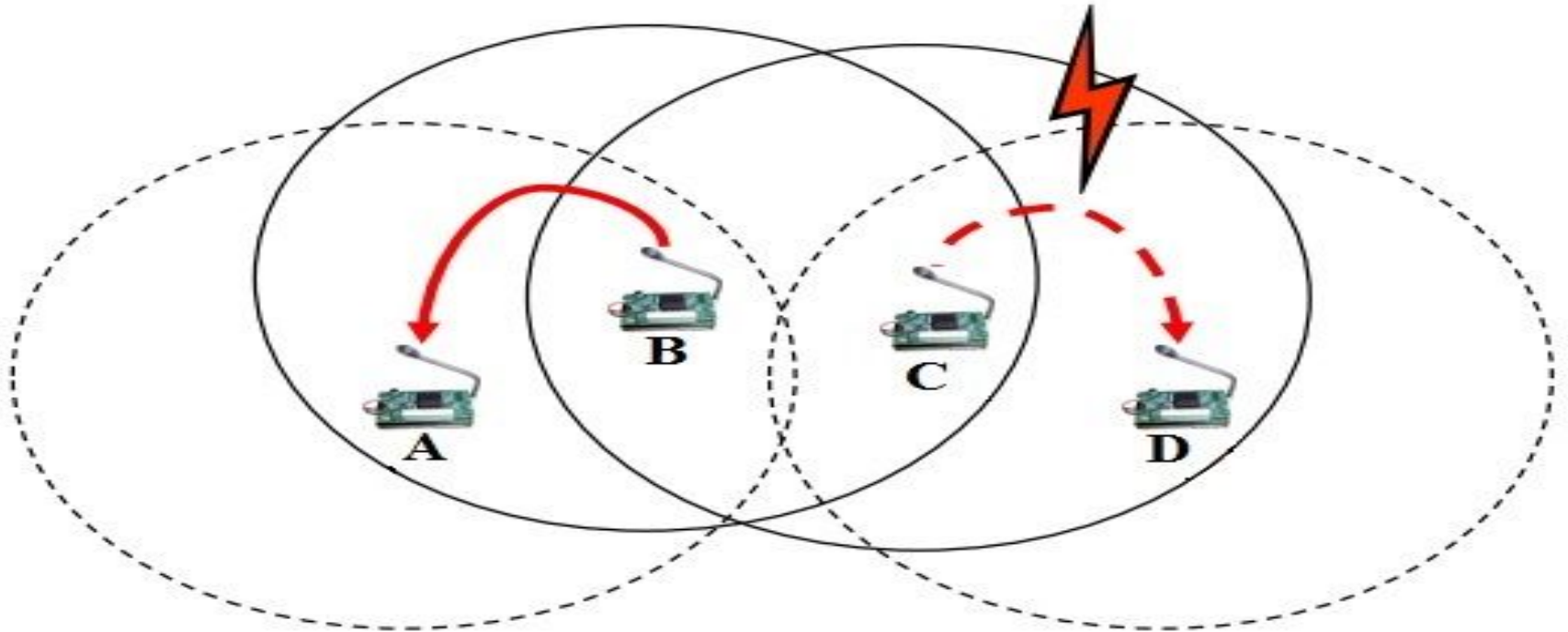
- ❖ we have three nodes A, B, and C that are arranged such that A and B are in mutual range, B and C are in mutual range, but A and C cannot hear each other.
- ❖ Assume that A starts to transmit a packet to B and some time later node C also decides to start a packet transmission.
- ❖ A carrier-sensing operation by C shows an idle medium since C cannot hear A's signals. When C starts its packet, the signals collide at B and both packets are useless.

Fundamentals of (wireless) MAC Protocols

Requirements and design constraints for wireless MAC protocols:

- ❖ Using simple **CSMA** in a hidden-terminal scenario thus leads to needless collisions.
- ❖ In the exposed-terminal scenario, B transmits a packet to A, and some moment later, C wants to transmit a packet to D. Although this would be theoretically possible since both A and D would receive their packets without distortions, the carrier-sense operation performed by C suppresses C's transmission and bandwidth is wasted.

Hidden-terminal scenario (circles indicate transmission



Fundamentals of (wireless) MAC Protocols

Requirements and design constraints for wireless MAC protocols:

- ❖ Using simple CSMA in an exposed terminal scenario thus leads to needless waiting.
- ❖ Two solutions to the hidden-terminal and exposed-terminal problems are busy-tone solutions and the RTS/CTS handshake used in the IEEE 802.11 WLAN standard and first presented in the MACA/MACAW protocols.
- ❖ These will be described in Section 5.1.2 in the context of CSMA protocols

Fundamentals of (wireless) MAC Protocols

Requirements and design constraints for wireless MAC protocols:

- ❖ it is often possible for the transmitter to detect a collision at the receiver immediately and to abort packet transmission. This feature is called collision detection (CD) and is used in Ethernet's CSMA/CD protocol to increase throughput efficiency.
- ❖ Such a collision detection works because of the low attenuation in a wired medium, resulting in similar SNRs at transmitter and receiver.

Fundamentals of (wireless) MAC Protocols

Requirements and design constraints for wireless MAC protocols:

- ❖ when the transmitter reads back the channel signal during transmission and observes a collision, it can infer that there must have been a collision at the receiver too. More importantly, the absence of a collision at the transmitter allows to conclude that there has been no collision at the receiver during the packet transmission.
- ❖ simple wireless transceivers work only in a half-duplex mode, meaning that at any given time either the transmit or the receive circuitry is active but not both.

Fundamentals of (wireless) MAC Protocols

Requirements and design constraints for wireless MAC protocols:

- ❖ Another important problem arises when there is no dedicated frequency band allocated to a wireless sensor network and the WSN has to share its spectrum with other systems.
- ❖ Because of license-free operations, many wireless systems use the so-called ISM bands, with the 2.4 GHz ISM band being a prime example.
- ❖ This specific band is used by several systems, for example, the IEEE 802.11/IEEE 802.11b WLANs, Bluetooth, and the IEEE 802.15.4 WPAN.

Fundamentals of (wireless) MAC Protocols

Requirements and design constraints for wireless MAC protocols:

- ❖ The design of MAC protocols depends on the expected traffic load patterns. If a WSN is deployed to continuously observe a physical phenomenon, for example, the time-dependent temperature distribution in a forest, a continuous and low load with a significant fraction of periodic traffic can be expected.
- ❖ The network is close to idle for a long time and then is faced with a bulk of packets that are to be delivered quickly. A high MAC efficiency is desirable during these overload phases. An example for this class of applications is wildfire observation.

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:

A huge number of (wireless) MAC protocols have been devised during the last thirty years. They can be roughly classified into the following classes:

1. Fixed assignment protocols,
2. Demand assignment protocols,
3. Random access protocols

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Fixed assignment protocols]

- ❖ In this class of protocols, the available resources are divided between the nodes such that the resource assignment is long term and each node can use its resources exclusively without the risk of collisions.
- ❖ Long term means that the assignment is for durations of minutes, hours, or even longer, as opposed to the short-term case where assignments have a scope of a data burst, corresponding to a time horizon of perhaps (tens of) milliseconds.

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Fixed assignment protocols]

- ❖ To account for changes in the topology – for example, due to nodes dying or new nodes being deployed, mobility, or changes in the load patterns – signaling mechanisms are needed in fixed assignment protocols to renegotiate the assignment of resources to nodes.
- ❖ This poses questions about the scalability of these protocols. Typical protocols of this class are TDMA, FDMA, CDMA, and SDMA.

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Fixed assignment protocols]

- ❖ The **Time Division Multiple Access (TDMA)** scheme subdivides the time axis into fixed-length superframes and each superframe is again subdivided into a fixed number of time slots.
- ❖ These time slots are assigned to nodes exclusively and hence the node can transmit in this time slot periodically in every superframe. TDMA requires tight time synchronization between nodes to avoid overlapping of signals in adjacent time slots.

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Fixed assignment protocols]

- ❖ In **Frequency Division Multiple Access (FDMA)**, the available frequency band is subdivided into a number of subchannels and these are assigned to nodes, which can transmit exclusively on their channel.
- ❖ This scheme requires frequency synchronization, relatively narrowband filters, and the ability of a receiver to tune to the channel used by a transmitter.

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Fixed assignment protocols]

- ❖ FDMA transceiver tends to be more complex than a TDMA transceiver.
- ❖ In Code Division Multiple Access (CDMA) schemes, the nodes spread their signals over a much larger bandwidth than needed, using different codes to separate their transmissions.
- ❖ The receiver has to know the code used by the transmitter; all parallel transmissions using other codes appear as noise. Crucial to CDMA is the code management.

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Fixed assignment protocols]

- ❖ **Space Division Multiple Access (SDMA)**, the spatial separation of nodes is used to separate their transmissions. SDMA requires arrays of antennas and sophisticated signal processing techniques and cannot be considered a candidate technology for WSNs.

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Demand assignment protocols]

- ❖ In **demand assignment protocols**, the exclusive allocation of resources to nodes is made on a short-term basis, typically the duration of a data burst. This class of protocols can be broadly subdivided into centralized and distributed protocols.
- ❖ In central control protocols (examples are the HIPERLAN/2 protocol, DQRUMA, or the MASCARA protocol; polling schemes can also be subsumed under this class),

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Demand assignment protocols]

- ❖ The nodes send out requests for bandwidth allocation to a central node that either accepts or rejects the requests.
- ❖ In case of successful allocation, a confirmation is transmitted back to the requesting node along with a description of the allocated resource.
- ❖ **for example**, the numbers and positions of assigned time slots in a TDMA system and the duration of allocation.

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Demand assignment protocols]

- ❖ The node can use these resources exclusively. The submission of requests from nodes to the central station is often done contention based, that is, using a random access protocol on a dedicated (logical) signaling channel.
- ❖ Another option is to let the central station poll its associated nodes. In addition, the nodes often piggyback requests onto data packets transmitted in their exclusive data slots, thus avoiding transmission of separate request packets.

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Demand assignment protocols]

- ❖ The central node needs to be switched on all the time and is responsible for resource allocation. Resource deallocation is often done implicitly:
- ❖ when a node does not use its time slots any more, the central node can allocate these to other nodes.
- ❖ This way, nodes do not need to send extra deallocation packets. Summarizing, the central node performs a lot of activities, it must be constantly awake, and thus needs lots of energy.

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Demand assignment protocols]

- ❖ This class of protocols is a good choice if a sufficient number of energy-unconstrained nodes are present and the duties of the central station can be moved to these. An example is the IEEE 802.15.4 protocol,
- ❖ If there are no unconstrained nodes, a suitable approach is to rotate the central station duties among the nodes like, for example, in the LEACH protocol.

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Demand assignment protocols]

- ❖ An example of **distributed demand assignment protocols** are token-passing protocols like IEEE 802.4 Token Bus.
- ❖ The right to initiate transmissions is tied to reception of a small special token frame. The token frame is rotated among nodes organized in a logical ring on top of a broadcast medium.
- ❖ Special ring management procedures are needed to include and exclude nodes from the ring or to correct failures like lost tokens.

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Demand assignment protocols]

- ❖ since token circulation times are variable, a node must always be able to receive the token to avoid breaking the logical ring.
- ❖ A nodes transceiver must be switched on most of the time. In addition, maintaining a logical ring in face of frequent topology changes is not an easy task and involves significant signaling traffic besides the token frames themselves.

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Random access protocols]

- ❖ The nodes are uncoordinated, and the protocols operate in a fully distributed manner.
- ❖ **Random access protocols** often incorporate a random element, for example, by exploiting random packet arrival times, setting timers to random values, and so on. One of the first and still very important random access protocols is the ALOHA or slotted ALOHA protocol, developed at the University of Hawaii.

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Random access protocols]

- ❖ In the pure ALOHA protocol, a node wanting to transmit a new packet transmits it immediately. There is no coordination with other nodes and the protocol thus accepts the risk of collisions at the receiver.
- ❖ To detect this, the receiver is required to send an immediate acknowledgment for a properly received packet.
- ❖ The transmitter interprets the lack of an acknowledgment frame as a sign of a collision, backs off for a random time, and starts the next trial.

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Random access protocols]

- ❖ ALOHA provides short access and transmission delays under light loads; under heavier loads, the number of collisions increases, which in turn decreases the throughput efficiency and increases the transmission delays.
- ❖ In slotted ALOHA, the time is subdivided into time slots and a node is allowed to start a packet transmission only at the beginning of a slot. A slot is large enough to accommodate a maximum-length packet.

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Random access protocols]

- ❖ If any node wants to start later, it has to wait for the beginning of the next time slot and has thus no chance to destroy the node's packet. In short, the synchronization reduces the probability of collisions and slotted ALOHA has a higher throughput than pure ALOHA.

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Random access protocols]

- ❖ In the class of **CSMA** protocols, a transmitting node tries to be respectful to ongoing transmissions.
- ❖ First, the node is required to listen to the medium; this is called **carrier sensing**. If the medium is found to be idle, the node starts transmission.
- ❖ If the medium is found busy, the node defers its transmission for an amount of time determined by one of several possible algorithms.
- ❖ For example, in nonpersistent CSMA, the node draws a random waiting time, after which the medium is sensed again.

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Random access protocols]

- ❖ In different **persistent CSMA** variants, after sensing that the medium is busy, the node awaits the end of the ongoing transmission and then behaves according to a **backoff algorithm**. In many of these backoff algorithms, the time after the end of the previous frame is subdivided into time slots.
- ❖ In the backoff algorithm executed by the IEEE 802.11 **Distributed Coordination Function (DCF)**, a node transmitting a new frame picks a random value from the current **contention window** and starts a timer with this value.

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Random access protocols]

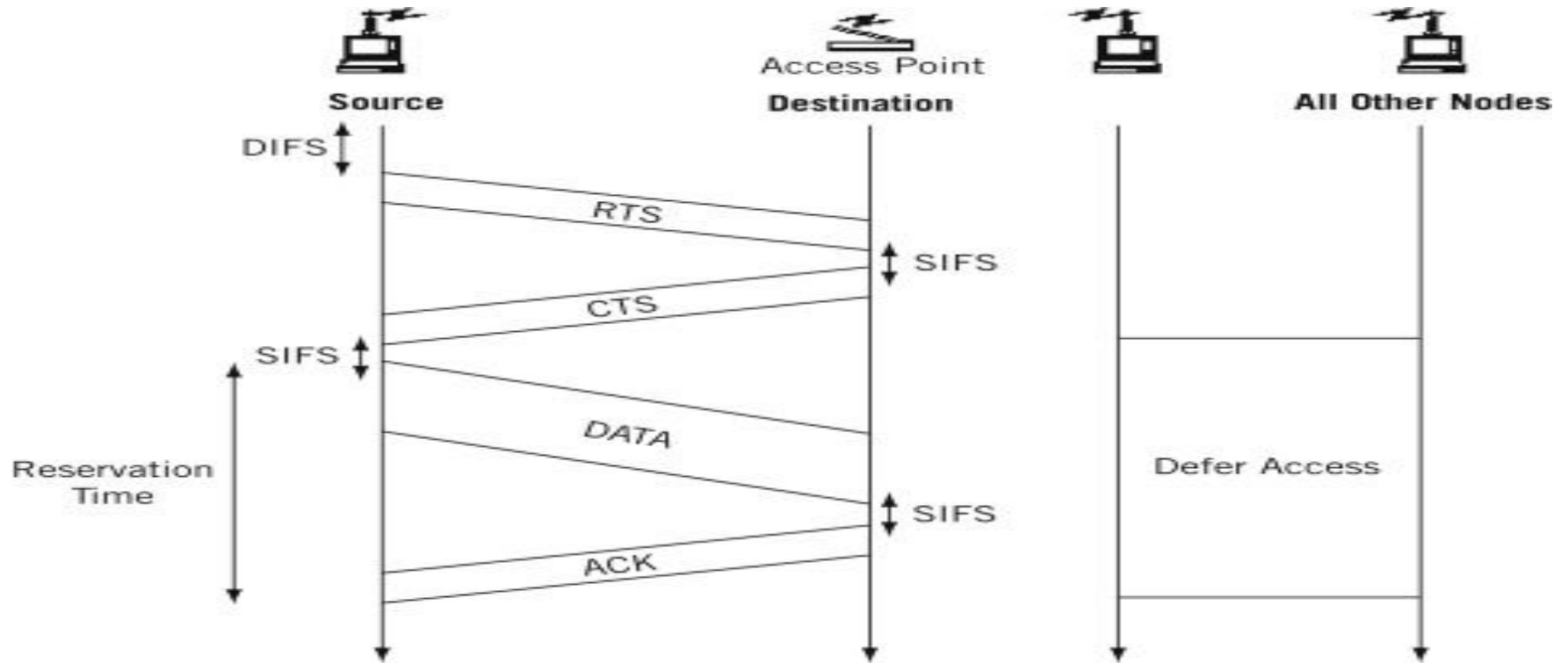
- ❖ The timer is decremented after each slot. If another node starts in the meantime, the timer is suspended and resumed after the next frame ends and contention continues. If the timer decrements to zero, the node transmits its frame.
- ❖ When a transmission error occurs (indicated, for example, by a missing acknowledgment frame), the size of the contention window is increased according to a modified binary exponential backoff procedure.

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Random access protocols]

- ❖ The carrier-sense protocols are susceptible to the hidden-terminal problem since interference at the receiver cannot be detected by the transmitter. This problem may cause packet collisions.
- ❖ The energy spent on collided packets is wasted and the packets have to be retransmitted.
- ❖ Several approaches have appeared to solve or at least to reduce the hidden-terminal problem; the busy-tone solution and **the RTS/CTS handshake**.

RTS/CTS handshake in IEEE 802.11



Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Random access protocols]

- ❖ A node that wishes to transmit a packet first senses the control channel for the presence of a busy tone. If it hears something, the node backs off according to some algorithm, for example similar to nonpersistent CSMA. If it hears nothing, the node starts packet transmission on the data channel.
- ❖ This protocol solves both the hidden- and exposed-terminal problem, given that the busy-tone signal can be heard over the same distance as the data signal.

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Random access protocols]

- ❖ If the busy tone is too weak, a node within radio range of the receiver might start data transmission and destroy the receiver's signal.
- ❖ If the busy tone is too strong, more nodes than necessary suppress their transmissions.
- ❖ The control channel does not need much bandwidth but a narrow bandwidth channel requires good frequency synchronization. A solution with two busy tones, one sent by the receiver and the other by the transmitter node,

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Random access protocols]

- ❖ One solution approach is to ensure that CTS packets are longer than RTS packets. For an explanation, consider the right part of Figure 5.5. Here, even if B's CTS arrives at C immediately after C starts its RTS, it lasts long enough that C has a chance to turn its transceiver into receive mode and to sense B's signal.
- ❖ An additional protocol rule states that in such a case node C has to defer any further transmission for a sufficiently long time to accommodate one maximum-length data packet.

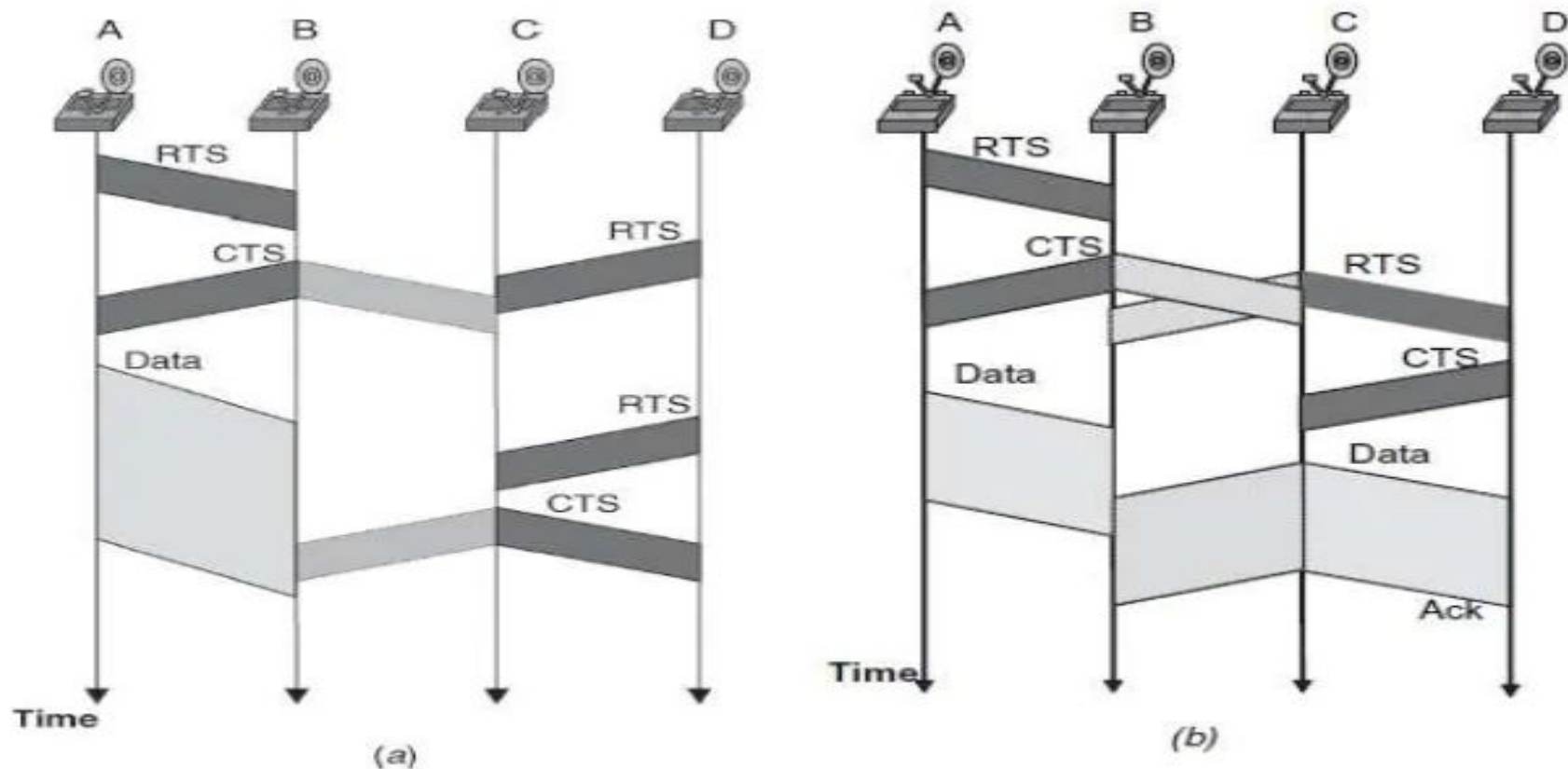


Figure 5.5 Collision avoidance failure using RTS/CTS handshake.

Fundamentals of (wireless) MAC Protocols

2 .Important classes of MAC protocols:[Random access protocols]

- ❖ Hence, the data packet between A and B can be transmitted without distortion.
- ❖ A further problem of the RTS/CTS handshake is its significant overhead of two control packets per data packet, not counting the acknowledgment packet. If the data packet is small, this overhead might not pay off and it may be simpler to use some plain CSMA variant.
- ❖ For long packets, the overhead of the RTS/CTS handshake can be neglected, but long packets are more likely to be hit by channel errors and must be retransmitted entirely, wasting precious energy (channel errors often hit only a few bits).

Welcome
to your new
class!



Fundamentals of (wireless) MAC Protocols

3 MAC protocols for wireless sensor networks:

- ❖ For the case of WSNs, the balance of requirements is different from traditional (wireless) networks. Additional requirements come up, first and foremost, the need to conserve energy.
- ❖ The importance of energy efficiency for the design of MAC protocols is relatively new and many of the “classical” protocols like ALOHA and CSMA contain no provisions toward this goal.

Fundamentals of (wireless) MAC Protocols

3 MAC protocols for wireless sensor networks:

- ❖ Other typical performance figures like fairness, throughput, or delay tend to play a minor role in sensor networks.
- ❖ Fairness is not important since the nodes in a WSN do not represent individuals competing for bandwidth, but they collaborate to achieve a common goal.
- ❖ The access/transmission delay performance is traded against energy conservation, and throughput is mostly not an issue either.

Fundamentals of (wireless) MAC Protocols

3 MAC protocols for wireless sensor networks:

Energy problems on the MAC layer:

- ❖ Collisions
- ❖ Overhearing
- ❖ Protocol overhead
- ❖ Idle listening

Collisions

- ❖ collisions incur useless receive costs at the destination node, useless transmit costs at the source node, and the prospect to expend further energy upon packet retransmission.
- ❖ Hence, collisions should be avoided, either by design (fixed assignment/TDMA or demand assignment protocols) or by appropriate collision avoidance/hidden-terminal procedures in CSMA protocols.
- ❖ However, if it can be guaranteed for the particular sensor network application at hand that the load is always sufficiently low, collisions are no problem.

Overhearing

- ❖ Unicast frames have one source and one destination node. However, the wireless medium is a broadcast medium and all the source's neighbors that are in receive state hear a packet and drop it when it is not destined to them; these nodes overhear the packet.
- ❖ The higher node densities overhearing avoidance can save significant amounts of energy. On the other hand, overhearing is sometimes desirable, for example, when collecting neighborhood information or estimating the current traffic load for management purposes

Protocol overhead

- ❖ Protocol overhead is induced by MAC-related control frames like, for example, RTS and CTS packets or request packets in demand assignment protocols, and furthermore by per-packet overhead like packet headers and trailers.
- ❖ A design constraint somewhat related to energy concerns is the requirement for low complexity operation. Sensor nodes shall be simple and cheap and cannot offer plentiful resources in terms of processing power, memory, or energy. Therefore, computationally expensive operations like complex scheduling algorithms should be avoided.

Idle listening

- ❖ A node being in idle state is ready to receive a packet but is not currently receiving anything. This readiness is costly and useless in case of low network loads; for many radio modems, the idle state still consumes significant energy.
- ❖ Switching off the transceiver is a solution; however, since mode changes also cost energy, their frequency should be kept at “reasonable” levels.
- ❖ TDMA-based protocols offer an implicit solution to this problem, since a node having assigned a time slot and exchanging (transmitting/receiving) data only during this slot can safely switch off its transceiver in all other time slots.

Fundamentals of (wireless) MAC Protocols

- ❖ Most of the MAC protocols developed for wireless sensor networks attack one or more of these problems to reduce energy consumption
- ❖ A design constraint somewhat related to energy concerns is the requirement for low complexity operation.
- ❖ Sensor nodes shall be simple and cheap and cannot offer plentiful resources in terms of processing power, memory, or energy. Therefore, computationally expensive operations like complex scheduling algorithms should be avoided.

Fundamentals of (wireless) MAC Protocols

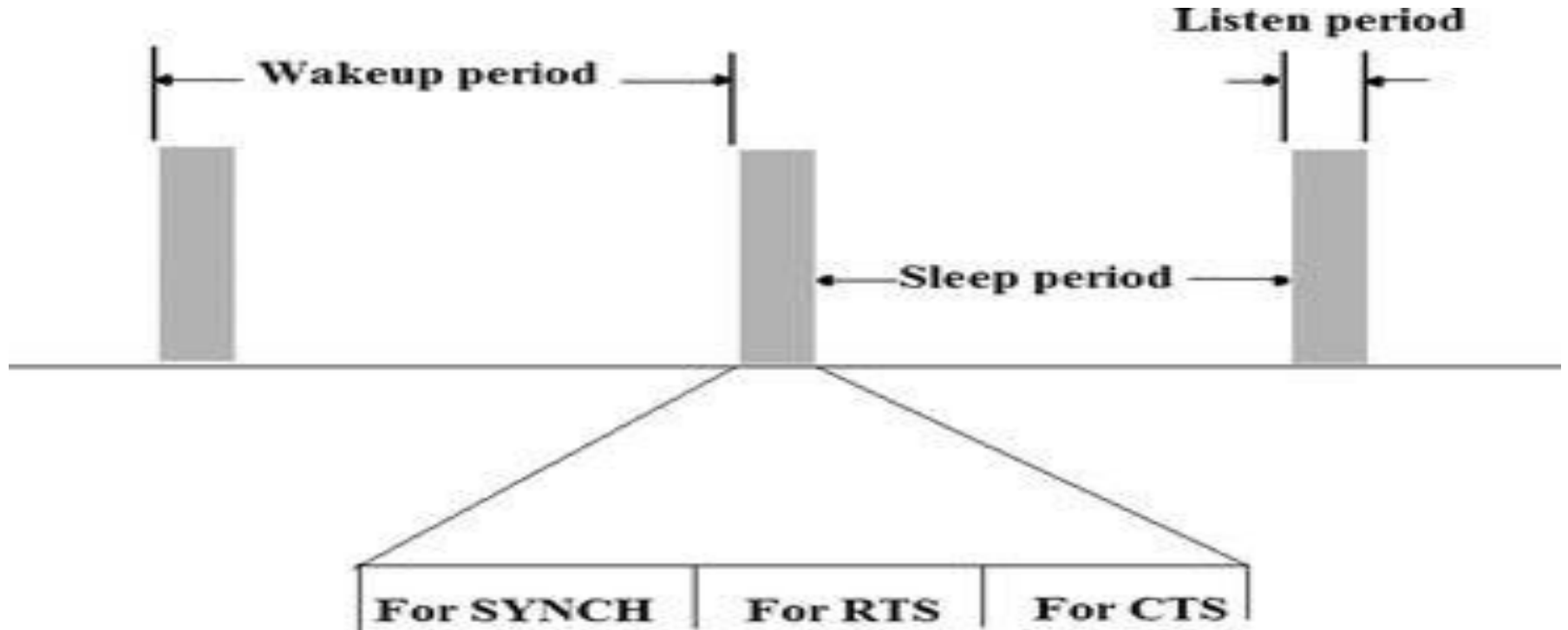
- ❖ The desire to use cheap node hardware includes components like oscillators and clocks. Consequently, the designer of MAC protocols should bear in mind that very tight time synchronization (as needed for TDMA with small time slots) would require frequent resynchronization of neighboring nodes, which can consume significant energy.



2 Low duty cycle protocols and wakeup concepts

- ❖ **Low duty cycle protocols** try to avoid spending (much) time in the idle state and to reduce the communication activities of a sensor node to a minimum. In an ideal case, the sleep state is left only when a node is about to transmit or receive packets.
- ❖ A periodic wakeup scheme is used. Such schemes exist in different flavors. One is the cycled receiver approach. In this approach, nodes spend most of their time in the sleep mode and wake up periodically to receive packets from other nodes.

2 Low duty cycle protocols and wakeup concepts



2 Low duty cycle protocols and wakeup concepts

- ❖ a node A listens onto the channel during its listen period and goes back into sleep mode when no other node takes the opportunity to direct a packet to A. A potential transmitter B must acquire knowledge about A's listen periods to send its packet at the right time.
- ❖ A transmit a short beacon at the beginning of its listen period to indicate its willingness to receive packets. Another method is to let node B send frequent request packets until one of them hits A's listen period and is really answered by A.

2 Low duty cycle protocols and wakeup concepts

- ❖ Node A only receives packets during its listen period. If node A itself wants to transmit packets, it must acquire the target's listen period. A whole cycle consisting of sleep period and listen period is also called a wakeup period. The ratio of the listen period length to the wakeup period length is also called the node's duty cycle.



2 Low duty cycle protocols and wakeup concepts

- ❖ By choosing a small duty cycle, the transceiver is in sleep mode most of the time, avoiding idle listening and conserving energy.
- ❖ By choosing a small duty cycle, the traffic directed from neighboring nodes to a given node concentrates on a small time window (the listen period) and in heavy load situations significant competition can occur.

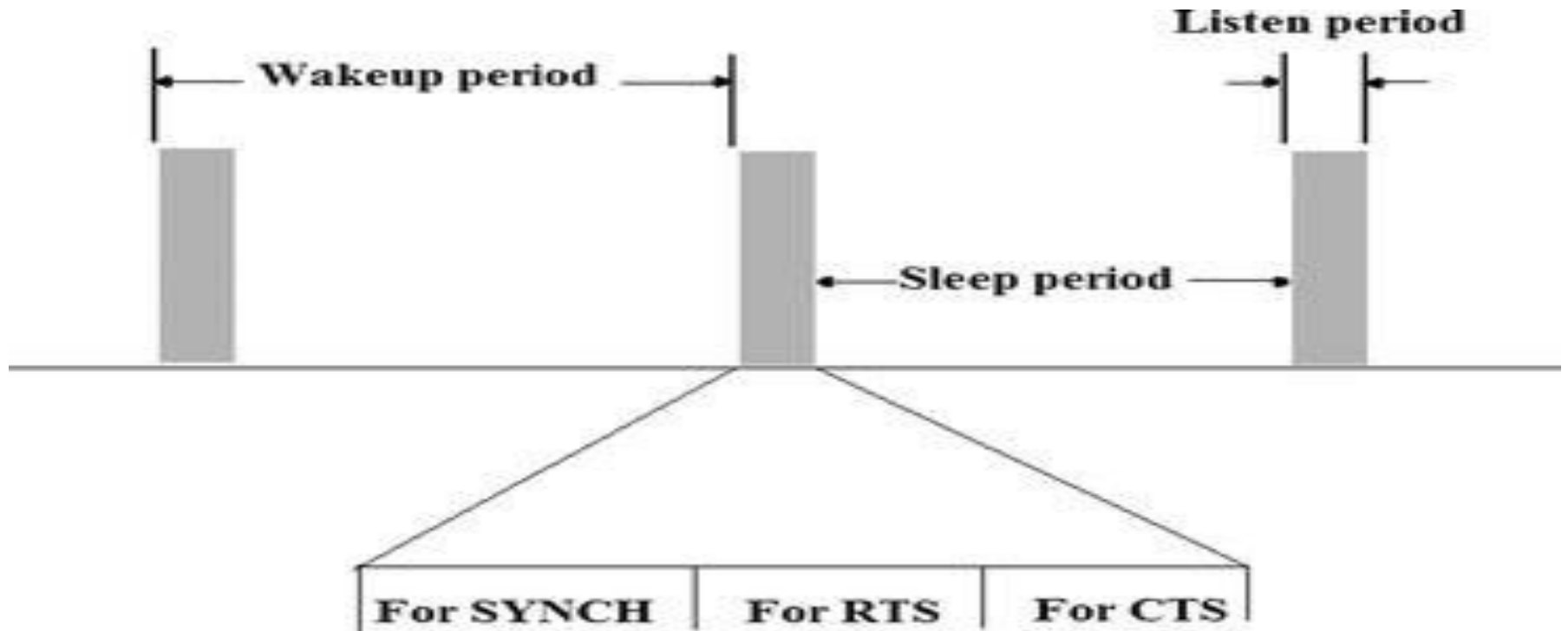
2 Low duty cycle protocols and wakeup concepts

- ❖ Choosing a long sleep period induces a significant per-hop latency, since a prospective transmitter node has to wait an average of half a sleep period before the receiver can accept packets.
- ❖ In the multihop case, the per-hop latencies add up and create significant end-to-end latencies. Sleep phases should not be too short lest the start-up costs outweigh the benefits.

S-MAC

- ❖ 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, 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:

S-MAC



S-MAC

- ❖ In the first phase (**SYNCH phase**), node x accepts SYNCH packets from its neighbors. In these packets, the neighbors describe their own schedule and x stores their schedule in a table (the schedule table). Node x 's SYNCH phase is subdivided into time slots and x 's neighbors contend according to a CSMA scheme with additional backoff.
- ❖ It is not required that x broadcasts its schedule in every of y 's wakeup periods. However, for reasons of time synchronization and to allow new nodes to learn their local network topology, x should send SYNCH packets periodically. The according period is called synchronization period.

S-MAC

- ❖ In the second phase (**RTS phase**), x listens for RTS packets from neighboring nodes. In S-MAC, the RTS/CTS handshake of data packets due 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 the previous phase. After this, the packet exchange continues, extending into x's nominal sleep time.

S-MAC

- ❖ In general, when competing for the medium, the nodes use the RTS/CTS handshake, including the virtual carrier-sense mechanism, whereby a node maintains a NAV variable.
- ❖ The NAV mechanism can 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 packets are dropped and the nodes use CSMA with backoff.

S-MAC

- ❖ If we can arrange that the schedules of node x and its neighbors are synchronized, node x and all its neighbors wake up at the same time and x can reach all of them with a single SYNCH packet.
- ❖ 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 of data packets is not influenced by virtual clustering
- ❖

S-MAC

- ❖ The periodic wakeup scheme adopted by S-MAC allows nodes to spend much time in the sleep mode, but there is also a price to pay in terms of latency. Without further modifications, the per-hop latency of S-MAC will be approximately equal to the sleep period on average when all nodes follow the same schedule.
- ❖ The **adaptive-listening scheme**, which roughly halves the 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.

S-MAC

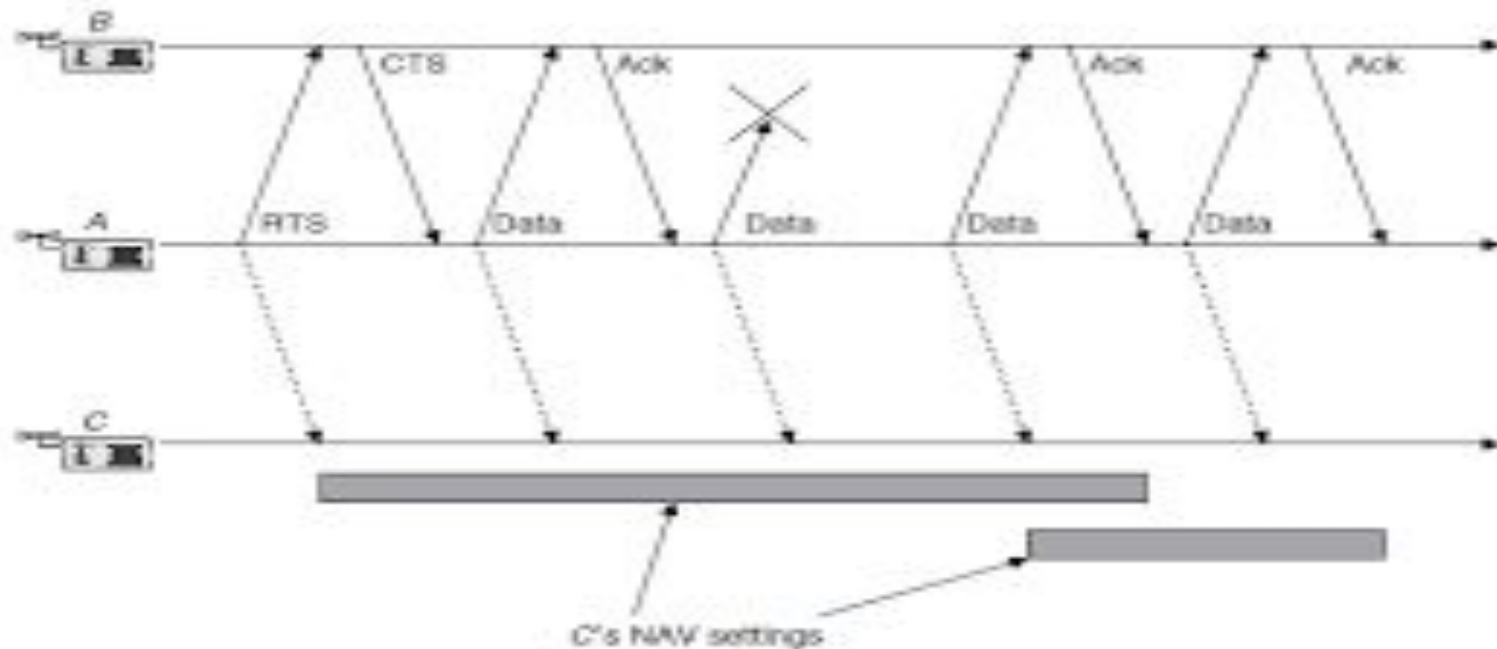


Figure 5.7 S-MAC fragmentation and NAV setting

S-MAC

- ❖ 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,). S-MAC includes a fragmentation scheme working as follows.

S-MAC

- ❖ 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.

S-MAC

- ❖ 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, there is 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

S-MAC

- ❖ 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.

S-MAC

- ❖ The approach taken by S-MAC reduces the latency of complete messages by suppressing intertwined transmissions of other packets. Therefore, in a sense, this protocol is unfair because single nodes can block the medium for long time.
- ❖ However, 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.



TYPES OF WIRELESS SENSOR NETWORK

Dr.P.Venkatesan
Associate Professor/ECE
Sri Chandrasekharendra Saraswathi Viswa
Mahavidyalaya University (SCSVMV)
Kanchipuram, Tamil Nadu, India.

Types of Wireless Sensor Networks

Depending on the environment, the **types of networks** are decided so that those can be deployed underwater, underground, on land, and so on. Different types of WSNs include:

1. Terrestrial WSNs
2. Underground WSNs
3. Underwater WSNs
4. Multimedia WSNs
5. Mobile WSNs

Terrestrial WSNs

Terrestrial WSNs are capable of communicating base stations efficiently, and consist of hundreds to thousands of wireless sensor nodes deployed either in an unstructured (ad hoc) or structured (Pre-planned) manner. In an unstructured mode, the sensor nodes are randomly distributed within the target area that is dropped from a fixed plane. The preplanned or structured mode considers optimal placement, grid placement, and 2D, 3D placement models.

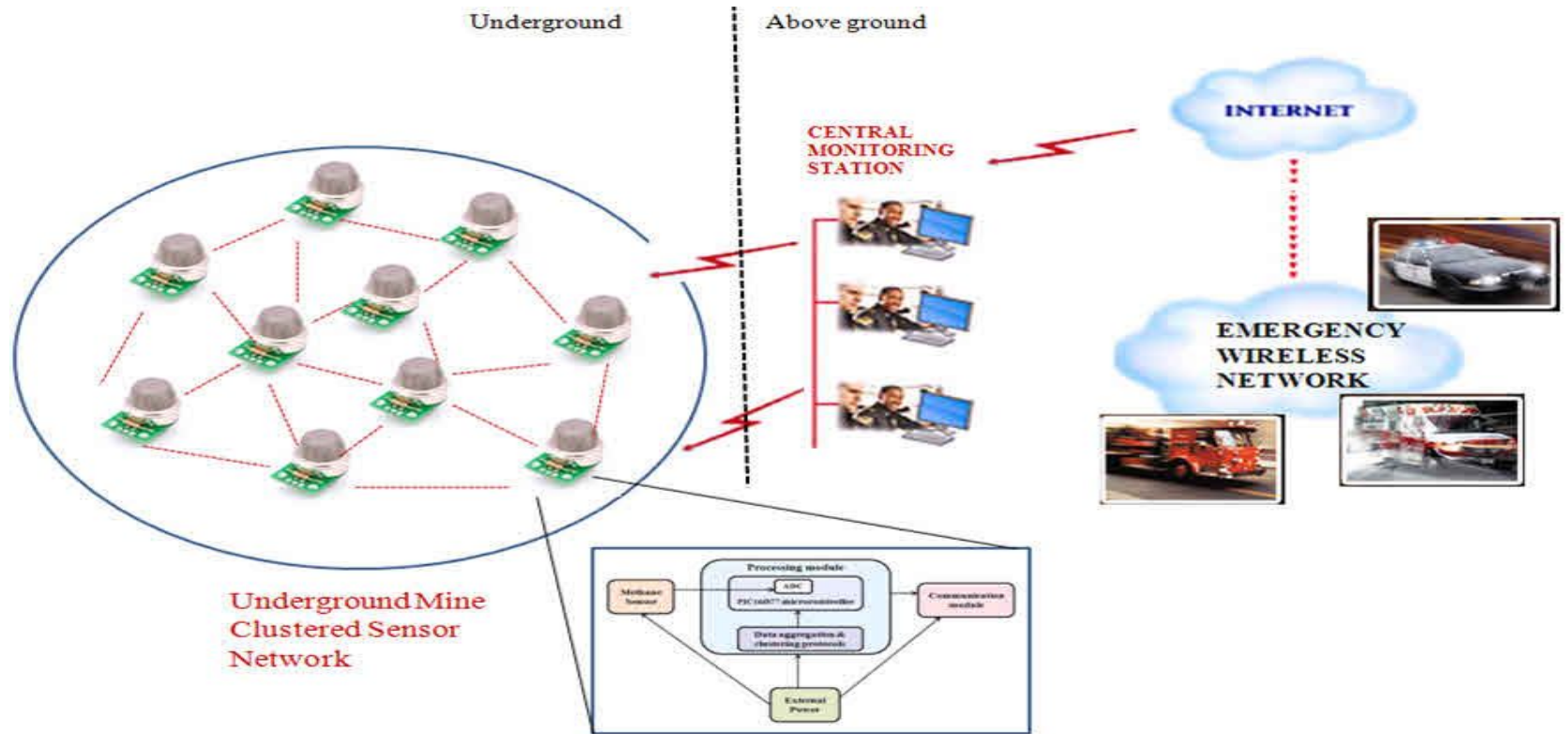
In this WSN, the **battery power** is limited; however, the battery is equipped with solar cells as a secondary power source. The Energy conservation of these WSNs is achieved by using low duty cycle operations, minimizing delays, and optimal routing, and so on.

Underground WSNs

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

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

Underground WSNs

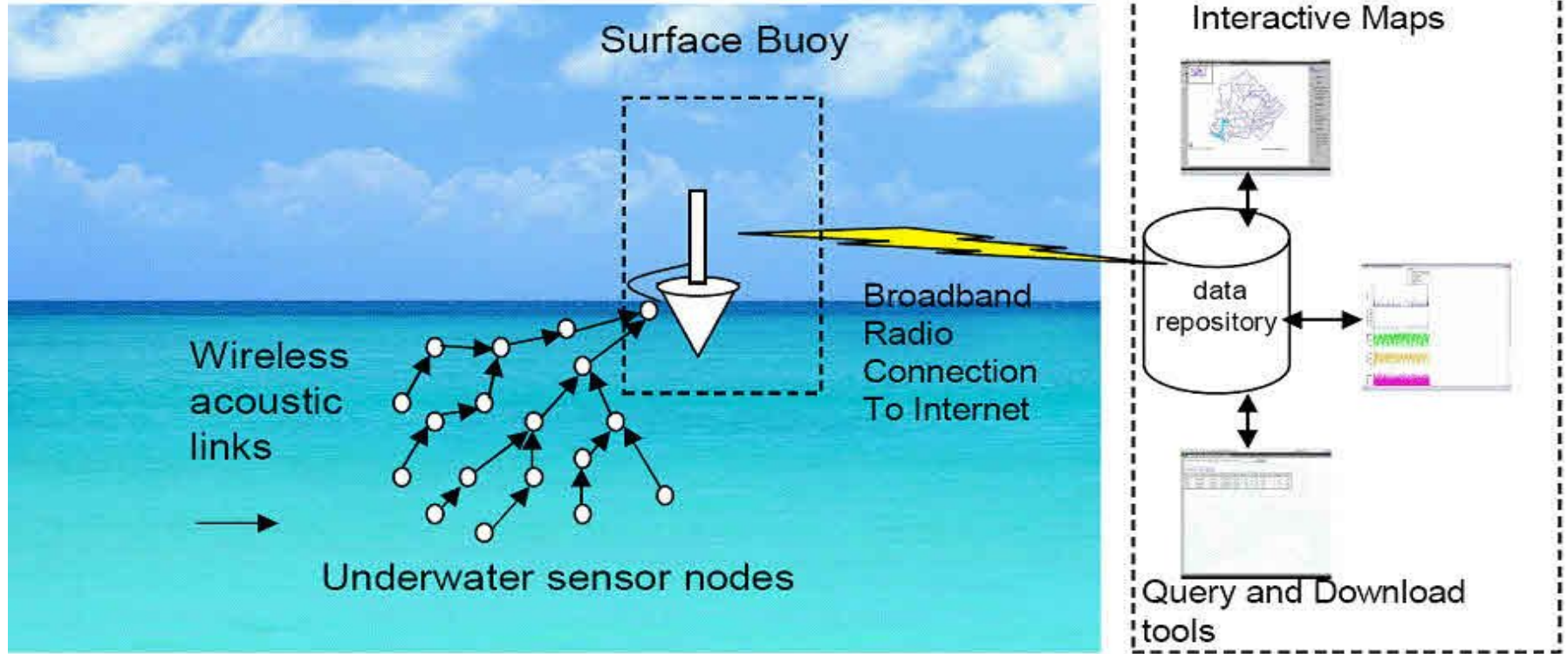


Under Water WSNs

More than 70% of the earth is occupied with water. These networks consist of several sensor nodes and vehicles deployed underwater. Autonomous underwater vehicles are used for gathering data from these sensor nodes. A challenge of underwater communication is a long propagation delay, and bandwidth and sensor failures.

Underwater, WSNs are equipped with a limited battery that cannot be recharged or replaced. The issue of energy conservation for underwater WSNs involves the development of underwater communication and networking techniques.

Under Water WSNs

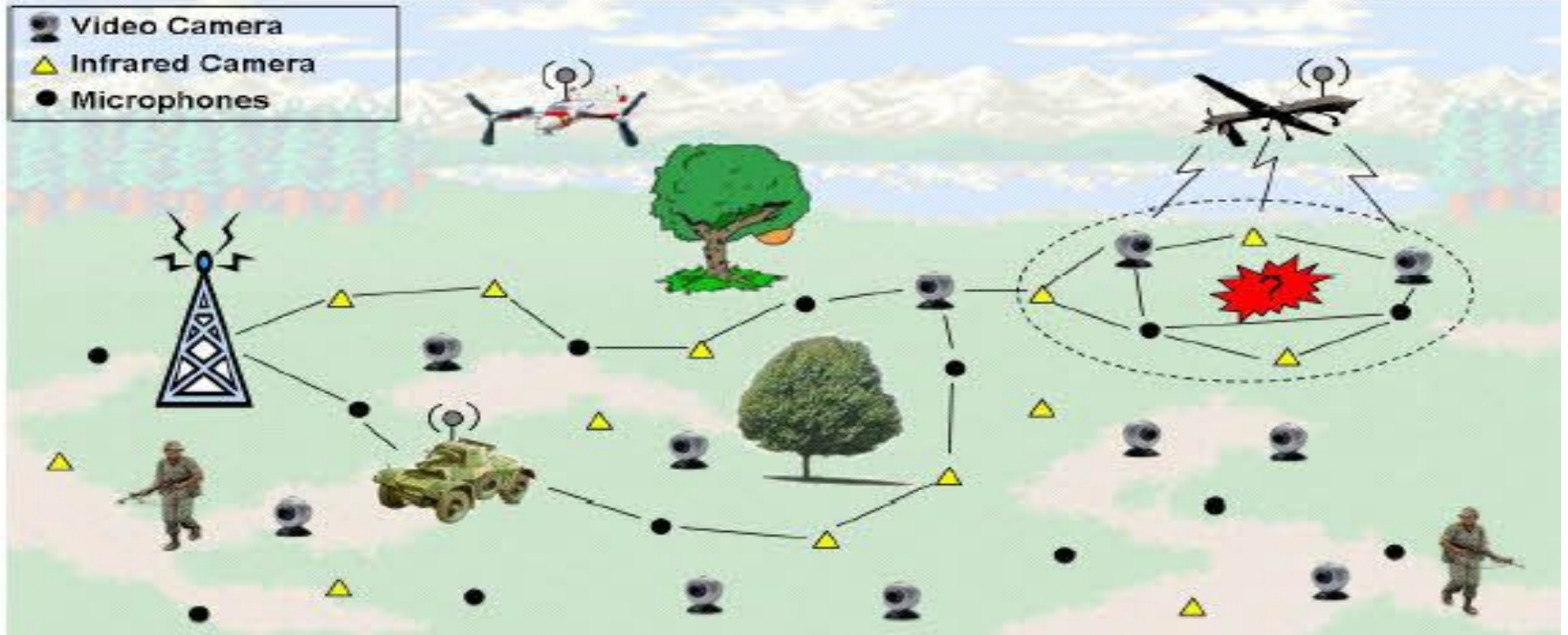


Multimedia WSNs

Multimedia wireless sensor networks have been proposed to enable tracking and monitoring of events in the form of multimedia, such as imaging, video, and audio. These networks consist of low-cost sensor nodes equipped with microphones and cameras. These nodes are interconnected with each other over a wireless connection for data compression, data retrieval, and correlation.

The challenges with the multimedia WSN include high energy consumption, high bandwidth requirements, data processing, and compressing techniques. In addition to this, multimedia contents require high bandwidth for the content to be delivered properly and easily.

Multimedia WSNs



Mobile WSNs

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



Thank You!

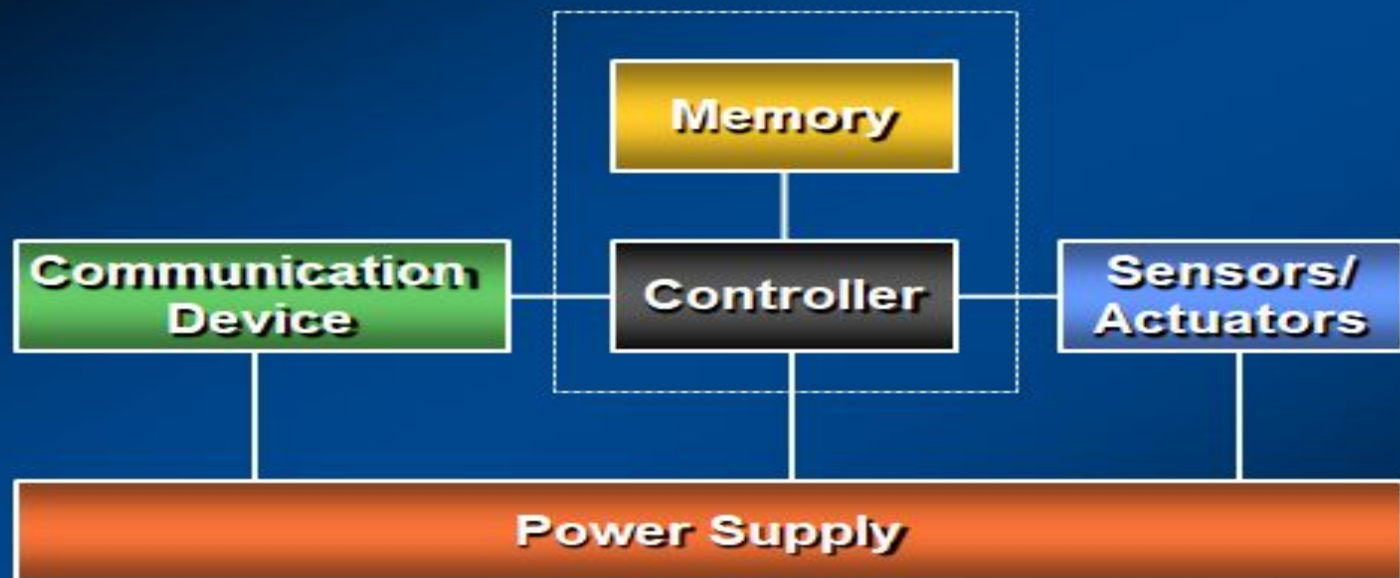
Single Node Architecture

Dr.P.Venkatesan
Associate Professor/ECE
SCSVMV University

Outline

- **Main components of a wireless sensor node**
 - Processor, radio, sensors, batteries
- **Energy supply and consumption**
- **Operating systems and execution environments**
 - IWING's MoteLib
 - TinyOS
 - Contiki
- **Sample implementations**

Main Components



Controller

- Main options:
 - **Microcontroller** – general purpose processor, optimized for embedded applications, low power consumption
 - **DSP** – optimized for signal processing tasks, not suitable here
 - **FPGA** – may be good for testing
 - **ASIC** – only when peak performance is needed, no flexibility

Microcontroller Examples

- **Texas Instruments MSP430**

- 16-bit RISC core, 4 MHz
- Up to 120 KB flash
- 2-10 KB RAM
- 12 ADCs, RT clock

- **Atmel ATmega**

- 8-bit controller, 8 MHz
- Up to 128KB Flash
- 4 KB RAM



Communication Device

- Medium options
 - Electromagnetic, RF
 - Electromagnetic, optical
 - Ultrasound



Transceiver Characteristics

- Service to upper layer: packet, byte, bit
- Power consumption
- Supported frequency, multiple channels
- Data rate
- Modulation
- Power control
- Communication range
- etc.

Transceiver States

- Transceivers can be put into different operational *states*, typically:

- **Transmit**

- **Receive**

- **Idle** – ready to receive, but not doing so

- **Sleep** – significant parts of the transceiver are switched off

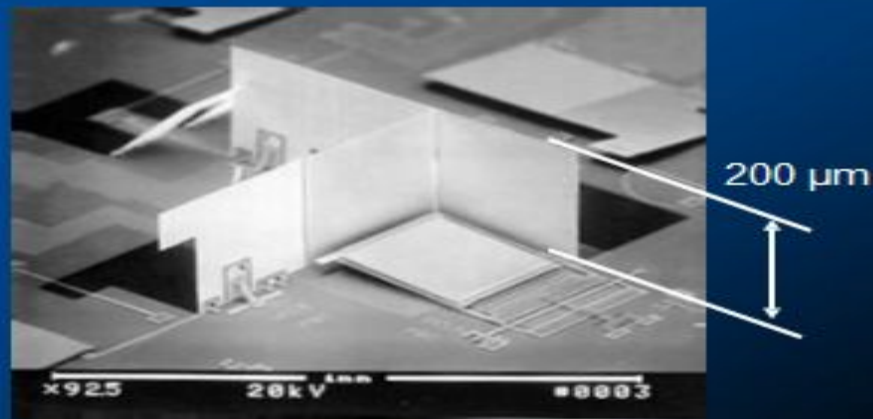


Wakeup Receivers

- When to switch on a receiver is not clear
 - Contention-based MAC protocols: Receiver is always on
 - TDMA-based MAC protocols: Synchronization overhead, inflexible
- Desirable: Receiver that can (only) check for incoming messages
 - When signal detected, wake up main receiver for actual reception
 - Ideally: *Wakeup receiver* can already process simple addresses
 - Not clear whether they can be actually built, however

Optical Communication

- Optical communication can consume less energy
- Example: passive readout via corner cube reflector
 - Laser is reflected back directly to source if mirrors are at right angles
 - Mirrors can be “tilted” to stop reflecting
 - Allows data to be sent back to laser source



Sensors

- **Main categories**

- **Passive, omnidirectional**

- Examples: light, thermometer, microphones, hygrometer, ...

- **Passive, narrow-beam**

- Example: Camera

- **Active sensors**

- Example: Radar

- **Important parameter: Area of coverage**

- Which region is adequately covered by a given sensor?

Outline

- **Main components of a wireless sensor node**
 - Processor, radio, sensors, batteries
- ***Energy supply and consumption***
- **Operating systems and execution environments**
 - IWING's MoteLib
 - TinyOS
 - Contiki
- **Example implementations**

Energy Supply

- Goal: provide as much energy as possible at smallest cost/volume/weight/recharge time/longevity
 - In WSN, recharging may or may not be an option
- Options
 - **Primary batteries** – not rechargeable
 - **Secondary batteries** – rechargeable, only makes sense in combination with some form of energy harvesting

Energy Supply - Requirements

- Low self-discharge
- Long shelf life
- Capacity under load
- Efficient recharging at low current
- Good relaxation properties (seeming self-recharging)
- Voltage stability (to avoid DC-DC conversion)

Energy Supply - Requirements

- Low self-discharge
- Long shelf life
- Capacity under load
- Efficient recharging at low current
- Good relaxation properties (seeming self-recharging)
- Voltage stability (to avoid DC-DC conversion)

Battery Examples

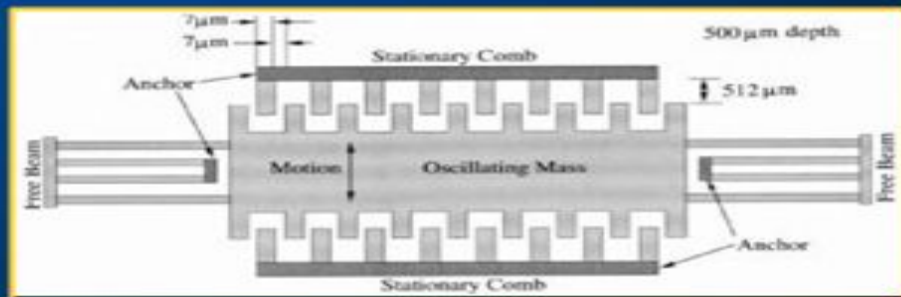
- Energy per volume (Joule/cc):

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

http://en.wikipedia.org/wiki/Energy_density

Energy Harvesting

- **How to recharge a battery?**
 - A laptop: easy, plug into wall socket in the evening
 - A sensor node? – Try to scavenge energy from environment
- **Ambient energy sources**
 - Light ! solar cells – between $10 \mu\text{W}/\text{cm}^2$ and $15 \text{mW}/\text{cm}^2$
 - Temperature gradients – $80 \mu\text{W}/\text{cm}^2$ @ 1V from 5K difference
 - Vibrations – between 0.1 and $10000 \mu\text{W}/\text{cm}^3$
 - Pressure variation (piezo-electric) – $330 \mu\text{W}/\text{cm}^2$ from the heel of a shoe
 - Air/liquid flow (MEMS gas turbines)



Portable Solar Chargers

- **Foldable Solar Chargers**

- <http://www.energy.gov.uk/FoldableChargers.asp>

- **Solargorilla**

- <http://powertraveller.com/iwantsome/primatpower/solargorilla/>



Multiple Power Consumption Modes

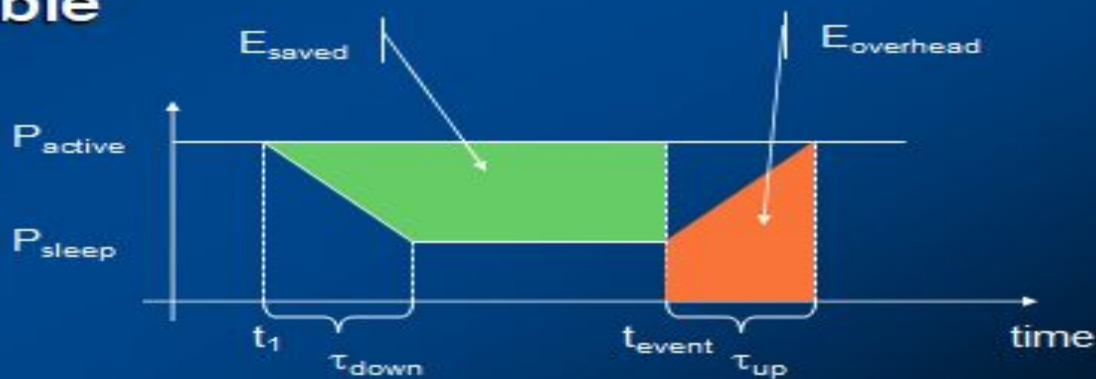
- Do not run sensor node at full operation all the time
 - If nothing to do, switch to power safe mode
- Typical modes
 - Controller: Active, idle, sleep
 - Radio mode: Turn on/off transmitter/receiver, both
 - Strongly depends on hardware
- Questions:
 - When to throttle down?
 - How to wake up again?

Energy Consumption Figures

- **TI MSP 430 (@ 1 MHz, 3V):**
 - Fully operation 1.2 mW
 - One fully operational mode + four sleep modes
 - Deepest sleep mode 0.3 μ W – only woken up by external interrupts (not even timer is running any more)
- **Atmel ATMega**
 - Operational mode: 15 mW active, 6 mW idle
 - Six modes of operations
 - Sleep mode: 75 μ W

Switching Between Modes

- Simplest idea: Greedily switch to lower mode whenever possible
- Problem: Time and power consumption required to reach higher modes not negligible



Should We Switch?

- Switching modes is beneficial if

$$E_{overhead} < E_{saved}$$

which is equivalent to

$$(t_{event} - t_1) > \frac{1}{2} \left(\tau_{down} + \frac{P_{active} + P_{sleep}}{P_{active} - P_{sleep}} \tau_{up} \right)$$

Computation vs. Communication Energy Cost

- **Sending one bit vs. running one instruction**
 - Energy ratio up to **23000:1**
 - I.e., send & receive one KB = running three million instruction
- **So, try to compute instead of communicate whenever possible**
- **Key technique – *in-network processing***
 - Exploit compression schemes, intelligent coding schemes, aggregate data, ...

Outline

- **Main components of a wireless sensor node**
 - Processor, radio, sensors, batteries
- **Energy supply and consumption**
- **Operating systems and execution environments**
 - IWING's MoteLib
 - TinyOS
 - Contiki
- ***Example implementations***

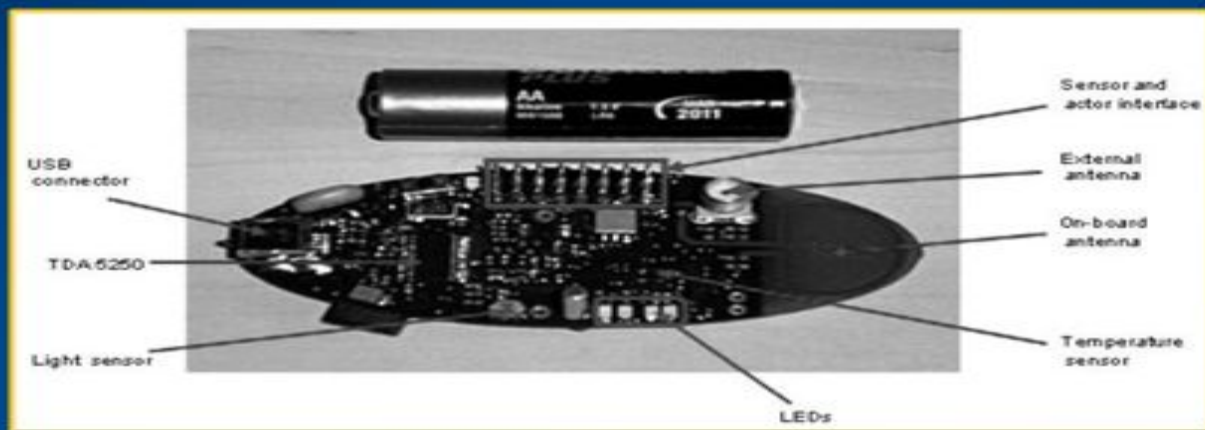
Mica Motes

- By Crossbow, USA
- MCU:
 - Atmel ATMega128L
- Comm: RFM TR1000



EYES Nodes

- By Infineon, EU
- MCU: TI MSP430
- Comm: Infineon radio modem TDA5250



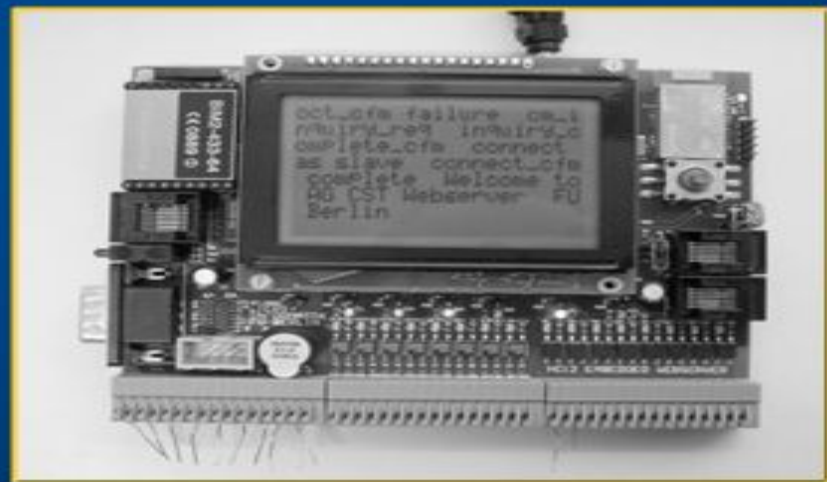
Btnote

- By ETH Zurich
- MCU:
 - Atmel ATMega128L
- Comm:
 - Bluetooth
 - Chipcon CC1000



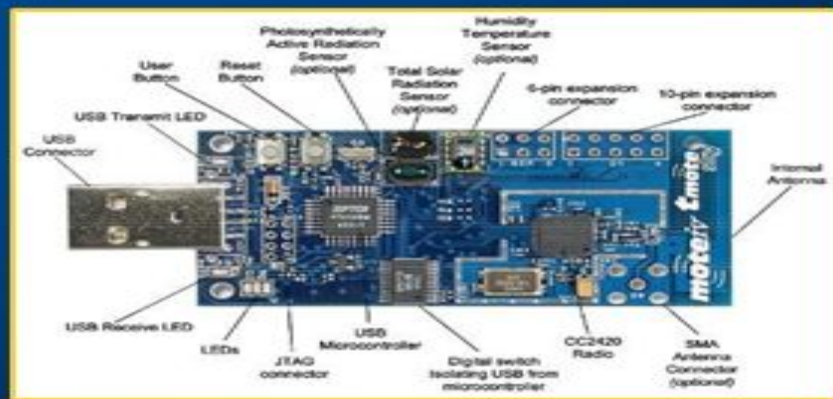
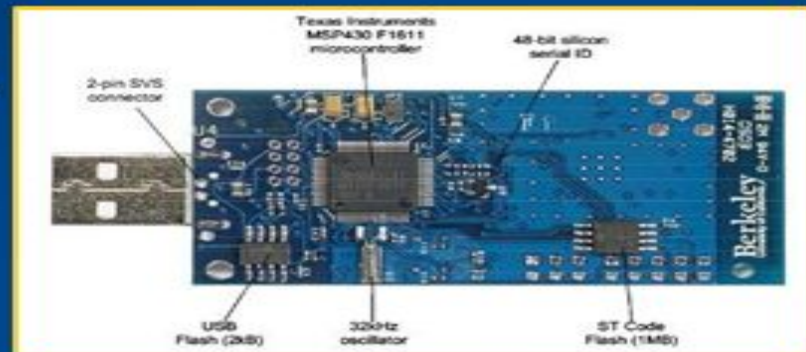
ScatterWeb

- By Computer Systems & Telematics group, Freie Universitat Berlin
- MCU:
 - TI MSP 430
- Comm:
 - Bluetooth, I²C, CAN



Tmote Sky

- By Sentilla (formerly Moteiv), USA
- MCU:
 - TI MSP430
- Comm:
 - Chipcon CC2420 (IEEE 802.15.4)



IRIS Motes

- By Crossbow, USA
- MCU: ATmega128L
- Comm: Atmel's RF230 (IEEE 802.15.4)
- 3x radio range compared to Tmote
- "Postage-stamp" form factor costs as low as \$29 per unit (when purchased in large volumes)

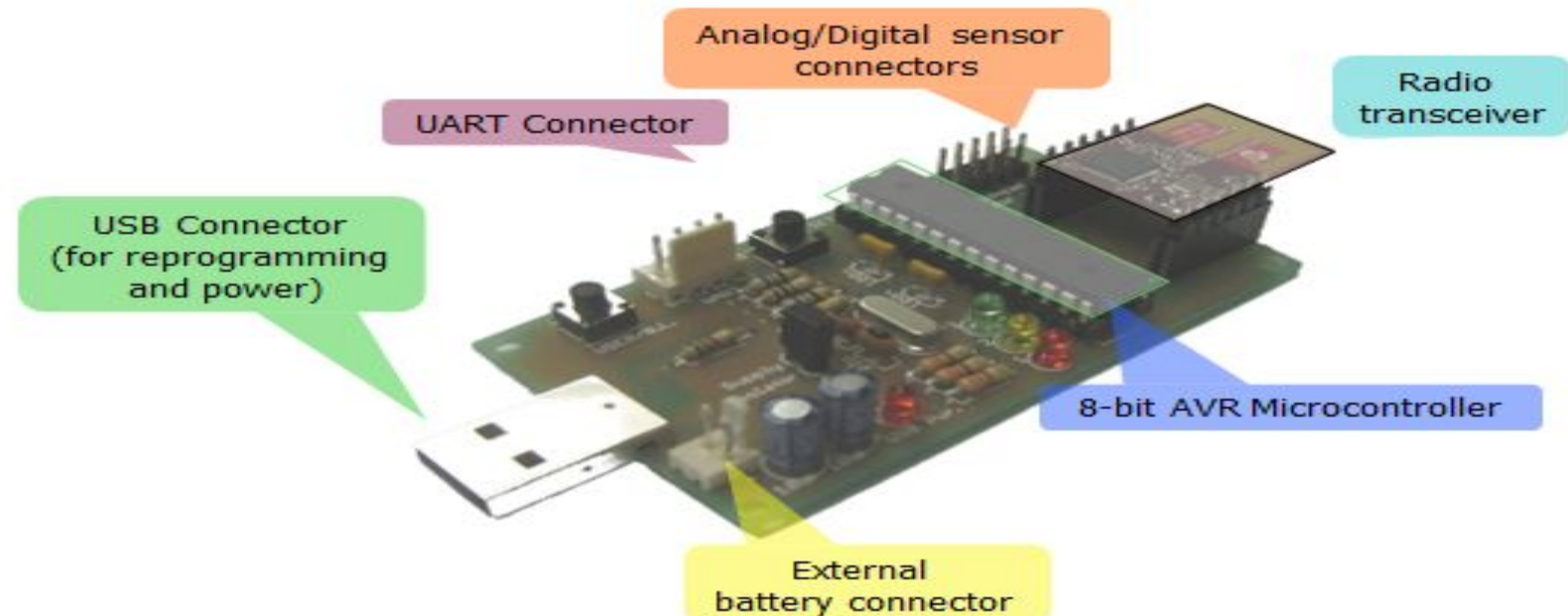


IMote2

- By Intel Research
- MCU: PXA271 XScale
- Comm: Chipcon CC2420 (IEEE802.15.4)



IWING-MRF Motes



Morakot Saravane, Chaiporn Jaikaeo, 2010. Intelligent Wireless Network Group (IWING), KU

IWING-MRF Motes

- Built from off-the-shelf components
- Built-in USB boot loader
 - Reprogrammed via USB
- Easy to modify and extend hardware



IWING-MRF Mote

- **Processor**
 - 8-bit AVR microcontroller ATmega88/168/328, 12 MHz
 - 16KB flash, 2KB RAM
- **RF transceiver**
 - Microchip's MRF24J40A/B/C, 2.4GHz IEEE 802.15.4
 - SPI interface
- **External connectors**
 - 6 ADC connectors (can also be used as TWI)
 - 1 UART
- **Power options**
 - 3 – 3.6 VDC
 - USB or 2 AA batteries

IWING-JN Motes

- Built on JN5168 wireless microcontroller
- 32-bit RISC architecture
 - Operating at 32 MHz
 - 256 KB flash, 32 KB RAM
- IEEE 802.15.4 RF transceiver
- 4 ADC channels (10-bit)
- ~20 general-purpose digital I/O
- 2 UART interfaces
- Hardware access via C-language API



Outline

- Main components of a wireless sensor node
 - Processor, radio, sensors, batteries
- Energy supply and consumption
- *Operating systems and execution environments*
 - IWING's MoteLib
 - TinyOS
 - Contiki
- Example implementations

Operating System Challenges

- Usual operating system goals
 - Make access to device resources abstract (virtualization)
 - Protect resources from concurrent access
- Usual means
 - Protected operation modes of the CPU
 - Process with separate address spaces
- These are not available in microcontrollers
 - No separate protection modes, no MMU
 - Would make devices more expensive, more power-hungry

Possible OS Options

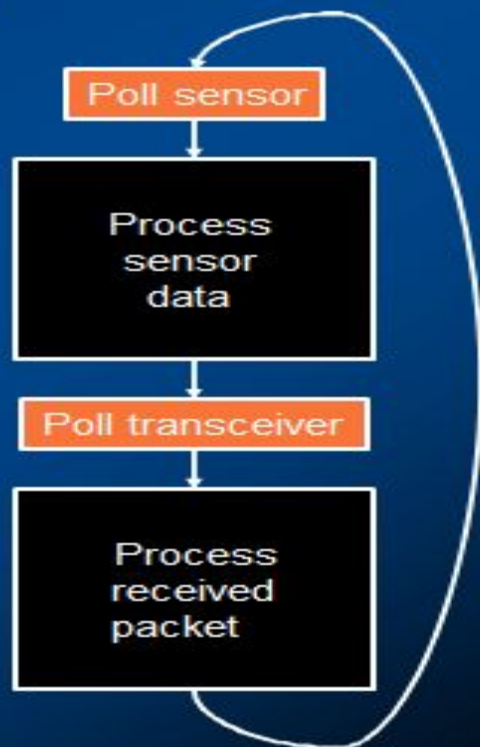
- Try to implement “as close to an operating system” on WSN nodes
 - Support for processes!
 - Possible, but relatively high overhead
- Stay away with operating system
 - There is only a single “application” running on a WSN node
 - No need to protect malicious software parts from each other
 - Direct hardware control by application might improve efficiency

Possible OS Options

- **Currently popular approach**
 - ⇒ **No OS, just a simple run-time environment**
 - **Enough to abstract away hardware access details**
 - **Biggest impact: Unusual programming model**

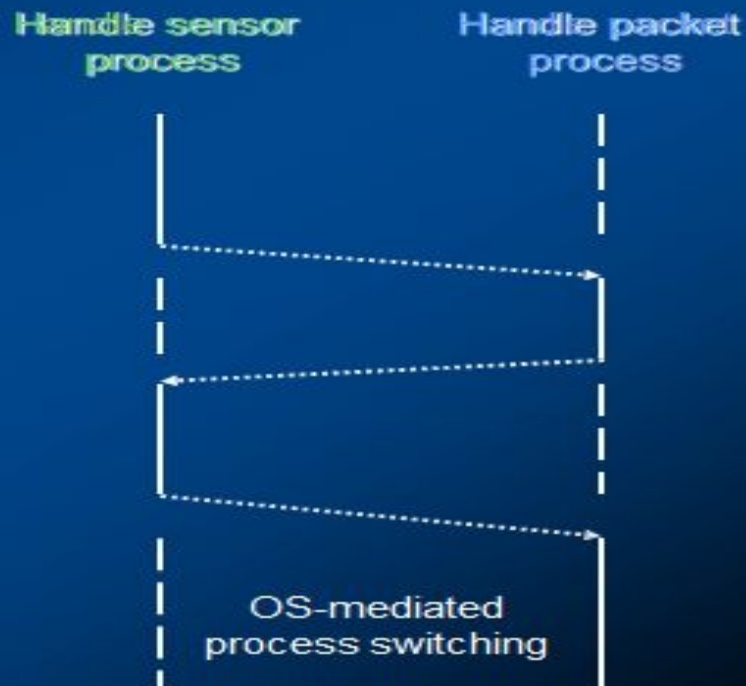
Concurrency Support

- **Simplest option: No concurrency, sequential processing of tasks**
 - **Risk of missing data**
 - **Should support interrupts/asynchronous operations**



Processes/Threads

- Based on interrupts, context switching
- Difficulties
 - Too many context switches
 - Most tasks are short anyway
 - Each process required its own stack



Event-Based Concurrency

- ***Event-based programming model***
 - Perform regular processing or be idle
 - React to events when they happen immediately
 - Basically: interrupt handler
- **Must not remain in interrupt handler too long**
 - Danger of losing events



Components Instead of Processes

- An abstraction to group functionality
- Typically fulfill only a single, well-defined function
 - E.g., individual functions of a networking protocol
- Main difference to processes:
 - Component does not have an execution
 - Components access same address space, no protection against each other

Event-based Protocol Stack

- Usual networking API: **sockets**
 - Issue: blocking calls to receive data
 - Not match to event-based OS
- API is therefore also event-based
 - E.g., Tell some component that some other component wants to be informed if and when data has arrived
 - Component will be posted an event once this condition is met
 - Details: see **IWING's Motelib** and **TimyOS**

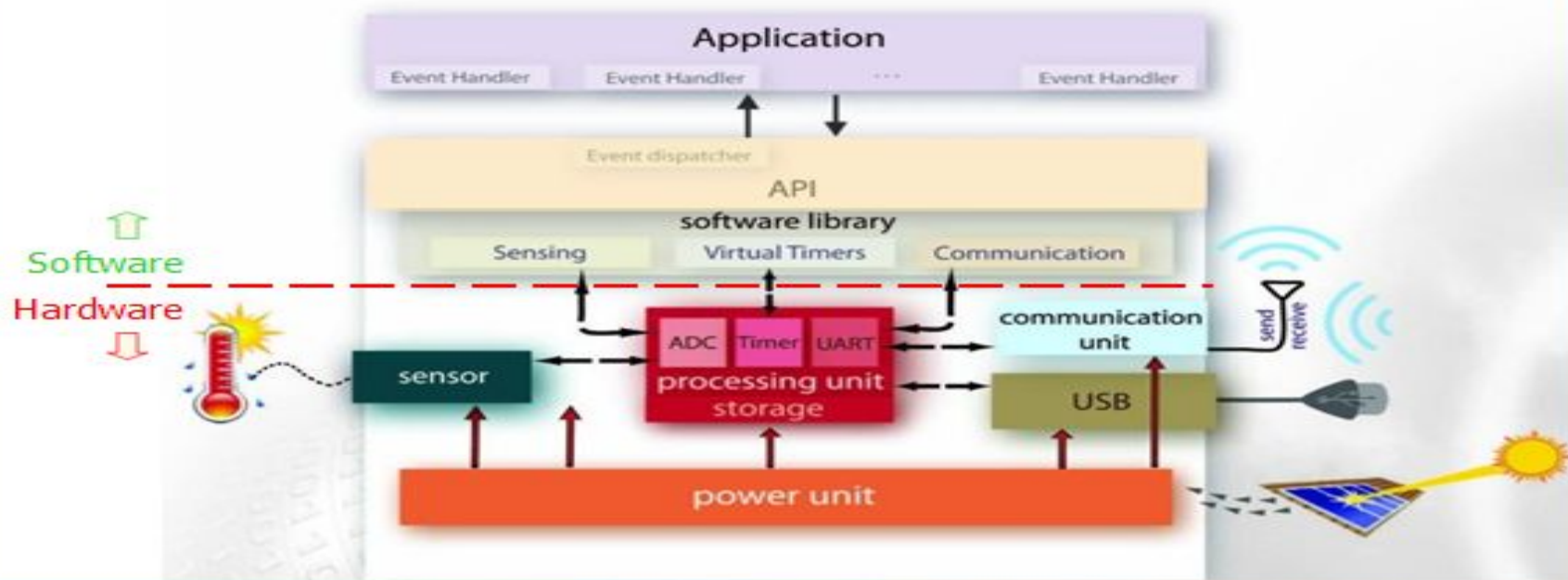
Outline

- **Main components of a wireless sensor node**
 - Processor, radio, sensors, batteries
- **Energy supply and consumption**
- **Operating systems and execution environments**
 - *IWING's MoteLib*
 - TinyOS
 - Contiki
- **Example implementations**

Case Study: IWING's MoteLib

- **Developed by IWING (CPE, KU) along with IWING motes**
- **Provides hardware abstraction and virtualization in standard C interfaces**
- **Follows event-based programming model**

MoteLib Architecture and API



Example: *Count and Send*

- **Node#0 runs a counter and broadcasts its value**
- **Other nodes display received values on LEDs**

Example: Count and Send

```
#include <motelib/system.h>
#include <motelib/timer.h>
#include <motelib/radio.h>
#include <motelib/led.h>

Timer timer;
uint8_t counter;

void timerFired(Timer *t)
{
    counter++;
    radioRequestTx(BROADCAST_ADDR, 0, (char*)&counter, sizeof(counter), NULL);
}

void receive(Address source, MessageType type, void *message, uint8_t len)
{
    ledSetValue(((char*)message)[0]);
}

void boot()
{
    counter = 0;
    if (getAddress() == 0)
    {
        timerCreate(&timer);
        timerStart(&timer, TIMER_PERIODIC, 500, timerFired);
    }
    else
        radioSetRxHandler(receive);
}
```

called when timer expires

called when node receives a radio packet

called when node booted

Outline

- Main components of a wireless sensor node
 - Processor, radio, sensors, batteries
- Energy supply and consumption
- Operating systems and execution environments
 - IWING's MoteLib
 - *TinyOS*
 - Contiki
- Example implementations

Case Study: TinyOS



- Developed by UC Berkeley as runtime environment for their motes
- nesC (network embedded system C) as adjunct programming language
- Design aspects:
 - Component-based system
 - Components interact by exchanging asynchronous events
 - Components form a program by *wiring* them together (akin to VHDL – hardware description language)
- Website: <http://www.tinyos.net>

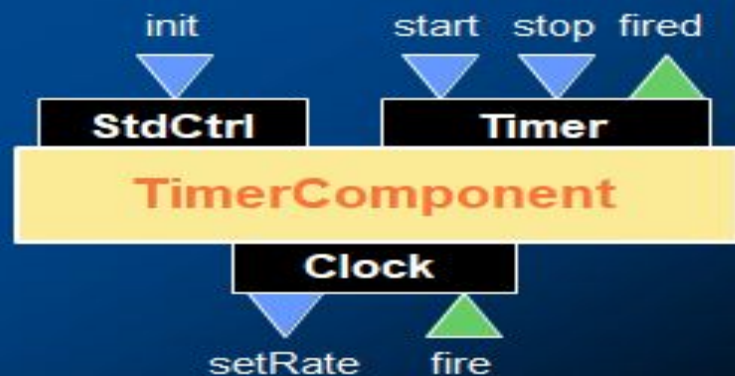
TinyOS Components

- **Components**
 - **Frame** – state information
 - **Tasks** – normal execution program
 - **Command handlers**
 - **Event handlers**
- **Hierarchically arranged**
 - Events are passed upward from hardware
 - Commands are passed downward

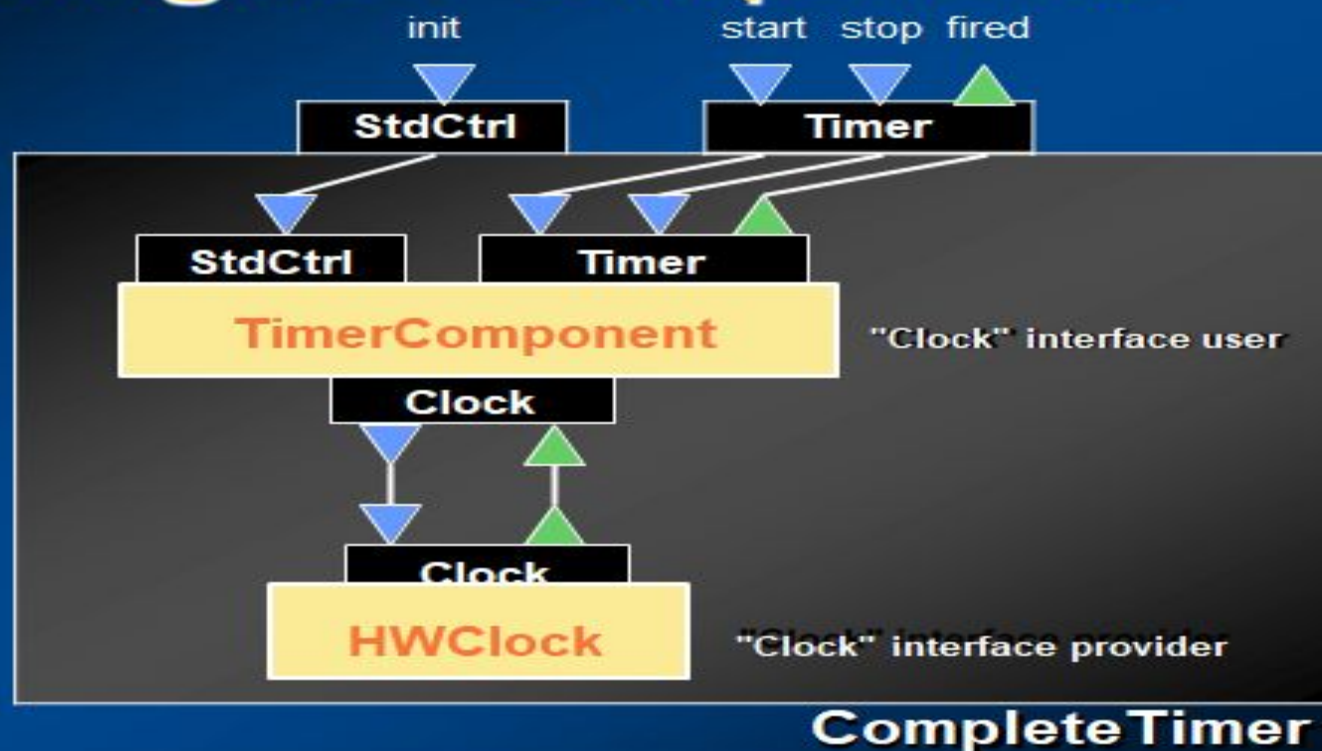


Interfaces

- Many commands/events can be grouped
- nesC structures corresponding commands/events into *interface types*
- Example: Structure timer into three interfaces
 - StdCtrl
 - Timer
 - Clock
- The TimerComponent
 - Provides: StdCtrl, Timer
 - Uses: Clock



Forming New Components

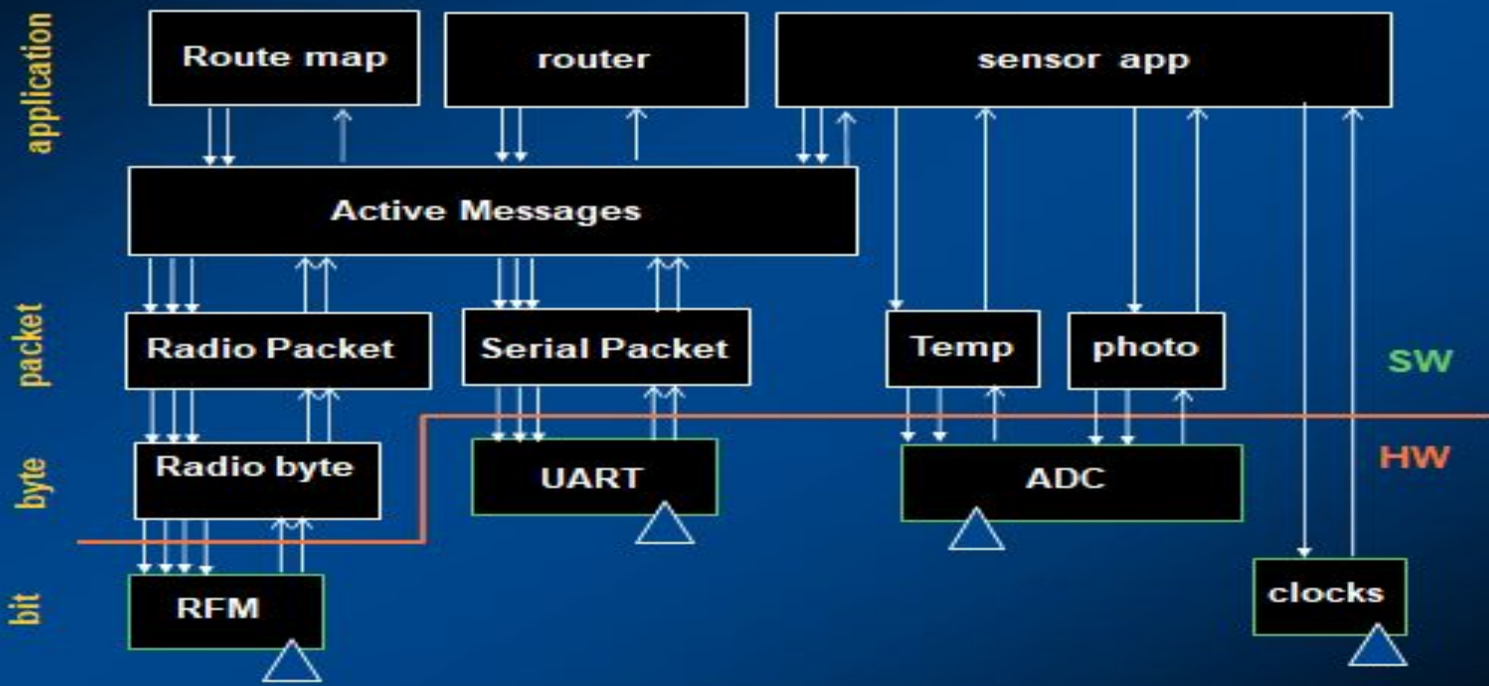


Sample nesC Code

```
configuration CompleteTimer
{
  provides
  {
    interface StdCtrl;
    interface Timer;
  }

  implementation
  {
    components TimerComponent, HWClock;
    StdCtrl = TimerComponent.StdCtrl;
    Timer = TimerComponent.Timer;
    TimerComponent.Clock -> HWClock.Clock;
  }
}
```

Sample App Configuration

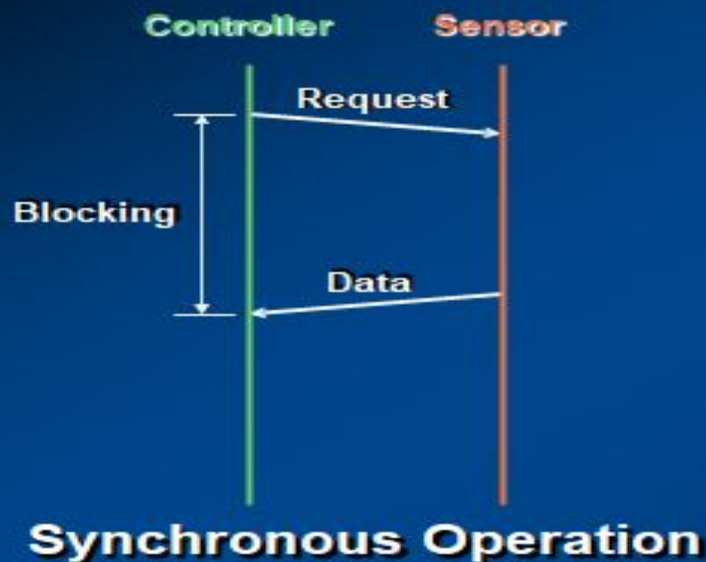


Handlers versus Tasks



- **Command/event handlers must run to completion**
 - Must not wait an indeterminate amount of time
 - Only a *request* to perform some action
- **Tasks can perform arbitrary, long computation**
 - Can be interrupted by handlers
 - Also have to be run to completion
 - Preemptive multitasking not implemented

Split-Phase Operations



Outline

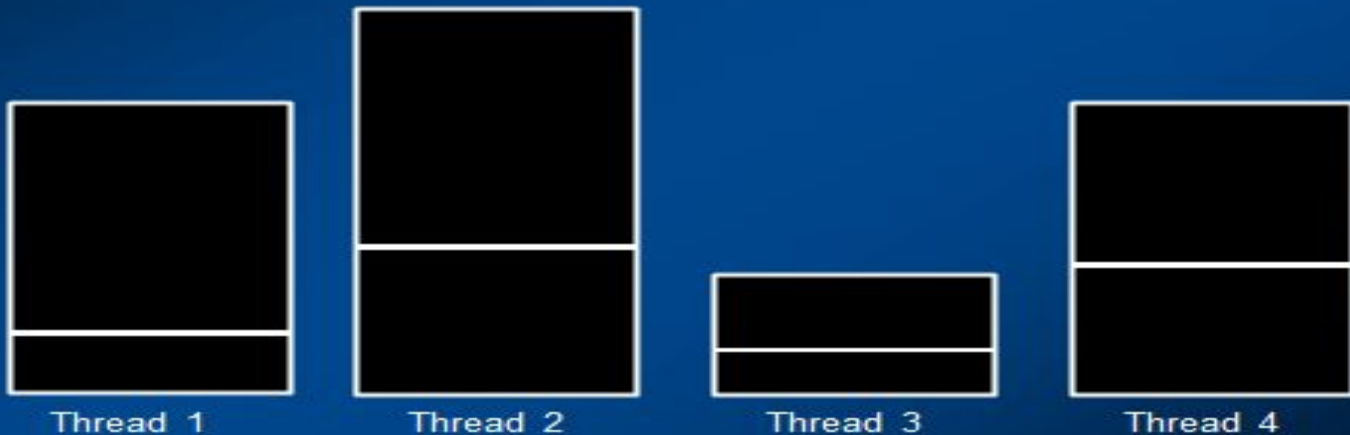
- **Main components of a wireless sensor node**
 - Processor, radio, sensors, batteries
- **Energy supply and consumption**
- **Operating systems and execution environments**
 - IWING's MoteLib
 - TinyOS
 - *Contiki*
- **Example implementations**

Case Study: Contiki

- Multitasking OS developed by **Swedish Institute of Computer Science (SICS)**
- The kernel is event driven
- Processes are **protothreads**
 - Very light weight threads
 - Provide a linear, thread-like programming model
- Comes with various communication stacks: uIP, uIPv6, Rime
- Website <http://www.contiki-os.org/>

Problem with Multithreading

- Four threads, each with its own stack



Events Require One Stack

- Four event handlers, one stack

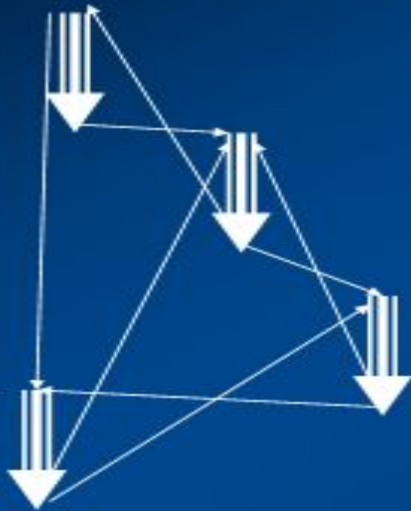
Stack is reused for every event handler



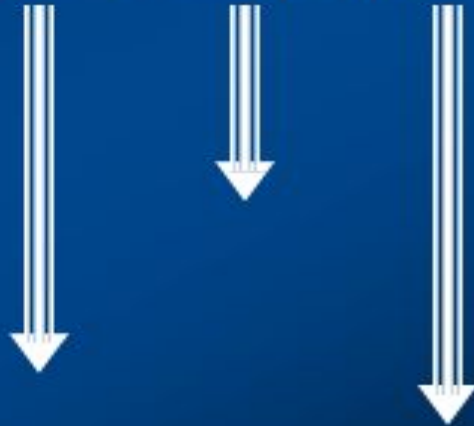
Eventhandler 34

Problem with Event-based Model

Events: unstructured code flow



Threads: sequential code flow



Very much like programming with GOTOs

Protothreads


- Protothreads require only one stack
- E.g, four protothreads, each with its own stack



Contiki Processes

- Contiki processes are protothreads

```
PROCESS_THREAD(hello_world_process, ev, data)
{
    PROCESS_BEGIN();
    printf("Hello, world!\n");
    while(1) {
        PROCESS_WAIT_EVENT();
    }
    PROCESS_END();
}
```



Contiki's Cooja Simulator

The screenshot displays the Cooja simulator interface with the following components:

- Network View:** A graph showing 32 nodes (represented by numbered circles) connected by blue lines, illustrating a network topology.
- Simulation Control:** A panel with buttons for Start, Pause, Step, and Reload. It displays the current Time: 00:07.737 and Speed: 40.76%.
- Mote Output:** A log window showing the following data:

Time ms	Mote	Message
1270	ID:32	MAC 00:12:74:20...
1275	ID:23	IPv6 addresses:...
1279	ID:32	CSMA ContikiMAC...
1282	ID:23	fe80::212:7417:...
1295	ID:32	Tentative link...
1298	ID:32	Starting 'Unica...
1300	ID:23	IPv6 addresses:...

- Timeline:** A window titled "Timeline showing 41 motes" displaying a sequence of events as colored bars (green, red, black) over time.

Summary

- **The need to build cheap, low-energy, (small) devices has various consequences**
 - **Much simpler radio frontends and controllers**
 - **Energy supply and scavenging are a premium resource**
 - **Power management is crucial**
- **Unique programming challenges of embedded systems**
 - **Concurrency without support, protection**
 - **De facto standard:**
 - **Event-based programming model: TinyOS**
 - **Multithreaded programming model: Contiki**



Thank You!

WIRELESS SENSOR NETWORKS

Dr.P.Venkatesan

Associate professor/ECE

SCSVMV University



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

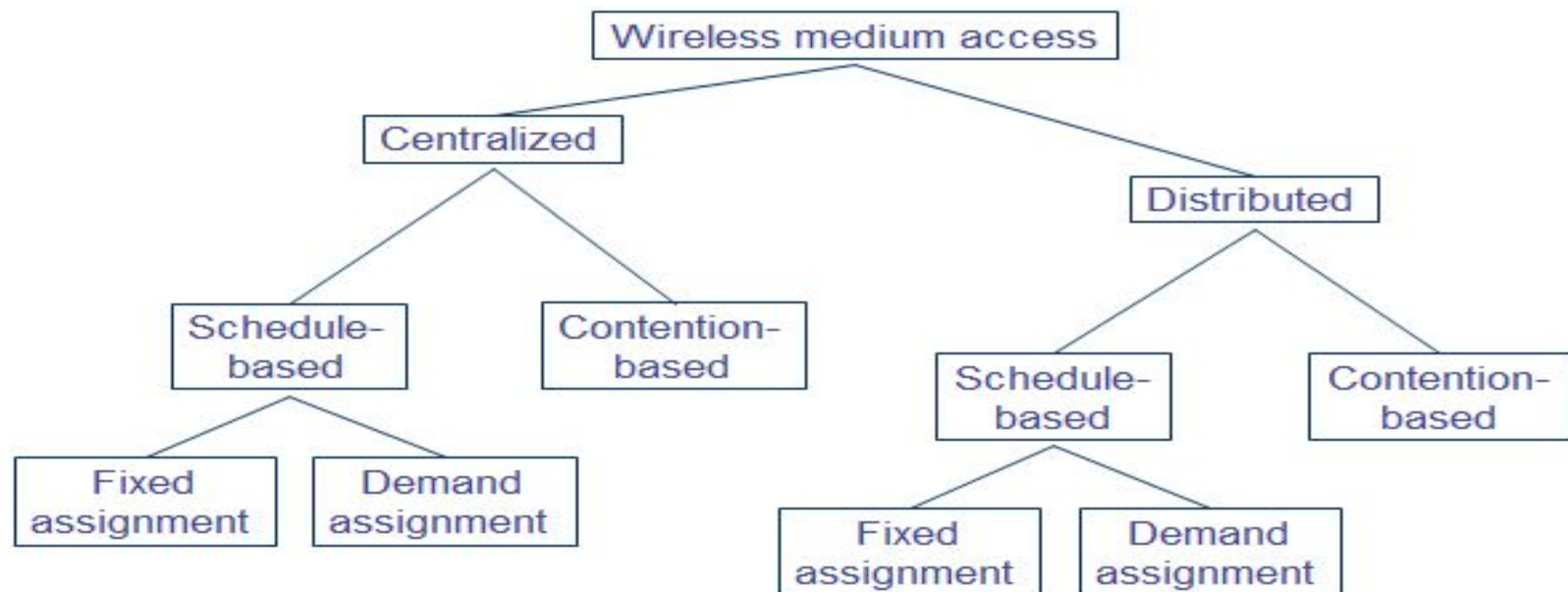
- Medium access in wireless networks is difficult mainly because of
 - Impossible (or very difficult) to send and receive at the same time
 - Interference situation at receiver is what counts for transmission success, but can be very different from what sender can observe
 - High error rates (for signaling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

- 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

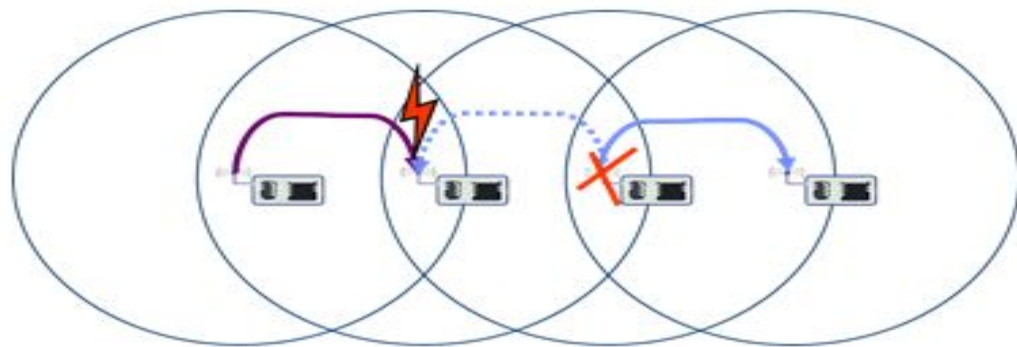
Overview

- Principal options and difficulties
- Contention-based protocols
 - MACA
 - S-MAC, T-MAC
 - Preamble sampling, B-MAC
 - FAMAS
- 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

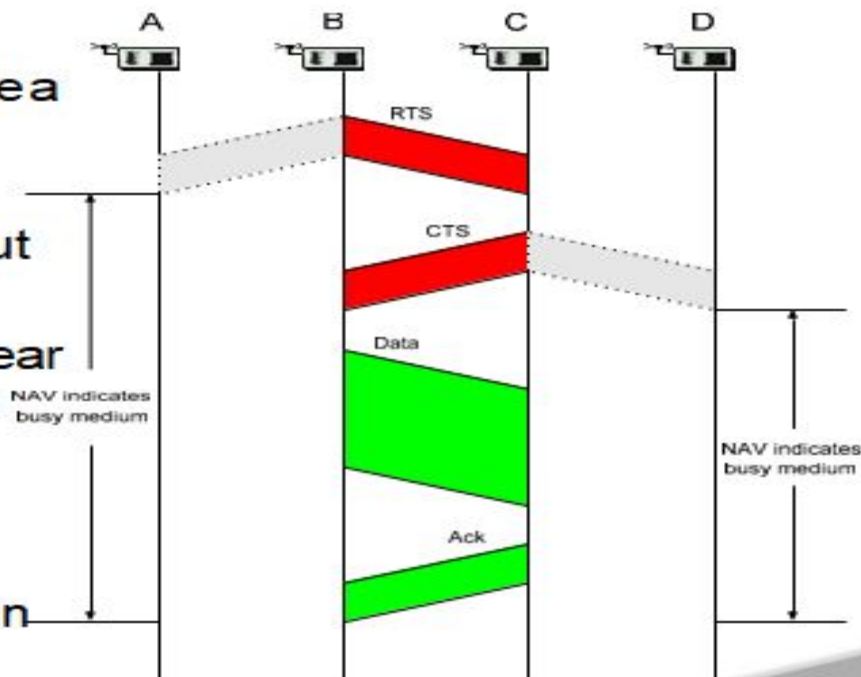


Main options to shut up senders

- 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

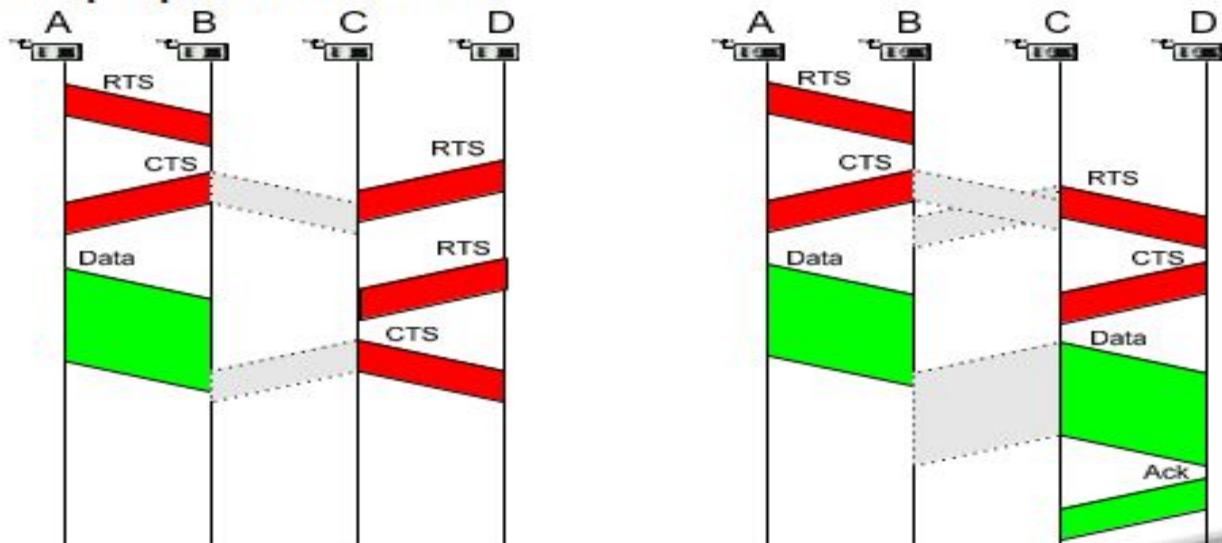
Receiver informs interferers before transmission – MACA

- 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

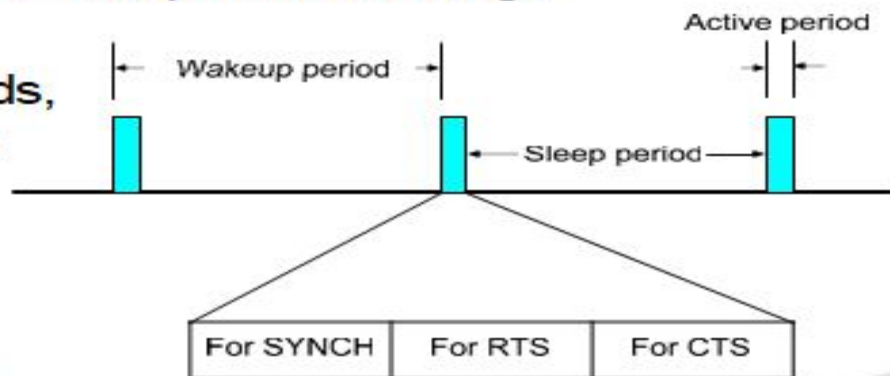
- RTS/CTS ameliorate, but do not solve hidden/exposed terminal problems
- Example problem cases:



- 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)

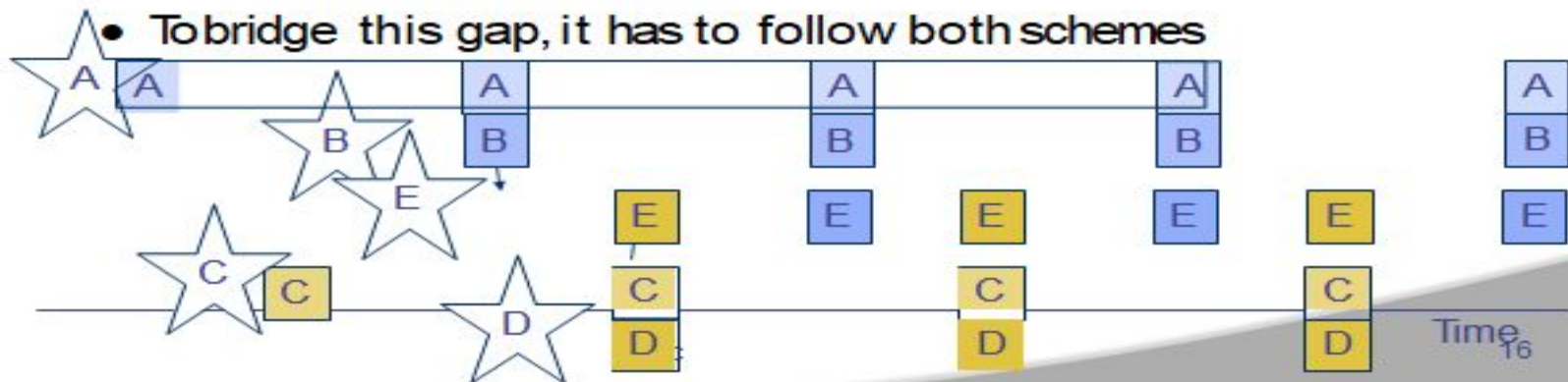
- MACA's idle listening is particularly unsuitable if averagedata rate is low
- Most of the time, nothing happens
- Idea: Switch nodes off, ensure that neighboring nodes turn on simultaneously to allow packet exchange (rendez-vous)
- Only in these active periods, packet exchanges happen
- Need to also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



S-MAC synchronized islands

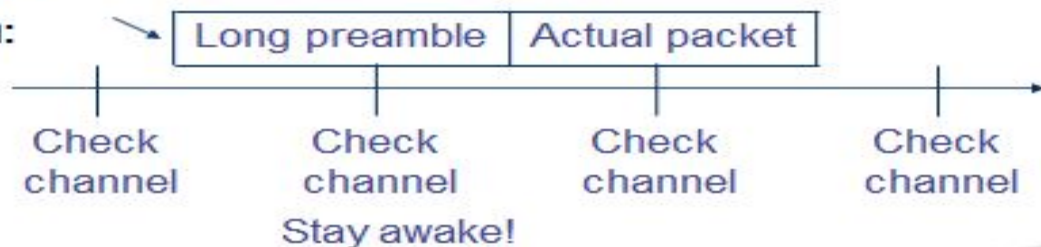
- Nodes try to pick up schedule synchronization from neighboring nodes
- If no neighbor found, nodes pick some schedule to start with
- If additional nodes join, some node might learn about two different schedules from different nodes
 - “Synchronized islands”

- To bridge this gap, it has to follow both schemes



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

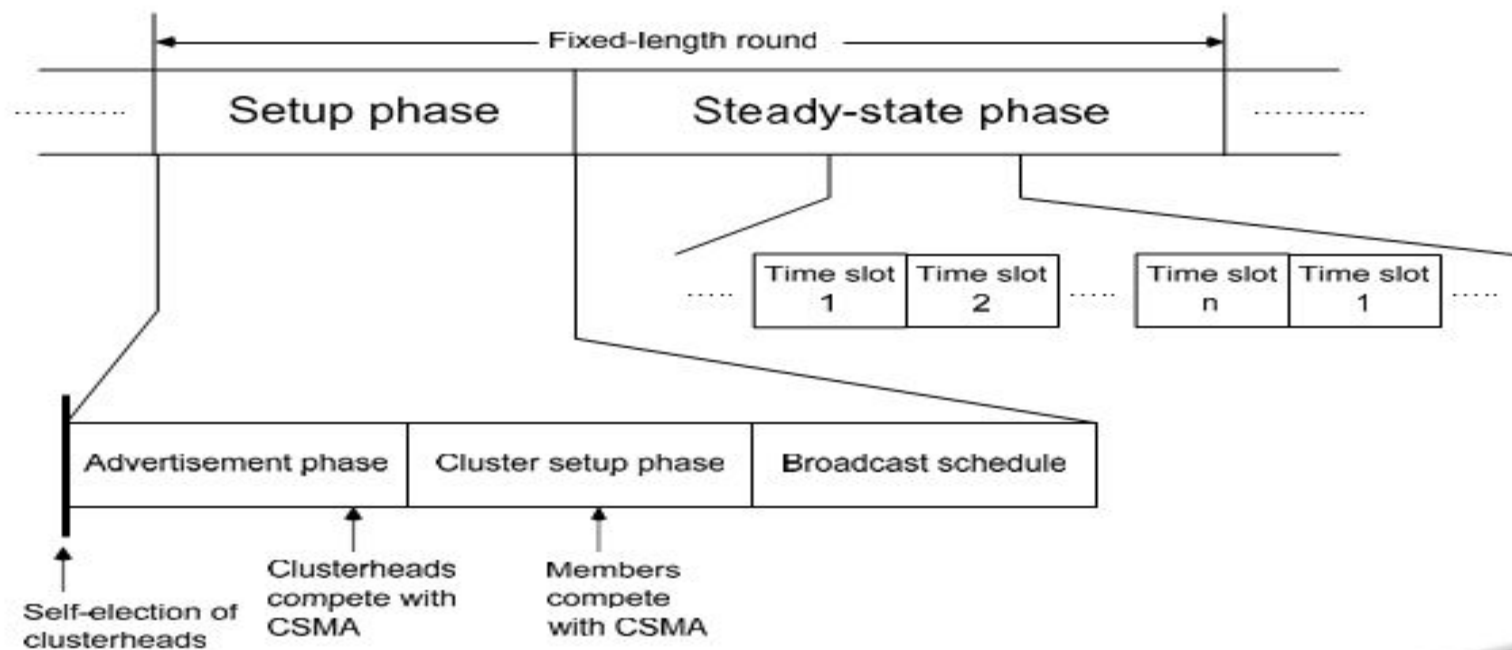
Overview

- 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

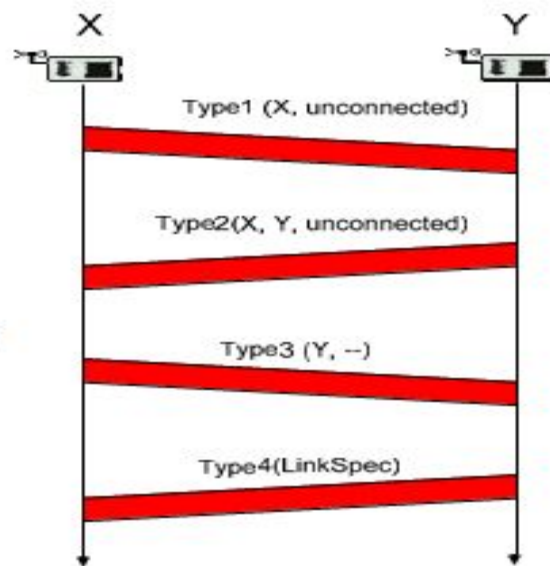


SMACS

- 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

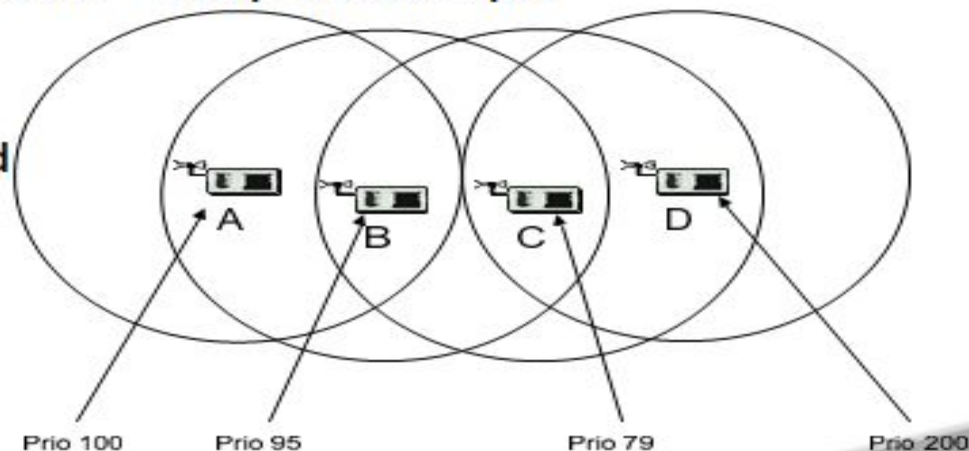
TRAMA

- 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 $p = h(x \odot t)$
 - 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

- 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

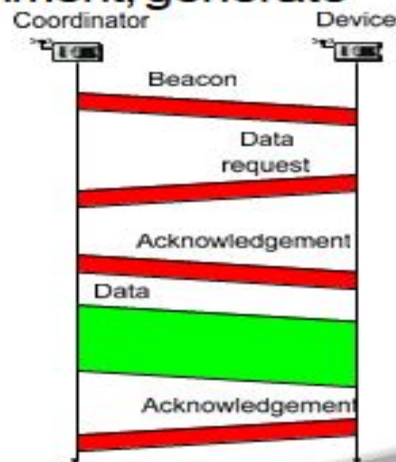
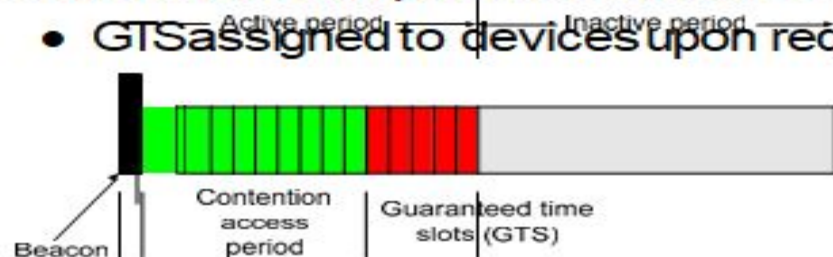
- Principal options and difficulties
- Contention-based protocols
- Schedule-based protocols
- IEEE 802.15.4

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

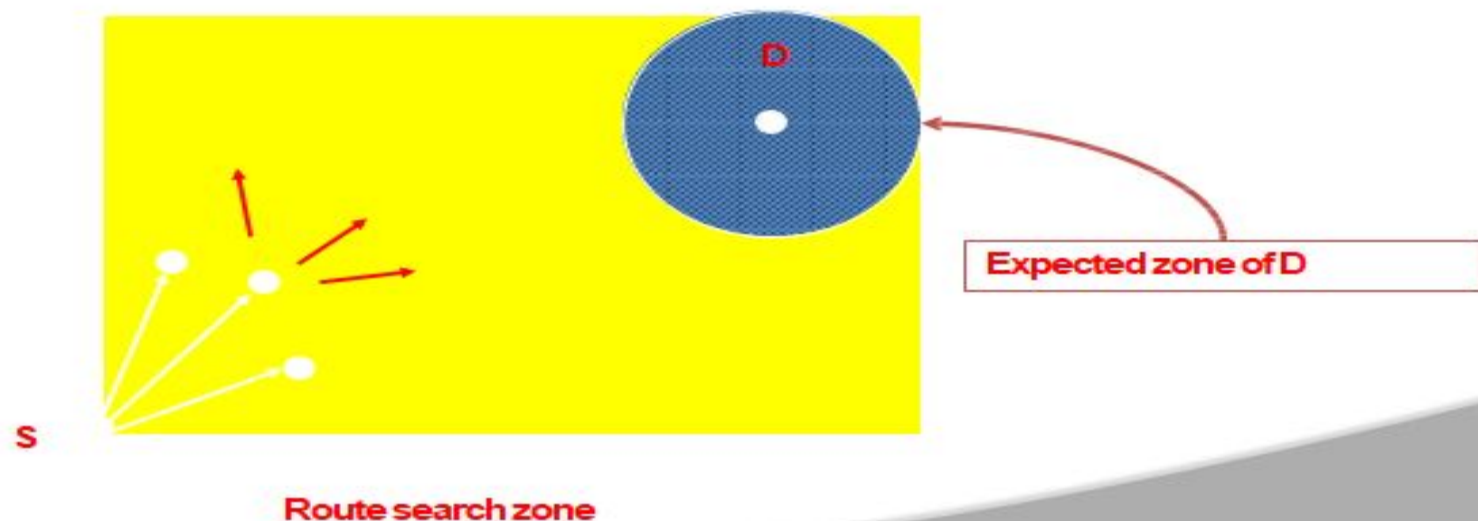
- Location-Aided Routing (LAR) in mobile ad hoc networks
- Young-Bae Ko and Nitin H. Vaidya
- Texas A&M University
- Wireless Networks 6 (2000) 307–321

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

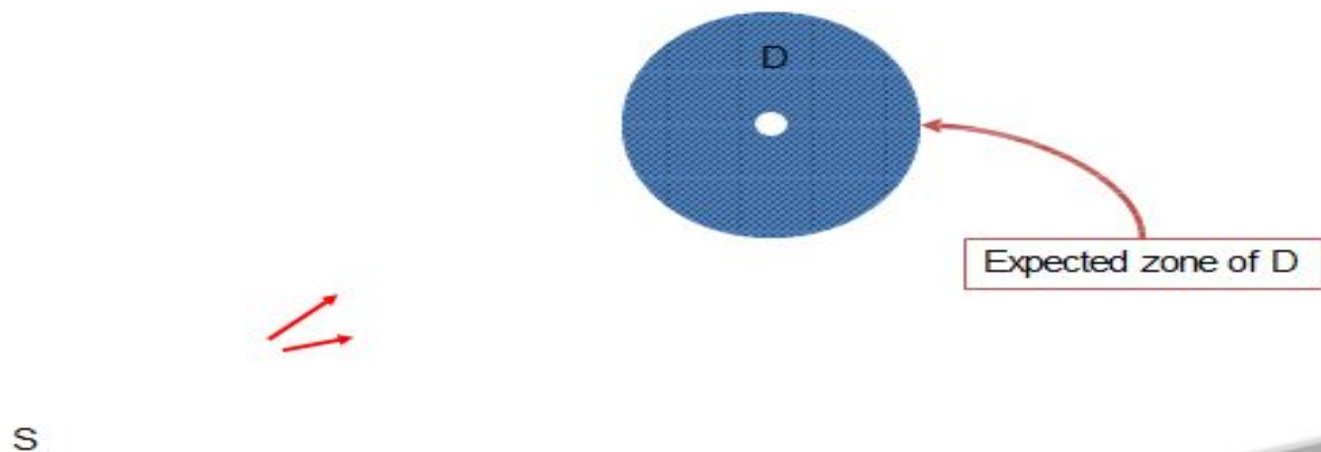
- **A**Distance **R**outing **E**ffect **A**lgorithm for **M**obility (DREAM)
- S. Basagni, I. Chlamtac, V.R. Syrotiuk, B.A. Woodward
- The University of Texas at Dallas
- Mobicom'98

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.**

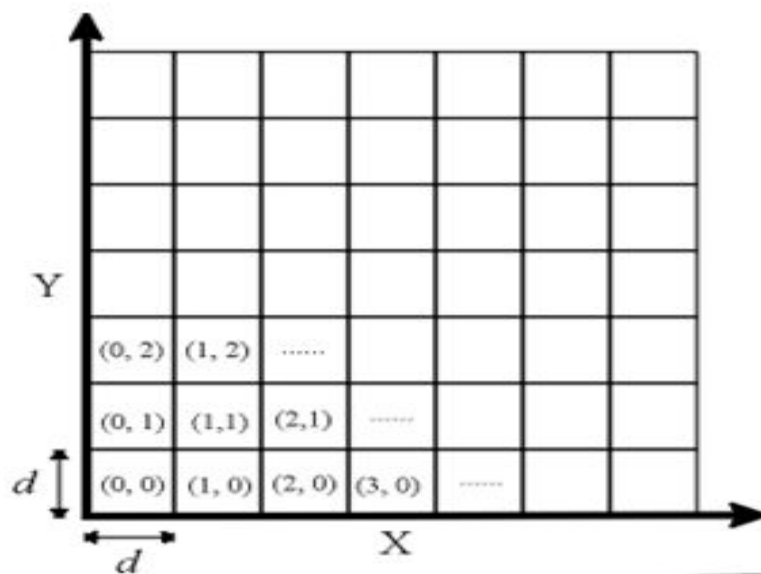


GRID Routing

- **“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

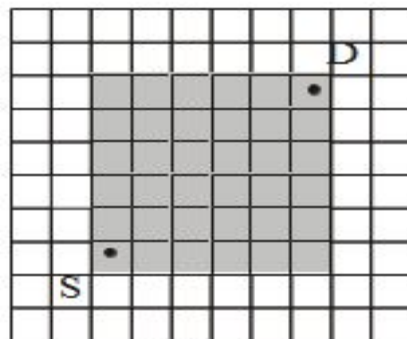
- Partition the physical area into $d \times d$ squares called *grids*.



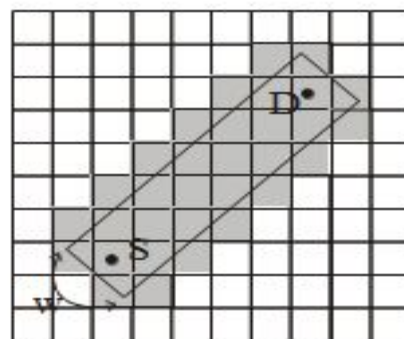
Protocol Overview

- ❑ 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.**

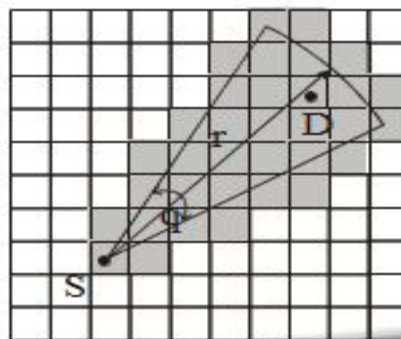
Route Search Range Options



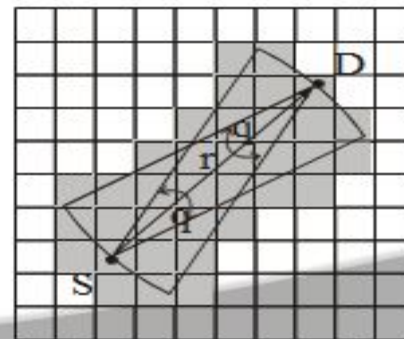
(a) Rectangle



(b) Bar(w)

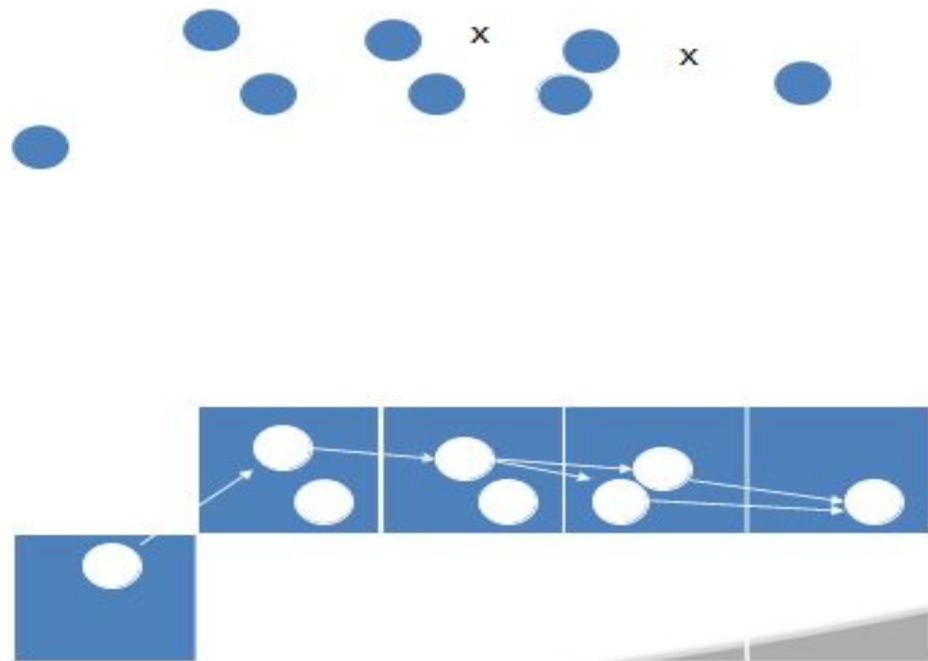


(c) Fan(q, r)



(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



Thank You!

WIRELESS SENSOR NETWORKS

Dr.P.Venkatesan

Associate professor/ECE

SCSVMV University



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 scarce 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.

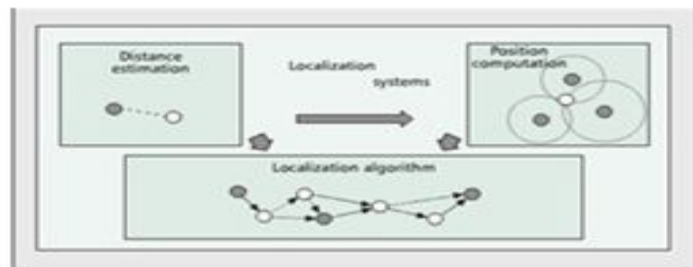
TIME SYNCHRONIZATION

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.

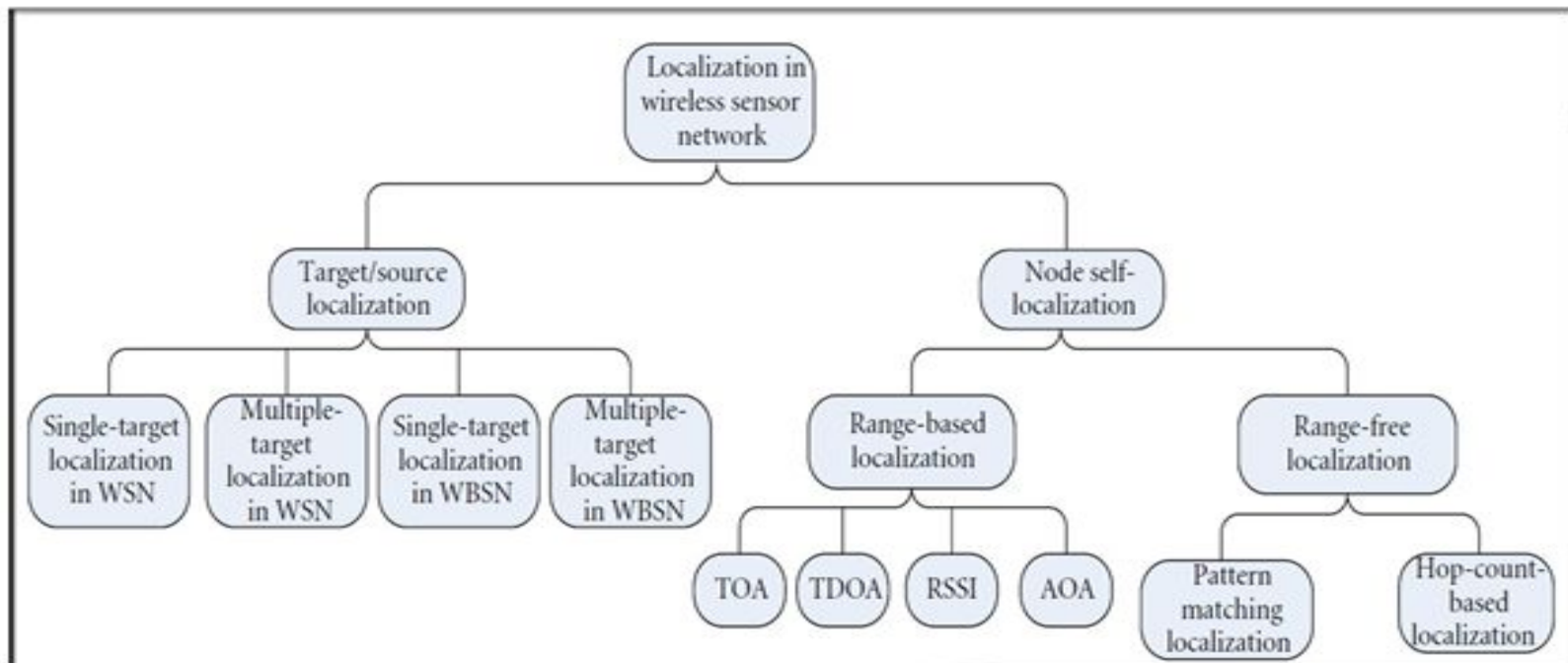
GPS: We need to determine the physical coordinates of a group of sensor nodes in a wireless sensor network (WSN).

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



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 :

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

where g_i represents the gain factor of the i th 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 i th sensor and the source. In addition n_i is the measurement noise modeled as zero mean white Gaussian with variance σ_i^2 , namely, $n_i \sim N(0, \sigma_i^2)$.

Target/Source Localization

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

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

where $d_{ik}(t)$ is the distance between the i th sensor and the k th source. K is the number of the sources. g_i is the gain of i th 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 k th source. α is the attenuation exponent.

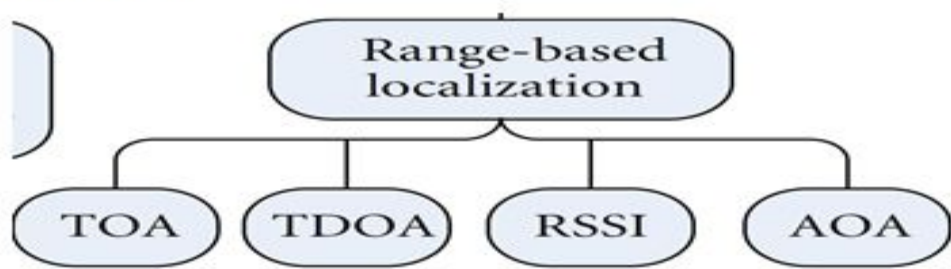
Target/Source Localization

- 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

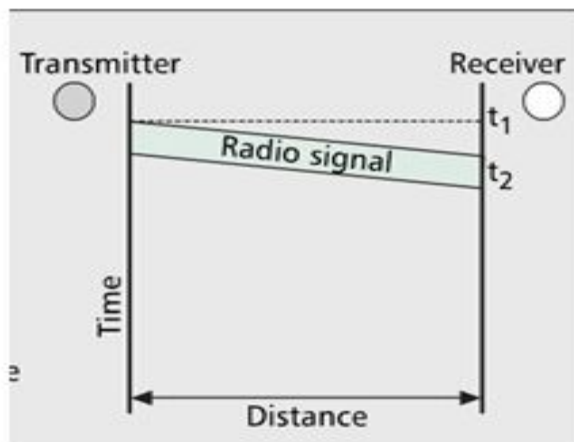
Range-based Localization: uses the measured distance/angle to estimate the indoor location using geometric principles.

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

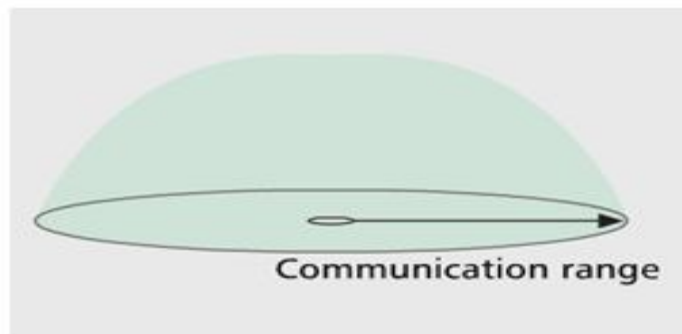
- Time of arrival: (TOA): It's a method that tries to estimate distance between 2 nodes using time based measures.
- Accurate but needs synchronization



RANGE BASED LOCALIZATION

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

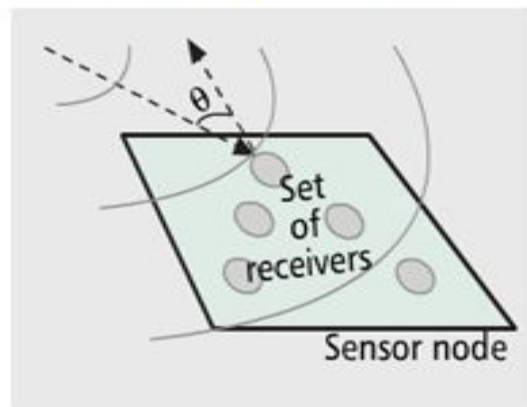
➤ Low cost but very sensitive to noise



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.

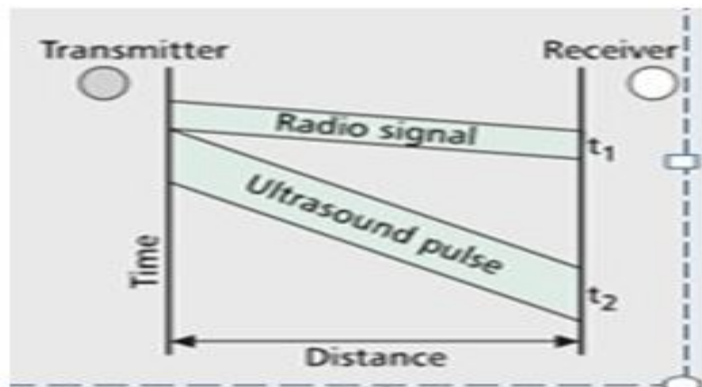


RANGE BASED LOCALIZATION

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.



RANGING TECHNIQUES

- Ranging methods aim at estimating the distance of a receiver to a transmitter. The first way is to use RSS (received signal strength) along with the 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 the RSS method.

RSS method con's:

- The distance to estimate is 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.

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.

Distance calculation:

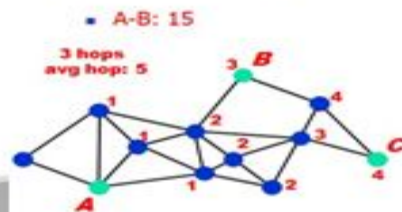
➤ According to the position of the beacon node and hop count, each beacon node uses the following equation to estimate the actual distance of every hop

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

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

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



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 be 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 utility and 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:

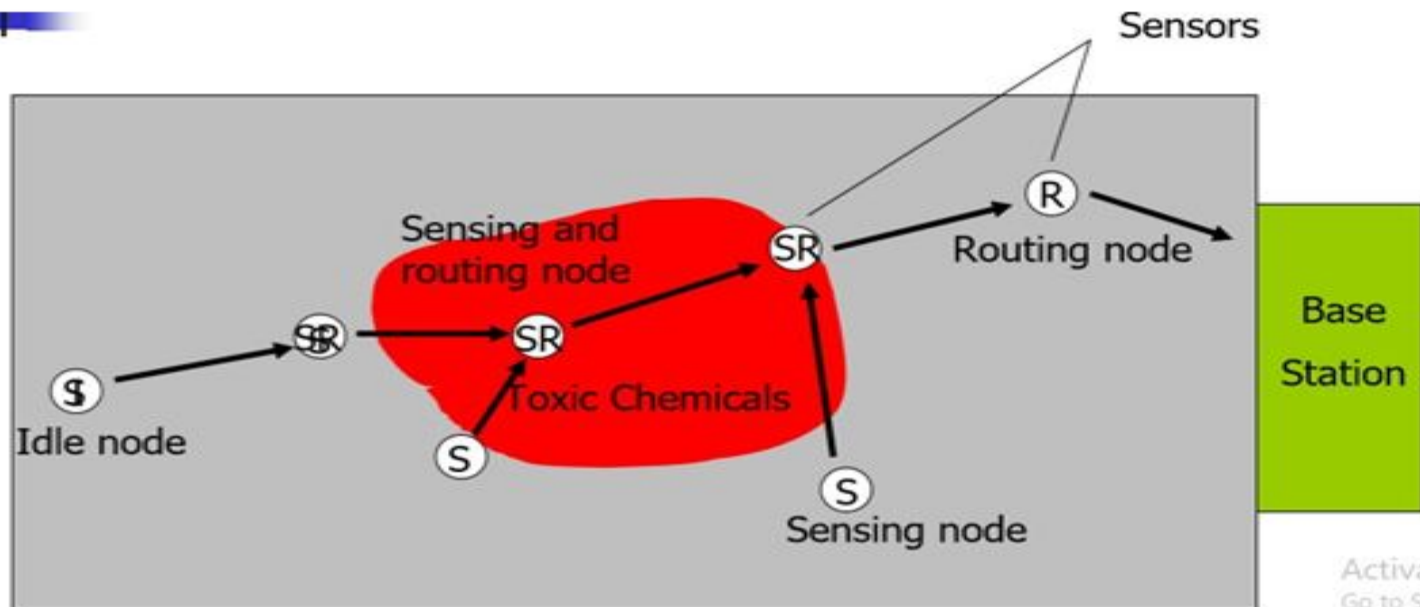
➤ „Nodes, denoted by SR, may participate in both sensing and routing.

➤ „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



$$T_0 = T_0 + \Delta T$$

Activate
Go to Setting

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

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

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

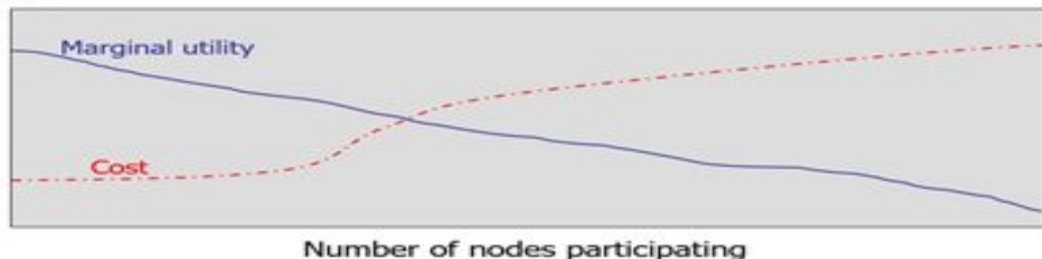
GENERIC MODEL OF UTILITY AND COST

„The constraint is defined as

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

receiving nodes as $V_r(t)$.

➤ 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:

„ C_s : the cost of a sensing operation

„ C_a : the cost of data aggregation

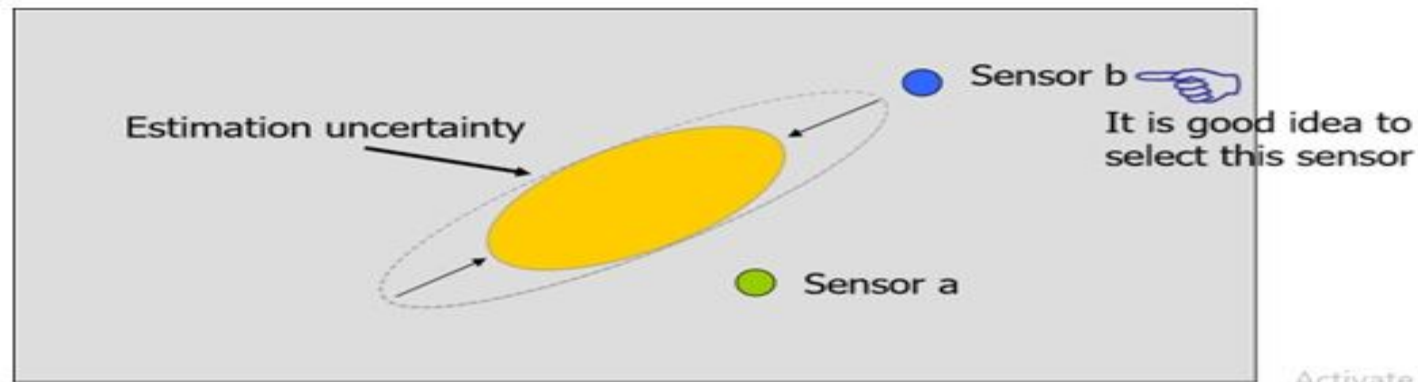
„ C_t : the cost of data transmission „

C_r : 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



Activate V
Go to Editor

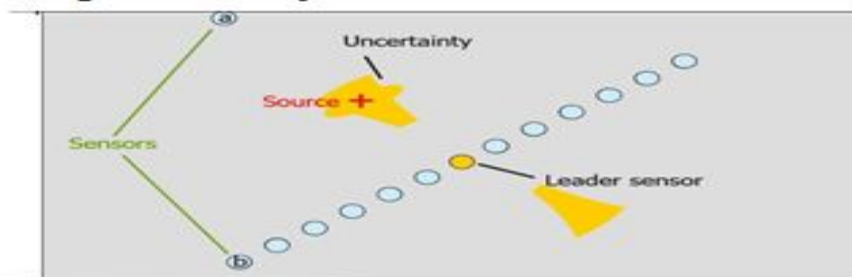
Sensor selection based on information gain of individual sensor contributions

SENSOR SELECTION

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

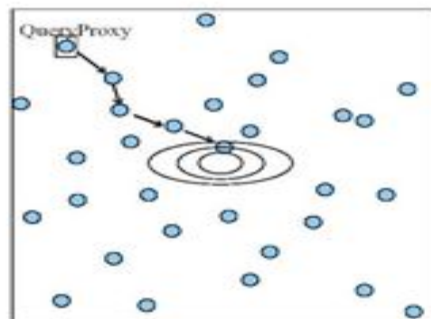
Sensor b would provide better information than a because sensor b lies close to the longer axis of the uncertainty ellipsoid and its range constraint would intersect this longer axis transversely.

Scenario: Localizing a Stationary Source



Joint Routing and Information Aggregation

Our primary purpose is to collect and aggregate information. „DSQ 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 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.

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

Joint Routing and Information Aggregation

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

➤ ζ_j is the position of the node j .

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

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

➤ „ ζ_k is the position of the current routing node.

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



Thank You!

WIRELESS SENSOR NETWORKS

Dr.P.Venkatesan

Associate professor/ECE

SCSVMV University

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 systems support.
- Typically support at least one programming language, such as C.
- Ex: mica, TinyOS, nesC

- 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.

Berkeley motes







Mote type		WeC	Rene	Rene2	Mica	Mica2	Mica2Dot
Example picture							
MCU	Chip Type	AT90LS8535	ATmega163L	ATmega103L	ATmega128L		
	Program memory (KB)	4 MHz, 8 bit	4 MHz, 8 bit	4 MHz, 8 bit	8 MHz, 8 bit		
	RAM (KB)	8	16	128	128		
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.

TinyOS

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

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()` \in `eventsendDone()`
- Avoid blocking the entire system
- Not accepting another packet Until `sendDone()` is called, avoid race condition

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.

nesC

- An event call is a method call from a lower layer component to a higher layer component. (signal)
- Acommand 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.

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
 - $A.a \rightarrow B.a$, interface is hidden from upper layers

Node-level software platforms

nesC

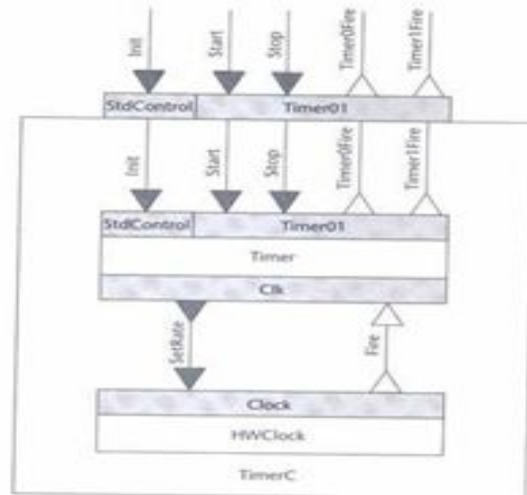


Figure 7.9 The TimerC configuration implemented by connecting Timer with HWClock.

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.

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.

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.timerGFire() {
    bool localBusy;
    atomic {
      localBusy = busy;
      busy = TRUE;
    }
    if (!localBusy) {
      call ADC.getData(); //start getting sensor reading
      return SUCCESS;
    } else {
      return FAILED;
    }
  }
}
```

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

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.

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.

TinyGALS

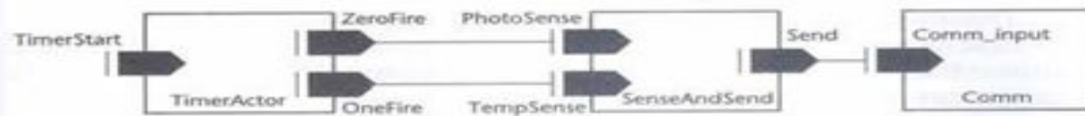


Figure 7.14 Triggering, sensing, and sending actors of the FieldMonitor in TinyGALS.

```
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

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

Node-Level Simulators

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

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.

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

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

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.

Example figure

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



- DE



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.

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 issues are discussed.

- 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?

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, GOGis useful.
- By broadcasting from one specific sensor, those have heard the packet will become the same group.

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

Publish/Subscribe group

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

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.



shutterstock.com · 1153070891