

Ad hoc and Sensor Networks

Chapter 5: Medium access control protocols

Holger Karl



Computer Networks Group
Universität Paderborn

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



SS 05

Ad hoc & sensor networks - Ch 5: MAC protocols

2

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!



SS 05

Ad hoc & sensor networks - Ch 5: MAC protocols

3



SS 05

Ad hoc & sensor networks - Ch 5: MAC protocols

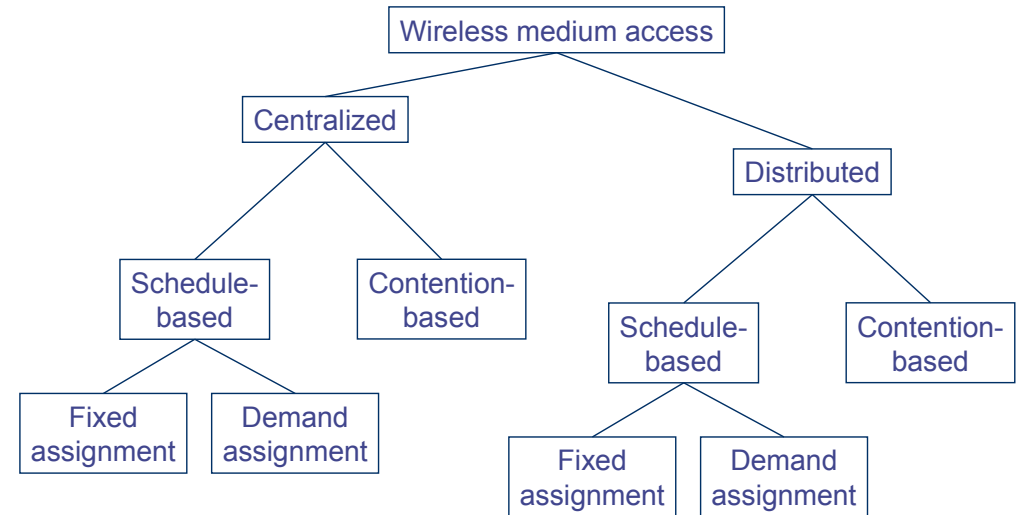
4

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

! Usually, distributed medium access is considered



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
 - Needed: time synchronization!
- **Contention-based** protocols
 - 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

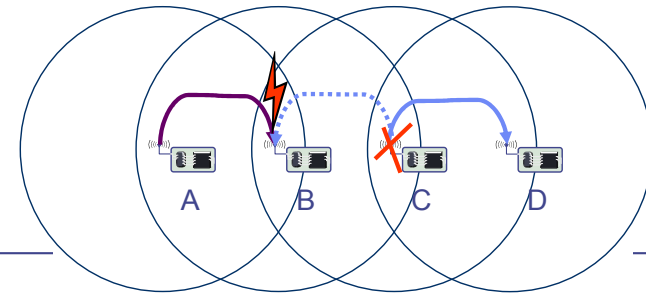
- Principal options and difficulties
- **Contention-based protocols**
 - MACA
 - S-MAC, T-MAC
 - Preamble sampling, B-MAC
 - PAMAS
- Schedule-based protocols
- IEEE 802.15.4



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)

Hidden terminal scenario:



Also:
recall
exposed
terminal
scenario



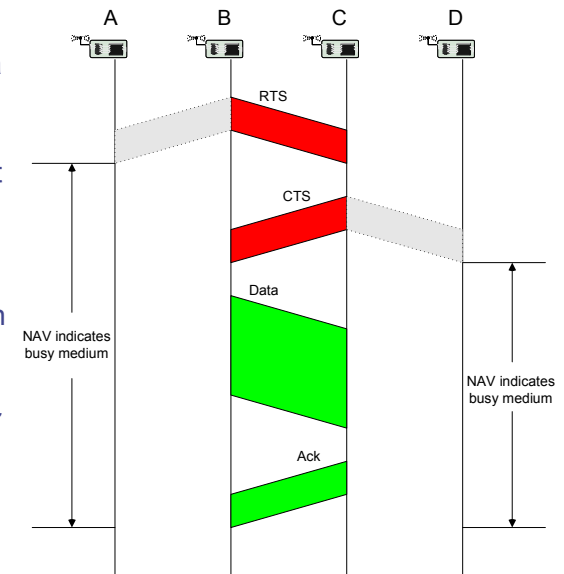
Main options to shut up senders

- 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



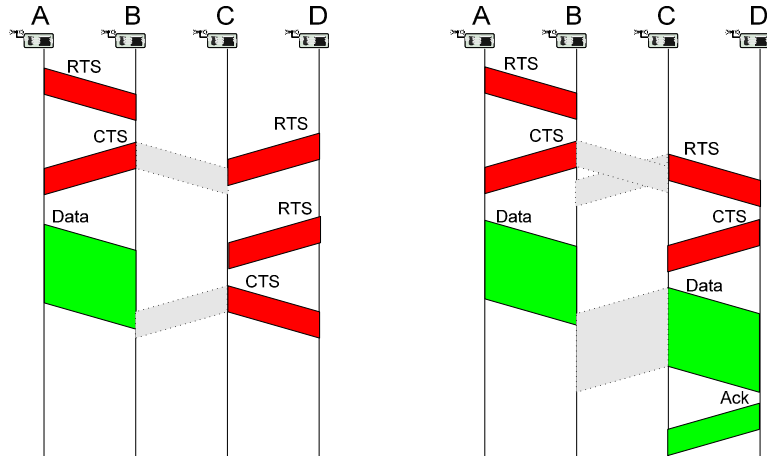
Receiver informs interferers before transmission – MACA

- 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
 - Store this information in a **Network Allocation Vector**
- B sends, C acks
- ! **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:



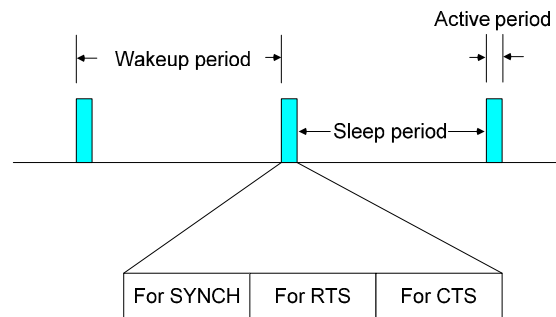
MACA Problem: Idle listening

- 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?
 - ...



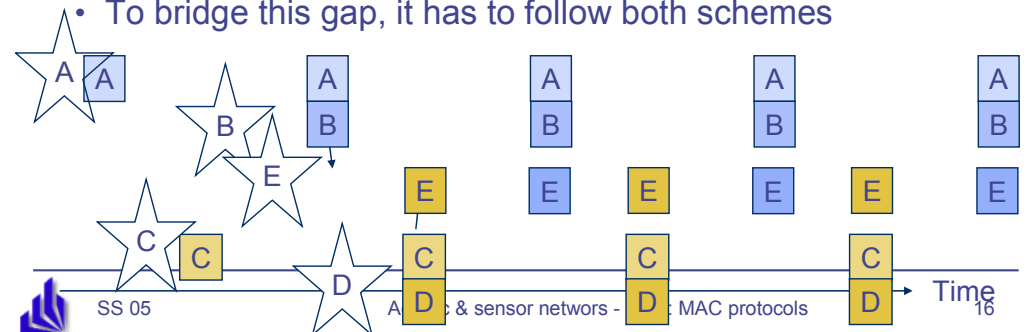
Sensor-MAC (S-MAC)

- MACA's idle listening is particularly unsuitable if average data 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



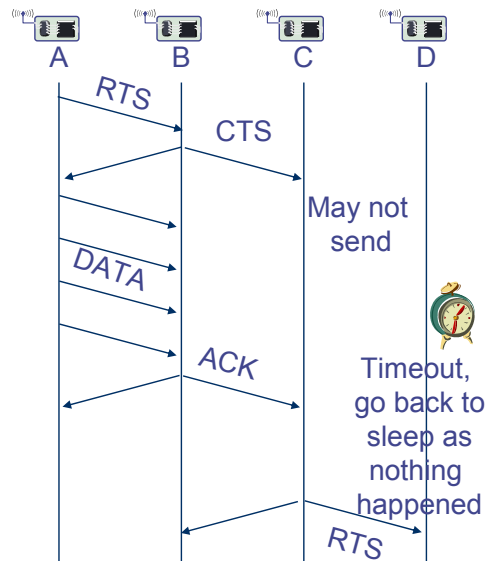
S-MAC synchronized islands

- 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



Timeout-MAC (T-MAC)

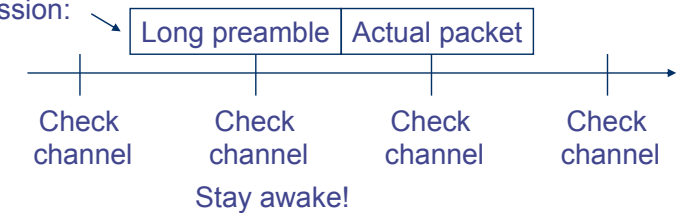
- In S-MAC, active period is of constant length
- What if no traffic actually happens?
 - Nodes stay awake needlessly long
- Idea: Prematurely go back to sleep mode when no traffic has happened for a certain time (=timeout) ! T-MAC
 - Adaptive duty cycle!
- One ensuing problem: Early sleeping
 - C wants to send to D, but is hindered by transmission A! B
 - Two solutions exist – homework!



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



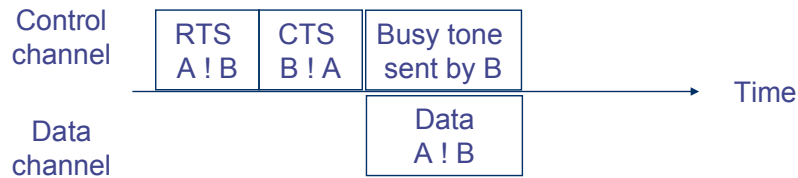
B-MAC II

- 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



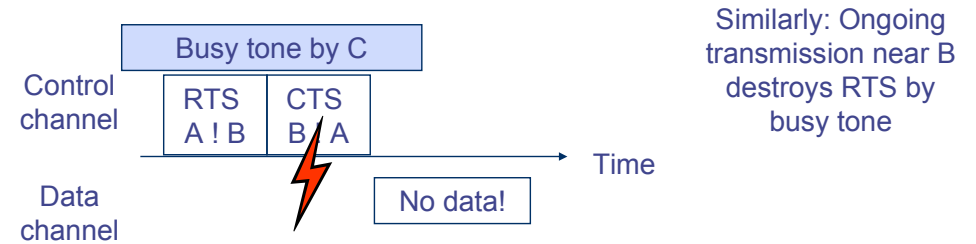
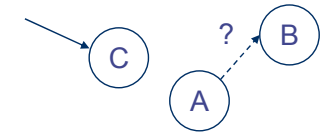
Power Aware Multiaccess with Signaling – PAMAS

- 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



PAMAS – Already ongoing transmission

- Suppose a node C in vicinity of A is already receiving a packet when A initiates RTS
- Procedure
 - A sends RTS to B
 - C is sending busy tone (as it receives data)
 - CTS and busy tone collide, A receives no CTS, does not send data



Overview

- Principal options and difficulties
- Contention-based protocols
- **Schedule-based protocols**
 - LEACH
 - SMACS
 - TRAMA
- IEEE 802.15.4

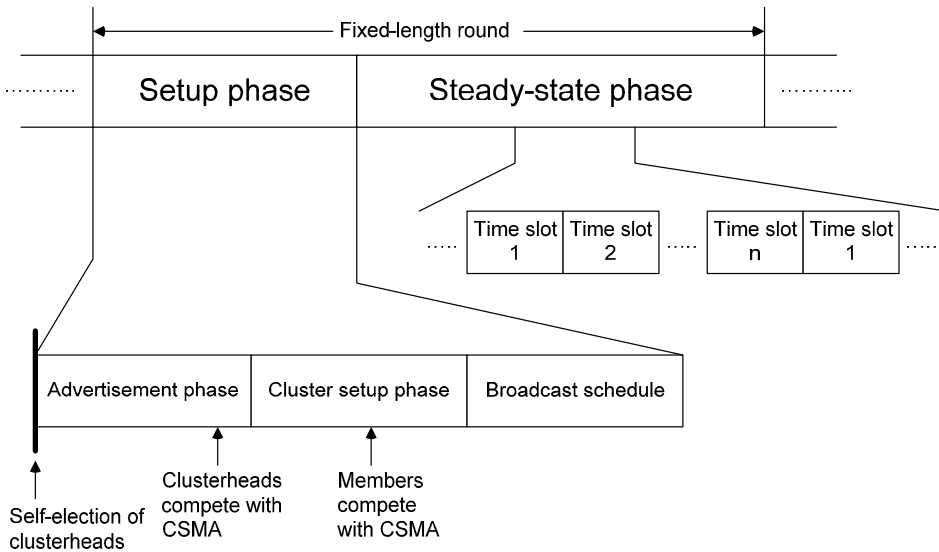


Low-Energy Adaptive Clustering Hierarchy (LEACH)

- 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



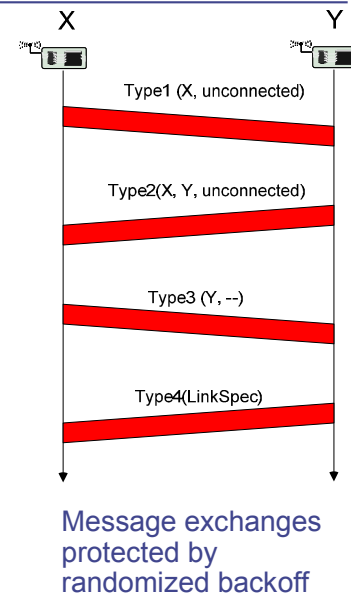
SMACS

- 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
- In effect: a local construction of a schedule



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
- Case 4: both nodes already have links
 - Nodes exchange their schedules and pick free slots/frequencies in mutual agreement



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



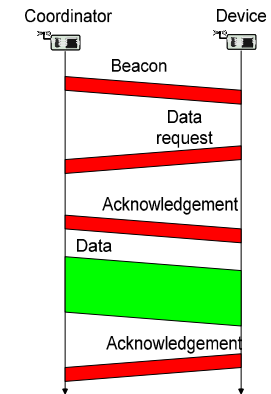
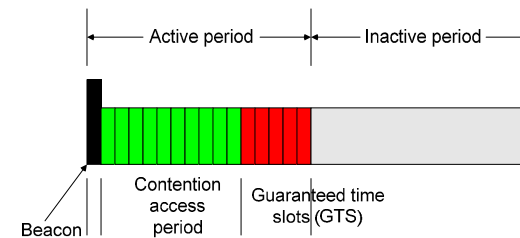
IEEE 802.15.4

- 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
 - Possible option: First send a short **filter packet** that includes the actual destination address to allow nodes to power off quickly
- Not quite so simple scheme: Send a wakeup burst including the receiver address
 - Wakeup radio needs to support this option
- 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...



Summary

- Many different ideas exist for medium access control in MANET/WSN
- Comparing their performance and suitability is difficult
- Especially: clearly identifying interdependencies between MAC protocol and other layers/applications is difficult
 - Which is the best MAC for which application?
- Nonetheless, certain “common use cases” exist
 - IEEE 802.11 DCF for MANET
 - IEEE 802.15.4 for some early “commercial” WSN variants
 - B-MAC for WSN research not focusing on MAC

