Computer Networking and IT Security (CNS)

INHN0012 – WiSe 2025/26

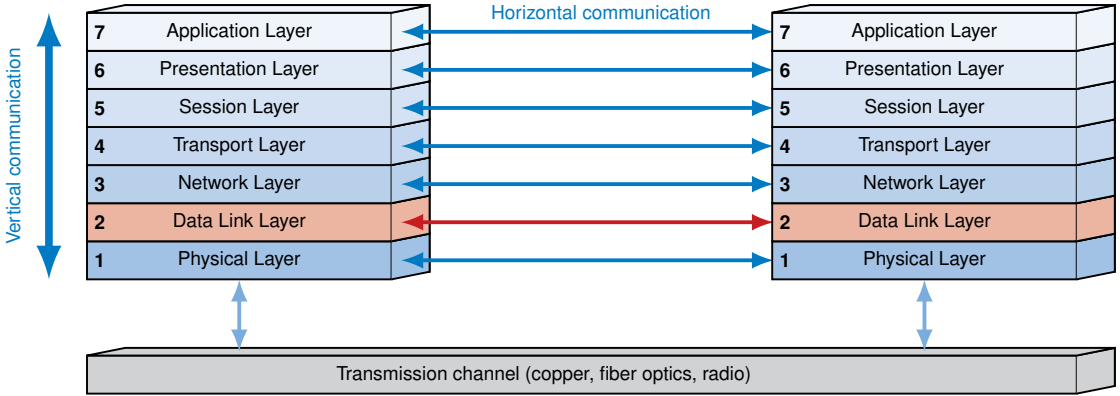
Lorenz Lehle, Stephan Günther

Chair of Distributed Systems and Security
School of Computation, Information and Technology
Technical University of Munich

Chapter 2: Data link layer

Chapter 2: Data link layer

Placement in the ISO/OSI model



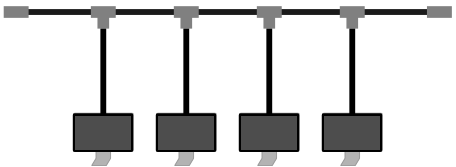
Chapter 2: Data link layer

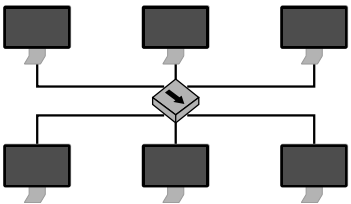
We first deal with local area networks (LANs), i. e.

- all attached nodes can be reached directly and
- are accessible by means of (mostly) unstructured addresses on layer 2,
- there is no routing,
- relaying messages based on layer 2 addresses (“bridging” and “switching”) is, however, possible.

Examples:

- Individual local networks (connections via bus / hub, but also by means of switches)





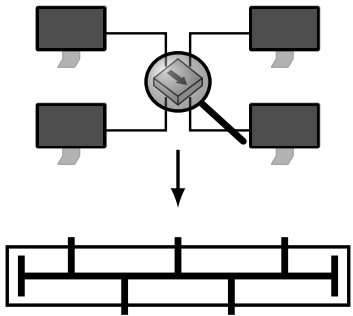
- Connection between a base station and a cellular phone
- Bus systems within a computer, e. g. USB, PCI, etc.

The essential tasks of the Data-Link Layer are

- media access control,
- error detection of transmitted frames, and
- addressing within the directly connected network.

Media access control:

- Hubs create a star topology only at a first glance
- All attached computer are internally connected to a single bus
- Concurrent transmissions result in collisions and loss of frames

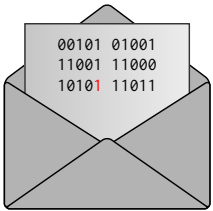


The essential tasks of the Data-Link Layer are

- media access control,
- error detection of transmitted frames, and
- addressing within the directly connected network.

Error detection:

- Transmission errors occur despite channel coding
- These must be detected
- Defective messages must not be forwarded to higher layers
- Retransmission of defective or missing frames is sometimes left to higher layers

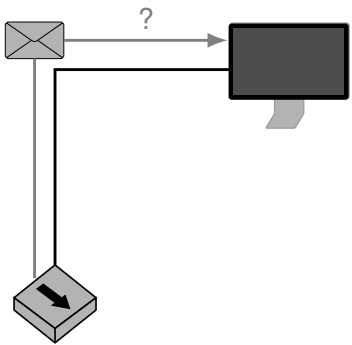


The essential tasks of the Data-Link Layer are

- media access control,
- error detection of transmitted frames, and
- addressing within the directly connected network.

Addressing within local area networks:

- A frame may be received by multiple nodes, e. g. in case of a bus or wireless network
- Receivers must decide to whom a frame was addressed



Chapter 2: Data link layer

Representation of networks as graphs

- Directed graphs
- Undirected graphs
- Paths in networks
- Network topologies
- Adjacency and distance matrix
- Creating tree structures

Characterizing connections, multiple access, and access control

Framing, addressing, and error detection

Connecting nodes on Layers 1 and 2

Security Considerations

Summary

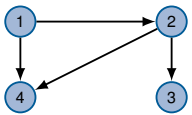
Motivation

- Directed or undirected graphs are commonly used to represent network topologies and node connections.
- In the following, we introduce the corresponding notation and basic terms.

An asymmetric network can be represented as directed graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ with

- $\mathcal{N}$ denoting the set of nodes (Nodes or vertices) and
- $\mathcal{A} = \{i,j\} \mid i,j \in \mathcal{N} \wedge i, j$ are connected by a directed arc$\}$ denoting the set of directed arcs.

Example: $\mathcal{N} = \{1,2,3,4\}$, $\mathcal{A} = \{(1,2),(2,3),(2,4),(1,4)\}$

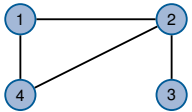


Undirected graphs

A symmetric network can be represented as undirected graph $\mathcal{G} = (\mathcal{N},\mathcal{E})$ with

- $\mathcal{N}$ denoting the set of nodes and
- $\mathcal{E} = \{\{i,j\} \mid i,j \in \mathcal{N} \wedge i, j \text{ are connected undirectedly}\}$ denoting the set of (undirected) edges.

Example: $\mathcal{N} = \{1,2,3,4\}$, $\mathcal{E} = \{\{1,2\},\{2,3\},\{2,4\},\{1,4\}\}$



Note:

Undirected graphs can be understood as directed graphs with symmetric arcs. An edge $\{i,j\}$ of an undirected graph with weight (cost) c_{ij} thus corresponds to the two arcs (i,j) and (j,i) of a directed graph with weights (costs) $c_{ji} = c_{ij}$.

Paths in networks

Paths can be represented in graphs:

- A path between two nodes[1] $s,t \in \mathcal{N}$ is a set

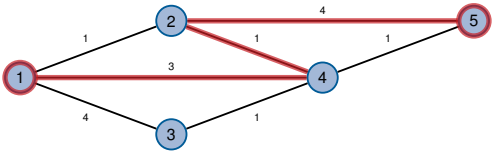
$$\mathcal{P}_{st} = \{(s,i),(i,j), \ldots ,(k,l),(l,t)\}$$

  of edges that connect s and t with each other over a number of intermediate nodes.

- A path's cost is the sum of the costs of the used edges: $c(\mathcal{P}_{st}) = \sum_{(i,j)\in\mathcal{P}_{st}} c_{ij}$.

- A path's length is the number of intermediate nodes: $l(\mathcal{P}_{st}) = |\mathcal{P}_{st}|$.
  On Layer 3 we refer to the path costs as hop count, which is not common on Layer 2.

Example: $\mathcal{P}_{15} = \{(1,4),(4,2),(2,5)\}$



$c(\mathcal{P}_{15}) = 3 + 1 + 4 = 8, \quad l(\mathcal{P}_{15}) = 3$

[1] Source and destination (terminal) are commonly denoted as s and t, respectively.

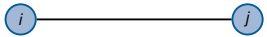
Network topologies

The topology describes the structure how nodes are connected with each other. We differentiate between

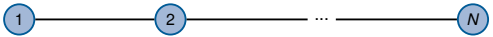
- physical and
- logical topology.

Well-known topologies:

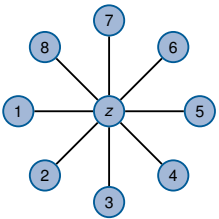
- Point-to-point



- Chain

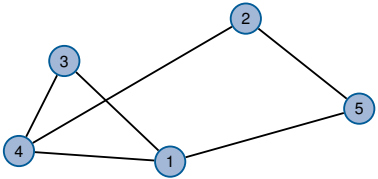


- Star

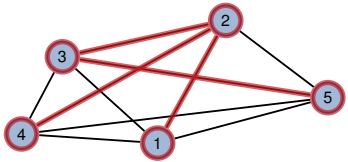


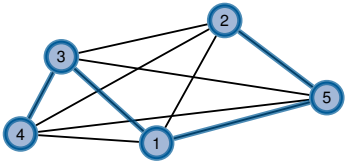
Network topologies

- Mesh

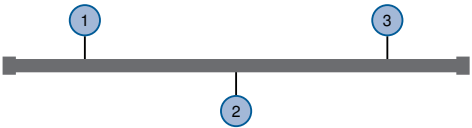


- Tree (mostly a logical topology)





- Bus



Adjacency matrix

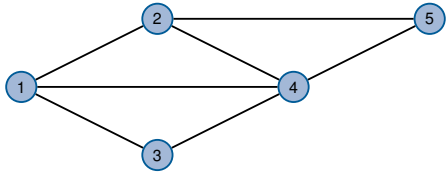
Networks can easily represented as matrices. The adjacency matrix

$$\boldsymbol{A} = (a)_{ij} = \begin{cases} 1 & \exists (i,j) \in \mathcal{A} \\ 0 & \text{otherwise} \end{cases}, \quad \forall i,j \in \mathcal{N}, \qquad \boldsymbol{A} \in \{0,1\}^{N \times N}$$

denotes whether node i is directly connected to node j.

Example:

$$\boldsymbol{A} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$



- The element a_{ij} of $\boldsymbol{A}$ is 1 if there is a connection from node i to node j.
- $\boldsymbol{A}$ is symmetric ($\boldsymbol{A} = \boldsymbol{A}^T$) if for each arch (i,j) there is also an anti-parallel arc (j,i).

Distance matrix

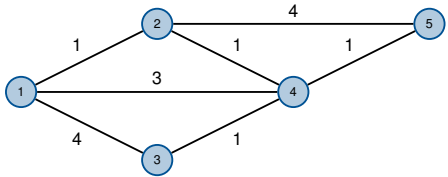
The Distance matrix

$$\boldsymbol{D} = (d)_{ij} = \begin{cases} c_{ij} & \exists (i,j) \in \mathcal{A} \\ 0 & \text{if } i = j \\ \infty & \text{sonst} \end{cases}, \quad \forall i,j \in \mathcal{N}, \qquad \boldsymbol{D} \in \mathbb{R}_{0+}^{N \times N}$$

contains the costs of paths of length 1 between all pairs of nodes.

Example:

$$\boldsymbol{D} = \begin{bmatrix} 0 & 1 & 4 & 3 & \infty \\ 1 & 0 & \infty & 1 & 4 \\ 4 & \infty & 0 & 1 & \infty \\ 3 & 1 & 1 & 0 & 1 \\ \infty & 4 & \infty & 1 & 0 \end{bmatrix}$$



- The element d_{ij} of $\boldsymbol{D}$ denotes the distance between i and j.
- If there is no direct connection between i and j, then we have that $d_{ij} = \infty$.
- $\boldsymbol{D}$ is symmetric, if the network is symmetric, i. e., for each arc (i,j) there is an anti-parallel arc (j,i) with the same costs.

Adjacency and distance matrix

Question: How do we obtain a matrix that denotes the costs of the shortest path between each pair of nodes?

Adjacency and distance matrix

Question: How do we obtain a matrix that denotes the costs of the shortest path between each pair of nodes?

Answer: Calculate powers of $\boldsymbol{D}$ with respect to the min-plus product

$$\boldsymbol{D}^n = \boldsymbol{D}^{n-1} \otimes \boldsymbol{D} \text{ with } d_{ij}^n = \min_{k \in \mathcal{N}} \left\{ d_{ik}^{n-1} + d_{kj} \right\}.$$

- The n-th power $\boldsymbol{D}^n$ contains the costs of the shortest path between each pair of nodes over a distance of at most n hops.
- The power series converges for a finite $n \in \mathbb{N}$, i. e., $\boldsymbol{D}^{n+1} = \boldsymbol{D}^n = \boldsymbol{D}^*$.

Adjacency and distance matrix

Question: How do we obtain a matrix that denotes the costs of the shortest path between each pair of nodes?

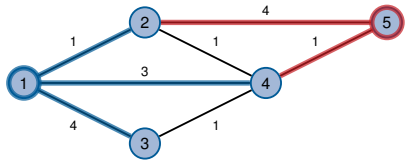
Answer: Calculate powers of $\boldsymbol{D}$ with respect to the min-plus product

$$\boldsymbol{D}^n = \boldsymbol{D}^{n-1} \otimes \boldsymbol{D} \quad \text{with} \quad d_{ij}^n = \min_{k \in \mathcal{N}} \left\{ d_{ik}^{n-1} + d_{kj} \right\}.$$

- The n-th power $\boldsymbol{D}^n$ contains the costs of the shortest path between each pair of nodes over a distance of at most n hops.
- The power series converges for a finite $n \in \mathbb{N}$, i. e., $\boldsymbol{D}^{n+1} = \boldsymbol{D}^n = \boldsymbol{D}^*$.

Example: How we obtain the element (1,5) of $\boldsymbol{D}^2$?

$$\boldsymbol{D} = \begin{bmatrix} 0 & 1 & 4 & 3 & \infty \\ 1 & 0 & \infty & 1 & 4 \\ 4 & \infty & 0 & 1 & \infty \\ 3 & 1 & 1 & 0 & 1 \\ \infty & 4 & \infty & 1 & 0 \end{bmatrix}$$



- Row 1 denotes the costs of the shortest path of length 1 hop from node 1 to each other note.
- Column 5 denotes the costs by which node 5 can be reached by each other node over a path of at most 1 hop.

How often do we have to multiply?

- The power n such that $\boldsymbol{D}^n = \boldsymbol{D}^{n+1} = \boldsymbol{D}^*$ is upper bounded by the longest simple path in the network.
- The longest simple path in turn is bounded above by the number of nodes N.

$$\Rightarrow n < N$$

In our example $n = 3$ is the minimum power such that $\boldsymbol{D}^n = \boldsymbol{D}^{n+1}$ although we have $N = 5$ nodes.

$\boldsymbol{D}^*$ contains the costs of the shortest path between any pair of nodes and therefore solves the all-pair-shortest-distance problem (apsd).

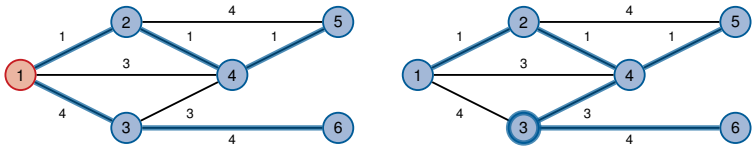
Creating tree structures

A tree is a connected and loop-free graph. In the following, We distinguish between two special types of trees:

- Shortest Path Tree (SPT)
  Connects a root with all other notes at pair-wise minimal costs (rooted tree).
- Minimum Spanning Tree (MST)
  Connects all nodes such that the costs of needed arcs is minimal (unrooted tree).

These trees minimize different metrics are, in general, not the same.

Example:



(a) Shortest Path Tree (SPT) with root node 1

(b) Minimum Spanning Tree (MST)

We will learn about two algorithms to derive SPTs in Chapter 3:

- Bellman-Ford (based on the min-plus product)
- Dijkstra (a greedy algorithm)

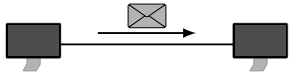
Chapter 2: Data link layer

Characterizing connections

A connection between nodes can be characterized with respect to some fundamental properties:

- data rate
- transmission delay
- direction of transmission
- multiple access (multiplexing)

For the beginning, we consider point-to-point connections:



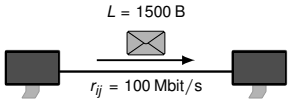
Data rate

Data rate and serialization time

The data rate r given in bit/s determines the time necessary to serialize L bits of data onto the medium. This time, also known as serialization time, is

$$t_s = \frac{L}{r}.$$

Example:



$$t_s = \frac{L}{r} = \frac{1500 \cdot 8\ \text{bit}}{100 \cdot 10^6\ \text{bit/s}} = 120\ \mu\text{s}$$

Question: When does node j receive the first bit of the message?

Propagation delay

We know from Chapter 1 that signals propagate at limited speed – in case of electromagnetic waves in a vacuum at the speed of light.

Propagation delay

The propagation delay over a distance d is determined by the limited propagation speed of signals, which is given relative to the speed of light in a vacuum denoted by $c_0 \approx 300\,000\,\mathrm{km/s}$:

$$t_p = \frac{d}{\nu c_0}.$$

The value $0 < \nu < 1$ is known as relative propagation speed within a medium. For typical copper conductors we may assume $\nu \approx 2/3$.

Example:

- In the examples of the previous slide we had a serialization time of $t_s = 120\,\mu\mathrm{s}$
- Let us assume that nodes i and j are $d_{ij} = 100\,\mathrm{m}$ apart from each other
- At the speed of light, a signal needs 334 ns for that distance
  $\Rightarrow$ j receives the first bit of the message when i is transmitting the 33rd bit.[1]

Question: How long does it take until j has received the whole message?

[1] $100\,\mathrm{Mbit/s} \cdot 334\,\mathrm{ns} \approx 33\,\mathrm{bit}$

Transmission time and network communication diagram

In a network communication diagram, the time sequence when sending and receiving messages can be graphically illustrated.

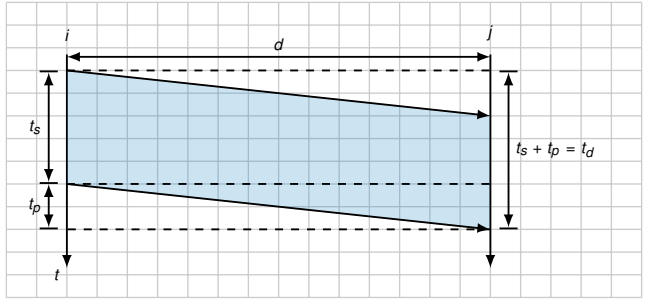


Figure 1: Network communication diagram

- The total delay t_d is therefore given as $t_d = t_s + t_p = \frac{L}{r} + \frac{d}{\nu c_0}$.
- Note that propagation delay t_p and serialization time t_s may differ by multiples orders of magnitude, depending on distance and data rate.

Bandwidth delay product

Due to the limited speed of propagation, a channel has certain “memory capacity”[1] C, which is known as bandwidth delay product.

Bandwidth delay product

The bandwidth delay product denotes the number of bits

$$C = t_p r = \frac{d}{\nu c_0} r$$

that are concurrently in flight in a specific transmitting direction.

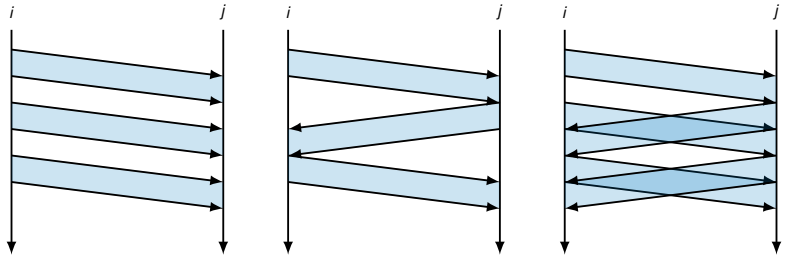
Example:

- Transmit rate $r = 1\,\text{Gbit/s}$
- Distance $d = 10\,\text{m}$
- $\nu = 2/3$ (copper conductor)
- $C = t_p \cdot r = \frac{d}{\nu c_0} \cdot r = \frac{10\,\text{m}}{2/3 \cdot 3 \cdot 10^8\,\text{m/s}} \cdot 10^9\,\text{bit/s} \approx 50\,\text{bit}$

[1] Not to be confused with the channel capacity introduced in Chapter 1.

Transmitting direction

With regard to the direction of transmission, a distinction is made between the following cases:



(a) Simplex (b) Half duplex (c) Full duplex

The type of connection thereby depends on

- the channel's capabilities,
- media access control, and
- requirements of the communication partners.

Multiple access (Multiplexing)

It is often advantageous to transmit messages from different subscribers together over one line:

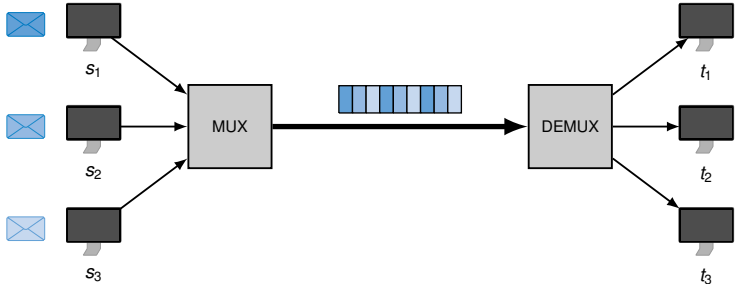


Figure 2: Deterministic time division multiplex

We already heard about another approach for time division multiple access:

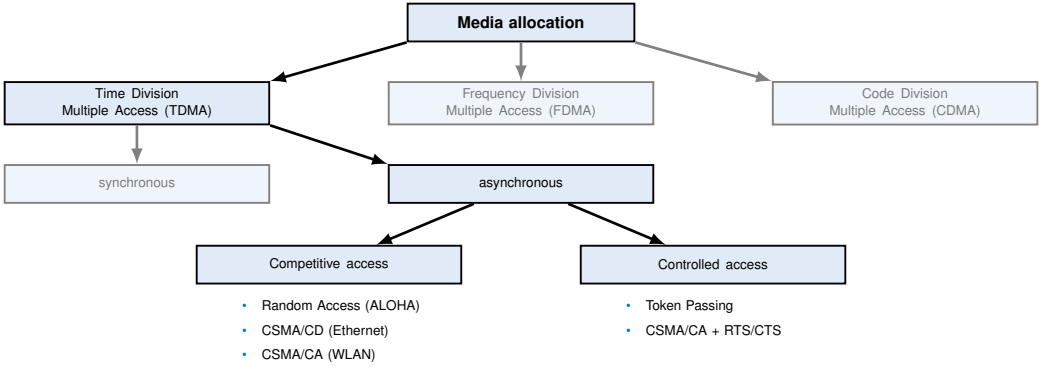
- If we connect multiple computers using a hub,
- a common shared medium (bus) is created, that
- is accessed by computers using non-deterministic media access control.

Overview of different approaches for multiplexing

- Time division multiplex (TDM) (see previous slide)
  - Deterministic approaches, e. g. POTS and ISDN
  - Non-deterministic approaches (competitive media access) in packet based networks such as Ethernet or WLAN
- Frequency division multiplex (FDM)
  Division of the channel into different frequency bands and assignment of different bands to communication partners (see Chapter 1):
  - Omnipresent with wireless transmissions (e. g. different radio stations)
  - Also used with fiber optics (so called “modes” with different colors)
  - Coexistence of ISDN and DSL on the same line
- Space division multiplex, SDM
  Usage of multiple parallel channels:
  - Link aggregation with Ethernet
  - MIMO (multiple-in multiple-out) in wireless transmissions (usage of multiple antennas creates multiple channels with slightly different characteristics on the same frequency)
- Code division multiplex, CDM
  Usage of orthogonal alphabets and assignment of alphabets to communication partners:
  - UMTS in cellular networks
  - CDMA2000 used by the US provider *Verizon* (“CDMA-iPhone”)

Multiple access and media access control

Some (static) multiplexing approaches can also be used for multiple access:



We will learn about these selected four methods in more detail below.

Evaluation criteria for media access control schemes are among others:

- Throughput, i.e. total number of successfully transmitted messages per time unit
- Delay for individual messages
- Fairness between nodes sharing the same medium
- Complexity for transmitter and receiver

Problem with synchronous TDMA

- The channel is statically divided between nodes
- However, data traffic tends to be bursty, i. e., individual nodes transmit data in short, irregular time intervals
- The unused bandwidth cannot be used by other nodes during idle phases

Approach: asynchronous TDMA

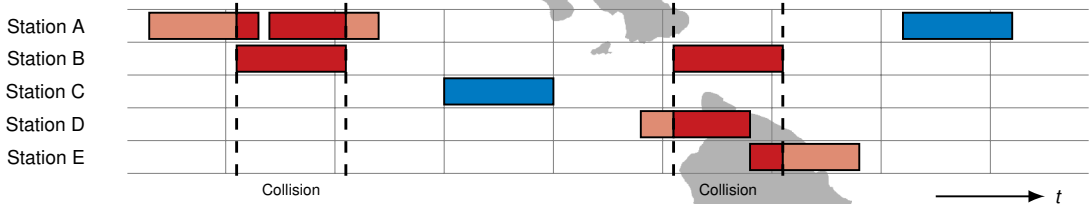
- No static allocation / assignment of time slots
- Instead: random, competitive access or dynamically controlled media access

Random access (ALOHA)

- Developed at the university of Hawaii (1971), cf. Prof. Abramson
- Originally meant for wireless communications to connect Oahu with other Hawaiian islands

Operation

- When a station (node) has data to send, it transmits to a central station (cmp. base stations in WLAN)
- If more than one station transmits concurrently, transmissions collide and are lost
- Successfully transmitted messages are acknowledged by the receiver on another channel (frequency) such that acknowledgements cannot collide with messages (“out-of-band” acknowledgement scheme on the link layer)



The channel model is comparatively simple. There are mathematical descriptions for the so called ALOHA random access channel.

Achievable channel usage with ALOHA

Simplifying assumptions:

- We assume at least $N \geq 15$ nodes (important such that we can use the Poisson distribution as approximation)
- Transmission probability of nodes are uniformly and independently distributed
- Messages have constant size (transmission time T)

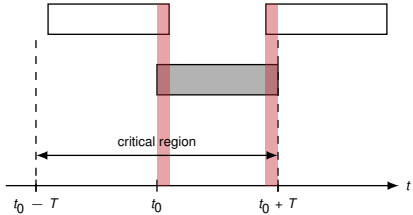
Modelling:

- Whether a certain node i transmits during the time slot $[t, t+T)$ or not corresponds to a Bernoulli experiment with success (transmit) probability p_i.
- Since transmit probabilities are the same for all nodes, we have that $p_i = p \; \forall i = 1, \ldots, N$.
- Since we assume N independent nodes, the same Bernoulli experiment is repeated N times per time slot.
- That is a Binomial distribution counting the number of successes in a series of identical and independent trials.
- For sufficiently large N, the binomial distribution can be approximated using the Poisson distribution[1].
- The average number of messages per time interval is then given as $Np = \lambda$.

The random variable X_t, denoting the number of nodes that transmits within $[t, t+T)$, is Poisson distributed: $\Pr[X_t = k] = \frac{\lambda^k e^{-\lambda}}{k!}$.

[1] Distribution of rare events

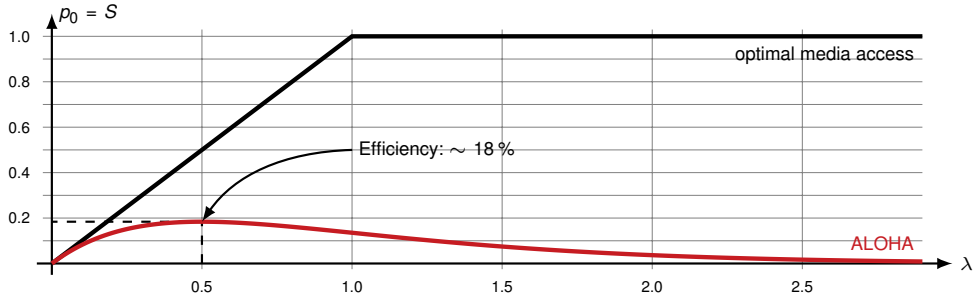
- Assume one station starts transmitting at time t_0.
- A collision occurs exactly when at least one other station also transmits in the interval $(t_0 - T, t_0 + T)$.
- The transmission is therefore successful if during $[t_0, t_0 + T]$ only one station exactly one transmission occurs and in the interval $(t_0 - T, t_0)$ no transmission has started.



With the probability density function $\Pr[X_t = k] = \frac{\lambda^k e^{-\lambda}}{k!}$ we obtain the probability p_0 for a successful transmission:

$$p_0 = \Pr\left[X_{t_0-T} = 0\right] \Pr\left[X_{t_0} = 1\right] = e^{-\lambda} \lambda e^{-\lambda} = \lambda e^{-2\lambda}$$

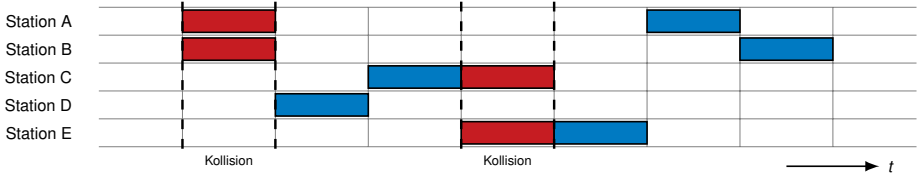
The success probability p_0 can be plotted against the number of messages to be sent per time interval, which is denoted by λ:



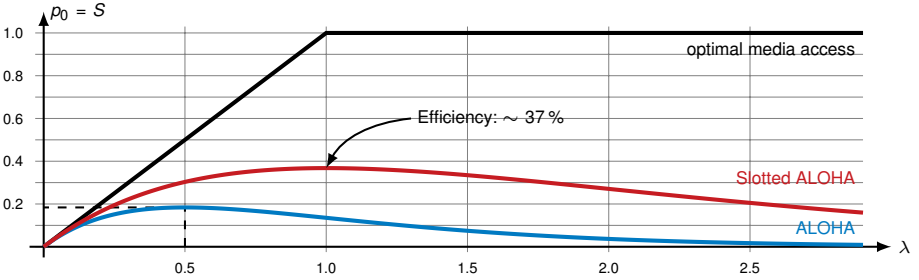
- Per time interval $[t,t + T)$ only a single transmission can be successful
- Therefore, the probability of a successful message also corresponds to the average number S of successfully transmitted messages per time interval.
- With optimal media access, the number of successful messages per time slot would linearly increase with λ and remain constant for $\lambda > 1$ (channel is fully and perfectly utilized).

ALOHA and slotted ALOHA

Variant: With slotted ALOHA nodes must no longer start transmitting at arbitrary points in time but only at multiples of transmission intervals $t = nT$ for $n = 0,1, ...$



The critical time frame is reduced from $2T$ to $T \Rightarrow S = \lambda \cdot e^{-\lambda}$



Carrier sense multiple access (CSMA)

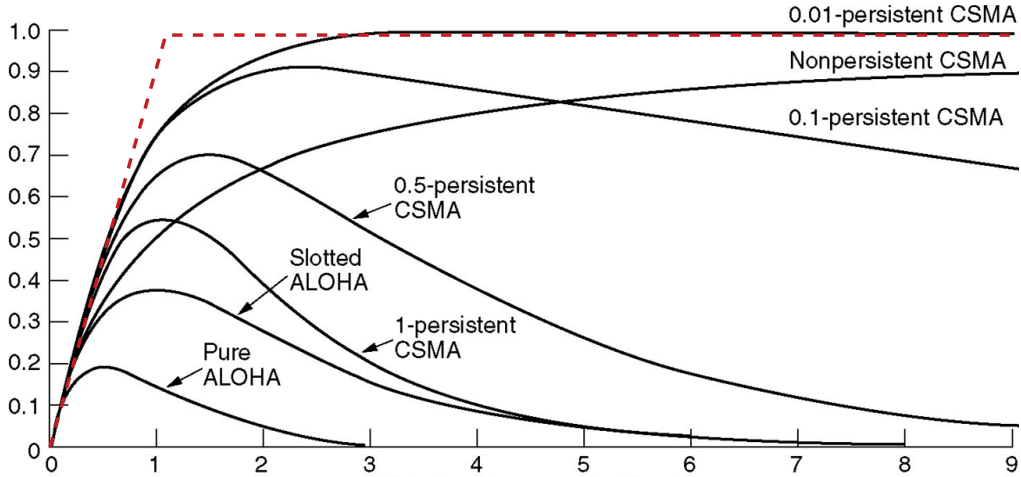
A simple improvement over slotted ALOHA: "listen before talk"

- Listen for ongoing transmissions on the medium
- Start transmitting only when the medium is idle

Different variants:

- 1-persistent CSMA
  1. If medium idle, start transmitting
  2. If medium busy, wait until it becomes free and start in the following time slot
- *p*-persistent CSMA
  1. If medium idle, start transmitting with probability p or delay the transmission with probability $1 - p$ by one time slot – then continue with 1.
  2. If medium busy, wait until it becomes idle – then continue with 1.
- non-persistent CSMA
  1. If medium idle, start transmitting
  2. If medium busy, wait a random amount of slot times – then continue with 1.

Comparison of all approaches discussed so far



CSMA/CD (collision detection)

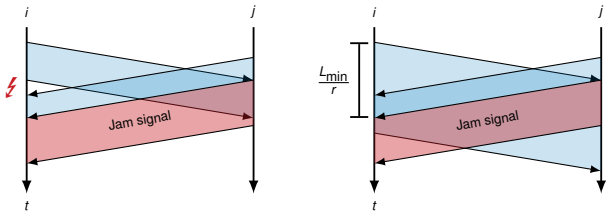
- Detect collisions and repeat the transmission if necessary
- A transmission is assumed to be successful if it is completed without errors, i. e., no collision is detected while transmitting
- Since no acknowledgements are used, the scheme is only suitable if the probability of random transmission errors is sufficiently small

Problem: The transmitter must be able to detect the collision while it is still transmitting.

Prerequisites for CSMA/CD [2]

Assuming that two stations *i* and *j* communicate over a distance *d* using CSMA/CD. In order that collisions can be detected, the messages must be at least of size

$$L_{\text{min}} = \frac{2d}{\nu c_0} r.$$



If we use 1-persistent CSMA with collision detection, the following problem occurs:

- The collision destroys both messages involved.
- At least one of the stations transmits a jam signal.
- When the medium becomes idle again, both stations concurrently start the retransmission.

  $\Rightarrow$ The messages collide again

Solution: Wait a “random” number of time slots if the previous transmission caused a collision

Binary exponential backoff

At the k-th attempt to transmit a specific message

- the transmitter chooses a random number $n \in \left\{0, \ldots, \min\{2^{k-1} - 1, 1023\}\right\}$ and
- delays the retransmission by n time slots.

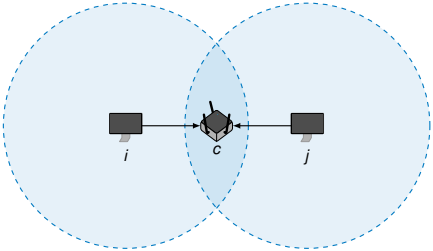
When the medium becomes busy in the mean time, the counter is stopped and resumed when the medium becomes idle again.

By these random delays that increase in case of consecutive collisions (which corresponds to a high load on the channel / large λ), the probability of further collisions is reduced.

CSMA/CA (collision avoidance)

In wireless networks, CSMA/CD cannot be used since the transmitter may be unable to detect a collision in every situation.

„Hidden station“:



- Nodes *i* and *j* transmit at the same time
- Node *c* detects the collision
- But neither *i* nor *j* detect the collision since they are out of range

CSMA/CA is based on *p*-persistent CSMA, i. e.,

1. If the medium is idle, start transmitting with probability p or delay the transmission with probability $1 - p$ for one slot time – then continue with 1.
2. If the medium is busy, wait until it becomes idle – then continue with 1.

Case study: IEEE 802.11 DCF (distributed coordination function)

- Fixed time interval between frames: DIFS (DCF interframe spacing).
- If the medium is idle for at least a DIFS period, uniformely and independently draw a number of backoff slots from the set $\left\{0,1,2,\ldots,\min\left\{2^{c+k-1}-1,255\right\}\right\}$ by which the transmission is deferred.
- c depends on the actual PHY (physical layer implementation), e. g. $c = 4$, k denotes the number of transmission attempts.

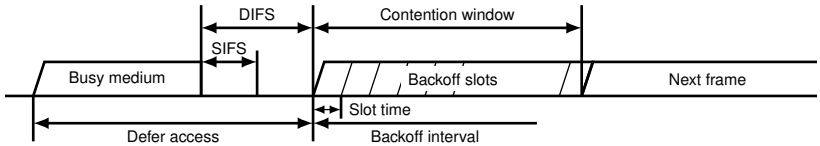


Figure 3: IEEE 802.11 DCF

- Since $c > 0$, we always have a contention window (main difference to Ethernet).
- With IEEE 802.11, a transmission is assumed to be successful in the following cases:
  - for unicasts the receiver transmits an acknowledgement after SIFS (which is shorter than DIFS)
    $\Rightarrow$ no other node may start transmitting after this time
  - for multicasts and broadcasts, the transmission is assumed to be successful if no errors were detected during transmission (in general, a station does not know from whom to expect acknowledgements, and acknowledgements from multiple stations may cause collisions themselves)

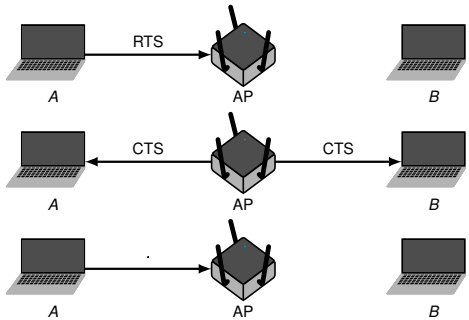
CSMA, CSMA/CD, CSMA/CA

Optional extension: RTS/CTS (request to send / clear to send)

- In infrastructure mode, transmissions are controlled by a base station (access point)
- Before a node may start a transmission, it transmits an RTS to the base station
- Only if the space station answers by CTS, a transmission may begin

Example:

1. *A* transmits a RTS that may in general not be received by *B* due to the distance.
2. The base station answers with a CTS that is received by both *A* and *B*, making *B* aware that a transmission is about to start
3. *A* may start transmitting while *B* has to wait a defined time period before it is even allowed to transmit a RTS.



Optional extension: RTS/CTS (request to send / clear to send)

Advantage:

- Collisions and the effect of hidden stations are reduced but not fully prevented.

Disadvantages:

- There may still be collisions, e. g. if *B* does not overhear the CTS.
- RTS/CTS takes some additional time before a transmission may begin, thus reducing the achievable data rate under optimal conditions.

Notes:

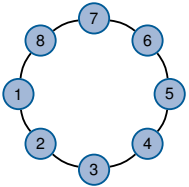
- RTS/CTS is part of the so called virtual carrier sensing since the medium is reserved for a specific duration.
- To reduce the loss probability for RTS/CTS messages, these are transmitted at the most robust modulation and coding scheme, i. e., the lowest supported data rate. This does not hurt much since these messages are very small.
- All devices receiving a CTS should consider the medium busy for the respective time frame, independent on whether or not they belong to the same service set[2]

[2] The term “service set” denotes a group of IEEE 802.11 devices communicating with each other – informally speaking: being associated with the same AP.

Token Passing

Idea: collision-free operation by forwarding a token in a ring

- Stations are connected to a logical ring.
- A token circulates through the ring by passing it on from node to node.
- If a stations wants to transmit a message, it waits until it obtains the token.
- After transmitting a message, the token ist passed on again.



Receiving messages:

- A message is passed through the ring just like the token.
- The receiver marks the message as read and passes it on.
- When the message arrives again at the transmitter, it removes the message and instead passes on the token.

What happens if the token is „lost“?

- There is a monitor station that is somehow elected, e. g. the station with the (numerically interpreted) largest address.
- That monitor station creates a new token when needed and removes endless circulating message or duplicate tokens.
- If the monitor station is removed, the remaining stations elect a new monitor station.

Advantages:

- No retransmissions due to collisions
- There is a guaranteed maximum delay until a node can start transmitting (determinism)

Disadvantages and problems:

- If the token is lost, it must be replaced
  $\rightarrow$ one station needs to take over responsibility (monitor station).
- Defective or selfish behavior of a node may impact the whole network.
- The transmission delay may be larger compared to CSMA since a transmitter has to wait for the token.
- Creating the ring topology can be complex.

Usage nowaday:

- Token Ring (IEEE 802.5) has been completely replaced by Ethernet (IEEE 802.3).
- FDDI (Fiber Distributed Data Interface) is a collective term for fiber optical rings up to a distance of several hundred km. The are, for instance, used as backbone by local service providers in metropolitan scales.

Summary

In this subchapter we learned about different dynamic approaches for time division multiple access.

In contrast to static time division multiplex the available channel bandwidth is not exclusively reserved for inactive nodes.

Competitive access:

- ALOHA and Slotted ALOHA
- CSMA (non-persistent, 1-persistent, *p*-persistent)
- CSMA/CD (collision detection)
  IEEE 802.3 Ethernet
- CSMA/CA (collision avoidance)
  IEEE 802.11 WLAN

Controlled access:

- CSMA/CA with RTS/CTS
  IEEE 802.11 WLAN
- Token Passing (collision prevention)
  IEEE 802.5 Token Ring, Fiber Distributed Data Interface (FDDI)

Chapter 2: Data link layer

Motivation

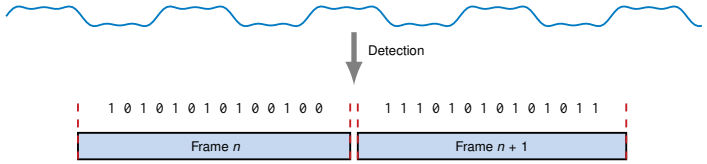
So far we have only talked about messages without thinking about their format. From the point of view of the physical layer, a message is just a sequence of bits. However, this conception is no longer sufficient for a consideration of the data link layer.

In the following, we will consider

- how individual messages can be kept apart,
- what additional information link layer protocols require, and
- how transmission errors that occur despite channel coding can be detected.

In the context of the data link layer we refer to messages as frames.

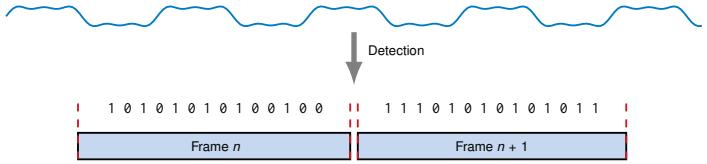
Frame boundary detection and code transparency



How can the receiver detect frames, especially if

- frames have different sizes and
- there is not constant user data on the line (idle periods)?

Frame boundary detection and code transparency



How can the receiver detect frames, especially if

- frames have different sizes and
- there is not constant user data on the line (idle periods)?

There are many approaches:

- Length specification of payload
- Control characters (start / end)
- Boundary fields and “bit stuffing”
- Code rule violation

The goal of all these procedures is to maintain code transparency, i. e., to allow the transmission of arbitrary strings.

Length specification of payload

- At the beginning of the frame is the length of the following payload (or the total length of the frame).
- Prerequisite: the length field and thus the beginning of a message must be clearly recognizable

How can the beginning of a frame be detected?

Length specification of payload

- At the beginning of the frame is the length of the following payload (or the total length of the frame).
- Prerequisite: the length field and thus the beginning of a message must be clearly recognizable

How can the beginning of a frame be detected?

- Control characters (start / end)
- Prepend bounding fields
- Due to loss of the carrier signal between the frames (code rule violation, see Chapter 1)

Control characters

In Chapter 1 we got to know the 4B5B code which is used in combination with line codes like MLT-3 on the physical layer.

- Each 4 bit input is mapped to 5 bit output
- A frame is preceded by the start symbols J/K
- After a frame the end symbols T/R are inserted

Input	Output	Meaning	Input	Output	Meaning
0000	11110	hex data 0	-	00000	quiet (signal loss)
0001	01001	hex data 1	-	11111	idle (pause)
0010	10100	hex data 2	-	11000	start #1 (J)
0011	10101	hex data 3	-	10001	start #2 (K)
0100	01010	hex data 4	-	01101	end (T)
0101	01011	hex data 5	-	00111	reset (R)
⋮	⋮	⋮	-	11001	set
1111	11101	hex data F	-	00100	halt

Example:

Input:			1011	0101	0110		
Output:	11000	10001	10111	01011	01110	01101	00111
	start #1	start #2	data	data	data	end	reset

Frame boundary detection and code transparency

Control characters are not only used on layers 1 and 2. On layer 6 (presentation layer), the ASCII code (American Standard Code for Information Interchange) is used (7 bit code words):

ASCII (hex)	Character	ASCII (hex)	Character	ASCII (hex)	Character	ASCII (hex)	Character
00	NUL	20	SP	40	@	60	`
01	SOH	21	!	41	A	61	a
02	STX	22	"	42	B	62	b
03	ETX	23	#	43	C	63	c
04	EOT	24	$	44	D	64	d
05	ENQ	25	%	45	E	65	e
06	ACK	26	&	46	F	66	f
07	BEL	27	'	47	G	67	g
08	BS	28	(	48	H	68	h
09	TAB	29	)	49	I	69	i
0A	LF	2A	*	4A	J	6A	j
0B	VT	2B	+	4B	K	6B	k
0C	FF	2C	,	4C	L	6C	l
0D	CR	2D	-	4D	M	6D	m
0E	SO	2E	.	4E	N	6E	n
0F	SI	2F	/	4F	O	6F	o
10	DLE	30	0	50	P	70	p
11	DC1	31	1	51	Q	71	q
12	DC2	32	2	52	R	72	r
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Frame boundary detection and code transparency

What if control characters occur randomly in the payload data?

1. In the case of the 4B5B code, this cannot happen:
   - 4 bit data words are injectively mapped to 5 bit code words
   - Some of the remaining 5 bit code words are used as control characters
2. The ASCII-Code is merely an interpretation rule:
   - Some code words represent text characters (digits, numbers, ...), others are control characters
   - To be able to transmit a control character as data, this is marked by a special control character: escape character
   - If this special control character itself is to be transmitted, it is doubled
   - This procedure is called character stuffing

Mostly code transparency is taken care of automatically, so users do not have to worry about it. This is not true for programming languages.

```
System.out.println("A \" must be escaped");
```

Within the string to be output, quotes must be escaped using a backslash.

More examples:

- Bash (`Ctrl+C`)
- Text editors (Emacs, vim)

Bounding fields and bit stuffing

- Mark start and end of a message with a specific bit sequence
- Make sure that the marker does not occur randomly in the payload data (“bit stuffing”)

Example:

- Let `01111110` be the start / end marker
- To prevent the marker from appearing in payload data, insert after five consecutive `1` bits a `0` bit in the payload

Input:		`1100101111110111111`	
Output:	`01111110`	`11001011111`**`0`**`10111111`**`0`**`1`	`01111110`

- Receiver removes the following `0` bit after five consecutive `1` bits.

Used e. g. by HDLC (High Level Data Link Control), layer 2 protocol for serial lines.

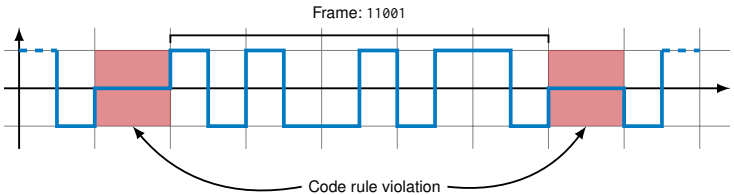
Code rule violation

Many line codes (e. g. Return-to-Zero and Manchester) have certain changes in signal levels independent of the actual data being transmitted.

Idea:

- Omit certain signal changes
- In this way an invalid symbol (not existing in the code) is created
- This can be used to mark the start and end of frames.

Example: Manchester code



IEEE 802.3a/i (Ethernet): 10 Mbit/s

- The Manchester code is used as line code
- The end of a frame is indicated by code rule violation

IEEE 802.3u (FastEthernet): 100 Mbit/s

- MLT-3 is used as line code in combination with 4B5B encoding
- Start and end of frames are marked by control characters of the 4B5B code

IEEE 802.3z (Gigabit Ethernet over fiber): 1000 Mbit/s

- NRZ is used as line code in combination with 8B10B encoding
- Start and end of frames are marked by control characters of the 8B10B code
- IEEE 802.3ab (Gigabit Ethernet over copper) uses another line codes since the attenuation would be too large with NRZ

In addition, in all these examples, each frame is still preceded by a preamble (see Chapter 1). This also ensures clock synchronization between transmitter and receiver, even when using the MLT-3 code.

Addressing and error detection

So far, we know

- how a binary data stream is transmitted and
- how the receiver recognizes frame boundaries.

However, we do not yet know

- how the payload coming from layer 3 and higher is handled by the data link layer,
- how the receiver of a frame is addressed, and
- how a frame is created from the payload (service data unit) and protocol-specific information.

Remark: All of the following concepts are explained using the IEEE 802 standards. The main points are transferable to other protocols with minor modifications.

Addressing in local area networks:

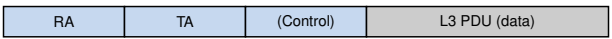
- All nodes can be reached directly
- There is no routing

Requirements for addresses on layer 2:

- Unique identification of nodes within the respective local area network
- Usually there is a broadcast address, which addresses all nodes in the respective local area network
- In addition, there may be multicast addresses[1], which address certain groups of nodes

Addresses on layer 2 are generally referred to as MAC addresses, where MAC stands for media access control.

Example:

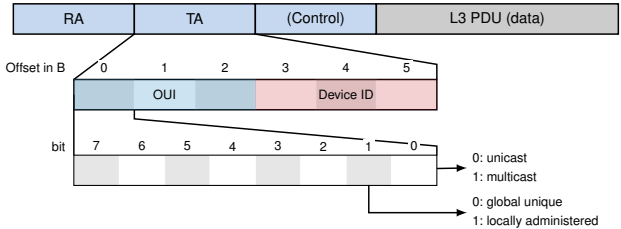


(RA = receiver address, TA = transmitter address)[2]

[1] Multicast addresses are handled like broadcasts in many cases on layer 2. However, they are extensively used in switched networks when IPv6 is used on layer 3.

[2] Since MAC addresses only address the next hop in each case, we avoid talking about source and destination addresses.

MAC addresses of all IEEE 802 standards (e. g. Ethernet, WLAN, Bluetooth) have the following structure:



- Network cards have a MAC address stored in the ROM (read-only memory).
- Separation into OUI (organizationally unique identifier) and device ID enables network card manufacturers to assign unique MAC addresses.
- Consequently, the manufacturer of a network card can be identified by its MAC address (e. g. `7c:6d:62` $\hat{=}$ Apple).
- The broadcast address is defined as `ff:ff:ff:ff:ff:ff` (“all ones”).
- Whether an address is a unicast or multicast address is determined by the lowest order bit of the first octet.

Remark: For certain applications, it makes sense to dispense with cross-vendor uniqueness, e. g. virtualized network adapters. For this purpose the so called local-administrated addresses (second bit of the first octet) are provided for this purpose.

Error detection

- Despite channel coding, transmission errors (bit errors) can occur.
- It can therefore happen that an incorrect payload is forwarded to higher layers.

To further reduce the probability of such errors being passed on, additional error-detecting codes are used (so-called checksums):

In contrast to channel coding (error-correcting codes), the checksum of a layer 2 protocol is usually not used for error correction but only for error detection.

Cyclic redundancy check (CRC)

Unlike error-correcting codes (channel codes, Chapter 1), CRC is a family of codes used primarily for error detection. The following objectives are pursued with their use:

- A large number of error types (single-bit, multi-bit, burst errors) should be detected.
- The amount of added redundancy should be small.
- Errors should primarily be detected, not corrected.

Unlike error-correcting codes (channel codes, Chapter 1), CRC is a family of codes used primarily for error detection. The following objectives are pursued with their use:

- A large number of error types (single-bit, multi-bit, burst errors) should be detected.
- The amount of added redundancy should be small.
- Errors should primarily be detected, not corrected.

Basics:

- A data word of length n bit can be represented by a polynomial

$$a(x) = \sum_{i=0}^{n-1} a_i x^i \text{ mit } a_i \in \mathbb{F}_2 \text{ mit } \mathbb{F}_2 = \{0,1\}.$$

- All data words of length exactly n bit constitute the set

$$F_q[x] = \left\{ a \;\middle|\; a(x) = \sum_{i=0}^{n-1} a_i x^i,\ a_i \in \mathbb{F}_2 \right\}.$$

- Together with appropriately defined addition and multiplication a so calles finite extension field $\langle F_q[x], +, \cdot \rangle$ with $q = 2^n$ elements is formed, on which the usual rules for addition and multiplication apply.

What means “appropriately defined”?

Summation: For the some of $a,b \in F_q[x]$ we define

$$c(x) = a(x) + b(x) = \sum_{i=0}^{n-1} (a_i + b_i)\, x^i,$$

where the sum $a_i + b_i$ of coefficients is defined according to the rules of GF(2)[2], i. e., the sum of two data words is the bitwise XOR.

[2] Galois field, finite field (cmp. Discrete Structures)

What means “appropriately defined”?

Summation: For the some of $a,b \in F_q[x]$ we define

$$c(x) = a(x) + b(x) = \sum_{i=0}^{n-1} (a_i + b_i)\, x^i,$$

where the sum $a_i + b_i$ of coefficients is defined according to the rules of GF(2)[2], i. e., the sum of two data words is the bitwise XOR.

Multiplication: Multiplication is more complex since the degree of $d(x) = a(x) \cdot b(x)$ can be larger than $n - 1$ and therefore $d(x) \notin F_q[x]$. We thus choose a reduction polynomial $r(x)$ with $\deg(r(x)) = n$ and define the product of $a,b \in F_q[x]$ as

$$d(x) = (a(x) \cdot b(x)) \bmod r(x).$$

- This corresponds to a normal polynomial multiplication (where the addition corresponds to an XOR-connection) followed by a modulo operation over $r(x)$.
- The modulo operation corresponds to a polynomial division with the remainder as result.
- This ensures that $\deg(d(x)) < n$.

[2] Galois field, finite field (cmp. Discrete Structures)

Example: $F_4[x] = \{0,1,x,x+1\}$ with $r(x) = x^2 + x + 1$. Is $r(x)$ irreducible, i. e. it cannot be represented as product of two polynomials with degree strictly less than n ($n = 2$ in this example)?

Example: $F_4[x] = \{0,1,x,x+1\}$ with $r(x) = x^2 + x + 1$. Is $r(x)$ irreducible, i. e. it cannot be represented as product of two polynomials with degree strictly less than n ($n = 2$ in this example)?

$\odot$	0	1	x	$x+1$
0	0			
1	0	1		
x	0	x	x^2	
$x+1$	0	$x+1$	$x^2 + x$	„$x^2 + 2x + 1$“ $= x^2 + 1$

Addition over $F_4[x]$ is equivalent to addition over $\mathbb{F}_2$ between monomials of the same degree.

$\Rightarrow$ Yes, $r(x)$ is irreducible since it cannot be represented as product $a \odot b$ with $a,b \in F_4[x]$.

Example: $F_4[x] = \{0,1,x,x+1\}$ with $r(x) = x^2 + x + 1$. Is $r(x)$ irreducible, i. e. it cannot be represented as product of two polynomials with degree strictly less than n ($n = 2$ in this example)?

$\odot$	0	1	x	$x+1$
0	0			
1	0	1		
x	0	x	x^2	
$x+1$	0	$x+1$	x^2+x	„$x^2 + 2x + 1$“ $= x^2 + 1$

Addition over $F_4[x]$ is equivalent to addition over $\mathbb{F}_2$ between monomials of the same degree.

$\Rightarrow$ Yes, $r(x)$ is irreducible since it cannot be represented as product $a \odot b$ with $a,b \in F_4[x]$.

For multiplication, we need $r(x)$ to reduce the colored results in the table above:

+	0	1	x	$x+1$
0	0			
1	1	0		
x	x	$x+1$	0	
$x+1$	$x+1$	x	1	0

·	0	1	x	$x+1$
0	0			
1	0	1		
x	0	x	$x+1$	
$x+1$	0	$x+1$	1	x

Multiplication examples:

$x^2 : (x^2 + x + 1) = 1$, remainder: $x + 1$ $\quad (x^2 + x) : (x^2 + x + 1) = 1$, remainder: 1 $\quad (x^2 + 1) : (x^2 + x + 1) = 1$, remainder: x

Remarks:

- If we choose an irreducible polynomial for $r(x)$, i. e., $r(x)$ cannot be represented as product of $a,b \in F_q[x]$, then we obtain a finite field with $q = 2^n$ elements.
- For CRC one usually chooses $r(x) = p(x)(x + 1)$ with $p \in F_q[x]$ as reduction polynomial of degree n:
  - $p(x)$ and $x + 1$ *are* elements of $F_q[x]$ and $r(x)$ is therefore obviously not irreducible.
  - With this choice for $r(x)$, $\langle F_q[x], +, \cdot \rangle$ is not a finite field.
  - However, this choice allows to detect all bit errors of odd number.
- The choice of $r(x)$ determines not only the length of the checksum but also the error detection properties.

Remarks:

- If we choose an irreducible polynomial for $r(x)$, i. e., $r(x)$ cannot be represented as product of $a,b \in F_q[x]$, then we obtain a finite field with $q = 2^n$ elements.
- For CRC one usually chooses $r(x) = p(x)(x + 1)$ with $p \in F_q[x]$ as reduction polynomial of degree n:
  - $p(x)$ and $x + 1$ *are* elements of $F_q[x]$ and $r(x)$ is therefore obviously not irreducible.
  - With this choice for $r(x)$, $\langle F_q[x], +, \cdot \rangle$ is not a finite field.
  - However, this choice allows to detect all bit errors of odd number.
- The choice of $r(x)$ determines not only the length of the checksum but also the error detection properties.

Back to CRC

- CRC calculates a checksum of fixed length for a given data block (e. B. L2-PDU).
- Code words are polynomials $a \in F_q[x]$.
- The degree n of the reduction polynomial $r(x)$ determines
  - the maximum degree $n - 1$ of valid code words $a \in F_q[x]$ and
  - which types of errors (single-bit, multi-bit, burst errors) can be detected.
- Ethernet uses CRC32 with the reduction polynomial

$$r(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1.$$

How does CRC work?

Suppose we have a reduction polynomial $r(x)$ of degree n and a message $m(x)$ of degree k, i. e. the message consists of $k + 1$ bit, that is to be secured using CRC:

1. Append n zeros to $m(x)$: $m'(x) = m(x) \cdot x^n$.
2. Determine the remainder $c(x) = m'(x) \bmod r(x)$, which represents the checksum.
3. The message to be transmitted is $s(x) = m'(x) + c(x)$.

How does CRC work?

Suppose we have a reduction polynomial $r(x)$ of degree n and a message $m(x)$ of degree k, i. e. the message consists of $k + 1$ bit, that is to be secured using CRC:

1. Append n zeros to $m(x)$: $m'(x) = m(x) \cdot x^n$.
2. Determine the remainder $c(x) = m'(x) \bmod r(x)$, which represents the checksum.
3. The message to be transmitted is $s(x) = m'(x) + c(x)$.

The receiver checks the incoming message $s'(x) = s(x) + e(x)$, which may contain an error $e(x) \neq 0$:

1. The remainder $c'(x) = s'(x) \bmod r(x) = (s(x) + e(x)) \bmod r(x)$ is determined.
2. If $c'(x) = 0$, then with high probability no error occurred. If $c'(x) \neq 0$, then there was an error for sure.

Example: Reduction polynomial $r(x) = x^3 + x^2 + 1$, data $m(x) = x^7 + x^5 + x^2 + 1$

1. Determine coefficients: $r(x) \hat{=} 1101$ and $m(x) \hat{=} 10100101$

Example: Reduction polynomial $r(x) = x^3 + x^2 + 1$, data $m(x) = x^7 + x^5 + x^2 + 1$

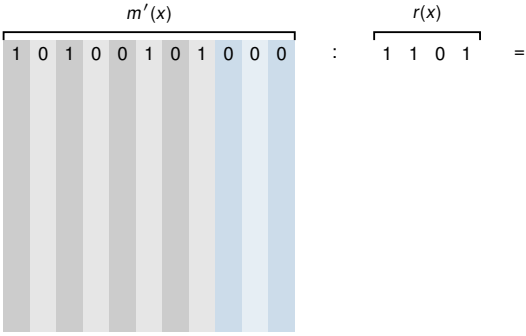
1. Determine coefficients: $r(x) \mathrel{\hat{=}} 1101$ and $m(x) \mathrel{\hat{=}} 10100101$
2. $\deg(r(x)) = 3 \Rightarrow$ multiply data by x^3. This corresponds to "appending" 3 zeros: $m'(x) = m(x) \cdot x^3 \mathrel{\hat{=}} 10100101\mathbf{000}$

Addressing and error detection

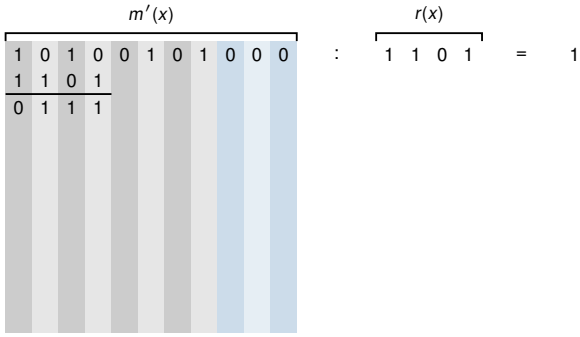
Example: Reduction polynomial $r(x) = x^3 + x^2 + 1$, data $m(x) = x^7 + x^5 + x^2 + 1$

1. Determine coefficients: $r(x) \mathrel{\hat{=}} 1101$ and $m(x) \mathrel{\hat{=}} 10100101$
2. $\deg(r(x)) = 3 \Rightarrow$ multiply data by x^3. This corresponds to "appending" 3 zeros: $m'(x) = m(x) \cdot x^3 \mathrel{\hat{=}} 10100101\mathbf{000}$
3. Determine the remainder of the polynomial division $m'(x)/r(x)$, which is the checksum $c(x)$.

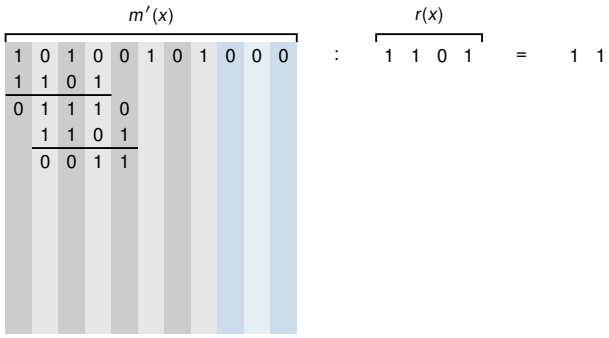
Example: Reduction polynomial $r(x) = x^3 + x^2 + 1$, data $m(x) = x^7 + x^5 + x^2 + 1$



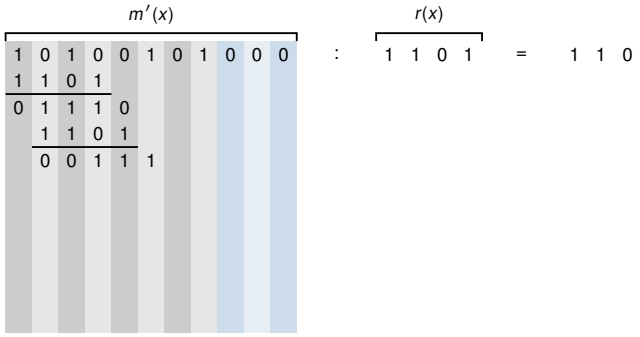
Example: Reduction polynomial $r(x) = x^3 + x^2 + 1$, data $m(x) = x^7 + x^5 + x^2 + 1$



Example: Reduction polynomial $r(x) = x^3 + x^2 + 1$, data $m(x) = x^7 + x^5 + x^2 + 1$



Example: Reduction polynomial $r(x) = x^3 + x^2 + 1$, data $m(x) = x^7 + x^5 + x^2 + 1$

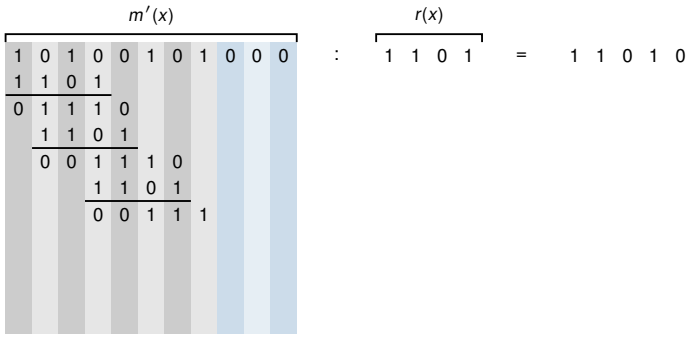


Example: Reduction polynomial $r(x) = x^3 + x^2 + 1$, data $m(x) = x^7 + x^5 + x^2 + 1$

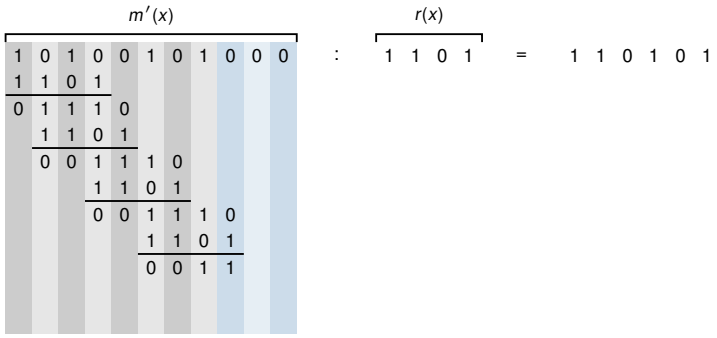
$m'(x)$: $r(x)$ = 1 1 0 1

1	0	1	0	0	1	0	1	0	0	0	
1	1	0	1								
0	1	1	1	0							
	1	1	0	1							
	0	0	1	1	1	0					
			1	1	0	1					
			0	0	1	1					

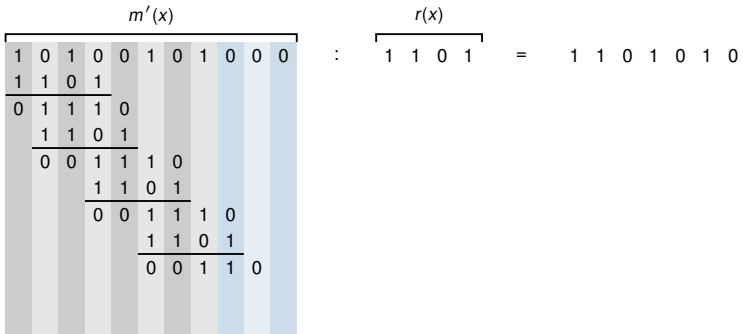
Example: Reduction polynomial $r(x) = x^3 + x^2 + 1$, data $m(x) = x^7 + x^5 + x^2 + 1$



Example: Reduction polynomial $r(x) = x^3 + x^2 + 1$, data $m(x) = x^7 + x^5 + x^2 + 1$



Example: Reduction polynomial $r(x) = x^3 + x^2 + 1$, data $m(x) = x^7 + x^5 + x^2 + 1$



Example: Reduction polynomial $r(x) = x^3 + x^2 + 1$, data $m(x) = x^7 + x^5 + x^2 + 1$

$m'(x)$: $r(x)$ = 1 1 0 1 0 1 0 1

1	0	1	0	0	1	0	1	0	0	0	
1	1	0	1								
0	1	1	1	0							
	1	1	0	1							
	0	0	1	1	1	0					
			1	1	0	1					
			0	0	1	1	1	0			
					1	1	0	1			
					0	0	1	1	0	0	
							1	1	0	1	
							0	0	0	1	$= c(x)$

Example: Reduction polynomial $r(x) = x^3 + x^2 + 1$, data $m(x) = x^7 + x^5 + x^2 + 1$

1. Determine coefficients: $r(x) \mathrel{\hat{=}} 1101$ and $m(x) \mathrel{\hat{=}} 10100101$
2. $\deg(r(x)) = 3 \Rightarrow$ multiply data by x^3. This corresponds to "appending" 3 zeros: $m'(x) = m(x) \cdot x^3 \mathrel{\hat{=}} 10100101\mathbf{000}$
3. Determine the remainder of the polynomial division $m'(x)/r(x)$, which is the checksum $c(x)$.
4. The message to be transmitted is $s(x) = m'(x) + c(x)$. The addition is a bitwise XOR operation since we operate over GF(2).

$$
\begin{array}{rcccccccccccl}
 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & = m'(x) \\
\oplus & & & & & & & & & 0 & 0 & 1 & = c(x) \\
\hline
 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & = s(x)
\end{array}
$$

Example: Reduction polynomial $r(x) = x^3 + x^2 + 1$, data $m(x) = x^7 + x^5 + x^2 + 1$

1. Determine coefficients: $r(x) \mathrel{\hat{=}} 1101$ and $m(x) \mathrel{\hat{=}} 10100101$
2. $\deg(r(x)) = 3 \Rightarrow$ multiply data by x^3. This corresponds to "appending" 3 zeros: $m'(x) = m(x) \cdot x^3 \mathrel{\hat{=}} 10100101\mathbf{000}$
3. Determine the remainder of the polynomial division $m'(x)/r(x)$, which is the checksum $c(x)$.
4. The message to be transmitted is $s(x) = m'(x) + c(x)$. The addition is a bitwise XOR operation since we operate over GF(2).

$$
\begin{array}{rcccccccccccl}
 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & = m'(x) \\
\oplus & & & & & & & & & 0 & 0 & 1 & = c(x) \\
\hline
 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & = s(x)
\end{array}
$$

The receiver checks the message by determining $c'(x) = (s(x) + e(x)) \bmod r(x)$, where $e(x)$ represents possible transmission errors:

- $c'(x) \neq 0$ means that there were errors.
- $c'(x) = 0$ means that with high probability there were no errors.

Which errors are detected by CRC32?

Let n denote the length of the checksum, i. e. $n = \deg(r(x))$. Then the following types of errors can be detected:

- all 1 bit errors
- isolated 2 bit errors, i. e., errors at bit positions i and j where $i > j$ such that $i - j > n$
- some burst errors that are longer than n

Depending on the concrete choice of the reduction polynomial we could also detect either

- all burst errors with length smaller than n or
- all error patterns with an odd number of flipped bits.

Which errors cannot or not reliably detected?

- errors with more than n flipped bits
- errors consisting of multiple bursts
- all errors that are a multiple of the reduction polynomial

Can CRC really not be used to correct errors?

It can, but ...

- CRC is initially designed as an error-detecting code, i. e. correction is no main objective.
- This does not exclude, however, that with a suitable choice of the reduction polynomial and with an appropriate ratio of useful data to redundancy, certain errors cannot also be corrected.

Examples[1]:

- ATM (Asynchronous Transfer Mode), for instance, uses a 1 B checksum to detect errors of 1,2,3 bit and to correct single-bit errors in the 4 B ATM Header.
- Bluetooth secures 10 bit data blocks with a 5 bit checksum, which allows to correct single-bit errors.

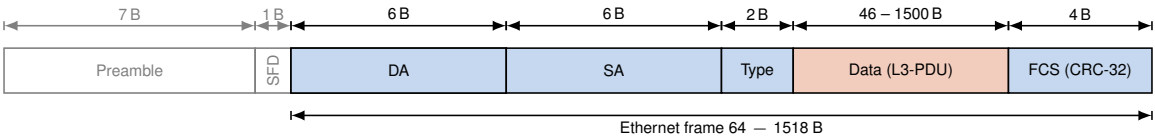
So in both cases, CRC is used as an error-correcting block code. The code rate is $4/5$ with ATM and $2/3$ with Bluetooth.

What about Ethernet?

- Ethernet frames have a default length of up to $1500 \cdot 8$ bit (not considering so called “jumbo frames”).
- The checksum is 32 bit long.
- That results in a code rate of approximately 0,997.
- CRC is not used here for error correction.

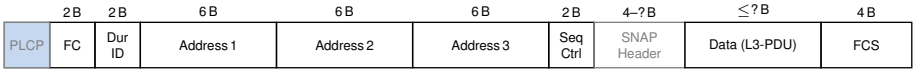
[1] http://einstein.informatik.uni-oldenburg.de/rechnernetze/fehlerkorrektur1.htm

Frame before 4B5B encoding:



- Preamble and Start Frame Delimiter (SFD) are used for clock recovery.
- One Byte of the preamble is replaced by the J/K symbol of the 4B5B code (SFD).
- After the Frame Check Sequence (FCS), the T/R symbol is appended.
- The data between the J/K and T/R symbols are encoded according to the 4B5B code.
- The type field indicates the type of the frame, e. g. `0x0800` ≙ IPv4 payload, `0x0806` ≙ ARP).
- The data field field must be at least 46 B before encoding – otherwise it is padded up to the minimal length.

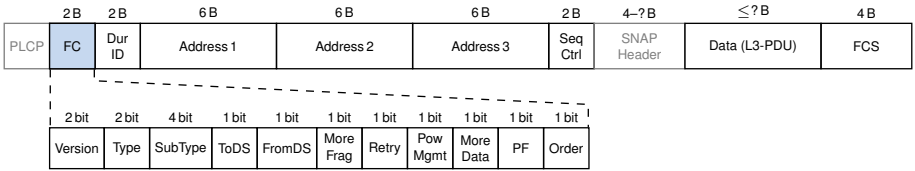
Data frame as used in the infrastructure mode, i. e., with access point, without encryption:



Physical Layer Convergence Procedure (PLCP)

- Header of the physical layer (comparable to the preamble in Ethernet)
- Serves for synchronization as well as for communication of transmission parameters (data rate, modulation, code rate, etc.)
- Not part of the L2 header

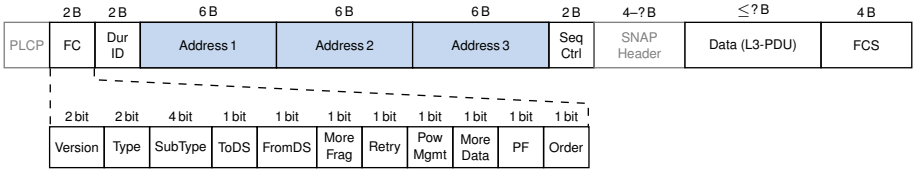
Data frame as used in the infrastructure mode, i. e., with access point, without encryption:



Frame Control (FC)

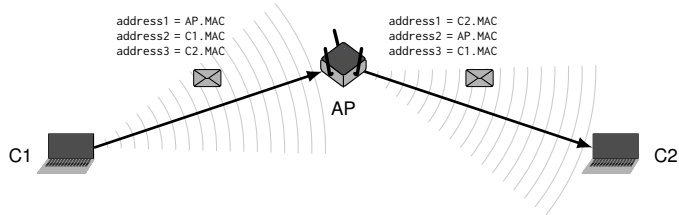
- Indicates the type of the frame (data, management, or control frame)
- Defines how the addresses contained in the header are interpreted (ToDS / FromDS Bits)
- Various other parameters:
  - Do other fragments follow that belong to the same frame?
  - Is it a retransmit of a previous frame that was assumed to be lost?
  - Does the transmitter has more data to send?
  - . . .

Data frame as used in the infrastructure mode, i. e., with access point, without encryption:

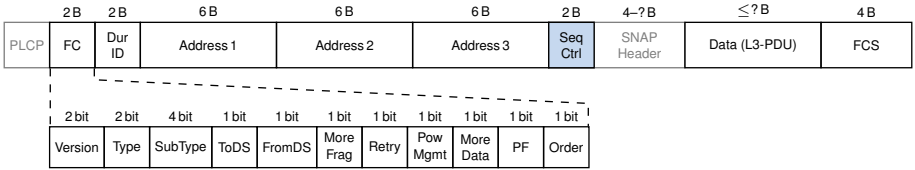


Variable number of MAC addresses depending on frame type and operation mode, here data frames in infrastructure mode

- Address 1 specifies the receiver (receiver address, RA)
- Address 2 specifies the transmitter (transmitter address, TA)
- Address 3 specifies the source (source address, SA) or the destination (destination address, DA)



Data frame as used in the infrastructure mode, i. e., with access point, without encryption:

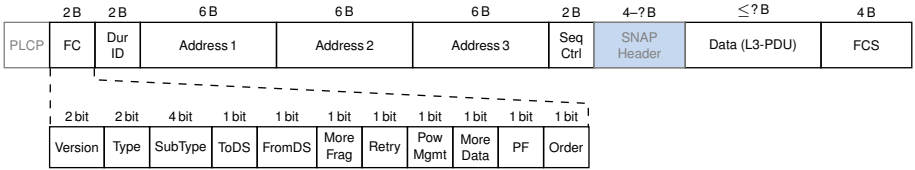


Sequence Control

- Sequence number of the frame
- Used to detect missing frames and to sort incoming frames

Note: In contrast to Ethernet, WLAN uses acknowledgements on layer 2 since the transmission medium is too unreliable. This is not a replacement for acknowledgements on higher layers, but a necessity that protocols on higher layers can operate at all.

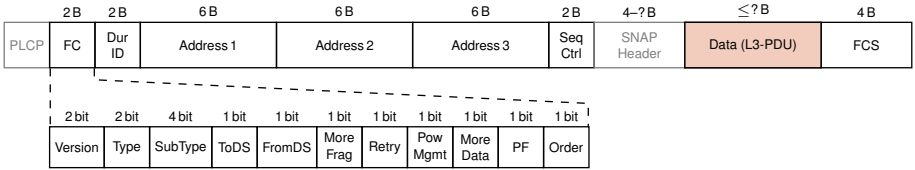
Data frame as used in the infrastructure mode, i. e., with access point, without encryption:



Subnetwork Access Protocol (SNAP)

- Variable length header indicating the type of the L3 PDU
- Roughly comparable to the Ethertype (but much more flexible)

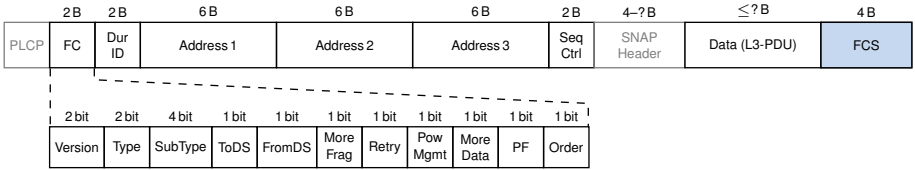
Data frame as used in the infrastructure mode, i. e., with access point, without encryption:



Data (L3 PDU)

- Variable length data field
- The maximum frame length with IEEE 802.11 is multiple times larger than with Ethernet due to the expensive media access
  - The media access requires a lot of time (large overhead)
  - The smaller individual frames are, the more time is consumed by repeated media access
    ⇒ Frames tend to be larger even so that increases the probability of transmission errors

Data frame as used in the infrastructure mode, i. e., with access point, without encryption:



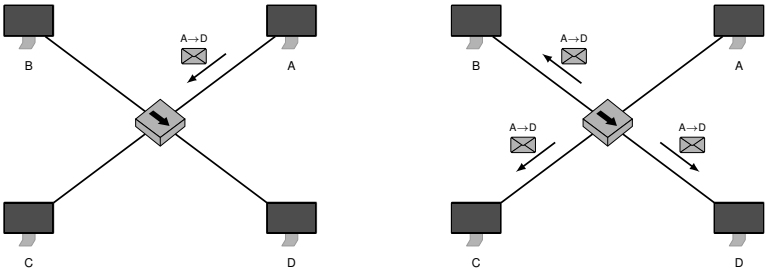
Frame Check Sequence (FCS)

- 32 bit CRC checksum over the whole L2 frame (everything exept the PLCP and the FCS itself)
- Except for implementation details identical to Ethernet
- A different reduction polynomial is used compared to Ethernet

Chapter 2: Data link layer

Links on layer 1: hub

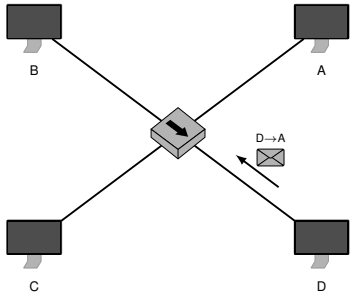
- Node A transmits a frame destined for node D
- The hub interconnects the individual links to a common bus
- The frame is received by all nodes
- Consequently, at any time only one node may transmit, otherwise collisions occur

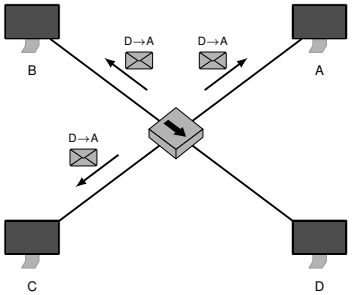


Important: Most layer 2 implementations are connectionless, i. e., there is no connection setup or teardown between source and destination[1].

[1] Ethernet uses the terms source and destination, although transmitter and receiver may be more precise and avoid confusion with source and destination from the perspective of layer 3. Making things even more confusing, all four terms are required to describe the addresses in wireless networks. Therefore, we stick to the usual naming convention.

- Node D answers to the frame of A
- This answer is again received by all nodes





Definition (Collision Domain)
A collision domain is that part of a local area network within which a collision can occur when multiple nodes are transmitting simultaneously. It is commonly referred to as segment.

Are hubs more than just star distributors?

A distinction is made between active and passive hubs:

- Active hubs (repeaters) amplify signals on the physical layer without checking the fields in frames like addresses or checksums
- Passive Hubs are really just star distributors – you might as well solder the individual wires of the patch cables together

Can you cascade hubs? Yes, but for Ethernet (IEEE 802.3a/i) the 5-4-3 rule applies[1]:

- Not more than 5 sections (of 100 m length each),
- connected by 4 repeaters,
- with active nodes in only 3 of these sections.

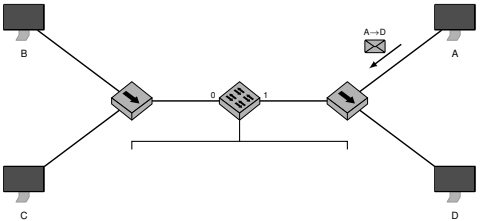
Note: Each section should be no longer than 185 m for 802.3a (10BASE-2), and no longer than 100 m for 802.3i (10BASE-T) between the hub and the end device due to attenuation. Due to reliable collision detection, 100BASE-TX results in a maximum extension of 500 m (will be discussed in the tutorial).

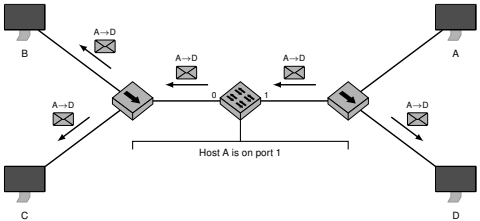
Can hubs connect different types of media?

- Yes, if the same media access method is used on all sections (for example, connecting Ethernet via BNC and patch cables, each with the same data rate).
- It is not possible if the media access control schemes differ.

[1] Note that this rule does not apply to modern, switched Ethernets.

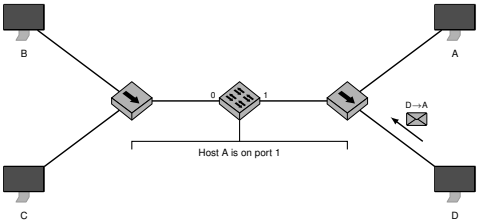
Links on layer 2: switch

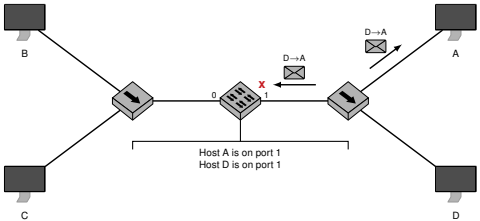




- Two groups of hosts, each connected by hubs, are coupled by a switch in the above example.
- The switch initially works like a hub with 2 ports (learning phase).
- Thereby the switch remembers over which port a frame was received.
- Thus, it assigns to ports 0 and 1 the MAC addresses of the nodes connected to the respective port.
- A switch with only two ports (which is more common again today in the context of virtualization) is also called bridge.

Links on layer 2: switch

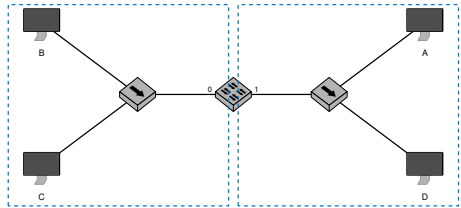




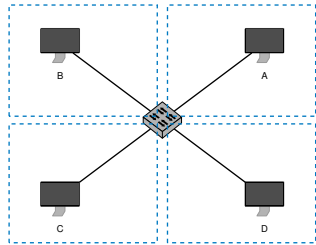
- The destination address of incoming frames is compared with the entries in the switching table.
- If an entry exists, the frame is forwarded only to the destination port in question.
- Otherwise the frame is broadcast over all ports (except the port from which it was initially received.)
- Entries receive a timestamp and are invalidated after a fixed time interval.
- One port may have mappings for several MAC addresses.

Links on layer 2: switch

- A switch or bridge interrupts collision domains (also called segmentation).
- If a switch is aware of all connected devices, one node in each segment is allowed to transmit at the same time.



- If exactly one host is connected per switch port, this is called microsegmentation. This is the norm today.
- In this case, any two hosts can communicate with each other simultaneously.
- Since all Ethernet standards using patch cables or fiber optics also support full-duplex operation between two nodes, nodes may transmit and receive concurrently.

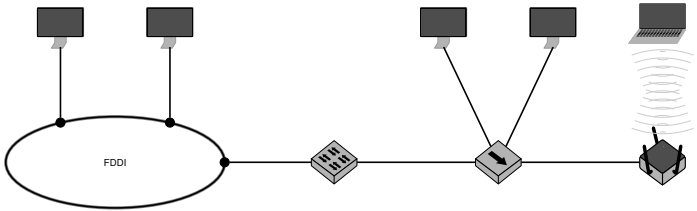


Switches can also be used to couple network segments with different access methods:

- FDDI-Ethernet switch between Token Passing und CSMA/CD
- WLAN access point between CSMA/CD and CSMA/CA

This coupling is transparent, i. e.,

- connected stations do not notice that a switch is being used, and
- during normal operation, a host will never communicate directly with a switch.



Prerequisite: The MAC addresses must be “compatible” to be able to identify the respective destination via its MAC address.

Remarks

- Switches are transparent to hosts, which means that a host does not know that it is communicating with other hosts through a switch.
- Source and destination addresses are not modified by switches.
- Switches do not restrict accessibility within a local area network.
- A broadcast (MAC address `ff:ff:ff:ff:ff:ff`) is relayed to all hosts (therefore it is also called broadcast domains as opposed to a collision domain).
- A switch does not need its MAC address to perform its basic tasks.
- Forwarding decisions are made based on the destination address and the current switching table.

Furthermore, a distinction is made between two different types of switching:

- Store-and-Forward: Incoming frames are received in full and their FCS is checked before being forwarded. If the output port is occupied, a limited number of frames can be buffered.
- Cut-Through: Start serializing the frame once the output port has been determined. The FCS is not checked in that case.

Loops on layer 2

- Loops at layer 2 result in multiple copies of a frame being created and circulating in the network.
- This would eventually lead to an overload situation and malfunction.

How are loops formed?

- Even if local area networks are spatially limited, it is easy to lose track of what is going on and unintentionally create loops.
- Topologies with redundant paths are used to build robust local networks.
- If a link or switch fails, traffic can be redirected. From redundant paths, loops can arise.

How to avoid loops?

- Switches used in larger deployments support the so-called Spanning Tree Protocol (STP).
- The goal is to deactivate redundant paths so that all network segments can be reached loop-free.
- Thus, protocol creates a Minimum Spanning Tree (MST), not to be confused with a Shortest Path Tree (SPT).
- If a connection fails, one of these paths may be reactivated.

WLAN access points

WLAN access points are essentially bridges between wired and wireless networks:

- one RJ45 interface connects to the Ethernet
- a wireless transceiver communicates with stations in the wireless network

However, there is a significant difference to bridges or switches:

- WLAN access points according to IEEE 802.11 standards are not transparent to wireless clients on layer 2!
  - Clients are aware of the access point.
  - The access point is explicitly addressed from within the wireless network.
- **Question:** What does that look like when a wired device (Ethernet) sends a frame to a WLAN client?

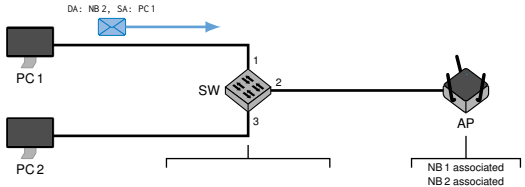
The following is common to both switches and access points:

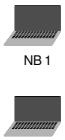
- Forwarding decisions are solely based on MAC addresses.
- They interrupt collision domains, i. e., a frame is not forwarded by an AP if the receiver is not connected (“associated”) to that AP.

Important: Since this is a broadcast medium, only one transmission can take place at a time, since frames would otherwise collide on layer 1.

Example: PC 1 wants to send a frame to NB 2. We assume that

- NB 1 and NB 2 are associated with the AP and
- there has not yet been any other communication on the network.

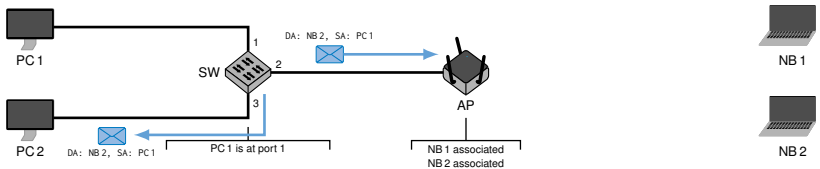




- PC 1 transmits a frame to NB 2.
- Source address (SA) and destination address (DA) are thus fixed.

Example: PC 1 wants to send a frame to NB 2. We assume that

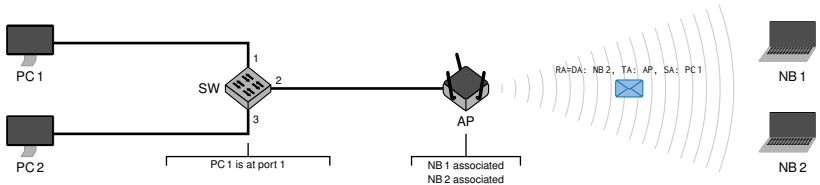
- NB 1 and NB 2 are associated with the AP and
- there has not yet been any other communication on the network.



- The switch SW learns that PC 1 is attached to port 1.
- The destination NB 2, however, is still unknown. The frame is thus broadcast on all ports.

Example: PC 1 wants to send a frame to NB 2. We assume that

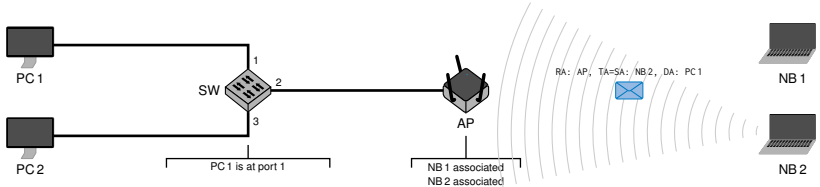
- NB 1 and NB 2 are associated with the AP and
- there has not yet been any other communication on the network.



- The access point AP receives the frame and knows that NB 2 is an associated station.
- It thus accepts and converts the frame from IEEE 802.3 to IEEE 802.11.
- The receiver address (RA) corresponds destination address (DA).
- The transmitter address (TA) is the MAC address of the AP.
- The source address (SA) remains the address of PC 1.
- NB 2 accepts the frame, NB 1 overhears but ignores it.

Example: PC 1 wants to send a frame to NB 2. We assume that

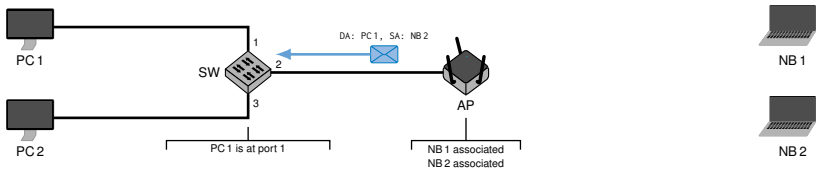
- NB 1 and NB 2 are associated with the AP and
- there has not yet been any other communication on the network.



- NB 2 answers with a new frame.
- The receiver address (RA) is the AP.
- The transmitter address (TA) is the same as the source address (SA).
- The destination address (DA) is PC 1.
- The AP overhears the frame and accepts it as it is directed to the AP.

Example: PC 1 wants to send a frame to NB 2. We assume that

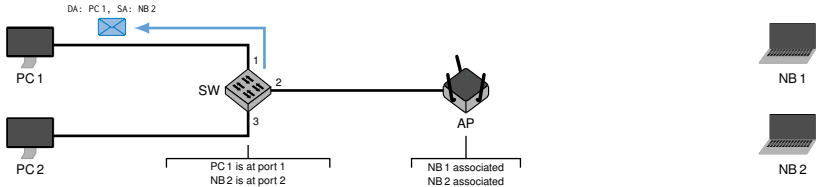
- NB 1 and NB 2 are associated with the AP and
- there has not yet been any other communication on the network.



- The AP knows that PC 1 is not an associated wireless station (in fact it knows that it is connected to the wired network as it previously received the Ethernet frame destined to NB 1).
- The AP thus converts the frame from IEEE 802.11 back to IEEE 802.3.
- The source address (SA) is NB 2.
- The destination address (DA) is PC 1.

Example: PC 1 wants to send a frame to NB 2. We assume that

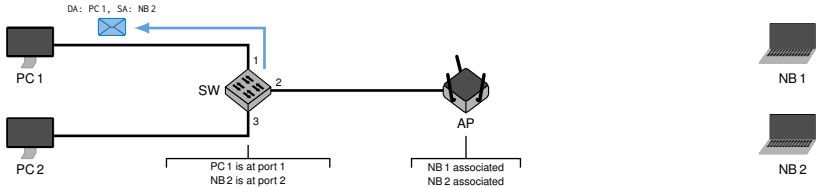
- NB 1 and NB 2 are associated with the AP and
- there has not yet been any other communication on the network.



- SW 1 now learns that NB 2 is reachable via port 2.
- Since it also knows that PC 1 is attached to port 1, it forwards the frame only over that port.
- PC 1 accepts the frame since the destination address (DA) matches its own MAC address.
- Neither PC 1 nor NB 2 noticed that the respective other communication partner is attached via a completely different media access control scheme to the same local area network.

Example: PC 1 wants to send a frame to NB 2. We assume that

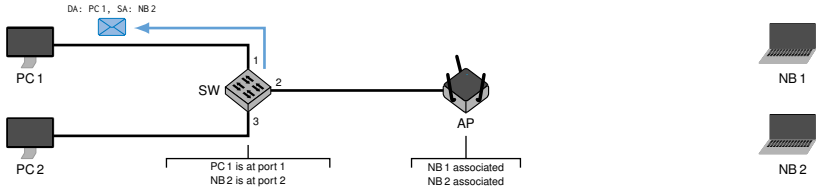
- NB 1 and NB 2 are associated with the AP and
- there has not yet been any other communication on the network.



Important: In contrast to switches, the AP is explicitly addressed by its clients. The AP is therefore not transparent within the wireless network, but invisible to the wired network.

Example: PC 1 wants to send a frame to NB 2. We assume that

- NB 1 and NB 2 are associated with the AP and
- there has not yet been any other communication on the network.



Important: In contrast to switches, the AP is explicitly addressed by its clients. The AP is therefore not transparent within the wireless network, but invisible to the wired network.

Question: How does PC 1 actually now a MAC address like `de:ad:be:ef:00:01` that belongs to NB 2?

The term “WLAN router” is, for the most part, wrong from a technical perspective:

- Manufactures sell devices that combine a
  - DSL or cable modem (layer 1),
  - Ethernet switch (layer 2),
  - WLAN access point (layer 2), and
  - router (Ethernet ↔ DSL/cable/etc.).
- Looking only at the capabilities as access point, it is nothing else than a switch with integrated media converters.

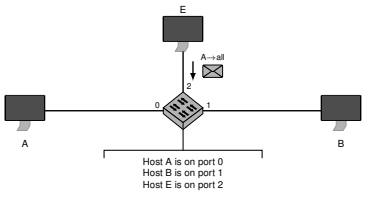
Routing (see Chapter 3) does not take place within wireless networks in infrastructure mode.

Be aware that this term is commonly used for devices with at least four different functions.

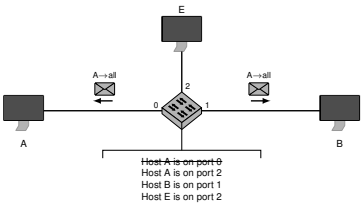
Chapter 2: Data link layer

CAM poisoning

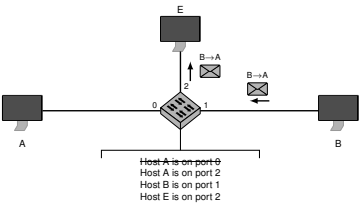
- Switches maintain a switching table, also known as Content Addressable Memory (CAM) Table, to associate SAs with switch ports.
- An attacker with physical access may forge his SA, making the switch believe that another host is connected to that port.
- The switch will incorrectly forward frames to the attacker while making the real destination unreachable.



(a) E sends a frame with faked SA



(b) The CAM of the switch is updated / poisoned



(c) A legitimate frame destined for A is forwarded to E

Denial of Service

- An attacker may also flood the switch with arbitrary source addresses.
- This can easily lead to an overload at the switch since CAM is very limited.

Mitigations

Only switch ports exposed to the outside (so called edge ports) are vulnerable. In most cases only a single host is attached to such a port (which is also why STP is disabled on those ports). Depending on the switch and its capabilities, there may be port security features:

- Allowed SAs may be statically configured per port, denying any unknown SAs.
  - Secure, but high level of maintenance required
- Allowed SAs may be learned over time and be saved administratively.
  - Still manual intervention needed
  - Set of allowed addresses may change over time
- Ports may be configured to accept only the first SA observed per port.
  - Dynamic for the most part
  - Manual intervention only required when a host is exchanged

Even when such port security features are used, attacks remain possible: there is a mapping between MAC and IP addresses (ARP / neighbor resolution, see Chapter 3).

- An attacker (e. g. a compromised host) may transmit unsolicited ARP replies or neighbor advertisements with its correct SA but forged payload that contains a wrong mapping between IP and MAC addresses.
- If other hosts on the network do not check the SA against the payload (which they likely don't), their ARP caches get poisoned.
- Such a node would direct frames to the wrong (malicious) node, which seems to be perfectly fine from the perspective of the switch.

Security Considerations

L2 vulnerabilities in IEEE 802.11

Cables are easy

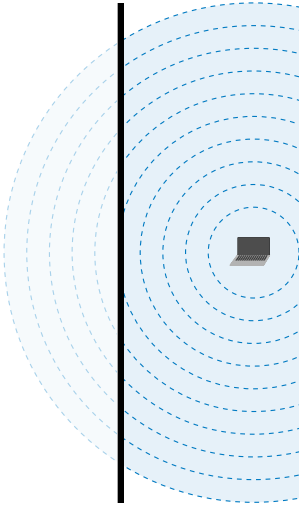
- Cables are very effective in controlling the physical propagation of the information transmitted
  - For a cabled network, access is restricted through the need to access it physically
  - That is, attaching a cable to your notebook
  - This allows to secure the medium by the means of a locked door
- This does not translate well to wireless media

Wireless is not

- Adversaries can see your wireless traffic, behind walls, without you noticing
- Adversaries can participate in your network, behind walls, without you seeing them directly

⇒ You cannot really do anything against adversaries using the same medium as you

⇒ We therefore need a way to keep the medium safe without relying on physical restrictions



Limiting access to the medium

- We want to limit medium access to clients that are authorized
- We need to decide for each client whether we trust it or not

Knowledge is access

- All clients shall prove knowledge of a (possibly unique) pre-shared secret
- Participating on the medium is only permitted if a client demonstrates knowledge of the secret
- The traffic of unauthorized clients is ignored

Knowledge is not power

- We can prevent adversaries from participating in the network
- We cannot prevent adversaries from blocking the medium
  - ! Anybody can jam on a known frequency
  - ⇒ This will prevent communication on that channel

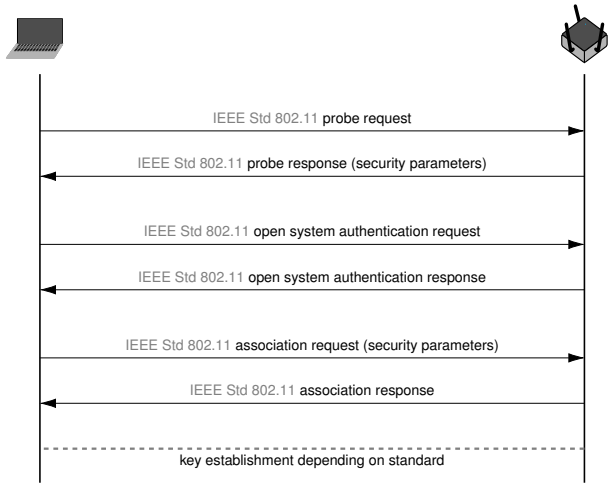


Figure 4: Associating with a wireless network according to IEEE 802.11 [1, p. 284]

Wired Equivalent Privacy (WEP)

- The first (1997) wireless security standard for IEEE 802.11
- Method: apply encryption[1] to the traffic

⇒ Goal: provide security comparable to that of a wire
  - It did exactly that, but not more
  - It provides no authentication[2] for data, allowing for replay attacks

- It did not last very long, as it had pretty terrible security
  - Replaying a recorded frame that generates a response many times generates an arbitrary amount of encrypted data, e. g. an ARP request
  - By deauthenticating a client (next slide), such a request can be triggered
  - Recording the generated traffic allows to derive the key within seconds
- It was about to ruin the image of Wi-Fi

[1] Encryption: process of changing data through a deterministic, reversible process, which hides the information and makes it retrievable to only parties in possession of a common secret

[2] Authentication: data is authenticated if its origin and integrity can be proven. This is commonly achieved through processes involving cryptography.

Security Considerations

L2 vulnerabilities in IEEE 802.11 – Wireless security standards

Wired Equivalent Privacy (WEP)

- The first (1997) wireless security standard for IEEE 802.11
- Method: apply encryption[1] to the traffic

⇒ Goal: provide security comparable to that of a wire
  - It did exactly that, but not more
  - It provides no authentication[2] for data, allowing for replay attacks

- It did not last very long, as it had pretty terrible security
  - Replaying a recorded frame that generates a response many times generates an arbitrary amount of encrypted data, e. g. an ARP request
  - By deauthenticating a client (next slide), such a request can be triggered
  - Recording the generated traffic allows to derive the key within seconds
- It was about to ruin the image of Wi-Fi

Wi-Fi Protected Access (WPA)

- WPA: Intermediate successor (2003) of WEP
  - TKIP mode is hardware-compatible with WEP but fixes the flaws that allowed calculating the key too easily
  - WPA provided a drop-in replacement until a new standard was defined
  - CCMP mode introduces better encryption (AES)
- WPA2 (2004) finally then switches to AES for encryption
- WPA3 is the most recent standard (2018)
  - Forward secrecy[3]
  - Longer keys (up to 256 bit key length)
  - Opportunistic encryption in open systems (no preshared keys)

[1] Encryption: process of changing data through a deterministic, reversible process, which hides the information and makes it retrievable to only parties in possession of a common secret

[2] Authentication: data is authenticated if its origin and integrity can be proven. This is commonly achieved through processes involving cryptography.

[3] Forward secrecy: an attacker is unable to decrypt recorded, encrypted traffic in case the shared secret becomes known in the future

Deauthentication attack

In an unencrypted WLAN nodes inherently have access to all data being transmitted. If they manage to associate with an AP, they can mount similar attacks as in a wired network. Encryption *should* protect from that. However, there is still an easy way for DoS attacks:

- IEEE 802.11 differentiates between data, management, and control frames.
  - Data frames are what the name suggests
  - Management frames are used to maintain a connection between client and AP, e. g. authentication, association, roaming, etc.
  - Control frames are used for media access, e. g. acknowledgements, RTS / CTS etc.
- In general, only data frames are encrypted (if encryption is used at all).
- There is a special management frame used to disconnect (deauthenticate) clients by the AP.
- This frame can easily be forged by anyone in range and directed to the victim, which will assume that it has been disconnected and starts a new association attempt.

Mitigation

- IEEE 802.11w is an extension to encrypt (some types of) management frames.
- WPA3 *required* protected management frames

What about control frames and jamming?

- An attacker may still impersonate the AP and transmit forged CTS frames, impacting the ability of nodes to acquire medium access
- There is no way to prevent jamming the medium

WPA2/3 works in one of two modes: personal and enterprise

Personal Mode

- Uses a pre-shared key
- Same master secret for all users
- Encryption keys are derived from this master secret

} What you find at home

Enterprise Mode

- Delegates authentication to an external provider
- Unique master secret per client
- Authentication using username and password or certificates

} What you find at university

Open Networks

- An open network provides no encryption and no authentication of data
- WPA3 offers (opportunistic) encryption also for open networks
  - This does not prevent malicious nodes from participating in the network
  - But it ensures data privacy, i. e. recorded traffic cannot be decrypted by an attacker

Chapter 2: Data link layer

Summary

We should now have an understanding

- how networks can be represented as graphs,
- what the difference between a MST and a SPT is,
- how different methods for media access control work,
- how they handle or avoid collisions,
- why historically Ethernets were limited to 500 m,
- how nodes are addressed in local area networks,
- how MAC addresses look like and for what purpose they are used,
- how multiple local area networks can be combined in a larger one,
- what the differences between hubs, bridges, and switches are,
- how switches learn which host is attached to which port and how forwarding decisions are made,
- what collision and broadcast domains are,
- how wired and wireless networks can be connected on layer 2, and
- what the difficulties in protecting against attacks on layer 2 are.

Chapter 2: Data link layer

References

[1] ieee standard for information technology–telecommunications and information exchange between systems - local and metropolitan area networks–specific requirements - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications.
IEEE Std 802.11-2020 (Revision of IEEE Std 802.11-2016), pages 1–4379, 2021.

[2] E. Stein.
Taschenbuch Rechnernetze und Internet, chapter Konzepte: Lokale Netzwerke, pages 191–218.
Fachbuchverlag Leipzig, 2. edition, 2004.