2.1 MAC PROTOCOLS FOR WIRELESS SENSOR NETWORKS

- The medium access control(MAC) is a sublayer of the data link layer of the open system interconnections reference model for data transmission.
- It is responsible for flow control and multiplexing for transmission medium.

Characteristics of MAC protocols:

Transmission Delay: Transmission delay is defined as the amount of time that a single message spends in the MAC protocol

Throughput: Throughput is defined by the rate at which messages are served. The throughput can be measured in messages or symbols per second.

Fairness: A MAC protocol is considered fair if it allocates a channel among the competing nodes according to some fairness criteria.

Scalability: Scalability describes the ability of the communication system to meet performance characteristics despite of the size of the network and number of competing nodes.

Robustness: Robustness is referred to as a composition of reliability, availability and dependability.

Stability: Stability describes how good the protocol handles fluctuation of traffic load over a sustainable period of time.

Goals of MAC Protocols:

- Minimize Energy Consumption
- Overhearing: Unnecessarily receive a packet destined to another node
- Idle listening : Staying active to receive even if there is no sender.

Design considerations for MAC protocols in wireless sensor networks

a) Balance of requirements

- The importance of energy efficiency for the design of MAC protocols is relatively new and many of the “classical” protocols like ALOHA and CSMA contain no provisions toward this goal.
- Other typical performance figures like fairness, throughput, or delay tend to play a minor role in sensor networks.
- Further important requirements for MAC protocols are scalability and robustness against frequent topology changes.

- It is caused by nodes powering down temporarily to replenish their batteries by energy scavenging, mobility, deployment of new nodes, or death of existing nodes.

b) Energy problems on the MAC layer

- A nodes transceiver consumes a significant share of energy.
- The transceiver has four main states: transmitting, receiving, idling, or sleeping.
- Transmitting is costly, receive costs often have the same order of magnitude as transmit costs, idling can be significantly cheaper but also about as expensive as receiving, and sleeping costs almost nothing but results in a "deaf" node.
- Some energy problems and design goals are mentioned below:
  - ✓ **Collisions**:
    - ❖ Collisions incur useless receive costs at the destination node, useless transmit costs at the source node, and the prospect to expend further energy upon packet retransmission.
    - ❖ Hence, collisions should be avoided, either by design (fixed assignment/TDMA or demand assignment protocols) or by appropriate collision avoidance/hidden-terminal procedures in CSMA protocols.
  - ✓ **Overhearing**
    - ❖ Unicast frames have one source and one destination node.
    - ❖ However, the wireless medium is a broadcast medium and all the source's neighbors that are in receive state hear a packet and drop it when it is not destined to them; these nodes overhear the packet.
  - ✓ **Protocol overhead**
    - ❖ Protocol overhead is induced by MAC-related control frames like, RTS and CTS packets or request packets in demand assignment protocols
  - ✓ **Idle listening**
    - ❖ A node being in idle state is ready to receive a packet but is not currently receiving anything.

- ❖ This readiness is costly and useless in case of low network loads; the idle state still consumes significant energy.
- ❖ Switching off the transceiver is a solution.
- ❖ A design constraint somewhat related to energy concerns is the requirement for low complexity operation.
- ❖ Sensor nodes shall be simple and cheap and cannot offer plentiful resources in terms of processing power, memory, or energy.
- ❖ Therefore, computationally expensive operations like complex scheduling algorithms should be avoided.

Important classes of MAC Protocols

a) Fixed assignment protocols

b) Demand assignment protocols

c) Random access protocols

a) Fixed assignment protocols

- In this class of protocols, available resources are divided between the nodes such that the resource assignment is long term and each node can use its resources exclusively without the risk of collisions.
- Long term means that the assignment is for durations of minutes, hours or even longer as opposed to the short term case where assignments have a scope of a data burst, corresponding to a time horizon of perhaps milliseconds.
- Typical protocols of this class are TDMA, FDMA, CDMA and SDMA
    1. The Time Division Multiple Access(TDMA) scheme subdivides the time axis into fixed length super frames and each super frame is again subdivided into a fixed number of time slots. These time slots are assigned to nodes exclusively and hence the node can transmit in this time slot periodically in every super frame.TDMA requires tight time synchronization between nodes to avoid overlapping of signals in adjacent time slots.
    2. In Frequency Division Multiple Access(FDMA), The available frequency band is subdivided into a number of subchannels and these are assigned to nodes,

which can transmit exclusively on their channel. This scheme requires frequency synchronizations to renegotiate the assignment of resources to nodes.

3. In Code Division Multiple Access(CDMA) schemes, the nodes spread their signals over a much larger bandwidth than needed, using different codes to separate their transmissions. The receiver has to know the code used by the transmitter all parallel transmissions using other codes appear as noise.
4. In Space Division Multiple Access(SDMA), The spatial separation of nodes is used to separate their transmissions. SDMA requires arrays of antennas and sophisticated signal processing techniques and cannot be considered a candidate technology for WSNs

b) Demand assignment protocols

In demand assignment protocols, the exclusive allocation of resources to nodes is made on a short-term basis, typically the duration of a data burst. This class of protocols can be broadly subdivided into centralized and distributed protocols. In central control protocols (examples are the HIPERLAN/2 protocol, DQRUMA, or the MASCARA protocol; polling schemes can also be subsumed under this class), the nodes send out requests for bandwidth allocation to a central node that either accepts or rejects the requests. In case of successful allocation, a confirmation is transmitted back to the requesting node along with a description of the allocated resource, for example, the numbers and positions of assigned time slots in a TDMA system and the duration of allocation. The node can use these resources exclusively. The submission of requests from nodes to the central station is often done contention based, that is, using a random access protocol on a dedicated (logical) signalling channel. Another option is to let the central station poll its associated nodes. In addition, the nodes often piggyback requests onto data packets transmitted in their exclusive data slots, thus avoiding transmission of separate request packets. The central node needs to be switched on all the time and is responsible for resource allocation. Resource deallocation is often done implicitly: when a node does not use its time slots any more, the central node can allocate these to other nodes. This way, nodes do not need to send extra deallocation packets. Summarizing, the central node performs a lot of activities, it must be constantly awake, and thus needs lots of energy. This class of protocols is a good choice if a sufficient number of energy-unconstrained nodes are present and the duties of the central station can be moved to these.

An example of distributed demand assignment protocols are token-passing protocols like IEEE 802.4 Token Bus. The right to initiate transmissions is tied to reception of a small special token frame. The token frame is rotated among nodes organized in a logical ring on top of a broadcast medium. Special ring management procedures are needed to include and exclude nodes from the ring or to correct failures like lost tokens. Token-passing protocols have also been considered for wireless or error-prone media, but they tend to have problems with the maintenance of the logical ring in the presence of significant channel errors. In addition, since token circulation times are variable, a node must always be able to receive the token to avoid breaking the logical ring. Hence, a nodes transceiver must be switched on most of the time. In addition, maintaining a logical ring in face of frequent topology changes is not an easy task and involves significant signalling traffic besides the token frames themselves.

c) Random access protocols

The nodes are uncoordinated, and the protocols operate in a fully distributed manner. Random access protocols often incorporate a random element, for example, by exploiting random packet arrival times, setting timers to random values, and so on. One of the first and still very important random access protocols is the ALOHA or slotted ALOHA protocol, developed at the University of Hawaii. In the pure ALOHA protocol, a node wanting to transmit a new packet transmits it immediately. There is no coordination with other nodes and the protocol thus accepts the risk of collisions at the receiver. To detect this, the receiver is required to send an immediate acknowledgment for a properly received packet. The transmitter interprets the lack of an acknowledgment frame as a sign of a collision, backs off for a random time, and starts the next trial. ALOHA provides short access and transmission delays under light loads; under heavier loads, the number of collisions increases, which in turn decreases the throughput efficiency and increases the transmission delays. In slotted ALOHA, the time is subdivided into time slots and a node is allowed to start a packet transmission only at the beginning of a slot. A slot is large enough to accommodate a maximum-length packet. Accordingly, only contenders starting their packet transmission in the same slot can destroy a node's packet. If any node wants to start later, it has to wait for the beginning of the next time slot and has thus no chance to destroy the node's packet. In short, the synchronization reduces the probability of collisions and slotted ALOHA has a higher throughput than pure ALOHA.

In the class of CSMA protocols, a transmitting node tries to be respectful to ongoing transmissions. First, the node is required to listen to the medium; this is called carrier sensing. If the medium is

found to be idle, the node starts transmission. If the medium is found busy, the node defers its transmission for an amount of time determined by one of several possible algorithms. For example, in nonpersistent CSMA, the node draws a random waiting time, after which the medium is sensed again. Before this time, the node does not care about the state of the medium. In different persistent CSMA variants, after sensing that the medium is busy, the node awaits the end of the ongoing transmission and then behaves according to a backoff algorithm. In many of these backoff algorithms, the time after the end of the previous frame is subdivided into time slots. In p-persistent CSMA, a node starts transmission in a time slot with some probability p and with probability 1 – p it waits for another slot.3 If some other node starts to transmit in the meantime, the node defers and repeats the whole procedure after the end of the new frame. A small value of p makes collisions unlikely, but at the cost of high access delays. The converse is true for a large value of p.

In the backoff algorithm executed by the IEEE 802.11 Distributed Coordination Function (DCF), a node transmitting a new frame picks a random value from the current contention window and starts a timer with this value. The timer is decremented after each slot. If another node starts in the meantime, the timer is suspended and resumed after the next frame ends and contention continues. If the timer decrements to zero, the node transmits its frame. When a transmission error occurs (indicated, for example, by a missing acknowledgment frame), the size of the contention window is increased according to a modified binary exponential backoff procedure.4 While CSMA protocols are still susceptible to collisions, they have a higher throughput efficiency than ALOHA protocols, since ongoing packets are not destroyed when potential competitors hear them on the medium.

As explained above, carrier-sense protocols are susceptible to the hidden-terminal problem since interference at the receiver cannot be detected by the transmitter. This problem may cause packet collisions. The energy spent on collided packets is wasted and the packets have to be retransmitted. Several approaches have appeared to solve or at least to reduce the hidden-terminal problem; we present two important ones: the busy-tone solution and the RTS/CTS handshake.

In the original busy-tone solution, two different frequency channels are used, one for the data packets and the other one as a control channel. As soon as a node starts to receive a packet destined to it, it emits an unmodulated wave on the control channel and ends this when packet reception is finished. A node that wishes to transmit a packet first senses the control channel for the presence of a busy tone. If it hears something, the node backs off according to some algorithm, for example similar to nonpersistent CSMA. If it hears nothing, the node starts packet transmission on the data

channel. This protocol solves both the hidden- and exposed terminal problem, given that the busy-tone signal can be heard over the same distance as the data signal. If the busy tone is too weak, a node within radio range of the receiver might start data transmission and destroy the receiver's signal. If the busy tone is too strong, more nodes than necessary suppress their transmissions. The control channel does not need much bandwidth but a narrow bandwidth channel requires good frequency synchronization. A solution with two busy tones, one sent by the receiver and the other by the transmitter node. Another variant of the busy-tone approach is used by PAMAS.

The RTS/CTS handshake as used in IEEE 802.11 is based on the MACAW protocol and is illustrated in Figure 2.2. It uses only a single channel and two special control packets. Suppose that node B wants to transmit a data packet to node C. After B has obtained channel access (for example after sensing the channel as idle), it sends a Request. To Send (RTS) packet to C, which includes a duration field indicating the remaining length of the overall transaction (i.e., until the point where B would receive the acknowledgment for its data packet). If C has properly received the RTS packet, it sends a Clear. To Send (CTS) packet, which again contains a duration field. When B receives the CTS packet, it starts transmission of the data packet and finally C answers with an acknowledgment packet. The acknowledgment is used to tell B about the success of the transmission; lack of acknowledgment is interpreted as collision (the older MACA protocol lacks the acknowledgment). Any other station A or D hearing either the RTS, CTS, data or acknowledgment packet sets an internal timer called Network Allocation Vector (NAV) to the remaining duration indicated in the respective frame and avoids sending any packet as long as this timer is not expired. Specifically, nodes A and D send no CTS answer packets even when they have received a RTS packet correctly. This way, the ongoing transmission is not distorted.

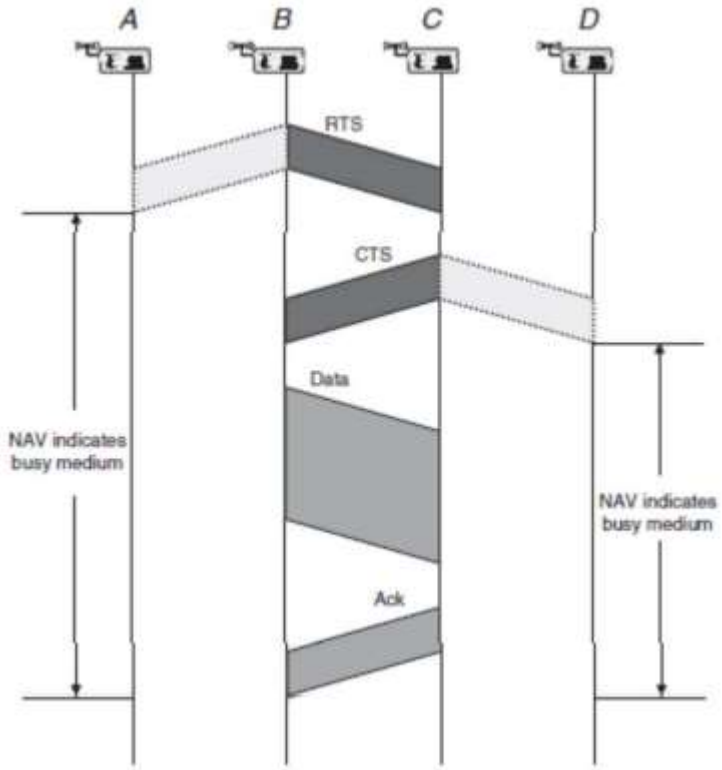

Fig 2.2 RTS/CTS handshake in IEEE 802.11

Does this scheme eliminate collisions completely? No, there still exist some collision scenarios. First, in the scenario described above, nodes A and C can issue RTS packets to B simultaneously. However, in this case, only the RTS packets are lost and no long data frame has been transmitted. Two further problems are illustrated in Figure 3: In the left part of the figure, nodes A and B run the RTS-CTS-Data-Ack sequence, and B's CTS packet also reaches node C. However, at almost the same time, node D sends an RTS packet to C, which collides at node C with B's CTS packet. This way, C has no chance to decode the duration field of the CTS packet and to set its NAV variable accordingly. After its failed RTS packet, D sends the RTS packet again to C and C answers with a CTS packet. Node C is doing so because it cannot hear A's ongoing transmission and has no proper NAV entry. C's CTS packet and A's data packet collide at B. In the figure's right part, the problem is created by C starting its RTS packet to D immediately before it can sense B's CTS packet, which C consequently cannot decode properly. One solution approach is to ensure that CTS packets are longer than RTS packets. For an explanation, consider the right part of Figure 3. Here, even if B's CTS arrives at C immediately after C starts its RTS, it lasts long enough that C has a chance to turn its transceiver into receive mode and to sense B's signal. An additional protocol rule

states that in such a case node C has to defer any further transmission for a sufficiently long time to accommodate one maximum-length data packet. Hence, the data packet between A and B can be transmitted without distortion.

A further problem of the RTS/CTS handshake is its significant overhead of two control packets per data packet, not counting the acknowledgment packet. If the data packet is small, this overhead might not pay off and it may be simpler to use some plain CSMA variant.

For long packets, the overhead of the RTS/CTS handshake can be neglected, but long packets are more likely to be hit by channel errors and must be retransmitted entirely, wasting precious energy (channel errors often hit only a few bits).

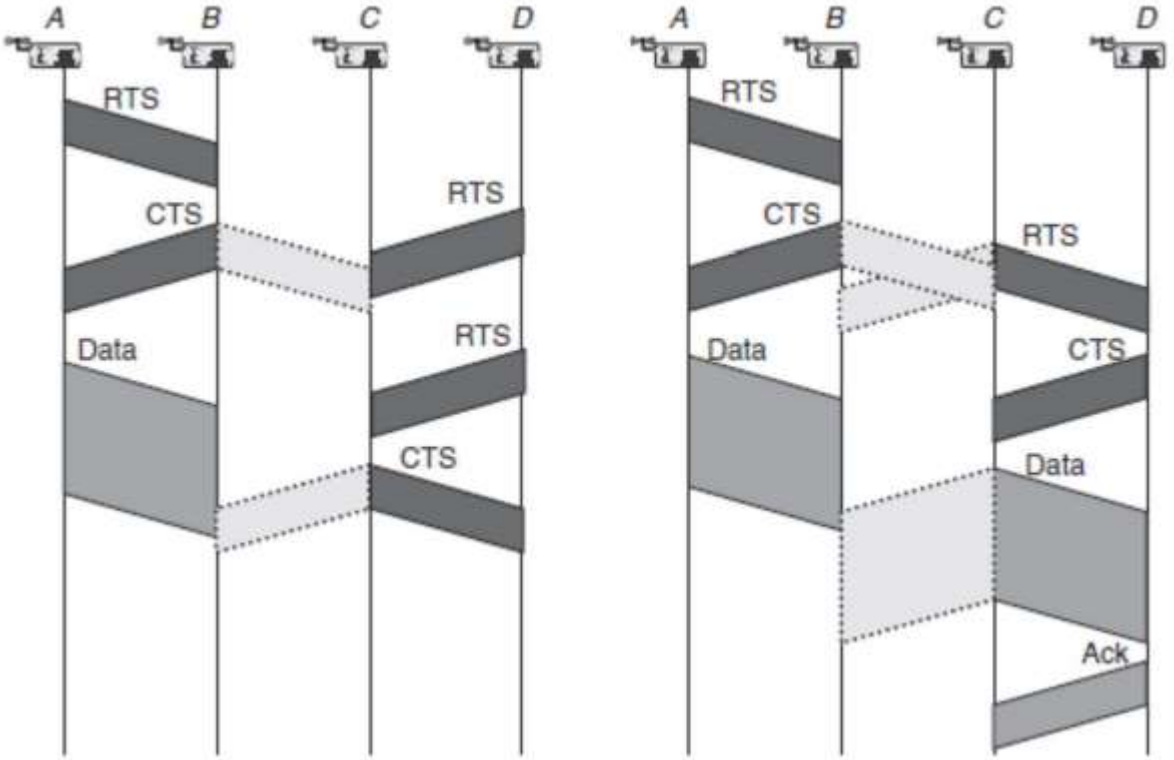

Fig: Two problems in RTS/CTS handshake

ROHINI COLLEGE OF ENGINEERING AND TECHNOLOGY
OBSERVE OPTIMIZE OUTSPREAD