Embedded Internet and the Internet of Things
WS 12/13

4. MAC Protocols

Prof. Dr. Mesut Güneş
Distributed, embedded Systems (DES)
Institute of Computer Science
Freie Universität Berlin
Overview

• Challenges for MAC protocols
• Contention based protocols
• Reservation based protocols
• Hybrid protocols
Challenges for MAC protocols
Objectives of MAC protocols

- Collision avoidance
- Energy efficiency
- Scalability
- Latency
- Fairness
- Throughput
- Bandwidth utilization
Power consumption

![Power consumption diagram]

- Sensor
- CPU
- TX
- RX
- IDLE
- SLEEP

Radio

Power (mW)

0 5 10 15 20

Prof. Dr. Mesut Güneş • 4. MAC Protocols
Power consumption

![Power consumption graph]

- (a) sleep
- (b) init radio
- (c) radio crystal startup
- (d) μc
- (e) rx
- (f) adc
- (g) sleep

Time (ms)

Current (mA)
Major sources of energy waste

- Idle listening
  - Transmitter
  - Receiver

Common to all wireless networks

- Objective: Reduce energy consumption!
**MAC protocol classes**

- **Fixed assignment protocols**
  - TDMA
  - FDMA
  - CDMA
  - SDMA
  - Available resources are assigned to nodes for long-term usage

- **Demand assignment protocols**
  - Token-passing
  - Polling
  - Available resources are assigned to nodes for short-term usage

- **Random access protocols**
  - ALOHA
  - CSMA
  - Nodes use resources fully distributed without assignment
Challenges for MAC in WSNs

1. WSN Architecture
   - High density of nodes
   - Increased collision probability
   - Minimize signaling overhead to prevent further collisions
   - Sophisticated and simple collision avoidance protocols required

100 nodes on 1000mx1000m

1000 nodes on 1000mx1000m
Challenges for MAC in WSNs

2. Limited energy resources
   • Connectivity and the performance of the network is affected as nodes die
   • Transmitting and receiving consumes almost same energy
   • Frequent power up/down eats up energy
   • Need very low power MAC protocols
   • Minimize signaling overhead
   • Avoid idle listening
   • Prevent frequent radio state changes (active<->sleep)
Challenges for MAC in WSNs

3. Limited processing and memory capabilities
   • Complex algorithms cannot be implemented
   • Conventional layered architecture may not be appropriate
   • Centralized or local management is limited
   • Simple scheduling algorithms required
   • Cross-layer optimization required
   • Self-configurable, distributed protocols required
Challenges for MAC in WSNs

4. Limited packet size
   • Unique node ID is not practical -> IoT?
   • Limited header space
   • Local IDs should be used for inter-node communication
   • MAC protocol overhead should be minimized

5. Cheap encoder/decoder
   • Cheap node requirement prevents sophisticated encoders/decoders to be implemented
   • Simple FEC codes required for error control
   • Channel state dependent MAC can be used to decrease error rate
Challenges for MAC in WSNs

6. Inaccurate clock crystals
   • Cheap node requirement prevents expensive crystals to be implemented
   • Synchronization problems
   • TDMA-based schemes are not practical

7. Event-based networking
   • Observed data depends on physical phenomenon
   • Spatial and temporal correlation in the physical phenomenon should be exploited

Bottom line:
Existing MAC protocols cannot be used for WSNs!
MAC Protocols for WSN

• ?-MAC (pick your letter!)

• µ-MAC, AI-LMAC, B-MAC, Bit, BMA, CC-MAC, CMAC, Crankshaft, CSMA-MPS, CSMA/ARC, DMAC, E2-MAC, EMACs, f-MAC, FLAMA, Funneling-MAC, G-MAC, HMAC, LMAC, LPL, MMAC, MR-MAC, nanoMAC, O-MAC, PACT, PEDAMACS, PicoRadio, PMAC, PMAC, Q-MAC, Q-MAC, QMAC, RATE EST, RL-MAC, RMAC, RMAC, S-MAC, S-MAC/AL, SCP-MAC, SEESAW, Sift, SMACS, SS-TDMA, STEM, T-MAC, TA-MAC, TICER, TRAMA, U-MAC, WiseMAC, X-MAC, Z-MAC, ...
Overview of MAC protocols for WSNs

MAC

Contention based
- S-MAC
- B-MAC
- T-MAC
- X-MAC
- CCMAC

Reservation based
- TRAMA
- FLAMA

Hybrid
- Z-MAC
Contention based protocols
Contention based MAC Protocols

- In fact random based MAC protocol
- Channel access through carrier sense mechanism
- Provide robustness and scalability to the network
- Collision probability increases with node density
  - Good performance with few nodes, bad performance with many nodes
Contention based protocols

CSMA/CA
IEEE 802.11

- IEEE 802.11, “Wireless LAN medium access control (MAC) and physical layer (PHY) specifications,” 1999
- Originally developed for WLANs
IEEE 802.11

- Reminder basic knowledge CSMA/CA (flowchart)

- Distributed Foundation Wireless Medium Access Control - Distributed Coordinated Function CSMA/CA (DFWMAC-DCF) for IEEE 802.11
Basic CSMA/CA

Station senses the channel and it is idle.

Direct access if medium is free \( \geq \text{IFS} \)

Contestation Window (Randomized Back-off Mechanism)

Slot Time (~20\(\mu\)s with DSSS)

Medium Busy

Next Frame

\( t \)

IFS = Inter Frame Spacing
Basic CSMA/CA

• A station with a frame to transmit senses the medium (channel).
• If IDLE -> waits to see if the channel remains idle for a time equal to IFS (Inter-frame spacing). If so, the station may transmit immediately.
• If BUSY -> (either because the station initially finds the channel busy or because the channel becomes busy during the IFS idle time), the station defers transmission and continues to monitor the channel until the current transmission is over.
• Once the current transmission is over, the station delays another IFS.
• If the medium remains idle for this period, the station backs off using a binary exponential backoff scheme and again keeps sensing the medium.
• Backoff scheme
  • The station picks up a random number of slots (the initial value of backoff counter) within a contention window to wait before transmitting its frame.
Basic CSMA/CA

• **MAC** runs a random number generator to set a backoff clock for every contending station
• When the **contention window** starts in which all stations having packets for transmission run down their backoff clocks
• The station with its clock expiring first starts transmission
• Other terminals sense the new transmission and freeze their clocks to be restarted after the completion of the current transmission in the next contention period
CSMA/CA Algorithm

• If Collisions (Control or Data)
  • Binary exponential increase (doubling) of CW
  • Length of backoff time is exponentially increased as the station goes through successive retransmissions
Inter-frame Spaces (IFS)

- Direct access if medium is free $\geq$ DIFS

- DIFS $\sim 50\mu s$: Distributed Coordination Function IFS
- PIFS $\sim 30\mu s$: Point Coordination Function IFS
- SIFS $\sim 10\mu s$: Short Inter Frame Spacing
Inter-frame Spaces (IFS)

• Priorities are defined through different inter frame spaces

• SIFS (Short Inter Frame Spacing)
  • Highest priority packets such as ACK, CTS, polling response
  • Used for immediate response actions
Inter-frame Spaces (IFS)

- **PIFS (PCF IFS):** Point Coordination Function Inter-Frame spacing
  - Medium priority, for real time service using PCF
  - SIFS + One slot time
  - Used by centralized controller in PCF scheme when using polls
Inter-frame Spaces (IFS)

• **DIFS (DCF, Distributed Coordination Function IFS)**
  • Lowest priority, for asynchronous data service
  • SIFS + Two slot times
  • Used as minimum delay of asynchronous frames contending for access
CSMA/CA with ACK

- Station has to wait for DIFS before sending data
- Receiver ACKs immediately
  - Receiver waits for SIFS < DIFS
- Receiver transmits ACK without sensing the medium
- If ACK is lost, retransmission is performed
- Automatic retransmission of data packets in case of transmission errors
CSMA/CA with ACK

![Diagram showing CSMA/CA with ACK protocol]

- **DIFS**: Delays sending for a specific time after detecting a carrier.
- **SIFS**: Short inter-frame space, used to synchronize the receiver.
- **CW**: Contention window, the time frame for which a node can send data without waiting for an acknowledgment.
- **Waiting Time**: Time for other stations to wait before sending data.
- **Data**: The actual data transmitted by the sender.
- **ACK**: Acknowledgment frame sent by the receiver.

**Prof. Dr. Mesut Güneş • 4. MAC Protocols**
Problems with CSMA/CA

- Hidden terminal problem
- Exposed terminal problem
Hidden terminal problem

- A senses the channel free and sends Data
- C cannot hear A and senses the channel free
- Data packet collides at B
Exposed terminal problem

- B sends Data to A (overheard by C)
- C inhibits its transmission to D since channel is busy
- A cannot hear C
  - C-D transmission can actually take place without collisions
CSMA/CA with RTS/CTS

• Transmitter sends an RTS (Request To Send) after medium has been idle for time interval more than DIFS

• Receiver responds with CTS (Clear To Send) after medium has been idle for SIFS

• Then data is transmitted

• RTS/CTS is used for reserving channel for data transmission so that the collision can only occur in control message
CSMA/CA with RTS/CTS

• Use short signaling packets for Collision Avoidance

• RTS Packet (Request To Send, 20 Bytes)
  • A sender requests the right to send from a receiver with a short RTS packet before it sends a data packet

• CTS Packet (Clear To Send, 16 Bytes)
  • The receiver grants the right to send as soon as it is ready to receive

• They contain
  (Sender Address, Receiver Address, Packet Size)
CSMA with RTS/CTS

- Sender
  - DIFS
  - RTS
  - SIFS
  - Data
  - SIFS

- Receiver
  - CTS
  - ACK
  - SIFS

- Other
  - DIFS
  - Contention Window
  - Next Frame
  - SIFS
  - Defer Access
  - Backoff After Defer

Time
Problems with CSMA/CA

- The problems are still not solved
  - Hidden terminal problem
  - Exposed terminal problem
Hidden terminal problem

- A sends RTS
- B sends CTS
- C overhears CTS
- C inhibits its own transmitter
- A successfully sends Data to B

![Diagram of the hidden terminal problem showing communication between nodes A, B, and C with RTS, CTS, and Data exchanges.](image-url)
Hidden terminal problem

- How does C know how long to wait before it can attempt a transmission?
  - A includes length of Data that it wants to send in the RTS packet
  - B includes this information in the CTS packet
  - C, when it overhears the CTS packet, retrieves the length information and uses it to set the inhibition time

- Network Allocation Vector (NAV)
  - MACA protocol used in 802.11
Exposed terminal problem

- B sends RTS to A (overheard by C)
- A sends CTS to B
- C cannot hear A’s CTS
- C assumes A is either down or out of range
- C does not inhibit its transmissions to D
Collisions

- Still possible - RTS packets can collide!

- Binary exponential backoff performed by stations that experience RTS collisions

- RTS collisions not as bad as data collisions in CSMA (since RTS packets are typically much smaller than DATA packets)
CSMA/CA with RTS/CTS

• Both Physical Carrier Sensing and Virtual Carrier Sensing used in 802.11

• If either function indicates that the medium is busy, 802.11 treats the channel to be busy

• Virtual Carrier Sensing is provided by NAV (Network Allocation Vector)
CSMA/CA with RTS/CTS

• Most 802.11 frames carry a duration field which is used to reserve the medium for a fixed time period
• Tx sets the NAV to the time for which it expects to use the medium
• Other stations start counting down from NAV to 0
• As long as NAV > 0, the medium is busy
• CHANNEL VIRTUALLY BUSY -> a NAV is turned on!
• The transmission will be delayed until the NAV expires
• When the channel is virtually available, then MAC checks for PHY condition of the channel
CSMA/CA with RTS/CTS

Sender

Receiver

NAV

RTS

CTS

DATA

ACK

SIFS

SIFS

SIFS

SIFS

Prof. Dr. Mesut Güneş • 4. MAC Protocols
CSMA/CA with RTS/CTS

• If receiver receives RTS, it sends CTS (Clear to Send) after SIFS

• CTS again contains duration field and all stations receiving this packet need to adjust their NAV

• Sender can now send data after SIFS, acknowledgement via ACK by receiver after SIFS
CSMA/CA with RTS/CTS

• Every station receiving the RTS that is not addressed to it, will go to the Virtual Carrier Sensing Mode for the entire period identified in the RTS/CTS communication, by setting their NAV signal on.

• Network Allocation Vector (NAV) is set in accordance with the duration of the field.

• NAV specifies the earliest point at which the station can try to access the medium.

• Thus, the source station sends its packet without contention.

• After completion of the transmission, the destination terminal sends an ACK and NAV signal is terminated, opening the contention for other users.
CSMA/CA with RTS/CTS

Sender

Receiver

NAV

Defer access
Contestion based protocols

S-MAC
S-MAC: Sensor-MAC

• Problem: “Idle Listening” consumes significant energy
• Solution: Periodic listen and sleep

During sleeping, radio is turned off
• Important parameter: “duty cycle” \( \rightarrow \) duty cycle = \( \frac{t_l}{t_p} \)
• Reduce duty cycle to \( \sim 10\% \)
  • Listen for 200ms and sleep for 1.8s
• Property: High latency, low energy
S-MAC

• Each node goes into periodic sleep mode during which it switches the radio off and sets a timer to awake later.

• When the timer expires it wakes up and listens to see if any other node wants to talk to it.

• The duration of the sleep and listen cycles are application dependent and they are set the same for all nodes.

• Requires a periodic synchronization among nodes to take care of any type of clock drift.
Periodic Sleep and Listen

- All nodes are free to choose their own listen/sleep schedules
- To reduce control overhead, neighboring nodes are synchronized together
- They listen at the same time and go to sleep at the same time
  - \( \rightarrow \) synchronized sleep
Synchronization

• SYNC packets are exchanged periodically to maintain schedule synchronization.

SYNC Packet

Sender Node ID

Next Sleep Time

• Synchronization period:
  • Period for a node to send a SYNC packet
  • Receivers will adjust their timer counters immediately after they receive the SYNC packet
Periodic listen and sleep
Periodic listen and sleep

- **listen**
- **sleep**
- **SYNCH**
- **RTS**
- **CTS**
Maintaining synchronization

Receiver

Listen

<table>
<thead>
<tr>
<th>for SYNC</th>
<th>for RTS</th>
<th>for CTS</th>
<th>Sleep</th>
</tr>
</thead>
</table>

Sender 1

Tx SYNC

CS

Sleep

Only SYNC packet

Sender 2

Tx RTS

Got CTS

CS

Send data

Unicast data packet

Sender 3

Tx SYNC

Tx RTS

Got CTS

CS

Send data

SYNC and Data packet
Choosing and maintaining schedules

- Each node maintains a schedule table that stores schedules of all its known neighbors

- For initial schedule:
  - A node first listens to the medium for a certain amount of time (at least the synchronization period)
  - If it does not hear a schedule (SYNC packet) from another node, it randomly chooses a schedule and broadcasts its schedule with a SYNC packet immediately
  - This node is called a Synchronizer

- If a node receives a schedule from a neighbor before choosing its own schedule
  - It follows this neighbor’s schedule
  - Becomes a Follower
  - Waits for a random delay and broadcasts its schedule
Coordinated sleeping

- In a large network, we cannot guarantee that all nodes follow the same schedule.
- The node on the border will follow both schedules.
- When it broadcasts a packet, it needs to do it twice, first for nodes on schedule 1 and then for those on schedule 2.
Coordinated sleeping

Virtual clusters in S-MAC

- Synchronizer
- Follower
- Border node
Border nodes

• Border nodes have less time to sleep and consume more energy than others.

• Option: Let border nodes adopt only one schedule
  • -> First received
Collision avoidance

• **S-MAC** is based on contention, i.e., if multiple neighbors want to talk to a node at the same time, they will try to send when the node starts listening.

• Similar to IEEE802.11, i.e., use RTS/CTS mechanism to address the hidden terminal problem

• Perform carrier sense before initiating a transmission
Collision avoidance

- If a node fails to get the medium, it goes to sleep and wakes up when the receiver is free and listening again.
- Broadcast packets are sent without using RTS/CTS.
- Unicast data packets follow the sequence of RTS/CTS/DATA/ACK between the sender and receiver.
- Duration field in each transmitted packet indicates how long the remaining transmission will be so if a node receives a packet destined to another node, it knows how long it has to keep silent.
- The node records this value in network allocation vector (NAV) and sets a timer for it.
- When a node has data to send, it first looks at NAV.
- If this value is not zero, then medium is busy (virtual carrier sense).
Collision avoidance

• The medium is determined as free if both virtual and physical carrier sense indicate the medium is free.

• All immediate neighbors of both the sender and receiver should sleep after they hear RTS or CTS packet until the current transmission is over.
Adaptive listening feature

- Reduce multi-hop latency due to periodic sleep
  - BASIC IDEA: Let the node which overhears its neighbor’s transmission stay awake
- Both neighbors will learn about how long the transmission is from the duration field in the RTS and CTS packets
- They are able to adaptively wake up when the transmission is over
- Reduce latency by at least half (e.g., CTS of 2 is heard by 3 also, 3 remains awake!!)
Message passing feature

• Long messages are broken down into smaller packets and sent continuously once the channel is acquired by RTS/CTS handshake.

• Increases the sleep time, but leads to fairness problems.
S-MAC: Example

- **Topology**
  - Two-hop network with two sources and two sinks

- **Sources periodically generate a sensing message which is divided into fragments**

- **Traffic load is changed by varying the inter-arrival period of the messages:**
  - For inter-arrival period of 5s, message is generated every 5s by each source
  - Here inter-arrival period varies between 1-10s
S-MAC: Example

- In an experiment, a source node generates 10 messages

- Each message has 10 fragments
  - 200 data packets to be passed from source to sink nodes

- Each fragment has 40 bytes
  - 8000 bytes at all

- The total energy consumption of each node is measured for sending this fixed amount of data
Experiments

- S-MAC consumes much less energy than 802.11-like protocol without sleeping.
- At heavy load, idle listening rarely happens, energy savings from sleeping is very limited. SMAC achieves energy savings by avoiding overhearing and efficiently transmitting long messages.
- At light load, periodic sleeping plays the key role.

**Average energy consumption in the source nodes A&B**

**Notice:** small value -> heavy traffic load
Energy consumption over multi-hops

- Ten-hop linear network at different traffic
  - Source
    1 → 2 → 3 → ... → 10 → 11
  - Sink

- Load
  - inter-arrival time 0-10s
  - source node sends 20 messages with each 100 bytes long

- 3 configurations for S-MAC
  - No sleep cycles
  - 10% duty cycle without adaptive listening
  - 10% duty cycle with adaptive listening

- Periodic listen interval 115ms
  - 10% duty cycle means a frame length for 1.15sec
Energy consumption over multi-hops

• At light traffic load, periodic sleeping has significant energy savings over fully active mode.

• Adaptive listen saves more at heavy load by reducing latency.

![Graph showing aggregate energy consumption at different traffic load in the entire network.](image)
Latency as hops increase

- Adaptive listen significantly reduces latency caused by periodic sleeping.

![Latency under lowest traffic load graph](image)

![Latency under highest traffic load graph](image)
Throughput as hops increase

- Adaptive listen significantly increases throughput
- Uses less time to pass the same amount of data
S-MAC: Summary

• A mainly static network is assumed
• Trades off latency for reduced energy consumption
• Redundant data is still sent with increased latency
• Increased collision rate due to sleep schedules
Contention based protocols

B-MAC
B-MAC: Berkeley MAC

• Properties
  • Keep core MAC simple
  • Provides basic CSMA access
  • Optional link level ACK -> no link level RTS/CTS
  • CSMA backoffs configurable by higher layers
  • Carrier sensing using Clear Channel Assessment (CCA)
  • Sleep/Wake scheduling using Low Power Listening (LPL)
Goals of B-MAC

• Low Power Operation
• Effective Collision Avoidance
• Simple Implementation, Small Code and RAM Size
• Efficient Channel Utilization
• Reconfigurable by Network Protocols
• Tolerant to Changing RF/Networking Conditions
• Scalable to Large Numbers of Nodes

Comparison of the size of B-MAC and S-MAC in bytes. Both protocols are implemented in TinyOS.

<table>
<thead>
<tr>
<th>Protocol</th>
<th>ROM</th>
<th>RAM</th>
</tr>
</thead>
<tbody>
<tr>
<td>B-MAC</td>
<td>3046</td>
<td>166</td>
</tr>
<tr>
<td>B-MAC w/ ACK</td>
<td>3340</td>
<td>168</td>
</tr>
<tr>
<td>B-MAC w/ LPL</td>
<td>4092</td>
<td>170</td>
</tr>
<tr>
<td>B-MAC w/ LPL &amp; ACK</td>
<td>4386</td>
<td>172</td>
</tr>
<tr>
<td>B-MAC w/ LPL &amp; ACK + RTS-CTS</td>
<td>4616</td>
<td>277</td>
</tr>
<tr>
<td>S-MAC</td>
<td>6274</td>
<td>516</td>
</tr>
</tbody>
</table>
B-MAC TinyOS Interfaces

• B-MAC Design
  • Clear Channel Assessment (CCA)
  • Packet Backoffs
  • Link Layer Acknowledgments
  • Low Power Listening (LPL)

• TinyOS Interface
  • Flexible control of B-MAC by higher layer services
  • Allow services to toggle CCA and ACKs
  • Set backoffs on a per message basis
  • Change the LPL mode for transmit and receive

interface MacControl {
    command result_t EnableCCA();
    command result_t DisableCCA();
    command result_t EnableAck();
    command result_t DisableAck();
    command void* HaltTx();
}

interface MacBackoff {
    event uint16_t initialBackoff(void* msg);
    event uint16_t congestionBackoff(void* msg);
}

interface LowPowerListening {
    command result_t SetListeningMode(uint8_t mode);
    command uint8_t GetListeningMode();
    command result_t SetTransmitMode(uint8_t mode);
    command uint8_t GetTransmitMode();
    command result_t SetPreambleLength(uint16_t bytes);
    command uint16_t GetPreambleLength();
    command result_t SetCheckInterval(uint16_t ms);
    command uint16_t GetCheckInterval();
}
Clear Channel Assessment

• Effective collision avoidance

• Find out whether the channel is idle
  • If too pessimistic: waste bandwidth
  • If too optimistic: more collisions

• Key observation
  • Ambient noise may change significantly depending on the environment
  • Packet reception has fairly constant channel energy
  • Need to tell what is noise and what is a signal

• Software approach to estimating the noise floor
  \[\rightarrow\text{B-MAC solution!}\]
Clear Channel Assessment

- Take a signal strength sample when the channel is assumed to be free/idle
  - WHEN?
  - Right after a packet is transmitted or when no valid data is received

- Samples are entered into a FIFO queue
Clear Channel Assessment

• Median of the queue $S_t$ is added to an exponentially weighted moving average with decay $\alpha$

• Median signal strength is used as a simple low pass filter to add robustness to the noise floor estimate $A_t$

$$A_t = \alpha S_t + (1 - \alpha)S_{t-1}$$

• Assumptions: $\alpha=0.06$ and FIFO queue size=10
• Once a good estimate of the noise floor is established, a request to transmit a packet starts the process of monitoring the received signal from the radio
Single-Sample Thresholding vs Outlier Detection

• Common approach: take single sample, compare to noise floor
  • Large number of false negatives \( \rightarrow \) lower effective channel BW

• B-MAC: search for outliers in received signal (RSSI)
  • If a sample has significantly lower energy than the noise floor during the sampling period, then the channel is clear
  • If 5 samples are taken and no outlier is found, the channel is busy

Prof. Dr. Mesut Güneş • 4. MAC Protocols
CCA vs. Threshold Techniques

Signal Strength Indicator (RSSI) from transceiver

Idle

Threshold

Outlier

Time (ms)

Prof. Dr. Mesut Güneş • 4. MAC Protocols
CCA vs. Threshold Techniques

- Threshold: waste channel utilization
- CCA: Fully utilize the channel since a valid packet could have no outlier significantly below the noise floor
- A packet arrives between 22 and 54ms.
  - The middle graph shows the output of a thresholding CCA algorithm. (1: channel clear, 0: channel busy)
  - Bottom shows the output of an outlier detection algorithm
Clear Channel Assessment

• Before transmission, take a sample of the channel
• If the sample is below the current noise floor, channel clear, send immediately
• If five samples are taken, and no outlier found -> channel busy, take a random backoff
• Noise floor updated when the channel is known to be clear, e.g., just after packet transmission
• CCA can be turned on/off (see B-MAC-TinyOS interface)
• If turned off, a schedule-based protocol can be implemented above B-MAC
• If turned on, B-MAC uses an initial channel backoff when sending a packet
Clear Channel Assessment

• B-MAC does not set the backoff time, instead an event is signaled to the service that sent the packet via the MacBackoff interface
• The service may either return an initial backoff time or ignore the event
• If ignored, a small random backoff is used
• After the initial backoff, the CCA outlier algorithm is run
• If the channel is not clear, an event signals the service for a congestion backoff time
• If no backoff time is given, again a small random backoff is used
• Enabling or disabling CCA and configuring the backoff allows services to change the fairness and available throughput
Low Power Listening

- **Goal:** Minimize “Listen Cost”
- **Principles**
  - Node periodically wakes up, turns radio on and checks activity on the channel
    - Wakeup time fixed (time spend sampling RSSI?)
    - “Check time” variable
  - If energy/activity on the channel is detected, node powers up and stays awake for the time required to receive the incoming packet
Low Power Listening

- Node goes back to sleep
  - If the packet is received successfully
  - After a timeout (if no packet received (a false positive))
- Preamble length matches channel checking period
  - No explicit synchronization required
- Noise floor estimation used to detect channel activity during LPL
Check Interval for Channel Activity

• To reliably receive data, the preamble length is matched to the interval that the channel is checked for activity.

• If the channel is checked for every 100 ms, the preamble must be at least 100 ms long for a node to wake up, detect activity on the channel, receive the preamble and then receive the message.

• Interval between LPL samples is maximized so that the time spent sampling the channel is minimized.

• Transmit mode ~ Preamble length

• Listening mode ~ Check interval
**LPL - Preamble Sampling**

Preamble is not a packet but a physical layer RF pulse (Minimize overhead)

---

**Diagram:**

Sender (S) sends a Preamble, followed by Send data. The receiver (R) samples the Preamble and then becomes active to receive a message. The condition for Preamble sampling is:

\[ |\text{Preamble}| \geq \text{Sampling period} \]
LPL Check Interval

• Sampling rate (traffic pattern) defines optimal check interval

• Check interval
  • Too small: energy wasted on idle listening
  • Too large: energy wasted on transmissions (long preambles)

• In general, it is better to have larger preambles than to check more often!

• More frequent checking of the radio
  • Shorter transmission time
  • More energy consumption
LPL Check Interval

![Graph showing LPL Check Interval](image)
Experimental Results: Throughput

- “B-MAC is about 4.5 faster than SMAC-unicast”
- Not as fast when ACK or RTS/CTS is used
- Differences less pronounced as number of nodes increases
- Another issue: B-MAC has CCA, thus it backs off less frequently (and perhaps the backoff timer is faster)
- What about hidden terminal without RTS/CTS?
Throughput vs Power Consumption

- 10 nodes in a neighborhood
- Data must arrive within 10 seconds
- Average power consumption per node
Throughput vs Power Consumption

- Low data rates: S-MAC is better
  - Very low duty cycle
- Power vs throughput
  - S-MAC: linear
  - B-MAC: sub-linear
- Reason: S-MAC duty cycle must increase
  - More active periods, more SYNC periods
  - Cost of resynchronizing?
- B-MAC: larger preambles at low throughput, progressively becoming smaller

<table>
<thead>
<tr>
<th>Length (bytes)</th>
<th>B-MAC</th>
<th>S-MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preamble</td>
<td>8</td>
<td>18</td>
</tr>
<tr>
<td>Synchronization</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Header</td>
<td>5</td>
<td>9</td>
</tr>
<tr>
<td>Footer (CRC)</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Data Length</td>
<td>29</td>
<td>29</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>46</strong></td>
<td><strong>60</strong></td>
</tr>
</tbody>
</table>
Energy vs Latency

- 10-hop network
- Source sends a 100-byte packet every 10 seconds
- S-MAC: 10% Duty Cycle with Adaptive Listening
- B-MAC: choose optimal check interval
- SMAC again performs worse...
  - Reason: sync packets, prob. of multiple schedules→less time to sleep
### Comparison of S-MAC and B-MAC

<table>
<thead>
<tr>
<th>Feature</th>
<th>S-MAC</th>
<th>B-MAC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collision avoidance</td>
<td>CSMA/CA</td>
<td>CSMA</td>
</tr>
<tr>
<td>ACK</td>
<td>Yes</td>
<td>Optional</td>
</tr>
<tr>
<td>Message passing</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Overhearing avoidance</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Listen period</td>
<td>Pre-defined + adaptive listen</td>
<td>Pre-defined</td>
</tr>
<tr>
<td>Listen interval</td>
<td>Long</td>
<td>Very short</td>
</tr>
<tr>
<td>Schedule synchronization</td>
<td>Required</td>
<td>Not required</td>
</tr>
<tr>
<td>Packet transmission</td>
<td>Short preamble</td>
<td>Long preamble</td>
</tr>
<tr>
<td>Code size</td>
<td>6.3KB</td>
<td>4.4KB (LPL &amp; ACK)</td>
</tr>
</tbody>
</table>
Contention based protocols

T-MAC
**T-MAC**

- **S-MAC** is based on static frame length

  ![S-MAC Diagram]

- **T-MAC** improves energy efficiency by using an adaptive duty cycle
  - Node goes in sleep mode after TA sec idle time (no traffic)

  ![T-MAC Diagram]
T-MAC

- T-MAC is an improvement on S-MAC
- All traffic is burst at the begin of an listen interval
- How to determine idle time $TA$?

$$TA > C + R + T$$

- $C$ Length of contention interval
- $R$ Time of RTS packet transfer
- $T$ Turnaround time between RTS and CTS
Contention based protocols

WiseMAC
WiseMAC


- Another problem with basic preamble sampling
  - Sender has to wait for the preamble and the wakeup of receiver
  - Sender wastes resources if receiver sleeps during preamble transmission

- Improvement
  - Sender transmits preamble just when receiver wakes up
  - Problem: How to know when receiver wakes up?
  - Approach: Sender learns the schedule of the receiver
Receiver adds schedule information into ACK
WiseMAC
Contention based protocols
PAMAS
PAMAS: Power Aware Multiaccess with Signaling

- **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

![Diagram showing PAMAS protocol](image)
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

Similarly: Ongoing transmission near B destroys RTS by busy tone
Reservation based protocols

TRAMA
Reservation based protocols

- Reservation based
  - First reserve a transmission slot
  - Then transmit

Traffic Adaptive Medium Access (TRAMA)

Collision free
TRAMA

• A time-slotted structure

115.2 kbps -> transmission slot 46ms (512-byte segments)
TRAMA

- Scheduled access period
  - Used for contention free data transmission
  - Supports unicast, multicast, and broadcast communication

- Random access period
  - Used for signalling
  - Synchronization and updating of two-hop neighbourhood information
  - Collision!!
TRAMA

- Transmission slots:
  - For collision free data exchange
  - For schedule propagation
  - Nodes use SEP (Schedule Exchange Protocol) to
    - exchange traffic based information or schedules with neighbors

- Signaling slots:
  - are used by the Neighbor protocol (NP)
  - Propagate one-hop neighbor information among neighboring nodes during the random access period
  - A consistent two-hop topology information across all nodes is obtained
TRAMA Components

• Neighbor Protocol (NP)
  • Gather 2-hop neighborhood information

• Schedule Exchange Protocol (SEP)
  • Gather 1-hop traffic information for SCHEDULING

• Adaptive Election Algorithm (AEA)
  • Select transmitters, receivers for current time slot leave other nodes in liberty to switch to low power mode using the NP and SEP results
Neighbor Protocol (NP)

- **Main Function:**
  - Gather two-hop neighborhood information by using signaling packets during the random access period

- If no updates, signaling packets are sent as “keep alive” beacons

- A node times out if nothing is heard from its neighbor

- Updates retransmitted to guarantee packet delivery
### Packet Formats

<table>
<thead>
<tr>
<th>Type</th>
<th>SourceAddr</th>
<th>DestAddr</th>
<th>DeleteNum</th>
<th>AddNum</th>
<th>Deleted NodeID’s</th>
<th>Added NodeID’s</th>
</tr>
</thead>
</table>

(a) Signal Header

<table>
<thead>
<tr>
<th>Type</th>
<th>SourceAddr</th>
<th>DestAddr</th>
<th>Timeout</th>
<th>NumSlots</th>
<th>Bitmap</th>
</tr>
</thead>
</table>

Short Schedule Summary

(b) Data Header
Schedule Exchange Protocol (SEP)

• Each node computes a SCHEDULE INTERVAL (SCHED) based on the rate at which packets are produced.
• SCHED represents number of slots for which the node can announce the schedule to its neighbors according to its current state (queue)
• The node pre-computes number of slots in the interval 

\[ [t, t+SCHED] \]

for which it has the highest priority among its two-hop neighbors (contenders) → WINNING SLOTS
Schedule Exchange Protocol (SEP)

- The node announces the intended receivers for these slots
- The last winning slot is used for broadcasting the node’s schedule for the next interval
- If these winning slots cannot be filled by the node the remaining vacant slots can be released to other nodes

Example: Node u -> SCHED is 100 slots.
- During time slot 1000, u computes its winning slots between [1000, 1100]
- Assume: These slots are 1009, 1030, 1033, 1064, 1075, 1098
- u uses slot 1098 to announce its next schedule by looking ahead from [1098, 1198]
Schedule Exchange Protocol (SEP)

• Nodes announce their schedules via SCHEDULE PACKETS

• BITMAP: with the length equal to number of one-hop neighbors

• Each bit corresponds to one particular receiver

• Example: One node with 4 neighbors 14,7,5, and 4
  • BITMAP -> size 4
  • For broadcast: all bitmap bits are set to 1
Schedule Packet Format

<table>
<thead>
<tr>
<th>Field</th>
<th>SourceAddr</th>
<th>Timeout</th>
<th>Width</th>
<th>NumSlots</th>
<th>Bitmaps</th>
<th>.......</th>
<th>.......</th>
<th>Width x Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field Size (bits)</td>
<td>32</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Bitmap (Width = 4)

Reserved for particular set of receivers depending on the node traffic.

Does not have data, these slots could be used by other one-hop neighbors with data.

Reserved for announcing the schedule.

RESERVED  GIVEUP

ChangeOver Slot

Prof. Dr. Mesut Güneş • 4. MAC Protocols
Schedule Packet Format

- Number of slots for which the schedule is valid
- Length of the neighbor bitmap (number of one hop neighbors)
- Total number of winning slots (number of bitmaps contained in the packet)

### Field Size (bits)

<table>
<thead>
<tr>
<th>Field</th>
<th>SourceAddr</th>
<th>Timeout</th>
<th>Width</th>
<th>NumSlots</th>
<th>Bitmaps</th>
<th>......</th>
<th>......</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field Size</td>
<td>32</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>Width x Size</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Bitmap (Width = 4)

- 1010: Reserved for particular set of receivers depending on the node traffic.
- 00100000: Does not have data, these slots could be used by other one-hop neighbors with data.
- 00001111: Reserved for announcing the schedule.

- ChangeOver Slot
- GIVEUP
- RESERVED
Adaptive Election

• Given: Each node knows its two-hop neighborhood and their current schedules
• How to decide which slot (in scheduled access period) a node can use?
  • Use node identifier $u$ and globally known hash function $h$
  • For time slot $t$, compute priority

\[
prio(u, t) = h(u \oplus t)
\]

• Compute this priority for next $k$ time slots for node itself and all two-hop neighbors
• Node uses these time slots for which it has the highest priority
Simulation Results

- Broadcast traffic using Poisson arrivals
- 50 nodes, 500mx500m area, 512 byte data, Average node density: 6
Energy Savings

![Graph showing energy savings comparison between different MAC protocols. The x-axis represents the mean interarrival time (in seconds), and the y-axis represents the percentage sleep time. The graph compares TRAMA-bcast, TRAMA-ucast, SMAC-10-bcast, SMAC-10-ucast, SMAC-50-bcast, and SMAC-50-ucast.]
TRAMA: Summary

- Complex election algorithm and data structure
- Overhead due to explicit schedule propagation
- Higher queueing delay
Hybrid protocols

Z-MAC
Z(ebra)-MAC

• Combines the strengths of both CSMA and TDMA at the same time offsetting their weaknesses
• High channel efficiency and fair

<table>
<thead>
<tr>
<th>Channel Utilization</th>
<th>Low Contention</th>
<th>High Contention</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSMA</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>TDMA</td>
<td>Low</td>
<td>High</td>
</tr>
</tbody>
</table>
Effective throughput: CSMA vs. TDMA

Channel utilization vs. Number of contenders

- CSMA
- TDMA
- IDEAL
Z-MAC

• Uses TDMA schedule created by DRAND as a “hint” to schedule transmissions
  • Distributed RAND

• The owner of a time-slot always has priority over the non-owners while accessing the medium

• Unlike TDMA, non-owners can “steal” the time-slot when the owners do not have data to send
Z-MAC

• This enables Z-MAC to switch between CSMA and TDMA depending on the level of contention

• Under low contention
  • Z-MAC acts like CSMA
  • i.e., high channel utilization and low latency

• Under high contention
  • Z-MAC acts like TDMA
  • i.e., high channel utilization, fairness, and low contention overhead
DRAND-Algorithm
Distributed TDMA Scheduling

Radio Interference Map

Input Graph → DRAND Slot Assignment

Prof. Dr. Mesut Güneş • 4. MAC Protocols
DRAND-Algorithm: Successful Round

Step 1: Broadcast Request

Step 2: Receive Grants

Step 3: Broadcast Release

Step 4: Broadcast Two Hop Release
DRAND-Algorithm: Unsuccessful Round

Step 1: Broadcast Request

Step 2: Receive Grants from A, B, D but Reject from E

Step 3: Broadcast Fail
Local Frames

• After DRAND, each node needs to decide on the frame size

• Conventional wisdom: Synchronize with rest of the network on Maximum Slot Number (MSN) as the frame size

• Disadvantage:
  • MSN must be broadcast across the entire network.
  • Unused slots if neighborhood small, e.g., A and B would have to maintain frame size of 8, in spite of having small neighborhood
Label is the assigned slot, number in parenthesis is maximum slot number within two hops
Local Frames

• Time Frame Rule (TF Rule)
  • Let node $i$ be assigned to slot $s_i$, according to DRAND and MSN within two hop neighborhood be $F_i$, then $i$'s time frame is set to be $2^a$, where positive integer $a$ is chosen to satisfy condition

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

• In other words, $i$ uses the $s_i$-th slot in every $2^a$ time frame
  • $i$'s slots are $L \cdot 2^a + s_i$, for all $L=1,2,3,\ldots$
Local Frames: Example

A → B → C → D

0(2) → 1(2) → 2(5) → 0(5)

E 1(5)
F 3(5)
G 4(5)
H 5(5)

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>H</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Transmission Control

• Slot Ownership
  • If current timeslot is the node's assigned time-slot, then it is the Owner, and all other neighbour nodes are Non-Owners

• Low Contention Level: Nodes compete in all slots, albeit with different priorities. Before transmitting:
  • Owner: Backoff = Random(T_o)
  • Non-Owner: Backoff = T_o + Random(T_no)
  • After backoff, sense channel, if busy repeat above, else send.

• Switches between CSMA and TDMA automatically depending on contention level

• Performance depends on specific values of T_o and T_no
  • From analysis, T_o = 8 and T_no = 32 are used for best performance
Transmission Control

- Ready to Send, Start Random($T_0$) Backoff
- Ready to Send, Start To + Random($T_{no}$) Backoff
- After Backoff, CCA Idle
- After Backoff, CCA Busy

<table>
<thead>
<tr>
<th>Time Slots</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>A(0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B(1)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Owner Backoffs

Non-Owner Backoffs
Explicit Contention Notification (ECN)

• Informs all nodes within two-hop neighborhood not to send during its time-slot

• When a node receives ECN message, it sets its HCL flag

• ECN is sent by a node if it experiences high contention

• High contention detected by lost ACKs or congestion backoffs.

• On receiving one-hop ECN from i, forward two-hop ECN if it is on the routing path from i.
Explicit Contention Notification

- C experiences high contention
- C broadcasts one-hop ECN message to A, B, D.
- A, B not on routing path, so discard ECN.
- D on routing path, so it forwards ECN as two-hop ECN message to E, F.
- Now, E and F will not compete during C's slot as Non-Owners.
- A, B and D are eligible to compete during C's slot, albeit with lesser priority as Non-Owners.
Two-Hop Experiments

• Setup: Two-Hop
  • Dumbbell shaped topology
  • Transmission power varied between low (50) and high (150) to get two-hop situations.
• Aim – See how Z-MAC works when Hidden Terminal Problem manifests itself.
Experimental Setup - Testbed

- 40 Mica2 sensor motes
- Wall-powered and connected to the Internet via Ethernet ports
- Programs uploaded via the Internet, all mote interaction via wireless
- Links vary in quality, some have loss rates up to 30-40%
- Asymmetric links also present (14→15)
Single-Hop Results: Throughput

One Hop Throughput; Mica2 Experiment

Throughput - Kbps

Number of Sources/Contenders

Z-MAC

B-MAC

Prof. Dr. Mesut Güneş • 4. MAC Protocols
Multi Hop Results: Throughput

Multi-hop Throughput; Mica2 Experiment

- Z-MAC-HCL: To = 8, Tno = 32
- B-MAC: Initial = 32, Congestion = 16

Average Throughput - Kbps

Packets per second
Multi Hop Results: Energy Efficiency

Throughput/Energy

- Z-MAC-HCL - To = 8, Tno = 32
- B-MAC - Initial = 32, Congestion = 16

Throughput/Energy - Kbits/mJ

- Z-MAC
- B-MAC

Packets per second

Prof. Dr. Mesut Güneş • 4. MAC Protocols
Z-MAC: Summary

- Hybrid protocol
  - Improved throughput in high traffic load
  - Acceptable latency in low traffic
IEEE 802.15.4
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

---

![Superframe Structure Diagram]
Summary
Summary

• Plenty of MAC protocols

• Discussed
  • Contention-based protocols
  • Reservation-based protocols
  • Hybrid protocols

• Open issues
  • Mobility support
  • Real-time communication
  • Which MAC suites best an application?
  • How to compare different MAC protocols?
  • Influence of MAC to higher layers!


