Layer-2 Objectives

- Layer 2 responsibilities
- Layer 2 format
- Layer 2 operation
- Layer 2 Devices: Bridge operation

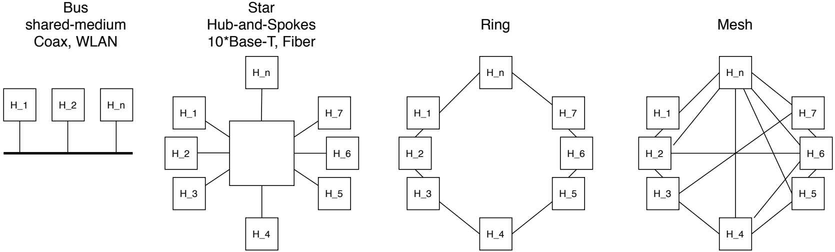
Layer-2: Übersicht

- Der *Data Link Layer* ist als *Sicherungsschicht* verantwortlich für die Verbindung zwischen zwei Knoten
- Die Bits von der Schicht 1 werden im Layer 2 zu *Frames* zusammengefasst und mit Zusatzinformationen für die Fehlererkennung ausgestattet
- Bei Ethernet/LAN erfolgt eine weitere Unterteilung der Schicht 2:
  - MAC: Media Access Control (Zugriff auf das Übertragungsmedium)
  - LLC: Logical Link Control (Sicherung)
- Es kann auch eine *Adressierung* erfolgen, d.h. gezielte Kommunikation zu spezifischen Geräten und oft auch "Broadcast"/Multicast[1]

[1] "an alle" und "an Gruppe"

Layer-2: Physikalische Struktur

Durch die *Adressierung* können mehr als zwei Geräte miteinander kommuniziern. Die Physikalische Struktur des Netzwerkes kann dabei folgende Formen annehmen[2]:



Heute üblich:

- "im Kleinen"[3]: Sternförmige Verkabelung, "Bus"/WLAN
- "im Grossen"[4]: Vermascht
- oft auch (hierarchische) Mischformen

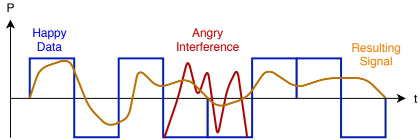
[2] und eine L2-Peerkommunikation ermöglichen

[3] Campus

[4] Internet, zwischen Provider

Layer-2: Fehlererkennung/-korrektur

Jede physikalische Übertragung ist Störungen unterworfen:



Das Zusammenfassen der einzelnen Bits in “Frames” bietet auch die Möglichkeit *Redundanz*[5] zu diesen Einheiten zuzufügen. Dadurch wird eine Fehlererkennung und eventuell eine Fehlerkorrektur möglich:

[5] “mehr Daten ohne mehr Informationsinhalt”

Kommunikationstechniken
Fehlererkennung/-Korrektur/-Vermeidung

ARQ : "automatic repeat reqeust" – eine Quittung (oder Timeout="keine Quittung") wird gesendet. In TCP/IP erst auf L4: TCP

FEC : "forward error correction": es werden *redundante* Daten gesendet, der Empfänger kann die Nachricht prüfen und je nach Verfahren korrigieren[6]. Im einfachsten Fall wird die Sendung $n > 2$ mal wiederholt

Hybrid : der Empfänger kann eine korruptes Frame neu anfordern[7]

[6] besonder für "Broadcast"-Kommunikation nützlich, wenn keine Quittung gesendet werden kann

[7] funktioniert z.B. bei Ethernet nicht, da die Adressierung auch im FEC eingeschlossen ist

Fehlererkennende Codes

Für die Erkennung von Übertragungsfehlern, werden verschiedene Methoden eingesetzt, bei denen zusätzliche Redundanz in die Daten eingefügt wird. Fehlererkennung ist immer ein Tradeoff zwischen zusätzlicher Redundanz und Auftretenswahrscheinlichkeit des Fehlers.

Paritätsbit(s): Pro Frame/Byte zusätzliche Bit(s), welche die Anzahl der "1" reflektieren[8] (odd/even).

Prüfsummen: z.B. CRC, Cyclic Redundancy Check. Es werden nach bestimmten Regeln Checksummen über die Daten gebildet und mitgesendet, auf der Empfangsseite wird die Prüfsumme erneut gebildet und mit der mitgesendeten verglichen.

"Robuste Codes": Die "Bit-Differenz" der Codewörter wird ausgenutzt, damit Fehler erkannt und korrigiert werden können. Ein 8-Bit Code mit nur 4 Code-Points:
00000000, 00001111, 11110000, 11111111,
Kann mit einem "Abstand" von mindestens 4 Bit definiert werden: (**Hammingdistanz**) damit werden bis zu $n-1$ Bit Fehler erkannt.

Folie : Degen

Beispiel für Paritätsbits

Folie : Degen

```
01001101 0
00001001 0
10001011 0
11101100 1
00110111 1
00010100 0
```

Korrekt

```
01001101 0
00001001 0
10000011 0
11101100 1
00110111 1
00010100 0
```

1 Bit Fehler

```
01001101 0
00011001 0
10001011 0
11100100 1
00110111 1
00010100 0
```

2 Bit Fehler

```
01001101 0
01001001 0
10001011 0
10101100 1
00110111 1
00010100 0
```

2 Bit Fehler

```
01001101 0
00001001 0
10001011 1
11101100 1
00110111 1
00010100 0
```

1 Bit Fehler

```
01101101 0
00001011 0
10001011 0
11101000 1
01110111 1
00010100 0
```

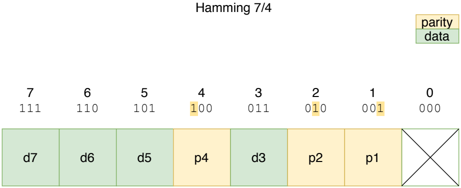
4 Bit Fehler

```
01001101 0
01001000 0
10001011 0
10101101 1
00110111 1
00010100 0
```

4 Bit Fehler

Hammingcodes

Hamming-Codes ermöglichen ein gutes Redundanz-Verhältnis[9] – d.h. möglichst wenig redundante Daten. Dies wird erreicht mit einem "Block"-Paritätsschema:



$$p_1 = d_3 \oplus d_5 \oplus d_7$$
$$p_3 = d_3 \oplus d_6 \oplus d_7$$
$$p_4 = d_5 \oplus d_6 \oplus d_7$$

https://en.wikipedia.org/wiki/Hamming_code

https://www.youtube.com/watch?v=b3NxrZOu_CE

https://www.youtube.com/watch?v=b3NxrZOu_CE

[9] ...das mit steigender Bitzahl/Blockgrösse besser wird

CRC – Cyclic Redundancy Check

CRC berechnet eine Prüfsumme *fester Länge* für beliebig lange Nachrichten[10]. Dazu wird ein *Generatorpolynom* binär mit der Nachricht verarbeitet[11], z.B:

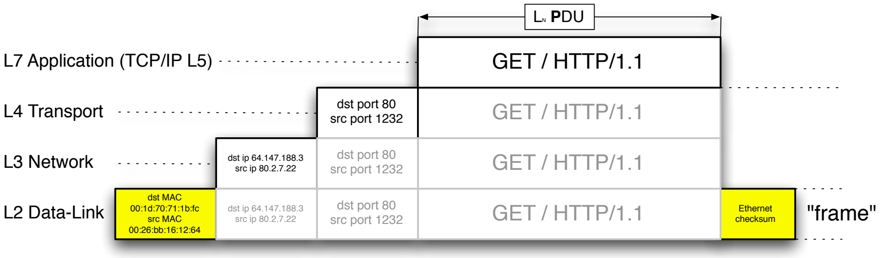
Generator $= x^3 + x + 1$ wird als Folge von 0 (kein Faktor) oder 1 (Faktor 1) codiert: 1011

```
11010011101100 000 <--- input right padded by 3 bits
1011               <--- divisor
01100011101100 000 <--- result (note the first four bits are the XOR with the divisor beneath
 1011              <--- divisor ... , the rest of the bits are unchanged)
00111011101100 000
  1011
00010111101100 000
   1011
00000001101100 000 <--- note that the divisor moves over to align with the next 1 in the dividend
       1011             (in other words, it doesn't necessarily move one bit per iteration)
00000000110100 000
        1011
00000000011000 000
         1011
00000000001110 000
          1011
00000000000101 000
           101 1
------------------
00000000000000 100 <--- remainder (3 bits).  Division algorithm stops here as dividend is equal to zero.
```

Beispiel von https://en.wikipedia.org/wiki/Cyclic_redundancy_check geklaut

[10] "undetected error rate" wird aber schlechter bei sehr grossen Verhältnissen. Ethernet 1500 Bytes -¿ 32 Bit CRC

Layer-2: Stack (Ethernet)



L2 Responsibilities (Ethernet)

- *framing*/ "packaging" of data for transport over links (ie, between *adjacent* nodes/LAN[12])
- implementation of device *addresses* in LAN[13]
- Ethernet/IEEE-802.3 allows for multicast- and broadcast destination[14]
- *Error detection* using a 32-bit CRC[15]
- Ethernet L2 does *not* assure delivery[16] (ie. no acknowledges sent, no attempt to retransmit)

[12] Local Area Network: typical in-house network connected to the Internet via a Router. WAN/Wide Area Network consist of many LANs → Internet

[13] ie, "local"

[14] message to some or all nodes on LAN

[15] err'd frames are simply dropped by bridges, routers, hosts. Ponder about the reason for this...

[16] ie, the layers above must handle lost messages

L2 Factlets

- messages on a Ethernet LAN are called *frames*
- most abundant LANs/L2-Networks today are 802.3/Ethernet and 802.11/Wireless
- devices for *building* LANs: L1:Hub/Repeater and L2:Bridge/Switch
- devices *interconnecting* LANs to other LANs or the "outside world": L3:Router or L3+:Firewall/Router
- L2 addressing is of *local* [17] interest only!
- a Link/L1 forms a "collision domain", transmissions from different devices may "collide" on a single wire/Hub
- a LAN/L2 denotes a "broadcast domain": `0xFF:FF:FF:FF:FF:FF` destination is sent to all nodes on the LAN[18]
- 802.x/Ethernet is a TDM[19] network

---

[17] there is no need for your computer to know the L2 address of a webserver in the Internet

[18] it is *limited*, ie it never leaves the LAN via a router

[19] Time Domain Multiplexing

L2 Frame-Header/Metadata

encapsulates – "frames" – a certain[20] amount of data[21] from above layer with metadata:

- **Preamble**: a special synchronize sequence[22]
- **Address**: destination- and source-address of adjacent nodes
- **Type**: identifies encapsulated data (type of SDU/upper-layer), eg `0x0800` for IP
- **Frame Checksum**[23]: allows the destination node to check consistency of data received

[20] on Ethernet maximum 1518 Bytes - layer-2 metadata, minimum 64 Bytes

[21] the "payload" from Layer-3, this is the "SDU" service-data-unit on Layer-2

[22] http://en.wikipedia.org/wiki/Ethernet_frame

[23] CRC32

L2 Addressing (1/2)

- Ethernet L2/MAC addresses consists of 6 Bytes (3 vendor-id[24], 3 serial)
- this allows for (theoretical) $2^{48} \sim 256$ trillion addresses
- the usual notation for MAC addresses are hex[25] bytes seperated by ":"
- MAC adresses are guaranteed[26] to be unique
- `0xFF:FF:FF:FF:FF:FF` is the *broadcast* [27] destination address
- any address with the `0x_1:__:__:__:__:__` bit set is *multicast* [28]

[24] https://db.uga.edu/network/public/vendorcode.cgi

[25] sometimes identified by 0x-prefix

[26] theoretically... most OS/network cards allows you to alter this address and sometimes the vendor just blows it

[27] "to all", limited to the LAN of course

[28] eg. to "all routers" in LAN

L2 Addressing (2/2)

- MAC/L2-Addresses exhibit no "grouping" network-coherence/pattern

```
IP/L3                         MAC/L2
-------------                 -----------------
192.168.1.247 dev eth0 lladdr 58:9c:fc:0d:09:dc REACHABLE
192.168.1.29  dev eth0 lladdr 00:0c:42:e9:25:57 PROBE
192.168.1.10  dev eth0 lladdr c8:2a:14:55:aa:e3 REACHABLE
192.168.1.87  dev eth0 lladdr 3c:2a:f4:eb:f0:fc REACHABLE
192.168.1.2   dev eth0 lladdr b8:69:f4:c5:3c:97 REACHABLE
192.168.1.16  dev eth0 lladdr 00:22:15:dd:59:16 REACHABLE
192.168.1.23  dev eth0 lladdr 00:10:6c:05:15:fe REACHABLE
192.168.1.26  dev eth0 lladdr 00:15:5d:01:16:01 REACHABLE
192.168.1.32  dev eth0 lladdr 10:40:f3:97:14:e4 REACHABLE
192.168.1.1   dev eth0 lladdr d4:ca:6d:f8:6e:7e REACHABLE
192.168.1.254 dev eth0 lladdr 00:21:cc:ca:d3:e8 REACHABLE
192.168.1.4   dev eth0 lladdr 00:0c:42:e9:25:57 REACHABLE
192.168.1.14  dev eth0 lladdr 00:08:9b:c1:17:fd REACHABLE
192.168.1.18  dev eth0 lladdr 00:08:9b:8c:c0:72 REACHABLE
192.168.1.25  dev eth0 lladdr 00:0c:42:e9:25:57 REACHABLE
192.168.1.79  dev eth0 lladdr 80:1f:02:51:d0:79 REACHABLE
192.168.1.83  dev eth0 lladdr 24:77:03:a3:29:0c REACHABLE
192.168.1.20  dev eth0 lladdr 08:00:27:5e:78:6c REACHABLE
192.168.1.22  dev eth0 lladdr d8:9e:f3:7c:dd:f5 REACHABLE
fe80::d6ca:6dff:fef8:6e7e dev eth0 lladdr d4:ca:6d:f8:6e:7e router STALE
```

- the address is specific to the device and not the location[29]

MAC-addresses are LAN/*local*-only

[29] ie, the address stays the same: on campus or at home

L2 Interlude

- find your computers MAC address[30]
- find the vendor of your computers NIC[31]
- find other MACs your computer had conversation with[32]
- find the vendor of the router[33] connecting you to the internet[34]
- find the MAC of your neighbours PC[35]
- find the MAC of `www.eff.org`
- listen to the network chit-chat using `tcpdump` (on `netbox`). Try to identify L2-broadcast, multicast and unicast

---

[30] UNIX: `ifconfig`, Microsoft Windows: `ipconfig /all`

[31] Network Interface Card

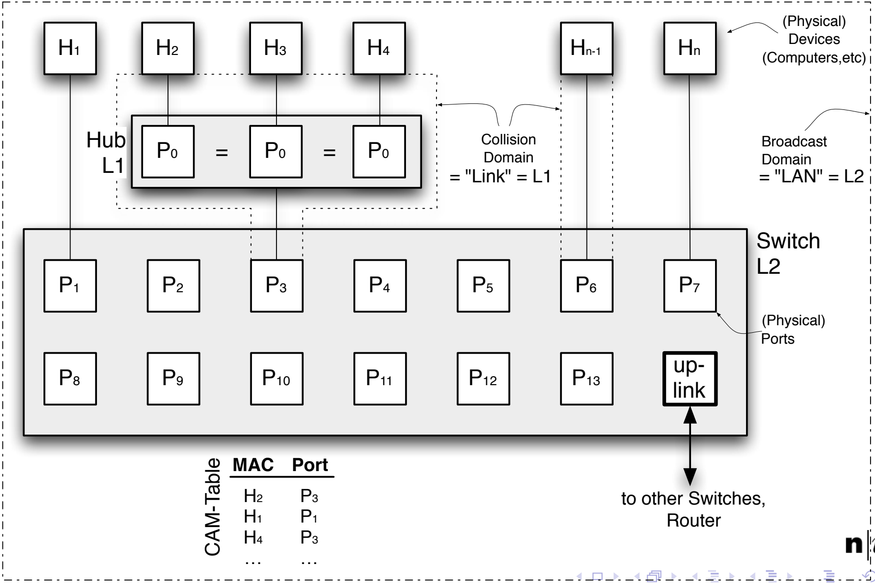
[32] `arp -a`, add another `-n` on UNIX for faster responses

[33] "default gateway"

[34] this is actually a L3 theme. . . use `netstat -rn` to find the routers IP and locate the corresponding MAC in the `arp -a` output

[35] use `ping` *IP* first then issue `arp -a` once again

L2 Bridging 1/2



L2 Bridging 2/2

- *bridges* are devices to extend the reach of a LAN. The resulting network is still a single LAN
- multiport[36] bridges are called (L3) *switches*
- bridges analyze the destination address of a frame and transmit it only on specific port(s)
  - ...thus providing some "privacy"[37]
  - this is achieved by building a MAC-address/port lookup table by storing the *source* MAC-address along with the receiving port number
- as long as a particular destination MAC-address is not known, frames must be *flooded* out to all except the receiving port
- broadcast frames are send out on all ports except on the receiving one

[36] anything with more than a few ports

[37] try yourself: use `wireshark` or `tcpdump` and see if you can spy on your neighbous traffic

L2 CSMA/CD, Collision-Domain

- CSMA: Carrier Sense Multiple Access/Collision Detection
- since the cable/medium[38] allows for at a single transmission only at any given time (TDM), the sender constantly monitors its transmission and cancels it in case of noise: *collision detection*
- such a L1-segment[39] is called a “collision-domain”
- bridged seperates “collision-domains”, thus a end-device connected to a switch has its private collision-domain[40]
- **today there are no longer collisions on wired networks**[41], in WLAN CSMA still applies, though

[38] in case of twisted-pair cables the send/receive lines are physically seperated allowing for full-duplex traffic. Traditional coax-cables are half-duplex only

[39] single broadcast-medium cable (coax) or repeater/hub interconnected

[40] and will never encounter collisions at all if configured correctly

[41] assuming all cabling is centralized in switches

L2 Bridging: Cut-Through vs Store-and-Forward

- traditionally bridges/switches receives a whole frame and forwards it if the frame-checksum matches
- this adds a certain *latency* [42] to the transmission
- some bridges/switches offer a *cut-through* forwarding mode, where the frame is forwarded as soon as the destination-address is received
- this mode allows for a *constant* and minimal latency
- in case of line-noise, the bridge may forward defective frames in cut-through mode
- advanced bridges mitigate this problem by fall-back to store-and-forward mode in presence of errors

[42] a delay, in this case dependent of the frame-length

L2 Briding: Loops and avoidance of

- complex LANs with multiple bridges may form *loops* [43]
- especially broadcast frames may lead to a (broadcast) *storm*
- advanced bridges employ a *spanning-tree* [44] protocol to avoid this

[43] try this at home: "short-circuit" your (auto-crossover) switch by connecting a cable back-to-back

[44] IEE 802.3D STP Spanning Tree Protocol: an application of the Djikstra-Algorithm, we'll study this in L3 OSPF

L2 Bridging: VLAN

- advanced bridges allow for *Virtual LANs* (VLANs)
- VLANs are seperated LAN/L2-segments[45]
- the L2 metadata is extended by a VLAN-identification number
- a physical port on the bridge can be configured to allow for one VLAN only[46] – usually to connect to end-devices
- physical ports may also be configured to operate in *trunking* mode – usually in bridge-to-bridge *aggregated* link or to allow for advanced end-devices to seperate VLANs internally
- typical applications: seperate external-, internal- and server-LAN for security reasons[47]

[45] ie, a router is required to interconnect VLANs

[46] the VLAN-id is *stripped*†from the metadata

[47] this is considered bad practice

L2: References for ND03

- http://en.wikipedia.org/wiki/Ethernet_frame, http://en.wikipedia.org/wiki/Ethernet_II_framing
- http://en.wikipedia.org/wiki/802.3
- http://en.wikipedia.org/wiki/IEEE_802.1D
- http://en.wikipedia.org/wiki/Bridging_(networking) especially the part "bridging makes no assumptions about where in the network a particular address is located" → "flooding"
- http://en.wikipedia.org/wiki/Frame_(networking)
- https://db.uga.edu/network/public/vendorcode.cgi, MAC vendor