Medium Access Control in Wireless IoT

Davide Quaglia, Damiano Carra
LIVELLO DATALINK
Goals

• Reliable and efficient communication between two nodes on the same physical medium
  – Cable (Wired)
  – Wireless

• Assumptions from the lower physical layer:
  – The concept of *bit* is defined
  – Bits, if received, arrive in the same order in which they have been transmitted
Functionality

- **Framing** = Bit grouping into layer-2 PDUs
- Error checking
- Ack and retransmission of corrupted/lost PDUs (not in all protocols)
- Policy of use of the channel if more than 2 nodes share the same physical medium
  - Node addressing
  - Channel arbitration
Services provided to the upper network layer

- Un-acknowledged connection-less service (e.g. Ethernet/IEEE802.3)
- Acknowledged connection-less service (e.g. WiFi/IEEE802.11, IEEE802.15.4)
- Connection-oriented service (e.g. IEEE802.16)

**REMARK:** the connection-oriented service is also acknowledged and furthermore it provides flow control
Framing

• Improve channel utilisation in case of more than two nodes sharing it
• Requested to check errors and recover PDUs
  – Error detection must be performed on blocks of bits (e.g. CRC)
  – The corrupted PDU can be retransmitted
• Issue: definition of start/end of frame
Framing

Sending machine

Packet

Header Payload field Trailer

Frame

Receiving machine

Packet

Header Payload field Trailer
Start/end of frame

- We need to use symbols which are not used to send data otherwise a sequence of data bits could be considered erroneously a start/end of frame
  - Physical signal configurations which are not used for data bits
    - Specific configuration choices can improve bit synchronization between TX and RX
  - Particular sequence of data bit values (FLAG)
    - Bit stuffing/de-stuffing is needed to avoid FLAG simulation in the PDU
  - Inter-packet gap minimum between 2 consecutive frames
Bit stuffing/de-stuffing

• Example taken from HDLC protocol
• Byte 01111110 is used as FLAG at the beginning/end of each frame
• The bits of the original frame are modified through stuffing
  – After five “1”s a “0” is automatically inserted
• At the receiver the Data Link layer operates de-stuffing
Bit stuffing: example

(a) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

(b) 0 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 1 1 1 1 0 1 0 0 1 0

(c) 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0

a) Original data from upper layer
b) Data transmitted on the wire or over-the-air
c) Data at the receiver after de-stuffing.
Error detection

• Some bits may have incorrect values at the RX
  – Interference, low-level signal
  – Often errors are not isolated but group into burst

• Hamming distance

• Redundant information must be added to the message to check errors
  – m bits of the original message
  – r bits of the code for error detection
  – n=m+r bits transmitted on the channel
  – Code rate = m/n

• Examples
  – Parity Bit
  – Checksum
  – Circular Redundancy Check (CRC)
Parity bit

• At the TX a bit is appended to the message
  – “1” if the amount of “1” in the message is even
  – “0” if the amount of “1” in the message is odd

• At the RX if the amount of “1” is even then at least one bit flipped its values
  – One bit or an odd number of bits (we cannot distinguish)
  – Errors affecting an even number of bits are not detected
Check sum

• Extension of the concept of parity bit
• The message is decomposed into \( r \) bit words
• The words are summed and overflow is not taken into account
• The sum (another \( r \)-bit word) is appended to the message
• The sum is recomputed at the RX
  • If it is different from the appended value an error occurred
• Errors are not detected if they affect different bits that do not change the sum
Circular Redundancy Check (CRC)

- The message is seen as the coefficients vector of a polynomial $M(x)$ having degree $m-1$
- Let $R(x)$ be the remainder of the polynomial division $x^rM(x)/G(x)$ where $G(x)$ is named *generating polynomial*
- By construction the polynomial $x^rM(x)-R(x)$ is exactly divided by $G(x)$ and it is transmitted on the channel $(m+r$ bit)
- At RX if the received sequence of bits is exactly divided by $G(x)$ then it is considered correct
Channel access methods

- Point-to-point (e.g. serial cable)
  - Point-Point Protocol
- Shared (bus, wireless)
  - Random access
    - CSMA/CD, CSMA/CA
  - Controlled access
    - Polling, token
  - Multiplexing
    - TDMA, FDMA, WDMA, CDMA
Point-to-point channel access

• In a point-to-point channel the arbitration is trivial since there are always two nodes
Limit of the point-to-point architecture

• In case of $N$ nodes the number of point-to-point channels is $N(N-1)$ with a quadratic cost increase

• A shared channel is needed
Access in case of shared channel

• Random access: the node which wants to transmit must wait for the channel to be free (carrier sense)

• Controlled access:
  – Polling: a master asks to each other node if it has something to transmit
  – Token: a token moves among the nodes; the node with the token can transmit for a given amount of time
Access in case of shared channel (2)

• Multiplexing: the physical channel is de-composed into logical channels used by nodes pairs as they were point-to-point channels

• De-composition methodology:
  – Radio frequency for wireless (Frequency Division Multiplexing o FDM) o light color for optical fibers (Wavelength Division Multiplexing o WDM)
  – Time interval (Time Division Multiplexing – TDM)
  – Frequency+time (Code Division Multiplexing – CDM)
    • 3G mobile and beyond
Problems in case of wireless transmission

• Interference and path loss
  – Non-negligible bit error rate

• Collision management more complex
  – Hidden node
  – Exposed Node
Interference and path loss

- More devices use the same frequency band (since it is un-licensed)
  - Other wireless nodes
  - Remote controls
  - Microwave ovens
- The signal energy decreases as a function of the distance between TX and RX
- Obstacles (e.g., walls)
- Multiple reflections of the signal cause signal distortion
Correct frame probability

• Probability to receive a correct bit

\[ (1 - P_{\text{bit}}^{\text{error}}) \]

• Probability to receive a PDU of length N

\[ P_{\text{frame}} = (1 - P_{\text{bit}}^{\text{error}})^N \]

  – E.g., N = 1518 byte = 12144 bit

• Caso Ethernet

\[ P_{\text{bit}}^{\text{error}} = 10^{-10} \Rightarrow P_{\text{ok}}^{\text{frame}} = 0.9999988 \]

• Caso WiFi

\[ P_{\text{bit}}^{\text{error}} = 10^{-4} \Rightarrow P_{\text{ok}}^{\text{frame}} = 0.2968700 \]
Limits of carrier sense:
Hidden node and exposed node

A wants to send to B but cannot hear that B is busy

B wants to send to C but mistakenly thinks the transmission will fail

(a) Hidden node
(b) Exposed node
Limit of collision detection (CSMA/CD)

• Collision Detection phase of CSMA/CD is not suitable
  – A double radio interface (to send and sense concurrently) is expensive...
  – … and useless since most of the collisions happen at the receiver

• --> Collision Avoidance
• --> Stop&Wait ack
CSMA/CA

- Carrier sense
- Collision avoidance via random back-off
- [optional] RTS/CTS
MEDIUM ACCESS CONTROL FOR WSN
MAC Challenges

• Traditionally
  – Fairness
  – Latency
  – Throughput

• For Sensor Networks
  – Power efficiency
  – Scalability
Power consumption of carrier sense

• Expected life time of many WSN applications: Months or years
• Actual lifetime
  – AA batteries: Max. 2000 mAh
  – CC2430 radio: 26.7mA in RX mode
  – 2000mAh / 26.7mA = 75 hours = 3 days

→ Keep radio asleep most of the time
→ Ideal duty cycle: 0.1% - 1%
Texas Instruments CC2430 architecture
# Power modes in TI CC2430

<table>
<thead>
<tr>
<th>Power Mode</th>
<th>Current (mA)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCU Active Mode, 16 MHz</td>
<td>4.3</td>
<td>Digital regulator on, High frequency (16 MHz) RCOSC running. No radio, crystals, or peripherals active.</td>
</tr>
<tr>
<td>MCU Active Mode, 32 MHz</td>
<td>9.5</td>
<td>MCU running at full speed (32MHz), 32MHz XOSC running. No radio or peripherals active.</td>
</tr>
<tr>
<td>MCU Active and RX Mode</td>
<td>26.7</td>
<td>MCU running at full speed (32MHz), 32MHz XOSC running, radio in RX mode, -50 dBm input power. No peripherals active.</td>
</tr>
<tr>
<td>MCU Active and TX Mode, 0dBm</td>
<td>28.1</td>
<td>MCU running at full speed (32MHz), 32MHz XOSC running, radio in TX mode, 0dBm output power. No peripherals active.</td>
</tr>
<tr>
<td>Power mode 1</td>
<td>190</td>
<td>Digital regulator on, High frequency RCOSC and crystal oscillator off. 32.768 kHz XOSC, POR and ST active. RAM retention.</td>
</tr>
<tr>
<td>Power mode 2</td>
<td>0.5</td>
<td>Digital regulator off, High frequency RCOSC and crystal oscillator off. 32.768 kHz XOSC, POR and ST active. RAM retention.</td>
</tr>
<tr>
<td>Power mode 3</td>
<td>0.3</td>
<td>No clocks. RAM retention. POR active.</td>
</tr>
</tbody>
</table>
### Power modes in TI CC2430

<table>
<thead>
<tr>
<th>Mode</th>
<th>Current (mA)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCU Active Mode, 16 MHz</td>
<td>4.3</td>
<td>Digital regulator on, High frequency (16 MHz) RCOSC running. No radio, crystals, or peripherals active.</td>
</tr>
<tr>
<td>MCU Active Mode, 32 MHz</td>
<td>9.5</td>
<td>MCU running at full speed (32MHz), 32MHz XOSC running. No radio or peripherals active.</td>
</tr>
<tr>
<td>MCU Active and RX Mode</td>
<td>26.7</td>
<td>MCU running at full speed (32MHz), 32MHz XOSC running, radio in RX mode, -50 dBm input power. No peripherals active.</td>
</tr>
<tr>
<td>MCU Active and TX Mode, 0dBm</td>
<td>28.1</td>
<td>MCU running at full speed (32MHz), 32MHz XOSC running, radio in TX mode, 0dBm output power. No peripherals active.</td>
</tr>
<tr>
<td>Power mode 1</td>
<td>190</td>
<td>Digital regulator on, High frequency RCOSC and crystal oscillator off. 32.768 kHz XOSC, POR and ST active. RAM retention.</td>
</tr>
<tr>
<td>Power mode 2</td>
<td>0.5</td>
<td>Digital regulator off, High frequency RCOSC and crystal oscillator off. 32.768 kHz XOSC, POR and ST active. RAM retention.</td>
</tr>
<tr>
<td>Power mode 3</td>
<td>0.3</td>
<td>No clocks. RAM retention. POR active.</td>
</tr>
</tbody>
</table>
Example of power-efficient MAC

- 1 s in sleep mode (power mode 2) → 0.5 μA
- 0.005 s in RX mode for carrier sense → 26.7 mA
- 0.005 s in TX mode to send packet → 28.1 mA
- Weighted current consumption
  \[ (0.0005 \times 1000 + 26.7 \times 5 + 28.1 \times 5)/(1010) \sim 0.27 \text{ mA} \]
- With AA batteries: 2000mAh / 0.27 mA ~ 7359 hours ~ 307 days
Sources of energy waste

- Collision
  - Retransmissions
- Idle listening
  - Continuously sense the channel
- Overhearing
  - Listen to packets addressed to other nodes
- Packet overhead
  - Header
  - Control packets (e.g., RTS/CTS)
Power Save Design Alternatives

• Wake-up radio
  – A sleeping node can be woken at any time by a secondary receiver (wake-up radio)

• Asymmetric polling

• Timer-Based
  – When a node enters sleep mode, it sets a timer to wakeup at a pre-determined time

• Hybrid
  – Timer-Based plus Wake-up radio
Wake-up radio

• Add second, low-power receiver to wakeup the main system on-demand
• Low-power could be achieved by:
  – Simpler hardware with a lower bit-rate and/or less decoding capability
  – Periodic listening using a radio with identical physical layer as data radio
Wake-up radio

Ultra low-power sub-system

Interrupt
Asymmetric polling

• Implemented in IEEE802.15.4
• Rules depend on the direction of the transfer
Timer-based MAC

- Scheduled contention (slotted access): Nodes periodically wake up together, contend for channel, then go back to sleep
  - S-MAC

- Channel polling (random access): Nodes independently wake up to sample channel
  - B-MAC, X-MAC

- TDMA (Time Division Multiple Access): Nodes maintain a schedule that dictates when to wake up and when they are allowed to transmit
  - DRAND

- Hybrid: SCP, Z-MAC, 802.15.4 (contention access period + contention free period)
S-MAC (Sensor MAC)

- A node sleeps most of the time
- Periodically wake up for short intervals to see if any node is transmitting a packet
- Accept latency to extend lifetime
S-MAC: SYNC interval

• Listen time consists of two parts: SYNC and RTS
• In the SYNC interval some nodes periodically send SYNC packet to synchronize clocks
• They use CSMA/CA for channel contention
S-MAC: RTS interval

- RTS/CTS is used to transmit data
- CSMA/CA followed by RTS/CTS

C. Lu, Washington Univ. Saint Louis

It lost contention
S-MAC: data transmission

- RTS/CTS contain the expected data TX time
  - Listeners not interested can sleep to save energy
- Sender does one RTS/CTS and then sends data for the rest of the frame
  - Prefer application performance to node level fairness
- ACK every data packet
  - Packet fragmentation for higher reliability

C. Lu, Washington Univ. Saint Louis
Pros and Cons of S-MAC

• More power conserving than standard CSMA/CA

• During the listening interval, everyone needs to stay awake unless someone transmits
  – Waste energy

• Time sync overhead

• RTS/CTS/ACK overhead

• Complex to be implemented
B-MAC (Berkeley MAC)

• Low Power Listening (LPL)
  – Periodic preamble sampling → Preamble > Sleep period
  – No sync between nodes

• Hidden terminal avoidance and multi-packet mechanisms not provided
Pros and Cons of B-MAC

• No need for everybody to stay awake when there is no traffic
  – Just wake up for preamble sampling and go back to sleep

• Better power conservation, latency and throughput than S-MAC

• Simpler to implement

• Low duty cycle $\rightarrow$ longer preamble
  – Little cost to receiver yet higher cost to sender
  – Longer delay
  – More contention
X-MAC: Early ACK

- Include destination address in short preambles
- Non-receiver avoids overhearing
- Receiver acknowledges preamble → Sender stops sending preamble
Thoughts on X-MAC

- Better than B-MAC in terms of latency, throughput and power consumption
- Energy consumption due to overhearing reduced
- Simple to implement
- On average the preamble size is reduced by half compared to B-MAC → Still considerable overhead
SCP-MAC

• Scheduled Channel Polling by everybody
  – Avoid long preambles in LPL (Low Power Listening) supported by B-MAC

• Wake up tone
  – Much shorter than preamble in LPL followed by data

• Assumption: the listening intervals must be synchronized
SCP-MAC (2)

(a) Low-power listening (LPL)

(b) Synchronized channel polling (SCP)

Figure 1. Sender and receiver synchronization schemes.
Time Division Multiple Access (TDMA)

- Predictable delay, throughput and duty cycle
- Little packet losses due to contention
- Scheduling and time sync are difficult
- Slots are wasted when a node has nothing to send
TDMA

Superframe

Time Sync | A Transmits | B Transmits | C Transmits | Sleep | Sleep

Slot
Z-MAC (Zebra MAC)

- Runs on top of B-MAC
- Rely on CSMA under light load → Switch to TDMA under high contention
Z-MAC (Zebra MAC)

CSMA
• Pros
  – Simple
  – Scalable
• Cons
  – Collisions due to hidden terminals
  – RTS/CTS is overhead

TDMA
• Pros
  – Naturally avoids collisions
• Cons
  – Complexity of scheduling
  – Synchronization needed
Thoughts on Z-MAC

• Good idea to combine strengths of CSMA and TDMA

• Complex

• Especially hard to implement TDMA part
  – How to deal with topology changes?
IEEE 802.15.4 superframe
IEEE 802.15.4 superframe

- Beacon frame sent periodically by the coordinator
  - It contains the superframe structure and the slot-transmitter association map
- CAP: Contention Access Period
  - CSMA/CA
  - For new nodes to join & reserve slots and for delay-insensitive flows
- CFP: Contention Free Period
  - TDMA for delay-sensitive flows
  - GTS: Guaranteed Time Slot
- Inactive period: also the coordinator can sleep
MAC protocols supported by TinyOS

- CC1100: experimental B-MAC
- CC2420: X-MAC