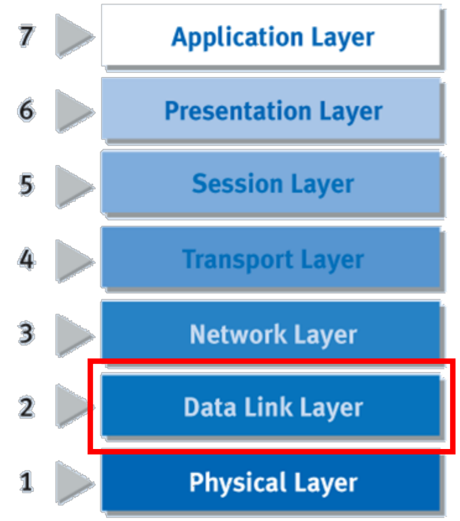
Medium Access

Sensor Networks

Prof. Dr. Kay Römer

Medium Access (MAC)

- Control of access of multiple nodes on shared communication medium
  - Avoidance of collisions
  - Part of link layer
- Strong impact on energy consumption
  - Radio dominates energy consumption



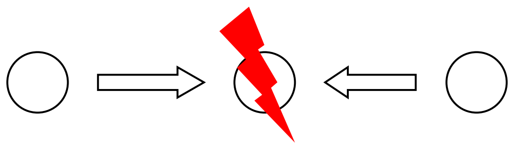
Radio Energy Consumption

- Radio energy consumption facts
  - Send / receive = 1 ... 2
  - Receive / listen = 1 ... 2
  - Listen / sleep >> 100
  - Mode switch takes time
    - 10us – 2ms
- Implications
  - Sleep most of the time (but then „deaf + dumb“)
  - Minimize number of mode switches

Energy Saving Potential

- Useless listening
  - ... for packets while nothing is sent
- Collisions
  - Transmission of packets by multiple nodes at same time
- Overhearing
  - ... of packets addressed to other nodes
- Protocol overhead
  - Headers
  - Control packets

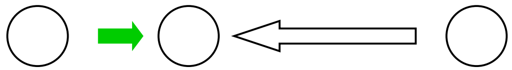
Collisions



- Receiver in range of multiple simultaneous transmissions
- Sender may not detect collisions
  - Not within range of other sender
  - Half duplex: cannot send and receive simultaneously

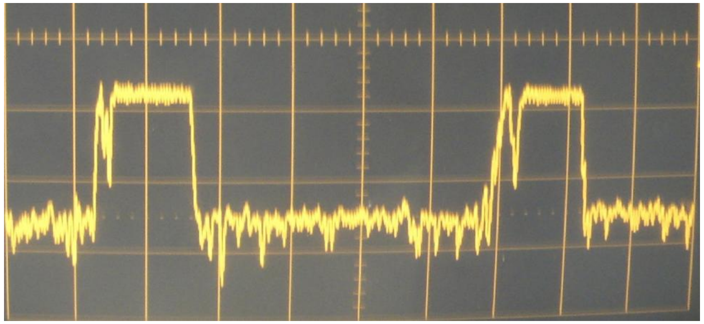
Capture Effect

- Reception of the stronger of two simultaneous transmissions
  - Cf. Sharp transition between FM radio stations instead of overlay of stations on AM radio
  - Depends on modulation scheme and implementation details of the radio

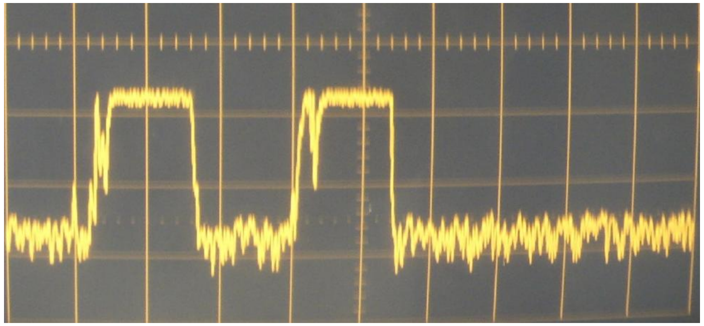


OR Channel

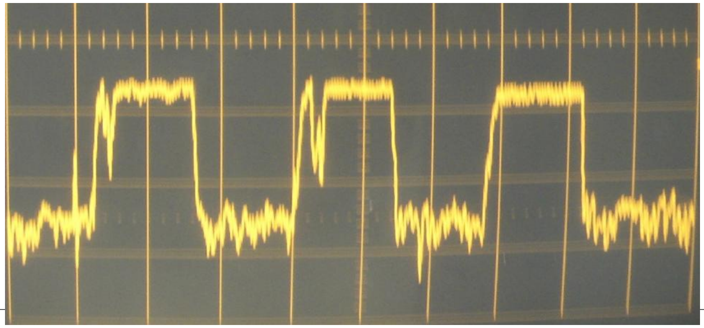
- Multiple bit-synchronized transmissions result in reception of bitwise OR
  - For OOK



+



=

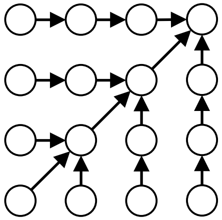


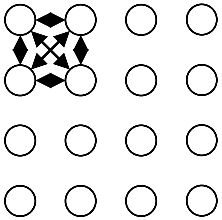
Tradeoffs

- Saving energy typically implies other drawbacks
  - Latency, esp. across multiple hops
  - Bandwidth (bits per second)
  - Fairness (some nodes prioritized over others)

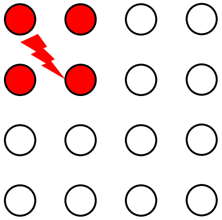
Traffic Patterns

- Exploit knowledge about traffic patterns for energy saving
- Message flow
  - Convergecast: All sensors to sink (potentially with aggregation)
  - Broadcast: Sink to all nodes
  - Local interactions among neighbors
  - Geo routing: send to nodes at certain position





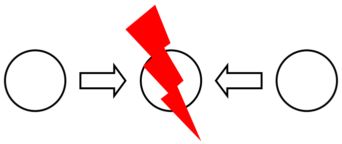
- Message generation
  - Streams
    - Periodic sensor readout / sending
  - Bursts
    - Most of the time idle
    - Event triggers transmission at multiple nearby nodes
- Application-specific properties
  - Cross layer!



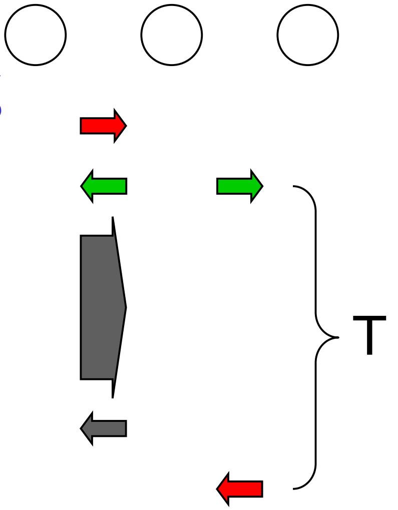
MAC Protocols

- Contention
  - Random access to medium
  - Avoidance of (most) collisions
    - Clear channel assessment (CSMA), random delays, channel reservation
  - Retransmission in case of collision
- Scheduling
  - Exclusive assignment of medium to nodes
    - Time slices (TDMA), frequency bands (FDMA)
  - Node only active in assigned time slot
  - No collisions, idle listening, overhearing
  - But: time synchronization, adaptivity?

CSMA/CA

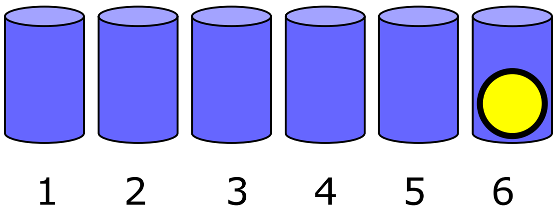


- CSMA: Listen, send when channel clear
  - Problem: „hidden terminals“
- CA: Channel reservation
  - RTS (Request to send): Request including duration T of planned transmission
  - CTS (Clear to send): Ack with T
  - DATA: After receipt of CTS
  - ACK: After receipt of DATA
  - All other nodes overhearing RTS/CTS delay all transmissions by T

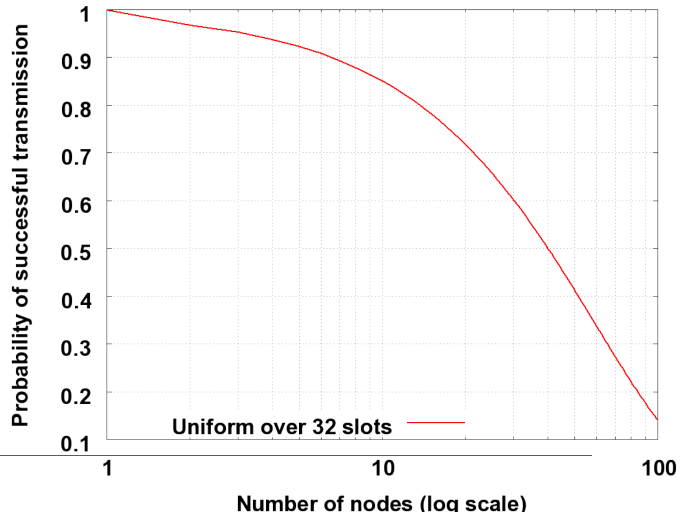


CSMA/CA: Collisions

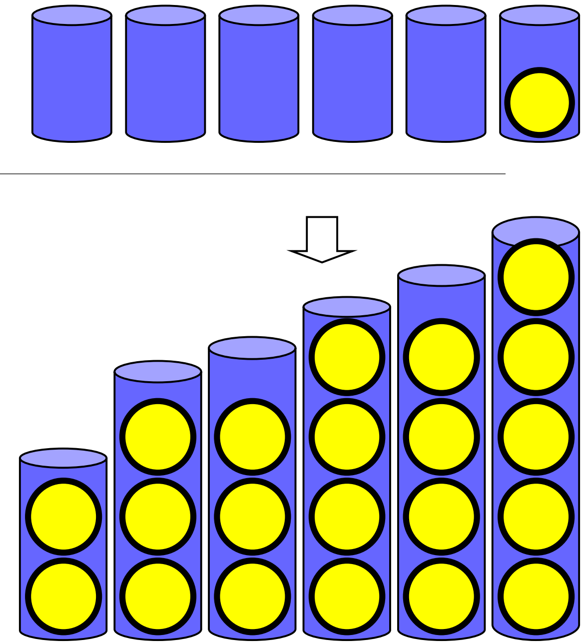
- Still collisions possible
  - Detected by missing ACK
  - Retransmit after random delay
- Delay strategies
  - Random uniform selection of a time slot from a time window
  - Problem: many nodes -> many collisions
    - Note: traffic bursts!



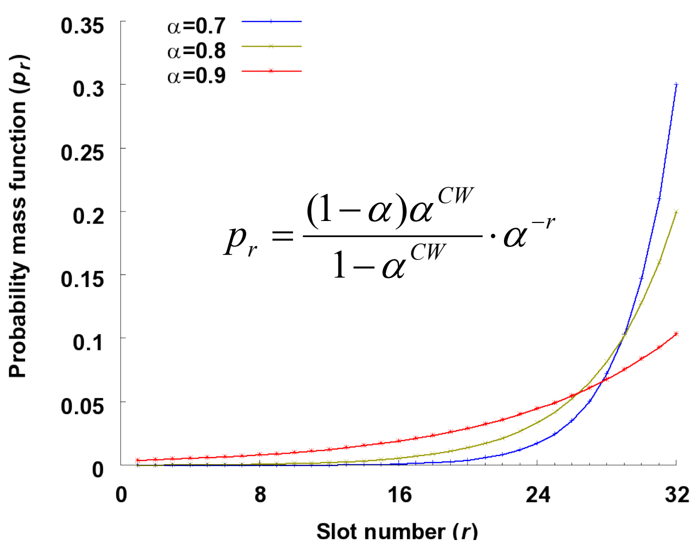
- Binary Exponential Backoff (BEB)
  - Double time window after each successive collision
  - Adapt to number of nodes
  - Problem: long delays

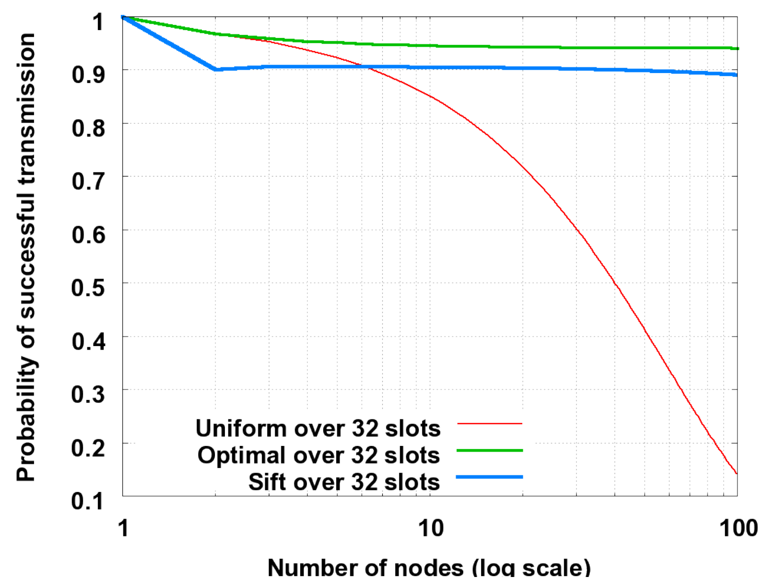


Sift



- BEB alternative
  - Fixed time window, but non-uniform probability of slot selection
  - Few nodes select early slot
  - Collision probability of first transmission small
  - Limited delay until first transmission(s)



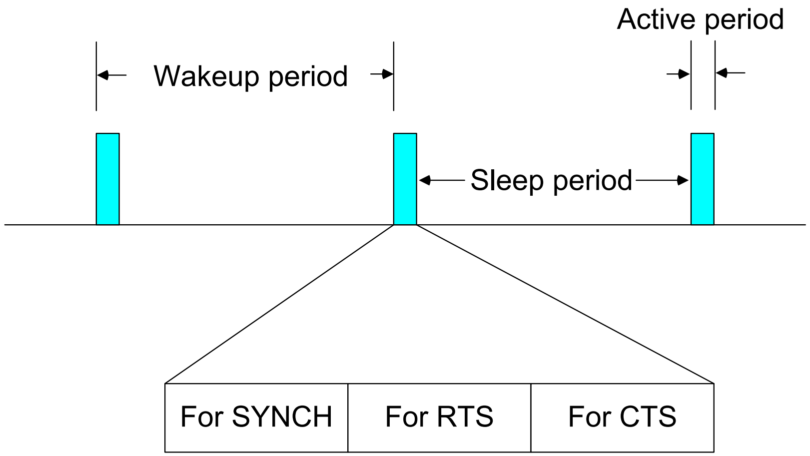


CSMA/CA for Sensor Nets

- Useless listening
  - Switch off radio during T
  - Problem: miss other RTS/CTS
    - Additional collisions
    - Bandwidth reduced by about 25%
- Protocol overhead
  - RTS/CTS/ACK for each payload packet!
    - Very inefficient for small messages
  - Reserve channel for multiple packets
    - Fairness vs. energy!

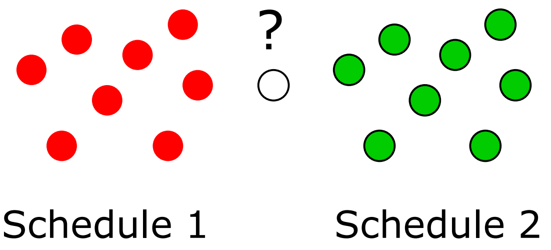
S-MAC

- CSMA/CA with duty cycling
  - Short active phase followed by long sleep phase
  - Nodes synchronized to wakeup simultaneously
  - RTS/CTS only during active phase
  - Transmission of data into sleep phase
- SYNCH/RTS/CTS
  - CSMA phases with slotted random delays
  - If collision retry in next active phase



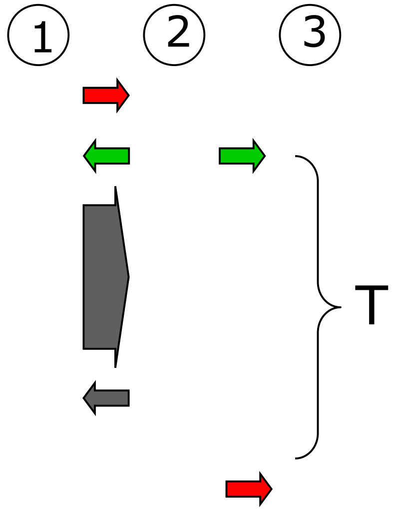
S-MAC: Synchronization

- Nodes transmit schedule packets during SYNCH
- New node
  - Listen and sync to schedule
  - If no neighbor define own schedule
- Problem: Merging islands with different schedules
  - Follow all neighboring schedules



S-MAC: Latency

- Expected per-hop latency = sleep duration / 2
  - Problem with long paths!
- Improvements:
  - Path: 1 -> 2 -> 3
  - Node 3: hears CTS, after T additional active phase
  - Node 2: After ACK send RTS immediately



S-MAC: Fragmentation

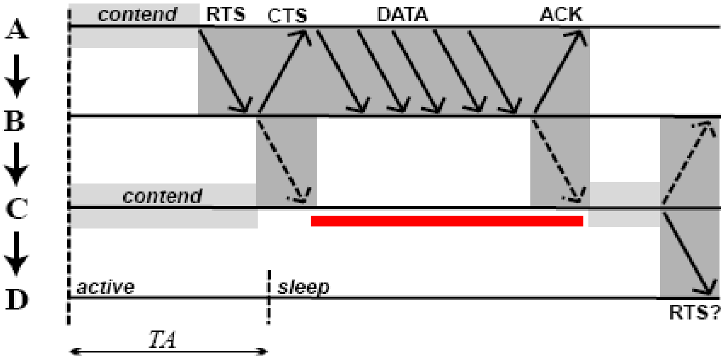
- High BER: Split long packets into many small ones
  - Problem: RTS/CTS/ACK for each fragment induces high overhead
- Solution in S-MAC
  - One RTS/CTS for all fragments
  - One ACK per fragment, retransmit if lost
  - All packets contain remaining duration T for transmission of all remaining fragments
- Fairness?

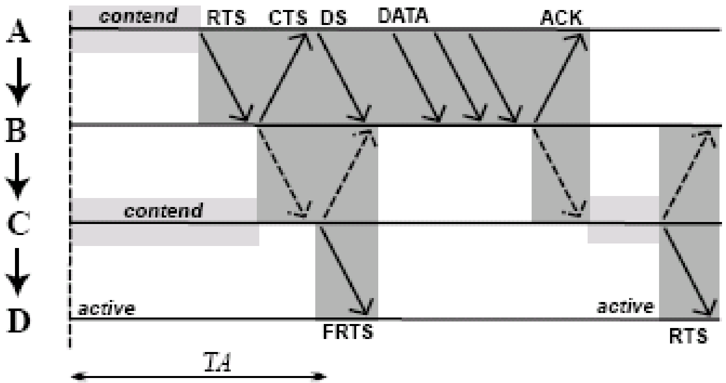
T-MAC

- S-MAC: Fixed duration of active phase
  - Waste when low traffic
  - Too short for high traffic
- T-MAC: adapt duration of active phase to traffic
  - Go to sleep early if no activity during timeout period TA
  - How does a node detect „no activity“
    - No messages received
    - No messages received at neighbors (implicitly known from overheard RTS/CTS)

T-MAC: Early Sleeping

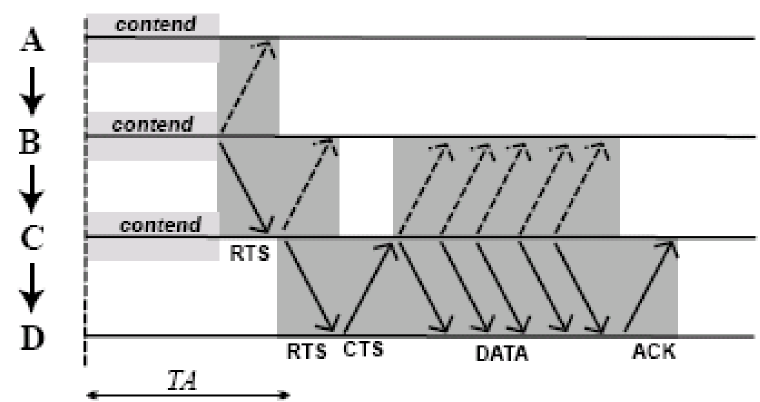
- Problem: D sleeps to early
  - Node C has data for D, but cannot send RTS to D as medium blocked by CTS
- Solution: Future RTS
  - Upon receipt of CTS, node C sends FRTS to D
  - Contains T from CTS
  - Dummy packet DS





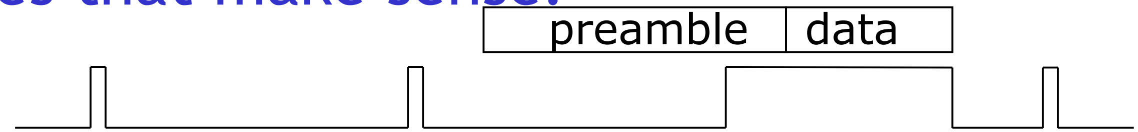
T-MAC: Buffer Overrun

- Problem: nodes forced to receive
  - RTS/CTS can force node to receive many packets without chance to forward them
  - Limited buffer space!
- Solution: RTS instead of CTS
  - Upon receipt of RTS node C sends RTS to D instead CTS
  - Only when C‘s buffers full and C has „lost“ contention multiple times



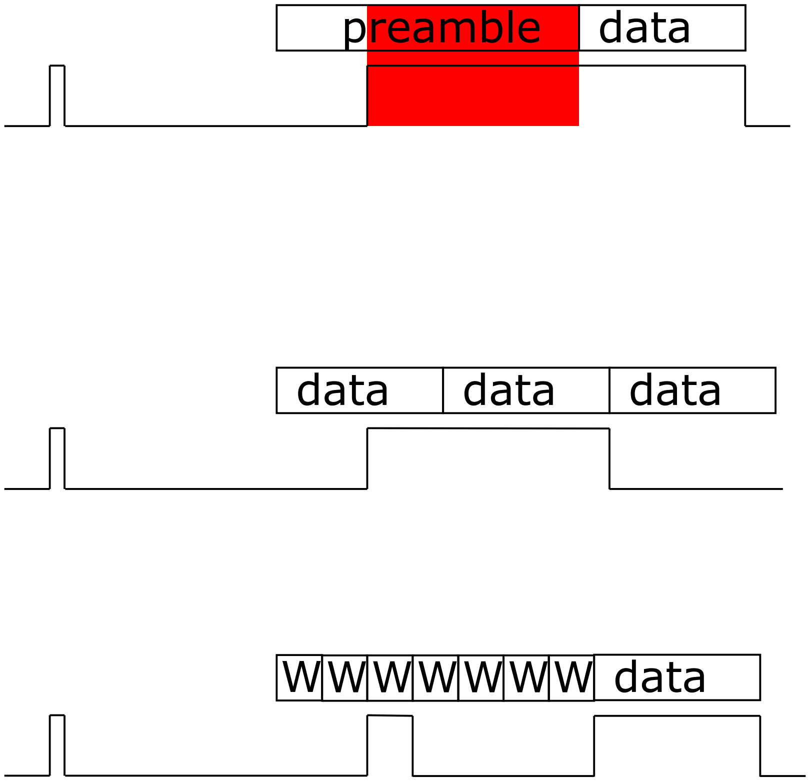
Wakeup Preambles

- Problem: Both S-MAC and T-MAC need synchronization
  - Multiple schedules per node ...
- Other approach
  - Unsynchronized short active phases
  - Sender transmits long wakeup preamble before message
    - Duration?
  - If node overhears preamble during active phase: stay awake to receive packet
- Shift load to sender
  - S-MAC / T-MAC: Useless active phases
  - Does that make sense?



Preamble Optimizations

- Problem: Transmit full preamble even if receiver already awake
  - Waste of energy
- Send repeated data instead of preamble
- Send repeated wakeup packet instead of preamble
  - Contains time until data

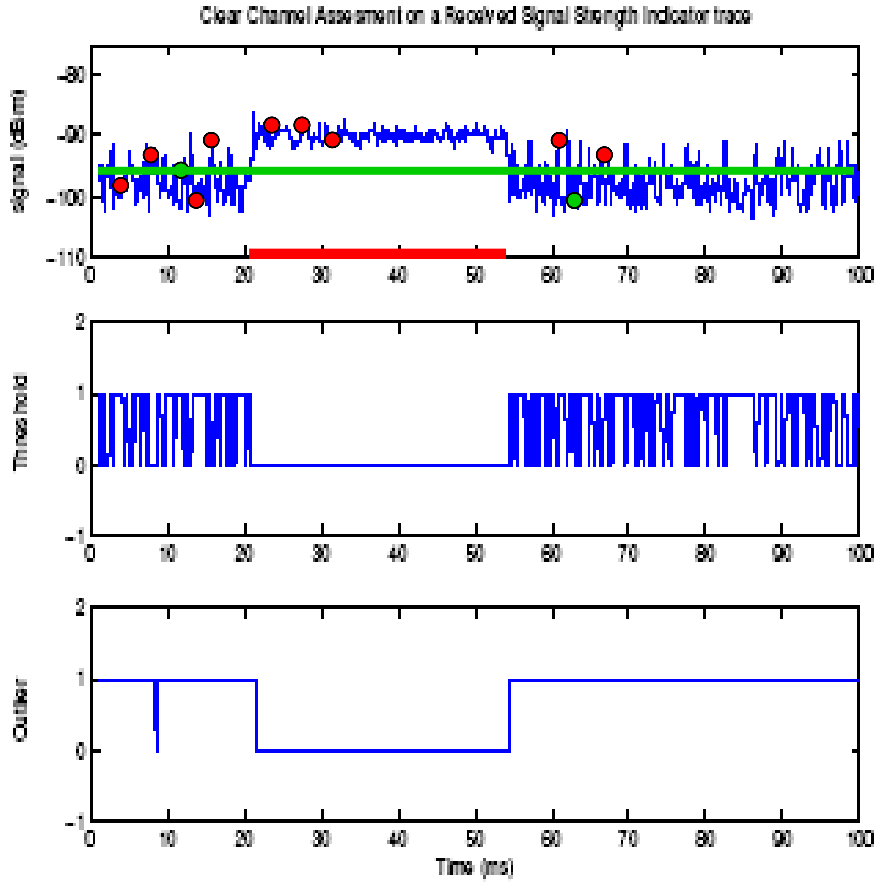


B-MAC

- Basic preambles
- Philosophy: minimum functionality with simple interface (TinyOS)
  - Configurable duration of sleep phase
  - Configurable preamble length
  - Fixed active phase with clear channel assessment
  - CSMA with simple random backoff
  - Optional ACKs
- Other functions have to implemented on top
  - RTS/CTS, ...
- Standard MAC for radios with byte interface (CC1000)

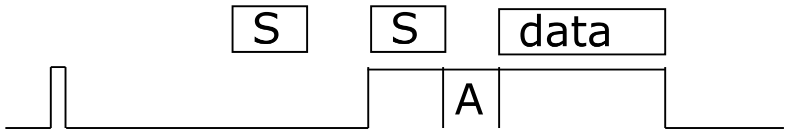
B-MAC: Clear Channel?

- Preamble detection
  - Reliable detection to avoid useless listening
- Estimate noise level
  - Measure signal strength when no transmission (when?)
  - Low pass filter: Median M of last 5 measurements
  - Noise level R: exponentially weighted moving average over M
    - $R = \alpha R + (1-\alpha)M$
- Clear channel?
  - Measure signal strength multiple times
  - Outlier < R? -> Channel clear!



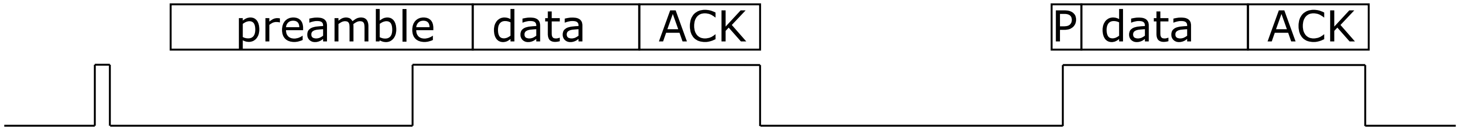
X-MAC

- Variant of preambles for packet-based radios
- Sender sends sequence of strobes and waits for ack from destination before transmitting data



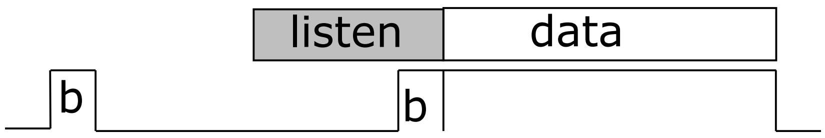
WiseMAC

- Preambles AND synchronization (!)
  - Initially long preamble
  - ACK contains wakeup schedule
  - Sender knows when receiver will wakeup next
    - Only approximately!
  - Short preamble
  - Does not work for broadcast
- Other features
  - „More!“ bit in header for packet bursts
  - Repeated data instead of long preamble
  - Sensitive clear channel assessment instead of RTS/CTS



Receiver-Initiated MAC

- Receivers periodically wake up and broadcast beacon message
- Transmitter waits to receive beacon from destination and then transmits data
- Minimized channel occupation for rendezvous
- Example: RI-MAC

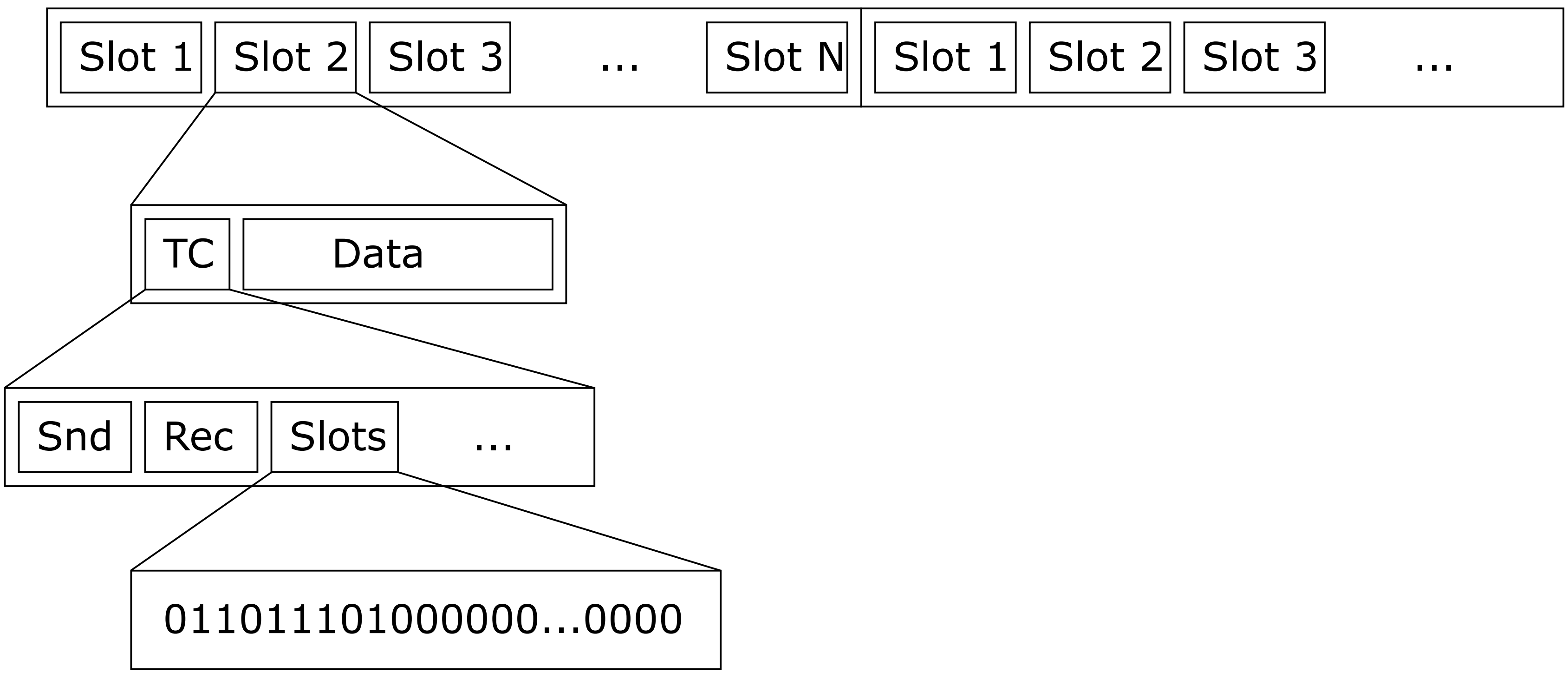


MAC with Scheduling

- N time slots per time frame
  - Each node is assigned one or more time slots per frame
- Many variants how to assign slots
  - Centralized vs. distributed
  - Static vs. on demand

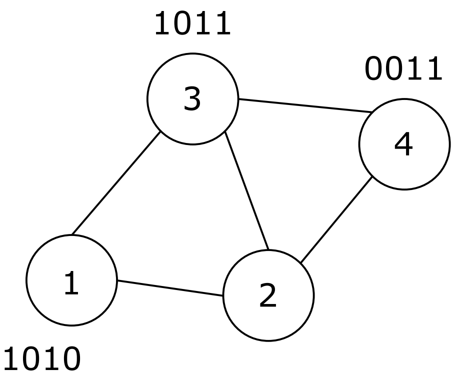
LMAC

- 32 slots per frame
- Each node is assigned one slot
- Node can send packet to neighbors during its slot
  - Neighbors overhear TC and synchronize
  - Only addressed neighbors receive data
- No ACKS, no retransmission in original LMAC!



LMAC: Slot Assignment

- TC.Slots contains bit vector with assigned slots
  - From sender's perspective
- New node
  - Overhears complete frame with TC.Slots of all neighbors
  - Chooses free slot
- Mobility and link fluctuation can lead to collisions (slot assigned to multiple neighbors)
  - Sender selected random new free slot

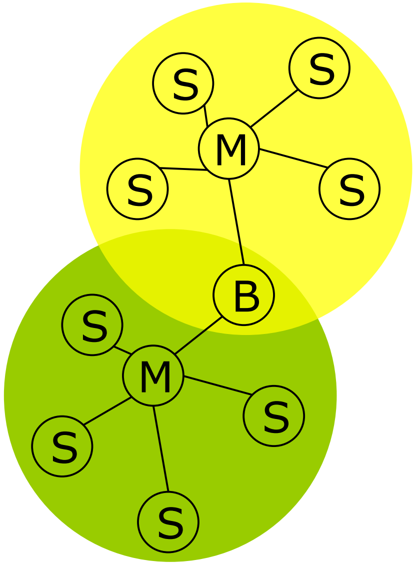


LMAC: Limitations

- Network density limited by number of slots per frame
  - 2 hop neighborhood must not reuse slot
- Bandwidth per node limited
  - Channel bandwidth / number of slots
  - Problematic when nodes generate different amount of traffic

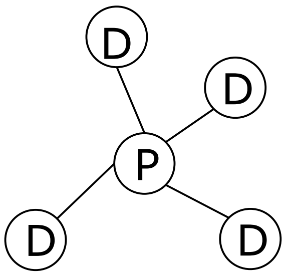
Bluetooth

- Physical Layer
  - 79 Channels
  - Pseudo-random frequency hopping
    - How to discover neighbors?
  - Bandwidth about 1 mbps
    - Energy efficient for high traffic load
- Scheduling-based MAC
  - Piconet: 1 master + max. 7 slaves
    - Master synchronizes slaves and defines hopping sequence
    - Time slot assigned to each slave for communication with master
  - Scatternets: overlapping piconets
    - Bridge: node participating in >= 2 piconets
    - Periodic switching between piconets

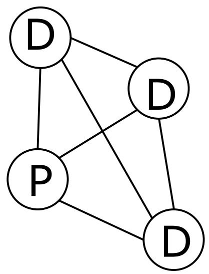


IEEE 802.15.4 („Zigbee“)

- Physical layer
  - 20 – 250 kbps in different ISM bands (868 kHz, 926 kHz, 2.4 GHz)
  - In some bands multiple channels with dynamic selection
- Node roles
  - PAN coordinator
  - Coordinator
  - Device



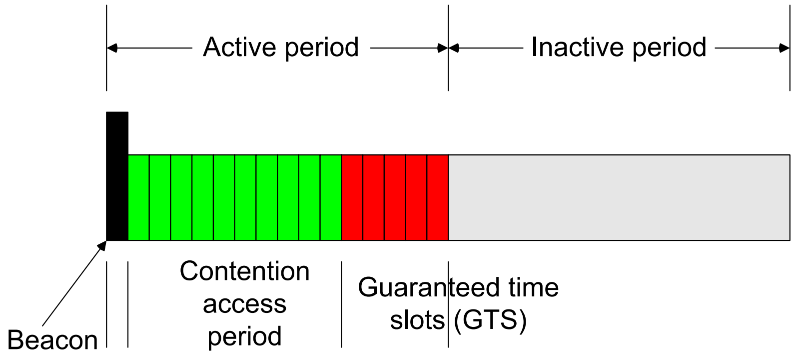
- Node types
  - Full Function Device
  - Reduced Function Device
    - Only „device“ role



- Network topologies
  - Star (MAC: Beacon Mode = CSMA/CA + time slots)
  - Peer-to-Peer (MAC: Non-Beacon-Mode = CSMA/CA)

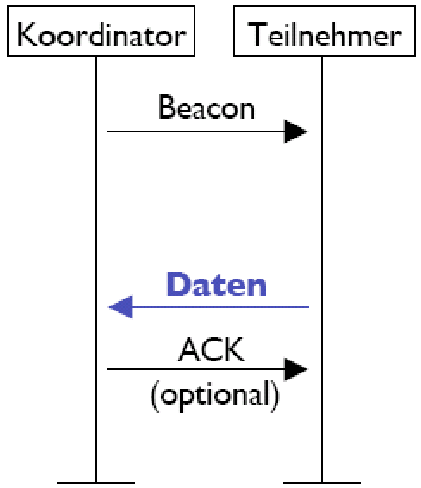
Beacon Mode

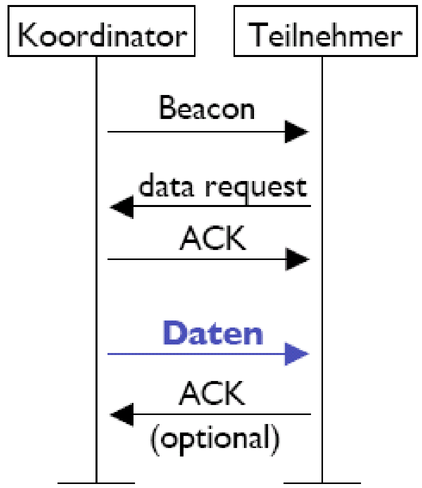
- Coordinator and devices form star network
  - Coordinator broadcasts Beacon periodically
    - Synchronization
    - Frame organization
    - Allocation of GTS
  - Active period
    - CAP: CSMA
    - GTS: TDMA
  - Inactive period
    - Sleep



Beacon Mode: Data Transfer

- Device to coordinator
  - GTS - only when slot allocated and data packet fits into GTS
  - CSMA - otherwise
- Coordinator to device
  - GTS – only when slot allocated and data packet fits into GTS
  - Announcement of receivers in Beacon
  - Receivers send request during CAP
  - Coordinator sends packet





Beacon Mode: CSMA

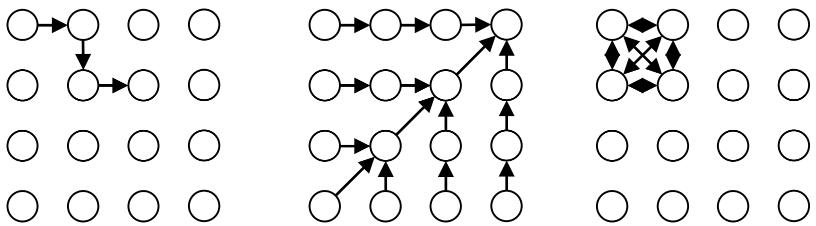
- Using during CAP
  - BEB before transmission
  - Send if channel clear
  - No RTS/CTS

Other Approaches

- Large number of other MAC protocols
  - Shown principles / protocols most relevant
- MACs exploiting further physical layer features
  - Concurrent use of multiple channels
  - Multiple radio modules per node
    - Wakeup radio
  - Ultra wideband radios
  - Directional antennas
  - Light, sound, …
  - ...

Which MAC is best?

- Is there a MAC which is always the best?
  - Resource consumption
  - Energy consumption
  - Throughput
  - Packet loss
- Traffic scenarios
  - Point to point
  - Convergecast
  - Local interaction



- Protocols
  - 802.11: plain CSMA/CA
  - S-MAC
  - T-MAC
  - LPL: preambles (B-MAC)
  - LMAC

Resource Consumption

	LMAC	LPL	802.11	S-MAC	T-MAC
code size (lines)	250+75	325	400	625	825
RAM usage (bytes)	15+40	49	51	78	80

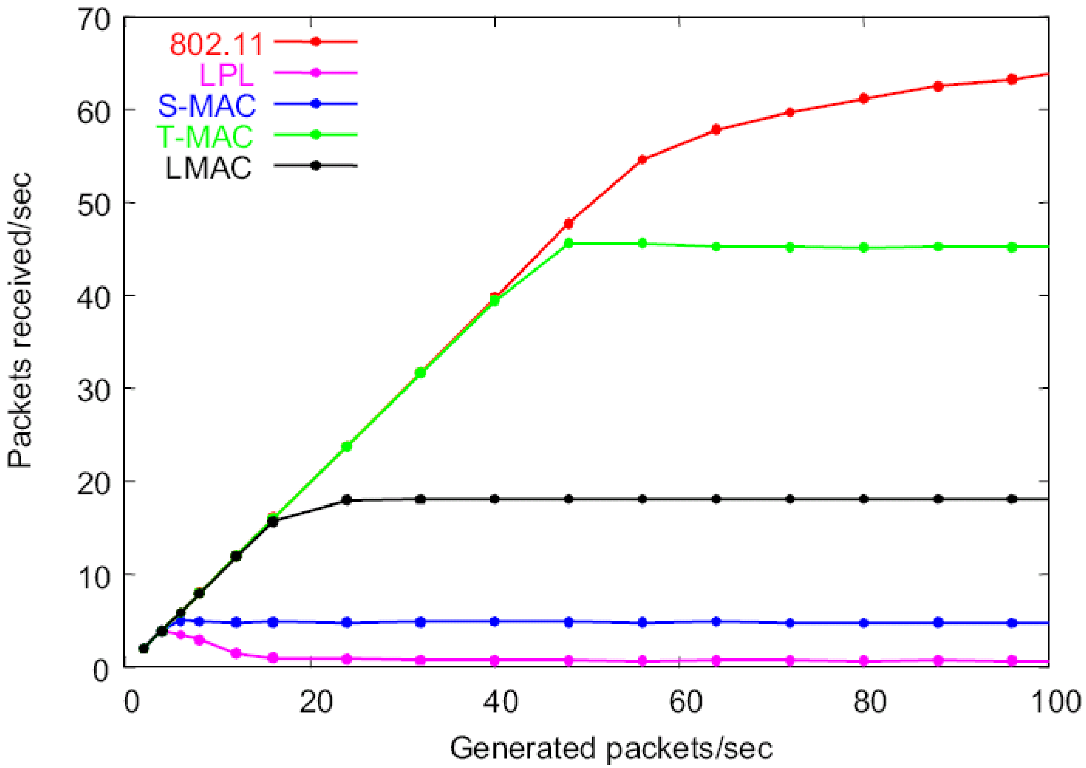
- LMAC: without / with ACK and retransmissions
- 802.11: RTS/CTS
- S-MAC: synchronization
- T-MAC: adaptation

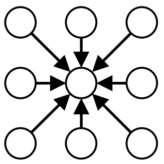
Idle Energy Consumption

	802.11	LPL	S-MAC	T-MAC	LMAC
energy consumption [mW]	11.4	1.14	1.21	0.37	0.75
effective duty cycle [%]	100	10	11	3.2	6.6

- S-MAC / T-MAC: synchronization

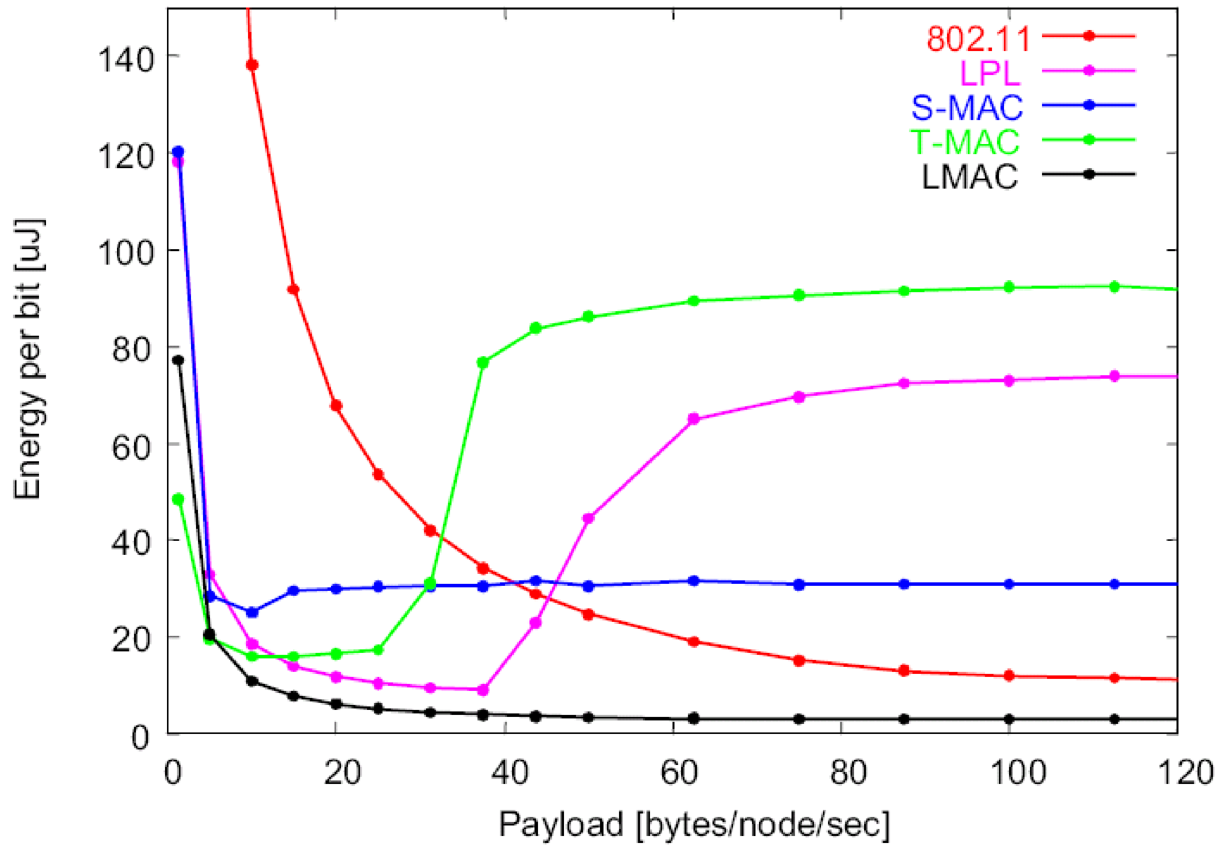
Throughout on 3x3 Grid





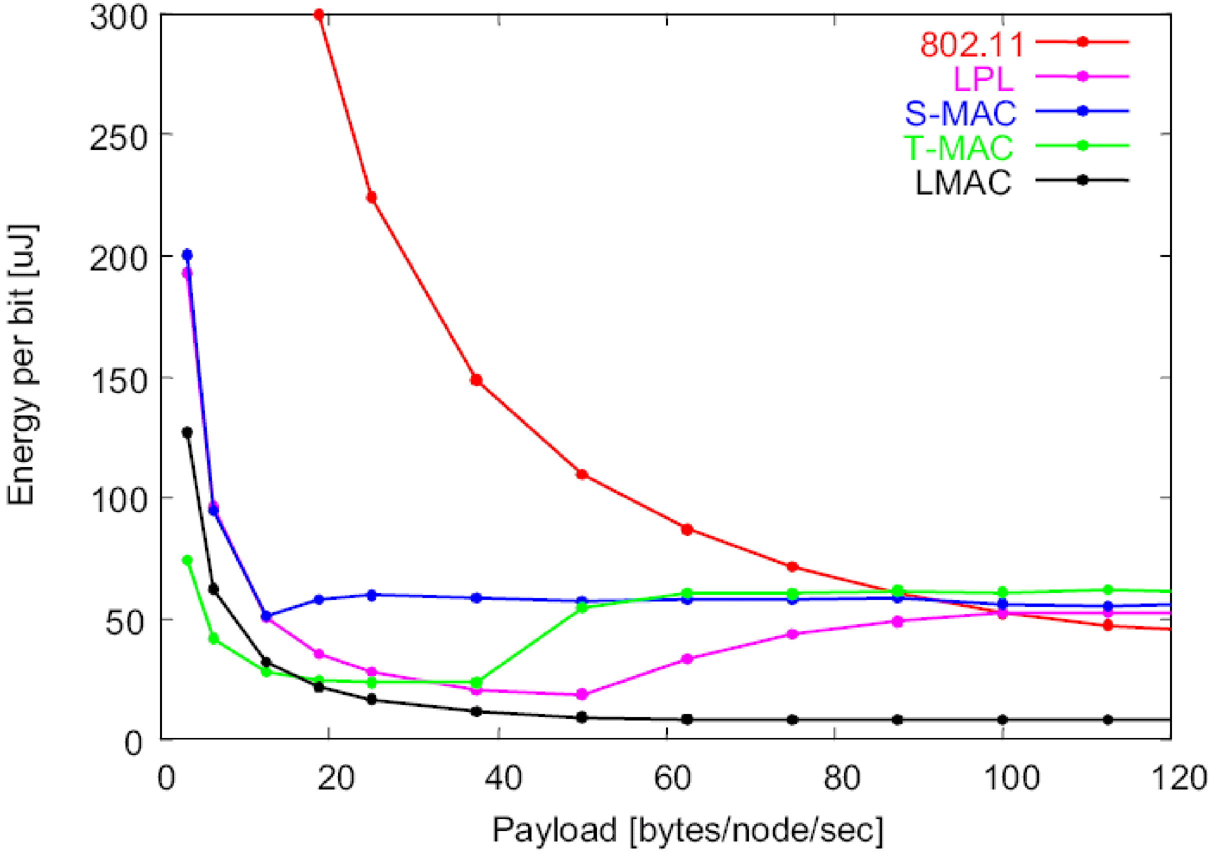
- T-MAC: fixed backoff time window
- LMAC: 1 slot per node and frame
- S-MAC: Fixed duty cycle
- LPL: hidden terminals

Energy for Point to Point



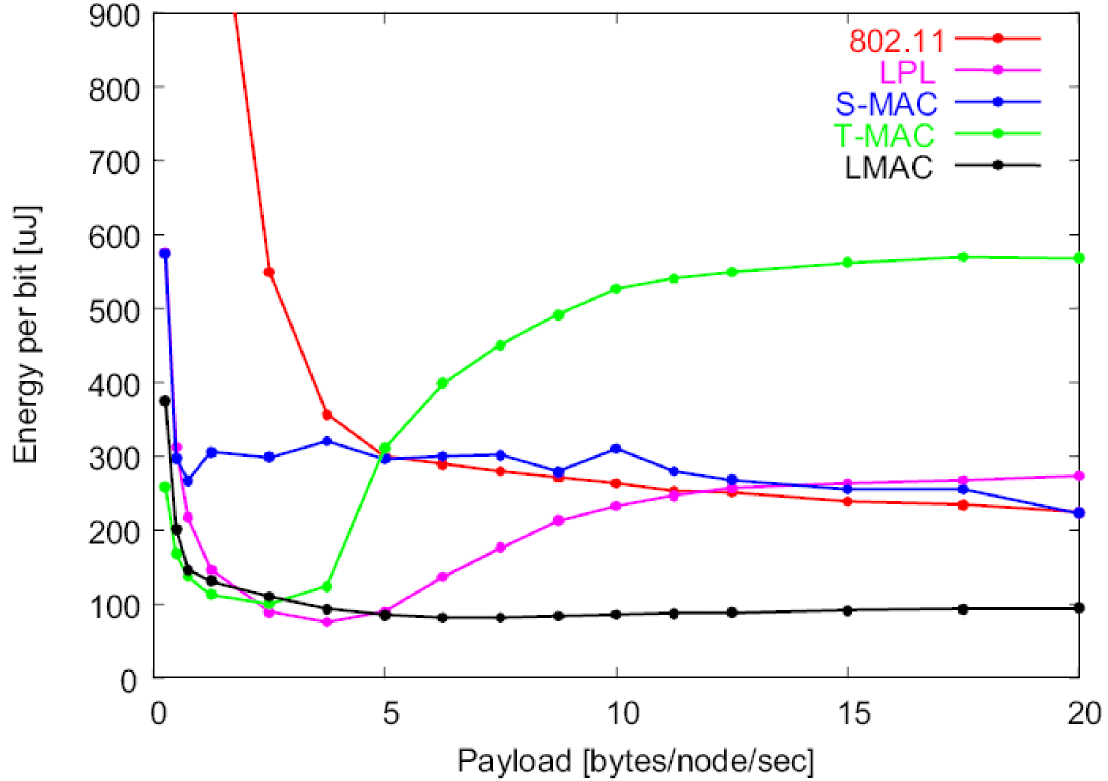
- T-MAC: early sleeping
- LPL: hidden terminal collisions
- S-MAC: fixed duty cycle

Energy for Local Interactions



- Similar trends, but less extreme

Energy for Convergecast



- Similar trends, but more pronounced again

Conclusions

- „Winner“ regarding energy consumptions
  - Low traffic load: T-MAC
    - Adaption
  - High traffic load: LMAC
    - No collisions
- Further observations
  - 802.11 competitive at high traffic loads
  - There is no best MAC, depends on application
  - Choice often non-trivial

References

- Slides contain material by the following authors
  - Holger Karl – Uni Paderborn
  - Koen Langendoen – TU Delft
  - Kyle Jamieson – MIT
  - Matthias Ringwald – ETH Zürich