panikzettel.htwr-aachen.de

Mobile Internet Technology

Version 2 | 24.10.2025

Jonas Schneider Adrian Groh

Contents

1 Preface and Prior Knowledge

This Panikzettel ("panic note") is open source at `https://github.com/htwr-aachen/panikzettel` . We encourage any and all contribution from students and official sources. See https://htwr-aachen.de/docs/panikzettel .

As this is a communication systems subject, knowledge of Ethernet, MAC and IP is assumed. Take a look over the Datkom Panikzettel to refresh.

2 Physical Layer

This subject concerns wireless communication and especially the two most used communication standards, 802.11 (Wi-Fi) and xG telecommunication networks. Wireless communication provides ways to let two systems communicate without directly connected via wire. The communication is encoded into electromagnetic waves that have the following characteristics:

- A Amplitude
- f: Frequency in $\frac{1}{s} = \text{Hz}$
- φ: Phase in rad

Electromagnetic waves travel with light speed and as such one up and down *wave* has a *length* $\lambda = \frac{c}{f}$

To encode binary data onto a fixed wave, we have to modulate (change) some wave characteristic over time. In this topic the notion of *bandwidth* is not really intuitively explainable especially in panic like you are currently.

Definition: Bandwidth

Width of frequency band needed to represent a signal

Trying it though: when decomposing a complex wave into its subcomponents with the Fourier transform we get waves of different frequencies. The range of these frequencies forms the bandwidth that we need.

The most basic and intuitive forms of modulation / keying (modification of a basic wave, later transferred to the actual carrier frequency f_c) are the **Amplitude Shift Keying** (ASK) and **Frequency Shift Keying** (FSK) modifying the amplitude and frequency on `0` or `1` respectively. ASK does not need much bandwidth, but is susceptible to distortion or noise. FSK needs much more bandwidth.

One improvement is the **Minimum Shift Keying** (MSK), which does not encode bits but bit transitions (somewhat similar to Manchester encoding on wires), where one of the frequencies is twice the other. This is used in GSM in combination with a Gaussian Low-Pass filter (GMSK). Figure 1 provides a visualization.

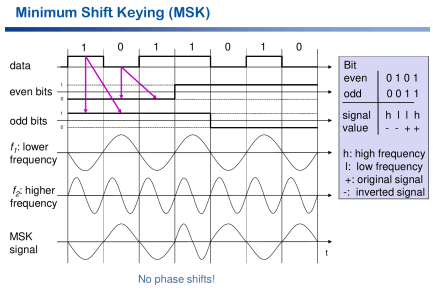

Figure 1: Minimum Shift Keying

The third keying is the phase. **Phase Shift Keying** modulates the phase of a wave. It is more robust against distortion than ASK and uses less bandwidth than FSK.

There are many different ways to modify the phase of the wave, an intuitive one is **Binary Phase Shift Keying** (BPSK), where `0` is a plain sine wave, `1` is inverted sine wave 180° phase shifted. Very robust, but inefficient bandwidth use.

More advanced versions are **Quaternary Phase Shift Keying** (QPSK), where we encodes two bits per signal by using four different phases. In Figure 2 each Ring is a QPSK. The tradeoff is robustness vs efficiency.

Quadrature Amplitude Modulation (QAM)

In **Quadrature Amplitude Modulation** we combine amplitude and phase keying to encode m bits using only one signal. There are 2^m discrete levels. This is widely used in modern communication where older standards tend to use e.g. 16- or 64-QAM and newer Wi-Fi 7 can go up to 4096-QAM. Figure 2 shows 16-QAM.

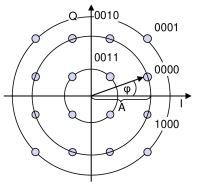

Figure 2: 16-QAM

Wireless communication has many advantages over wires, but comes with many challenges.

- Restricted frequency bands
- Low transmission rates
- High loss rates
- Higher delays/jitter
- Lower security

Nyquist and Shannon

Nyquist Theorem

Nyquist provides a maximum for our signal / data rate for a given Bandwidth. Of course as with all these theorems, this is only a theoretical maximum.

Definition: Signal Rate

Possible number of signal parameter changes per second

$$S_{\text{max}}\ [\text{Hz, baud}] = 2 \cdot B\ [\text{Hz}]$$

This signal rate is not directly translatable to bits, for this the encoding is needed.

Satz: Nyquist Theorem

$$R_{\text{max}}\ [\text{bit/s}] = 2 \cdot B\ [\text{Hz}] \cdot \text{ld}(n)\ [\text{bit}]$$

where R_{max} is the maximum possible data rate, B is the Bandwidth and n is the number of states of the encoding

Here 16-QAM gives us $\text{ld}(16) = \log_2(16) = 4$ bits per signal.

Shannon Theorem

Shannon provides us with another theoretical, this time in combination with the noise in our environment.

Definition: Signal-to-noise ratio

Power of a signal S relative to the power of noise N

$$\text{SNR} = \frac{S}{N}$$

Usually given in *dezibel* [dB]: $\text{SNR}_{\text{db}} = 10 \cdot \log_{10}\left(\frac{S}{N}\right)$

Satz: Shannon Theorem

$$R_{\text{max}}\ [\text{bit/s}] = B\ [\text{Hz}] \cdot \text{ld}\left(1 + \frac{S}{N}\right)\ [\text{bit}]$$

Since in reality, both bandwidth and noise are constraints, the minimum of Nyquist and Shannon theorems gives the maximum achievable data rate. In the exam there is probably a question with a combination of both systems, where Shannon gives a maxmimum for a given noise level and Nyquist a then theoretical achievable maximum due to constraints of the encoding (we can either select BPSK, 16, 32 or 64-QAM nothing in the middle) which reduces the possible data rate further.

2.1 Signal Propagation

The received power P_r of a signal can be calculated as follows:

$$P_r = P_t \cdot \left(\frac{\lambda}{4\pi}\right)^2 \cdot \left(\frac{1}{d}\right)^{\alpha} \cdot G_r \cdot G_t$$

P_r : received signal power

P_t : transmitted signal power

G_r : gain of receiver antenna

G_t : gain of transmitter antenna

λ : wave length of carrier frequency

d : distance between transmitter and receiver

α : Environmental factor

$\alpha = 2$ in a vacuum for a straight-line path. In reality it is more, $\alpha \in (2.7; 5)$ in a city and $\alpha \in (4; 6)$ indoors.

The received power is influenced by attenuation and environmental factors shown in

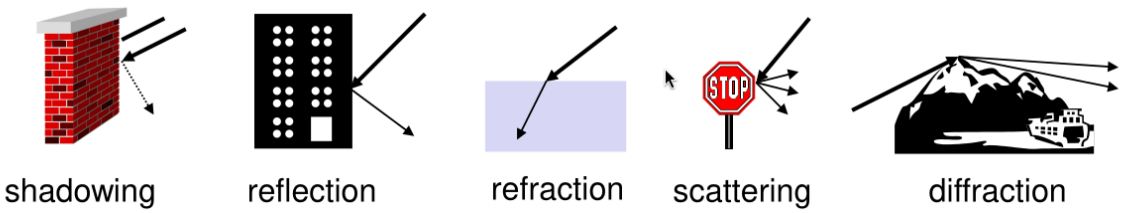

Figure 3: Power Influencers

Multipath Propagation

Signals often take multiple paths of different length to the same destination (line of sight and bouncing from a stop sign to you), causing the same signal to be received multiple times with a short delay with a different phase and different signal strengths. This causes interference and noise. This is especially bad when the last signal we send interferes with the next one, which is called **Inter-Symbol Interference** (ISI)

In mobile communication, channel characteristics change over time, enhancing this effect.

2.2 Multiplexing

We have a given bandwidth i.e. $2.412 - 2.484$GHz, how do we make use of this bandwidth, so that multiple people can use our Wi-Fi. This is called **multiplexing**

Frequency Multiplex

In **Frequency Multiplexing** the spectrum is separated into smaller frequency bands. This does not require any sort of dynamic coordination, but wastes bandwidth if the traffic is distributed unevenly. We also need to keep guard frequencies between the bands to account for inaccuracies and noise.

We can also use this to allow duplex communication (one sub-band for sending, one for receiving).

Time Multiplex

Each sender gets the whole spectrum for a certain time slot. However, this requires precise time synchronization between hosts thus needing guard times to avoid overlaps caused by delays because of the speed of light.

Can also be used for Duplex communication (Time Division Duplex (TDD)), where downlink and uplink are separated with timeslots or combined with Frequency Multiplexing like GSM.

Code Division Multiplex (CDM)

All channels use the same spectrum at once. Each sender has a unique code (chipping sequence) that the receiver can use to filter out a specific sender. The used chipping sequences should be orthogonal to each other.

Sadly this requires perfect synchronization as well. A visualization is shown in Figure 4.

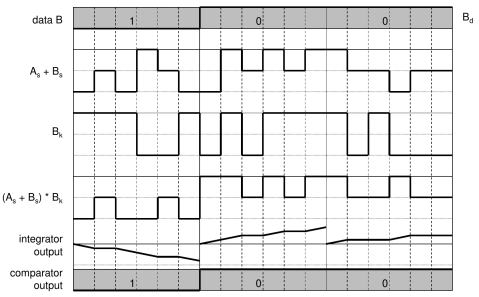

Figure 4: CDM

Direct Sequence Spread Spectrum (DSSS)

Here we XOR the signal with a pseudo-random number chipping sequence. Requires precise power control because of the near-far problem, where a nearby station drowns out the sender which is further away.

Frequency Hopping Spread Spectrum (FHSS)

Change the carrier frequency by a sequence code. I guess hoping from frequency to frequency

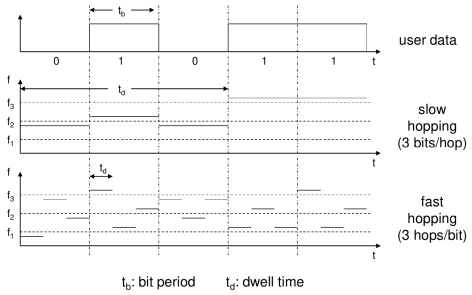

Figure 5: FHSS

Orthogonal Frequency Division Multiplexing (OFDM)

A given bandwidth b is subdivided into n sub-bands on which several narrowband signals are transmitted simultaneously. The sub-channels are orthogonal, which means that the maximum signal power of one sub-channel is on the same frequency as the minimum of the two neighboring sub-channels. The sub-streams are merged/separated using Fourier transform, which can be implemented very efficiently with fast Fourier transform (FFT). This leads to more efficient bandwidth usage than regular FDM.

Space Multiplexing

By limiting the transmit power, the range is restricted to only one pair of sender and receiver. Outside of that range, other transmissions on the same frequency are possible. This is only really useful in combination with other multiplexing methods, as there are typically multiple transmitters within range of one another. In GSM: cell sizes vary from 100m to 35km

2.3 Coding

In wireless communication, the Bit Error Rate (BER) is relatively high with the Packet Error Probability $P_p = 1 - (1 - \text{BER})^n$, the probability that a packet with a length of n bits contains an error. This means that for packets of length 12.000 bit and a BER of 10^{-3}, almost all frames (99.999%) are damaged during transmission.

To deal with this, forward error correcting codes (we always send this with our data), like the hamming code, are used.

Block Codes

Data is processed in blocks where input blocks have k bit and protected output blocks have n bit.

E.g. $(255, 247)$ code protects 247 data bits with 8 FEC-bits.

The disadvantage is that expensive calculations are necessary in certain time intervals (when a block is full). And we need to wait for blocks.

Convolutional Codes

In addition to n and k, a constraint factor K is used that specifies the amount of previous blocks the code is also dependent on. E.g. for an (n, k, K) code: the n-bit long output does not only depend on the k Bits of the current block but also on the previous $K-1$ blocks. $\Rightarrow n$ output bits are a function of the last $K \cdot k$ input bits. An encoder for this can be represented as a shift register and state diagram or a trellis diagram.

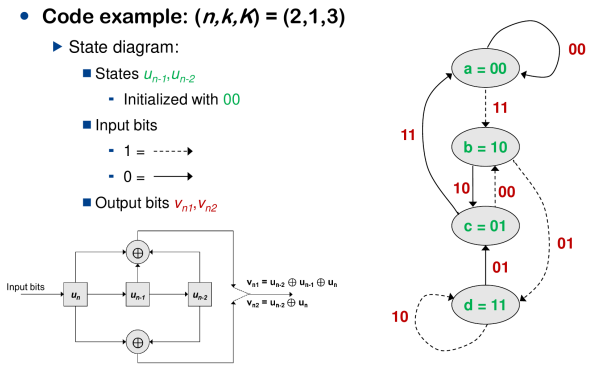

Figure 6: Convolutional Code

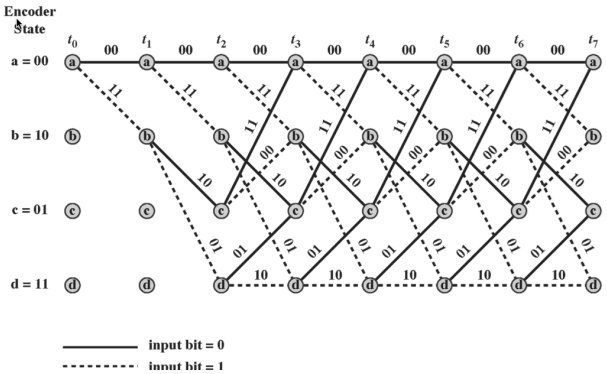

Figure 7: Trellis diagram

Correction Capabilities

Satz: Free Distance

Hamming distance $d \geq 2 \cdot t + 1$ can correct t errors in a block

For a convolutional code we use the Viterbi algorithm

Algorithmus: Viterbi

Basic idea: check if received string gives a valid path through the trellis

1. Compare received bit sequence with all possible transmitted sequences
2. Choose path with lowest hamming distance to received sequence
3. Only select valid paths (always start at (0,0) but most often they are **terminated** and need to end in (0,0) as well)

Turbo Codes are a development on convolutional codes and provide faster decoding.

3 Medium Access Control (MAC)

Medium access provides the rules for transmitting, so that everybody can clearly listen in. This can happen multiplexed but not necessarily. Controlling medium access is especially important in wireless networks, as everybody can hear each other (within range) and everything would interfere otherwise. A second strong argument is the notion of power consumption which is far more important on probably mobile nodes without a direct power source.

A problem with just complete multiplexing is, that it is often too inflexible and the bandwidth could be used more efficiently if traffic has many bursts.

3.1 The Hidden and Exposed Station Problem

Carrier Sense Multiple Access / Collision Detection is the standard MAC algorithm for Ethernet.

This does not really work in wireless networks because of

- Hidden station problem: where a further station sends to our target, which we cannot detect. See Figure 9.
- Exposed station problem: where a nearby station sends to a different station which blocks us. See Figure 10.

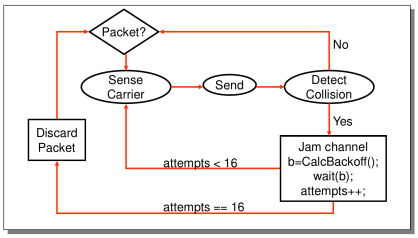

Figure 8: State Diagram of CSMA/CD

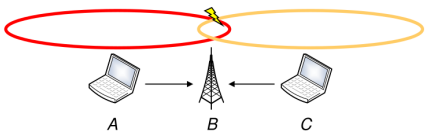

Figure 9: Hidden Station Problem

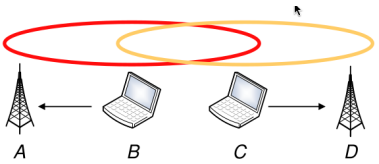

Figure 10: Exposed Station Problem

In general due to these two problems, detecting collisions in wireless networks is hard, because sending and receiving on the same channel is not really possible (can't listen for interference while transmitting). So new MAC schemes have been development specialized for wireless communication. These can be divided into *centralized* and *distributed*, as well as *schedule-* and *contention*-based.

In a centralized MAC algorithm a "central" station defines when which node may have access to the medium. We will focus on the distributed ones (coming back a bit with Beacon frames of Wi-Fi).

3.2 Schedule-based MAC

A schedule (either fixed or computed on demand) regulates, which participant may use the medium at which time. This obviously requires time synchronization similar to time multiplexing.

One example of which is **Demand Assigned Multiple Access** (DAMA) where a sender reserves future time slots and therefore no collisions in the actual transmission are possible if everybody receives the reservation. Fun Fact: This is typically used in satellite links.

| Question: How do we reserve time slots?

Here DAMA is pretty pragmatic and we allow everybody to attempt a reservation and take the ones which succeed. Everybody keeps a full list of reservations. (ALOHA-Phase)

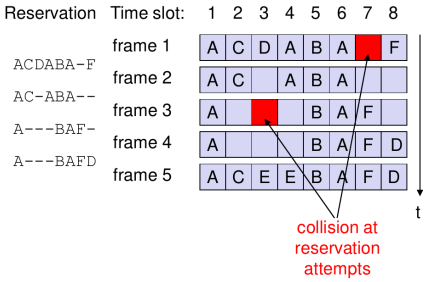

Figure 11: DAMA Example

3.3 Contention-based MAC

Just hoping that having collisions and dealing with them is more efficient than coordinating medium access is the premise of **Contention-based protocols**. This usually involves randomization in terms of backoff.

The simplest ones are the ALOHA protocol suite. Just send the data, when no `ACK` is received before a timeout, retransmit. This results in $\sim 18\%$ efficiency. We have no timeslots we can start transmitting the middle of another transmission. Variant: Slotted Aloha, have fixed timeslots in which sending is possible the rest is the same, but this makes collisions less likely and results in $\sim 36\%$ efficiency.

These are all very much vulnerable to hidden station problems, not exposed stations though because we just send regardless :).

Busy Tone Protocols

Collision avoidance by informing nodes of ongoing transmission (busy tone). This can be on a different frequency or the same. In **Busy Tone Multiple Access** (BTMA) each station receiving a transmission sends busy tone (even the ones not addressed by the transmission). In **Receiver initiated BTMA** (RI-BTMA) only receiver sends busy tone.

Wireless Collision Detect (WCD)

In **Wireless Collision Detect** (WCD) we further improve the busy tone with another tone type, to remove the problem of collisions during the address decoding on RI-BTMA (nobody sends a busy tone if the receiver does not yet know he is the receiver).

1. Start with regular BTMA
2. Once the receiver knows he is addressed, send a “feedback” tone, so that other stations stop their busy tone.

Multiple Access with Collision Avoidance (MACA)

MACA is somewhat in the middle of schedule and contention. We use signaling packets (also either in-band or out-of-band) for collision avoidance:

- `RTS` (request to send), which is sent before starting the actual transmission
- `CTS` (clear to send), where the receiver grants right to send.

  As shown in Figure 12 this allows both stations near the sender and the receiver to have knowledge of the soon to start transmission, because either `RTS` or `CTS` (or both) is received and stored in the *Network Allocation Vector*. Thus MACA does not have the problem of hidden stations, and exposed stations can potentially be detected (though it is not a good idea to send)

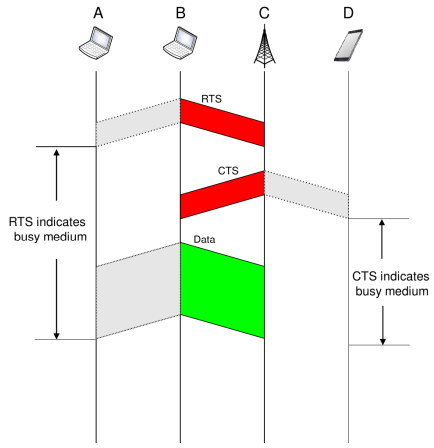

Figure 12: MACA

MACA is an optional mechanism within the popular CSMA/CA MAC.

Well there are some problem though: idle listening is necessary which is bad for low-powered devices. However, we can use “Beacon” frames on a fixed regular interval to wake up in synchronized time frames like Wi-Fi does it.

4 Wireless LAN (Wi-Fi)

4.1 Standards & Structure

Wi-Fi is defined by the IEEE 802.11 standard with each generation (study group) defined by letter at the end Later on the Wi-Fi Alliance switched to simpler naming convention:

- 802.11n (Wi-Fi 4)
- 802.11ac (Wi-Fi 5)
- 802.11ax (Wi-Fi 6)
- 802.11be (Wi-Fi 7)

Wi-Fi runs on layer 1 and 2 of the OSI model, so the physical layer (channel selection, modulation, error coding) and mac layer (access mechanisms, fragmentation, encryption, authentication, …).

General Structure

There are 3 key types of Wi-Fi deployments

1. Infrastructure Network: Access Points are attached to existing network, each AP manages communication in its reception range. Most probably what you are using right now.

2. Ad-hoc Network: stations can build up an own LAN without APs. Example Camera Connect of Sony, Canon, etc…. Or wireless sensor networks which will become important later on.

3. Mesh Networks: Where APs are connected with WLAN, not cables

4.2 Technology

Wi-Fi has obviously changed drastically since the old days. We nonetheless will focus on at least the og-802.11 and 802.11b, 802.11a due to the topic selection in prior exams and exercises. We will then summarize the changes made since 1997 □.

Original 802.11 standard

The og-802.11 standard has two 2.4 GHz band and one unknown infrared band for ad-hoc only. It can use either use *FHSS* (Section 2.2.5) . Which can use 79 different channels with 1 MHz bandwidth each and a minimum of 2.5 hops/s. We either encode in 2-GFSK (1 MBit/s) or 4-GFSK (2 MBit/s).

or *DSSS* variant (Section 2.2.4) where we encode in either PSK (1 MBit/s) or QPSK (2 MBit/s) and use pre-defined chipping sequences.

802.11b

This is the first common version in Europe and only uses *DSSS* with a adaptive modulation based on the *SNR* allowing up to $1, 2, 5.5, 11$ MBit/s. For 5.5 and 11 MBit/s we use Complementary Code Keying which uses a sequence of codes codes instead of one code for all transmission like the barker ones below that. Because the channels in the available 2.4 GHz range are so tight, we must also limit the power on the edges of our spectrum to reduce interference with neighboring channels. This allows for more overlapping channels.

802.11a

This is the first common version in the USA and the next evolution, somewhat parallel to 802.11b, which only enables the usage of the higher-frequency bands (5GHz) for more channels and higher data rates. We combine this with *OFDM* (Section 2.2.6) and possibly up to 64-QAM allowing up to 54 MBit/s. We have 52 subcarriers without guard spaces but 4 of these are phase references transmitting a known signal and allow for as the name suggests phase calibration but also Noise measurements.

Since 1999

802.11g: builds up on 802.11b, but replaces DSSS/CCK with OFDM. 802.11n: allows up to 600 MBit/s with compatibility to 802.11a/b/g where we use 2.4 and 5GHz and guard spaces and the introduction of MIMO with 4 possible simultaneously streams 802.11ac: only uses the 5GHz band with 7 GBit/s and 8 MIMO streams, 256-QAM 802.11ax: similarly allowed more MIMO stuff more QAM **and** OFDMA where the AP can assign stations subbands 802.11be: more QAM, more MIMO, more OFDMA 802.11bn: More QAM, More MIMO, More bandwidth… (estimated 2026)

4.3 Medium Access

Wi-Fi has a prioritized MAC layer where different information is prioritized via different waiting time intervals. See Figure 13.

- `SIFS` (Short Inter-Frame Space) $10\mu s$
- `DIFS` $50\mu s$

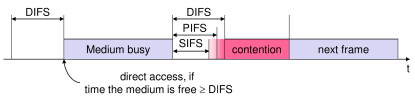

Figure 13: IFS in Wi-Fi

The first important rule is: Only send after carrier sense (CS) and sent out `ACK` on successful transmission (prioritized by `SIFS`). Imagine we want to send a packet then there are a couple different cases:

1. We have sent prior to this, now we have to wait a `DIFS` and select a random backoff. If we don't hear anything during the entire `DIFS` and backoff, we are able to send. If not goto 5. This is necessary for a fair scheme where everybody is able to send sometime.
2. Somebody is sending currently. Now you wait until you receive the `ACK` of that transmission, indicating that it is completed, and **then** wait a `DIFS` and a random backoff (if you already have one use that one). Again if you heard nothing send, else goto 5.
3. Nobody sends currently. Wait `DIFS` and send immediately if no conflicts.
4. We have sent prior to this but no `ACK` is received. Assume a collision and goto 1. for a retransmission.
5. If we hear someone during our waiting period, we save the time left to wait (During the `DIFS` there is no backoff yet) which will be used in the next waiting period. This makes a somewhat fair MAC scheme.

This whole thing can be seen in Figure 14.

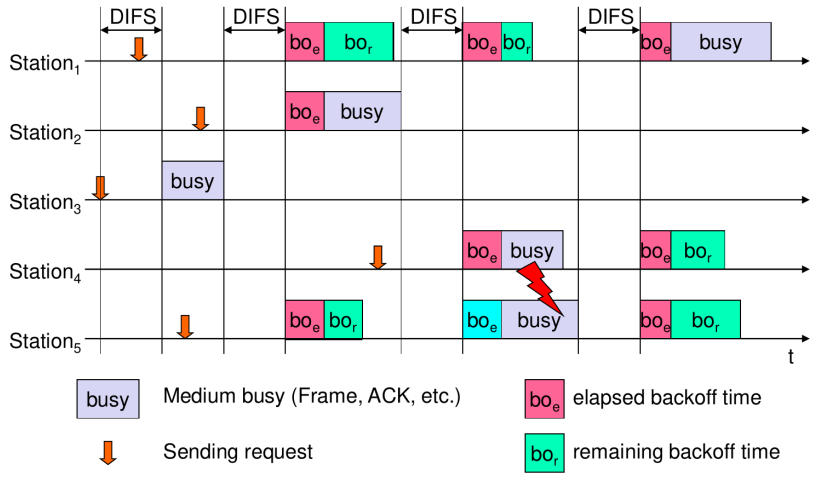

Figure 14: Example Transmission 802.11b

This concludes the CSMA/CA.

The optional part of which is the `RTS`, `CTS` (each prioritzed with `SIFS` instead of `DIFS`) implementation described in Figure 12.

4.4 Frames

Each Wi-Fi generation has different parameters and therefore headers they want, to combine all of these there is the **Physical Layer Convergence Protocol** (PLCP) which allows the PHY layer to decouple from the signalling and data layer. There are 4 MAC address fields (sender, receiver), 2 of which are optional for the different kinds of Wi-Fi deployment methods (Section 4.1.1).

Then there is physical and mac layers with the following headers:

`ACK`, `RTS`, `CTS` in order

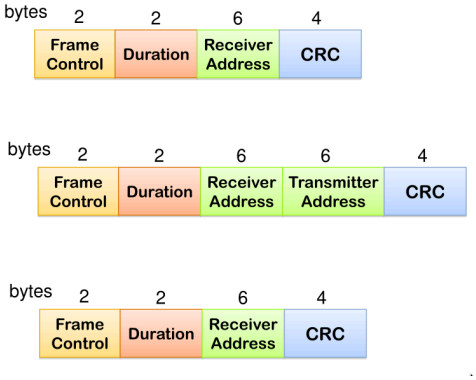

Figure 15: ACK,RTS,CTS frames

And the original Wi-Fi PHY frames

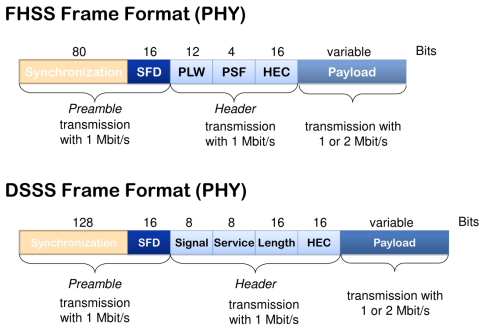

Figure 16: 802.11 PHY frames

The synchronization and SFD are known constant sequences and the header and `ACK`,`RTS`,`CTS` are **always** transmitted with 1 MBit/s to allow even the worst devices to understand the signal parameters

4.5 Management

Things that need to be managed are: synchronization, power management, security, ..

Synchronization

We can decide to use Beacon frames, send out by the APs in a fixed interval. This allows the AP to remember which stations have messages and buffer them until the next beacon, where we notify the device that there is stuff to get (*Traffic Indication Map*). The devices can then sleep if nothing is there. Beacon frames also contain information on roaming and important stuff like the `SSID` a time synchronization mechanism and supported codecs, rates and security protocols.

Roaming and Security

Security is not important right now. Roaming however allows something like eduroam (bad role model), where we connect to one AP in a network and can move freely to another without loss of connections. We have to have mobility database to know which AP is the correct one and a centralized DHCP server, so that we keep our IP. Then if we connect to a new AP it sends out a fake message with our MAC address to update the switches switching table from the old to the new AP.

5 Multi-Hop Networks (Routing)

Now we assume a Ad-Hoc Wi-Fi community most probably a wireless sensor network. The problem is routing between these nodes. Traditional routing protocols are not designed for mobility and frequent updates and thus take very long to converge (Distance Vectorrrr) or have way too much overhead for low powered sensors (Link State). Both also don't account for high packet loss and low computation capabilities of this environment.

There are two types of Multi-Hop Networks we are interested in **Mobile Ad-Hoc Network** (MANET) where devices connect directly without the need for an AP, allowing devices to act as a router to extend the network range. The structure of such a network can vary because of

mobility. And **Wireless Mesh Network** (WMN) where there are routers and APs, but they are connected wireless.

5.1 Routing metrics for multi-hop networks

First we introduce new *metrics* to measure the usefulness of a path. Metrics from wired networks, like hop count, are often not suitable here.

Expected Transmission Count (ETX)

The expected number of transmissions required to successfully transmit a packet over a link.

$$\text{ETX} = \frac{1}{\text{FDR} \cdot \text{RDR}}$$

- FDR: Formward Delivery Ratio, estimated probability of a packet to successfully reach the destination
- RDR: Reverse Delivery Ratio, same for other direction

ETX of a path: sum of all link ETXs.

Expected Transmission Time (ETT)

Considers differences in packet sizes and data rates in different links.

$$\text{ETT} = \text{ETX} \cdot \frac{S}{B}$$

- S: average packet size
- B: bandwidth of the link

ETT of a path: sum of all link ETTs.

Weighted Cumulataive ETT (WCETT)

Consider ETT as well as interference of used links.

Now to routing we classify into these categories:

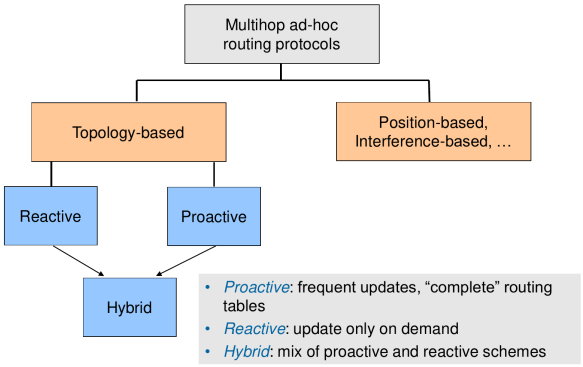

Figure 17: Routing Algorithm Categories

5.2 Reactive Protocols

In reactive protocols there is not active exchange of link information or changes. The entire route is found during a demanded request.

The first option is the **Dynamic Source Routing** (DSR) a path is searches for using flooding (`RREQ` messages with a request to the destination). Each station that receives a `RREQ` For the first time appends its own address and broadcasts it after a random delay (to avoid collisions) to all others (in MIT also to the original sender, in AIT this does not happen). If the station is the destination, it returns a `RREP` with the same path discovery. Though it is not necessary to be the same path for both directions. DSR has the benefit of extremely low overhead and no updates but of course there is high delay for sending a packet to the target. As well as high strain on the network due to the flooding, which is not energy efficient.

The other reactive option is the **Ad-hoc On-demand Distance Vector routing** (AODV). Route discovery is very similar to DSR with the notable change of `RREQ` building up reverse path caching of each station pointing towards source. This means the destination can reply without needing second path discovery (AODV assumes symmetric links) and we must periodically check for link availability. To avoid loops we also integrate sequence numbers into the `RREQ`.

5.3 Proactive Protocols

As opposed to reactive ones, *proactive* protocols exchange link information before a demand of a link.

In **Optimized Link State Routing** (OLSR) instead of broadcasting knowledge about local link costs to all neighbors, only sent them to a reduced number of nodes using *multipoint relays*. Multipoint relays are chosen as such that **every two-hop neighbor** of a node X is a **one-hop neighbor of at least one** multipoint relay of X. Each node periodically transmits its neighbor list, so that every node knows their 2-hop neighbors.

You can guess, it is more overhead and routing traffic but less latency for established routes.

In **Destination Sequenced Distance Vector** (DSDV) each host manages a table containing the distances (in terms of a metric) to other hosts and sequence numbers (the version of the table). On changes these routing table are (incrementally) exchanged if the sequence number is higher than ours. This reduces the traffic caused by link information gossip and guarantees that updates happen in the correct order and avoids loops.

⚠Warning: AIT content (not MIT exam relevant!)

In AIT these are all covered as well + one more, logically it makes sense to include it and let AIT panikzettel forward here

Beacon Vector Rounting (BVR) we select a few nodes as *beacons* and let every node (periodic beacon flooding) know its distance to each beacon. This distance vector $< d_1, d_2, d_3 >$ for beacon 1,2,3 then serves as coordinates. These coordinates together with truly unique id is then used as the address. We have 3 modes of routing, each being prioritized over the next one:
1. Greedy: Distance of node p to destination d is a tuple

$$(\delta_k^+(p,d), \delta_k^-(p,d))$$

$$\delta_k^+(p,d) = \sum_{\{i \in C_k(d)\}} \max(p_i - d_i, 0) \text{ beacons *closer* to } d \text{ than } p$$

$$\delta_k^-(p,d) = \sum_{\{i \in C_k(d)\}} \max(d_i - p_i, 0) \text{ beacons *further away* from } d \text{ than } p$$

$$C_k(d) \text{ being the } k \text{ closest beacons to } d$$

Then to decide on the next hop we select the smallest distance, where $\delta_k^+(p,d)$ has strict precedence over $\delta_k^-(p,d)$

2. Fallback Mode. Sometimes the greedy forwarding fails to improve the distance, then we just forward to the closest beacon of the destination
3. If even at the closest beacon there is no improvement, we resort to flooding scoped to the number of hops needed (known through beacon vector).

6 From GSM to 5G

Nextup telecommunication services. We will not delve deep into the technical descriptions of the physical technologies but more into the backbone structure of the network. Again we will concentrate on old GSM and only briefly discuss 3G,LTE and 5G

The 1st generation was analog, cellular mobile systems in different countries that were not compatible to each other.

2nd generation (2G/GSM)

The 2nd generation transitioned to digital and was first specified and almost universally adopted with GSM900 (Global System for Mobile communication) (frequency band around 900MHz). Though it still uses circuit switching instead of just packet switching.

What was required?

- Total mobility and support for both voice and data services.
- Worldwide Connectivity with only one number (Roaming)
- High Capacity

For the capacity we use a kind of space multiplexing with hexagonal cells which size are not uniform but dependent on environment and expected traffic amount (i.e. more smaller cells in cities). See Figure 18

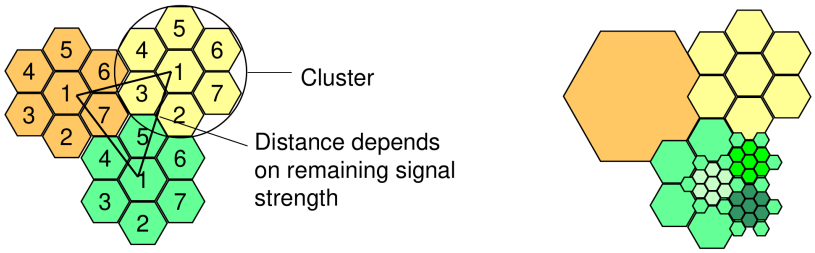

Figure 18: GSM Cell Example

A cluster is a set of cells in which the whole available bandwidth is distributed. This means more cells per cluster: less channels per cell, lower system capacity, but less co-channel interference.

Less cells per cluster: more channels per cell, higher system capacity, but more co-channel interference

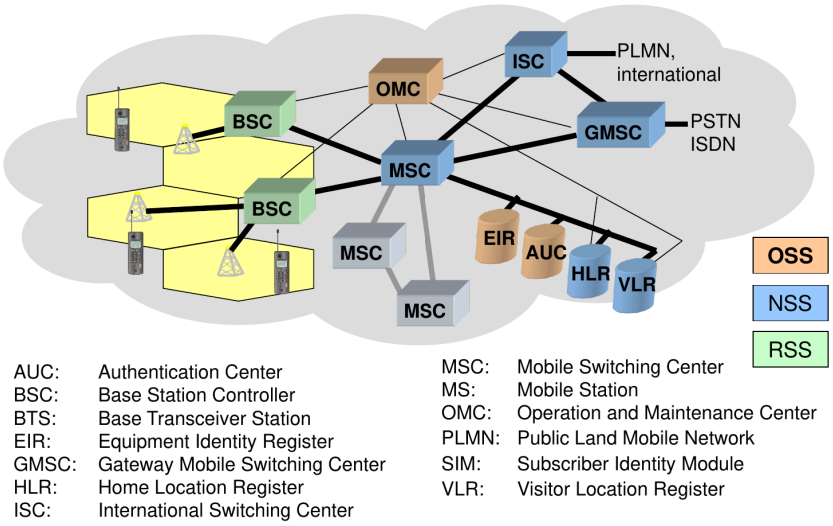

Figure 19: GSM System Overview

Figure 19 shows the a "rough" (sadly) overview of the system. We will go over the most important parts:

1. The mobile stations/nodes (MS) are the phones (not that smart) in the picture
2. Connect to Base Transceiver Station (BTS) connected Base Station Controller (BSC) these handle the communication in one or few cells. If a higher component wants to talk to the phone, it will be forwarded to the particular *BSC*, this will then determine the exact position and *BTS*, or if unknown initiate a paging (search) request on all BTS of that *BSC*. $BSS = BSC +$ all BTS
3. The Mobile Switching Center (MSC) connects different regions together. This is connected to both the *VLR* and *HLR*
4. The Visitor Location Register (VLR) is scoped to a *MSC* and knows the *BSS* of **all currently** connected devices in the network of this *MSC*, even visitors as the name suggests. This gets updated frequently and propagates it to *HLR*.
5. The Home Location Register (HLR) is scoped to the entire network and stores customer data of this provider and the last connected *MSC* for a faster search, though this is not updated immediately to avoid overload.
6. Gateway MSC (GMSC) handles connections to other providers and the wired telecommunication systems forwards to the correct *MSC* via the information of the *HLR*.

The handover process is best described by Figure 20 and is initiated by the BSC.

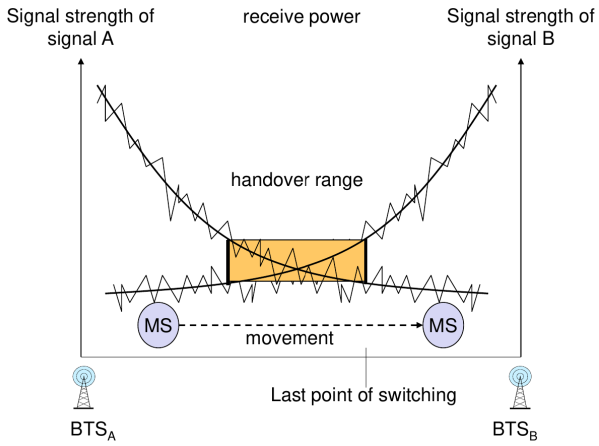

Figure 20: Handover Process

2.5G

GSM only allowed 9.6 kBit/s, thus until 3G you probably used the 2.5 extension of 2G one being EDGE.

HSCSD allows for several channels but is still connection oriented. **GPRS** was the first packet-oriented transmission which needed extra GSN support nodes. The problem was also that you cannot be sure that in an handover the new BTS has the same capacity and data rate as the old one. **EDGE** was just an improvement over GPRS with 8-PSK instead of GMSK and again new hardware but this time not in the core network.

Since 2003

UMTS still has compatibility to GSM and uses a whole new backbone system called UTRA with a simple generation of orthogonal chipping codes and scrambling codes. After release 4 and 5 almost all core networks make use of IP

In LTE the network is optimized for TCP performance and VoIP. We implement a QoS differentiation and achieve higher data rates with new radio subsystems. This comes with OFDMA and more.

5G uses OFDMA in down and upling and considers new territories, Internet of Things and smart Industries in 3 different scenarios it also has much more bandwidth now and is defined from 0.4-100GHz □:

1. Enhanced Mobile Broadband: The improvement over LTE
2. Massive Machine Type Communication: Integrate IoT, smart cities and all those buzz words
3. Ultra-reliable and Low Latency Communications: IIot, smart cars, e-health,… bzzz bzz

We have also the concept of Network/RAN Slicing, which allows to create multiple, independent virtual networks on a single physical infrastructure, where each slice is dynamically customized with the needed resource and performance requirements for example using Software Defined Networks and Network Function Virtualization. For more on this topic look for the AIT lecture and Panikzettel.

7 IP and Mobility

Last chapter! Don't give up.

The core problem of IP and mobility is the double role of the ip address as both an **identifier** and **locator**. This moving to a different subnet changes our IP address and DNS is way too slow and inconvenient to save us this time. This is only the most problematic one of several mobility variants.

1. Intra-cell mobility, handled by PHY layer
2. Inter-cell but intra-network mobility (micro mobility)
   - requires routing updates, IP address can be kept
   - handled within network backbone
3. inter-network mobility (macro mobility), IP address change required

7.1 Mobile IP (MIP)

Mobile IP is a system that tried to solve this problem, while failing terribly.

The **Mobile Node** (MN) keeps its original home ip address and a new **Care-of-Address** (CoA). At home we have a **Home Agent** (HA, could be the router itself) which manages the location of the MN, its CoA and tunnels IP packets to this CoA that are received on the home-address. The **Foreign Agent** (FA, the "exit" router of the foreign network the MN is in) forwards then this CoA to the MN. The **Correspondent Node** (CN, our communication partner) then just uses our home address without knowing the CoA. For the tunneling we use the IP-in-IP-encapsulation of and IPSec VPN. See Figure 21.

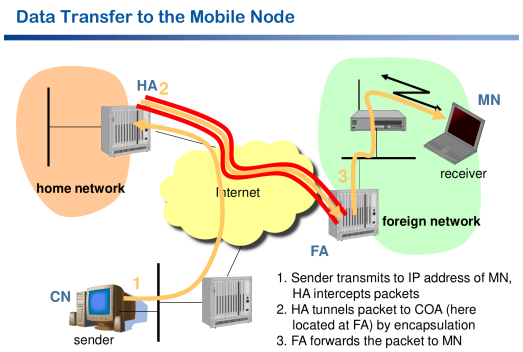

Figure 21: From CN to MN Communication

There are a couple of options to consider.

1. After a first contact the MN could send to the CN, to use the CoA instead of the home-address removing the triangle routing problem (what a sentence).
2. The foreign agent could either unpack the IP-in-IP encapsulation and send the MN only the original home-address packet or it directly send the IP-in-IP packet to the MN (called **co-located** CoA)

Problems

- Triangular routing can cause high latencies for high geographic distances
  - can be improved by informing CN about location of MN (route optimization)
- Firewalls might discard MIP packets
- TTL not consistent
- Privacy, if route optimization is used, CN can learn MNs movements
- And finally NAT breaks everything, because the CoA could be a 192.168.178.xxx private address

Some of these problems may be fixable with reverse tunneling of the home agent.

IPv6 MIP

IPv6 makes MIP a bit easier since security is directly integrated and FA are not needed anymore and CoA are always co-located resulting. We also have automatic path optimization (informing CN of CoA) with IPv6 extension headers and better handover abilities.

7.2 Host Identity Protocol (HIP)

HIP completely replaces the role of IP as an identifier and runs in between IP and the transport layer. This ensures the identifier stays the same during mobility.

For this the *Host Identity* (HI) is provided by public and private key pair which allows for authentication and IPSec is used for encryption. A *Host Identity Tag* (HIT) or *Local Scope Identity* (LSI) is a hash of this host identity that is only 128 or 32 Bit long respectively and is used for sending. The application then sees the HIT and converts it to the correct IP address for sending. To prevent L1 flooding DoS attacks we require a small cryptographic puzzle which the MN needs to solve for the Remote Host to create the IPSec tunnel.

However like all these things HIP can be affected by legacy middleboxes and we avoid this by encapsulating HIP in UDP

7.3 Peer-to-Peer-based Wi-Fi Internet Sharing Architecture (PISA)

Wi-Fi sharing means that we share our AP with others and get access to their AP thus building a community. But well do you want to trust just any AP? And do the AP want to trust that not mobile node is misbehaving in **your** network?

Again we use a indirection and HIP for an encrypted tunnel to our trust point and block all other traffic to the Community AP. This works securely and privatly. We also need to proof that our mobile node is actually part of our Home AP, which the community attests.

7.4 Multipath TCP (MPTCP) & QUIC

We would like to have one TCP connection over multiple paths, for example over you phone's Wi-Fi and simultaneously over 5G, that when you disconnect you immediately switch to 5G without reconnection. But again Middleboxes often drop unknown TCP options, like those used by MPTCP.

Have a look at the CSE lecture and or Panikzettel for details into this. The same holds for QUIC which is replacing TCP slowly and learns a lot from the last 20 years in communication systems engineering.