X-MAC and Ri-MAC

JeongGil Ko
CS450 TA Section
Oct. 9 2009
MAC Protocol

- Medium Access Control Protocols
- Many nodes share a common wireless medium and MAC protocols control which node will be occupying the wireless medium
MAC Protocol

MAC protocols determine which of TX1 and TX2 will send its packet.
MAC Protocol

Without a MAC protocol, packets from different senders can potentially collide!
MAC Protocol

- Carrier Sense Multiple Access
- Time Division Multiple Access
X-MAC

- ACM SenSys 2006
- University of Colorado at Boulder
- We will be implementing a simpler version of X-MAC
X-MAC

- Address some inefficiencies of LPL
- Transmitter initiates packet transmission with continuous short packet transmissions
- Receiver acknowledges the reception of message to finish data transfer or initiate a new packet transfer from the transmitter
Operation of X-MAC

Figure v: Diagram of X-MAC operations

The sending node $S$ tries to send a packet to $R$ by continuously transmitting packets in the air. Upon $R$'s wake-up, $R$ receives the data and replies with an acknowledgement packet to $S$. The light gray background indicates that a node's radio is on.

Figure w: Diagram of R-MAC operations with a single sender. A node that intends to send a packet stays awake until it hears a beacon from the receiver nodes. The beacon from $R$ initiates the data transmission. The node goes to sleep if it has no more pending data packets. The maximum time in the acknowledgment beacon is used to initiate new data transfers. If $R$ does not receive any incoming data packets before the maximum time expires, it goes back to sleep until the next scheduled wake-up time.

When multiple senders exist, packet collisions can happen. For example, both $S_v$ and $S_w$ want to send packets to $R$, but they randomly select close backoff times. If $R$ detects packet collisions on the channel, it broadcasts a new beacon with a larger maximum time after the previous maximum time expires. By having a larger maximum time for the next period, R-MAC intends to increase the probability of multiple senders selecting backoff times farther away.

Note that if the sender cannot deliver the packet within one duty cycle period, it stops and signals FAIL to the user applications. We will use Binary Exponential Backoff (BEB) strategy to calculate the maximum time at the receivers. Specifically, BEB computes the maximum time as $m \times i^n - v$, where $i$ starts from $v$ and increments by $v$ every time the node detects a collision. Note that $i$ is initialized to $v$ at the beginning of each wake-up cycle. Also, the receiver node stays awake for $z$ more ms to assure that the packets being propagated can make it to the receiver.

4 Skeleton Code

You can download the skeleton code from the class websites. The code is divided into four subfolders:

- The /tos/ folder adds necessary features to your original tree, such as congestion notification for R-MAC.
- The /apps/ folder will contain the application file that uses the MAC implementation.
- The other two folders, /rimac/ and /xmac/, will contain the actual implementation of the two different MAC protocols.

We simplify the MAC implementations by assuming that only one packet is processed at the MAC at any given time. If the MAC is already trying to send a packet and the application requests a new packet transmission, the MAC should return EBUSY. The MAC is also in charge.
Ri-MAC

- ACM SenSys 2007
- Rice University
- We will be implementing a simpler version of Ri-MAC
Ri-MAC

- Avoid unneeded packet transmissions
- Deal with cases where multiple transmitters compete and potentially cause packet collision
Operation of Ri-MAC

Start data transmission upon receiving R’s beacon

S: Wake up to send and wait for beacon

Node sends a beacon but goes to sleep since no incoming DATA

R: Node sends a beacon when it wakes up

Figure v: Diagram of XrMAC operations

The sending node S tries to send a packet to R by continuously transmitting packets in the air. Upon R’s wakeup, R receives the data and replies with an acknowledgement packet to S. The light gray background indicates that a node’s radio is on.

Figure w: Diagram of RirMAC operation with a single sender. A node that intends to send a packet stays awake until it hears a beacon from the receiver nodes. The beacon from R initiates the data transmission; S goes to sleep if it has no more pending data packets.

The max in the acknowledgment beacon is used to initiate new data transfers. If R does not receive any incoming data packets before the t max expires, it goes back to sleep until the next scheduled wake-up time.

When multiple senders exist, packet collisions can happen. For example, both S v and S w want to send packets to R, but they randomly select close backoff times. If R detects packet collision on the channel, it broadcasts a new beacon with a larger t max after the previous t max expires.

By having a larger t max for the next period, RirMAC intends to increase the probability of multiple senders selecting backoff times farther away. Note that if the sender cannot deliver the packet within one duty cycle period, it stops and signals FAIL to the user applications. We will use the Binary Exponential Backoff (BEB) strategy to calculate t max at the receivers. Specifically, BEB computes t max as m + i n − v, where i starts from v and increments by v every time the node detects a collision.

Note that i is initialized to v at the beginning of each wake-up cycle. Also, the receiver node stays awake for z more ms to assure that the packets being propagated can make it to the receivers.

Skeleton Code

You can download the skeleton code from the class websites. The code is divided into four subfolders:

- The /tos/ folder adds necessary features to your original tree.
- The /apps/ folder will contain the application file that uses the MAC implementation.
- The other two folders /rimac/ and /xmac/ will contain the actual implementation of the two different MAC protocols.

We simplify the MAC implementations by assuming that only one packet is processed at the MAC at any given time. If the MAC is already trying to send a packet and the application requests a new packet transmission, the MAC should return EBUSY.
Operation of Ri-MAC

• What happens in case of collision?
• Increase $t_{\text{max}}$
  • Increase the possibility of separated transmission times between multiple nodes
• Additional 5 ms wake up of the transmitter for easy implementation
Skeleton Code
THE END!

• Questions to me or mailing list :)

• Any one that did not get their homework submission receipts please raise your hands!