Exercise Sensor Networks

Lecture 7: MAC in radio networks

Exercise 7.1: Wise-MAC

A sender wants to transmit a message to a receiver using WiseMAC. Therefor it emits a preamble prior to the estimated wake-up time of the receiver and then adds the message.

a) In contrast to Aloha with preamble sampling a sender using WiseMAC knows when the receiver will wake up. What is the preamble good for in WiseMAC?

Solution:

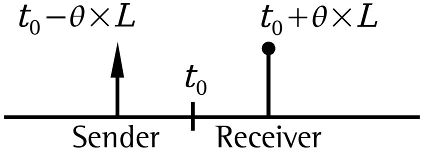
The purpose of the preamble is less focused on waking up the node as the data packet could almost be sent immediately at the known wakeup time of the recipient. A short preamble is still useful for synchronizing the beginning of the data chunk because the clocks of sender and receiver could have drifted apart. Another more important property of the small preamble is its use as contention phase if more than one sender wants to address a receiver.

b) The type of clocks being used for specific sensor nodes exhibit a maximum inaccuracy of theta time units per time unit (theta can be considered to be a small fraction, e.g., in the degree of magnitude of 10^{-5} seconds). The authors of Wise-MAC claim that after L time units a sender has to extend its preamble up to 4 x theta x L. Explain why. When does a senders have to start sending the preamble if it expects the receiver to wake up at time t_0 and if the receiver was silent for L time units?

(continued)

Exercise Sensor Networks

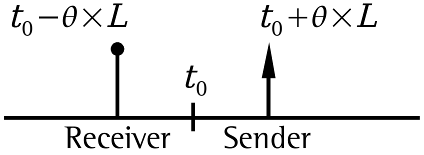
Lecture 7: MAC in radio networks



Exercise 7.1: WiseMAC

Solution:

Case sender early, receiver late: Sender (arrow) was too fast – in fact as fast as possible so after L time units the inaccuracy accumulates to (L x theta) in the worst case. Though the sender thinks its measured time is t_0 it actually is t_0-(L x theta). The receiver was in contrast as slow as possible and advanced (L x theta) time units further than t_0. If the sender is aware that this case can happen it has to send at least (2 x L x theta) time units to reach the receiver. If this scenario was reality the sender would reach the receiver in the very last moment.



Case sender late, receiver early: This case is very similar to the first one, however the roles between sender and receiver are swapped. If the sender was aware of this case it knew that it would have started its preamble much too late. In order to reach the wakeup time of the receiver it would have had to go back (L x theta) to the true point of time t_0 (to account for the inaccuracy of its own clock) and another (L x theta) to account for the inaccuracy of the receiver's clock which is too fast in this case (note that a fast clock means that a node starts of listen or to send too early while a slow clock causes a note to wait too long before taking action).

Obviously, the sender can not tell whether the first or the second scenario actually occurs. So it should account for both of them at the same time. This means that it has to go back (2 x L x theta) to start as shown in the second case. Intuitively speaking we could say that the sender assumed to be much to fast and the receiver much too slow. But if both clocks were synchronous the sender would have to send 2 x L x theta to reach the receiver in the last moment. Even worse if the sender had been to fast and the receiver much to slow it would even have had to continue the preamble (4 x theta x L) time units.

Exercise Sensor Networks

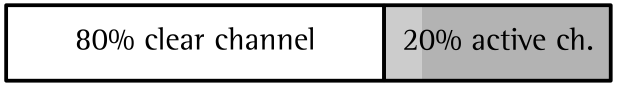
Lecture 7: MAC in radio networks

Exercise 7.1: WiseMAC

A sender wants to transmit a message to a receiver using WiseMAC. Therefor it emits a preamble prior to the estimated wakeup time of the receiver and then adds the message.

c) We consider a channel which is clear at about 80% of the time and active for the rest. The occupied 20% are further subdivided into 10% preamble time and 90% time for the actual data. How long does a node have to listen who is i) the receiver of a message all the time or who is ii) always uninvolved (not addressed by a sender)? Short wake up times are not considered and we assume that the ID of the receiver is included into the message (actual data transmission phase) at the very beginning.

Solution:



Addressed nodes wakeup in the preamble phase which is no coincidence but which is planned by the sender. On average 50% of the preamble has to be overheard so that nodes are awake 0.2x0.1/2. In addition the whole data phase has to be heard of course so the addressed node listens in total:

$0.2 \times 0.1/2 + 0.2 \times 0.90 = 0.19$

An uninvolved node wakes up in the active phase with a probability of 20% whereas 10% of the active phase is preamble time. Again 50% of the preamble time has to be overheard totaling to 0.2x0.1/2. The data phase can be omitted almost entirely as the nodes realized based on the ID in the packet header that is was not addressed and goes to sleep again. With an complementary 90% the node wakes up in the data phase and has to overhear 50% of it on average. The result for the uninvolved node is

$0.2 \times 0.1/2 + 0.2 \times 0.9/2 = 0.1$

Final result: The uninvolved node is still active half as long as compared to the one being addressed.

Exercise Sensor Networks

Lecture 7: MAC in radio networks

Exercise 7.2: AMRIS protocol

a) Is it possible that an msmID is used more than once in AMRIS. Why and when does it happen?

Solution:

Equal IDs can occur if stations do not hear one another due to the hidden station problem. They may even choose the same parent node not knowing about a brother node at the same level which chose the same ID. The probability of duplicate IDs in increasing with increasing distance from the root node.

b) Is it possible to address a particular node from the root even though msmIDs are not pairwise different? And is it possible for every node to address the root (we don't consider packet loss or node failure)?

Solution:

AMRIS is not a routing protocol for forwarding information for a particular node. It is used to deliver information to a group. So if a node has previously subscribed to a group it will get the information addressed to the group. The purpose of the ID is to build up a neighborhood table and to address the neighbors locally.

Note that in the worst case a nodes may have two neighbors with the same ID!

AMRIS is however suitable for finding the way from a node back to the root because everyone chose a unique parent node.

Exercise Sensor Networks

Lecture 7: MAC in radio networks

Exercise 7.3: SMACS

a) How does the SMACS protocol try to avoid collisions? What is the difference with regard to allocating a communication channel compared to the approaches we got to know to far?

Solution:

The protocol avoids collisions by establishing communication links in advance. These links can either be found or they will fail. Other contention based channel allocation approaches negotiate channel access repeatedly ad-hoc. Another major difference of SMACS is that it extends the space for communication by using multiple frequencies.

Exercise Sensor Networks

Lecture 7: MAC in radio networks

Exercise 7.3: SMACS

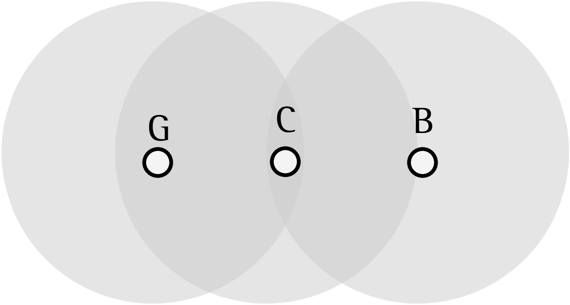
b) How does SMACS solve the hidden/exposed station problem? How can collisions still happen?

Solution:

A unique combination of a time slot and a frequency is defined for each communication partner, one for each direction. The combination of time and frequency will make collisions unlikely. Because of the larger number of frequencies several station can talk to one another at the same time within the same proximity. However, collisions can still happen.

Example:

Node C invites other local nodes to agree on a common slot. Node B responds and is chosen like shown on the lecture slides. Then B is allowed to find a free slot which coincides with its own schedule and the one of C.



Though this might work there is no guarantee that a node G unknown to B but in the neighborhood of C uses the same frequency at the same time slots as C and B. This is a typical instance of the hidden station problem.

Exercise Sensor Networks

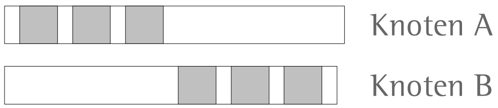
Lecture 7: MAC in radio networks

Exercise 7.3: SMACS

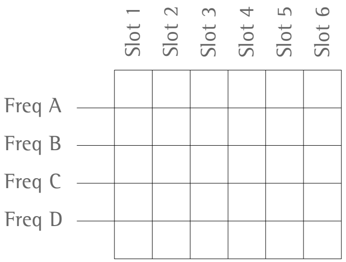
c) If two clusters meet it can happen that one cluster can not connect the other. How can this happen and what is the rare resource? Design an example in which one cluster in unable to connect another one.

Solution:

In the example on the right 50% of node A and B's slots are free but none of them coincides so that no communication can take place.



The combination of slots and frequencies extends the possibilities for communication significantly but in all 1-to-n relationships the limited number of time slots on the 1-side is still the sparse resource.



d) Can bottle necks identified in c) be resolved by sorting the schedules in another way? If yes how, if not why?

Solution:

A permutation of slots on one side is no option as this would only shift the problem to the peer node at the other side of the swapped slot. No good local solution is known at this time. Only a global optimization could mitigate the problem. E.g., a linear program could solve the problem.