Lecture on Sensor Networks

Communication in sensor networks

Problems of the MAC-layer in sensor networks (radio networks):

The following circumstances cause problems concerning energy efficiency:

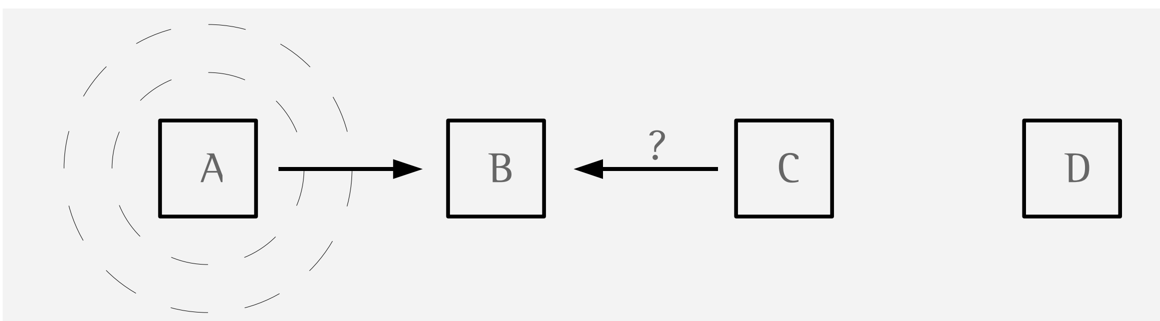
idle listening:	A node monitors the medium although no sender is active. Without special precautions it is unknown when the next sender will become active, so the monitoring cannot be omitted easily.
over emitting:	A node sends a message while the recipient is at least temporarily unable to listen.
overhearing:	A node receives a message (completely), which is addressed to someone else. Without special precautions the message has to be overheard to receive the following message.
collisions:	Two nodes transmit at the same time. Hence, the messages destroy each other on the recipient's side. At least two senders waste energy.

Communication in sensor networks

Problems of the MAC-layer in radio networks:

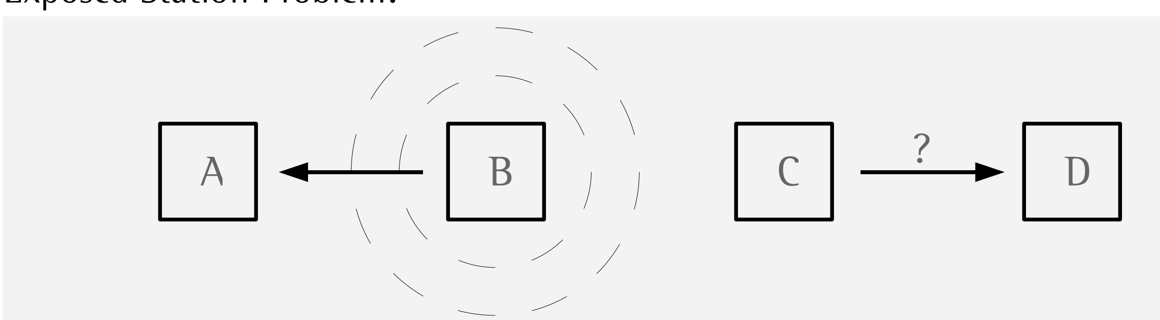
In contrast to cable-based channels the sending range of a station in a radio network is limited.
By transmitting a message a station can create collisions in a distance where it cannot detect them any more (hidden station problem). On the other hand, it can detect a collision locally that does not occur any more on the recipient's side (exposed station problem):

Hidden Station Problem:



C cannot hear that B is busy. If C sends, it causes a collision.

Exposed Station Problem:



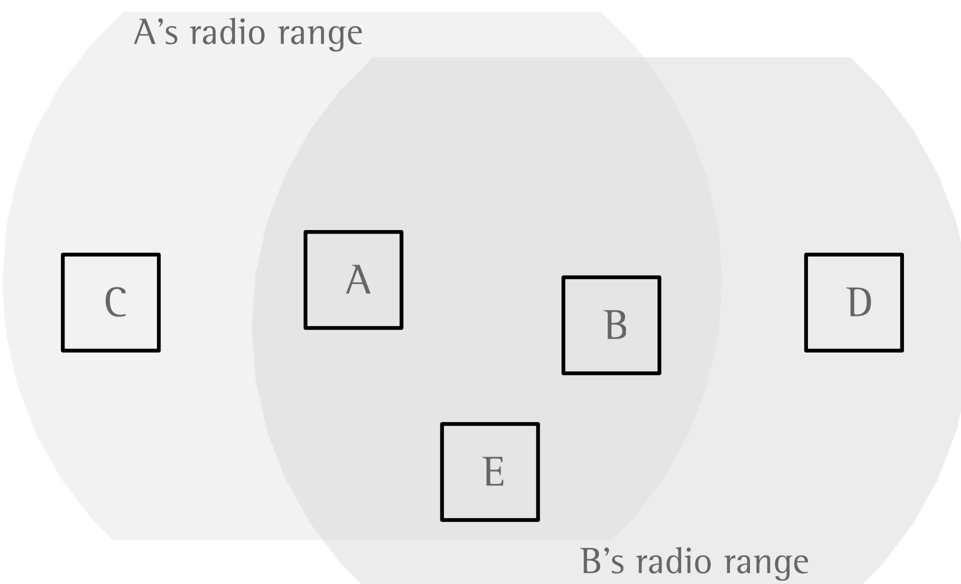
C wants to send to D, but thinks that the transmission will fail.

Communication in sensor networks

Problems of the MAC-layer in radio networks:

Difference between medium access in cable communication and radio transmission:
On a cable the conditions on the sender's side equal those on the recipient's side. The sender can derive from its local observations whether a transmission makes sense for the recipient. Generally, only the recipient's ability to listen undisturbed is of interest. In radio networks the remote conditions cannot be derived from the local observations.

Solution: The busy-detection does not happen „electronically" only (by listening to the channel), but has to be handled by a protocol.



Remark:
In reality, in addition to its transmitting range a sender also has a disturbance range, in which its packets are not understood any more, but still cause collisions. The circular resp. spherical area around a sender is a simplification, too. From interactions with the environment the real reception areas look different in most cases.

Communication in sensor networks

The RTS/CTS-Scheme:

Implemented in MACA (Medium Access with Collision Avoidance) and adopted in 802.11 to ensure an undisturbed reception:

(1) Sender A sends a "Request to Send" (RTS) to B. An RTS packet contains the recipient of the message and the length of the message to follow.

(2) If the RTS packet arrives on the recipient's side, the latter sends a "Clear to Send" (CTS). The CTS packet contains the length of the message to be expected soon copied from the RTS packet.

(3) After the original sender A receives the CTS packet, the message itself is transmitted.

Optional: (4) The recipient can use the checksum to verify whether the message was received (hopefully) correctly and send an acknowledgment (ACK).

Alternative: It can also be left to the higher transport layer to detect a missing message and to request it again (resp. to provoke a repeated transmission by not acknowledging it). But this takes remarkably longer and requires a new contention-phase (idle medium, RTS/CTS etc.).

Communication in sensor networks

The RTS/CTS-Scheme:

Ensuring an undisturbed reception:

What are the others doing?

(1) An RTS could collide with another RTS, e. g. if E also wanted to send. Neither of the senders receive a CTS, because their messages collided (possibly undetected by them) on the recipient's side. Both senders wait for a random amount of time before they assume that no CTS is to be expected. The node with the shorter timer will make the first attempt to retransmit, the other node detects this and waits.

(2) The RTS is e. g. also heard by C. But C realizes that it is not the recipient. C would not disturb B by sending a message, because they cannot hear each other anyways. Nevertheless, C has to be quiet until at least a possible CTS arrives safely at the sender's side. C would otherwise sabotage communication between A and B. But: C could send afterwards. In this case, A does no more understand its own message, but this has no effect on the reception at B. Yet an unsolved problem: C could send but it would not be able to receive anything (e.g. a CTS from someone else).

(3) D cannot hear A's RTS, but receives B's CTS, too. Because the length of the expected message is contained in the CTS packet, D knows that it has to be quiet for a certain amount of time.

Communication in sensor networks

The RTS/CTS-Scheme:

Ensuring an undisturbed reception:

How is the channel accessed in 802.11?

After a successful transmission, a new frame time begins. Nodes can express their intention to send. But they have to wait for the amount of time that an ACK needs. If the ACK does not arrive at the receiver's side, the last transmission has to be considered invalid. Afterwards, a contention phase with 31 time slots begins.

Every sender draws a random number out of the interval [0,31] and monitors the channel until the corresponding time slot appears. If the channel has been occupied before by another sender, the later sender defers its intention to send. Otherwise it „wins" the channel, unless another sender has the same number. In this case both RTS packets will interfere and a CTS is not to be expected.

Beyond that the time slots in the contention phase are doubled to 63, another failure leads to another doubling and so on. Hence, the protocol automatically adapts to different numbers of participants.

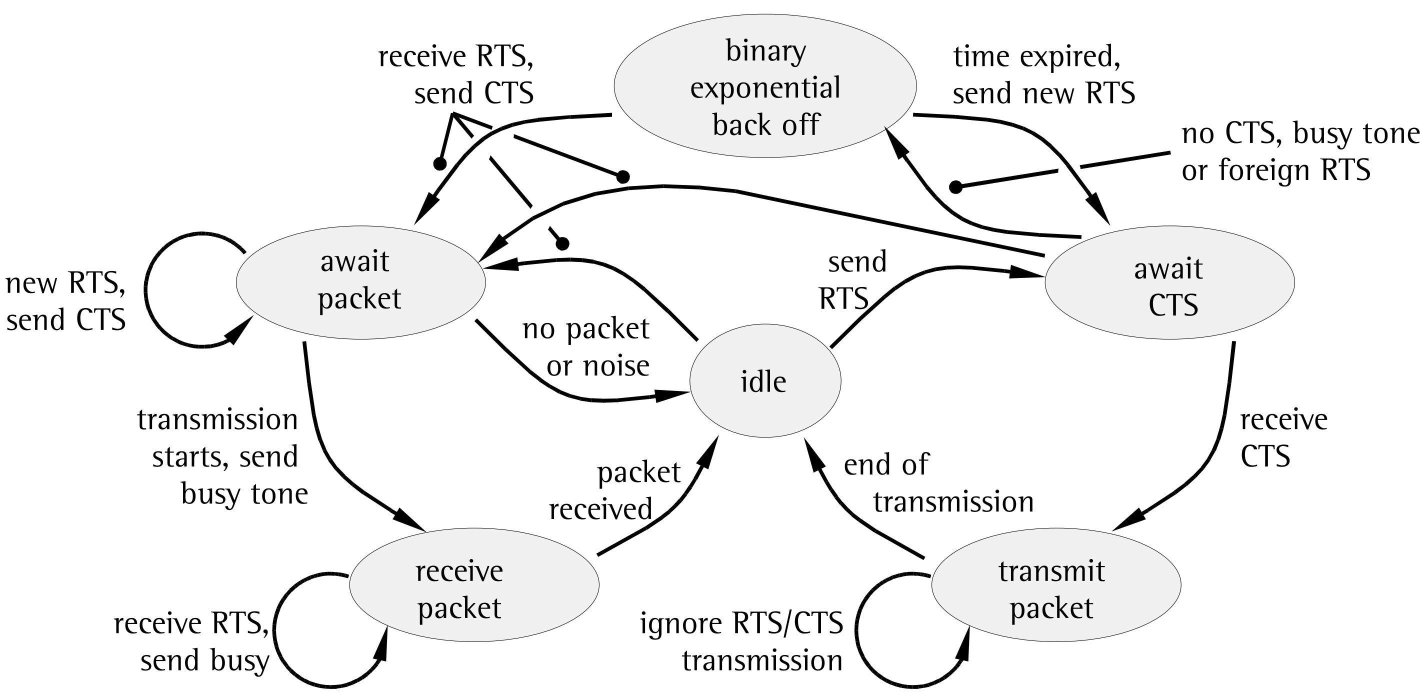
Why are discrete time slots used and not constantly distributed waiting periods?

Answer: Because switching from receiving to sending takes a certain amount of time. The time slot is chosen long enough to regard switching times.

Communication in sensor networks

Signalling channel as solution to the hidden terminal/exposed station problem:

PAMAS – Power Aware Multi-Access protocol with Signalling for Ad Hoc-Networks
by S. Singh and C. S. Raghavendra



Communication in sensor networks

Problems of the MAC-layer especially in radio networks:

S-MAC protocol: An adaptation for sensor networks

Principle: All nodes have a so-called schedule, which divides time into listen- and sleep-periods. Nodes try to join into groups as large as possible that follow this scheme of sleeping and listening periods.

How can the relation between listen- and sleep-periods be optimized?

Involved values: Maximum resulting data rate (specified by external events and sensors), maximum transmission rate, buffer size of the node, tolerable latency.

Communication in sensor networks

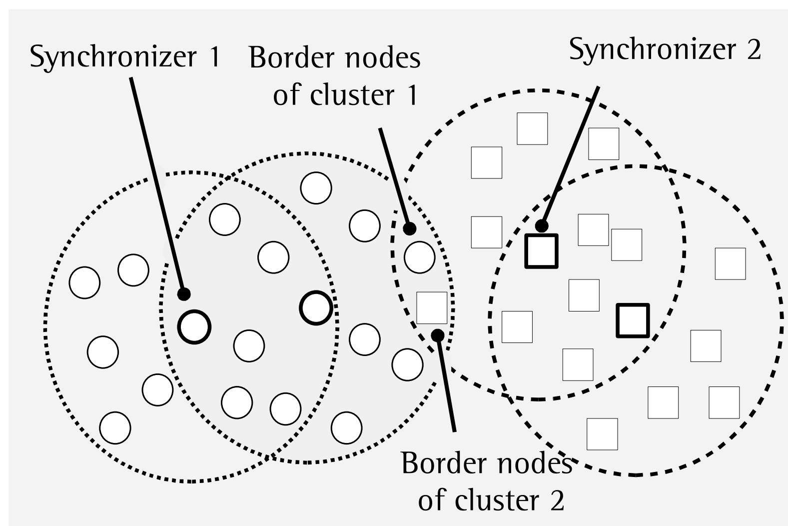
Problems of the MAC-layer especially in radio networks:

S-MAC protocol: An adaptation for sensor networks

Listen period:
As short as possible, because energy-consuming. But long enough for the max. resulting data rate. Otherwise data have to be rejected at the end of the sending period.

Sleep period:
As long as possible. But meanwhile data has to be stored. The buffer size limits the length of the sleeping period.

Total frame length:
= Listen+Sleep:
Has to guarantee the max. latency.
On the other hand:
Turning the RF-unit on and off costs time and energy, too.



Communication in sensor networks

Problems of the MAC-layer especially in radio networks:

S-MAC protocol: An adaptation for sensor networks

Initialization phase

Synchronizer: After the nodes are dropped, they wait for a random amount of time and listen whether one of the other nodes sends the first message. The node with the shortest timer sends a SYNC-message. It contains the information that the node will go to sleep in t seconds. The first node is also called a synchronizer, all others are followers.

Follower: All nodes in the radio range of the Synchronizer receive the initial SYNC-message and decide to go to sleep at the same time. They wait for a random time d and announce after the d seconds that they are going to sleep in t-d time units, like the synchronizer. Without the waiting period all nodes would transmit at the same time and would certainly create collisions. By the repeated SYNCs other nodes in the vicinity are also integrated in the virtual cluster, if they do not already follow another synchronizer.

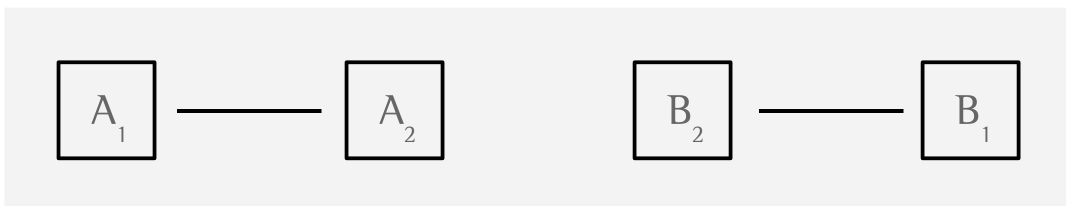
All recipients: Every SYNC-packet also tells all recipients which nodes are in sending range and which schedule they belong to. This will often, but not necessarily, be the recipient's own schedule. The knowledge of the „other" listening periods is important for addressing the neighborhood.

Communication in sensor networks

Problems of the MAC-layer especially in radio networks:

S-MAC protocol: An adaptation for sensor networks

What happens to nodes within the same sending range, who already belong to different virtual clusters, like A and B in the example below?



Strategy 1 (Receiving at the time of the others): A node records both schedules.
Advantage: It can participate in both clusters and works as a transponder which can transport messages from cluster to cluster, especially broadcast messages.
Disadvantage: It has to listen for a time twice as long as normal nodes. Although it has an important function for connecting the clusters, it will fail much sooner than other nodes.

Strategy 2 (Sending at the time of the others): The node sticks to the schedule it took on first, however it saves the nodes of other clusters in sending range together with their schedules. Hence, it can reach the other nodes if required (by rising early).
Disadvantage: Broadcast messages can not be delivered across cluster borders safely.

Communication in sensor networks

Problems of the MAC-layer especially in radio networks:

S-MAC protocol: An adaptation for sensor networks

Exceptional situation:

From time to time every node has to listen to the channel for a full frame length. Why?

Because new nodes which have already synchronized with other neighbors could be added. These new neighbors can only be safely detected by listening for a complete frame length. It is also possible that the new neighbors connect two sub-networks (so-called partitions) that did not know each other before.

When should a node transmit a broadcast-message for everyone (using strategy 1)?

(a) at the time of the own schedule
(b) at the time of the neighboring cluster's schedule

Answer: Always at the time of the own schedule. Because at least with strategy 1 it is certain that everyone is listening, even those not belonging to the own cluster. If the node sent at the time of a neighboring cluster's schedule there would probably be sleeping nodes within the sending range.

But with strategy 1 no node in the sender's range would sleep at the time of the sender's schedule. Why? Because the latter announced its own sending time and all nodes could register it.

Communication in sensor networks

Problems of the MAC-layer especially in radio networks:

S-MAC protocol: An adaptation for sensor networks

Time synchronization

The listening phase is divided into two sections, a SYNC- and an RTS-phase. The SYNC-phase is used to readjust the timers to avoid a long-term drift.

In larger intervals, every node sends a SYNC-packet like the one known from the initialization phase. This SYNC has the task to (1) re-synchronize other nodes and (2) give newly joining nodes the chance to integrate into a cluster. Access to the medium is in both cases negotiated by using a random waiting period. The node that occupies the medium first, „wins“ it. In the SYNC-phase the first node is also the one that may announce its time.

The random syncing scheme does not guarantee that all nodes adjust their timers in the same way but it does prevent very large clock drifts.

Communication in sensor networks

Problems of the MAC-layer especially in radio networks:

S-MAC protocol: An adaptation for sensor networks

Collision avoidance and avoidance of overhearing

In principle collision avoidance works like the RTS/CTS scheme. But a node should go to sleep whenever a packet for a foreign addressee is sent. This is possible because both the RTS and CTS packets contain the length of the packet to expect.

If the node wants to execute other tasks, at least the RF unit should be turned off. A counter, the so-called Network Allocation Vector NAV, with the waiting period has then to be set. Every time unit the NAV is decremented by one (e.g. interrupt controlled). The node is allowed to send only when NAV == 0. Otherwise it is known that the medium is occupied anyways, even without „physically“ testing it. This procedure is also called virtual carrier sense.

At the end of the virtual carrier sense the medium is in fact monitored and after a random waiting period an occupation attempt is started (using RTS/CTS). Otherwise the node has to wait again for the length of the incoming packet.

Conclusion: Avoidance of overhearing by virtual carrier sense
Avoidance of collisions by a random timer and physical carrier sense in combination with the RTS/CTS scheme

Communication in sensor networks

Problems of the MAC-layer especially in radio networks:

Improvement T-MAC:

In order to handle the largest amount of data that can possibly occur, the active periods of S-MAC are comparatively long. Apart from this case, the unnecessarily long waiting periods have to be accepted. An intuitive solution would be...

...to define a timeout TA. TA should be chosen a bit longer than the longest max. random waiting period for channel access (backoff-time). If the TA expires, there may be no new desire to communicate. As a consequence all nodes may go to sleep.

Problem:
The approach works only within a 1-hop neighborhood. Other nodes, situated in the proximity of the communication pair could have sending intentions at the same time .

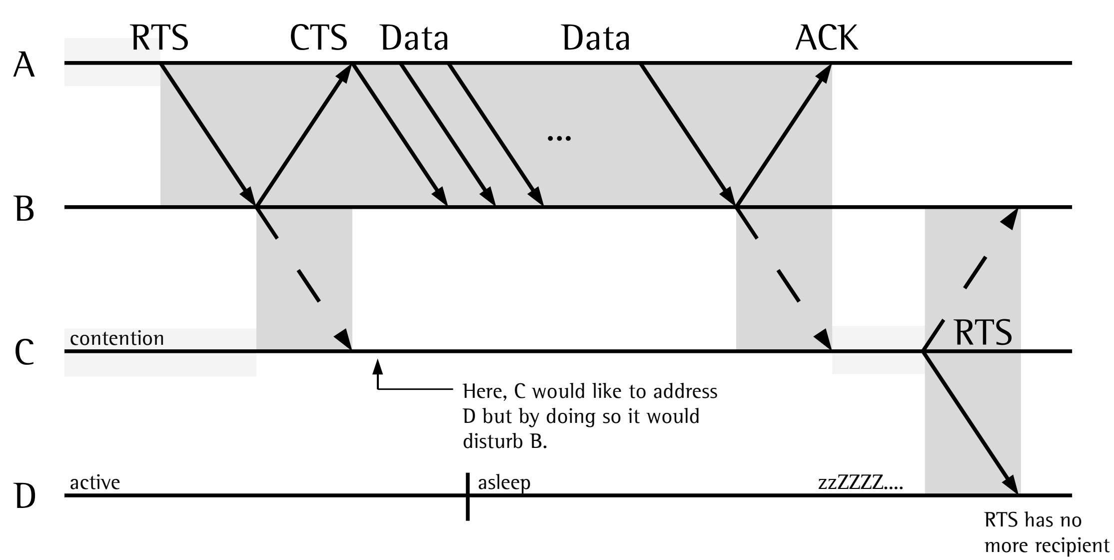
Communication in sensor networks

Problems of the MAC-layer especially in radio networks:

Improvement T-MAC:

Node C is in the direct neighborhood of the sending pair A/B and would like to address more distant nodes, like D. However, C can not inform D about the intended communication, because it would disturb the communication between A and B.

Node D does not hear any communication at all and goes to sleep. Therefore it can not be reached by C anymore, even after A and B have finished their communication.



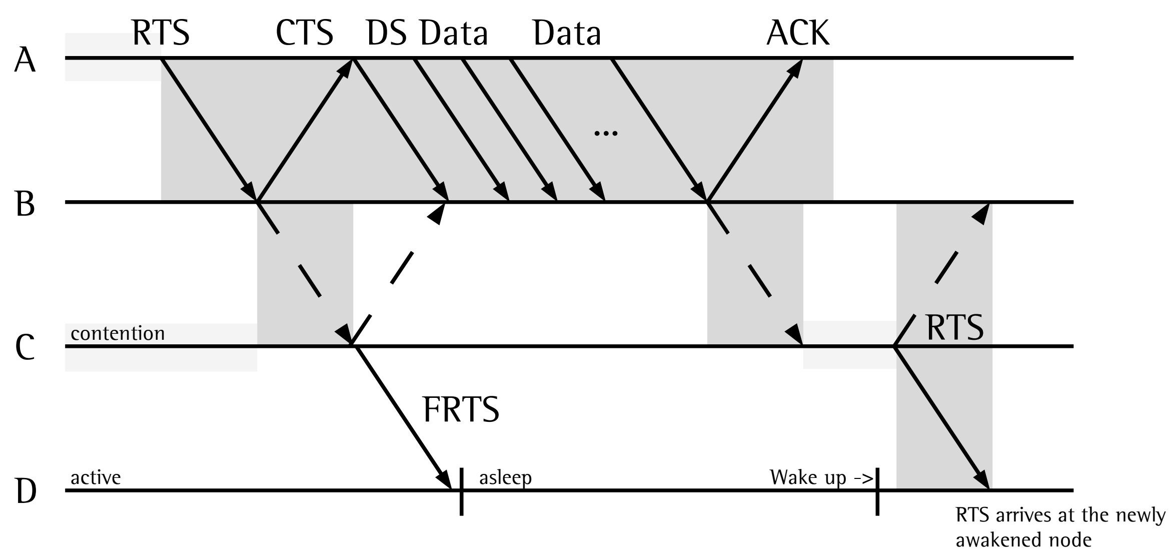
Note: The problem arises in particular with asymmetric communication with one predominant direction (from the source(s) to a sink).

Communication in sensor networks

Problems of the MAC-layer especially in radio networks:

Improvement T-MAC:

Solution 1: The prevented sender C may announce its future sending intention after all. In this example node C overhears B's CTS and sends its own FRTS (Future Ready-To-Send). In the normal case, this would lead to a collision with A's data and garble B's data packets. In this case, A knows that an FRTS could possibly follow and sends a DS (Data-Send). The DS has no deeper meaning and can hence be destroyed without causing problems. D can sleep for the time contained in the FRTS (copied by C from B's CTS).



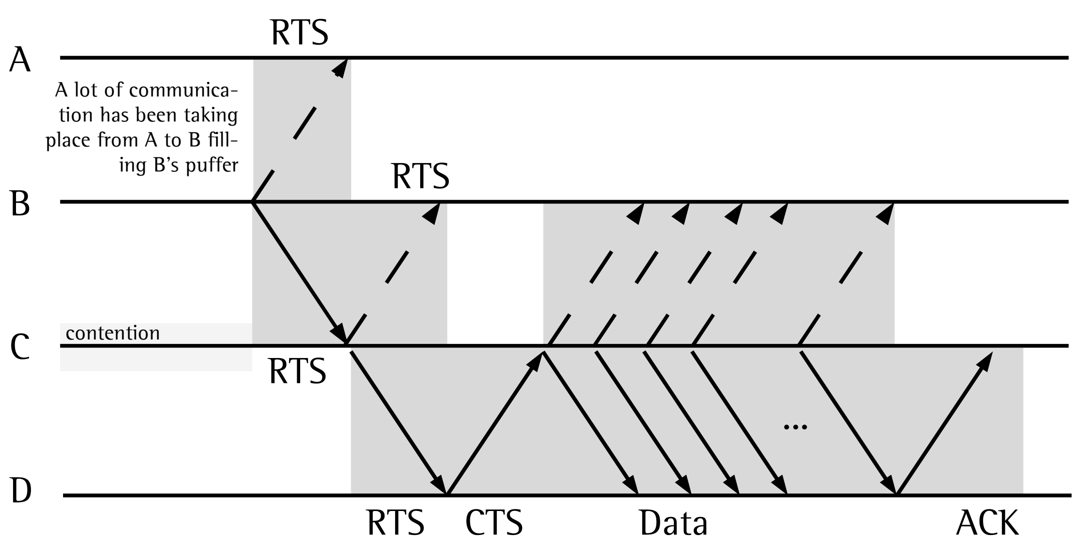
Communication in sensor networks

Problems of the MAC-layer especially in radio networks:

Improvement T-MAC:

Solution 2 (full-buffer priority): We accept that nodes in a further distance go to sleep. All mutual communication between neighboring nodes is done in one pass (if possible). If (in a one-way communication) a receiver's buffer is in danger of overflowing, it simply stops accepting data. Hence, nodes ignore further RTS requests and try to win the medium themselves by issuing own RTS to get rid of the full buffer. Which problems may occur?

The approach only works if the transmission is addressed to a sink. Otherwise the nodes may mutually ignore one another and create dead-lock situations.



Communication in sensor networks

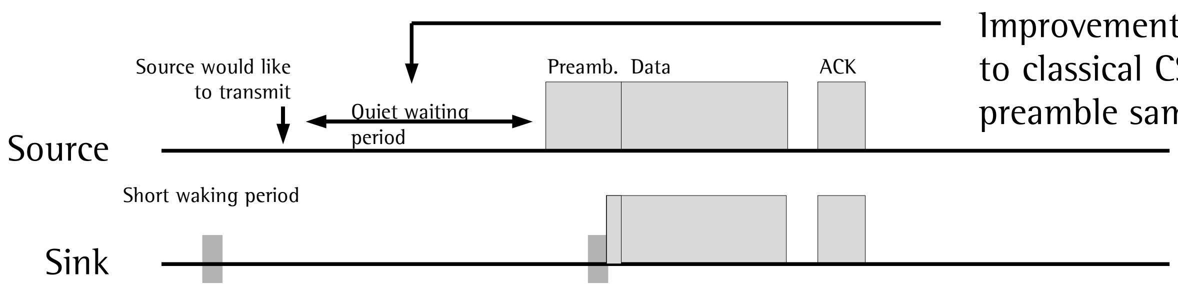
Problems of the MAC-layer especially in radio networks:

WiseMAC basically works like CSMA with Preamble Sampling.

Problems with Preamble Sampling

(a) The long preamble time a sender needs, only for occupying the channel without transmitting relevant information.

(b) In the preamble phase the recipient is so to speak „picked up“ to eventually be available for the real data transmission. Unfortunately in the meantime all other recipients are also picked up, but without any benefit. Although they can „oversleep“ the data frame, they have to wait at least for an RTS resp. CTS. Thus, lots of energy is wasted.



Improvement compared to classical CSMA with preamble sampling

*letztlich, nicht eventuell

Communication in sensor networks

Problems of the MAC-layer especially in radio networks:

WiseMAC

Improvement of WiseMAC: Every node remembers its neighbors' waking moment (relative to its own). Only shortly before the recipient awakes the channel is occupied by a sender.

The waking moment of a node is contained as additional information in its ACK packets (possibly piggy-packed onto the RTS or CTS as well). The ACK packets can also be used by accidentally woken, but uninvolved nodes to apply a correction of their knowledge about the waking moment of the ACK-sender. In contrast to S- and T-MAC every node has its own random waking moment in WiseMAC.

Synchronization problem: If sender and recipient do not hear from each other for some time their clocks may be out of sync. The longer this time, the sooner the sender's preamble-time has to start before the expected awakening of the recipient.

What's the upper limit for the preamble-time?

Communication in sensor networks

Problems of the MAC-layer especially in radio networks:

WiseMAC

Collision problem: If several senders decide to address a data frame to the same recipient, the preamble and even more severe the following data frames collide in any case. Thus, the preamble length has to be increased by a random amount of time. Again, a multiple of the RF unit's switching time is chosen for this, as with 802.11. That way a single sender probably wins the contention phase created by the slightly differing beginnings of the preamble. A later sender hears an ongoing preamble and defers its communication.

Advantages of WiseMAC: Low overhead for senders due to short preambles.

Disadvantages: As not all nodes listen all the time, broadcast messages are not possible any longer (solution: The sender switches to long preamble times).

In any case, uninvolved nodes wake up during the data frame. Unfortunately the data then can not be overheard as easily any longer, because the packet's length is unknown to accidentally involved nodes. On average, 50% of a data frame has to be overheard, if it has been caught in the middle.

Communication in sensor networks

AMRIS: Multicast Protocol for Ad hoc Wireless Networks

Idea:
The purpose of the AMRIS protocol is to create a multicast-tree in a wireless ad-hoc network with many recipients that want to join a certain session. The protocol is especially stable for moving, vanishing and newly appearing nodes.

Application:
In the context of sensor networks it is rather rare that a source sends large amounts of data to the nodes. But the protocol is also appropriate to build a distribution tree for requests. Nodes with different equipments may join different groups, e.g. one for each sensor type (temperature, light, humidity etc.) or they may join a group which aggregates nodes from the same location.

The tree can be used for an efficient distribution of requests. Furthermore it can be used to send collected data back to the sink (often at the same time the source of the request) on an optimal way.

Communication in sensor networks

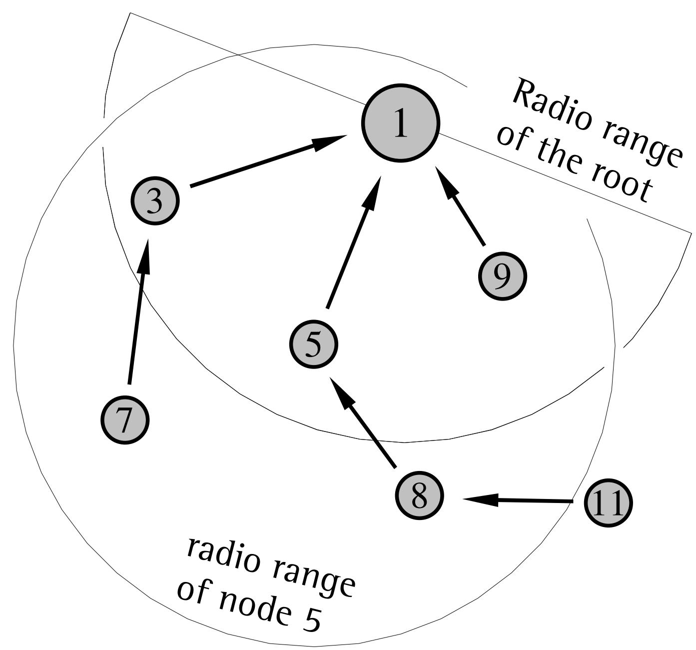
AMRIS: Multicast Protocol for Ad hoc Wireless Networks

Task: A source wants to send data to potentially many recipients in a wireless ad-hoc network (task of layer 3)

Requirement: It should not matter whether the underlying layer 2 (link layer) supports unicast only or broadcast as well.

2 phases of the protocol:

(1) Initialization phase
(2) Join phase
(3) Maintenance phase



Remark:
Broadcast: Message to everyone
Multicast: Message to a group

Communication in sensor networks

AMRIS: Multicast Protocol for Ad hoc Wireless Networks

Initialization phase

The source sends initial messages: NEW_SESSION(sessionID, msmID, routingMetrics)

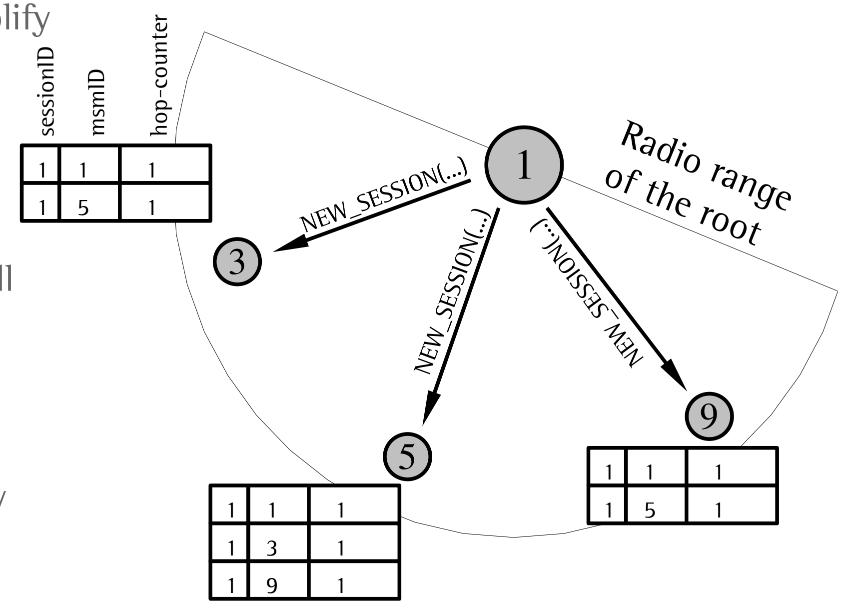
Recipients wait for a random period. The one with the shortest waiting period will repeat the NEW_SESSION-message. It replaces the msmID (multicast membership ID) by a larger, but not successive value.

The gap between the IDs is needed to simplify the later inclusion of additional nodes.

As a routing metrics we can use e.g. the number of hops to the root node.

Every participant remembers a table with all known neighbors. The entries of the table have an expiration date, for the case that a neighbor is not heard for a longer period.

Double msmIDs can only be avoided locally due to the hidden-station problem.



Communication in sensor networks

AMRIS: Multicast Protocol for Ad hoc Wireless Networks

Initialization phase

Every node is responsible itself for the choice of its parent node. It chooses a node with a good metrics, hence e.g. a short connection to the root node (the source of the later requests).

After the node's timer expires it announces a NEW_SESSION-message itself with the already mentioned higher msmID of the best parent node it has found so far. There is no guarantee that more suitable parents could emerge later. These are – like all nodes – added to the neighborhood table, but the parent is <u>not</u> changed any more after the NEW_SESSION message has been issued. Otherwise the recently sent NEW_SESSION-message would contain wrong information (because it has to conform to the parent's ID, plus an offset).

Up to this moment, no nodes joined the multicast-session. Only the neighborhood information was exchanged so far.

Communication in sensor networks

AMRIS: Multicast Protocol for Ad hoc Wireless Networks

Joining a group

A node X wants to join a group.

Its neighborhood table contains fellow nodes with larger and smaller msmIDs. The smaller IDs tend to be closer to the root node (but not necessarily). So X sorts the neighbors in order of increasing msmIDs and sends JOIN_REQ messages to potential parent nodes Y. If the number of hops is the chosen routing metrics only those nodes with a smaller number of hops (to the root) should be addressed.

Reaction of a potential parent node Y:

Y checks whether it is a member of the requested group already. If so, it sends a JOIN_ACK immediately. Now X belongs to the group and will receive future requests that are directed to this group.

If Y is not yet member of the group it acts as if it wanted to join itself. If a group-participant is found on the way up to the root the recursion will descend down to the initial requester. Another branch of the tree has now established.

Note that dead ends can emerge if nodes move away or stop working. In such a case the way via parent nodes to the root node might be blocked.

Communication in sensor networks

AMRIS: Multicast Protocol for Ad hoc Wireless Networks

Maintenance-Phase:

After the initialization nodes send so-called beacons regularly which contain the well known triple (sessionID,msmID,hops). Neighbors will hear these beacons and update their table entries accordingly.

In particular every table entry contains an expiry date. It is set to zero each time a beacon arrives for a neighbor but is incremented each time the beacon is missing. After three missing beacons in succession a node concludes that it has missed its neighbor and deletes the entry from the neighborhood table.

Disconnected branches in the tree:

Table updates as described above are common in AMRIS unless a node's father is lost. In this case the node has to find another link under all circumstances in order to reconnect itself and all of its children. This is called a Branch Reconstruction (BR) which has two phases BR1 and BR2.

BR1:

Just like in the initial joining phase a JOIN_REQ is sent to all possible parent nodes in turn. On success a father acknowledges the request with a JOIN_ACK. Otherwise it has to search its own environment for possible parents which can satisfy the request. Nodes which only serve the request of their children are called intermediate nodes.

Communication in sensor networks

AMRIS: Multicast Protocol for Ad hoc Wireless Networks

Maintenance-Phase: BR1 (continued)

If the reconstruction BR1 was not successful JOIN_NACKs are sent as negative acknowledgments on the reverse path back to the requester. The initiator (which is the only non-intermediate node on the path) gets to know the result at last. It will then iterate over other possible parent nodes (with a smaller number of hops to the root). If all requests have been unsuccessful the second branch reconstruction phase BR2 is started which is the most expensive phase of AMRIS.

BR2:

So far only unicast JOIN_REQ messages were sent. Now requests will be sent as broadcast messages directed to all neighbors at the same time which includes nodes with a smaller and larger number of hops. If the link layer does not support broadcast (like e.g. in WiseMAC) messages have to be sent individually to each neighbor. A node Y receives a broadcast BJOIN_REQ. In case it is already member of the group it will answer with a JOIN_ACK as a unicast response. However, this time the JOIN_ACK does not yet establish a branch of the tree, because the initiator X will possibly get a larger number of positive answers and will choose the best among all of them. Only the optimal partner will get a JOIN_CONF which eventually establishes the branch of the tree. Every broadcast BJOIN_REQ includes a TTL (Time-To-Live) counter which is decremented by one in each hop. This prevents the flood from reaching all parts of the network. Especially those far beyond the root are not likely to be helpful for the reconstruction.

Communication in sensor networks

AMRIS: Multicast Protocol for Ad hoc Wireless Networks

Maintenance-Phase: BR2 (continued)

Finally a node which receives a BJOIN_REQ and is not member of the group broadcasts the request further itself.

General remarks:

The BR2 phase can also be used (in the same fashion) if joining a group via one of the parent nodes was not successful.

Newly deployed nodes which do not yet have an msmID will listen to the beacons of the neighbors for a while before choosing a parent and announcing their chosen msmID. The reason why nodes do not use successive msmID in the beginning is due to the fact that these gaps can now be used by the newly deployed nodes.

The protocol is particularly stable in scenarios with moving nodes. In the context of sensor networks it deals well with vanishing and newly deployed nodes – however, at the expense of having to send beacons. Note that beacons can only be combined with a few energy-efficient MAC layer protocols.