MAC Addresses

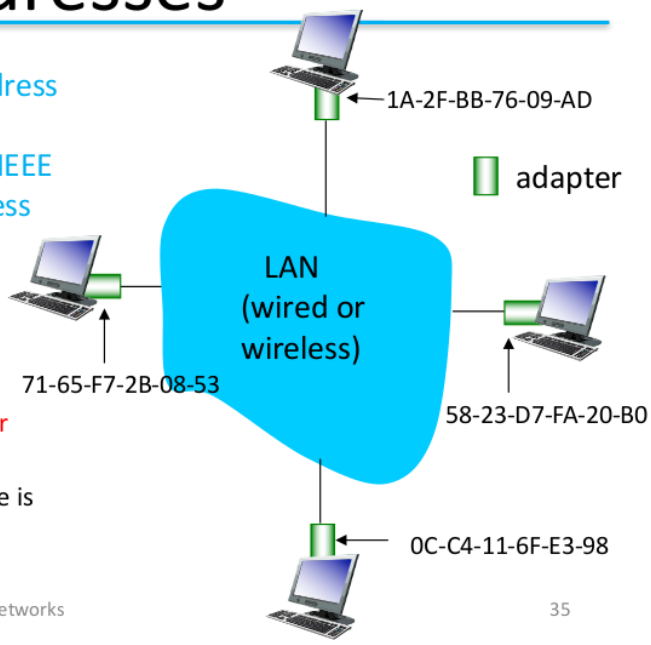
- 32/128-bit IP address
  - Network-layer address for interface
  - Used for layer 3 (network layer) forwarding
- MAC (or LAN or physical or Ethernet) address
  - Function: used "locally" to get frame from one interface to another physically-connected interface (same network, in IP-addressing sense)
  - 48 bit MAC address (for most LANs) burned in NIC ROM, also sometimes software settable
  - E.g.: 1A-2F-BB-76-09-AD

hexadecimal (base 16) notation
(each "numeral" represents 4 bits)

Layer 4 - Port ↔ Prozess
Layer 3 - IP ↔ Destination Layer -- 3 findet Fehlgesendete Packete
Layer 2 - Mac ↔ Frame-routing in Subnets

LAN Addresses

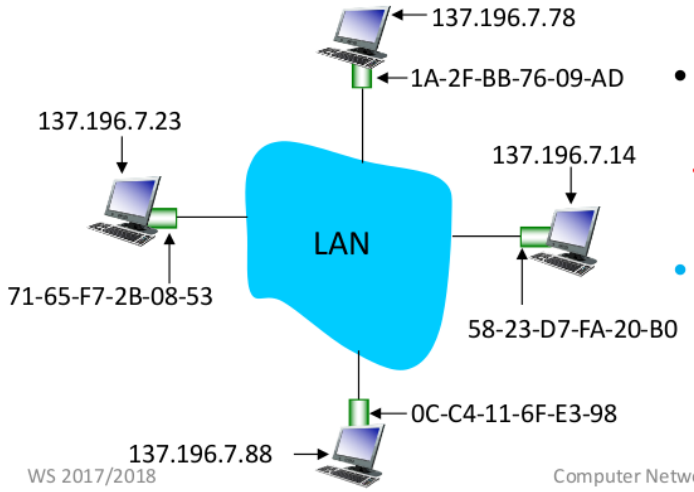
- Each adapter on LAN has unique LAN address
- MAC address allocation administered by IEEE
- Manufacturer buys portion of MAC address space (to assure uniqueness)
- Analogy
  - MAC address: like Social Security Number
  - IP address: like postal address
- MAC flat address ➜ portability
  - Can move LAN card from one LAN to another
- IP hierarchical address not portable
  - Address depends on IP subnet to which node is attached



Eindeutige Lan-Addresse, bestellbar bei IEEE
→ IP/MacAddressen veränderbar ↔ Bei Gleichsetzung mit Nachbar = Fehler

ARP: Address Resolution Protocol

Question: how to determine interface's MAC address, knowing its IP address?



- ARP table: each IP node (host, router) on LAN has table
- IP/MAC address mappings for some LAN nodes:
  < IP address; MAC address; TTL>
- TTL (Time To Live): time after which address mapping will be forgotten (typically 20 min)

-

ARP: Within same LAN

- A wants to send datagram to B
  - B's MAC address not in A's ARP table
- A broadcasts ARP query packet, containing B's IP address
  - Destination MAC address = FF-FF-FF-FF-FF-FF
  - All nodes on LAN receive ARP query
- B receives ARP packet, replies to A with its (B's) MAC address
  - Frame sent to A's MAC address (unicast)

- A caches (saves) IP-to-MAC address pair in its ARP table until information becomes old (times out)
  - Soft state: information that times out (goes away) unless refreshed
- ARP is "plug-and-play":
  - Nodes create their ARP tables without intervention from network administrator

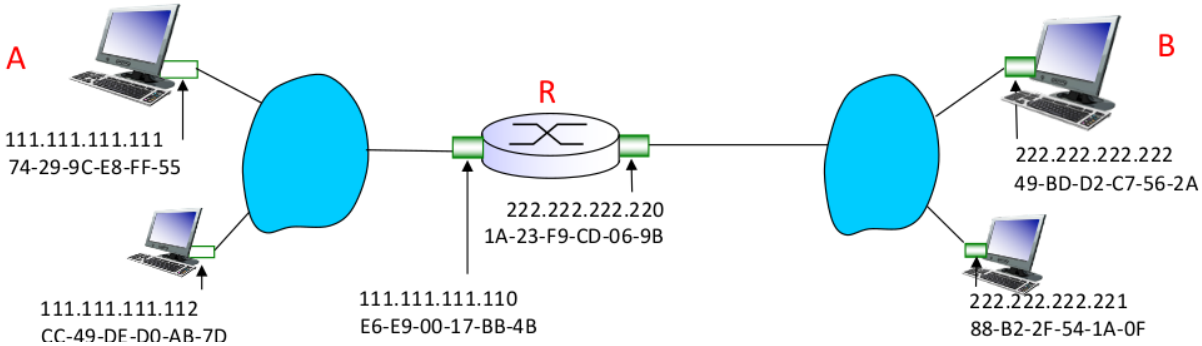
Query-Packet: Bis Schicht 3
Abhängig von IP-Addressen wird im Subnetz broadcasted

-

ARP: Routing to another LAN

Walkthrough: send datagram from A to B via R

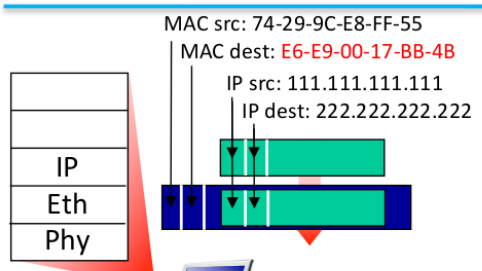
- Focus on addressing – at IP (datagram) and MAC layer (frame)
- Assume A knows B's IP address
- Assume A knows IP address of first hop router, R (how?)
- Assume A knows R's MAC address (how?)



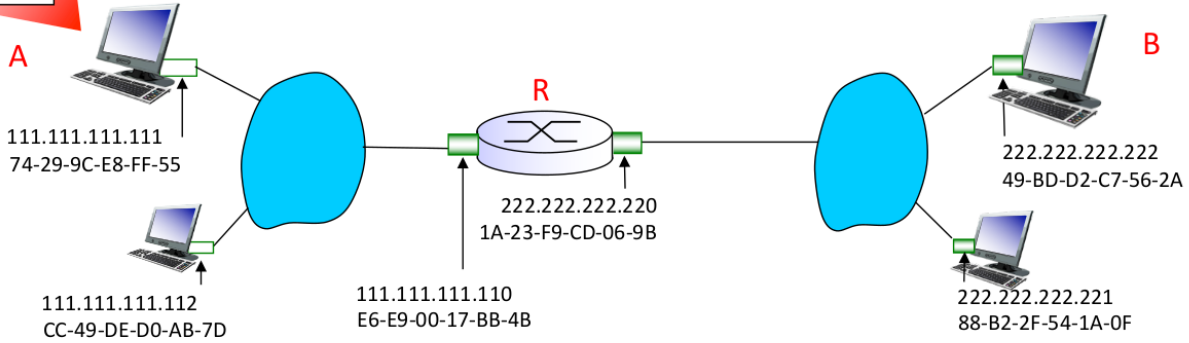
2. DNS
3. THCP
4. Broadcast

-

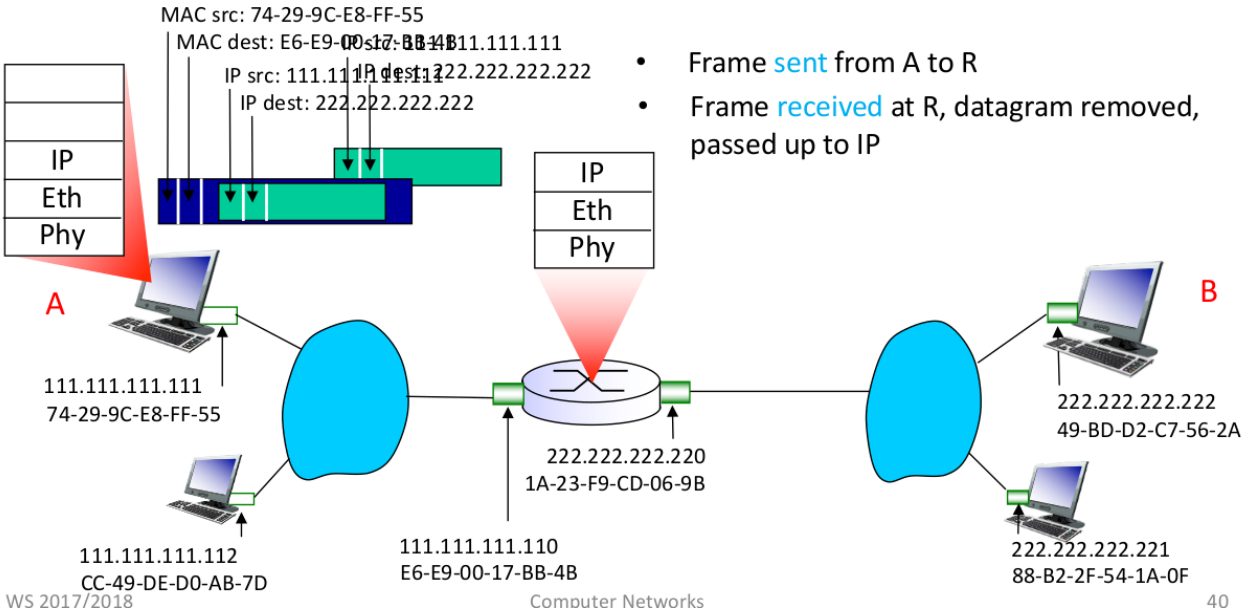
ARP: Routing to another LAN



- A creates IP datagram with IP source A, destination B
- A creates link-layer frame with R's MAC address as destination address, frame contains A-to-B IP datagram

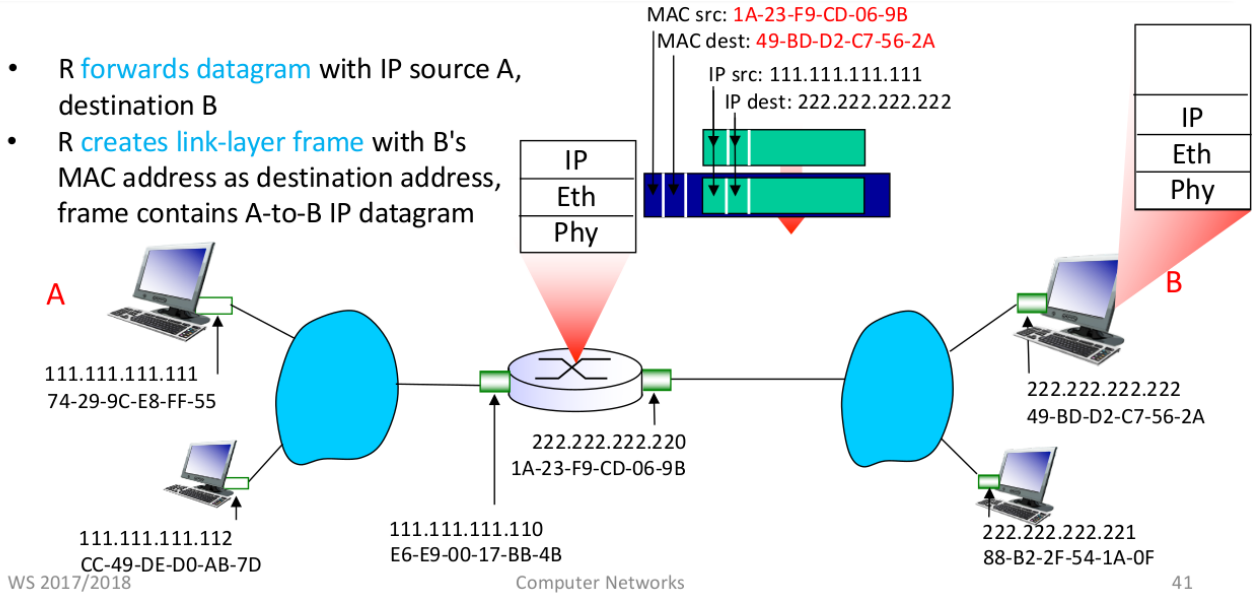


-



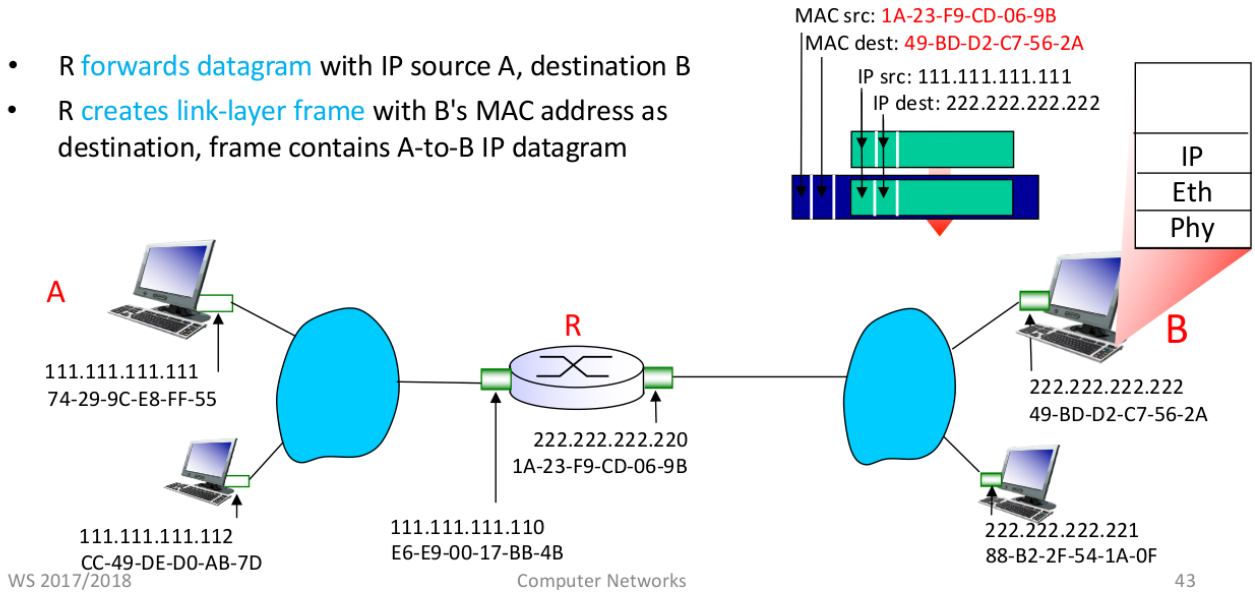
- Frame sent from A to R
- Frame received at R, datagram removed, passed up to IP

-



- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as destination address, frame contains A-to-B IP datagram

-

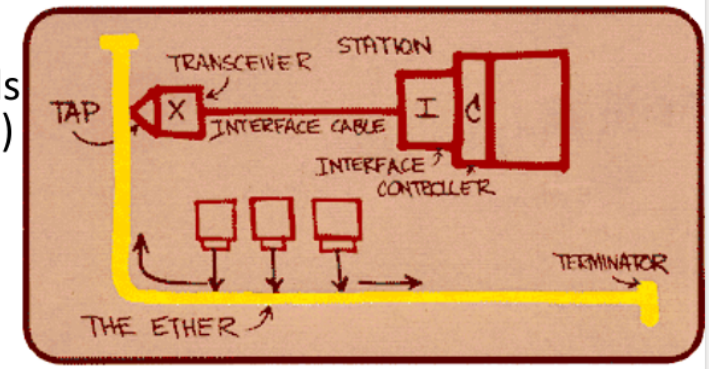


- R forwards datagram with IP source A, destination B
- R creates link-layer frame with B's MAC address as destination, frame contains A-to-B IP datagram

-

Ethernet

- "Dominant" wired LAN technology
- Single chip, multiple speeds (e.g., Broadcom BCM5761)
- First widely used LAN technology
- Simpler, cheap
- Kept up with speed race: 10 Mbps – 10 Gbps



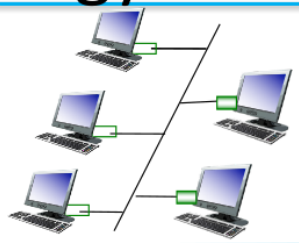
Metcalfe's Ethernet sketch

-

Ethernet: Physical Topology

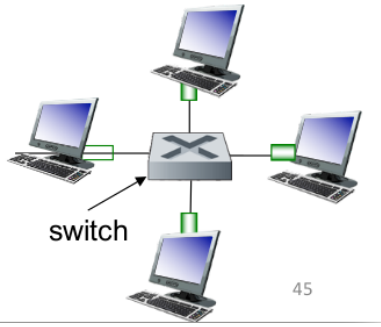
past

- Bus: popular through mid 90s
  - All nodes in same collision domain (can collide with each other)



today

- Star: prevails today
  - Active switch in center
  - Each "spoke" runs a (separate) Ethernet protocol (nodes do not collide with each other)



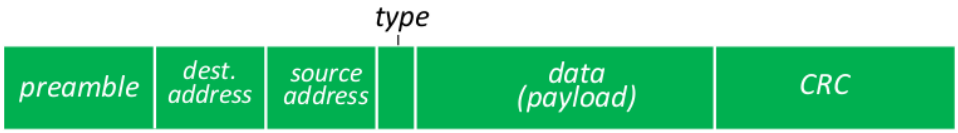
future?

Bustopologie existiert großteils in eingebetteten Systemen.
Future: Ad-hoc: Jeder kommuniziert direkt
Infrastrukturbetreiber haften teils für ihre Nutzer in öffentlichen Netzwerken

-

Ethernet Frame Structure

- Sending adapter encapsulates IP datagram (or other network layer protocol packet) in Ethernet frame



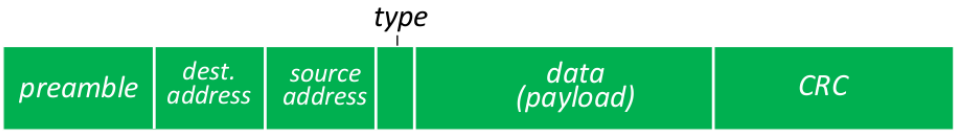
- Preamble
  - 7 bytes with pattern 10101010 followed by one byte with pattern 10101011
  - Used to synchronize receiver, sender clock rates

Carrier-Sense ↔ Identifiziert 7-Byte Pattern

-

Ethernet Frame Structure

- Addresses: 6 byte source, destination MAC addresses
  - If adapter receives frame with matching destination address, or with broadcast address (e.g., ARP packet), it passes data in frame to network layer protocol
  - Otherwise, adapter discards frame (except if in promiscuous mode)
- Type: indicates higher layer protocol (mostly IP but others possible, e.g., Novell IPX, AppleTalk)
- CRC: cyclic redundancy check at receiver
  - Error detected: frame is dropped



Payload ist unverschlüsselt ↔ Es ist jederzeit vorsicht geboten
Type = Protokolnummer/Port in anderen Protokollen

-

Ethernet: Unreliable, Connectionless

- Connectionless
  - No handshaking between sending and receiving NICs
- Unreliable
  - Receiving NIC doesn't send acks or nacks to sending NIC
  - Data in dropped frames recovered only if initial sender uses higher layer rdt (e.g., TCP), otherwise dropped data lost
- Ethernet's MAC protocol: unslotted CSMA/CD with binary backoff

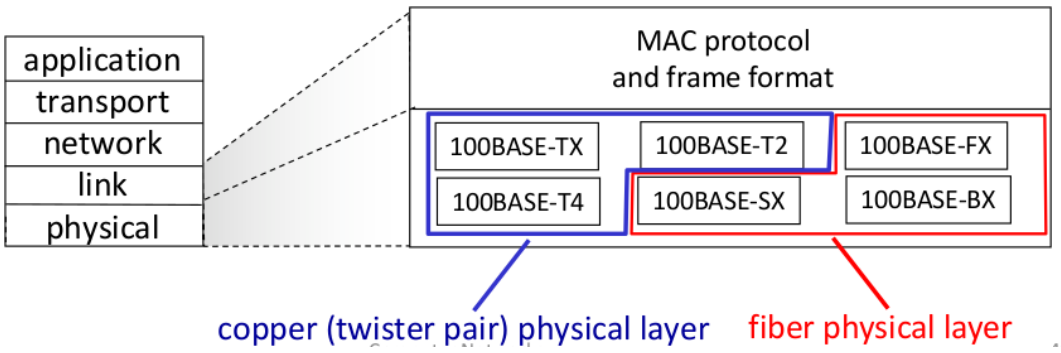
CSMA: Carrier Sense Multiple Access/Collision Detection
Backoff Algorithm: Bei Kollission wird gewartet bis zum wiedersenden (Exponentielle Wartezeit).

-

802.3 Ethernet Standards: Link & Physical Layers

Many different Ethernet standards

- Common MAC protocol and frame format
- Different speeds: 2 Mbps, 10 Mbps, 100 Mbps, 1Gbps, 10 Gbps, 40 Gbps
- Different physical layer media: fiber, cable



-

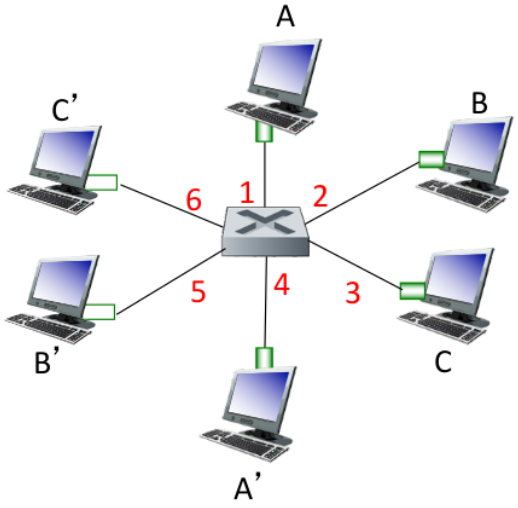
Ethernet Switch

- Link-layer device: takes an active role
  - Store, forward Ethernet frames
  - Examine incoming frame's MAC address, selectively forward frame to one-or-more outgoing links when frame is to be forwarded on segment, uses CSMA/CD to access segment
- Transparent
  - Hosts are unaware of presence of switches
- Plug-and-play, self-learning
  - Switches do not need to be configured

(Siehe unteres Bild: Switchlichter)
→Erstere Versionen hatten Packetvermittlung an alle Ports
→Heute werden MacAddressen ermittelt und dementsprechend weitergeleitet.
→Transparenz: Rechner bekommt von der Arbeit des Switches nichts mit.
-

Switch: Multiple Simultaneous Transmissions

- Hosts have dedicated, direct connection to switch
- Switches buffer packets
- Ethernet protocol used on each incoming link, but no collisions; full duplex
  - Each link is its own collision domain
- Switching: A-to-A' and B-to-B' can transmit simultaneously, without collisions



switch with six interfaces (1,2,3,4,5,6)

Switches buffern Pakete & Broadcasten bei Paketempfang.
-

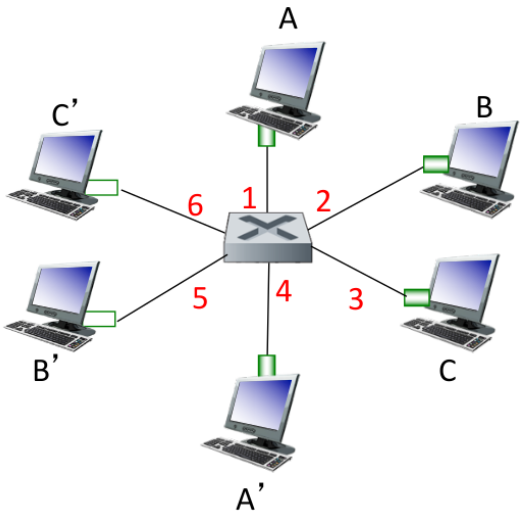
Q: how does switch know A' reachable via interface 4, B' reachable via interface 5?

A: each switch has a switch table, each entry:

- (MAC address of host, interface to reach host, time stamp)
- Looks like a routing table!

Q: how are entries created, maintained in switch table?

- Something like a routing protocol?



switch with six interfaces (1,2,3,4,5,6)

Switch: Frame Filtering/Forwarding

When frame received at switch:

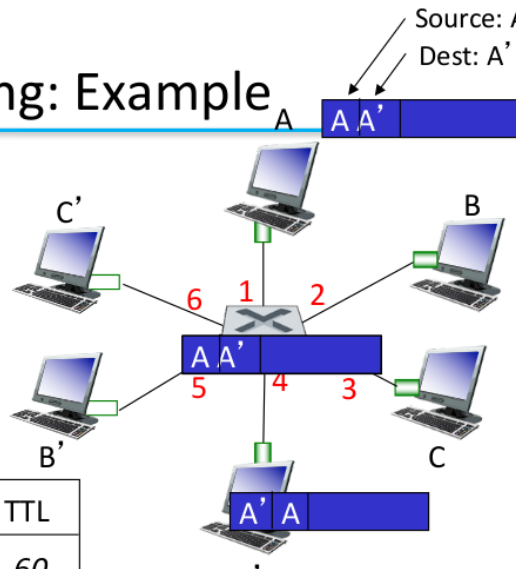
1. Record incoming link, MAC address of sending host
2. Index switch table using MAC destination address
3. if entry found for destination then

```
{
    if destination on segment from which frame arrived then drop frame
    else forward frame on interface indicated by entry
}
else flood  /* forward on all interfaces except arriving interface */
```

Self-Learning, Forwarding: Example

Frame destination, A', location unknown: *flood*

Destination A location known: selectively send on just one link



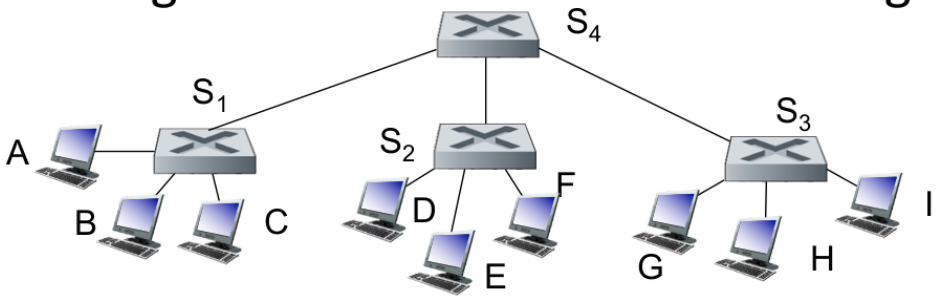
MAC addr	interface	TTL
A	*1*	*60*
A'	*4*	*60*

switch table (initially empty)

Floodet an alle außer an A selbst
→ A' beantwortet dem Switch
→ Andere Interfaces ignorieren.
-

Interconnecting switches

- Self-learning switches can be connected together:

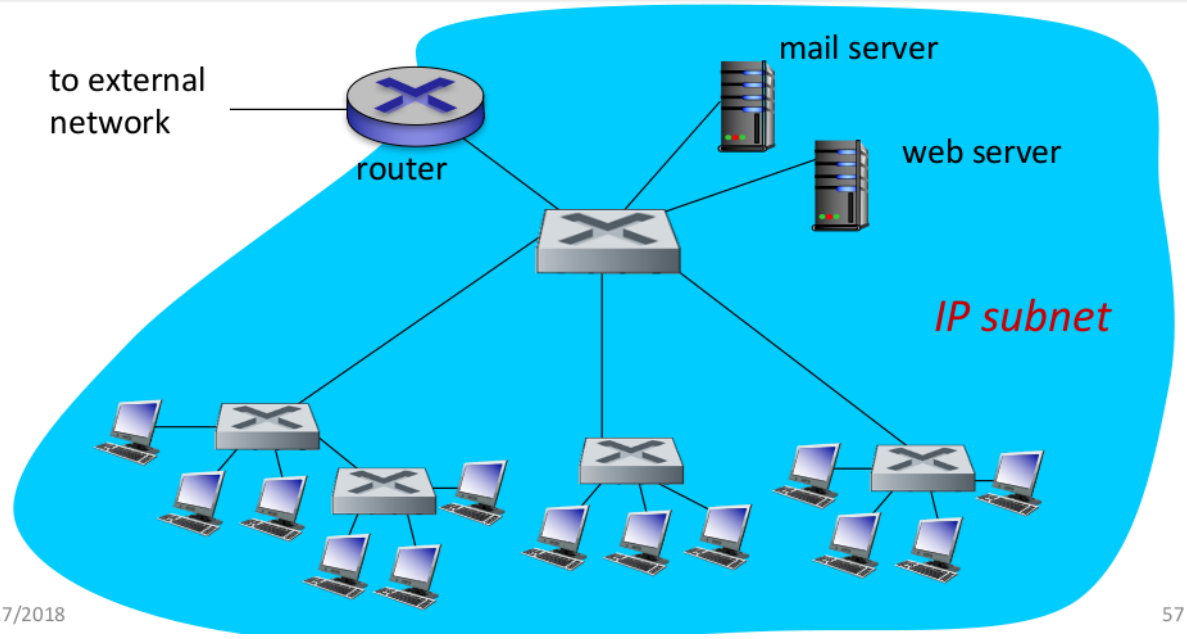


Q: sending from A to G - how does S_1 know to forward frame destined to G via S_4 and S_3?

A: self learning! (works exactly the same as in single-switch case!)

A sendet → S1 Floodet → S4 Floodet → S3 & S2 Floodet → G antworet.
– Alle Subinterfaces bekommen diese Nachfrage mit.
-

Institutional Network



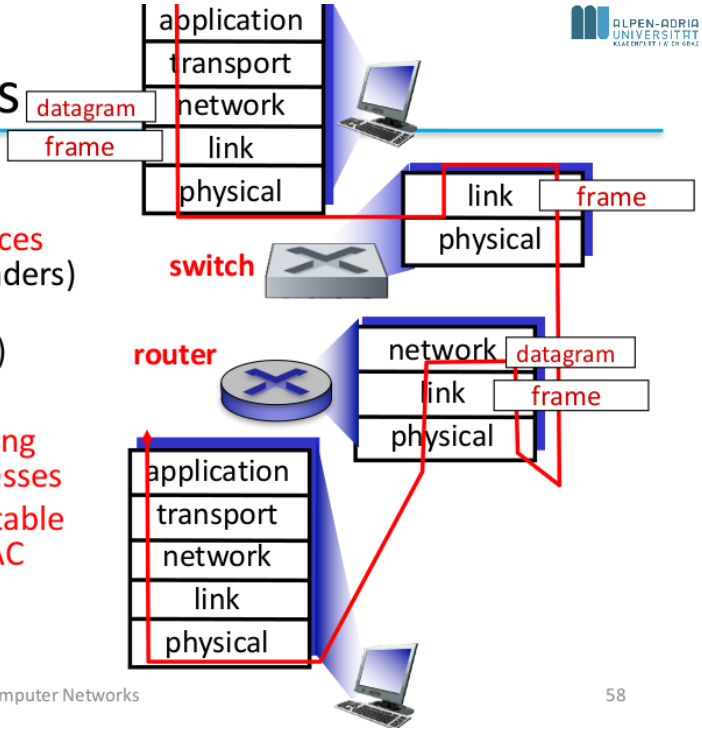
-

BITMOVIN

ALPEN-ADRIA UNIVERSITÄT KLAGENFURT | GRAZ

Switches vs. Routers

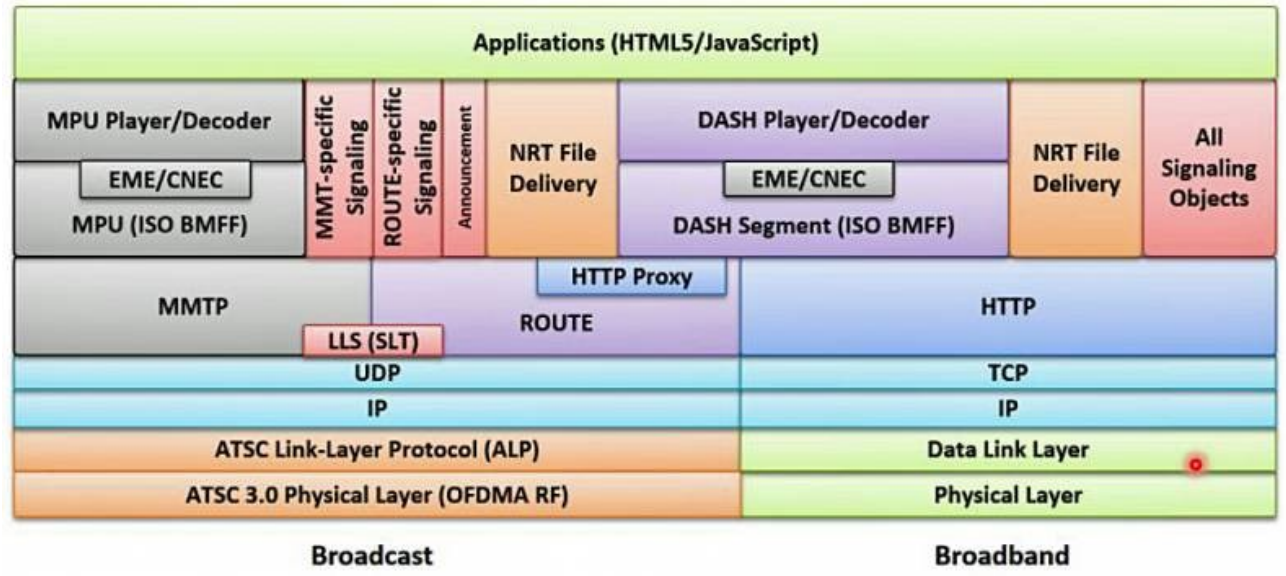
- Both are store-and-forward:
  - Routers: network-layer devices (examine network-layer headers)
  - Switches: link-layer devices (examine link-layer headers)
- Both have forwarding tables:
  - Routers: compute tables using routing algorithms, IP addresses
  - Switches: learn forwarding table using flooding, learning, MAC addresses



→ Router betrachtet Payload
→ Switch betrachtet Frame

→ Router: Routingalgorithmen
→ Switch: Routing via Flooding

-



ATSC A/331:2017 – Signaling Delivery, Synchronization and Error Protection
ATSC 3.0 – Nicht Klausurrelevant
→ Route/Mpeg Media Transport Protokoll: Realtime-Object-Protokole
-