

Betriebssysteme und Netzwerke

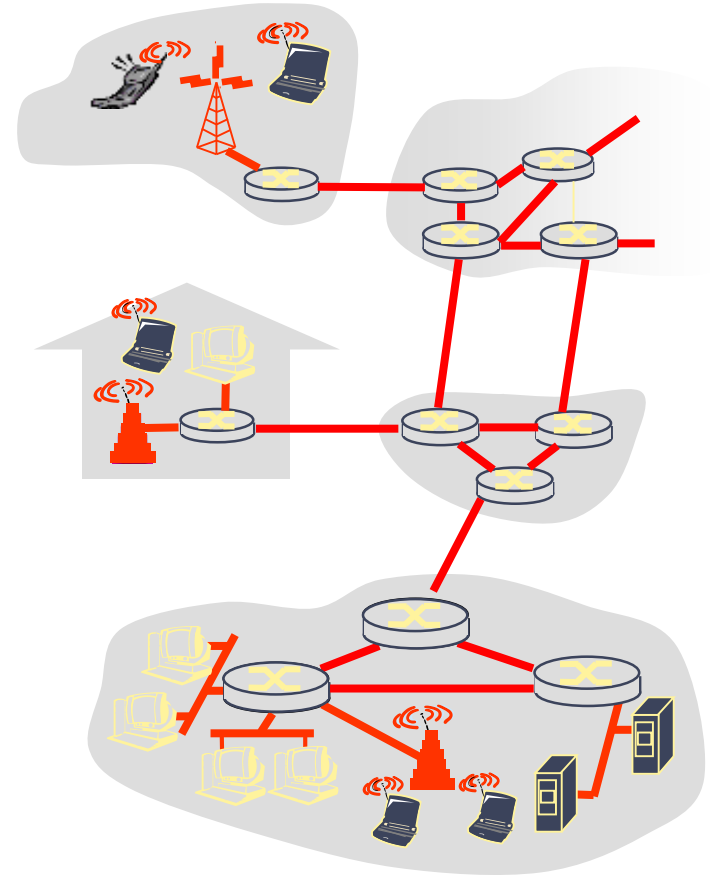
Vorlesung N08

Artur Andrzejak

Sicherungsschicht: Einführung

Sicherungsschicht: Begriffe

- ▶ Zur Vereinfachung behandeln wir Hosts und Router gleich und bezeichnen sie als **Knoten (nodes)**
- ▶ Die Kommunikationskanäle zwischen zwei (benachbarten) Knoten nennen wir **Links** (seltener als **Leitungen**)
- ▶ Ein Paket der Sicherungsschicht nennt man einen **Rahmen (frame)**
 - ▶ Rahmen kapseln Datagramme (= Pakete der Netzwerkschicht) ein



Sicherungsschicht (S-Schicht) ist dafür zuständig, ein Datagramm nur zwischen zwei benachbarten Knoten über ein Link zu übermitteln

Erinnerung: Internet-Protokollstapel

- ▶ Wie identifiziert man die Schicht?
 - ▶ Ein gutes Kriterium sind die Endpunkte der Kommunikation

Name	Endpunkte
Anwendungsschicht (application layer)	Mehrere <u>Programme</u> , jedes kann aus einem oder mehr Prozessen bestehen (z.B. Browser + Server)
Transportschicht (transport layer)	<u>Prozesse</u> (bzw. zugehörige Sockets – APIs des BS) auf einem oder mehreren Hosts
Netzwerkschicht (network layer)	Start- und End-Hosts (es wird <u>nicht</u> zwischen Prozessen auf einem Host unterschieden)
Sicherungsschicht (data link layer)	Zwei <u>direkt</u> verbundene Geräte (Host, Router, Switch) an den beiden Enden einer Teilstrecke
Bitübertragungss. (physical layer)	Elektronik der Leitungen oder Glasfaser an den Enden einer Teilstrecke

Dienste der Sicherungsschicht

- ▶ **Erzeugen von Rahmen (framing)**
 - ▶ Ein S-schichtprotokoll verkapselt alle Datagramme vor dem Transfer über einen Link in einem **Rahmen**
- ▶ **Verlässliche Übertragung** zwischen benachbarten Knoten (durch Wiederholungen der Übertragung)
 - ▶ Selten verwendet bei „fehlerfreien“ Medien (Glasfaser, manche Kabel)
 - ▶ Wichtig bei Links mit hoher Fehlerrate, z.B. Mobilfunkverbindung
- ▶ **Medienzugriff (link access)**
 - ▶ Ein **Medienzugriffsprotokoll** (**MAC**-Protokoll, **media access control protocol**) legt die Regeln fest, mit denen ein Rahmen über einen physischen Link übertragen wird

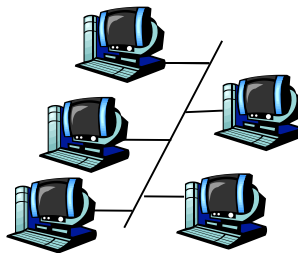
Einschub: Typen von Links

▶ **Punkt-zu-Punkt Links (point-to-point links)**

- ▶ D.h. 1 Sender, 1 Link, 1 Empfänger
- ▶ MAC-Protokoll ist dann einfach, z.B. PPP für Einwahlmodems

▶ **Broadcast-Übertragungskanal**

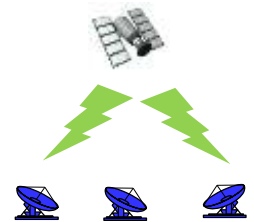
- ▶ Es können sich mehrere Knoten einen einzelnen Kanal teilen
=> **Mehrfachzugriff** auf das Medium möglich
 - ▶ Ethernet
 - ▶ 802.11-Protokolle



Gem. Kabel (z.B.
Kabel-Ethernet)



Radio als Medium
(z.B. 802.11 WiFi)



Radio als Medium
(Satellit)

Dienste der Sicherungsschicht /2

▶ **Flusskontrolle**

- ▶ Wichtig, da Knoten auf jeder Seite eines Links nur begrenzte Pufferkapazität für Rahmen haben

▶ **Fehlererkennung**

- ▶ Ein sehr häufiger Dienst von S-Schichtprotokollen, da es sinnlos ist, fehlerhafte Pakete weiterzuleiten

▶ **Fehlerkorrektur**

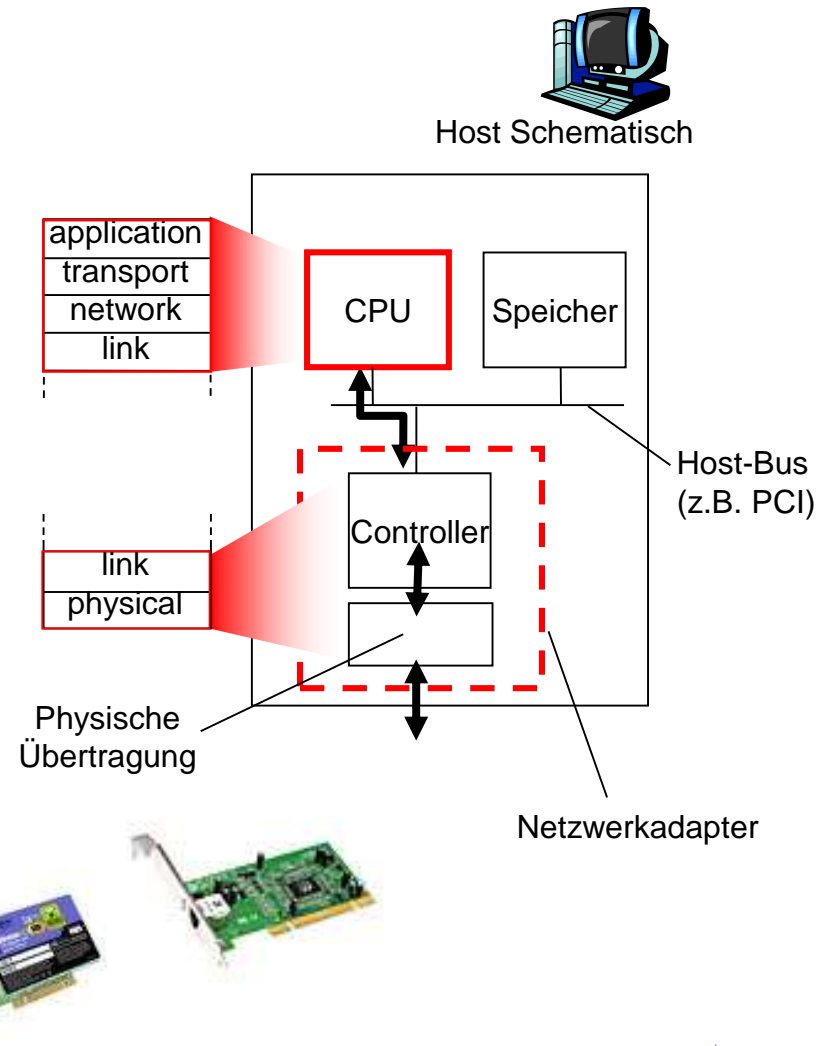
- ▶ Identifizieren und korrigieren von Bitfehler ohne wiederholte Übertragung

▶ **Halbduplex und Vollduplex Betrieb** (Eigenschaft)

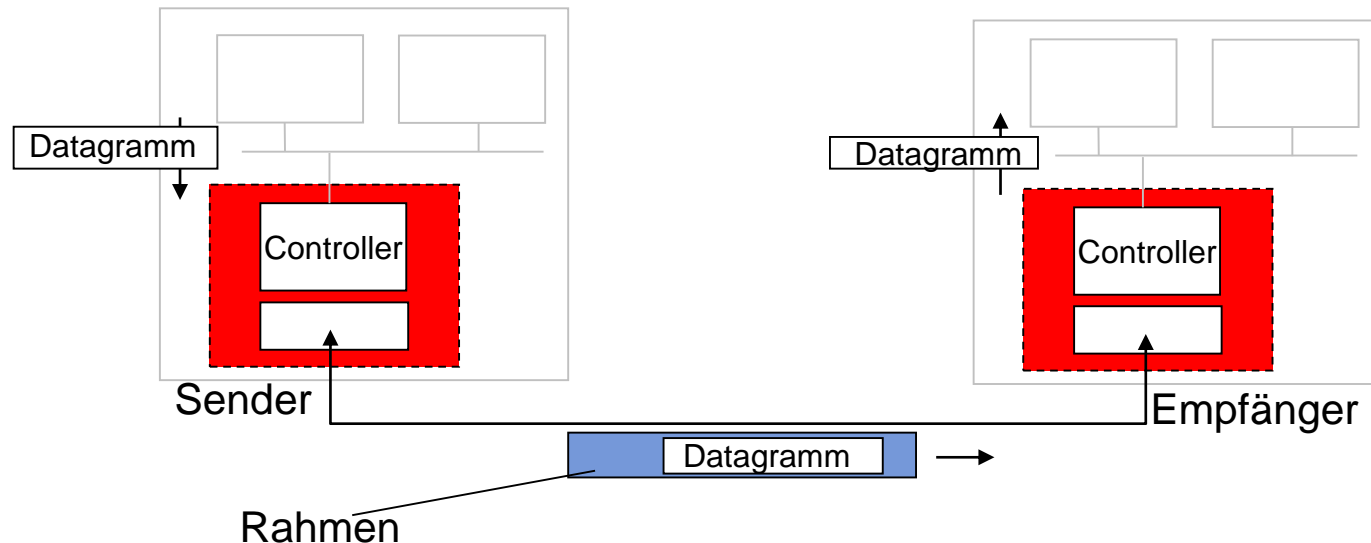
- ▶ Bei einer Halbduplex-Übertragung kann ein Knoten nicht gleichzeitig senden und empfangen, bei Vollduplex schon

Wo ist die S-Schicht implementiert?

- ▶ In jedem Host, Router und Switch
- ▶ Hosts: Im **Netzwerkadapter** (**network interface card, NIC**)
 - ▶ Ethernet-Karte, PCMCIA-Karte, 802.11-Karte (bzw. Chips auf dem Mainboard)
 - ▶ Implementiert S-Schicht und Bitübertragungsschicht
- ▶ Eine Kombination von Hardware, Software, Firmware



Kommunikation der Netzwerkadapter



► Sender:

- Kapselt das Datagramm in dem Rahmen ein
- Fügt Daten der Fehlerkorrektur hinzu, führt Flusskontrolle aus, sorgt für verlässliche Übertragung usw.

► Empfänger

- Überprüft auf Fehler; führt Flusskontrolle aus, sorgt für verlässliche Übertragung usw.
- Extrahiert Datagramm, gibt an die höhere Schicht weiter

Video: CCNA 1 DATA LINK LAYER Fundamentals CHAPTER SEVEN

<https://www.youtube.com/watch?v=6qS5S78iQjo>

MAC-Adressierung und ARP

- Sicherungsschicht -

MicroNugget: What are IP and MAC Addresses?

<https://www.youtube.com/watch?v=V2SpN-OePzc>

Erinnerung: Internet-Protokollstapel

- ▶ Wie identifiziert man die Schicht?
 - ▶ Ein gutes Kriterium sind die Endpunkte der Kommunikation

Name	Endpunkte
Anwendungsschicht (application layer)	Mehrere <u>Programme</u> , jedes kann aus einem oder mehr Prozessen bestehen (z.B. Browser + Server)
Transportschicht (transport layer)	<u>Prozesse</u> (bzw. zugehörige Sockets – APIs des BS) auf einem oder mehreren Hosts
Netzwerkschicht (network layer)	Start- und End-Hosts (es wird <u>nicht</u> zwischen Prozessen auf einem Host unterschieden)
Sicherungsschicht (data link layer)	Zwei <u>direkt</u> verbundene Geräte (Host, Router, Switch) an den beiden Enden einer Teilstrecke
Bitübertragungss. (physical layer)	Elektronik der Leitungen oder Glasfaser an den Enden einer Teilstrecke

MAC-Adressen

- ▶ Wir betrachten nun Adressen, die eine physische Einheit (NIC oder Port eines Routers) identifizieren
- ▶ Das sind **MAC (Media Access Control)**-Adressen
 - ▶ Andere Namen: **LAN-** oder **physische-** oder **Ethernet**-Adr.
- ▶ Funktion: Übermittlung des Rahmens von einem NIC zu einem NIC auf dem gleichen Link
 - ▶ Länge: **48 Bit** (6 Bytes)
 - ▶ Festgelegt im ROM der NIC, oft kann sie auch via Software verändert werden

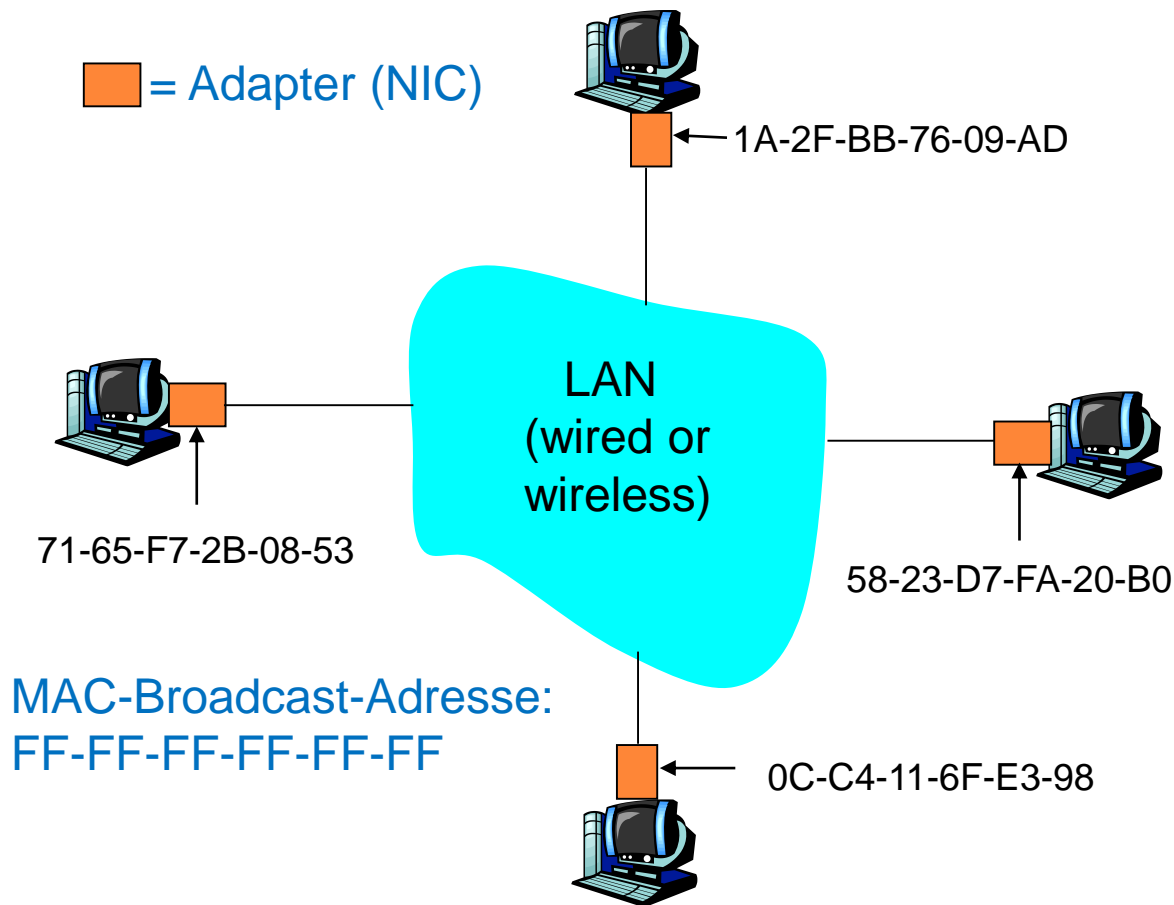
NIC = network interface card, d.h. Netzwerkkarte, Adapter

Bei einem *falschen* MAC wird die **MAC-Adresse** immer noch korrekt sein!



MAC-Adressen /2

Jeder NIC hat eine einmalige MAC-Adresse



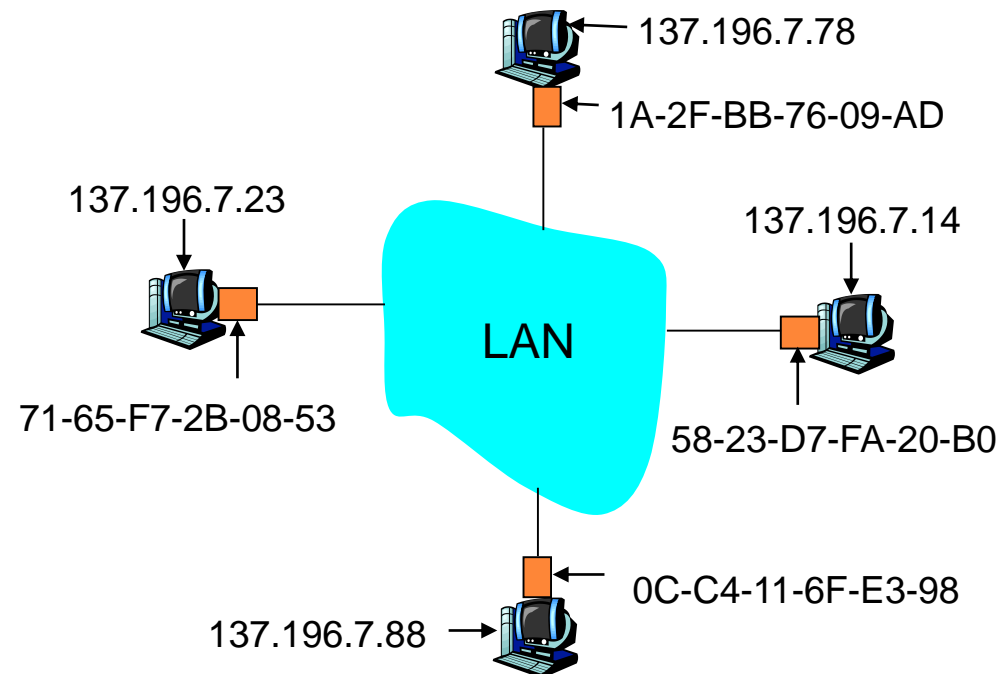
- Jeder Adapter (NIC) überprüft, ob die im Rahmen befindliche MAC-Adresse des Ziels seiner eigenen entspricht
- Bei Übereinstimmung extrahiert der Adapter das beigefügte Datagramm und reicht es an die höheren Schichten im Protokollstapel weiter
- Sonst löscht der Adapter das Paket

MAC-Adressen /3

- ▶ Zuordnung von MAC-Adressen wird von IEEE (**Institute of Electrical and Electronics Engineers**) vorgenommen
 - ▶ Hersteller kaufen (für eine symbolische Summe) eine 24-Bit “große” Menge an MAC-Adressen
- ▶ Analogie:
 - ▶ MAC-Adressen: wie Social Security Number (Link), in D. wie Rentenversicherungsnummer (ortsunabhängig)
 - ▶ IP-Adressen: wie Postadressen, identifizieren den “Ort” (Subnetz), aber nicht die “Bewohner”
- ▶ MAC-Adressen haben eine flache Hierarchie => portabel
 - ▶ Eine NIC und ihre MAC-Adresse kann sofort in allen Subnetzen eingesetzt werden
- ▶ IP-Adressen sind hierarchisch und nicht portabel
 - ▶ Adresse hängt von dem Subnetz ab

ARP: Address Resolution Protocol

Wie bestimmt man die MAC-Adresse eines Hosts aus seiner IP-Adresse?



- ▶ **ARP** ist ein Protokoll, das bei IPv4-Adressierung auf Ethernet benutzt wird
 - ▶ IPv6: **Neighbor Discovery Protocol (NDP)**
- ▶ Jeder Knoten (Host, Router) hat eine **ARP-Tabelle**
- ▶ **ARP-Tabelle:** IP-zu-MAC Adressenzuordnung für (einige) Knoten, genauer:
<IP address; MAC address; TTL>
 - ▶ TTL (Time To Live): Zeit, nach der die Zuordnung verfällt
 - ▶ ~20 Minuten

ARP: Gleiches Subnetz

- ▶ A möchte B ein Paket schicken, und die MAC-Adresse von B ist nicht in der Tabelle von A
- ▶ A macht einen **Broadcast** der Nachricht via “**ARP Query**” ...
 - ▶ Die die IP-Adresse von B enthält
 - ▶ Und Ziel-MAC-Adresse = FF-FF-FF-FF-FF-FF (Broadcast)
- ▶ B erhält das ARP-Query-Paket und antwortet A mit eigener MAC-Adresse
 - ▶ Rahmen wird nur an die MAC-Adresse von A geschickt (**unicast**)
- ▶ A speichert die IP-zu-MAC Zuordnung in der eigenen ARP-Tabelle, bis diese Information ungültig wird

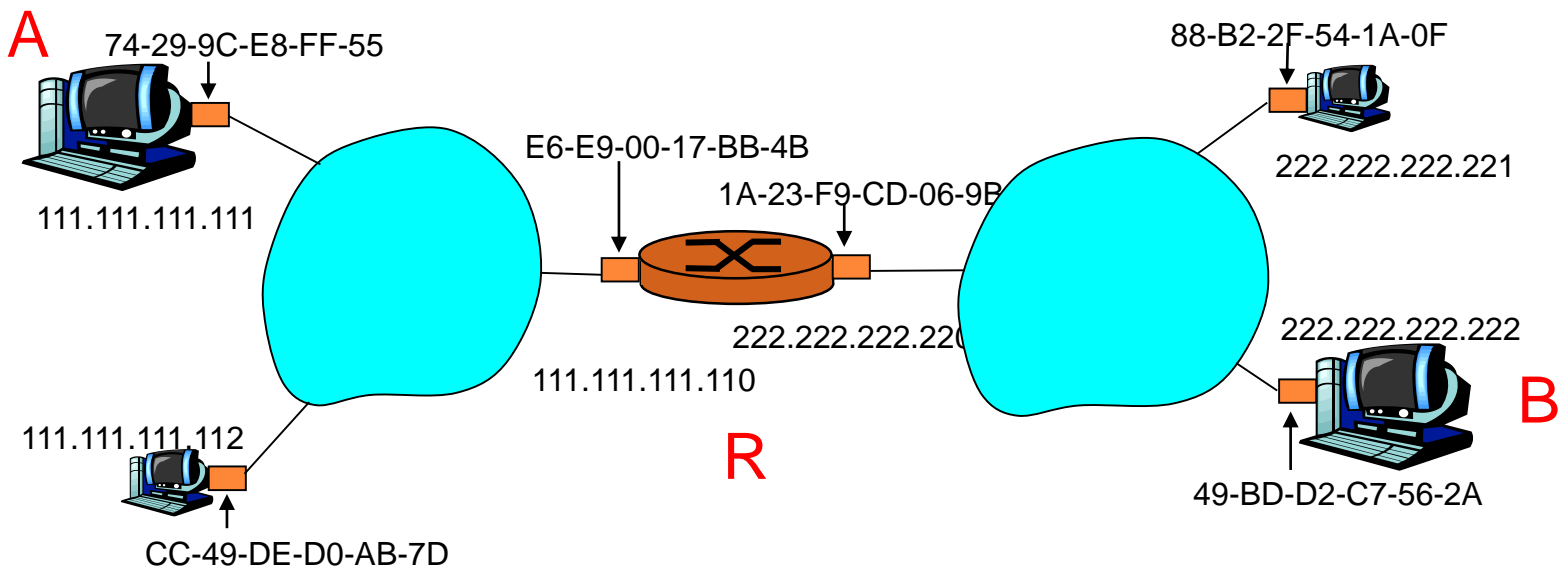
ARP-Nachrichtenformat

Bit 0–7		Bit 8–15		Bit 16–23		Bit 24–31	
Hardwareadrestyp (1)				Protokolladrestyp (0x0800)			
Hardwareadressgröße (6)		Protokolladressgröße (4)		Operation			
Quell-MAC-Adresse							
Quell-MAC-Adresse				Quell-IP-Adresse			
Quell-IP-Adresse				Ziel-MAC-Adresse			
Ziel-MAC-Adresse							
Ziel-IP-Adresse							

ARP: Routing in ein anderes Subnetz

Übertragung eines Pakets von A nach B via R

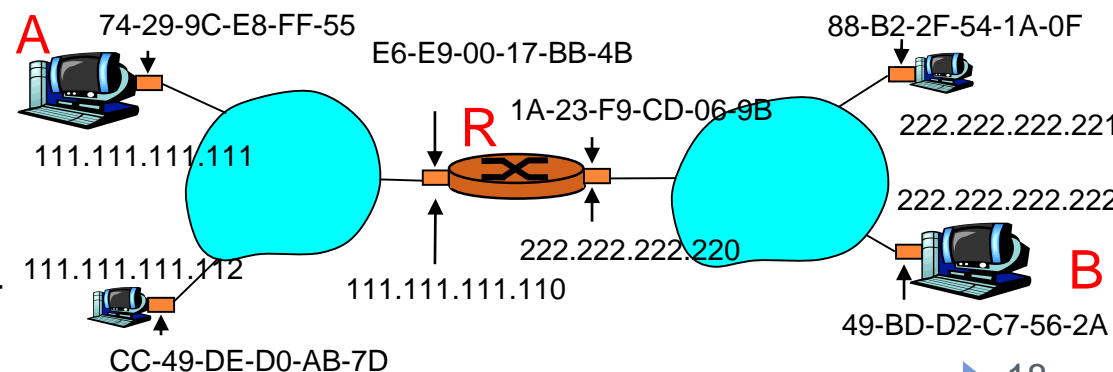
Annahme: A kennt die **IP-Adresse** von B



- ▶ Es gibt zwei ARP-Tabellen in diesen Router, je eine pro NIC zu einem Subnetz

Routing in ein anderes Subnetz /2

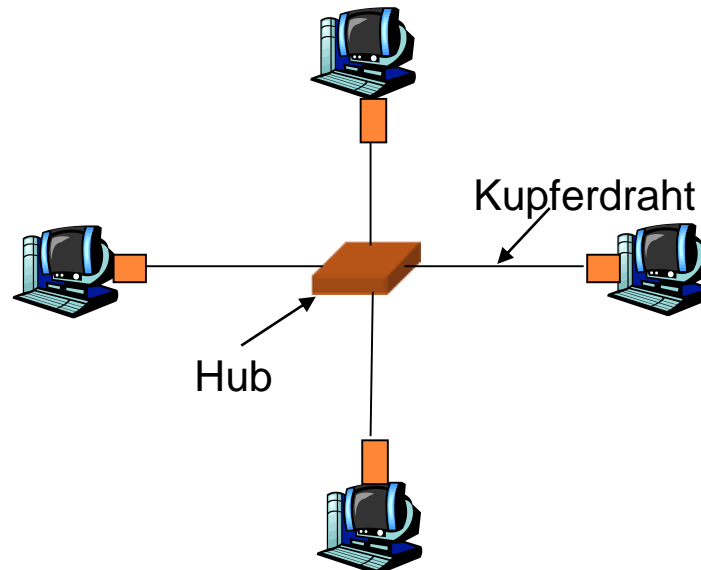
- ▶ 111.111.111.111 erstellt ein IP-Datagramm mit dem Ziel 222.222.222.222
- ▶ 111.111.111.111 schlägt in seiner IP-Weiterleitungstabelle nach und stellt fest, dass dieses Paket über R (111.111.111.110) weitergeleitet werden muss
- ▶ 111.111.111.111 verwendet ARP, um die MAC-Adresse von 111.111.111.110 zu bestimmen
- ▶ 111.111.111.111 erstellt einen Rahmen der Sicherungsschicht mit E6-E9-00-17-2B-4B als Zieladresse
 - ▶ Dieser Rahmen enthält das IP-Datagramm von 111.111.111.111 an 222.222.222.222
- ▶ Die Netzwerkkarte von 111.111.111.111 sendet den Rahmen
- ▶ Die Netzwerkkarte von 111.111.111.110 empfängt den Rahmen
- ▶ R packt das IP-Datagramm aus und stellt fest, dass es für 222.222.222.222 bestimmt ist
- ▶ Über die IP-Weiterleitungstabelle stellt R fest, dass er das Datagramm direkt an 222.222.222.222 ausliefern kann
- ▶ R verwendet ARP, um die MAC-Adresse von 222.222.222.222 zu erfahren
- ▶ R erstellt einen Rahmen, der das Datagramm von 111.111.111.111 an 222.222.222.222 enthält, und sendet es an die so ermittelte MAC-Adresse



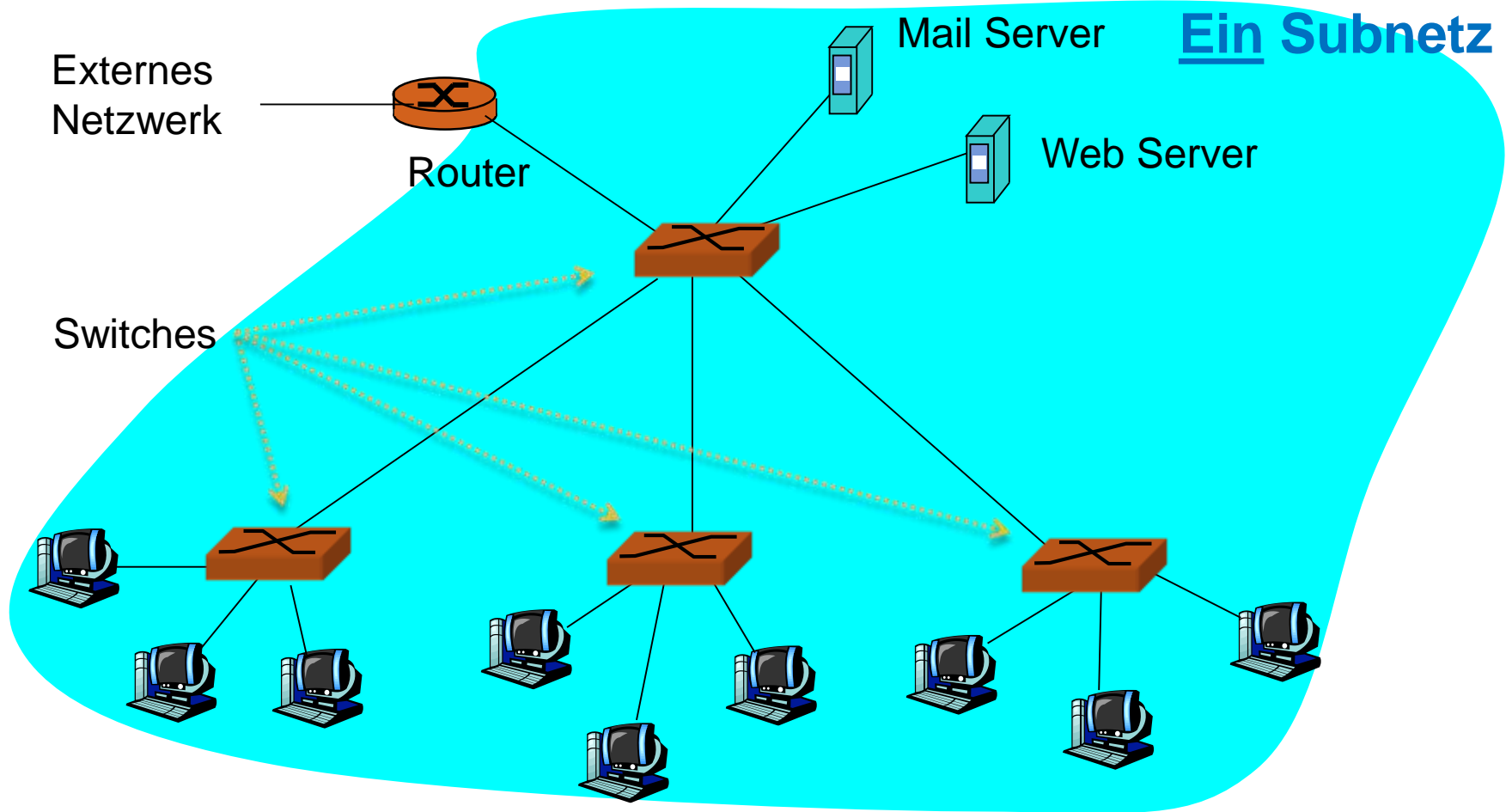
Hubs und Switches

Hubs: Einfache „Mehrfachsteckdosen“

- ▶ Bits, die bei einem Port ankommen, werden mit gleicher Geschwindigkeit an alle anderen weitergeleitet
- ▶ Alle Knoten am Hub können miteinander kollidieren
- ▶ Keine Pufferung, kein Kollisionserkennung am Hub

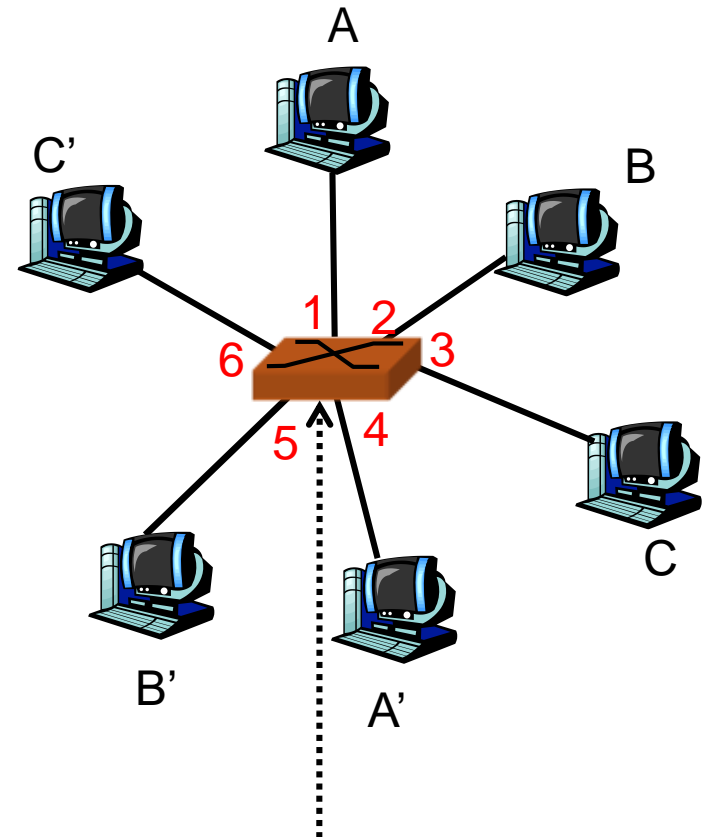


Switches: Paketverteiler **in einem Subnetz**



Switches: Eigenschaften

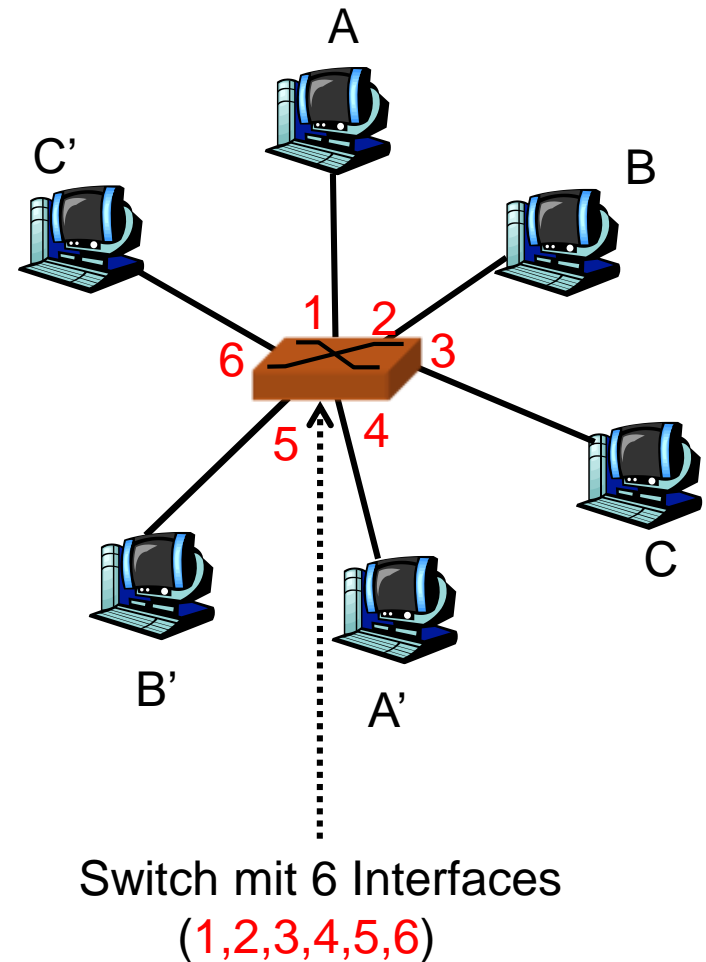
- ▶ Jeder Host hat einen dedizierten Link zum Switch
 - ▶ Links an einem Switch können unterschiedlichen Typ und Geschwindigkeit haben
- ▶ Switches puffern die Pakete: Hilfe bei Überlast
- ▶ Switches sind “Plug-and-Play” – keine Eingriffe nötig
 - ▶ Die Weiterleitung erfolgt anhand der MAC-Adressen
- ▶ Keine Kollisionen: wenn A an C schickt, bekommt nur C das Paket, sonst keiner
- ▶ Wie geht das?



Switch mit 6 Interfaces
(1,2,3,4,5,6)

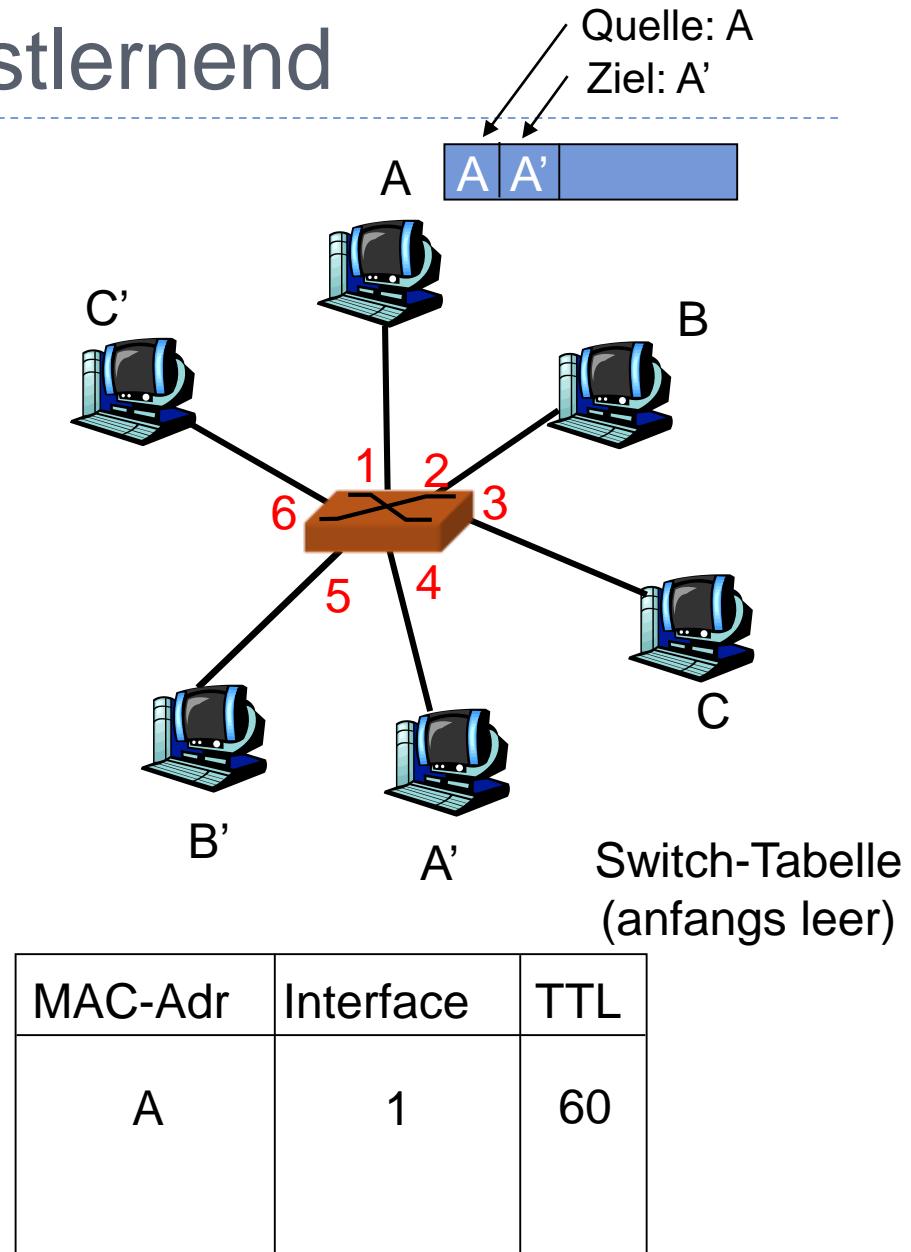
Switch-Tabelle

- ▶ Woher weiß dieser Switch, dass A über Link 1 erreichbar ist und z.B. C über Link 3?
- ▶ Jeder S. hat eine Switch-Tabelle mit Einträgen:
 - ▶ (MAC-Adresse des Hosts, Interface/Link-Nummer zum Host, Zeit des Eintrags)
- ▶ Wenn A zum C ein Paket schickt, erkennt S. anhand der Ziel-MAC-Adresse, dass das Paket nur an Interface 3 gehen soll
- ▶ Was passiert, wenn keine passende MAC-Adresse gefunden wird?



Switch-Tabelle ist Selbstlernend

- ▶ Switch lernt, welche NICs (bzw. MAC-Adressen) an welchen Links erreichbar sind!
 - ▶ Wenn ein Rahmen über Link x ankommt, extrahiert der Switch die MAC-Adresse der Quelle
 - ▶ Falls es diese noch nicht in der Switch-Tabelle gibt, wird sie eingetragen / erneuert



Switch: Filterung und Weiterleitung

Neuen Rahmen erhalten:

1. Aktualisiere die Switch-Tabelle (wie oben beschrieben)
2. Finde in der Switch-Tabelle den Eintrag X mit der Ziel-MAC-Adresse

3. **if** gefunden? **then** {


if Ziel liegt auf dem Link, über den das Paket ankam?

then ignoriere den Rahmen

else leite den Rahmen an den Link weiter, der von dem Eintrag X angegeben wurde

 }

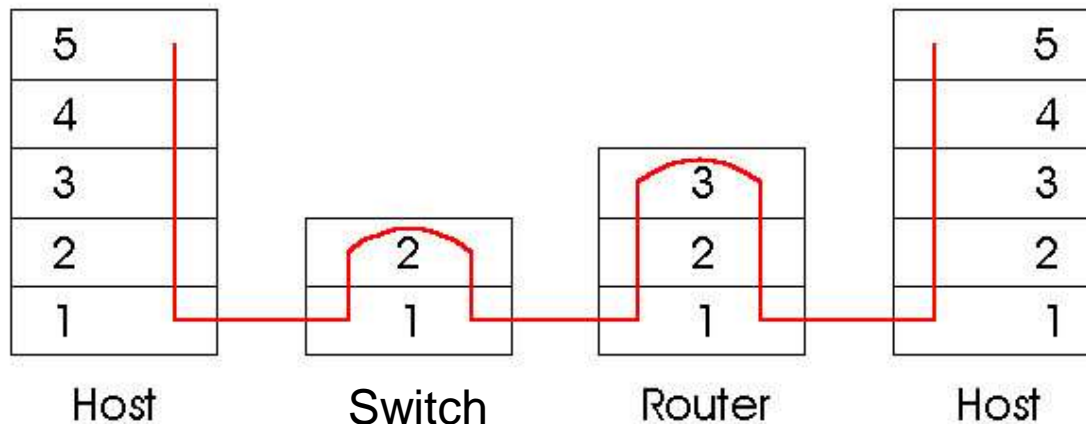
else "Fluten"



schickt das Paket an alle Interfaces
abgesehen von dem, an dem das Paket ankam

Switches vs. Router

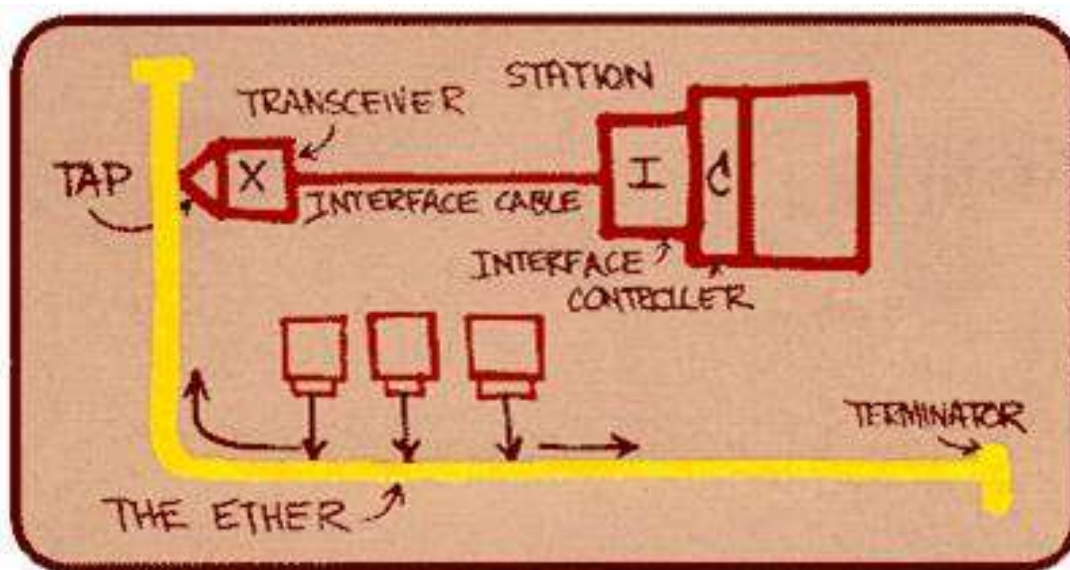
- ▶ Router sind komplizierter und etwas langsamer
 - ▶ Sie manipulieren das Paket: TTL + Prüfsumme ändern, MAC-Adresse ändern....
- ▶ Switches lernen selbständig ohne Sys-Admins
 - ▶ Aber: Sie können nur innerhalb eines Subnets arbeiten
- ▶ Beide puffern und leiten weiter
 - ▶ Router: überprüfen Header der Netzwerkschicht (IP-Datagramme)
 - ▶ Switches schauen sich nur die Header der Sicherungsschicht an



Ethernet

Ethernet (Sicherungsschicht)

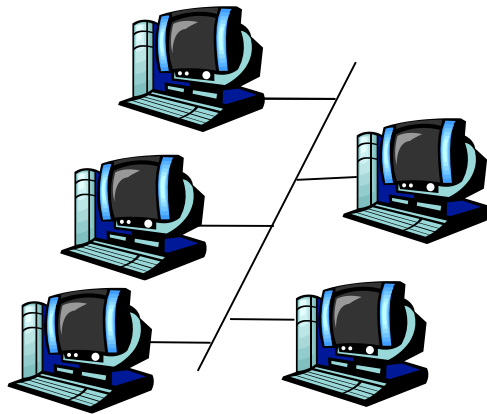
- ▶ Dominante Technologie für leitungsgebundene LANs
- ▶ Unter \$20 für eine NIC
- ▶ Erste umfassend eingesetzte Hochgeschwindigkeits-LAN-Technologie
- ▶ Einfacher und billiger als Alternativen
 - ▶ Token Ring, FDDI, ATM
- ▶ Geschwindigkeit gestiegen mit der Zeit: 10 Mbps – 10 Gbps



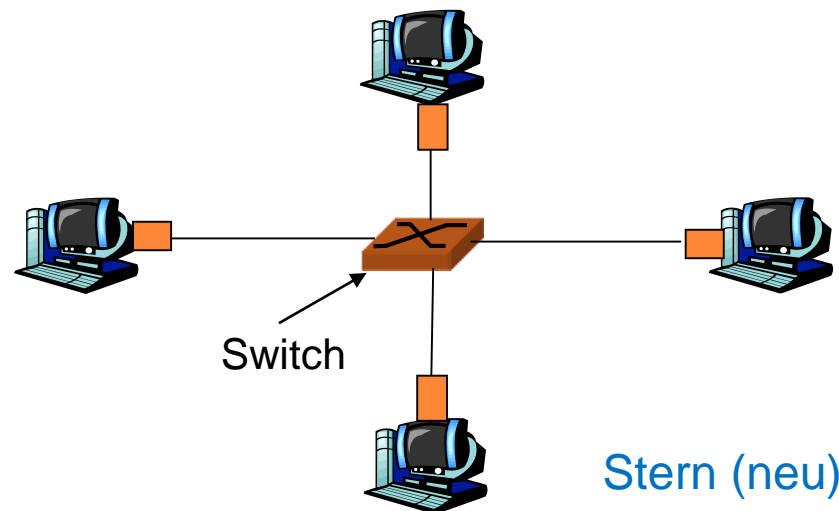
Ursprünglicher Ethernet-Entwurf (1973) von Robert Metcalfe
Urheber und Gründer von 3Com (in 1979)
3Com: verkauft 2010 an HP

Topologien

- ▶ **Bus-Topologie** wurde bis zu 2000er Jahre eingesetzt
 - ▶ Die Nachrichten aller Knoten können miteinander kollidieren
 - ▶ Ggf. war der “Bus” ein Stern mit einem Hub in der Mitte
- ▶ Heutzutage: Switch-basierte **Stern-Topologie**
 - ▶ Ein aktiver Switch verhindert Kollisionen der Pakete, da nur diese nur an die tatsächlichen Ziele geschickt werden



Bus mit einem
koaxialem Kabel (alt)



Stern (neu)

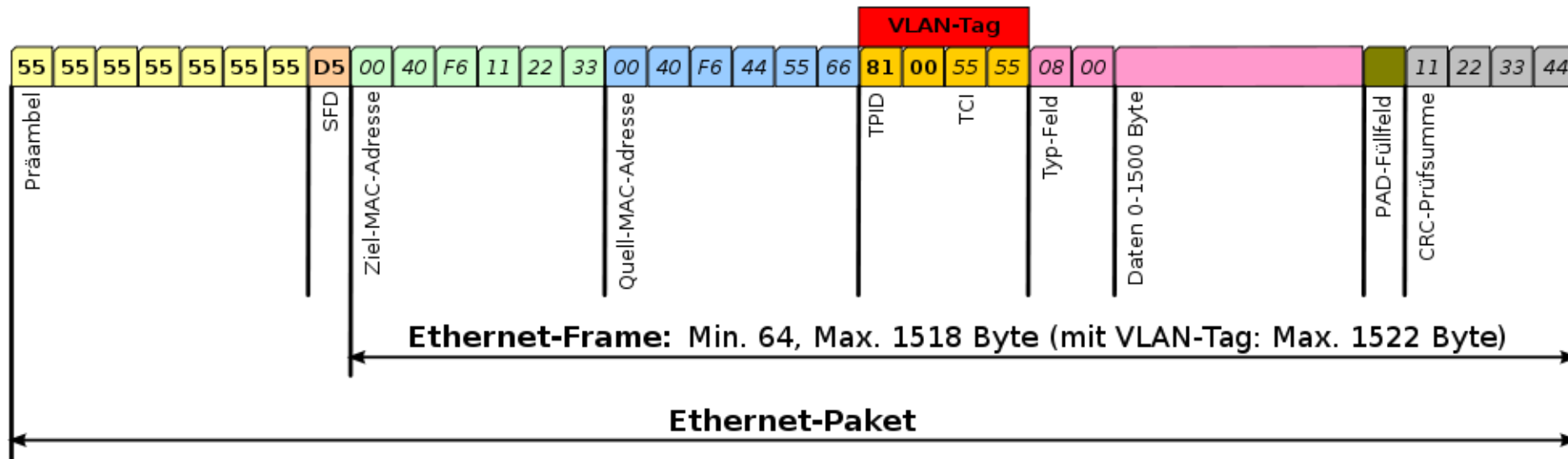
Ethernet - Rahmenstruktur



- ▶ Die sendende NIC kapselt ein IP-Datagramm in einen **Ethernet-Rahmen** ein
- ▶ **Präambel (Preamble)**:
 - ▶ 7 Bytes mit Muster 10101010 gefolgt von einem Byte mit Muster 10101011
 - ▶ Benutzt, um die Taktgeber des Sender und Empfängers zu synchronisieren & Beginn des Paketes zu erkennen
- ▶ **Typ**: Typ des nächsthöheren Protokolls (meistens IP, aber andere möglich: z.B. Novell IPX, AppleTalk)
- ▶ **CRC (Prüfsumme)**: Wird beim Empfänger überprüft, bei Fehler wird der Rahmen verworfen

Ethernet – Rahmenstruktur /2

- ▶ **Adressen** sind 6 Bytes lang
 - ▶ Der Adapter überprüft, ob die im Rahmen befindliche MAC-Adresse des Ziels seiner eigenen entspricht
 - ▶ Bei einer Übereinstimmung extrahiert der Adapter das beigefügte Datagramm und reicht es an die höheren Schichten im Protokollstapel weiter
 - ▶ Gibt es keine Übereinstimmung, löscht der Adapter den Rahmen

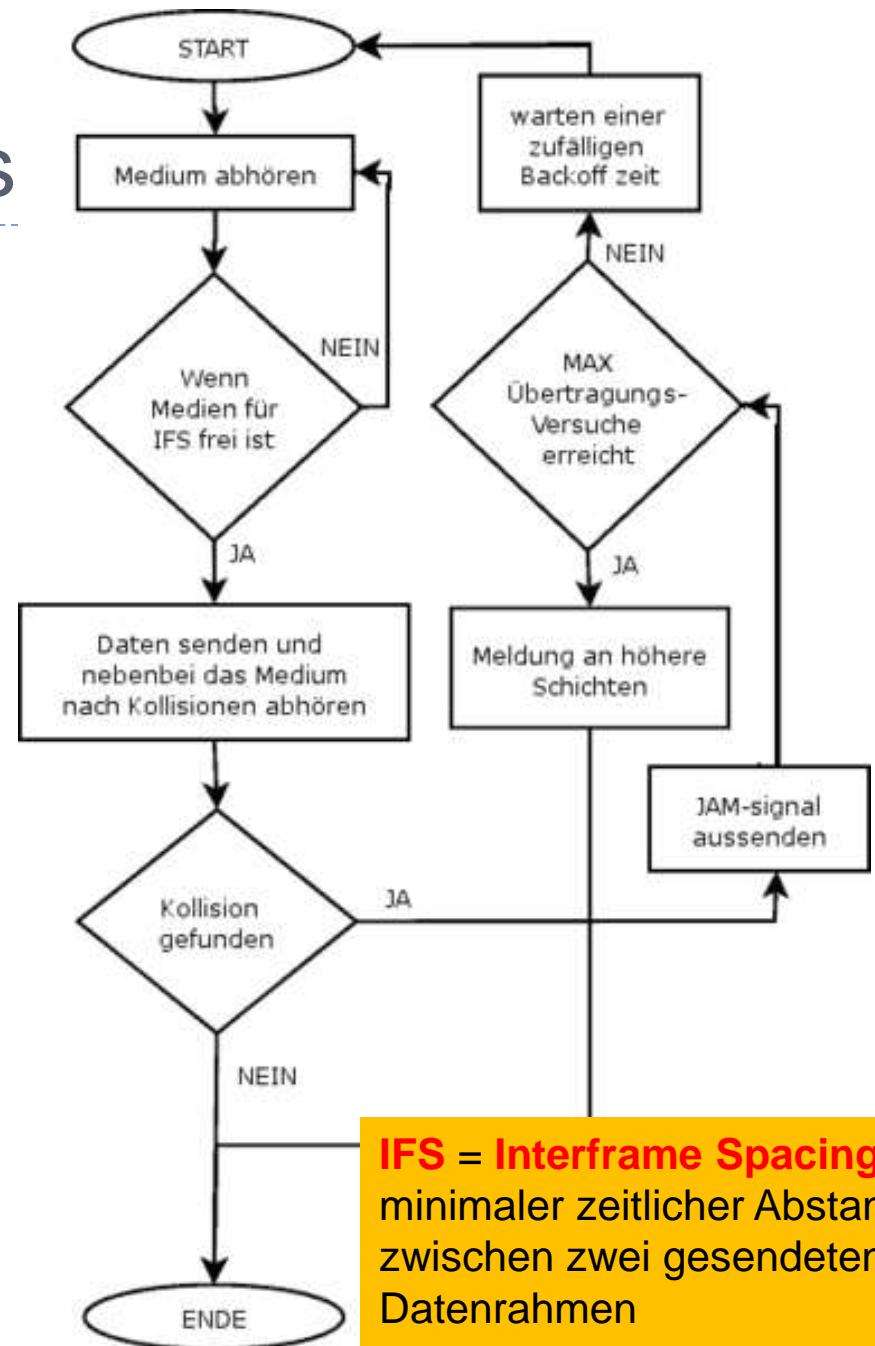


Ethernet: Eigenschaften

- ▶ Ethernet ist verbindungslos und unzuverlässig
 - ▶ Der Empfänger schickt keine ACKs zu dem Sender
 - ▶ Verlässliche Übertragung wird durch höhere Schichten sichergestellt
- ▶ Der **CSMA/CD**-Algorithmus regelt den Zugriff der Systeme auf das gemeinsame Medium (Link)
 - ▶ **Carrier Sense Multiple Access with Collision Detection**
 - ▶ Eine Weiterentwicklung des ALOHAnet aus den 1960er-Jahren
- ▶ CSMA/CD funktioniert wie eine Party, auf der alle Gäste über ein gemeinsames Medium (die Luft) kommunizieren
 - ▶ Bevor sie zu sprechen beginnen, warten sie höflich darauf, dass der andere Gast zu reden aufgehört hat
 - ▶ Wenn zwei Gäste zur gleichen Zeit zu sprechen beginnen, stoppen beide und warten für eine kurze, zufällige Zeitspanne, bevor sie einen neuen Anlauf wagen

CSMA/CD-Algorithmus

- ▶ Entdeckt der Sender eine Kollision, sendet er zunächst ein 48-Bit langes **Jam-Signal**
- ▶ Danach legt er eine Pause ein, deren Länge mit **exponentiellen Backoff**-Verfahren bestimmt wird:
 1. Wähle ein zufälliges **K** aus $\{0, 1, 2, \dots, 2^m - 1\}$, dabei ist **m** = Anzahl der Kollisionen (aufeinanderfolgenden)
 2. Warte **K***512 Bitübertragungszeiten



CSMA/CD-Algorithmus /2

Exponential Backoff:

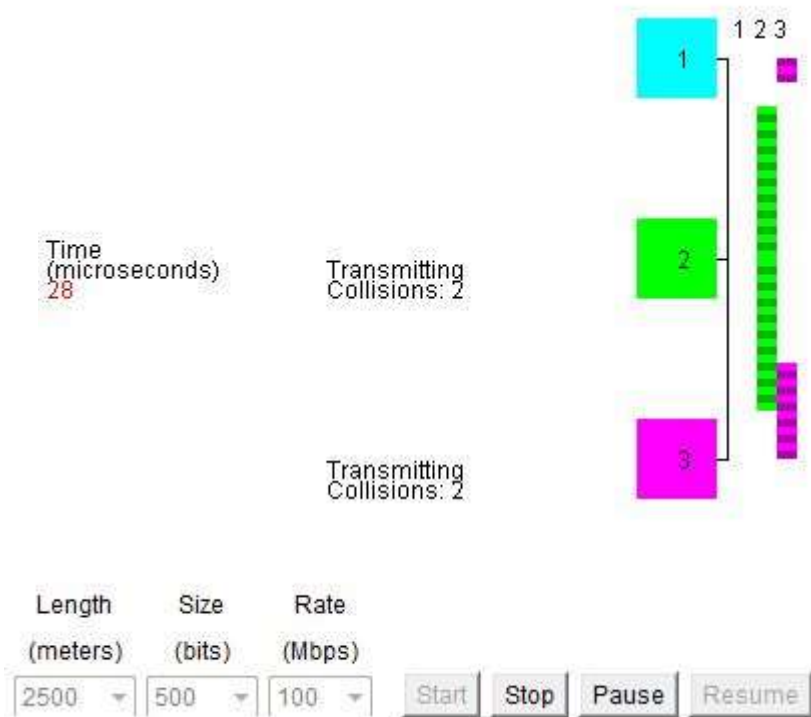
- ▶ Ziel: Passe die Länge der Pause an die gegenwärtige Buslast (load) an
 - ▶ Hohe Last: Zufälliges Warten soll länger sein

Jam-Signal:

- ▶ Stellt sicher, dass alle anderen Sender die Kollision bemerken

Bitübertragungszeit:

- ▶ 0.1 Mikrosek. für 10 Mbps Ethernet; => bei K=1023 beträgt die Wartezeit ca. 50 Millisekunden

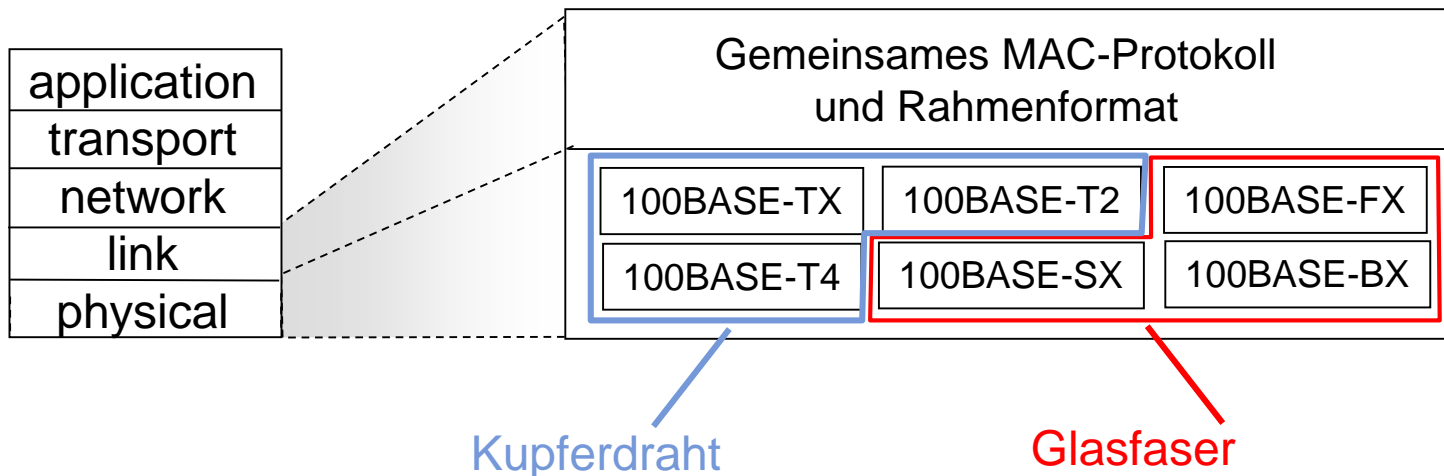


Siehe Java-Applet: [Link](http://media.pearsoncmg.com/aw/aw_kurose_network_2/applets/csmacd/csmacd.html)
(http://media.pearsoncmg.com/aw/aw_kurose_network_2/applets/csmacd/csmacd.html)

802.3 Ethernet-Standards

- ▶ **Es gibt viele** verschiedene Ethernet-Standards
 - ▶ Gemeinsames MAC-Protokoll und Rahmenformat
 - ▶ Verschiedene Geschwindigkeiten: 2 Mbps, 10 Mbps, 100Mbps, 1Gbps, 10G bps
 - ▶ Verschiedene Medien der Bitübertragungss.: Draht, Glasfaser
- ▶ Braucht man in den modernen Switch-basierten Ethernet noch ein Medienzugriffsprotokoll (wie CSMA/CD)?

Nein, eigentlich nicht; was nur gleich geblieben ist, ist das Rahmenformat



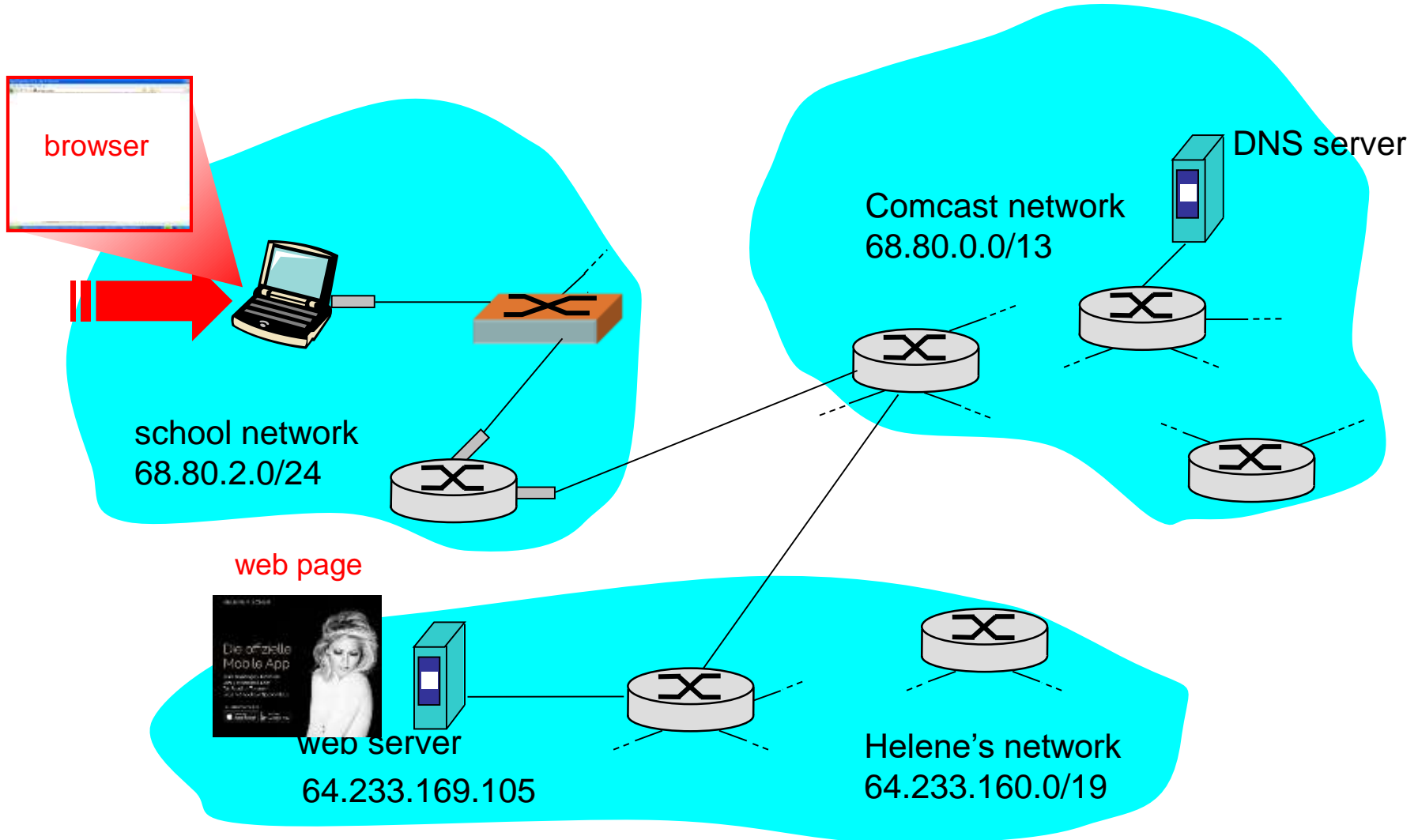
Tag der offenen Tür ... bei Ihrem Webbrowser!

A day in the life of a web request

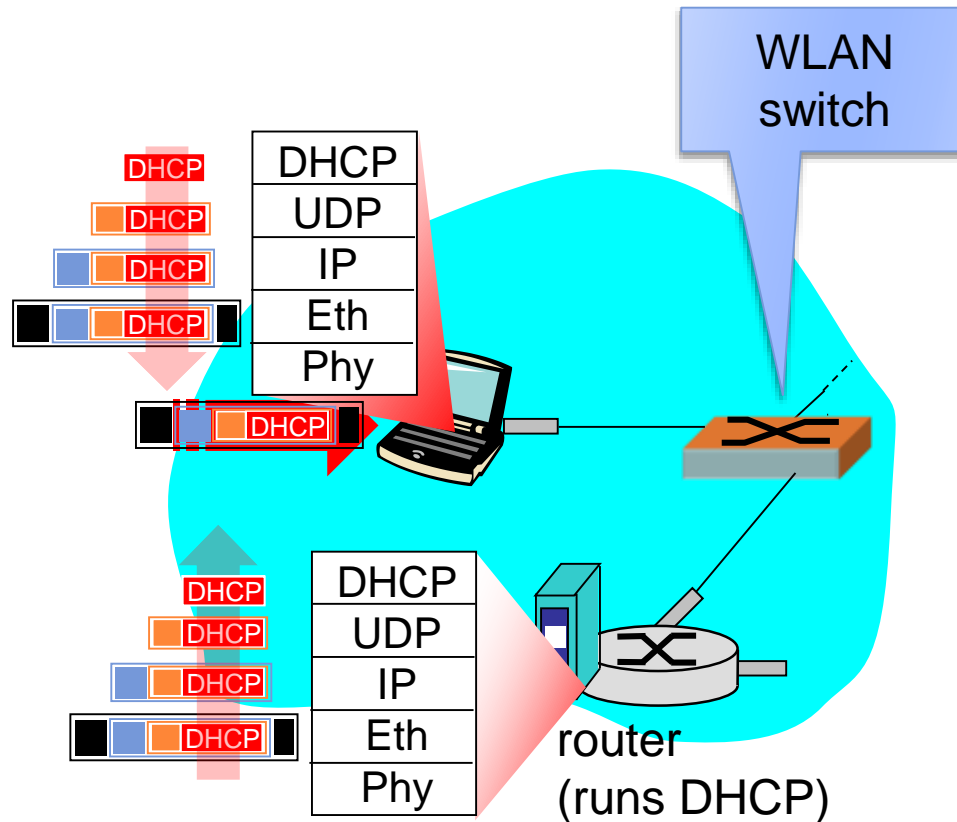
Synthesis: a day in the life of a web request

- ▶ journey down protocol stack complete!
 - ▶ application, transport, network, link
- ▶ putting-it-all-together: synthesis!
 - ▶ **goal:** identify, review, understand protocols (at all layers) involved in seemingly simple scenario: requesting www page
 - ▶ **scenario:** student attaches laptop to campus network, requests & receives web page
<http://www.helene-fischer.de/>

A day in the life: scenario

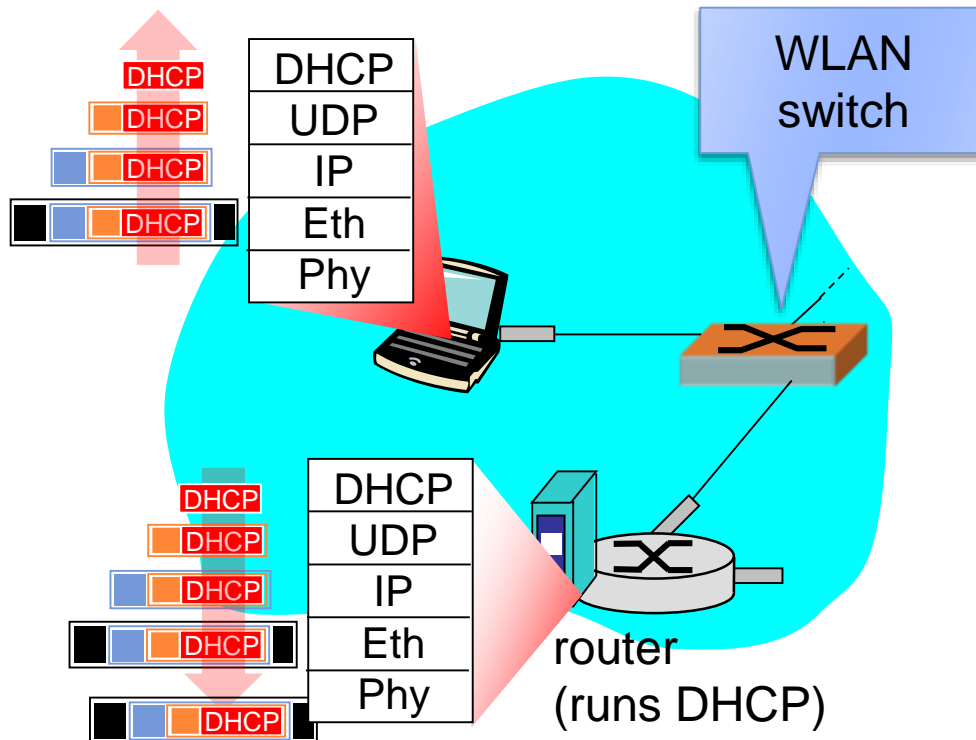


A day in the life... connecting to the Internet



- ▶ connecting laptop needs to get its own IP address, addr of first-hop router, addr of DNS server: use **DHCP**
- ❑ DHCP request **encapsulated** in **UDP**, encapsulated in **IP**, encapsulated in **802.1 Ethernet**
- ❑ Ethernet frame **broadcast** (destination **FFFFFFFFFFFF**) on LAN, received at router running **DHCP** server
- ❑ Ethernet **demux'ed** to IP demux'ed, UDP demux'ed to DHCP

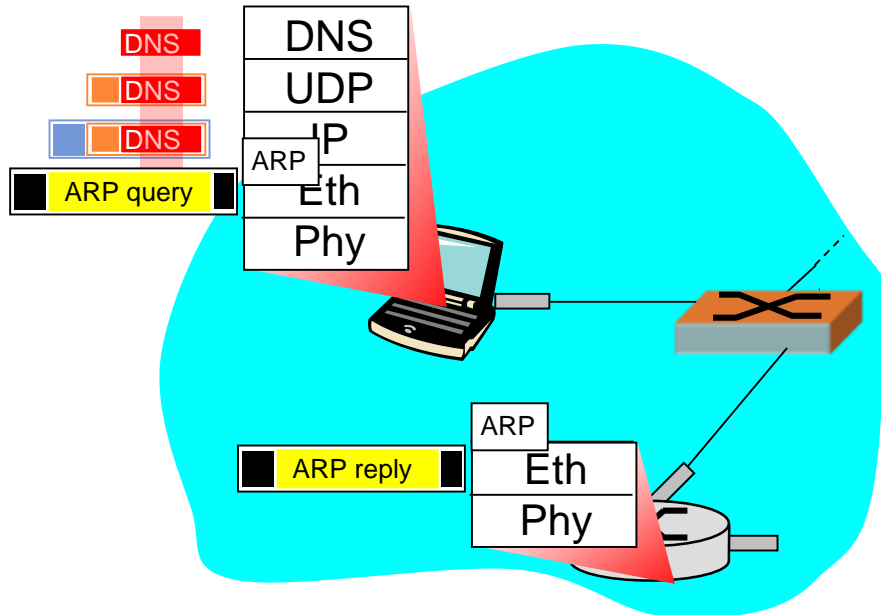
A day in the life... connecting to the Internet



- ▶ DHCP server formulates ***DHCP ACK*** containing client's IP address, IP address of first-hop router for client, name & IP address of DNS server
- encapsulation at DHCP server, frame forwarded (**switch learning**) through LAN, demultiplexing at client
- DHCP client receives DHCP ACK reply

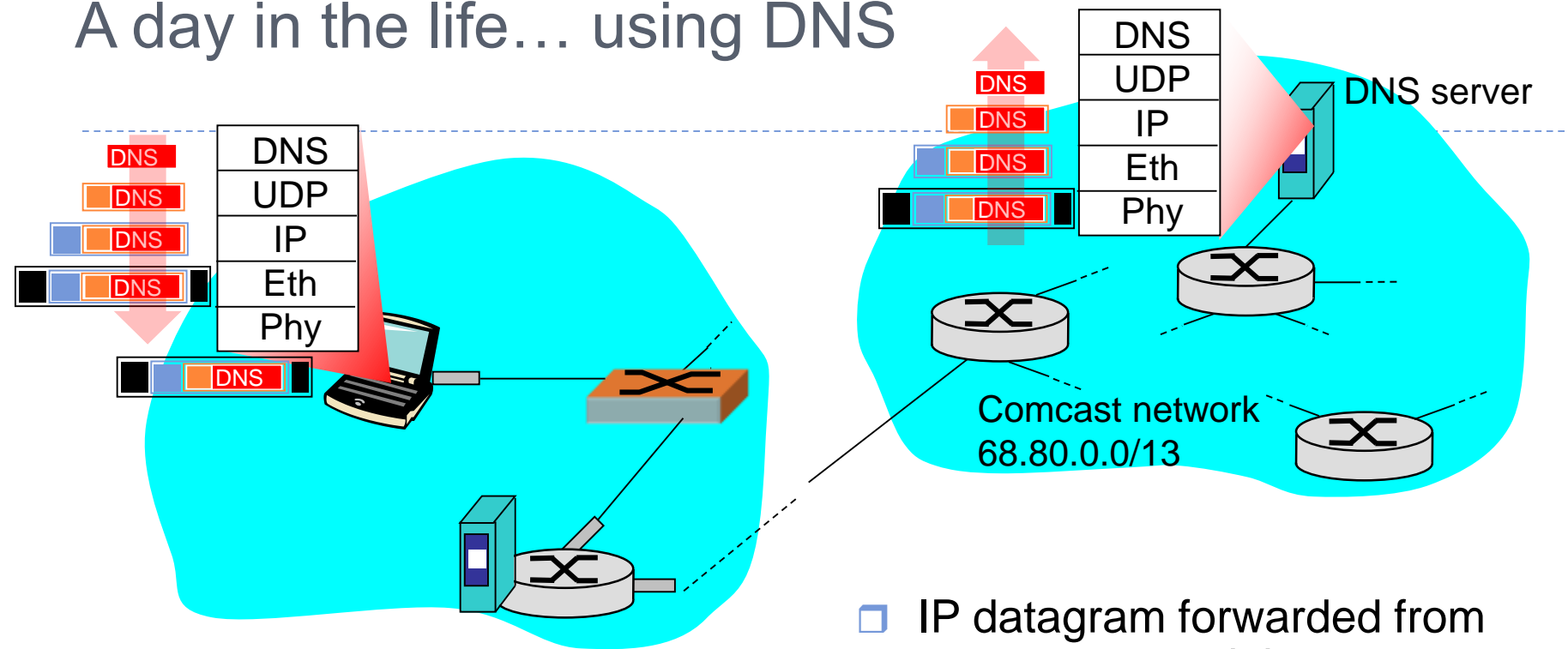
Client now has IP address, knows name & addr of DNS server, IP address of its first-hop router

A day in the life... ARP (before DNS, before HTTP)



- ▶ before sending **HTTP** request, need IP address of <http://www.helene-fischer.de/> : **DNS**
- DNS query created, encapsulated in UDP, encapsulated in IP, encapsulated in Eth. In order to send frame to router, need MAC address of router interface: **ARP**
- **ARP query** broadcast, received by router, which replies with **ARP reply** giving MAC address of router interface
- client now knows MAC address of first hop router, so can now send frame containing DNS query

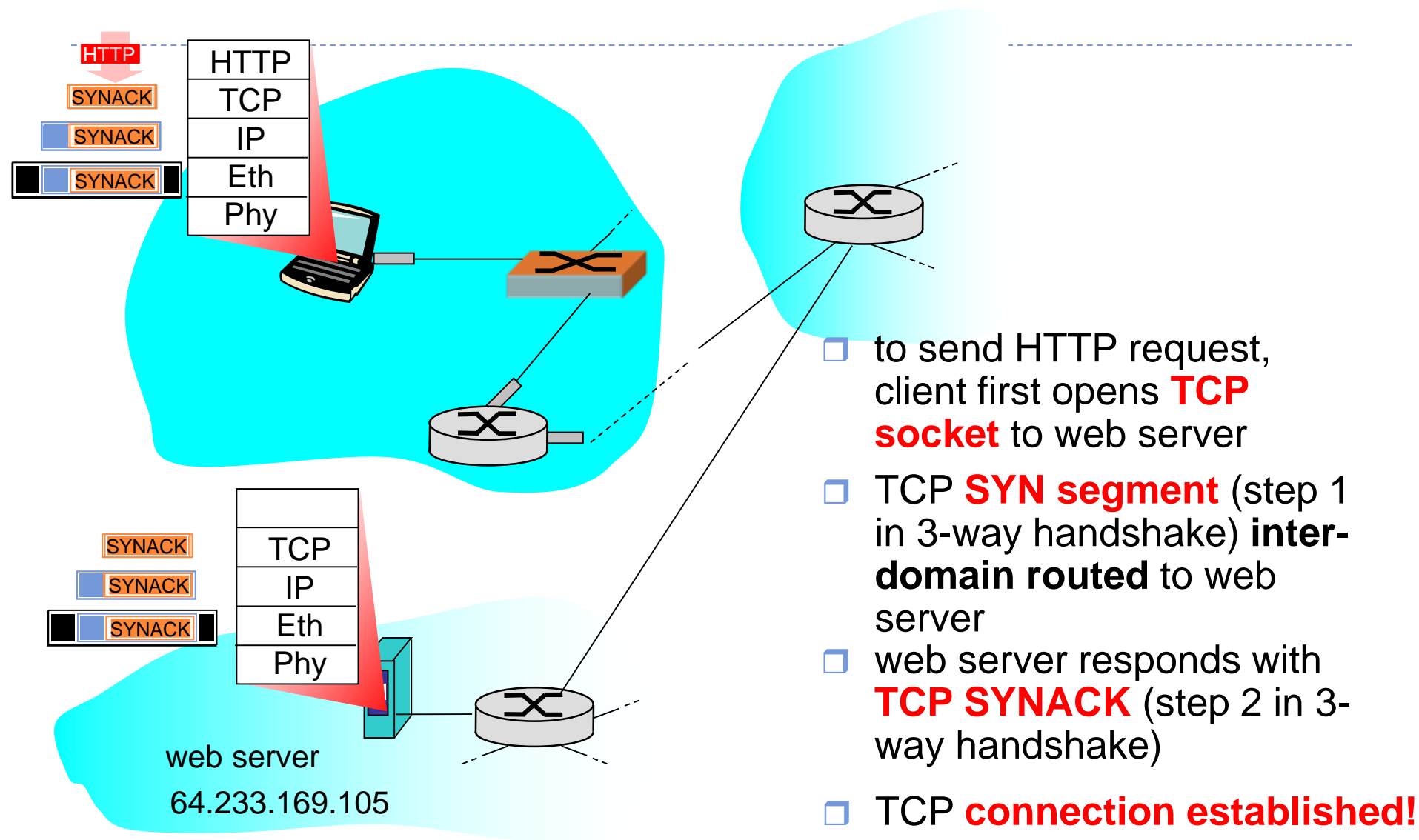
A day in the life... using DNS



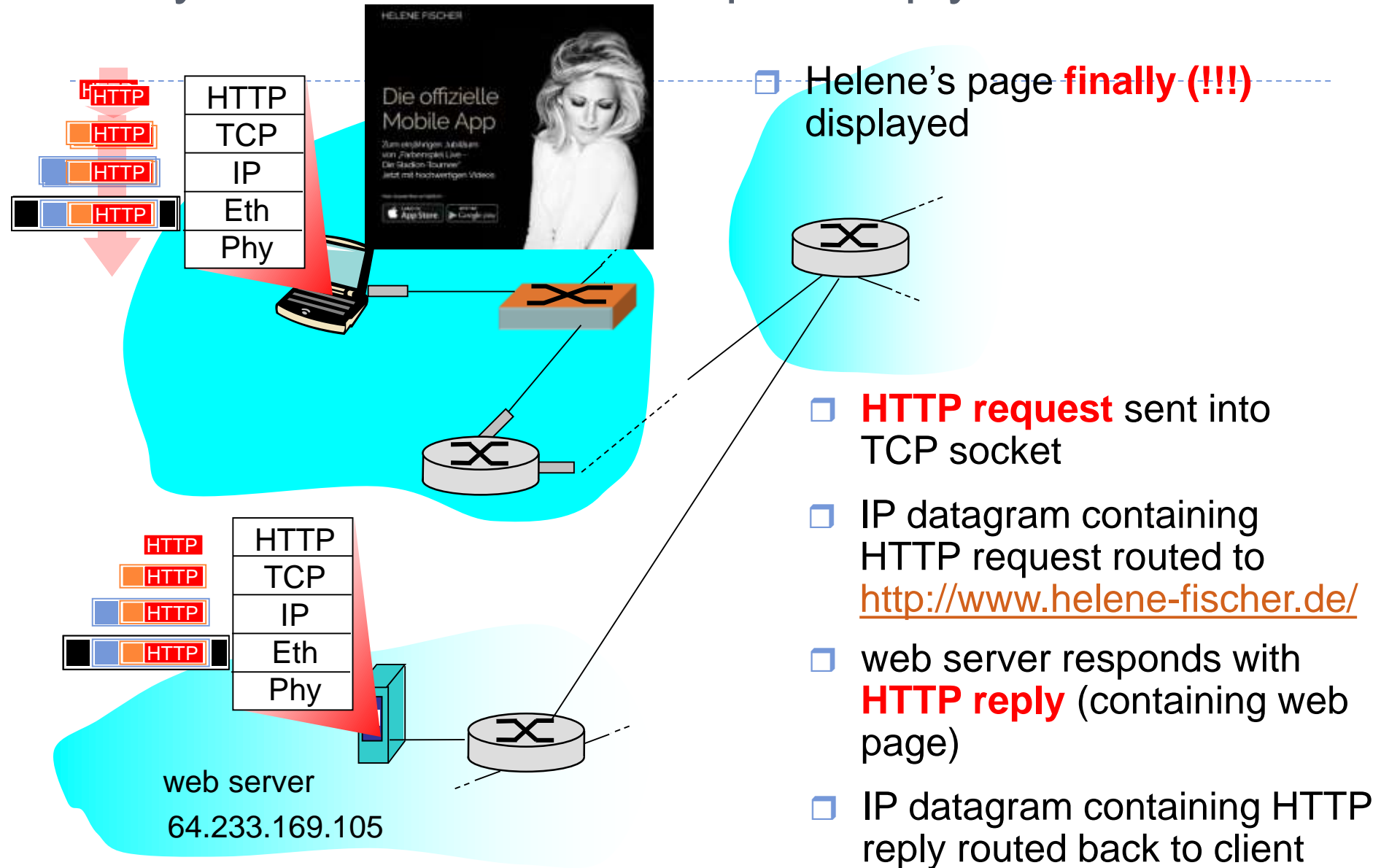
- ❑ IP datagram containing DNS query forwarded via LAN switch from client to 1st hop router

- ❑ IP datagram forwarded from campus network into comcast network, routed (tables created by **RIP, OSPF, IS-IS** and/or **BGP** routing protocols) to DNS server
- ❑ demux'ed to DNS server
- ❑ DNS server replies to client with IP address of <http://www.helene-fischer.de/>

A day in the life... TCP connection carrying HTTP



A day in the life... HTTP request/reply



Zusammenfassung

▶ Sicherungsschicht

- ▶ MAC-Adressierung und ARP
- ▶ Switches
- ▶ Ethernet

▶ Quellen:

- ▶ Kurose / Ross Kapitel 5, Abschnitte 5.1, 5.4, 5.5, 5.6

– **Alle Schichten** –

- ▶ *A day in the life of a web request*

▶ Quellen:

- ▶ Kurose / Ross Kapitel 4, Abschnitte 4.5-4.6
- ▶ Wikipedia

Danke.