MAC Layer Protocols for Sensor Networks

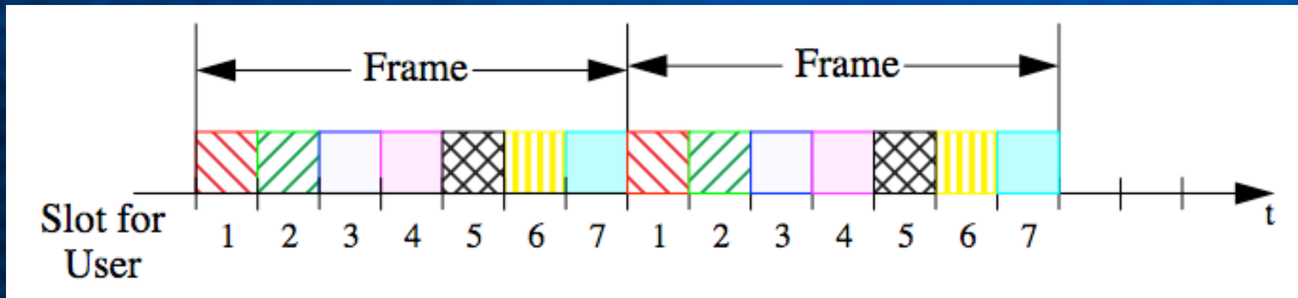
Leonardo Leiria Fernandes

Contents

- Basic Concepts
- S-MAC
- T-MAC
- B-MAC
- P-MAC
- Z-MAC

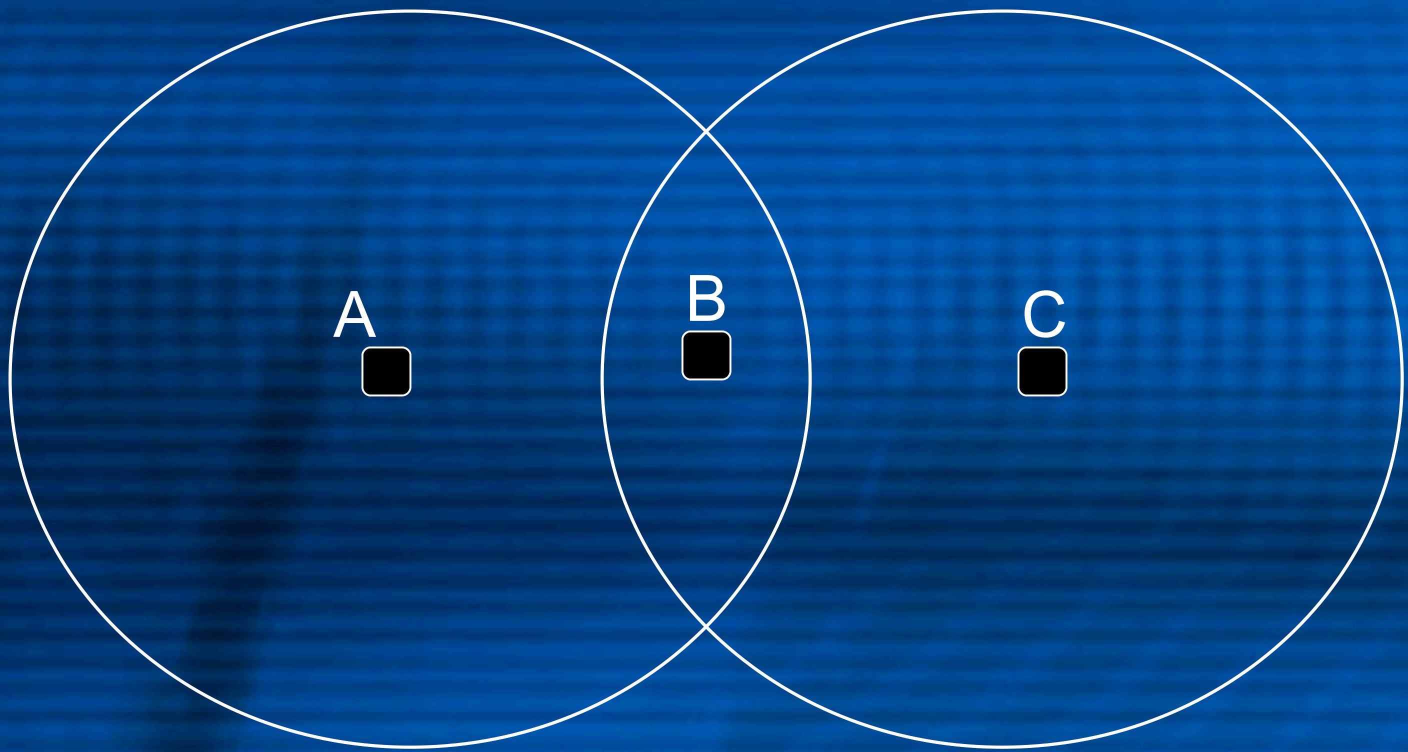
Basic Concepts

- Problem
- TDMA



- CSMA
- RTS / CTS

Hidden Nodes



MAC Challenges

- Traditionally
  - Fairness
  - Latency
  - Throughput
- For Sensor Networks
  - Power efficiency
  - Scalability

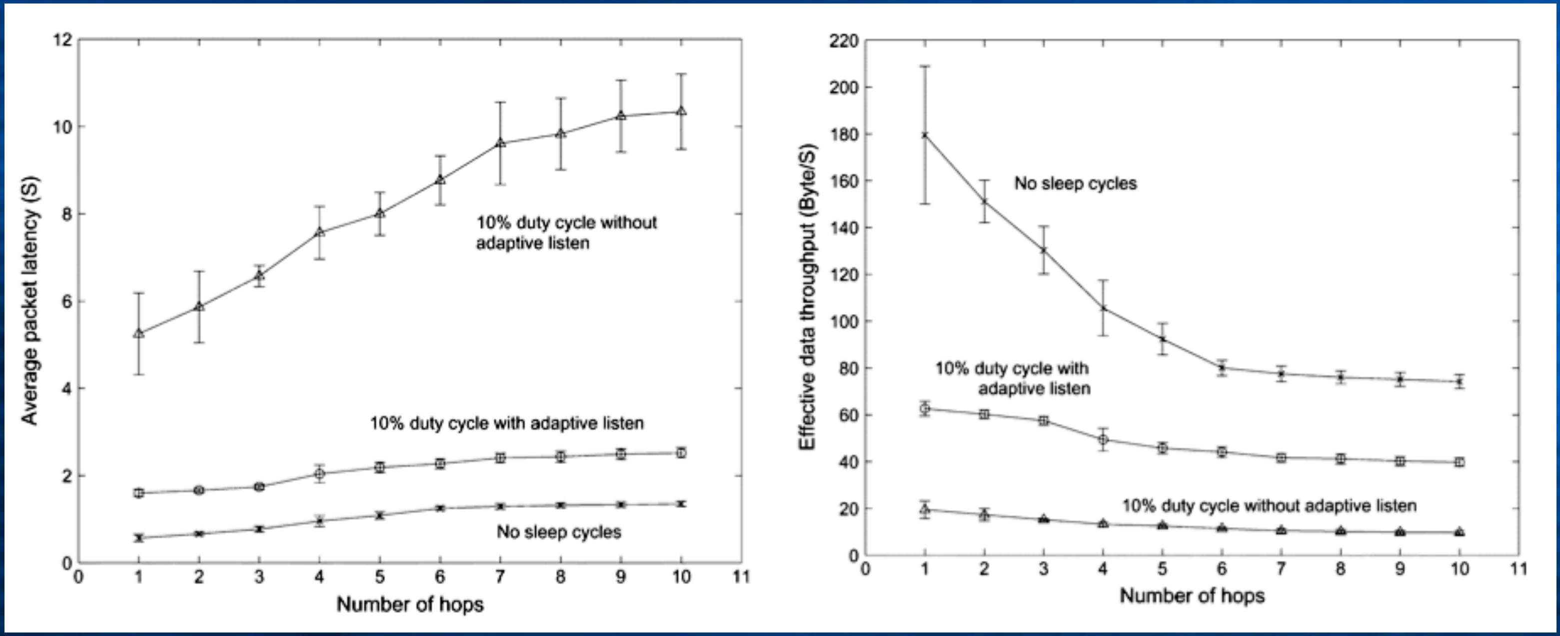
S-MAC - Sensor MAC

- Nodes periodically sleep
- Trades energy efficiency for lower throughput and higher latency
- Sleep during other nodes transmissions

S-MAC

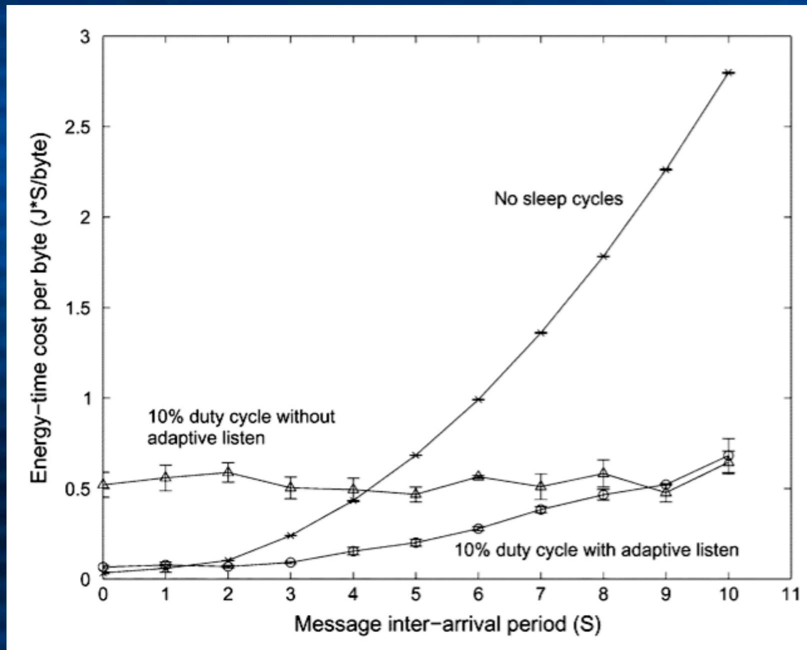
- Listen significantly longer than clock drift
- Neighboring nodes exchange SYNC msgs
- Exchanged timestamps are relative rather than absolute
- RTS/CTS avoids hidden terminal
- Message passing provided
- Packets contain expected duration of message
- Every packet must be acknowledged
- Adaptive listening can be used so that potential next hop nodes wake up in time for possible transmissions

S-MAC Results



Latency and throughput are problems, but adaptive listening improves it significantly

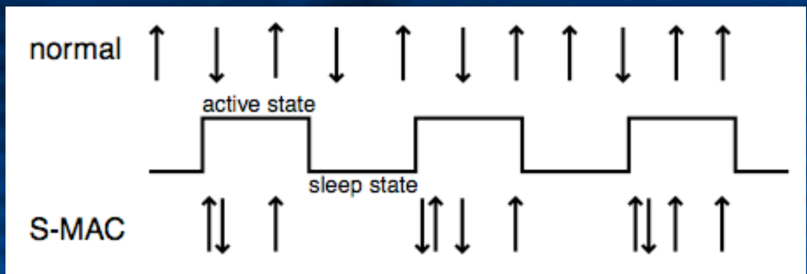
S-MAC Results

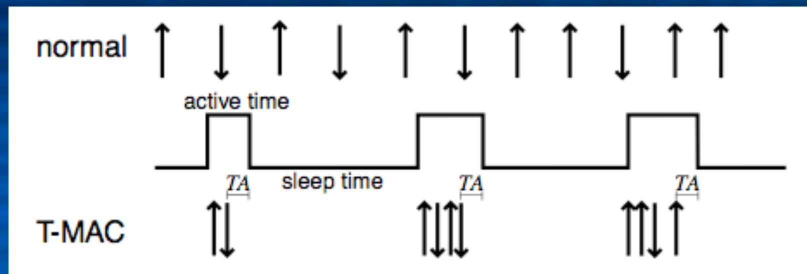


- Energy savings significant compared to “non-sleeping” protocols

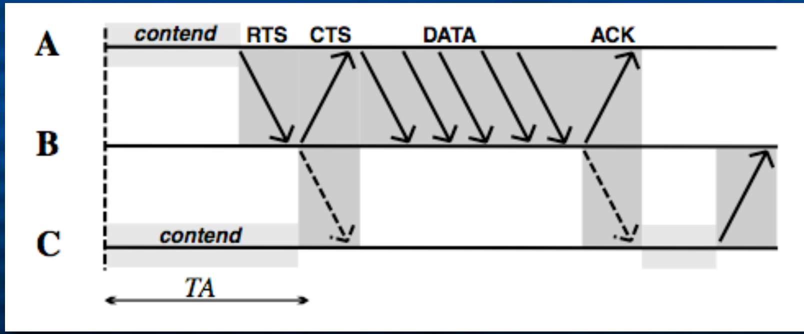
T-MAC - Timeout MAC

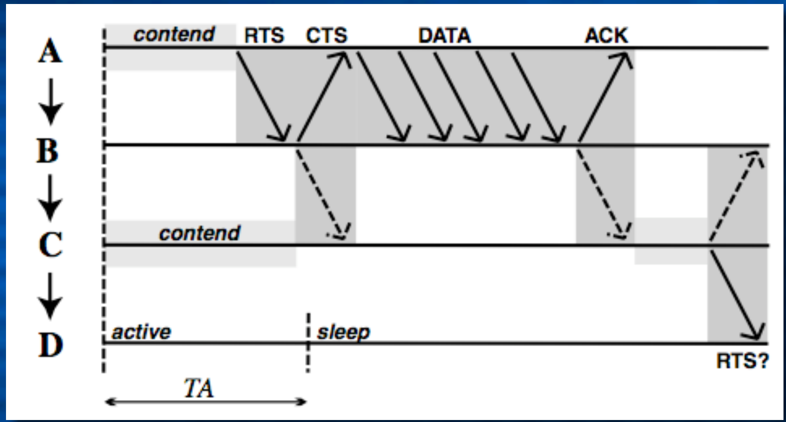
- Transmit all messages in bursts of variable length and sleep between bursts
- RTS / CTS / ACK Scheme
- Synchronization similar to S-MAC

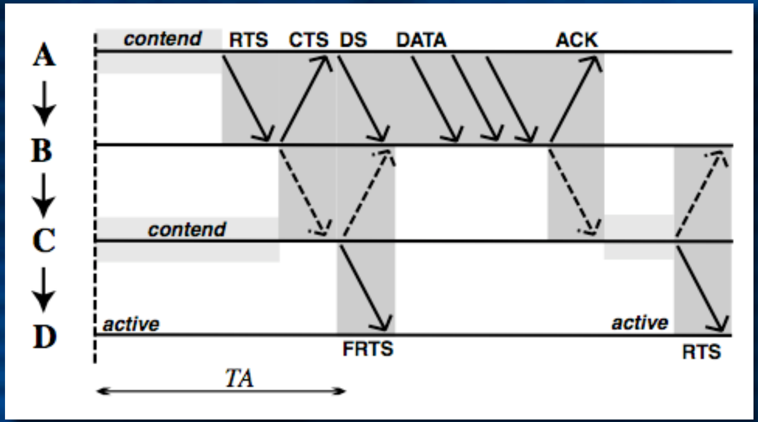


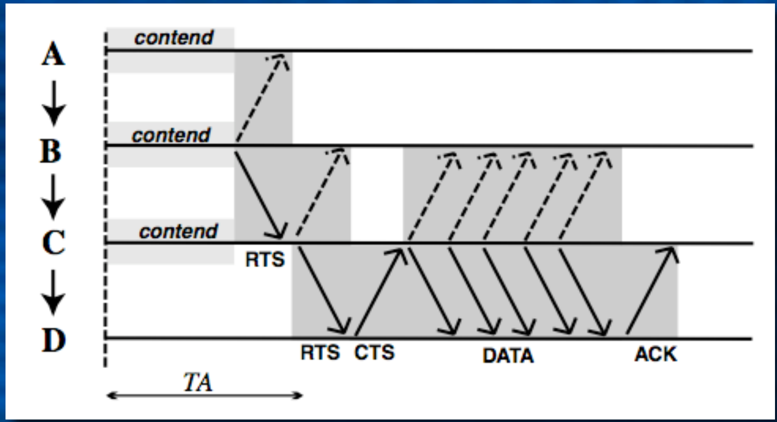


T-MAC Operation

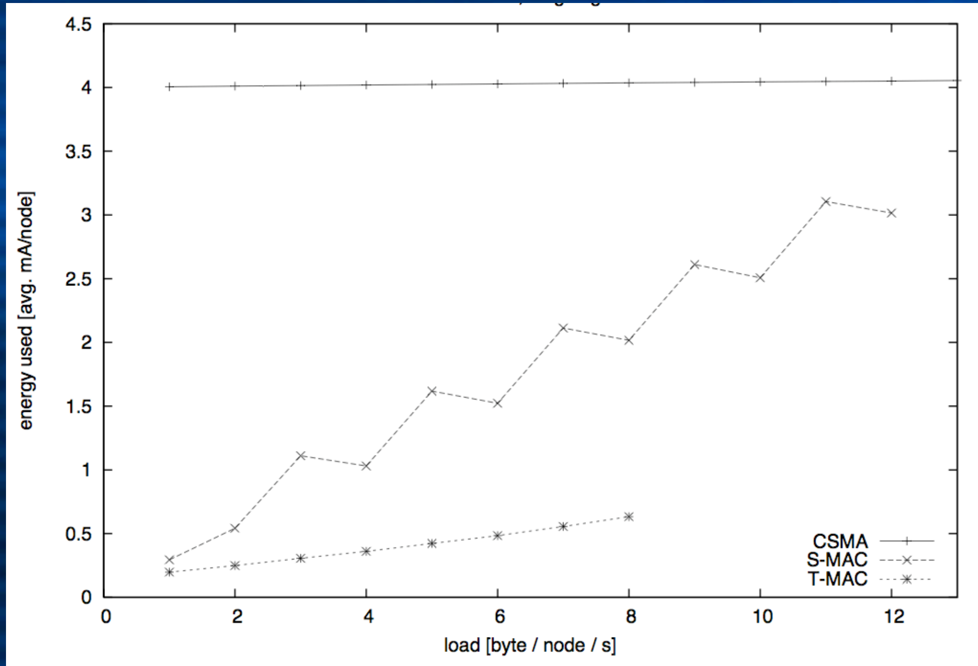








T-MAC Results



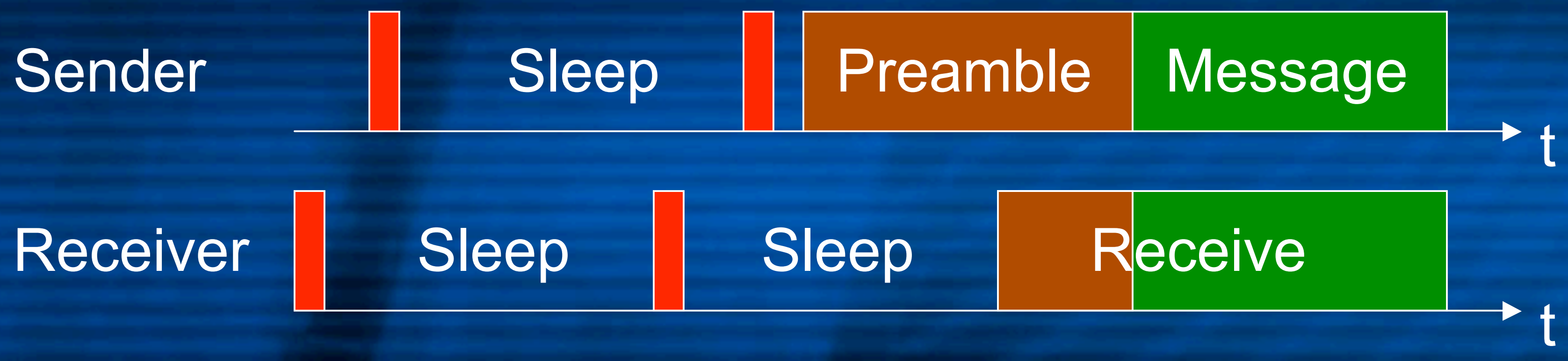
- T-MAC saves energy compared to S-MAC
- The "early sleeping problem" limits the maximum throughput
- Further testing on real sensors needed

B-MAC - Berkeley MAC

- B-MAC's Goals:
  - Low power operation
  - Effective collision avoidance
  - Simple implementation (small code)
  - Efficient at both low and high data rates
  - Reconfigurable by upper layers
  - Tolerant to changes on the network
  - Scalable to large number of nodes

B-MAC's Features

- Clear Channel Assessment (CCA)
- Low Power Listening (LPL) using preamble sampling
- Hidden terminal and multi-packet mechanisms not provided, should be implemented, if needed, by higher layers



B-MAC Interface

- CCA on/off
- Acknowledgements on/off
- Initial and congestion backoff in a per packet basis
- Configurable check interval and preamble length

B-MAC Lifetime Model

$$E = E_{rx} + E_{tx} + E_{listen} + E_{d} + E_{sleep}$$

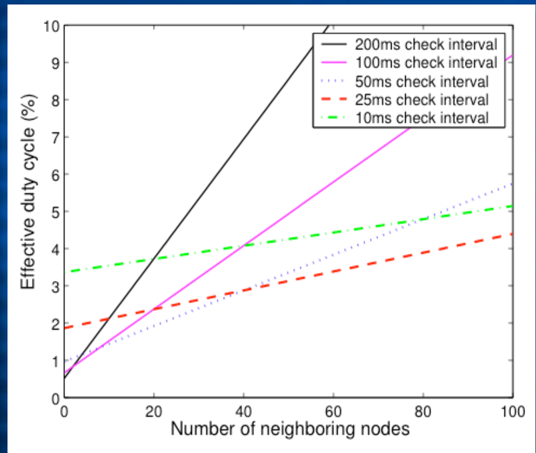
$$E_{rx} = t_{rx} c_{rxb} V$$

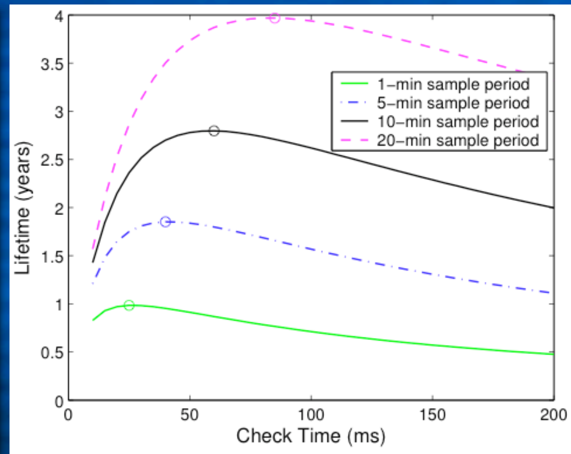
$$E_{tx} = t_{tx} c_{txb} V$$

$$E_{listen} \le E_{sample} \frac{1}{t_i}$$

$$E_{d} = t_{d} c_{data} V$$

$$E_{sleep} = t_{sleep} c_{sleep} V$$



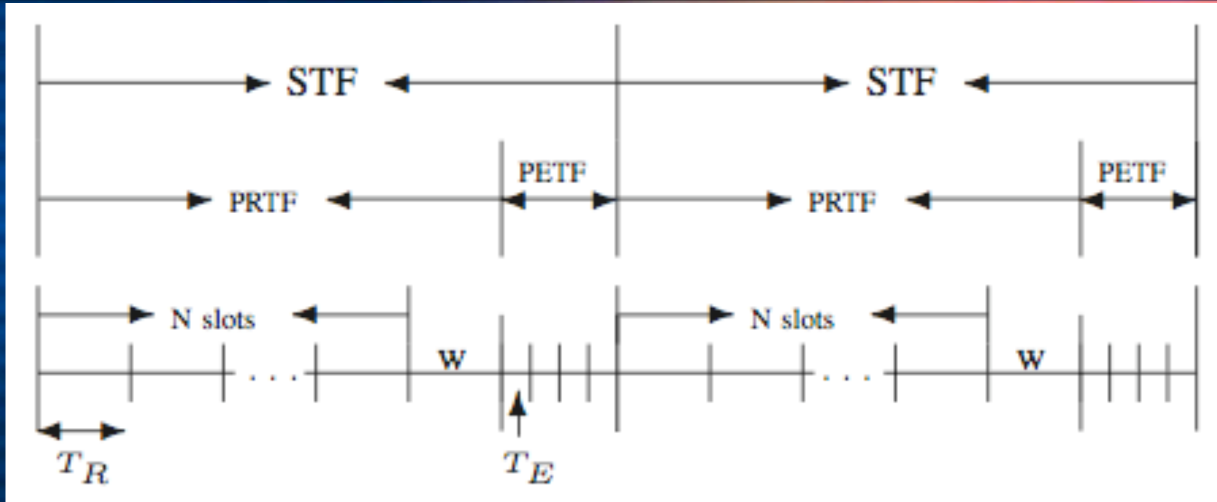


- *E* can be calculated if hardware constants, sample rate, number of neighboring nodes and check time/preamble are known
- Better: *E* can be minimized by varying check time/preamble if constants, sample rate and neighboring nodes are known

B-MAC Results

- Performs better than the other studied protocols in most cases
- System model can be complicated for application and routing protocol developers
- Protocol widely used because has good results even with default parameters

P-MAC - Pattern MAC



- Patterns are 0*1 strings with size 1-N
- Every node starts with 1 as pattern
- Number of 0's grow exponentially up to a threshold δ and then linearly up to N-1
- TR = CW + RTS + CTS + DATA + ACK
- N = tradeoff between latency and energy

Patterns vs Schedules

Local Pattern Bit	Packet to Send	Receiver Pattern Bit	Local Schedule
1	1	1	1
1	1	0	1-
1	0	*	1-
0	1	1	1
0	1	0	0
0	0	*	0

P-MAC Evaluation

- Simulated results are better than SMAC
- Good for relatively stable traffic conditions
- Adaptation to changes on traffic might be slow
- Loose time synchronization required
- Needs more testing and comparison with other protocols besides S-MAC

Z-MAC - Zebra MAC

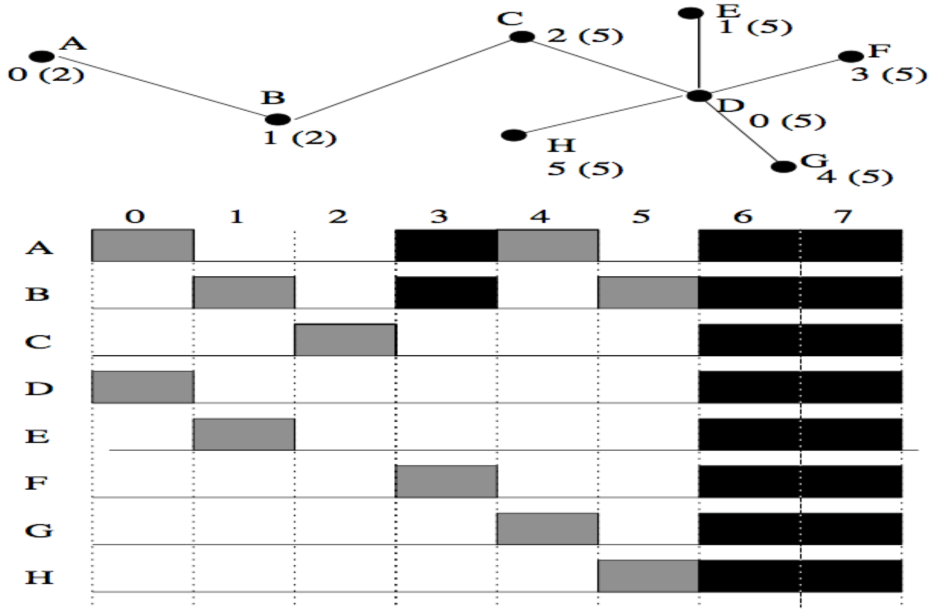
- Runs on top of B-MAC
- Combines TDMA and CSMA features

CSMA	TDMA
• Pros ■ Simple ■ Scalable • Cons ■ Collisions due to hidden terminals ■ RTS/CTS is overhead	• Pros ■ Naturally avoids collisions • Cons ■ Complexity of scheduling ■ Synchronization needed

Z-MAC Initialization

- Neighborhood discovery through ping messages containing known neighbors
- Two-hop neighborhood used as input for a scheduling algorithm (DRAND)
- Running time and message complexity of DRAND is $O(\delta)$, where δ is the two-hop neighborhood size
- The idea is to compensate the initialization energy consumption during the protocol normal operation

Z-MAC Time Slot Assignment



$$2^{a-1} \le F_i < 2^a - 1$$

$$l2^a + s_i (for : l = 0,1,2...)$$

Z-MAC Transmission Control

The Transmission Rule:

- If owner of slot
  - Take a random backoff within To
  - Run CCA and, if channel is clear, transmit
- Else
  - Wait for To
  - Take a random backoff within [To,Tno]
  - Run CCA and, if channel is clear, transmit

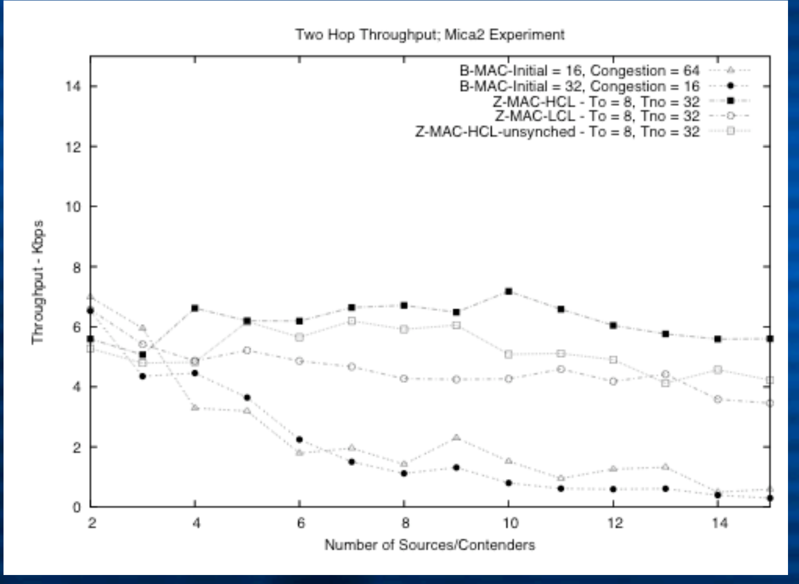
Z-MAC HCL Mode

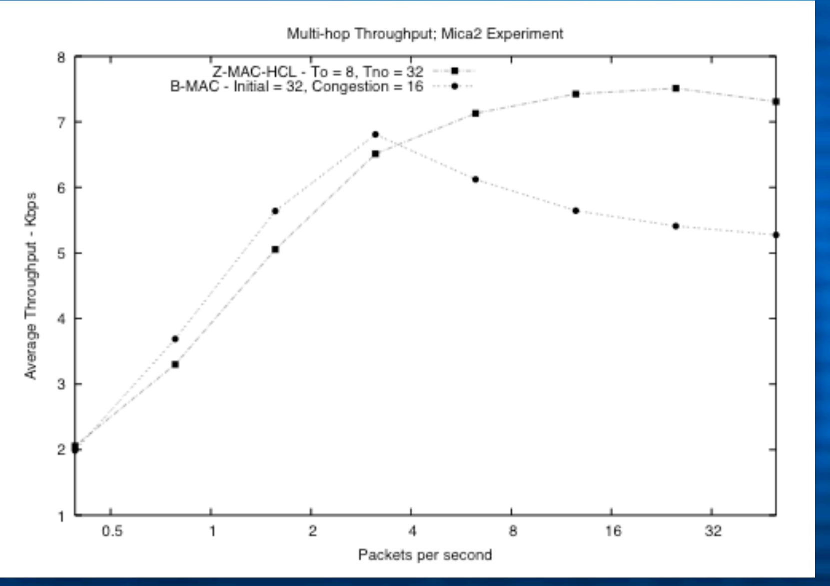
- Nodes can be in *"High Contention Level" (HCL)*
- A node is in HCL only if it recently received an *"Explicit Contention Notification" (ECN)* from a two-hop neighbor
- Nodes in HCL are not allowed to contend for the channel on their two-hop neighbors' time slots
- A node decides to send an ECN if it is losing too many messages (application ACK's) or based on noise measured through CCA

Z-MAC Receiving Schedule

- B-MAC based
- Time slots should be large enough for contention, CCA and one B-MAC packet transmission
- Slot size choice, like in B-MAC, left to application

Z-MAC Results





- Z-MAC performs better than B-MAC when load is high
- As expected, fairness increases with Z-MAC
- Complexity of the protocol can be a problem

Conclusions

- Between the protocols studied, B-MAC still seems to be the best one for applications in general
- Application developers seem not to use B-MAC's control interface
- Middleware service could make such optimizations according to network status

Thank You

Questions or comments?

Thank you for coming!