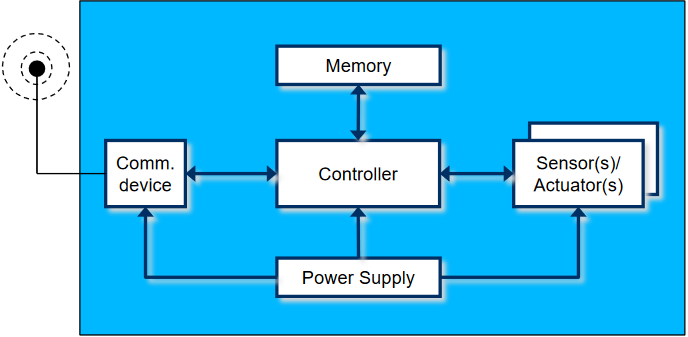
#### Q1: Describe the architecture and primary tasks of a WSN node including the main HW components characteristics and their energy consumption characteristics. Explain the typical methods to save energy according to the energy consumption characteristics of different hardware components. Give examples.

In the motes we have five major components



* Controller
  + Microcontroller

Suitable for building computationally less intensive, standalone applications, because of its compact construction, small size, low-power consumption, and low cost. It also features power different power states. Provides a high speed programming and eases debugging, because of higher-level programming languages.

Not as powerful and as flexible as some custom-made processors (such as DSPs and FPGAs).

* + Digital Signal Processors (DSPs)

Powerful, and complex digital filters can be realized with commonplace DSPs. useful for applications that require the deployment of nodes in harsh physical settings.

Lack of flexibility tasks require protocols (and not numerical operations) that need periodical upgrades or modifications (i.e., the networks should support flexibility in network reprogramming)

* + FPGA (Field Programmable Gate Array)

Flexible in their application, support parallel processing, can be reprogrammed.

Design and realization process is costly and reprogrammed take time

and energy.

* + ASIC (Application-specificIntegrated Circuit)

They ment to complement the Microcontroller or DPS for rudimentary and low-level tasks.

Simple design, can be optimized to meet a specific customer demand.

High development costs and lack of re-configurability.

* Memory
  + Ram is fast but losses content, when power is lost.
  + ROM can be read fast, but write slow.
  + Flash can be reprogrammed and erased.
* Communication Device
  + Radio Frequency (e.g. CC2420)

Long range and working in NLOS scenarios. Offer high data rates. Different states.

* + Another option: Optical, Need LOS, power energy per bit.
* Power Supply
  + Primary battery

This type is not rechargeable (Alkaline and Lithium)

* + Secondary battery

This type is rechargeable (Lithium), only makes sense in a setup with form of energy harvesting.

* Sensors / Actuators
  + Active sensors
    - Radar
  + Passive sensors
    - Thermometer, microphones
  + Actuators

Controls motors that can be used in many different usecases ex open a garden house window.

Energy Savings (More in Q2)

Controlling transceivers

* Similar to the controller, low duty cycle is necessary
* Easy to do for transmitter, difficult for receiver

Computation vs. communication energy cost

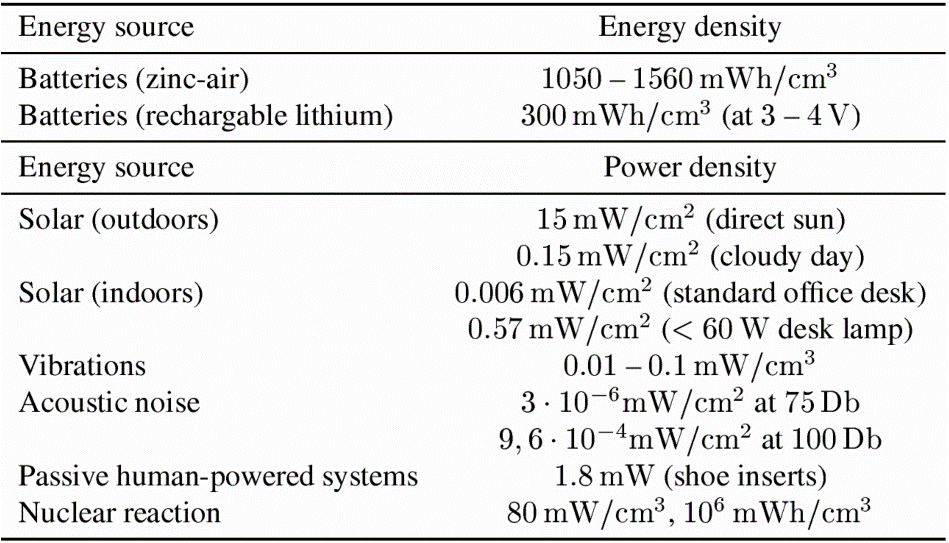
* Energy ratio of “sending one bit” vs. “computing one instruction” is between 220 and 2900 in the literature. To communicate (send & receive) one kilobyte = computing three million instructions!

SMAC, CSMA/CA, TDM etc.

#### Q2: Describe the possible energy sources for WSN nodes including energy harvesting. What are the pro and cons of the different sources? Describe the energy modeling of transmitter and receiver. What aspects should be taken into account when deciding if using single hop and multi-hop? Explain the typical methods to save energy using examples.

**Energy Sources and Harvesting**

The problem with WSNs is the fact that wireless nodes, are only able to stay alive for a given amount of time because of their battery. Often less than a year. To overcome this, one can implement energy harvesting into the nodes



Figur 1 – Energy Source and Harvesting Methods Overview

Solar

Pros: Easy maintainable, low price

Cons: Only efficient outside in the sun

Passive humen-powered systems

Pros: Generate power when in use.

Cons: No power while not in use.

**Energy Models for Transmitter and Receiver**

Consumption during transmission

Energy consumed during transmission includes:

* power spent on power amplifier for RF signal generation,
* power spent on other circuitry, e.g. baseband processor,

(Efficiency calculation in percentage)

is the desired transmission power.

and are constants depending on process technology and amplifier architecture.

The energy to transmit a packet of size *n*-bits, with bit rate *R* and coding rate , startup energy

Consumption during receiving

Energy consumption during receiving depends on the time receiver is on, the power for receiver circuitry, and the decoding energy.

is a more or less constant power, e.g., to drive LNA (Low Noise Amplifier) in the RF front end.

is the decoding overhead, which can be substantial depending on the specific FEC (Forward Error encoding) in use.

**Single-hop vs. Multi-hop**

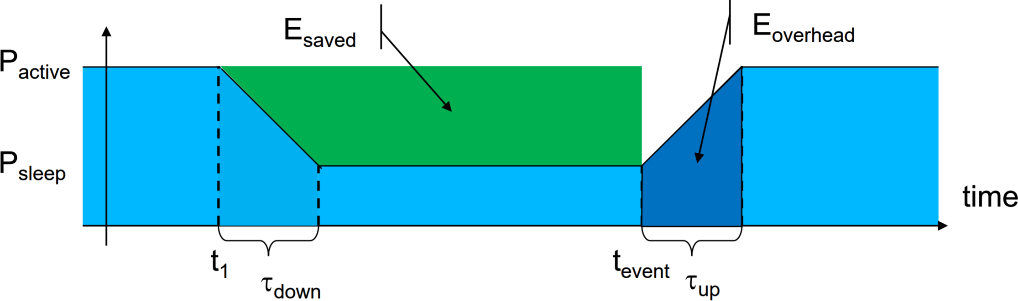
Transmit and receive = expensive = don’t do it!

Computation vs. communication energy cost

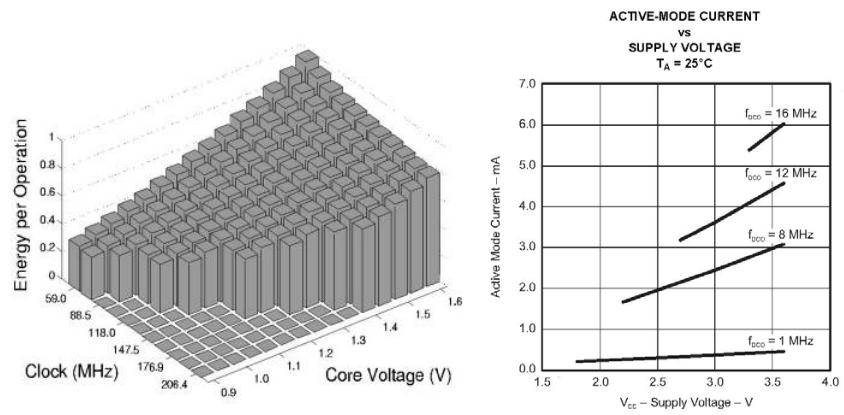
* Energy ratio of “sending one bit” vs. “computing one instruction” is between 220 and 2900 in the literature. To communicate (send & receive) one kilobyte = computing three million instructions!

**Save energy example**

Active/Idle/Sleep. Find the duty cycle where . (used in SMAC)



Dynamic Voltage Scaling (DVS). Lower clock allows lower supply voltage.



#### Q3: What are the optimization goals of WSNs? What are the basic design principles in WSNs? Give one or two examples in the protocol design which can reflect the design principles.

**Optimization goals**

**Quality of Service - WSN metrics missing important events**

high-level QoS attributes in WSN highly depend on the application. Two such examples is

**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.

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

**Energy Efficiency - How much energy is spend for each important event, Lifetime.**

Both the scalability and the robustness optimization goal, and to some degree also the other goals, make it imperative to organize the network in a distributed fashion. That means that there should be no centralized entity in charge – such an entity could, for example, control medium access or make routing decisions, similar to the tasks performed by a base station in cellular mobile networks. The disadvantages of such a centralized approach are obvious as it introduces exposed points of failure and is difficult to implement in a radio network, where participants only have a limited communication range.

**Scalability**

Scalability is not a free property you can get for free. It costs energy. Because with scalability more collision and that equals wasted energy. Ex Using a contention based mac, can scale to many nodes. But with more nodes there is an increased probability of collisions.

**Robustness**

Nodes 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 example of this could be a routing protocol that takes the nodes battery into account.

**Design principles**

**Distributed Organization**

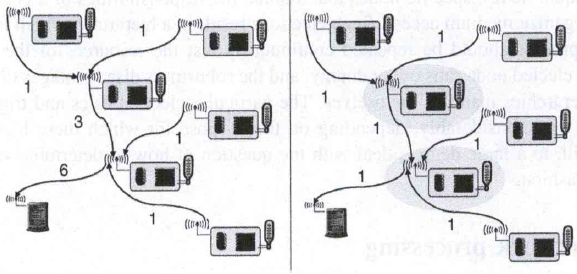
The WSNs nodes should cooperatively organize the network, using distributed algorithms and protocols. **Self-organization** is a commonly used term for this principle.

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). 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. The particular election rules and triggering conditions for reelection vary considerably, depending on the purpose for which these hierarchies are used.

**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. Several techniques for in-network processing exist, and by definition, this approach is open to an arbitrary extension – any form of data processing that improves an application is applicable.

The Perhaps 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. In such a case, it is evidently not necessary to transport are readings from all sensors to the sink, but rather, it suffices to send the average or the minimum and maximum value. Recalling from Section 2.3 that transmitting data is considerably more expensive than even complex computation shows the great energy-efficiency benefits of this approach.



The idea of aggregation. In the left half, a number of sensors transmit readings to a sink, using multihop communication. In total, 13 messages are required (the numbers in the figure indicate the number of messages traveling across a given link). When the highlighted nodes perform aggregation – for example, by computing average values (shown in the right half of the figure) – only 6 messages are necessary.

**Adaptive fidelity and accuracy**

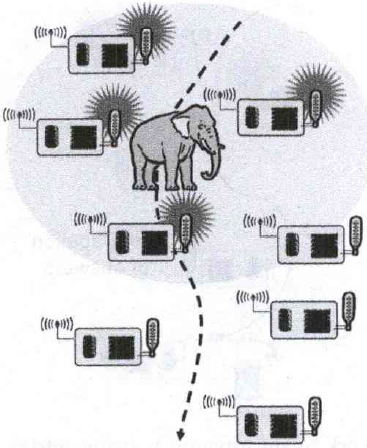
Section 2.3.4 has already discussed, in the context of a single node, the notion of making the fidelity of computation results contingent upon the amount of energy available for that particular computation. This notion can and should be extended from a single node to an entire network [246].

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. But in return for this, more energy has to be invested. Similar examples hold for event detection and tracking applications and in general for WSNs.

Hence, it is up to an application to somehow define the degree of accuracy of the results (assuming that it can live with imprecise, approximated results) and it is the task of the communication protocols to try to achieve at least this accuracy as energy efficiently as possible. Moreover, 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), and so forth. Therefore, the application needs feedback from the network about its status to make such decisions.

But as already discussed in 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, let alone with communication protocols that implement these decisions. This is still one of the core research problems of WSN.

**Data-centricity**

In traditional communication networks the relationship is usually the pair of communicating nodes – the sender and the receiver of data. In a WSN 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. 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. This fact that not the identity of nodes but the data are at the center of attention is called data-centric networking. For an application, this essentially means that an interface is exposed by the network where data, not nodes, is addressed in requests.

In a data-centric application, all the application would have to do is state its desire to be informed about events of a certain type – “presence of elephant” – and the nodes in the network that possess “elephant detectors” are implicitly informed about this request.

An example is the **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.

* **Decoupling in space** –neither sender nor receiver need to know their partner.
* **Decoupling in time –** answer not necessarily directly triggered by question asynchronous communication.

Entities can publish data under certain names, e.g., sensors

Entities can subscribe to updates of such named data by registering an interest in a particular type of data, e.g., sink.

Button line: In WSNs, sinks are more interested in querying an attribute of the phenomenon rather than querying an individual node.

Construct routes based on the interest of query message, forward the data packets to the next hop based on the routes (e.g. Direct Diffusion)

Q4: Explain the pro and cons of contention based MAC protocols and schedule based MAC protocols. What are the main energy issues in WSN MAC? Describe the basic schemes of SMAC and its additional functions. Explain how SMAC tackles the energy issues in its design.

The objectives of a WSN MAC protocols is Collision Avoidance, Energy Efficiency, Scalability. But it also have Energy problems, such as. Collision wasted energy when two packets collide. Overhearing Receiving packets that was destined for another node. Idle listings wasted energy to receive when no one is sending.

Contention based MAC can be scaled to many nodes, but with more nodes there is an increased probability of collisions.

Contention-based: CSMA/CA, SMAC, BMC. Scheduled-based: TRAMA

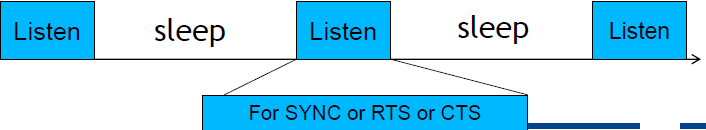
SMAC have features that can help solve energy waste.

* Sleep/listening ( Idle Listening and overhearing energy waste )

Trades energy efficiency for lower throughput and higher latency

Nodes can go to sleep during other nodes transmission

While sleeping, radio is off. Hence packet exchanges can’t happen

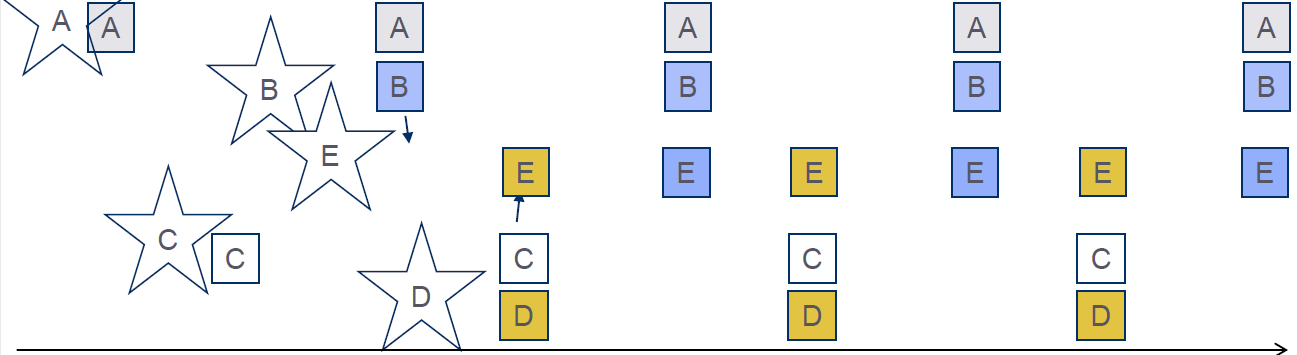


* Synchronization

First listens to the medium and If it receives a SYNC it becomes a follower.

Uses a sync PHASE to exchange wakeup schedules between neighbors, if none received it will choose it’s own schedule and broadcast it with a SYNC packet.

If a node receives, a different schedule after it chooses and announces its own. If it had no neighbors, it discards its own schedule, if It has then adopts both.



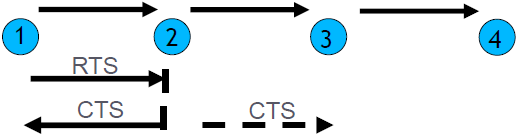
* Message passing ( Minimize overhead )

A receiver often needs to need get all messages before it can do in-network processing like aggregation. But with more packets there is a higher probability of corruption.

The solution is to fragment long messages into many small fragments and send them in a burst, *using only one set of RTS/CTS to reserve the channel and acks for each fragment*. Nodes that wake up in the middle of a burst by back to sleep by hearing acks.

* Adaptive listening ( Reduce latency, but paid in energy )

Let a node who overhears it’s neighbors transmission stay awake instead of following Sleep/listening. Reduces latency by staying awake(node 3 ) and forward to 4 if it is still hearing.



#### Q5: Explain the difference of distributed MAC and centralized MAC. What is preferable for most use cases in WSNs and Why? What are the main energy issues in WSN MAC? Clear channel assessment (CCA) and low power listening (LPL) are the two highlight features of BMAC. Explain how CCA and LPL work in BMAC.

**Centralized MAC.**

Having a central station control when a node may access the medium. Example is the polling and centralized of TDMA schedules. For this to work there is a need for time synchronization.

Simple , Usually no issues with collisions, overhearing and idle listening. Only burdens the central station

Not directly feasible for non-trivial wireless network sizes, but can be usefully if the network is divided into smaller groups.

Centralized multiplexing access, therefore, lacks flexibility and scalability to adapt to the variation of WSN applications

**Distributed MAC.**

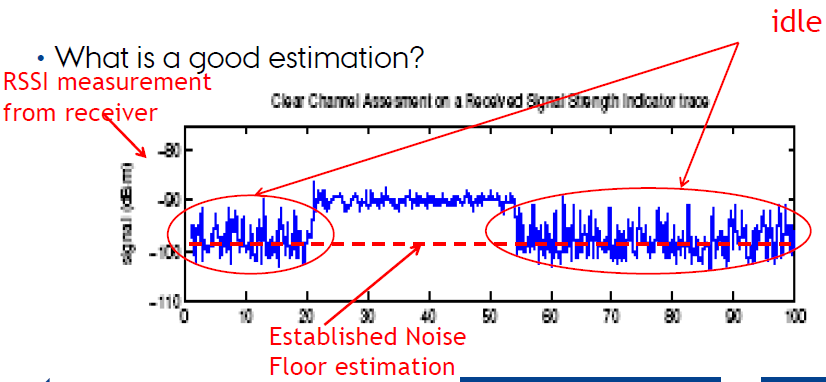
Using distributed MAC protocols allow nodes operate in a decentralized manner. Therefor it is easier to implement and perform more flexible and scalable control mechanisms, which may fit well with the requirements of WSNs.

But it is not collision-free protocols and need listen-before-talk(CSMA) schemes so nodes to keep sensing the channel. This results in high energy wastage due to collisions, idle listening, overhearing, and message overhead. S-MAC attempts to reduce all four types of energy wastage.

#### B-MAC

**Carrier sensing using clear channel Assessment ( CCA )**

For effective collision avoidance, BMAC must accurately determine if the channel is clear. To do that it needs to tell what is the signal and what is the noise. If not handled correctly it could lead to waste if a real signal is taken as noise or packet collisions.

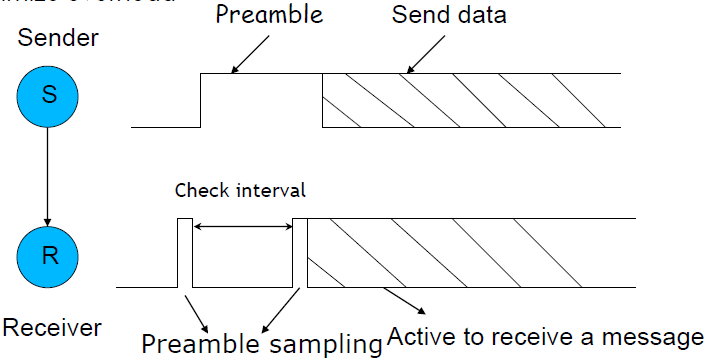


**Sleep/Wake scheduling using low power Listening (LPL) Reduce listening cost**

Nodes periodically wakes up, turns radio on and checks the channel

If activity on the channel is detected, node powers up and stays awake for the time required to receive packet.

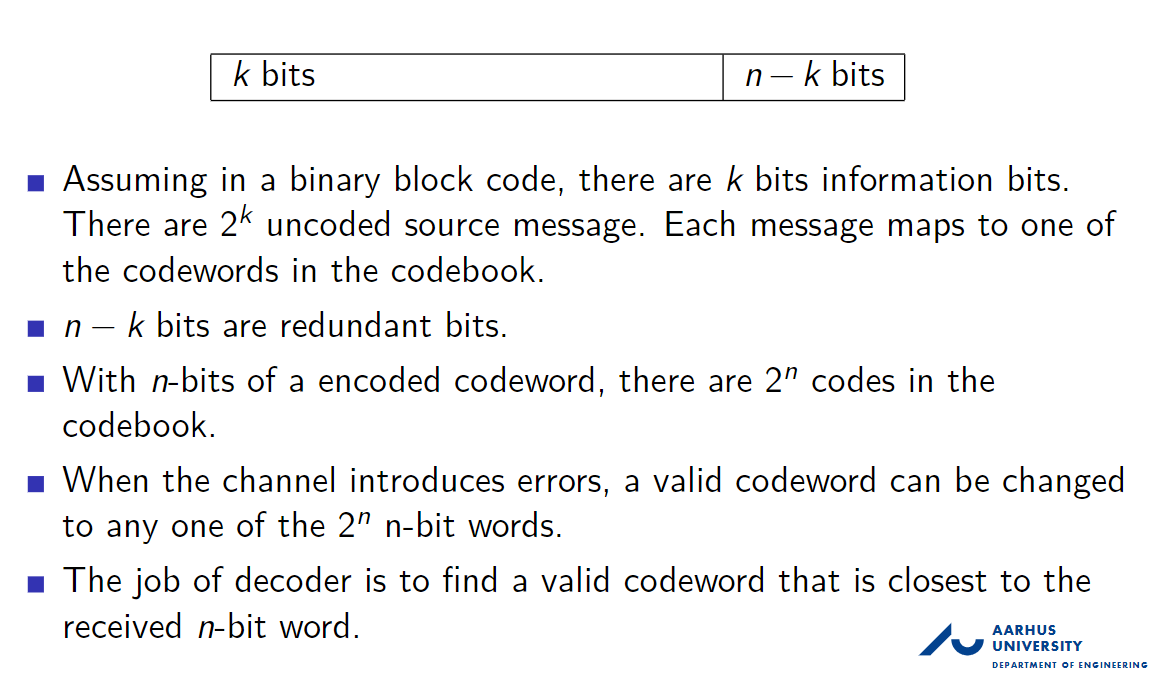
It goes back to sleep if a received or after timeout.



#### Q6: Explain the basic two approaches for error control in WSNs. What are the pros and cons of each approach? What are the tradeoffs between FEC, ARQ and transmission power? What is the difference of reliability in the link layer and in the transport layer? Explain the tradeoff between pure end-to-end reliability and end-to-end reliability plus link layer retransmission.

Error-control to ensure: Error-free, in-sequence, duplicate-free and loss free.  
Causes: Fading, interference. We have two protocols:

**Forward-error-control (FEC):** Proactive. Adds error control (data redundant) bits to each packet by using an error-correcting code (ECC) but at the cost of encoding/decoding, energy and longer packets. The receiver can then detect and correct errors using these extra bits. E.g. Satellites orbiting Uranus use FEC because retransmissions are costly!

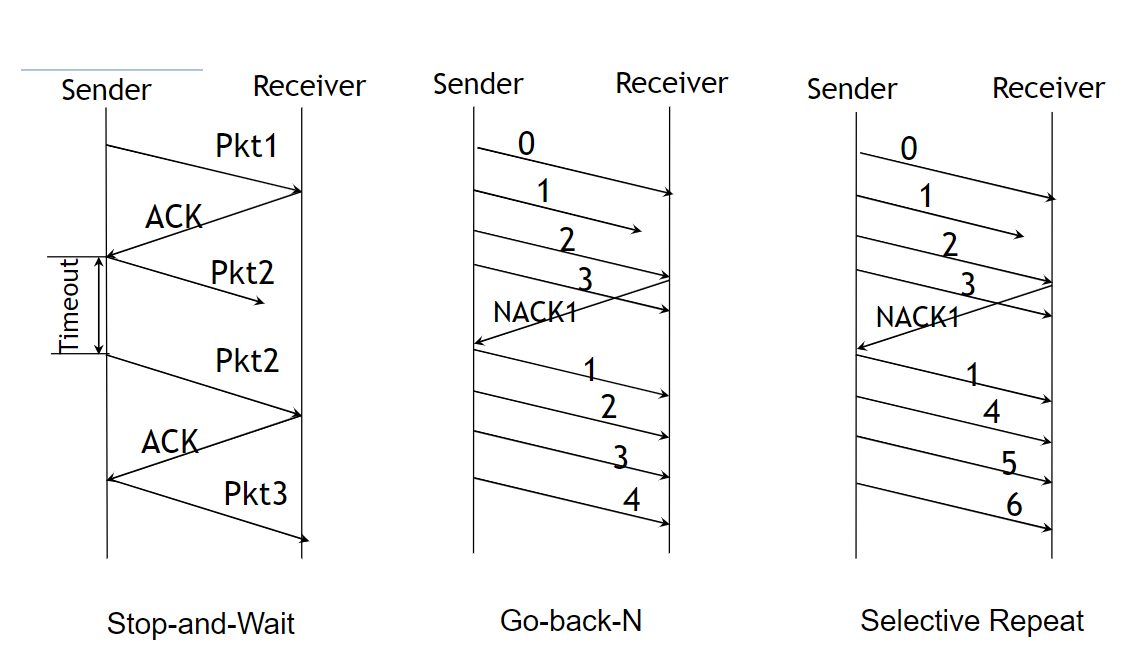


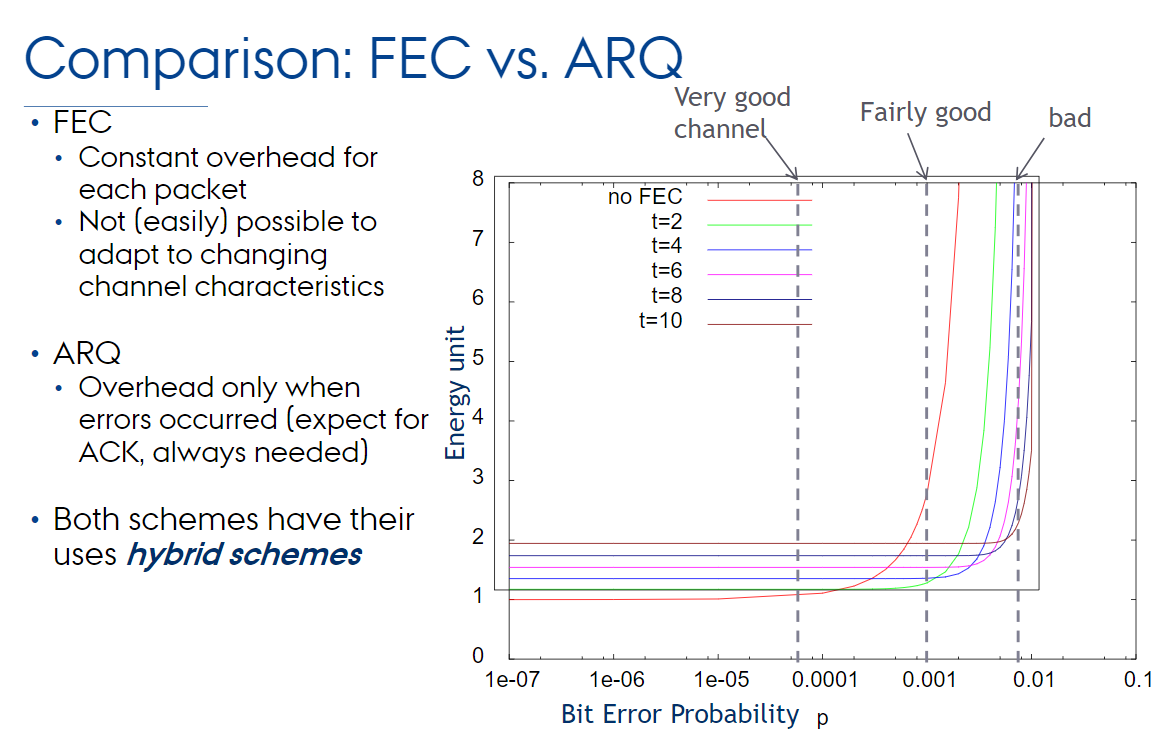
Advantages:

* Reduce the cost of communication systems.
* Extend the wireless network coverage and mitigate deep fading.
* Prolong life-time of the WSN by saving energy.

**Automatic Repeat request (ARQ):** Reactive. Compute checksum and add to the packet being sent (cyclic redundancy check CRC). The receiver sends positive or negative acknowledgment. Sender uses timers to resend packet if no response. Set maximum number of retransmission attempts.

* Stop-and-wait (one packet outstanding). USE IN NOISY ENVIRONMENT, but more packets!
* Go-Back-N (sender sends N packets. Receiver drops packets without expected sequence nr. In this example it drops packet 2 and 3. Then sends NACK1. Sender retransmits packet 1 to N). FEWER PACKETS IF NO NOISE! But using less power.
* Selective-repeat (receiver buffers packets out of sequence. Receiver sends NACK1 and sender resends only that packet, then sends packets to N).





Transport-layer reliability: End-to-end. Use this for good channels and low noise.

Link-layer reliability: Between two nodes. Use this for bad channels and high noise. OR long routes (> 10 hops).

#### 

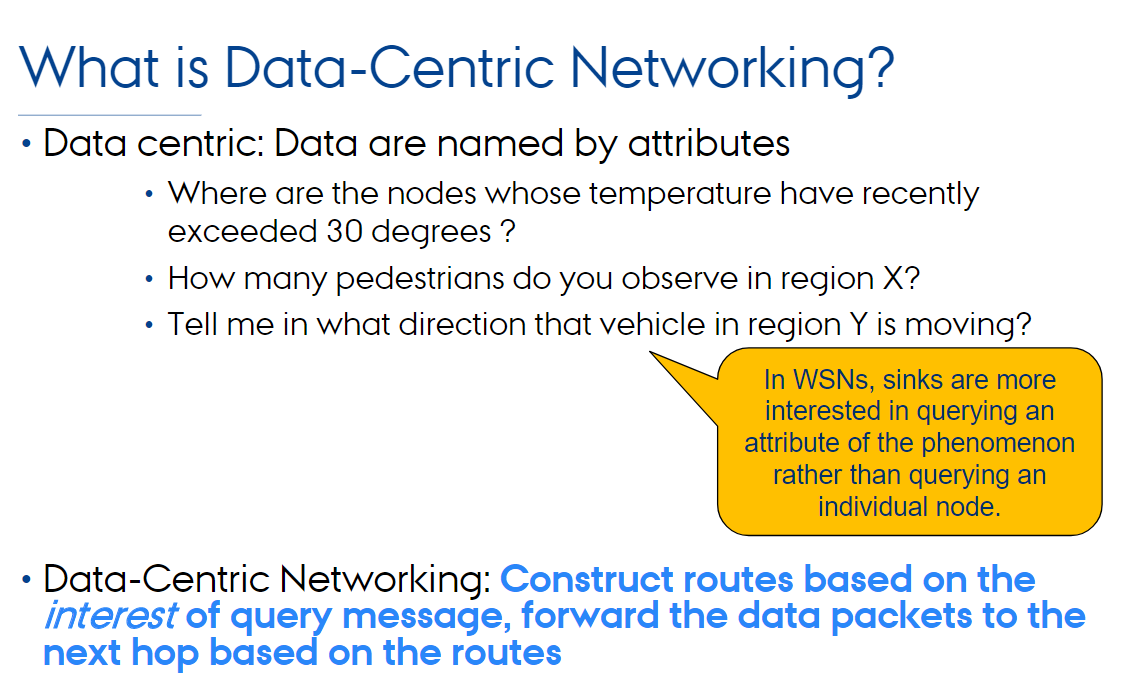
#### Q7: Explain the data centric networking, the motivation using it in WSNs and its difference comparing with ID-based routing. Explain how directed diffusion works.

Routing is about how we get from A to B in a network the most efficient way. Requirements for routing:

Efficient (low overhead), resilient (changing topology), self-configuring, distributed).  
  
**ID-centric:** Each node has a unique ID. Construct routing tables about how we reach each node by their ID. Then forward packets using these tables. Challenges: Shortest packet delay/hop count is not always best. We should look at energy/packet content/event data.

Examples: Flooding, gossiping.

**Data-centric:** Examine each data packet and route it based on content. Decoupled in space (sender and receiver does not know each other). Decoupled in time (asynchronous communication). We can implement publish-subscribe pattern in WSN. Entities/nodes publish data by certain properties. Entities/nodes subscribe to updates of certain named data.



**Directed Diffusion**

Works by propagation interests of data throughout the WSN. This maps publications and subscriptions to each other. This way we tell each node what data is available and who can provide it.

It has four phases (IRGD):

* Interest Propagation (An interest specifies what a user wants, which events. Sent from sink).
* Reinforcement (Sink links specific links to create a strong route for events).
* Gradient Establishment (A gradient is a direction to send data should event occur).
* Data delivery (Events sent from source to sink.)

Interest: Use attribute-value pairs. Set type, interval, duration, which sensors.  
Reply: Sensor detected event. Includes type, location, timestamp.

**Negative reinforcement:** The sink can send out these to delete established routes when creating new.

DD (Directed Diffusion) and SPIN: In DD the sink queries sensors for data. In SPIN the sensors advertise that they have data.

#### Q8: Explain the motivation to use data aggregation in WSN. What is the relation between data aggregation and networking in WSNs? Explain how data aggregation works, the challenges of data aggregation, and the pros and cons using data aggregation.

Aggregation is a part of the network layer

Providing robust aggregation solution with resource constrained sensor node is challenging; however, aggregation or other advanced in-network processing is a key enabler for efficient networking

Data Aggregation Motivation

Redundancy and correlation

* A certain degree of overlap and redundancy is created as measured sensor data is often also generated by nearby nodes
* Measured data can be expected to be highly correlated allowing further improvements of the information quality by using data fusion approaches (possibly exploiting further available meta information)

Energy constraints and network congestion

* Data transmission in sensor networks is much more energy expensive compared to local computation efforts
* The reduced number of transmitted messages towards the base station or sink helps reducing network congestion (especially near the base station)

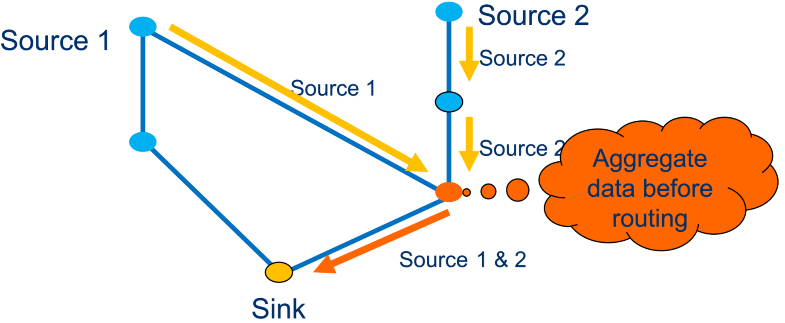
Data Aggregation Concept

Data Aggregation:

* Process of combining data or information to estimate or predict events, i.e., computing a smaller representation of the messages that is equivalent in its content to all the individual messages.

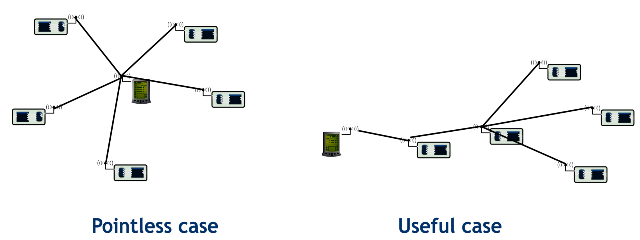
Idea:

* Take advantage of the routing hierarchy and high network density



The benefit of aggregation depending on the location of data sources and sinks

* Aggregation can be useful or pointless



Metrics for data aggregation performance evaluation

**Accuracy**: Difference between value(s) the sink obtains from aggregated packets and from the actual value (obtained in case no aggregation/no faults occur)

**Completeness**: Percentage of all readings that are included in computing the final aggregation at the sink

**Latency**

**Message overhead**

Data Aggregation Components

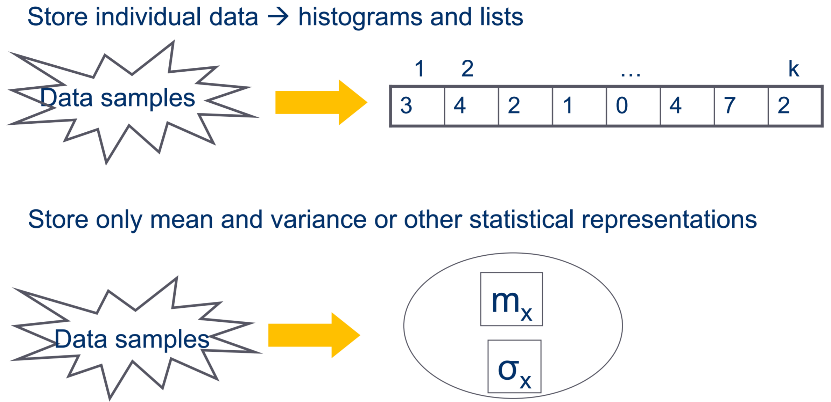
**Data Storage**: Store sensor data in a memory efficient way, while preserving the accuracy of the information.

**Aggregation Functions**

**Aggregation Paths**: Which are the optimal aggregation points on the path from sensors to sink? Which is the most suitable path from source to sink to favor data aggregation?

Data Storage Representations

Data can be represented with different degree of accuracy



Aggregation Functions

Very simple functions

* Average, max, min, median
* Suppression of duplicates

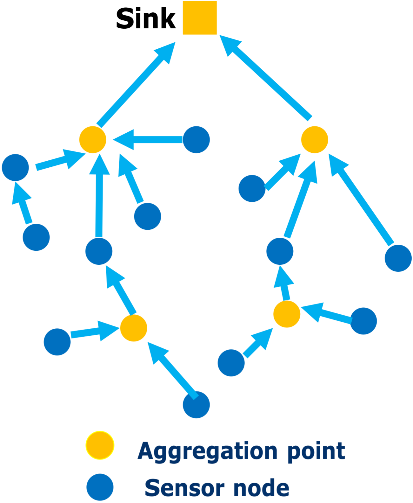
More sophisticated functions

* Exploit spatial and temporal correlation
* Signal processing (convolution, filtering, etc.)

Aggregation Paths

Challenges:

* Find the optimal number of aggregation points
* Selection of aggregation points
* Dynamic change of aggregation points (Energy efficiency)



Data Aggregation Pros & Cons

Pros:

* In-network processing reduces traffic load by aggregating data in route
  + Energy and memory efficiency
* Scalable to large numbers of both sinks and sensors

Cons:

* Requires careful design to tradeoff *accuracy* and *storage* and *message size*
* Incur information loss, making robust estimation is difficult:
  + E.g. a single outlier reading can damage MAX/MIN aggregates

#### Q9: Explain why TCP is not suitable for transport layer in WSN. Explain how PSFQ and ESRT work? What is the difference between PSFQ and ESRT?

**TCP is not suitable for transport layer in WSN**

* TCP has a large overhead, each TCP segment has a minimum header of 20 bytes for port numbers, sequence numbers, checksum, window size, and more. In a WSN packets will be small, just a few bits of sensor data.
* TCP requires perfect reliability and accept no loss. But in a WSN as long as we detect events we may not need all packets.
* TCP connects two end nodes and the intermediate nodes just forward blocks of bits and don’t care the content. In sensor networks, intermediate perform in-network processing or aggregation of data.

**Pump Slow Fetch Quick (PSFQ)** - packet reliability. It is applied in scenarios that are *pack losses are not tolerable, delay not critical* such as firmware update, Eg Downstream sink to source.

The user slowly injects packets into the network. Intermediate nodes store packets, forward if *in-sequence*

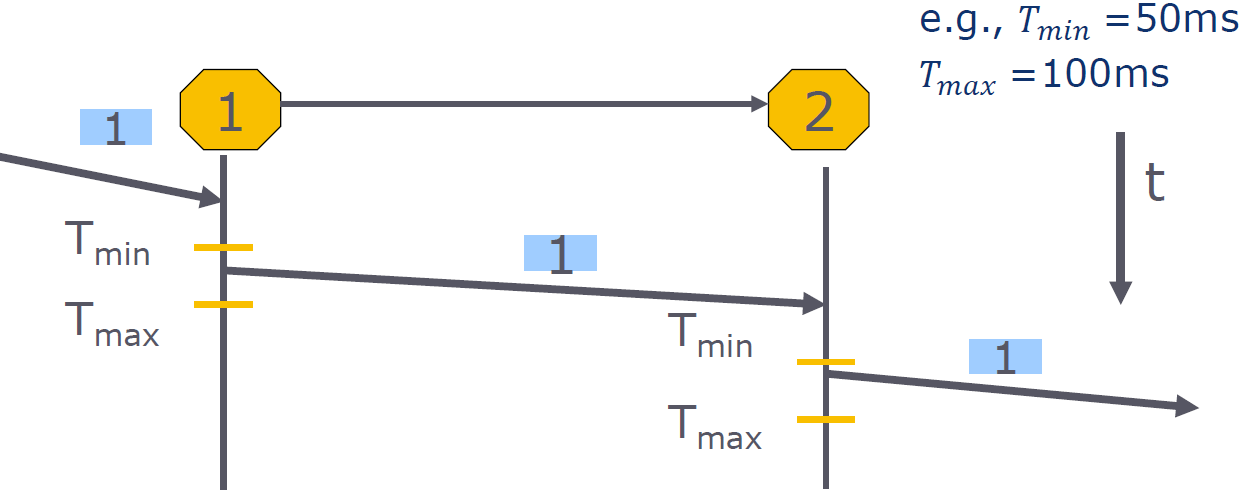
Out-of-sequence: buffer, request missing packet(s) hence *a fetch* operation (a NACK). This allows a previous node to resends missing *packet => local recovery*.

PSFQ has three functions:

**Message Relaying (PUMP Operation)**

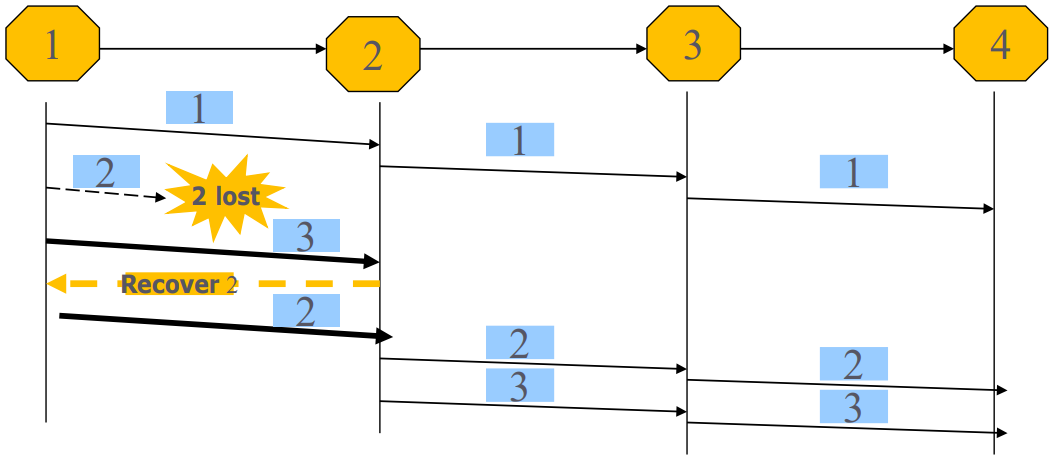
A user node broadcasts a packet to its neighbors every until all the data fragments have been sent out. If *not* *duplicate* and *in-order* and *TTL not 0*, cache and schedule for forwarding at time where .

Random delay allows a downstream node to recover the missing packets before the next packet arrives from an upstream node Also allows reducing the number of redundant broadcasts of the same packet by neighbors.



**Relay initiated error recovery (FETCH Operation)**

A node goes into fetch mode when a sequence number gap is detected. A node aggressively sends out NACK messages to its immediate upstream neighbors to request missing packets. If no reply is heard or only a partial set of missing packets are recovered within a period then the node will resend the NACK every interval until all packets are recovered. Since it is very likely that consecutive packets are lost because of fading conditions, PSFQ aggregates losses such that the fetch operation deal with a “window” of lost packets

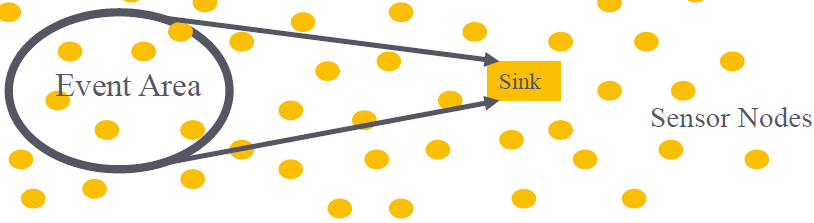


**Selective status reporting (REPORT Operation)**

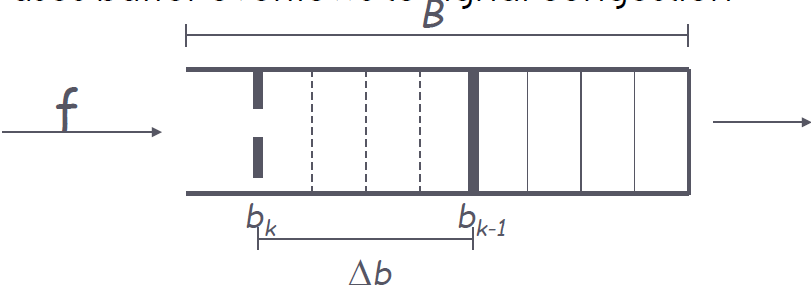
The report operation allows the data source to assess how many nodes have already received the complete code block and can thus switch to the new software. The sink node requests reporting by setting a reserved bit in the TTL field of an inject message. Report messages are generated by the most distant nodes (those that receive packets with a TTL of one) and travel back to the data source. The data generated by an end node or an intermediate node contains the node’s own address and a summary of the already received segments.

**Event-to-Sink Reliability Transport (ESRT)**

GOAL: To reliably detect/estimate event features based on the collective reports of several sensor nodes observing the event. (to guarantee event reliability)



ESRT uses buffer overflows to signal congestion

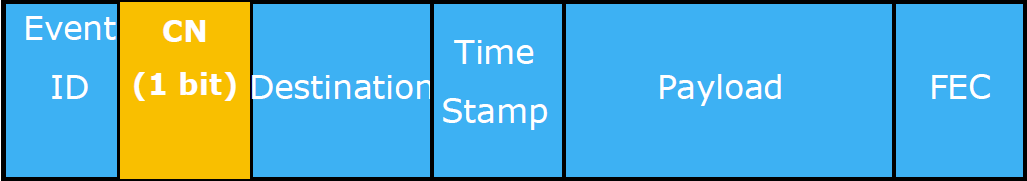


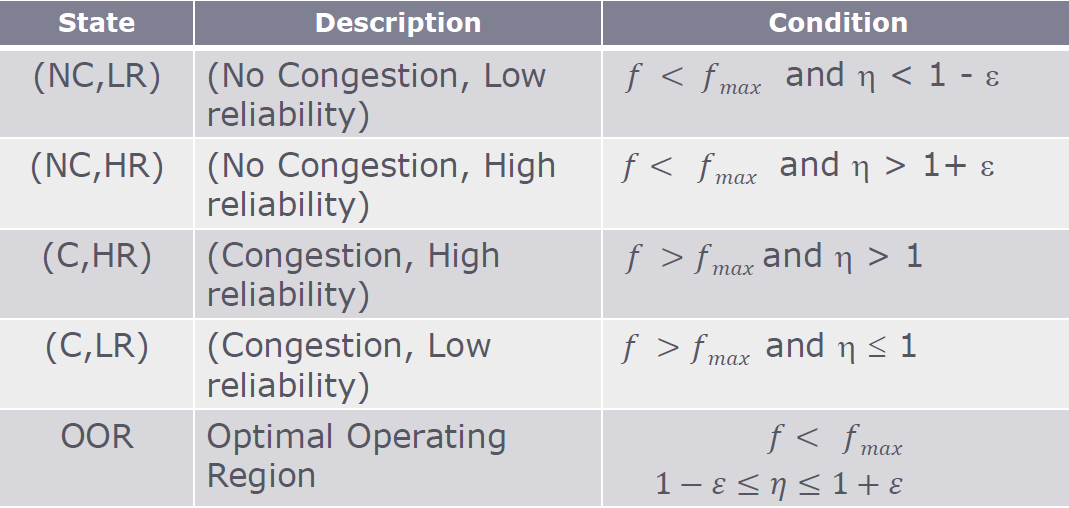
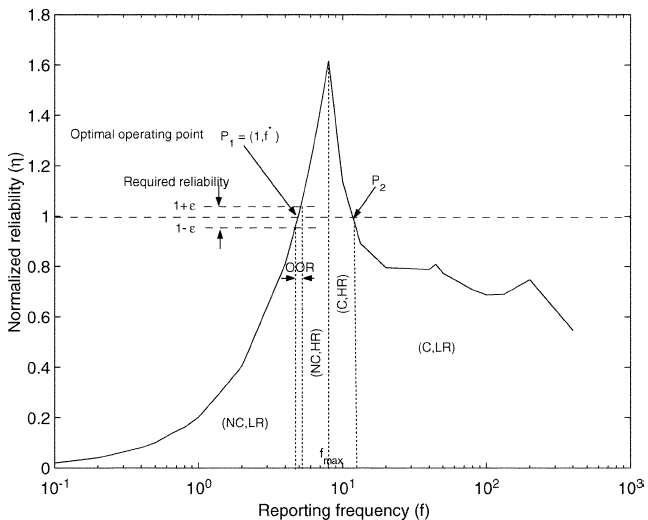
bk: Buffer fullness level at the end of reporting interval k

Δb: Buffer length increment over past interval Δb=bk-bk-1

B: Buffer size

f: Reporting frequency





: observed event reliability

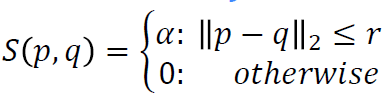
: desired event reliability

: reporting frequency rate

: maximum frequency without congestion

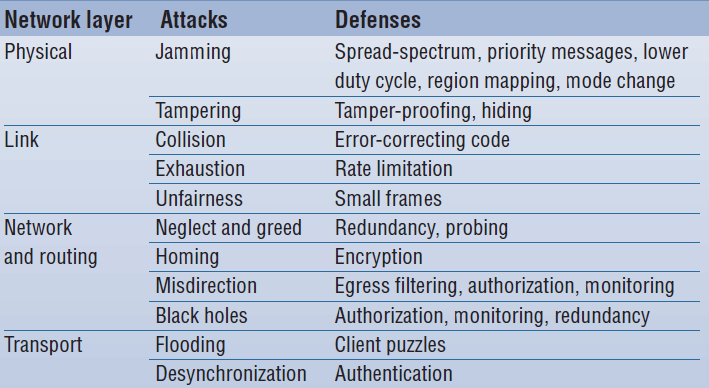
Two sensing models (One commonly used)

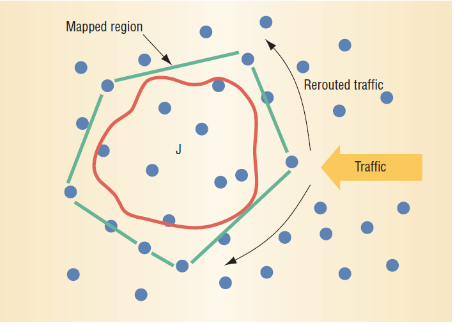
Boolean sensing model: all sensors of the same sensor modality. Assumes they are counting packets.



#### Q10: Explain the possible vulnerabilities and security attacks at different protocol layers in WSNs and the corresponding countermeasures.

A attack on a WSN could do a denial of service or be used to steal information using a passive attack such as overhearing. Active attacks can also happen such as reporting cheap routes to the routing protocols or exploiting access to the medium.



**Physical layer**

Jamming - by jamming the wireless medium, you can just the WSN from working. If the attack is intermittent nodes may be able to send a few messages with high power and priority back to the BS. The nodes can work together ensure that such messages will be received. Using the Jamming reports the nodes can form a zone around the jamming spot and work around it.

**Link - layer**

The link layer is responsible for node to node delivery and must do it in an energy efficient and fair manner. Attackers can abuse this fact and send messages to make collisions and this uses less energy than full on jamming. To defend this, we can use Error correcting codes for tolerating variable levels of corruption in messages at any layer. However, this introduces additional overhead. There are no really good defense agents this and A subverted node could intentionally and repeatedly deny access to the channel, expending much less energy than in fulltime jamming.

**Network - layer**

As the network layer has the responsible for routing packets attack type would be to advertise routes as basically free and then drop the packets this is known as a black hole attack. To defend against this type of attack, one could use Authorization.

**Transport - layer**

As we know the transport layer protocols provide end-to-end reliability. So the obvious attack scenario is abusing this fact by flooding. An adversary node sends out multiple end-to-end connection establishment requests, effectively exhausting the memory of a node. A defense requires clients to demonstrate the commitment of their own resources to each connection by solving client puzzles. The server can create and verify the puzzles easily, and storage of client-specific information is not required while clients are solving the puzzles. Servers distribute the puzzle, and clients wishing to connect must solve and present the puzzle to the server before receiving a connection. An adversary must therefore be able to commit far more computational resources per unit time to flood the server with valid connections. Under heavy load, the server could scale the puzzles to require even more work by potential clients. This solution is most appropriate for combating adversaries that possess the same limitations as sensor nodes. It has the disadvantage of requiring more computational energy for legitimate sensor nodes, but it is less costly than wasting radio transmissions by flooding.