Lecture on Sensor Networks

Communication in sensor networks

Terms

Source and sink	The sender of a message is often called the source, the recipient is called the sink. In the context of a data stream, information is floating from the source to the sink. The assignment of these roles may change dynamically.
Frame length resp. Frame	A frame is a time interval which is fully occupied by one sender. Usually one data packet on the MAC layer occupies a frame.
Number of hops	Denotes the number of participants a message has to pass between the source and the sink before reaching its destination. The distance between a source and a direct neighbor is counted as 1 hop.
ACK	Acknowledgment that the message was received correctly.
Routing	Means forwarding a packet into the right direction in order to reach a destination.
Carrier	Usually means a carrier frequency. The carrier frequency has to be detected before the modulated information can be decoded.
Idle	A participant in idle-state does not want to transmit anything and monitors the medium only if necessary, because it is not clear when a new message will arrive. In the case of sensor networks, idle-times are particularly undesired because they waste energy.

Communication in sensor networks

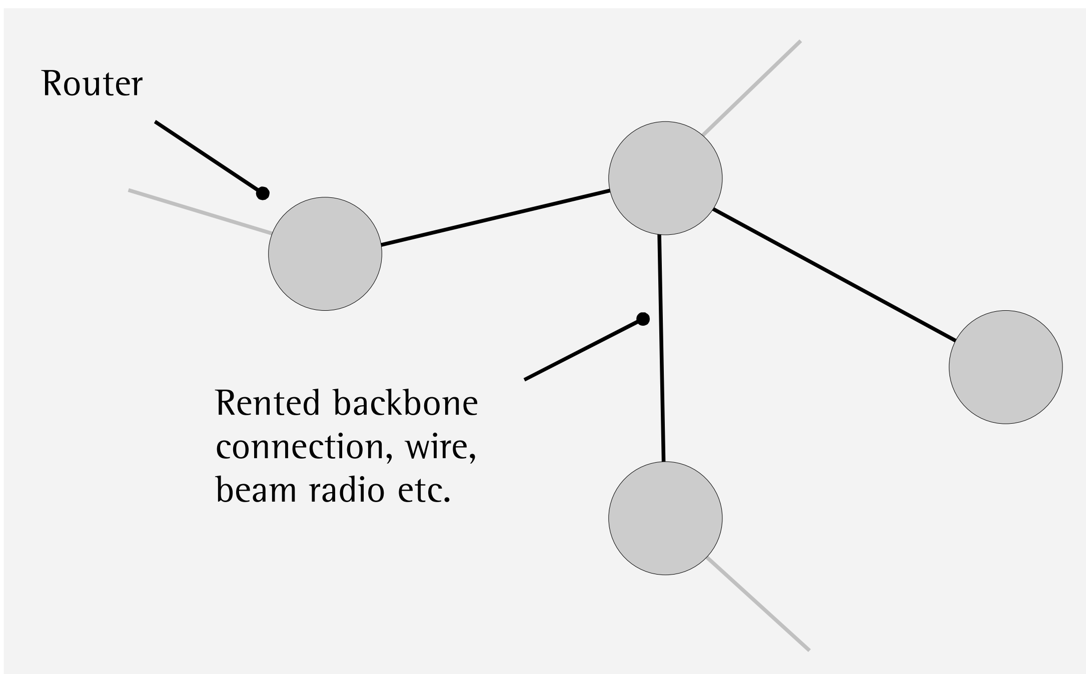
Classification and Motivation: The (ISO/OSI-network-) layer model

Application layer Presentation layer Session layer	
Transport layer	Is about end-to-end connections like flow control, preservation of the packet order etc. Single packets were already delivered by the network layer.
Network layer	How are data packets dispatched (especially: routed etc.) between distant participants?
Link layer	How is data dispatched between **two nodes** / routers in 1-hop distance? Also MAC-problems.
Physical layer	Specification of the wiring, carrier frequencies, etc.

Communication in sensor networks

Classification and Motivation

Conventional wired computer networks consist of special routers which are connected by dedicated channels. There is only one router at the beginning and end of a (virtual) channel. Both of them use the medium exclusively. Hence, gaining access to the medium causes no problems. Instead, a way through the graph of connected routers has to be chosen.



Communication in sensor networks

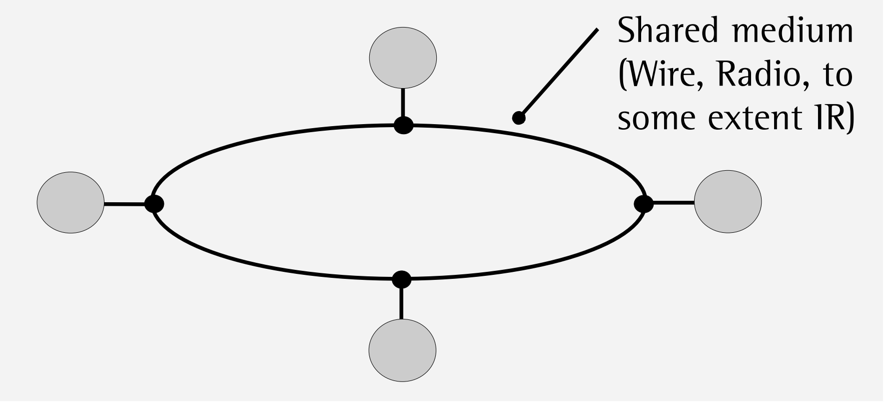
Classification and Motivation

If many computers communicate through a shared channel, the assignment of the „right to talk“ has to be organized. On the other hand there is no routing problem if everyone listens to the same channel.

Example:

In discussions, every participant should have an exclusive right to talk. The negotiation about the right to talk often takes place via the „visual channel“. The intention to talk is signaled to the discussion leader who then assigns the right to talk. Yet access control to the medium does not always work very well.

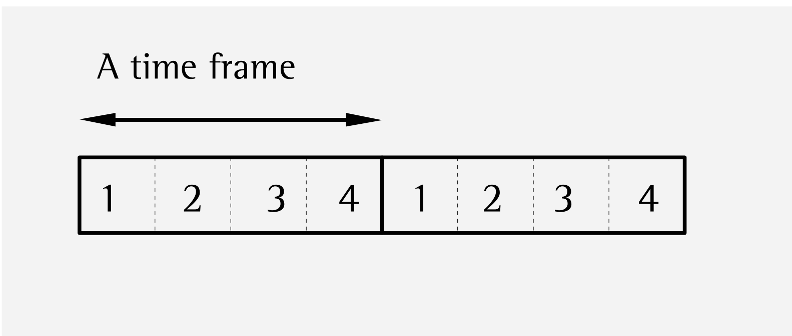
In the context of mobile ad-hoc networks the shared medium often is an entire radio band (e. g. the ISM band on 2,4 GHz). However, many sensor nodes use only a single frequency or infrared signals.

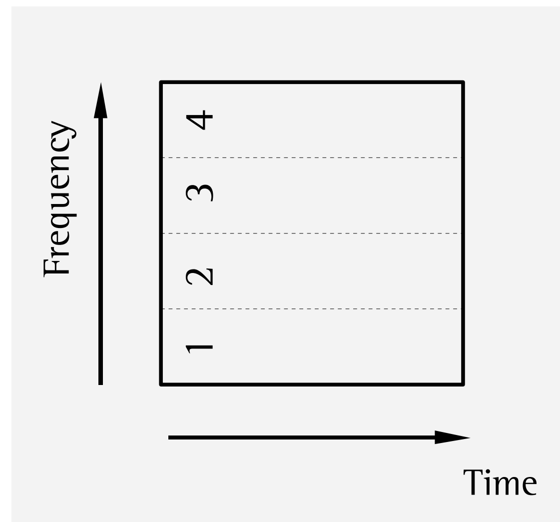


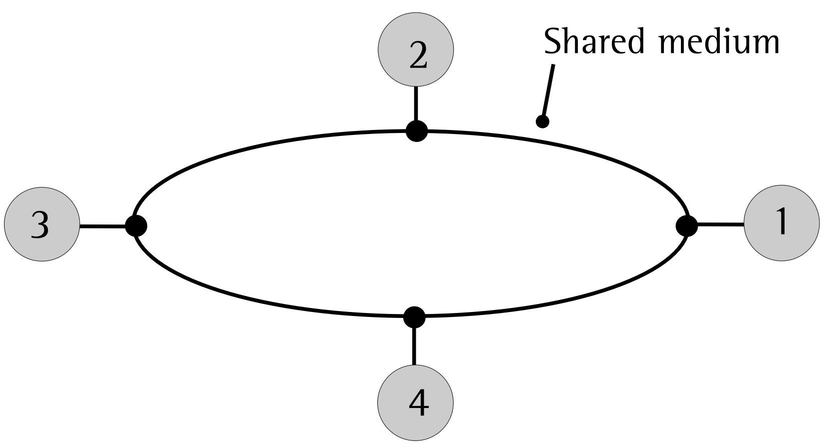
Communication in sensor networks

Static channel assignment

The channel is divided 1) into time slots
2) into frequency spectra







Communication in sensor networks

Static channel assignment: Problems

Subdivision is finer than the number of participants	Every participant gets access, but bandwidth is wasted.
Subdivision is coarser than the number of participants	Some participants do not get a slot. Nevertheless, bandwidth is wasted because not every participant uses its channel for the entire time.

Example: Classical telephony (mixed dynamic/static)

The assignment of wires (channels) to connection desires is dynamic – and can therefore fail. Once the connection is established, the required bandwidth is available for the duration of the conversation – this is guaranteed.

Example: Voice-over-IP (dynamic only)

Voice is split up into small packets and transmitted over the IP-based network. The packet only arrives (on time or at all) if enough bandwidth exists (mostly works due to sufficiently large bandwidth; so-called over provisioning). But guarantees cannot be given.

Communication in sensor networks

Dynamic channel assignment: Basic assumptions

The following channel assignment procedures assume these principles:

Station model:	There are N independent stations. The stations want to transmit data packets over the medium. Stations send statistically independent of one another. Packets are only transmitted completely.
Single channel model:	Only one channel is used for communication. Everyone is able to send and receive. If not explicitly ruled out, a sender is able to receive its own message. This is often not the case with sensor nodes (consider our ESB nodes).
Collision assumption:	A collision occurs if two frames overlap in time . This can be detected by all stations. There are no other errors rather than collisions.

Subdivision of time (particularly important in sensor networks):

a) continuous:	Stations may send at any time. Frames are not synchronized.
b) subdivided:	Time is split up into intervals. In every interval 0, 1 or more packets are possible (corresponding to inactive, success, collision).

Communication in sensor networks

Dynamic channel assignment: Basic assumptions

The following channel assignment procedures assume these principles:

Carrier detection:

With:	A station can detect an occupied channel. It does not try to send in order not to disturb the current transmission.
Without:	The availability of the channel is not checked, everybody may send at any time. Collisions of overlapping packets are detected afterwards. Detection if the channel is a wire: The amplitudes of the signals are superimposed, the untypical high amplitude can already be detected technically on the level of the network adapter. Less clear with radio channels, especially if sender and receiver are both mobile. Error protection takes place mainly on the logical level.

In the case of our sensor nodes no error protection is implemented on level 2 of the hardware. Thus, we have to implement it in software. What follows is an excursus on error protection.

Communication in sensor networks

Error control: Detecting vs. Correcting

Basically two variants are distinguished:
a) Error detecting codes
b) Error correcting codes (Forward Error Correction)

Both variants need a certain amount of redundancy. Which variant makes sense in which situation?

Error detection: Detection might be sufficient if the sink can ask the source for repeated transmission. Therefore, a feedback channel and a defined protocol are necessary. Error correction on the protocol level only makes sense with small error rates and / or high variance.

Example: On average every 1,000th byte is disturbed. Packets also consist of 1,000 bytes. Does a repeated transmission of defective packets make sense here?

Error correcting codes: A large amount of redundancy is necessary. This makes sense if a) no feedback channel is available or b) a retransmission delays the packet considerably (e. g. telephony, video conferences).

The applied technique also depends on the typical error distribution, particularly on its variance. If errors occur in so-called bursts, hence very clustered, a repeated transmission of a packet may be the solution. Whereas statistically independent errors promote error correcting codes.

Communication in sensor networks

Error control: The Hamming-Distance

Given two codewords:

10001001	0011	Operand 1
10110001	0101	Operand 2
00111000	0110	XOR

The **Hamming-Distance of two codewords** corresponds exactly to the number of bits the codewords differ in. This is calculated using the XOR-operation. In other words: If two codewords have the Hamming-Distance d, then it is necessary to „toggle“ d bits to transform one codeword into the other.

The **Hamming-Distance of a code** is defined as the smallest possible distance two arbitrary (but different) codewords of a code can have.

To detect errors we need a code with valid and invalid words. To be able to detect d bit errors in a codeword the code must have a distance of d+1. Why does not less distance suffice?

A code in which toggling one bit turns a codeword into another has the distance 1. Because all codewords are valid there is no redundancy. Every word is allowed. With a distance of 2 at least 2 bits have to be toggled to create a new word. Toggling only one bit instead creates by definition an invalid word. The same applies to more bits.

Communication in sensor networks

Error control: Error correcting codes

Error correcting codes need a distance of 2d+1 if d errors need to be detected and corrected. Why does less distance not suffice?

With a distance of 2d+1 we need to toggle 2d+1 bits to get from one valid codeword to another. If we toggle d bits only then the effort to get back to the original codeword is also exactly toggling d bits. But to get to a different (not the original) codeword when there are d bit errors it is necessary to toggle d+1 bits. If only d bit errors may occur then the shortest (and only) way to get to the next codeword is toggling d bits.

Example: 2 bits are encoded by 10

Orig. code:	00	01	10	11
E.corr.Code:	0000000000	0000011111	1111100000	1111111111

How many bit errors can be detected and corrected here?

Communication in sensor networks

Error control: Redundancy estimation of error correcting codes

How many bits do we need at least for the correction of a bit error?

We want to have 2^m valid codewords. r correction bits are needed. The error correcting code will have n = (m + r) bits in total. By toggling one of the n bits (redundant bits may be toggled, too) we get an (illegal) codeword with a distance of 1.That means, for each of the 2^m valid words we can create n invalid words. Hence follows

$$(n+1)2^m \leq 2^n$$

(n + 1) is composed of n invalid (each created by one bit error) and one valid codeword. The number of words of the error correcting code stands on the right side of the inequality.

n can be written as m data bits plus r correction bits.

$$(m+r+1)2^m \leq 2^{m+r} \Rightarrow \quad (m+r+1) \leq 2^r$$

With a given m we can estimate the number of correction bits a code needs in any case.

Example: 8 data bits require 4 correction bits / 1,024 data bits require 11 correction bits
Apparently useful especially for larger packets, but only 1-bit errors can be corrected.

Communication in sensor networks

Terms
Classification and Motivation
Static channel assignment
Dynamic channel assignment
Error control

Error control: The error correcting Hamming code

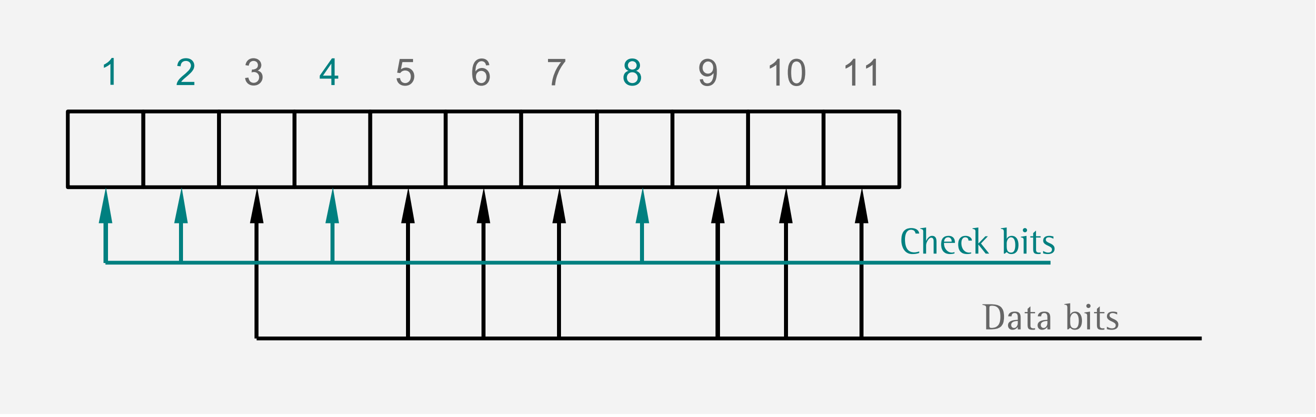
The Hamming code achieves the minimum of the previous estimation.

Algorithm: Number the bits from the LSB to the MSB. All 2^n bits are check bits, the rest are data bits. The data bits are filled up from left to right with the actual data, the check bits only depend on the data bits. Which data bit influences which check bit?

To see this, convert the number of a bit into its binary representation:

$11 = 8 + 2 + 1$

Data bit 11 hence influences check bits 8, 2 and 1. Check bit 1 is of course also influenced by the data bits 3, 5, 7 and 9. By definition all check bits combined with their data bits have to exhibit an even (or odd, depending what was agreed upon) parity.



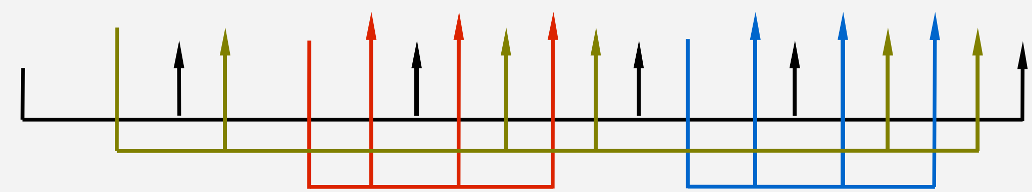
Communication in sensor networks

Terms
Classification and Motivation
Static channel assignment
Dynamic channel assignment
Error control

Error control: The error correcting Hamming code

Example: Data: 1101101

1	2	3	4	5	6	7	8	9	10	11	digit
?	?	1	?	1	0	1	?	1	0	1	data w/out check bits
1	2	1+2	4	4+1	4+2	4+2+1	8	8+1	8+2	8+2+1	Conversion



1	1	1	0	1	0	1	0	1	0	1	Data and check bits

Correction procedure: First, a counter is set to zero. Then, the check bits are checked one by one. If one of them produces the wrong parity, the number of the corresponding check bit is added to the counter. In the end, the counter points to the toggled bit.

Single bit errors can be corrected like this.

Why does the counter value reference the defective bit?

Bit 1 defective? yes: Bits 3,5,7,9 and 11 may be defective
Bit 2 defective? no: Bits 5 and 9 are left
Bit 4 defective? yes: Only bit 5 is left