Contention based Protocols.

In contention based a transmit signal to a receiver node is taken by any of its neighboos. If one neighboor tries a luck, the packet goes through the channel. If two or more neighbor tries their luck they have to compete with each other, then collision might occur.

→ compared to the contention-based protocols with Periodic wakeup Scheme (S-MAC), these protocols have no idle listening avoidance & make no restrictions as to when a node can receive a packet.

Steps a node passes through in case of a transmission as finite State automation.

→ After the node gets a new packet for transmission from its upper layers, it starts with random delay & initializes its trial counter num_retires with zero. Random delay used for desynchronizing the nodes that are initially synchronized by external event.

→ During random delay node's transceiver put into sleep mode. During the following listen Period node do carrier sensing. If medium is busy & no. of trials < the maximum trial number, node goes into backoff mode. In backoff mode node waits for a random time. After backoff mode the node listens again. If medium is busy & node has exhausted its maximum number of trials, the packet is dropped.

→ If medium is idle node transmits RTS packets & enters the "Await CTS" State. In case no CTS packet arrives or CTS Packet for another transaction received, the node either enters backoff mode or drops the Packet, depending on num_retires value. If CTS Packet arrives the node sends it data Packet & wait for the Acknowledgement

PAMAS: Power Aware Multiaccess with Signaling) is originally designed for adhoc networks. It Provides a detailed Overhearing avoidance mechanism while it doesn't considers the idle listening Problem.

→ PAMAS uses two channels: a data channel & a control channel. All signaling packets RTS, CTS, busy tones are transmitted on control channel, while the data channel is reserved for data packets.

→ considering to idle node x a packet form neighboring node y comes. Then x sends RTS packet on control channel without doing carrier sensing. This packet carries both x's & y's MAC address. If y receives this packet it answers with CTS; if it doesn't have any ongoing transmission. With obtained CTS x transmit data packet to y through data channel. If x fails to gets CTS packet with some time window, it enters the backoff mode, with binary exponential backoff scheme used where backoff time is uniformly Chosen from a time interval that is doubled after each failure to receive a CTS.

and y after receiving Provide busy-tone packet on control channel.

→ When nodes receiving x's RTS Packet on control channel. There is intended receiver y & other nodes let z is one of them. If z is currently receiving a packet, it reacts by sending a busy tone tone Packet which overlap with y's CTS. an node x & destroys the CTS. Therefore x, can't start transmission & z's packet reception is not disturbed. As busyton packet is longer than CTS, we can ensure that CTS is destroyed.

→ considering intended receiver y. If y has ongoing transmission. it supresses CTS, causing x to back-off. Node y obtain this knowledge either by sensing data channeler by checking any noise on control channel. The noise can be RTS or CTS of another node colliding at y.

→ In other way y answers with a CTS & starts to send out a busy-tone packet as soon as x's transmission has occured.

→ A node that receives RTS packet while in backoff state starts it's packet reception procedure. i.e it checks conditions for sending CTS.

→ When a node put its transceivers (Control & data) into sleep mode. The node knows it can't transmit & receive. However, decision to go for sleep is a question, i.e how long it will sleep. This decision is easy if node x knows the length of transmission by overhearing the RTS or CTS packets or the header of data packet in data channel. However, this length is often unknown to x, because these packets may be corrupted when x is sleeping. Hence, additional procedures are needed to resolve this.

→ For example that x wakes up and find the data channel busy. There are 2 cases to distinguish either x has no own packet to send or x wants to transmit. In first case x goes to sleep mode & wake up exactly when ongoing transmission ends to be able to receive an immediately following packet.

→ Waking up early has advantage of avoiding unwanted delays. However Since x may not have overhead the RTS, CTS, or data packet header belonging to ongoing transmission. It runs a Probing Protocol on control channel to enquire length of ongoing packet. This Probing Protocol ~~on control~~ works similarly to binary search algorithm. Let l be maximal packet length in seconds. First x sends a t_Probe($l/2, l$) packet and any transmitter node who finds in time interval $[l/2, l)$ answers with a t_probe_response(t) packet. It knows exactly when this single ongoing transmission ends & when to wake up next time. If x receives only noise in responses several t_Probe_response(t) may collide at x & x starts to search in interval $[3l/4, l]$, hoping for a single answer only. If

no answer arrives at all upon t-Probe(3l/4, l), x must checks the interval (l/2, 3l/4) & so on.

→ In othercase, x wakes up during an ongoing transmission and wants to transmit a packet. Therefore, x has not only to take care of ongoing transmission but also ongoing reception in its vicinity. To find time for next wakeup, x runs the described Probing Protocol for the set of transmitters, giving a time t when the longest ongoing transmission ends. In addition x runs a similar Probing Protocol for the set of receivers in neighborhood, indicating the time r when the longest ongoing reception ends. Finally, x schedules it wakeup for time max[r,t]. If t<r, waking up at t might give another node y a chance to transmit a packet to x without any additional delay. On the other hand, if r<t, there is some chance that x can start its own transmission.

Schedule-based Protocols:

Schedule-based Protocols that do not explicitely address idle listening avoidance but do implicitely using TDMA, through assignment of transmission and reception to nodes & let them sleep all other times. Also through schedule-based Protocols no special mechanism are needed to avoid collision.

→ The disadvantages of these schemes is to maintenance of schedules that involves signalling traffic, overlapping of time slots to be avoided, Hence maintainance of time synchronization involves extra signalling traffic. In TDMA it is difficult for a node to give unused time slots to its neighbors.

LEACH: The Low-energy Adaptive clustering Hierarchy assumes a dense sensor network of homogeneous, energy-constrained nodes, which report their data to sink node.

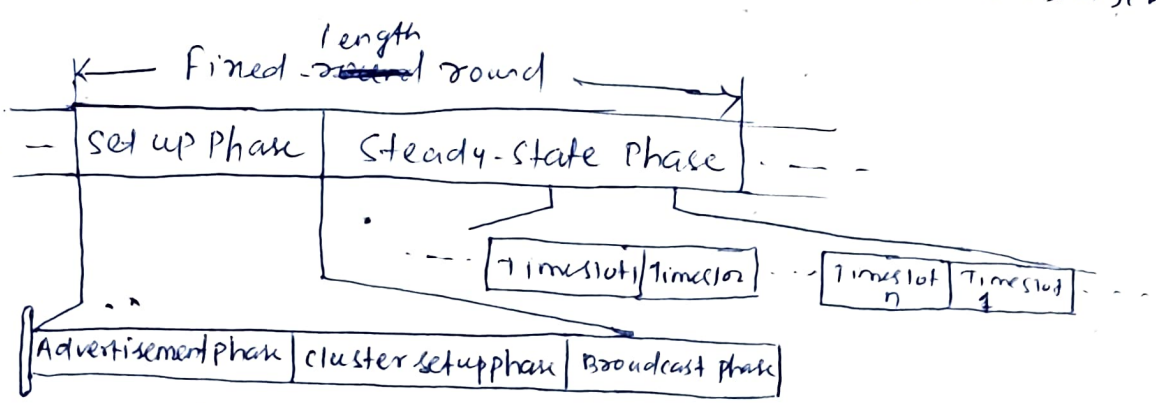
→ In LAECH a TDMA based MAC Protocol is integrated with clustering and simple routing Protocol.

-) The network partioning into cluster is time variable and the Protocol assume global time synchronization. Once the clusters have formed each cluster ~~choose~~ head picks a random CDMA code for its cluster, and it's member nodes use the same. This avoids situation where border node belonging to clusterhead A distorts transmissions directed to cluster B.

-) If the no. of clusterheads is less then there is more distance between the member nodes and cluster heads, hence more energy is required to reach the clusterhead. Similarly, if there are more cluster heads then there is more energy expensive transmissions from clusterheads to sink and less aggregation.

→ This protocol is organized in rounds and each round is divided into a setup phase and steady phase. Setup phase starts with self election of nodes and cluster heads. In following advertisement phase, the cluster heads inform their neighborhood with an advertisement packet. The Clusterheads (contend the medium) using CSMA) with no provision of hidden terminal problem. In the following cluster setup phase the members inform their clusterhead again using CSMA. After the cluster setup phase, the clusterhead knows the number of members and their identifiers. It constructs a TDMA schedule, picks a CDMA code randomly and broadcasts the information in the broadcast schedule subphase.

⇒ Because of collisions of advertisment or join packets, the protocol cannot guarantee that each non cluster head node belongs to a cluster. The clusterhead is switched on during the entire round and the member nodes have to be switched on during the setup phase and occasionally in steady-state phase according to their position in cluster's TDMA schedule.



(* LEACH would not be able to cover large geographical areas of square miles and more, because a clusterhead two miles away from sink likely does not)

have energy to reach the link at all.

SMACS. (1st phase Link (bidirectional) is set up, then node wakeup periodically for broadcast)

The Self-Organizing Medium Access Control for Sensor Networks (SMACS) Protocol is a Part of WSN Protocol Suite that address MAC, neighbor discovery, attachment of mobile nodes, a multihop routing protocol, and a local routing protocol for cooperative Signal Processing Purposes.

-> SMACS combines (neighborhood discovery and assignment of TDMA) Schedules to nodes.

Assumptions in SMAC:
- CDMA codes available
- Stationary Sensor N/wk, hence assignment valid for long time.
- Each node has subframes, all nodes have same superframe requires time synchronization

(i) The (available spectrum is subdivided into many channels) and each node can tune its transceiver to an arbitrary one, hence it is assumed that many (CDMA codes are available.)

(ii) (Most of the nodes in sensor network are stationary) and such (assignment is valid fairly long times.)

(iii) Each node devicles its time locally into fixed-length (subframes) (of duration T_{frame} seconds) which don't have necessarily same phase as neighbors Superframe. However, (all nodes have the same Superframe Length and this requires time synchronization.)

-) the (goal of SMACS is to (detect neighboring nodes and setup exclusive links or channels to these)) The links are directional and a link occupies TDMA slot. ~~A link can be bidirectional~~ for bidirectional operation two links required. Hence, a node has a (receive slot and a transmit slot) to another node. The (link assignment is made in such a way that there is no collision at the receiver) To acheive this SMACS ensures that for single node time slots for different links do not overlap and for each link random CDMA codes are allocated to the link. Therefore it is not required that node and its neighbors transmit at different times (same transmit time)

-> After link setup, the nodes wakeup periodically on the respective receive time slots with the receives tuned to the corresponding frequency or with correct CDMA code at hand.

Traffic-adaptive medium access protocol (TRAMA)

Access single channel without collisions

Schedules are established in distributed way.

TRAMA Protocols creates schedules that allows nodes to access a single channel in a collision-free manner. These schedules are constructed in a distributed manner and on demand basis.

→ This protocol assumes that all (nodes are time synchronized) and (devides time into (random access periods) and (scheduled access periods)). A random access followed by scheduled access is Period is called a cycle.

→ Nodes broadcast their neighborhood information by capturing respective packets from their neighbors. Nodes also broadcast their schedule information, i.e. they periodically provide the neighbors the updated receivers list for the packets that are in nodes queue. Based on this the nodes execute a (distributed scheduling algorithm) for each time slot of the schedule access period the transmitting and receiving nodes and the nodes that can go to sleep mode.

→ This protocol has 3 different components: the neighborhood protocol, schedule exchange protocol and adaptive election algorithm.

Neighborhood Protocol: It is (executed solely in random access phase), which is subdivided into small time slots. A (node randomly) picks number of time slots and (transmits small (control packets)) in these (without carrier sensing). (These packets ~~contain~~ indicate (node identification) and contain (incremental neighborhood information), i.e only those neighbor identification are included that belong to new neighbor or neighbors that were missing in last cycle. (When a node doesn't transmit, it listens to pick up its neighbors control packets.) The (length of random access phase) is choosen that a node receives neighbors packets with high probability.

Schedule exchange Protocol

Here a node transmits its current transmission schedule (indicating in which time slot it transmits to which neighbor) and also picks up its neighbors schedules. This information is used to allocate slots to transmitters and receivers. The question is how a node know which slots it can use. All nodes possess a global hash function h, a node whose identification is x and time slot is t, the priority value P is $P = h(x \oplus t)$, where $x \oplus t$ is the concatenation of node identification with time t. To compute schedule a node use certain time slots called schedule interval (let 100 slots) and for each slot it computes own priority and priority of two hop neighbors. The slots for which x has highest priority value can be used by x to transmit packets. These slots are called as winning slots. Let these are 17, 34, 90, 94. Looking a Packet's queue x can use all slots or leave some slots for other nodes. Node x allows each winning slot with receiving node or receivers and sends this assignment as schedule packet. The last winning slot (94) is always used for broadcasting x's next schedule, i.e whole schedule computation has to be repeated immediately before slot 94. (Using last winning slot, the schedule can be transmitted without risk of collision.) The neighbors of x should wake up at slot 94 to receive x's next schedule and to determine when they have to leave sleep mode to receive packet from x. So from x should also wakeup when its neighbors have announced to transmit their next schedule.

Adaptive election algorithm: The question is when x prepare for receptions and when can x go into sleep mode during the scheduled access phase. There are two cases:

(i) A one hop neighbor y of x has highest priority in x's

IEEE 802.15.4 MAC Protocol

(Low to medium bitrates, moderate avg. delay
In MAC layer 2 types of nodes (coordinator & device) (54)

This standard covers the physical layer & MAC layer of a low-rate WPAN. IEEE 802.15.4 is confused with ZigBee. ZigBee uses services offered by IEEE 802.15.4 and adds network construction (star network, peer-to-peer / mesh networks, cluster-tree networks), security, application services.

→ Targeted applications for IEEE 802.15.4 are in areas of WSNs, home automation, home networking, connecting devices to PC, home security etc. Most applications require only low to medium bitrates, moderate average delays. Physical layer offers bitrate of 20 kbps (single channel freq. range 868-868.6 MHz). 40kbps (ten channels in the range bet^n 905 and 928 MHz) & 250kbps (16 channels in the 2.4GHz ISM band bet^n 2.4 & 2.485 GHz. There are 27 channels available, but the MAC protocol uses only one of these channels at a time. It is not a multichannel protocol.

The MAC protocol combines both schedule-based as well as contention based schemes. The protocol is asymmetric in that different types of nodes.

Network architecture & types/roles of nodes. (coordinators and devices)
FFD
(Device) → RFD
This standard destinguishes on the MAC layer two types of nodes.

→ A full function device (FFD) can operate in different roles, it can be a PAN coordinator (PAN = personal area network), a simple coordinator or a device

→ A Reduced function device (RFD) can operate only as a device.

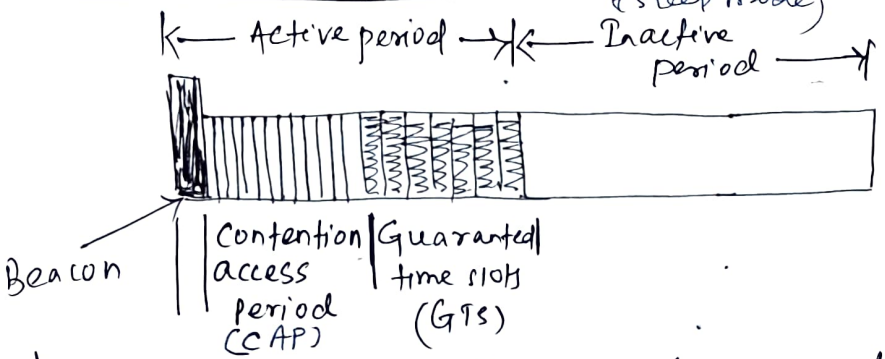
A coordinator handles among others the following task.

1. It manages a list of devices. Devices are required to explicitely associate & disassociate with a coordinator using certain signalling packets.
2. It allocates (short addresses) to its devices. All IEEE 802.15.4 nodes have a 64 bit device address. When a device associate with a coordinator, it request assignment of a 16 bit short address to be used subsequently in all communications bet^n device & coordinator. The assigned address is indicated in the (association response Packet) issued by coordinator.

→ In the beaconed mode of IEEE 802.15.4, it (transmits regularly frame beacon packets) announcing the (PAN number), a list of (outstanding frames), and other parameters. The coordinator can accept & process requests to reserve fixed time slots to nodes & the allocation are indicated in the beacon.

→ It exchanges data packets with devices & other peer coordinators.

Superframe structure of IEEE 802.15.4



* In beacon mode of transmission a beacon frame packet transmitted by coordinator to organize channel access & data transmission with the help of superframe.

→ The coordinator of star n/w operating in beacon mode organizes channel access & data transmission with help of superframe. All superframes have same length. The coordinator starts each superframe by sending a frame beacon packet. The frame beacon includes a superframe specification describing length of various components of superframe.

→ Superframe sub divided into an active & inactive period. During inactive period, all nodes including coordinator can switch of their transceivers & go to sleep state. The nodes have to wakeup immediately before the inactive period ends to receive the next beacon. The inactive period may be void.

→ The active period is subdivided into 16 time slots. First time slot is occupied by the beacon frame & the remaining time slots are partitioned into a contention access period (CAP) followed by a (maximum seven) Guaranteed Time slots (GTS).

→ The coordinator is active during entire active period. The associated devices are active in GTS Phase only on time slots allotted to them. In all other GTS slots they can enter sleep mode.

In CAP a device can shutdown its transceiver if it has neither any own data to transmit nor any data to fetch from the coordinator.

GTS Management (GTS descriptor is used)

- GTS slot is given to a device when it has requested during CAP.
- Flag in GTS slot indicate Tx or Rx slot.

Coordinator allocates GTS to devices only when the latter send appropriate request packet during the CAP. A flag in the request indicates whether the requested time slot is a transmit slot or a receive slot. In transmit slot the device transmits packets to the coordinator and in a receive slot the data flows in reverse direction.

→ Coordinator answers the request packet in 2 steps. An immediate acknowledgement packet that confirms the coordinator has received the request packet but it contains no information about success & failure of the request. After receiving the acknowledgement packet the device track the coordinator's beacons for some specified time (called a GTSDescPersistenceTime). When coordinator has sufficient resources to allocate GTS to the node, it inserts an appropriate GTS descriptor into one of next beacon frames. This GTS descriptor provides a (short address) of requesting node & the no. and position of time slot with in GTS phase of superframe. A device can use the allocated slots each time they are announced by the coordinator in the GTS descriptor. If coordinator has insufficient resources it generates GTS descriptor for (invalid) time slot zero, indicating the available resources in the descriptors length field. If device receives no GTS descriptor with in a GTSDescPersistenceTime time after sending the request, it concludes that allocation request has failed.

Data transfer Procedure:

Assuming that the device to transmit a data to the Coordinator. If the device has allocated transmit GTS, it wakes up just before the time slot starts and send packet immediately without any carrier sensing or collision mechanism. This can be only done if full transaction consisting of data packet and immediate acknowledgement sent by coordinator

as well as interframe space (IFSs) fit into the allocated time slots. If this is not the case when device doesn't have allocated slots it sends data packet during the CAP using slotted CSMA Protocol. The coordinator sends immediate acknowledgement for data packet.

→ Another case of data transfer from coordinator to a device. If device has allocated receive GTs and when packet/acknowledgement /IFS cycle fits into these, the coordinator simply transmits packet in allocated time slot without further coordination.

Slotted CSMA-CA Protocol :

When nodes have to send data or management / control packets during the CAP, they use a slotted CSMA Protocol. The protocol has no provisions against hidden-terminal situations, i.e there is no RTS/CTS handshake. TO reduce Probability of collision, the Protocols use random delays, it is thus a CSMA-CA protocol (CSMA with collision Avoidance)

→ The timeslots making up the CAP are subdivided into smaller time slots, called backoff periods. One backoff period has length corresponding to 20 channel symbol times & the slots considered by slotted CSMA-CA Protocols are just these backoff periods.

Nonbeaconed mode

The IEEE 802.15.4 protocol offers a nonbeaconed mode beside the beaconed mode. Some difference bet? these modes are

→ In nonbeaconed mode the coordinator doesn't send beacon frames nor is there any GTS mechanism. Lack of beacon packets takes away a good opportunity for devices to acquire time synchronization with coordinator.

→ All packets from devices are transmitted using an unslotted CSMA-CA Protocol. As opposed to slotted CSMA-CA Protocol, there is no synchronization to backoff Period boundaries and in addition the device performs only a single CCA operation.

→ Coordinators must be switched on constantly but devices can follow their own sleep schedule. Devices wakeup for two reasons

(59) (i) to send a data/control packet to the coordinators, or ii) to fetch a packet destined to itself from the coordinator by using the data request/acknowledgement data. The data request packet is sent through the unslotted CSMA-CA mechanism. When the coordinator has a data packet for the device, it transmits it using the unslotted CSMA-CA access method and the device sends an immediate acknowledgement for the data.

How about IEEE 802.11 and bluetooth:

There are two popular and commercially available systems, Bluetooth and IEEE 802.11, whats wrong with these systems.

→ Bluetooth is designed as a WPAN with one major application, the connection of device to a personal computer. It already been used as a prototyping WSNs applications. The PHY is based on a FHSS scheme having a hopping frequency of 1.6KHz. Nodes are organized into Piconets with one master and seven active slave nodes. The slaves follow the hopping sequence set by master. Master polls the active slave continuously. Two major drawbacks are, Bluetooth have the constant need of master node, which spends much energy in polling his slaves. this is not compatible in case of dense WSNs where huge number of master nodes needed. An active slave always need to be on as it doesn't know when to be polled by master. A passive slave has to apply at master to become active slave. Furthermore, each node must have ability to take role of masters or slaves and require considerable complexity. Also first frequency hopping requires tight synchronization betn. the nodes in a piconet.

→ In IEEE 802.11 family of protocols, several physical layers are specified sharing a single MAC protocol. In IEEE 802.11 requires any node x to constantly be in listen mode since another node y attempt to transmit a frame to x at any time. Secondly nodes are

required to overhear RTS and CTS Packet to adjust their NAV timers properly. IEEE 802.11 system is targeted towards high bit rates and available transceivers require orders of magnitude more energy than acceptable in low bit rate sensor n/wk. applications. IEEE 802.11 is a single hop protocol for both infrastructure & adHoc n/wk scenarios and is targeted at letting a number of independent and competing users share a common channel in a fair manner. Those goals don't match the goals of WSNs.