MAIN IOT

Contents

1. Radio propagation

The mechanism of radio propagation can generally be described as reflection, diffraction and scattering. Propagation models are traditionally focused on prediction of the average received signal strength used to define the coverage area in a mobile cellular environment. In addition, some statistical properties of the

radio channel, such as fading parameters, multipath and associated intersymbol interference heavily impact the system design and various system parameters.

1.1. **Free-space propagation.** The simplest possible scenario is that both transmit and receive antenna are placed in a free space. The power received at a destination in case of no obstacles and in LOS[1] can be described as the **Friis transmission equation**:

$$P_r = P_t G_t G_r \left(\frac{\lambda}{4\pi d}\right) \frac{1}{L}$$

where the power received depends on the power transmitted, multiplied for both the gain in transmission and reception; $\frac{1}{L}$ represents hardware loss, λ is the wavelength, and d is the distance between the transmitter and the receiver.

1.1.1. *Path loss.* Path loss, which is the rate between the transmitted power and the received power, can be described as:

$$P_l = \frac{P_t}{P_r} = \left(\frac{\lambda}{4\pi d}\right)^{-2}$$

Intuitively, path loss increases with higher frequencies and greater distance, assuming that the gains and hardware loss are equal to 1.

1.1.2. *Multi-path fading.* Multiple copy of a signal can arrive at destination because the signal may be scattered, reflected, and diffracted. The receiver may get multiple copies of the signal, each travelling different paths and resulting in different attenuation and delay (**delay spread**).

1.2. **Radio cell size.** Many small cells concentrated in highly populated areas are able to maintain a stable connection. Small cells are also faster to deploy and easier to maintain, however they have a shorter range and can handle fewer simultaneous sessions. One the other hand, larger cells must be used to cover wide, low-dense areas.

2. Cellular systems

2.1. **2G: GSM.** In this system the signal is digital and both TDMA and FDMA schemas are employed. Frequency bands are split between downlink and uplink. The **GSM** network can be split into three subsystems: the radio access network, the core network and the management network. These respective three subsystems can also be called the Base-Station Subsystem (BSS), the Network Switching Subsystem (NSS) and the Operation Support Subsystem (OSS).

In order to support the new concept of packet service, the **GPRS** system was deployed as an overlay of GSM, introducing the concept of sharing the pool of available channels in the cell between different users. Several timeslots on one carrier frequency may be allocated to one user (bundling); timeslots can be bundled on uplink and downlink . The allocation of timeslots may also be asymmetric between uplink and downlink. Furthermore, a timeslot may be shared by several users according to their priority or on round-robin basis. **EDGE** is a further evolution of GPRS, giving the option for an increased system data rate without changing the cellular mobile system architecture

[1]Line of Sight, i.e. the signal has not been modied by scattering, reflections, etc.

2.2. **TDMA and FDMA.** In wireless communications, it is necessary to utilize limited frequency bands at the same time, allowing multiple users to share radio channels simultaneously. The scheme that is used for this purpose is called multiple access. To provide simultaneous two-way communications, a forward channel (downlink) from the BS to the MS and a reverse channel (uplink) from the MS to the BS are necessary. Two types of duplex systems are utilized: frequency division duplexing (FDD) and time division duplexing (TDD). If a system employs different carrier frequencies to transmit the signal for each user, it is called a **FDMA** system. If a system uses distinct time slots to transmit the signal for different users, it is a **TDMA** system.

2.2.1. *Base Station Subsystem.* The **BSS** interfaces the mobile users with the cellular system operator network. It manages radio communication and radio resources, configuration of cells, and must handle handovers. It is formed by the Base Transmission Station (**BTS**), whose task is to provide radio coverage on the cell, and the Base Station Controller (**BSC**), which monitors and manages groups of BTS. The BSC tells the BTS when initiate a call or perform a handover, and reserves/relases radio channels.

2.2.2. *Network Switching Subsystem.* Performs routing on the network to the Gateway MSC (**GMSC**). The Home Location Register (**HLR**) mantains information about all the users that had contact with the cellular system, while the Visited Location Register (**VLR**) store the current location of a mobile user.

Two other databases perform **security** functions: the Authentication Centre (**AuC**) stores security-related data such as keys used for authentication and encryption; the Equipment Identity Register (**EIR**) stores information about the hardware of mobile stations. The GSM security is based on information stored in the SIM. When ciphering is active, all information exchanged between the mobile and the network on the dedicated radio channels is encrypted. Encryption is performed at the transmitting side after channel coding and interleaving and immediately preceding modulation. On the receiving side, decryption directly follows the demodulation of the data stream. Authentication parameters are generated by the AuC, stored in the HLR, and must be sent to the VLRs that requests them.

2.2.3. *Handovers and timing advance.* When a MS connects to a cell, the BSC sends a list of alternative channels, whose signal strength should be monitored by the MS. The network might start a **handover** for multiple reasons, and it could be due to the radio link, network management, or service issues. While the user is on call, a new channel must be allocated on the cell, the old channel must be deallocated, and the active connection will be routed to the new channel. Four types of handover can be identified:

- **intra-cell** / **intra-BSC**, triggered by low quality channel but high signal strength
- **inter-cell** / **intra-BSC**, the user is moving to another cell (inside the same BSC)
- **inter-BSC**, the user is moving to a cell of a different BSC
- **inter-MSC**, the user is crossing another MSC

The **timing advance** value corresponds to the length of time a signal takes to reach the BS from a MS. A continually adjusted timing advance value avoids interference

between users in adjacent timeslots, minimizing data loss and maintaining Mobile QoS. In combination with other variables it allows GSM localization to find a device position and track its user.

Handover performance can be defined in terms of probability of rejecting a handover (**Pdrop**) and probability of rejecting a new call (**Pblock**); inside systems which deal with requests for handover as a new incoming request, $Pdrop = Pblock$. If the system reaches its maximum capacity, it is better to block incoming connections rather than loosing one still active. A number of channels can be also reserved for handover requests: this lowers the Pdrop value however the overall capacity decreases.

2.3. **3G: Universal Mobile Telecommunication System.** Being backwards compatible, GSM and UMTS networks are able to inter-operate between them. The **UMTS** system has an overall network structure similar to GSM. The physical layer consists of the UMTS Terrestrial Radio Access Network (**UTRAN**), made of the base station (**nodeB**) and Radio Network Controller (**RNC**), which allows connectivity between the User Equipment (UE) and the core network. The multicarrier technology is Wideband Code Division Multiple-Access (**W-CDMA**), which allows several users to share bandwidth. Furthermore, The **HSPA** technology supports much higher data rates, higher capacity in both uplink and downlink directions and significantly reduces latency compared with the initial WCDMA system.

2.3.1. *CDMA.* Each user is assigned a different code, and multiple users can share the same frequency. A basic implementation of CDMA is **Frequency Hopping**, where a pseudorandom sequence is used to change the radio signal frequency, enabling simultaneous transmissions from several users (as long as non of them collides).

2.3.2. *Energy consumption.* In 3G, a large fraction of the energy, referred to as the **tail energy**, is wasted in high-power states after the completion of a typical transfer. The ramp energy on the other end, is described as the energy spent to switch to a high power state, and does not consume much. The tail-ender is a protocol designed to reduce the waste from tail energy, and does so in three ways: combine the usage of 3G and wifi, transfer data in batch for delay tolerant applications, and prefetch for application that may benefit from it.

2.4. **4G: Long-Term Evolution.** New requirements for capacity, user experience and higher cost-efficiency, have led to the specification of the **LTE** standard. A further goal was the redesign and simplification of the network architecture to a packed-switched model. The new high-level network is made of three main components: the User Equipment (**UE**), the Evolved Packet Core (**EPC**), and the Evolved-UTRAN (made of **e-nodeB**) handling communications between UE and EPC.

The **E-UTRAN** protocol stack offers increased peak rates, lower data transfer latency, lower power consumption, and interoperability with other radio technologies. The multicarrier technology is **OFDMA** in downlink and **SC-FDMA** in uplink, with support for MIMO antennas, i.e. the use of multiple antennas at the transmitter and receiver ends.

2.4.1. *OFDMA.* A high-rate radio channel is split into multiple low-rate subchannels, simultaneously transmitted over multiple carrier frequencies. Each user is then mapped to an orthogonal sub-carrier, which means they do not interfere with each other.

2.5. **5G: New Radio.** Strong need for ubiquitous ultra-high broadband access and QoS requirements define the fifth generation (5G) mobile network. Initial 5G/NR launches will depend on existing 4G/LTE infrastructure in non-standalone mode, before maturation of the standalone mode with the 5G core network.

3. Ad-hoc wireless networks

An ad-hoc wireless network is a wireless multi-hop infrastructure-less network whose devices act as source and/or destination of messages and act as relay for packets; it self organizes, self congure and self maintain. Main features include: highly dynamic topology, simple protocols with low energy consumption and low overhead (ideally), number of hosts depends on the requirements, traffic can be low to high.

3.1. **Wireless Sensor Network. WSN** networks have a mostly static topology, e.g. networks to monitor the environment, but it is also possible that hosts move. Ideally, the number of hosts must not degrade the performance. Hosts are typically sensors with limited battery capacity and low processing power, so protocols must be simple, low energy consuming, and light, since the memory is also limited; traffic can flows from sensors to one ore more sink, where data is gathered and processed.

3.1.1. *CSMA.* **CSMA** is a MAC protocol in which a node tries to detect the presence of a signal from another node, before attempting to transmit. Using CSMA, multiple nodes may, in turn, send and receive on the same medium. Carrier-sensing can be of two types, (i) physical by detecting activity on radio, or (ii) **virtual**, by sending RTS/CTS control packets. Furthermore, two common variants of CSMA exists, based on (i) Collision Detection (**CSMA/CD**) or (ii) Collision Avoidance (**CSMA/CA**). Since CSMA/CD is not possible in wireless networks, due to wireless transmitters desensing their receivers during packet transmission, CSMA/CA has to be used, however it is unreliable due to the **hidden terminal** problem. **Virtual carrier sensing** can be used to mitigate the hidden terminal phenomenon:

(1) the source sends a RTS frame after the medium has been sensed idle for a time interval exceeding **DIFS**
(2) on RTS arrival, the destination responds with a CTS frame, which can be sent after the medium has been sensed idle for a time interval exceeding **SIFS**
(3) on a successful RTS/CTS exchange, the data frame can be sent after waiting for a time interval SIFS
(4) on receiving a data frame, the destination waits for a time interval SIFS and sends an ACK. Since $SIFS < DIFS$, no other node is expected to use the shared medium at that time

If the handshake is not correctly completed, the node performs a retransmission attempt after an exponential backoff. The RTS/CTS frame includes a NAV field, which is used to estimate for how long nodes should sleep, before they can try to access the medium again.

3.2. **Cross-factor.** A substantial fraction of energy is consumed by the processing of packets throughout the protocol stack. Such **cross-factor** energy toll exhibits two notable features: (i) in some (common) radio settings, it may become the dominant source of energy consumption, and (ii) it is primarily associated to the frame processing itself, rather than to the amount of bytes handled.

The power consumed by an 802.11 device consists of the following components: (i) the idle consumption, ρ_{id}, (ii) the cross-factor for the packets generated by the application, P_{xg}, (iii) the power required to transmit them, P_{tx}, (vi) the power consumed in retransmissions, P_{retx}, (v) the power spent in receiving frames, P_{rx},

(vi) the cross-factor for the received frames, P_{xr}, and (vii) the power spent on sending and receiving ACK frames, $P_{rx,ack}$ and $P_{tx,ack}$:

$$P = \rho_{id} + P_{xg} + P_{tx} + P_{retx} + P_{rx} + P_{xr} + P_{rx,ack} + P_{tx,ack}$$

4. Routing protocols for ad-hoc networks

There are two standard approaches for intra-AS routing: **link state**, where each switching node keeps information on the topology and then applies shortest path algorithms, e.g. Dijkstra, to construct its own routing table; **distance vector**, where switching nodes exchange information (such as the routing table and hop count) and determinate the best route through the Bellman-Ford algorithm. However, ad-hoc wireless networks have different needs from standard networks, and routing protocol should have: minimal control overhead, low processing overhead, no loops, multi-hop path capability, dynamic topology maintenance, and it must be self-starting.

4.1. **Proactive routing protocol.** This type of protocols maintains routes to each other node by periodically distributing routing tables throughout the network. This means that routes are alway available. The main disadvantages of such algorithms are (i) high overheads in high mobility networks, (ii) low scalability, and (iii) slow route convergence time.

4.1.1. *Destination Sequence Distance Vector.* Based on the Distributed Bellman-Ford (**DBF**) algorithm, incorporating sequence numbers to avoid routing loops. Each routing table is tagged with a sequence number which is originated by the destination. To maintain the consistency of routing tables in a dynamic topology, each node periodically transmits updates, and transmits updates immediately when significant new informations are available. Broken links can be detected by layer-2 protocols, or it may instead be inferred if no broadcasts have been received for a while from a former neighbor. Any route with a more recent sequence number is preferred, while routes with older sequence numbers are discarded.

4.1.2. *Optimized Link State Routing Protocol.* Nodes periodically transmit **HELLO** messages, containing the sender address and its own neighbor, including each link status. Upon receiving a HELLO message, a node can thus gather informations about its neighborhood and two-hop neighborhood, as well as the quality of their link. Such informations are considered valid for a short period of time.

In **OLSR**, the problem of duplicate messages (within a region) is addressed through the notion of Multi-Point Relays (**MPR**). Each node selects its MPRs independently, which must be able to reach all the node's two-hop neighbors. All nodes with a non-empty MPR set periodically generate a topology control message, effectively announcing reachability to all its MPRs. The result is that all nodes will receive a partial topology graph of the network, and it is possible to run a shortest path algorithm for computing optimal routes from a nodes to any reacheable destination.

4.2. **Reactive routing protocol.** This type of protocol finds a route on-demand by flooding the network with Route Request packets, resulting in less control overhead and better scalability. Nodes that do not lie on active paths neither maintain any routing information nor participate in any periodic routing table exchanges.

The main disadvantages of such algorithms are (i) high delays in route finding, and (ii) excessive flooding leading to bottlenecks.

4.2.1. *Dynamic Source Routing.* The sender builds a **source route** in the packet header, giving the address of each host in the network through which the packet should be forwarded in order to reach the destination host. When a host receives a packet, if this host is not the final destination of the packet, it simply forwards it to the next hop in the source route.

Each node maintains a **route cache** in which it caches source routes it has learned. When one host sends a packet to another host, the sender first checks its route cache for a source route to destination. If no route is found, the sender starts the **route discovery** protocol: it broadcasts a **route request** packet to its neighbor identifying the destination, and (if successful) it receives a **route reply** listing a sequence of hops through which it may reach the destination.

While a source route is in use, the node does **route maintenance**, monitoring its working state. If it detects a problem, route discovery may be used again to discover a new, correct route to the destination.

4.2.2. *Ad hoc On-demand Distance Vector.* **AODV** relies on dynamically enstablishing routing table entries at intermediate nodes. This difference pays off in highly dense networks, where a large overhead is incurred by carrying source routes as in DSR. To maintain fresh routing information between nodes, sequence numbers are used as in DSDV, but each node maintains a monotonically increasing sequence number counter which is used to supersede stale cached routes.

The **path discovery** process is initiated whenever a source node needs to communicate with another node for which it has no routing informations in its table. The source node broadcasts a Route Request (**RREQ**) packet to its neighbor, which may respond with a Route Reply (**RREP**) back to the source or rebroadcast the RREQ to their own neighbor, after increasing the hop counter. When an intermediate node receives a RREQ, if it has already received a RREQ with the same ID and source address, it simply drops the redundant packet.

The intermediate node can reply only when it has a route with a sequence number that is greater than or equal to that contained in the RREQ. If it does have an current route to the destination, and if the RREQ has not been processed previously, the node then unicasts a RREP back to its neighbor from which it received the RREQ. A node receiving a RREP propagates the first RREP for a given source node towards that source, which can begin data transmission ad can update its routing informations (if it learns a better route).

4.3. **Geographical routing protocol.** Routing decisions are not based on network addresses and routing tables; instead, messages are routed towards a destination location. With knowledge of the neighbors' location, each node can select the next-hop that is closer to the destination, and thus advance towards the destination in each step.

The fact that neither routing tables nor route discovery activities are necessary makes geographic routing a good fit for dynamic networks such as wireless adhoc and sensor networks. In such networks, acquiring and maintaining routing information is costly as it involves additional message transmissions that require energy and bandwidth and frequent updates. In contrast, georouting algorithms work nearly stateless and can provide high message delivery rates under mobility,

assuming that (i) nodes can determine their own position, (ii) nodes are aware of their neighbors' positions, and (iii) the destination position is known.

The main prerequisite to meet the three assumptions is a positioning system. If this is available, geographic routing provides an efficient and scalable solution for routing in wireless and mobile networks. However, a simple **greedy forwarding** by minimizing the distance to the destination location in each step cannot guarantee message delivery. Greedy algorithms cannot resolve such dead-end or local minimum situation. Therefore, recovery methods have been developed to improve greedy forwarding.

4.3.1. *Location-Aided Routing.* When a source nodes needs to find a route to a destination node, it broadcasts a RREQ message to all its neighbor. A node receiving a message first compares the destination with its identifier. If it does not match, the node broadcasts the request to its neighbor. Duplicate requests are identified through sequence numbers, and discarded. As the RREQ travels the network, the path followed is included in the packet. On receiving the RREQ, the destination sends a RREP to the sender. If the next hop on the route is broken, nodes send a RERR message, and initiates a route discovery for that destination.

Furthermore, **LAR** makes use of location information to reduce routing overhead. The expected zone of a destination node D, from the viewpoint of a source node S at time t_1, is the region that node S expects to contain node D at time t_1. Node S can calculate the expected zone based on the knowledge that node D was at a certain location a time t_0. If node S does not know a previous location of node D, the expected zone may cover the entire network, reducing the algorithm to a basic flooding algorithm. Node S defines a request zone for the RREQ. A node forwards a RREQ only if it belongs to the expected zone.

4.3.2. *Geographic Random Forwarding.* Nodes follow a duty cycle, with asynchronous awake/asleep schedules. The **forwarding area** of each node is divided into N regions, 1 through N, such that any node in region i is closer to the sink than any node in region j. Whenever a node wants to send a packet it first senses the channel, then transmits a Request-to-Send (RTS) message that silences nodes offering negative advancement and serves as a polling message for region 1. The awake nodes in this region report back using a Clear-to-Send (CTS) message. If more than one respond, the sender issues a COLLISION message to solicit the choice of a single node. After the identification of the next hop, the node sends a data packet, waits for an ACK, and then the sender goes back to sleep, so that the relay finds the channel free and can forward the packet. If no relay is found in any region or the channel is sensed busy, the transmitter backs off and reschedules a later attempt.

Note that, in **GeRaF**, nodes are not required to know the position of any other sensor but their own and the sink's, since any other information is exchanged through RTSs and CTSs. Also nodes do not need to know their neighbors and their wake-up schedule: a relay is selected among the awake neighbors.

4.3.3. *Adaptive Load-Balanced Algorithm.* **ALBA** is designed to take congestion and traffic load balancing into consideration, other than just advancement. All the eligible relays of a node compute two values, the Geographic Priority Index (**GPI**), i.e. the index of the region the node would belong to in the GeRaF framework, and the Queue Priority Index (**QPI**), which is a measure of forwarding effectiveness as perceived by the relay.

To solicit the election of a forwarder, the source first senses the channel and then issues RTS messages to scan the QPI values in increasing order. If more than one nodes answer the RTS message, the sender tries to select the one with the best GPI, starting a second contention with progressively higher GPIs. This process eventually ends with the selection of a single relay. If no relay is found, the sender backs off. Like GeRaF, ALBA exchanges relevant data such as the geographic coordinates through RTSs and CTSs, requiring little information to be stored inside sensors.

In order to solve the dead-end problem, **ALBA-R** introduces the **Rainbow** node-coloring algorithm, with no need for additional signaling packets. Nodes that recognize themselves as dead-ends progressively stop advertising as relays, and to route traffic out of the dead-end, they begin to transmit packets backward, in the negative advancement zone. Hopefully, a relay that has a greedy forwarding path to the sink will be found.

5. Routing protcols for Wireless Sensor Networks

5.1. **Collection Tree Protocol.** The idea is to send beacon messages to build a tree structure, where nodes can relay their information to their best parent. The number of expected transmissions needed to send data between two nodes, **ETX**, is used as the routing metric. In a route that includes multiple hops, the metric is the sum of the ETX of the individual hops. Each node that wishes to collect data advertises itself as a root, which has always an ETX of zero. Each node sends its data to the nearest root, i.e. the root from which it is separated by the smallest ETX. Each node only keeps the smallest ETX (to the nearest root). Rapidly changing link qualities, e.g. sensor networks with moving nodes, cause routing information to become outdated which can lead to routing loops. CTP attempts to address these issues through datapath validation and adaptive beaconing.

Beacon messages are transmitted regularly when the sink starts the beaconing process, to exchange informations about ETXs. Data packets embed the estimated ETX, and sequence numbers allow to know the percentage of packets that has been received (link quality). The Bellman-Ford algorithm is used to compute the best route, with ETX values as weights. When a node receives a packet, if its current ETX is smaller than the one embedded into the packet, the node updates the packet's ETX and forwards it. Otherwise, the route must be recomputed by starting a beaconing process.

5.2. **Wake-up enabled protocols.** Many recent activities of IEEE 802.11 Working Group have been focused on improving power efficiency of Wi-Fi to make it viable for massive IoT scenarios, in which swarms of battery supplied sensors rarely communicate with remote servers. The 802.11ba standard introduces Wake-Up Radios (**WuR**), low-power hardware in charge of monitoring the channel. The use of wake-up radios enables on-demand communications, allowing nodes to keep their main radio off when not needed, and virtually eliminating idle listening.

5.2.1. *FLOOD-WUP.* A new flooding protocol, which can solve broadcast storm and collisions, while achieving high reliability and better latency vs. energy consumption performance. In **FLOOD-WUP** nodes are assigned shared broadcast addresses. We assume that each node is assigned two **broadcasting wake-up addresses**, named w_a and w_b. The sink sends the first broadcast packet to its

neighbors, preceding it with the wake-up sequence w_a. The following broadcast packet will be sent using the wake-up sequence w_b, the third one using the wake-up sequence w_a, and so on. Nodes in the network are initially asleep, with the wake-up radio active with address set to w_a. When a node receives a wake-up signal with sequence w_a, it wakes up and sets its radio to receive the broadcast packet B. Upon reception of B, each node changes its broadcast wake-up address from w_a to w_b. After receiving a broadcast packet, each node can wait a random time and retransmit the packet, by using the same wake-up sequence it was received with.

Nodes consume energy only when they receive or transmit packets. Energy consumption due to reception is limited to the reception of the first copy of the broadcast packet, as nodes do not wake up when duplicate broadcast packets are transmitted. Energy consumption for transmission is limited to the transmission of the broadcast packet and the wake-up sequence once.

5.2.2. *GREEN-WUP.* Packet forwarding is realized by opportunistically selecting the next hop relay among the whole set of neighbors, rather than just among the awake ones. To this end, we assume that nodes activate a subset of possible **unicast wake-up addresses** based on their state. In particular, nodes activate a given wake-up sequence w_i, i.e. they wake up if such sequence is transmitted, in case they are in the corresponding state i. Such a state expresses how good a node is to serve as a relay, and can be computed based on factors such as its current and expected energy intake from harvesting, its residual energy, its hop count, and so on.

In the specific case of **GREEN-WUP**, each node is associated with a wake-up address w composed by two subsequences, i.e. $w = w_l w_e$. The first subsequence, w_l, is set based on the node hop count. The second subsequence, w_e, is set by each node based on its residual energy. By computing its fractional residual energy, each node determines the class index i it belongs to, and it sets the subsequence w_e of its address accordingly.

6. Energy-efficient MAC protocols

A low-power MAC protocol aims to minimize energy waste - introduced by collisions, overhearing, control packets overhead, idle listening - and minimize end-to-end latency, while guaranteeing fairness and scalability. Among all reasons mentioned above idle listening is a major cause of energy waste. So it is important to design a suitable MAC protocol which can reduce or prevent energy wastes. There are four techniques to avoid idle listening - static sleep scheduling, dynamic sleep scheduling, preamble sampling, and off-line scheduling. Based on these techniques, many MAC protocols adopt CSMA.

6.1. **Sensor MAC.** At startup, nodes listen on the medium. If a node hears nothing it sends a SYNC packet with a schedule defining listen and sleep periods, and becomes a **synchronizer**. On the other hand, if a node receives a SYNC packet, it becomes a **follower**, and adopts the corresponding schedule. Border nodes may adopt two or more schedules (if different neighbors have different schedules), resulting in more energy consumption.

Each node keeps a table with the schedules of their neighbors. During a listen period, a node with a packet to send executes a procedure similar to CSMA/CA, transmitting a request-to-send (RTS) frame and the receiver node answering with

a clear-to-send (CTS) frame. All nodes not involved in the conversation enter a sleep state while the communicating nodes send data packets and ACKs. The transmission of ACKs are sufficient to deal with the hidden terminal problem. Sleeping decreases energy consumption but introduces latency, since communication with a sleeping node must wait until it wakes up. Furthermore, clocks may drift and SYNC packets may go lost, so they have to be frequently exchanged (introducing more control overhead).

6.2. **Timeout MAC. T-MAC** improves on **S-MAC** to find a balance between wake time (potentially wasting energy in idle listening) and the repetition of the duty cycle (potentially increasing latency). Active time is changed dynamically adapting on the traffic level of the network: higher the traffic, longer the active time. If a node does not receives transmissions from neighbors for a period of time, the active time is aborted and the node will go to sleep. Nodes do not exchange asleep/awake schedules, since a fixed schedule is not good in practice. T-MAC still suffers of low throughput, since every channel is active only for a brief period of time. Furthermore, synchronization is very inefficient because of clock drifting.

6.3. **Berkeley MAC. B-MAC** employs an adaptive preamble to reduce idle listening, a major source of energy usage in many protocols. When a node has a packet to send, it waits a backoff time before checking the channel. If the channel is clear, the node transmits; otherwise it begins a second (congestion) backoff. If the channel is idle and the node has no data to transmit, the node returns to sleep. B-MAC does not require RTS, CTS, ACK, or any other control frame by default, and no synchronization is required. However, the preamble creates large overhead, consumes more energy in transmission, and every node that overhear the preamble will stay awake until the end.

6.4. **X-MAC. X-MAC** improves upon B-MAC by embedding into the preamble the address of the destination, so nodes overhearing can return to sleep. The preamble is a series of short preambles divided by pauses: this allows the destination to send an ACK to interrupt the preamble and to start the transmission of data.

6.5. **Wireless sensor MAC.** Needs a network infrastructure composed of several AP, each serving a number of sensor. In **downlink**, the AP can transmit configuration messages and query request from sensors, and transmit without sensors always listening. In **uplink**, a sensor can transmit acquired data, and the AP can listen without energy constraints.

WiseMAC is based on CSMA with **preamble sampling**: a sensor regularly listens on the channel; if the channel is busy, it listens until the frame is received or the channel is idle again. The AP transmits a wake-up preamble in front of every data frame, ensuring the receiver will be awake when the frame arrives. The preamble is minimized by exploiting knowledge of sensors schedule. With every ACK apackets comes an up-to-date sampling schedule.

7. Low-Power WAN

7.1. **LoRa.** Defines a proprietary PHY and an open LoRaWAN MAC layer on top of it. The **physical layer** objective is to reach long distances by encoding signals in a larger spectrum, i.e. increase the frequency bandwidth to protect it from noise. End-nodes are connected to gateways through **LoRa-links**, while gateways can

be connected to other networks through LoRa-links or other technologies such as cellular systems or ethernet.

The **LoRaWAN** protocol defines three different device classes, where A is the most energy efficient, followed by B and then C. **Gateways** collect frames from sensors and relay them to the **network server**, which identify and filter duplicates, validate data and send it to the correct application server. A **frame** starts with a preamble, followed by header and CRC, then the payload. Furthermore, there are two layers of **security**: network and application. The former authenticates users, adds integrity checks, and encrypts the whole payload. The latter encrypts the payload.

LoRaWAN also describes a join procedure prior to participating in data exchange with the network server (**Over The Air Activation**). Every node must have a device ID and an application ID, where the second is the join server and is used for session key derivation. Furthermore, a node must have an application key, and a network key (which is given by the network). When a node wants to join, it will send a request to the network server, with a specific join ID and device ID. The network server replies with a message that is encrypted with the network key.

7.2. **Narrowband IoT.** A radio standard for cellular devices and services. **NB-IoT** is a subset of LTE, with limited bandwidth, focusing on indoor coverage, low costs, and long battery life. It defines two modes for uplink and three coverage classes (normal, robust, extreme).

8. Low-Rate WPAN

8.1. **IEEE 802.15.4.** Defines the PHY and MAC (CSMA/CA) layers for WPANs focused on low-cost, low-speed communication between devices. Networks can be built as either P2P or star topologies, with at least one FFD coordinator. In order to address several limitations, the IEEE working group has released an extension of the standard, which includes realtime guarantees, resilience to interferences, and the ability to increase capacity. The IEEE 802.15.4*e* introduces new features such as BLINK, AMCA, DSME, LLDN, and TSCH.

8.2. **6LoWPAN / Thread IoT.** Defines encapsulation and header compression mechanisms which allow IPv6 packets to be sent and received over IEEE 802.15.4 based networks. A **6LoWPAN** network is connected to an IPv6 network through an **edge router**, which handles (i) data exchange between 6LoWPAN devices and other networks; (ii) local data exchange between devices inside the 6LoWPAN; and (iii) the maintenance of the radio subnet. Edge routers may also support IPv6 transition mechanisms to connect to IPv4 networks. Like other networks with edge routers, it does not maintain any application layer state, because such networks forward datagrams at the network layer. This means that 6LoWPAN remains unaware of application protocols and changes. This lowers the processing power burden on edge routers.

6LoWPAN introduces an **adaption layer** between the link and network layer to enable transmission of IPv6 datagrams over 802.15.4 radio links. In order to improve transmissions over low-power and lossy networks, a new **header compression** mechanism is defined. Header fields are elided when they (i) can be derived from the link layer, (ii) carry common values, or (iii) assumptions such as a common network prefix can be made.