2.2 Low duty cycle protocols and wakeup concepts

- Low duty cycle protocols try to avoid spending time in the idle state and to reduce the communication activities of a sensor node to a minimum.
- In an ideal case, the sleep state is left only when a node is about to transmit or receive packets.
- A concept for achieving this is called wakeup radio.
- In several protocols, a periodic wakeup scheme is used. Such schemes exist in different flavors. One is the cycled receiver approach is illustrated in below Figure

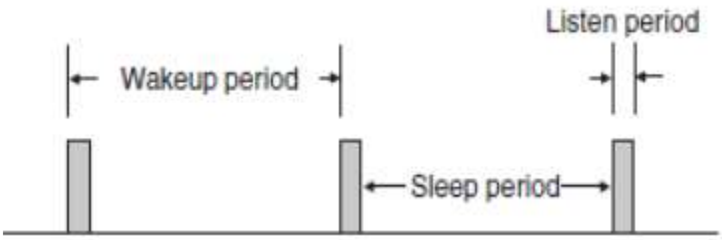

Fig 2.2.1 Periodic Wakeup Scheme

- In this approach, nodes spend most of their time in the sleep mode and wake up periodically to receive packets from other nodes.
- Specifically, a node A listens onto the channel during its listen period and goes back into sleep mode when no other node takes the opportunity to direct a packet to A.
- A potential transmitter B must acquire knowledge about A's listen periods to send its packet at the right time – this task corresponds to a rendezvous.
- This rendezvous can be accomplished by letting node A transmit a short beacon at the beginning of its listen period to indicate its willingness to receive packets.
- Another method is to let node B send frequent request packets until one of them hits A's listen period and is really answered by A.
- However, in either case, node A only receives packets during its listen period.
- If node A itself wants to transmit packets, it must acquire the target's listen period.
- A whole cycle consisting of sleep period and listen period is also called a wakeup period.
- The ratio of the listen period length to the wakeup period length is also called the node's duty cycle.
- By choosing a small duty cycle, the transceiver is in sleep mode most of the time, avoiding idle listening and conserving energy.

- By choosing a small duty cycle, the traffic directed from neighboring nodes to a given node concentrates on a small time window (the listen period) and in heavy load situations significant competition can occur.
- Choosing a long sleep period induces significant per-hop latency. In the multihop case, the per-hop latencies add up and create significant end-to-end latencies.
- Sleep phases should not be too short lest the start-up costs outweigh the benefits.
- In other protocols like S-MAC, there is also a periodic wakeup but nodes can both transmit and receive during their wakeup phases.
- When nodes have their wakeup phases at the same time, there is no necessity for a node wanting to transmit a packet to be awake outside these phases to rendezvous its receiver.

S-MAC

- The S-MAC (Sensor-MAC) protocol provides mechanisms to circumvent idle listening, collisions, and overhearing.
- S-MAC adopts a periodic wakeup scheme, that is, each node alternates between a fixedlength listen period and a fixed-length sleep period according to its schedule.
- The listen period of S-MAC can be used to receive and transmit packets.
- S-MAC attempts to coordinate the schedules of neighboring nodes such that their listen periods start at the same time.

Phases in listen period:

- A node x's listen period is subdivided into three different phases:

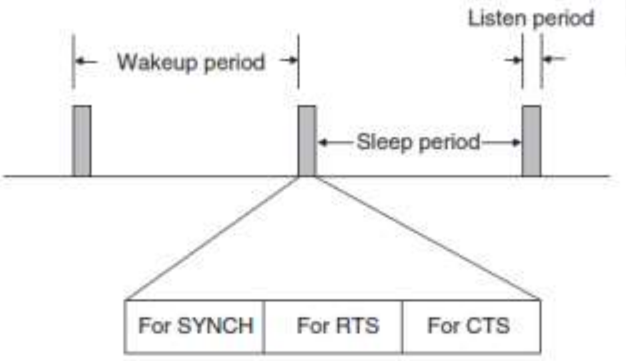

Fig 2.2.1 S-MAC Principle

First phase

- In the first phase (SYNCH phase), node x accepts SYNCH packets from its neighbors.

- In these packets, the neighbors describe their own schedule and x stores their schedule in a table (the schedule table).
- Node x's SYNCH phase is subdivided into time slots and x's neighbors contend according to a CSMA scheme with additional backoff.
- Each neighbor y wishing to transmit a SYNCH packet picks one of the time slots randomly and starts to transmit if no signal was received in any of the previous slots.
- In the other case, y goes back into sleep mode and waits for x's next wakeup. ϖ In the other direction, since x knows a neighbor y's schedule, x can wake at appropriate times and send its own SYNCH packet to y (in broadcast mode).
- It is not required that x broadcasts its schedule in every of y's wakeup periods.
- However, for reasons of time synchronization and to allow new nodes to learn their local network topology, x should send SYNCH packets periodically. The according period is called synchronization period.

Second phase

- In the second phase (RTS phase), x listens for RTS packets from neighboring nodes.
- In S-MAC, the RTS/CTS handshake is used to reduce collisions of data packets due to hidden-terminal situations.
- Again, interested neighbors contend in this phase according to a CSMA scheme with additional backoff.

Third Phase

- In the third phase (CTS phase), node x transmits a CTS packet if an RTS packet was received in the previous phase. After this, the packet exchange continues, extending into x's nominal sleep time. Working of S-MAC Protocol
- When competing for the medium, the nodes use the RTS/CTS handshake, including the virtual carrier-sense mechanism.
- When transmitting in a broadcast mode (for example SYNCH packets), the RTS and CTS packets are dropped and the nodes use CSMA with backoff.

Working of S-MAC Protocol

- When competing for the medium, the nodes use the RTS/CTS handshake, including the virtual carrier-sense mechanism.

- When transmitting in a broadcast mode (for example SYNCH packets), the RTS and CTS packets are dropped and the nodes use CSMA with backoff.
- If we can arrange that the schedules of node x and its neighbors are synchronized, node x and all its neighbors wake up at the same time and x can reach all of them with a single SYNCH packet.
- The S-MAC protocol allows neighboring nodes to agree on the same schedule and to create virtual clusters.
- The clustering structure refers solely to the exchange of schedules; the transfer of data packets is not influenced by virtual clustering.
- The S-MAC protocol proceeds as follows to form the virtual clusters:
  - ✓ A node x, newly switched on, listens for a time of at least the synchronization period.
  - ✓ If x receives any SYNCH packet from a neighbor, it adopts the announced schedule and broadcasts it in one of the neighbors' next listen periods.
  - ✓ In the other case, node x picks a schedule and broadcasts it.
  - ✓ If x receives another node's schedule during the broadcast packet's contention period, it drops its own schedule and follows the other one.
  - ✓ It might also happen that a node x receives a different schedule after it already has chosen one, for example, because bit errors destroyed previous SYNCH packets.
  - ✓ If node x already knows about the existence of neighbors who adopted its own schedule, it keeps its schedule and in the future has to transmit its SYNCH and data packets according to both schedules.
  - ✓ On the other hand, if x has no neighbor sharing its schedule, it drops its own and adopts the other one.
  - ✓ Since there is always a chance to receive SYNCH packets in error, node x periodically listens for a whole synchronization period

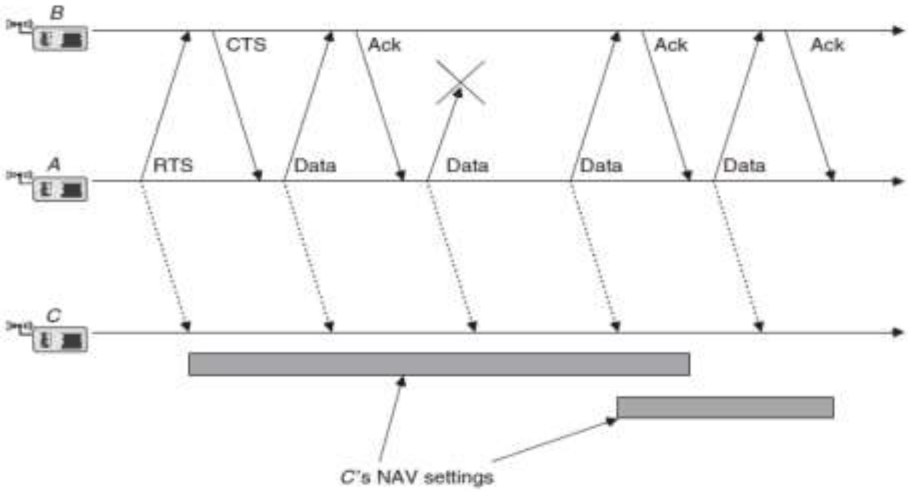

Fig:2.2.1 S-MAC Fragmentation and NAV Settings

S-MAC includes a fragmentation scheme

- ✓ A series of fragments is transmitted with only one RTS/CTS exchange between the transmitting node A and receiving node B.
- ✓ After each fragment, B has to answer with an acknowledgment packet.
- ✓ All the packets (data, ack, RTS, CTS) have a duration field and a neighboring node C is required to set its NAV field accordingly.
- ✓ In S-MAC, the duration field of all packets carries the remaining length of the whole transaction, including all fragments and their acknowledgments. Therefore, the whole message shall be passed at once.
- ✓ If one fragment needs to be retransmitted, the remaining duration is incremented by the length of a data plus ack packet, and the medium is reserved for this prolonged time.
- ✓ However, there is the problem of how a nonparticipating node shall learn about the elongation of the transaction when he has only heard the initial RTS or CTS packets.

Drawbacks:

- It is hard to adapt the length of the wakeup period to changing load situations, since this length is essentially fixed, as is the length of the listen period.

Wakeup radio concepts

- If a node were always in the receiving state when a packet is transmitted to it, in the transmitting state when it transmits a packet, and in the sleep state at all other times; the idle state should be avoided

- The wakeup radio concept strives to achieve this goal by a simple, "powerless" receiver that can trigger a main receiver if necessary.
- One proposed wakeup MAC protocol assumes the presence of several parallel data channels, separated either in frequency (FDMA) or by choosing different codes in a CDMA schemes.
- A node wishing to transmit a data packet randomly picks one of the channels and performs a carrier sensing operation.
- If the channel is busy, the node makes another random channel choice and repeats the carrier-sensing operation.
- After a certain number of unsuccessful trials, the node backs off for a random time and starts again.
- If the channel is idle, the node sends a wakeup signal to the intended receiver, indicating both the receiver identification and the channel to use.
- The receiver wakes up its data transceiver, tunes to the indicated channel, and the data packet transmission can proceed. Afterward, the receiver can switch its data transceiver back into sleep mode.
- It has the significant advantage that only the low-power wakeup transceiver has to be switched on all the time while the much more energy consuming data transceiver is nonsleeping if and only if the node is involved in data transmissions.
- Furthermore, this scheme is naturally traffic adaptive, that is, the MAC becomes more and more active as the traffic load increases.
- Periodic wakeup schemes do not have this property. However, there are also some drawbacks.
- First, there is no real hardware yet for such an ultralow power wakeup transceiver.
- Second, the range of the wakeup radio and the data radio should be the same.
- If the range of the wakeup radio is smaller than the range of the data radio, possibly not all neighbor nodes can be woken up.
- On the other hand, if the range of the wakeup radio is significantly larger, there can be a problem with local addressing schemes.
- These schemes do not use globally or network wide-unique addresses but only locally unique addresses, such that no node has two or more one-hop neighbors with the same address.
- Since the packets exchanged in the neighbor discovery phase have to use the data channel, the two hop neighborhood as seen on the data channel might be different from the two-hop neighborhood on the wakeup channel.

- Third, this scheme critically relies on the wakeup channel’s ability to transport useful information like node addresses and channel identifications;
- This might not always be feasible for transceiver complexity reasons and additionally requires methods to handle collisions or transmission errors on the wakeup channel.
- If the wakeup channel does not support this feature, the transmitter wakes up all its neighbors when it emits a wakeup signal, creating an overhearing situation for most of them.
- If the transmitting node is about to transmit a long data packet, it might be worthwhile to prepend the data packet with a short filter packet announcing the receiving node’s address.
- All the other nodes can go back to sleep mode after receiving the filter packet. Instead of using an extra packet, all nodes can read the bits of the data packet until the destination address appeared.
- If the packet’s address is not identical to its own address, the node can go back into sleep mode.