Technische Hochschule Ingolstadt
Fakultät Informatik

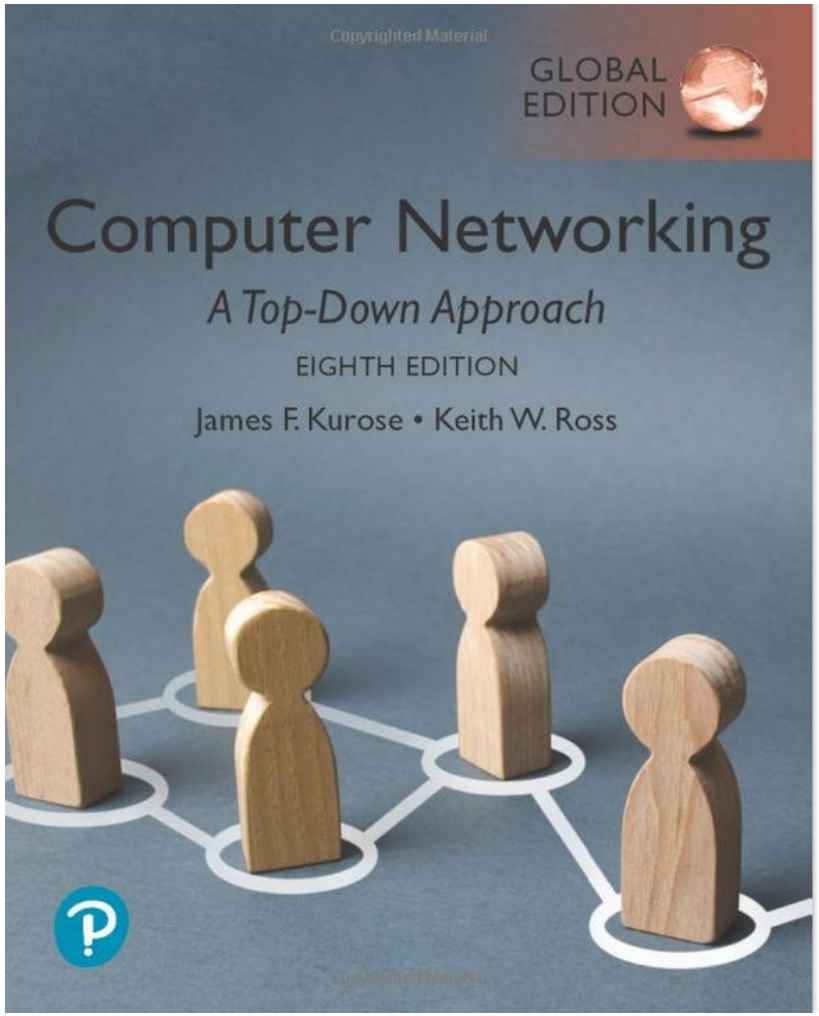
Kapitel 4: Vermittlungsschicht - Datenpfad

FFI_NW WS 2024

Vorlesung „Netzwerke“

20.09.2024

Der Inhalt des Foliensatzes basiert auf bzw. ist adaptiert aus:



Computer Networking: A Top-Down Approach

8th edition [Global Edition]
Jim Kurose, Keith Ross
Pearson, 2021

ISBN-10 : 1292405465
ISBN-13 : 978-1292405469

Mehrere Ausgaben (auch deutsche Editionen) in der Bibliothek verfügbar

Unsere Ziele:

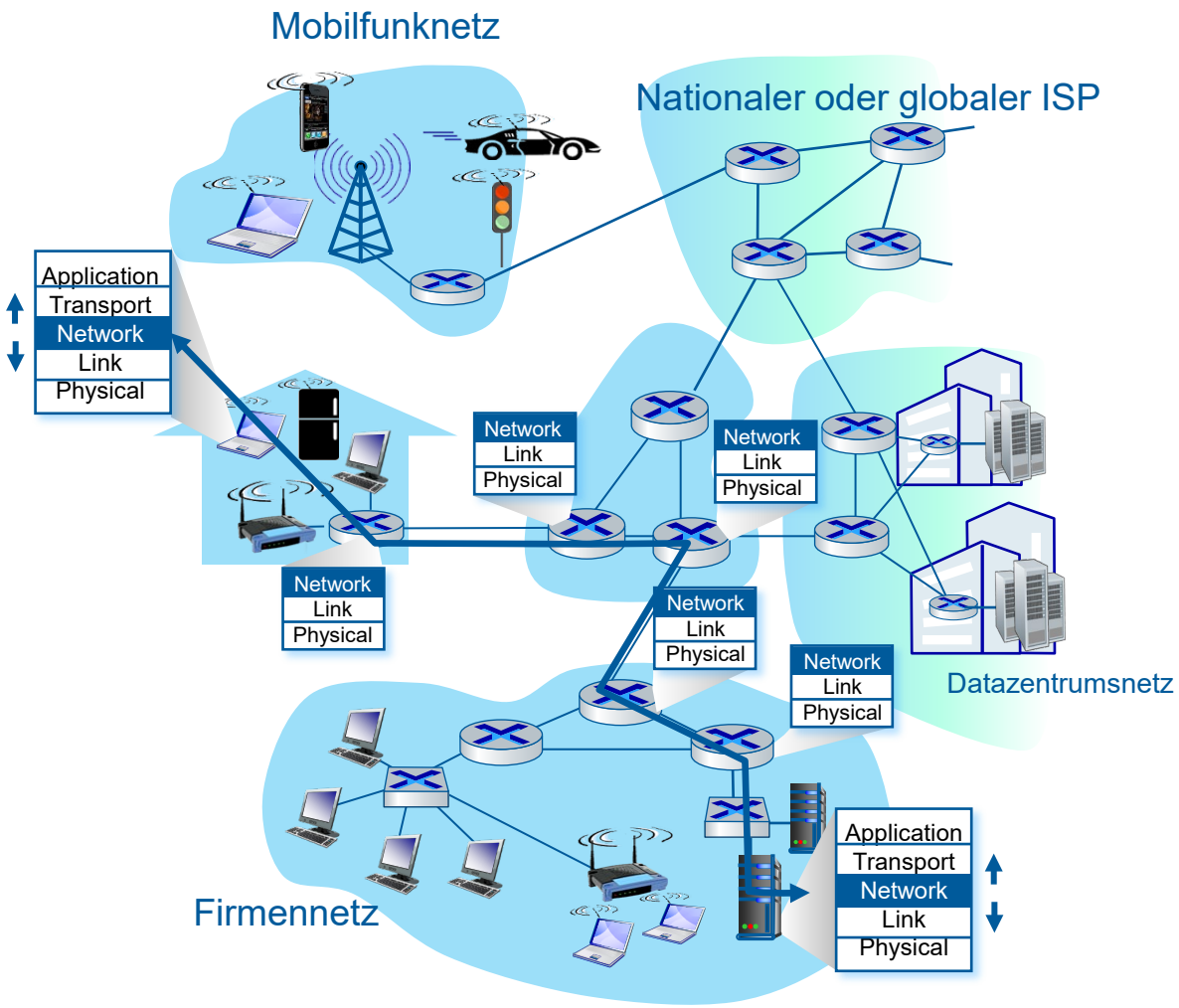
- **Verstehen der Prinzipien hinter den Diensten der Vermittlungsschicht mit Fokus auf den Datenpfad:**
  - Dienstmodelle
  - Forwarding versus Routing
  - Wie ein Router funktioniert
  - Adressierung
  - Generalized Forwarding
  - Internet Architektur
- **Instanziierung & Implementierung im Internet**
  - IP-Protokoll
  - NAT, Mittelboxen

Übersicht: Vermittlungsschicht - Datenpfad

- **Vermittlungsschicht: Überblick**
  - Datenpfad
  - Kontrollebene
- **Was steckt in einem Router**
  - Eingangsports, Switching, Ausgangsports
  - Puffer Management, Scheduling
- **IP: das Internet Protokoll**
  - Datagramm Format
  - Adressierung
  - Network Address Translation
  - IPv6
- **Generalized Forwarding, SDN**
  - Match+Action
  - OpenFlow: Match+Action
- **Mittelboxen**
  - Funktionen von Mittelboxen
  - Evolution & Prinzipien der Internet Architektur

Dienste und Protokolle der Vermittlungsschicht

- Transport eines Segments von Sender zum Empfangs-Host
  - **Sender**: kapselt Segmente in Datagramme zur Übergabe an die Sicherungsschicht
  - **Empfänger**: liefert ausgepackte Segmente an ein Protokoll der Transportschicht
- Internet Protokoll in **jedem Internet-fähigen Gerät**: Hosts, Router, …
- **Router**:
  - schaut auf die Header-Felder in allen IP-Datagrammen, die ihn passieren
  - bewegt Datagramme von Eingangsports zu Ausgangsports, um sie entlang des Ende-zu-Ende Pfades zu transferieren



Zwei Schlüsselfunktionen der Vermittlungsschicht

Funktionen der Vermittlungsschicht:

- **Forwarding:** Bewegen von Paketen von Eingangslink eines Routers zum passenden Ausgangslink

- **Routing:** Bestimmen der gewählten Route von Quelle bis zum Ziel
  - Routing Algorithmen

Analogie: Eine Reise unternehmen

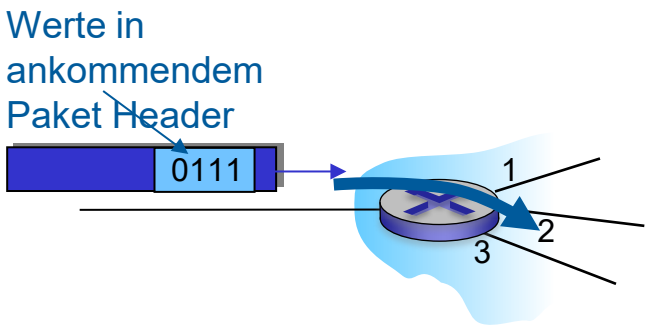
- **Forwarding:** Einen einzelnen Autobahnknoten durchfahren
- **Routing:** Reiseplanung von Startort zum Ziel

Forwarding

Routing

Datenpfad (Data plane):

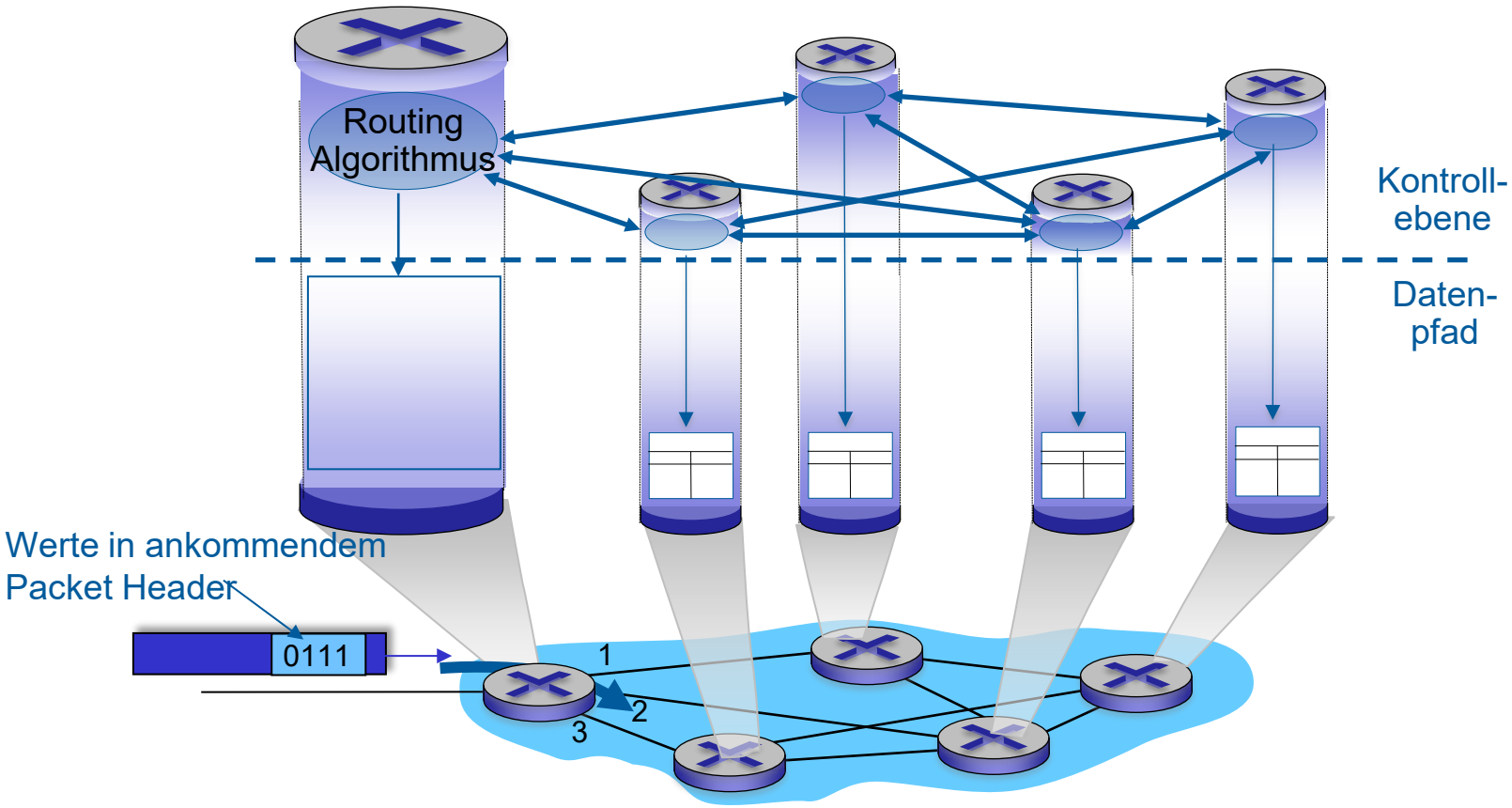
- **lokal**, individuell per Router
- bestimmt wie eine Datagramm von einem Router Eingangsport zum Ausgangsport weitergeleitet wird



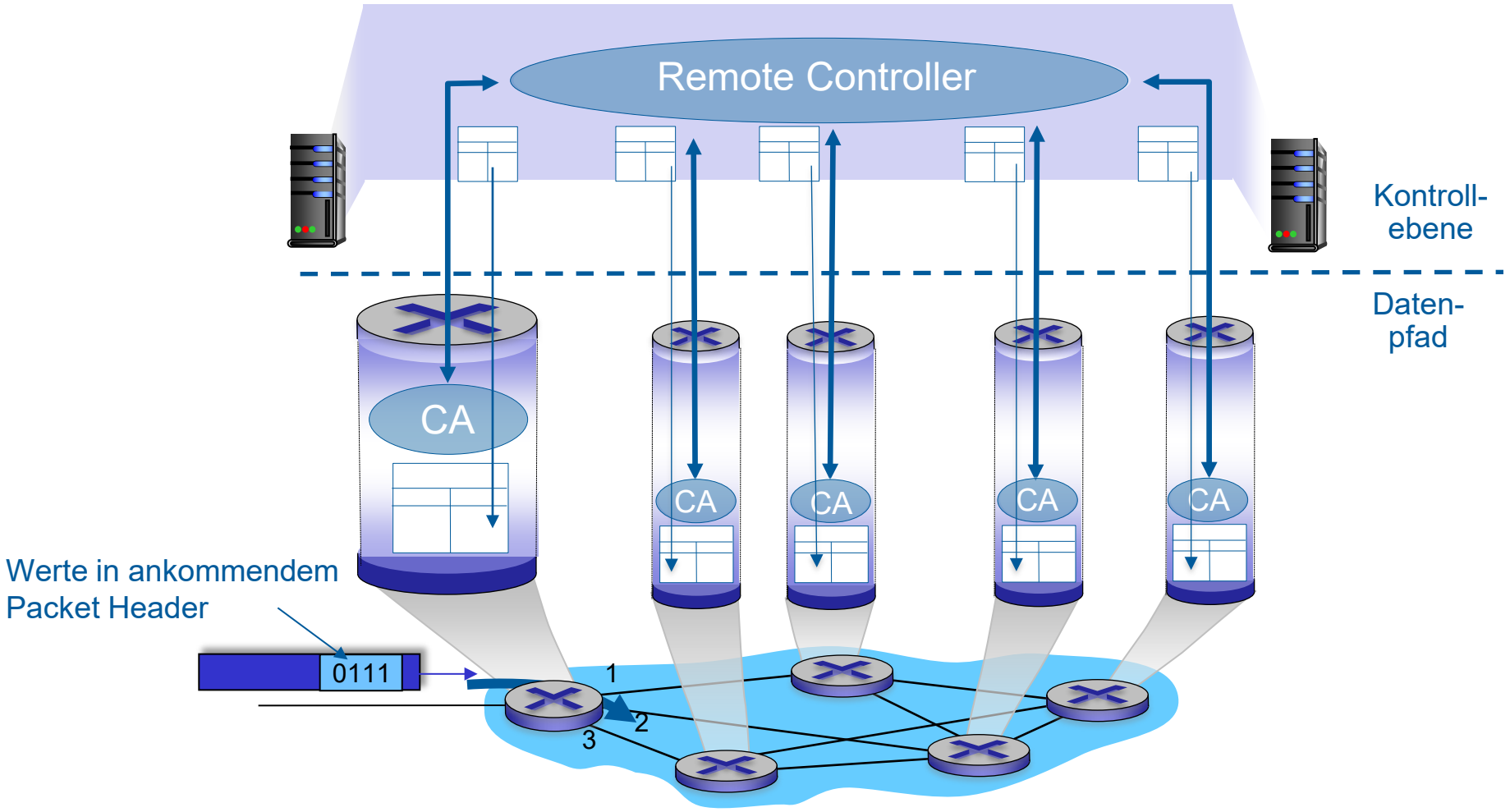
Kontrollebene (Control plane)

- **netzweite** Logik
- Bestimmt wie ein Datagramm zwischen Router entlang des Ende-zu-Ende Pfades von Quell- zu Zielhost geroutet wird
- Zwei Kontrollebenen-Ansätze :
  - **Traditionelle Routing Algorithmen:** implementiert in Routern
  - **Software-defined Networking (SDN):** implementiert auf (entfernten) Servern

Eigenständige Routing Algorithmen Komponenten **in jedem einzelnen Router** interagieren auf der Kontrollebene



Entfernter Controller berechnet und installiert Forwarding Tabellen in Netzelemente (Router)



Frage: Wie sieht das **Dienstmodell** für den “Kanal” aus, der Datagramme von Sender zum Empfänger transportiert?

Beispiel-Dienste für Lieferung einzelner Datagramme:

- Garantierte Lieferung
- Garantierte Lieferung mit weniger als 40 ms Verzögerung

Bespiel-Dienste für einen “Flow” aus Datagrammen:

- Datagramm Auslieferung in Reihenfolge
- Garantierte Minimum Bandbreite für einen Flow
- Einschränkungen auf Veränderungen des Zwischenpaketabstands (Jitter)

Dienstmodell der Vermittlungsschicht

Netz-Architektur	Dienstmodell	Quality of Service (QoS) Garantien ?			
		Bandbreite	Verlust	Reihenfolge	Latenz
Internet	best effort	nein	nein	nein	nein

Internet "best effort" Dienstmodell

Keine Garantien bezüglich*:*

- erfolgreicher Datagramm Lieferung zum Ziel
- Verzögerung und Einhaltung der Paketreihenfolge
- Bandbreite für einen Ende-zu-Ende Flow

Dienstmodell der Vermittlungsschicht

Netz-Architektur	Dienstmodell	Quality of Service (QoS) Garantien ?			
		Bandbreite	Verlust	Reihenfolge	Latenz
Internet	best effort	nein	nein	nein	nein
ATM	Konstante Bit Rate	Konstante Rate	ja	ja	ja
ATM	Verfügbar Bit Rate	Garantiertes Min.	nein	ja	nein
Internet	Intserv garantiert (RFC 1633)	ja	ja	ja	ja
Internet	Diffserv (RFC 2475)	möglich	möglich	möglich	nein

Einigen Gedanken zu Best-Effort Diensten:

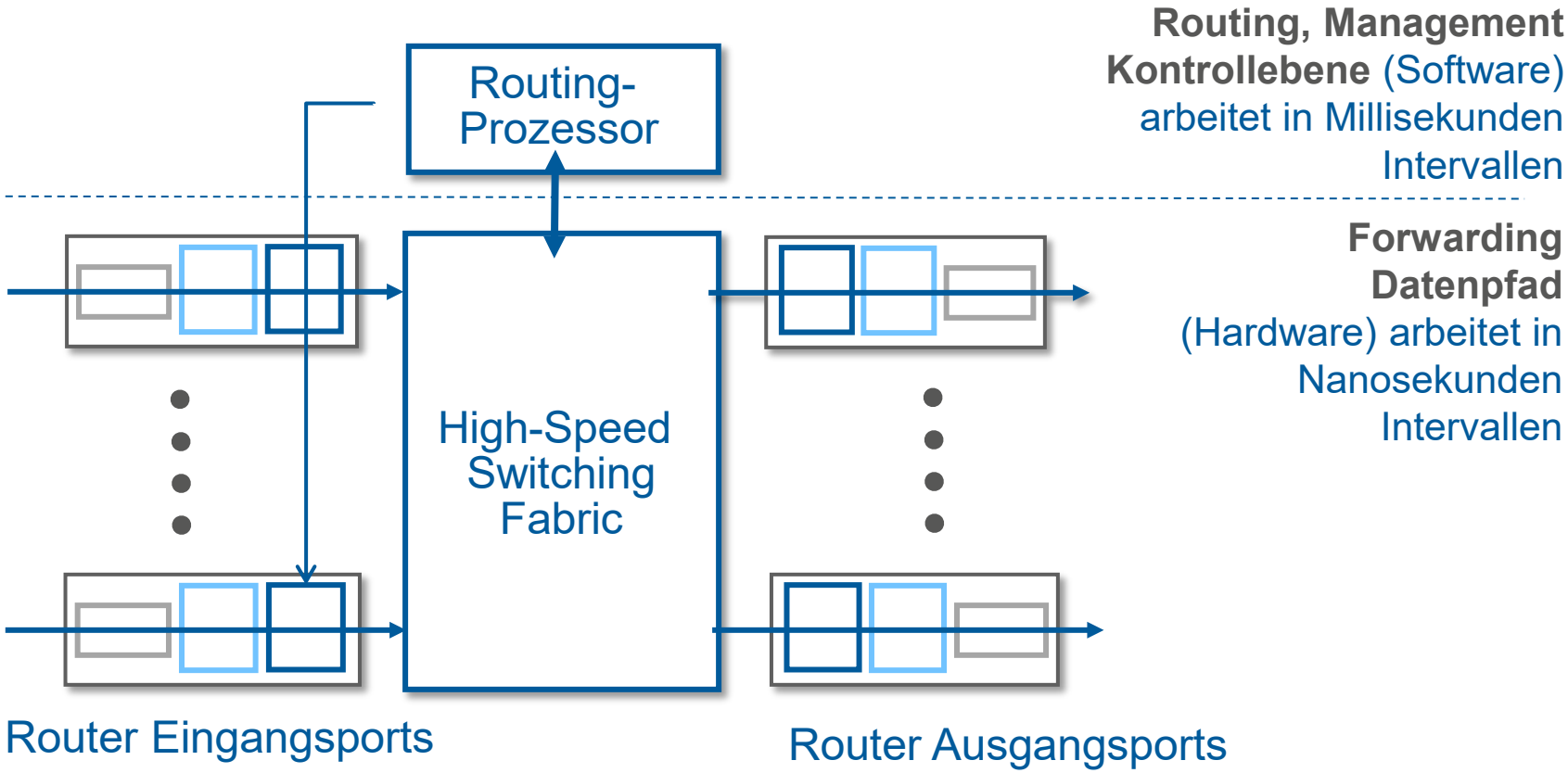
- Die **Einfachheit des Mechanismus** hat die schnelle Verbreitung des Internets ermöglicht
- Das zur **Verfügung stellen von ausreichend Bandbreite** sorgt dafür, dass die Leistung von Echtzeitapplikationen (z.B., interaktive Sprache, Video) “meistens gut genug” ist”
- **Verteilte, replizierte Dienste der Applikationsschicht** (Datenzentren, Content Distribution Networks), die sich nahe bei Endnutzer Netzen befinden, erlauben es, dass Dienste von mehreren Orten zur Verfügung gestellt werden
- Überlastkontrolle von “elastischen” Diensten hilft

Es ist schwer den Erfolg der best-effort Dienstmodells zu bestreiten!

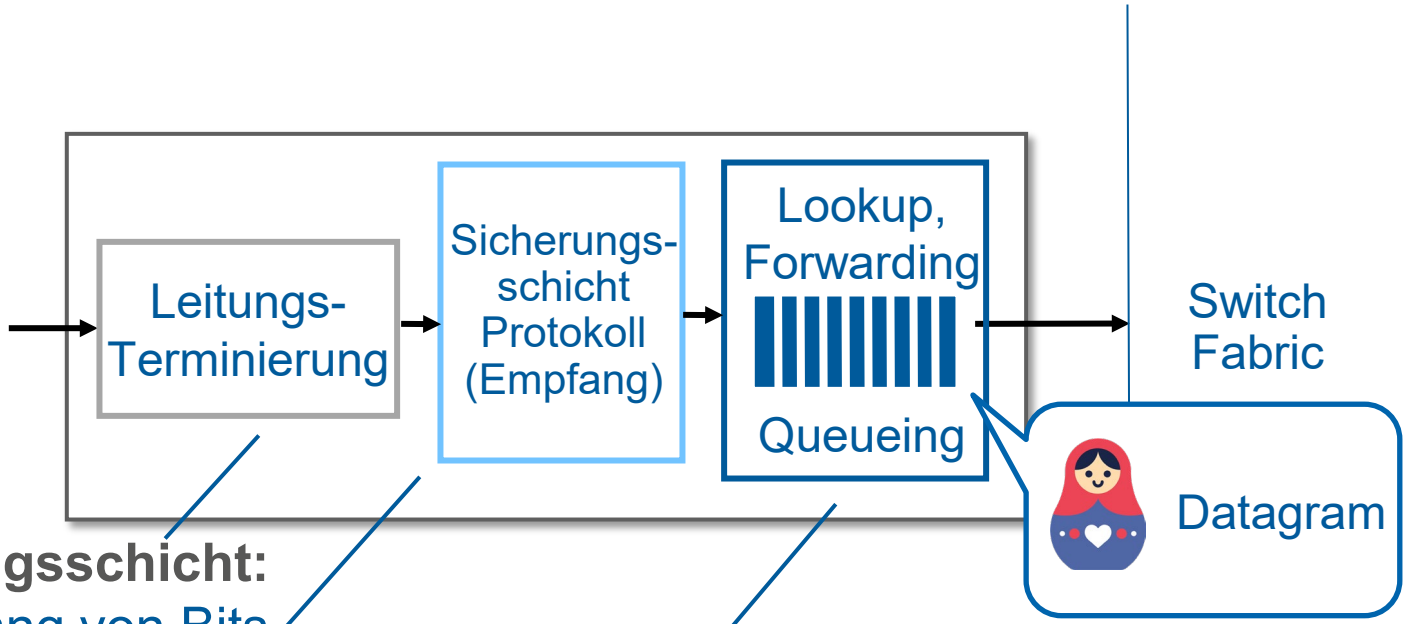
Übersicht: Vermittlungsschicht - Datenpfad

- **Vermittlungsschicht: Überblick**
  - Datenpfad
  - Kontrollebene
- **Was steckt in einem Router**
  - Eingangsports, Switching, Ausgangsports
  - Puffer Management, Scheduling
- **IP: das Internet Protokoll**
  - Datagramm Format
  - Adressierung
  - Network Address Translation
  - IPv6
- **Generalized Forwarding, SDN**
  - Match+Action
  - OpenFlow: Match+Action
- **Mittelboxen**
  - Funktionen von Mittelboxen
  - Evolution & Prinzipien der Internet Architektur

Grobdarstellung einer generischen Router-Architektur:



Funktionen von Eingangsports



Bitübertragungsschicht:
Empfang von Bits

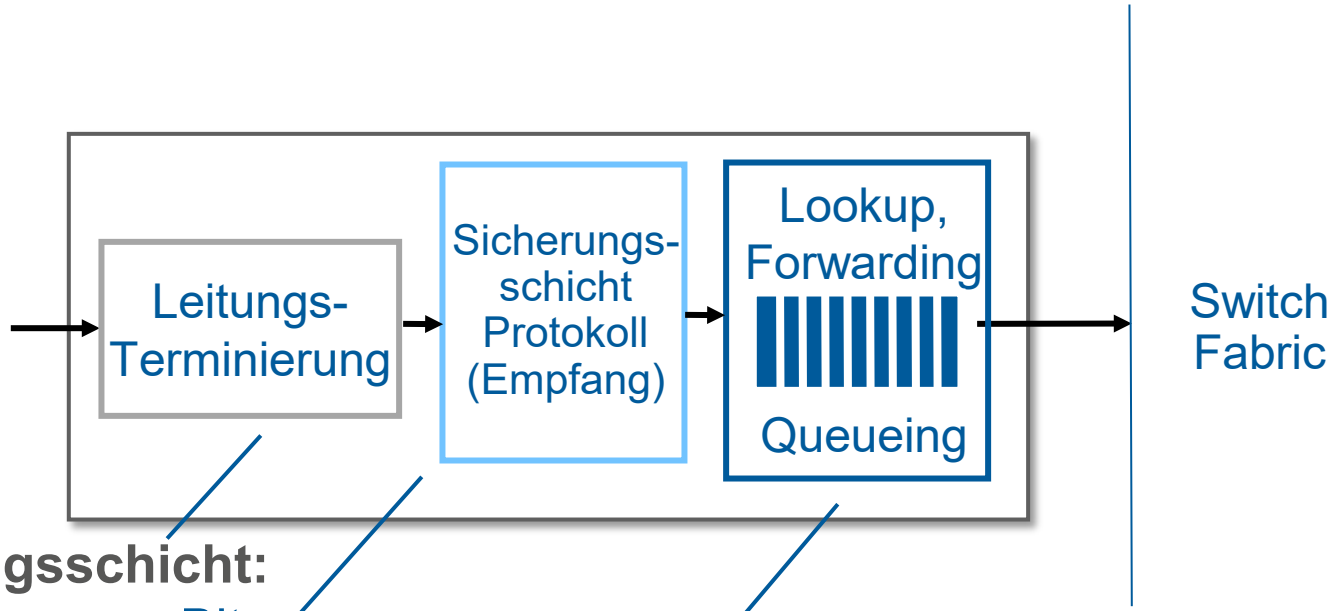
Sicherungsschicht:
z.B., Ethernet

dezentrales Switching*:*

- Auslesen von Header-Feldern, nachschlagen des Ausgangsports in der Forwarding Tabelle im Eingangsport Speicher (“*Match* + *Action*”)
- Ziel: Eingangsport Verarbeitung mit ‘Leitungsgeschwindigkeit’
- **Eingangsport Queuing:** passiert, wenn Datagramme schneller ankommen, als sie durch das Switching Fabric weitergeleitet werden können

Rahmen

Funktionen von Eingangsports



Bitübertragungsschicht:
Empfang von Bits

Sicherungsschicht:
z.B., Ethernet

dezentrales Switching:

- Auslesen von Header-Feldern, nachschlagen des Ausgangsports in der Forwarding Tabelle im Eingangsport Speicher ("*Match* + *Action*")
- **Ziel-basiertes Forwarding:** weiterleiten ausschließlich basierend auf der Ziel IP Adresse (traditionell)
- **Generalisiertes Forwarding:** weiterleiten basierend auf einer beliebigen Kombination von Header-Feldern

Ziel-basiertes Forwarding

forwarding table

Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00010000 00000100 through 11001000 00010111 00010000 00000111	3
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

?

längstes Präfix Match

Bei der Suche nach einem passenden Eintrag in der Forwarding-Tabelle, nutze das **längste** Adress-Präfix, dass mit der Zieladresse übereinstimmt.

Ziel-Adressbereich	Port
11001000 00010111 00010*** ********	0
11001000 00010111 00011000 ********	1
11001000 00010111 00011*** ********	2
ansonsten	3

Beispiele:

11001000 00010111 00010110 10100001 Welches Interface?

11001000 00010111 00011000 10101010 Welches Interface?

längstes Präfix Match

Bei der Suche nach einem passenden Eintrag in der Forwarding-Tabelle, nutze das **längste** Adress-Präfix, dass mit der Zieladresse übereinstimmt.

Ziel-Adressbereich	Port
11001000 00010111 00010*** ********	0
11001000 00010111 00011000 ********	1
1100 Übereinstimmung! 1*** ********	2
ansonsten	3

Beispiele:

11001000 00010111 00010110 10100001 Welches Interface?

11001000 00010111 00011000 10101010 Welches Interface?

längstes Präfix Match

Bei der Suche nach einem passenden Eintrag in der Forwarding-Tabelle, nutze das **längste** Adress-Präfix, dass mit der Zieladresse übereinstimmt.

Ziel-Adressbereich	Port
11001000 00010111 00010*** ********	0
11001000 00010111 00011000 ********	1
11001000 00010111 00011*** ********	2
ansonsten	3

Übereinstimmung!

Beispiele:

11001000 00010111 00010110 10100001 Welches Interface?

11001000 00010111 00011000 10101010 Welches Interface?

längstes Präfix Match

Bei der Suche nach einem passenden Eintrag in der Forwarding-Tabelle, nutze das **längste** Adress-Präfix, dass mit der Zieladresse übereinstimmt.

Ziel-Adressbereich	Port
11001000 00010111 00010*** ********	0
11001000 00010111 00011000 ********	1
11001000 00010111 00011*** ********	2
ansons	3

Übereinstimmung!

Beispiele:

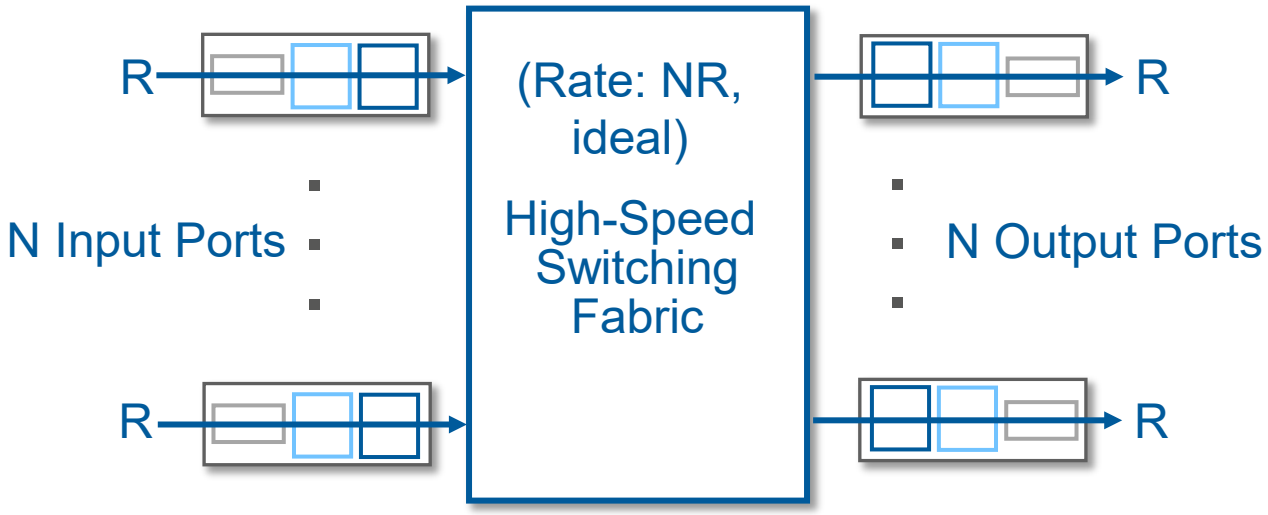
11001000 00010111 00010110 10100001 Welches Interface?

11001000 00010111 00011000 10101010 Welches Interface?

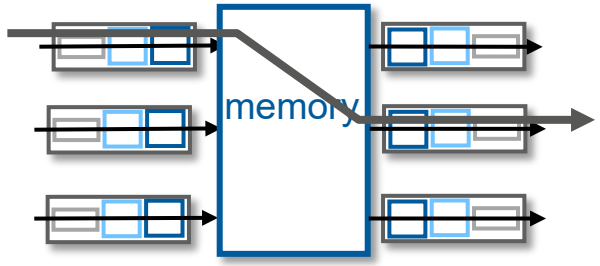
Längstes Präfix Matching

- Wir werden bei der Adressierung sehen **warum** längstes Präfix Matching verwendet wird
- Längstes Präfix Matching: wird oft mit Ternary Content Addressable Memories (TCAMs) durchgeführt
  - **Adressierbarer Inhalt:** finden der Adresse in einem Taktzyklus, unabhängig von der Tabellengröße
  - Cisco Catalyst: ~1M Einträge in der Routingtabelle im TCAM

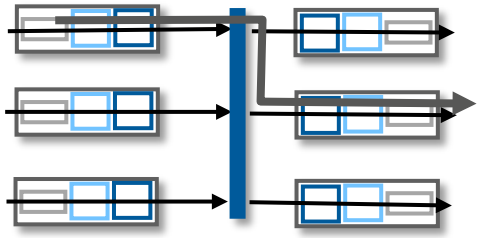
- Transferieren eines Pakets vom Eingangsport zum entsprechenden Ausgangsport
- **Switching Rate:** Rate, mit der Pakete von Eingangs- zu Ausgangsports transferiert werden könne
  - Oft als Vielfaches der Eingangs/Ausgangsverbindungsrate angegeben
  - N Eingänge: gewünschte Switching Rate entspricht N Mal der Verbindungsrate



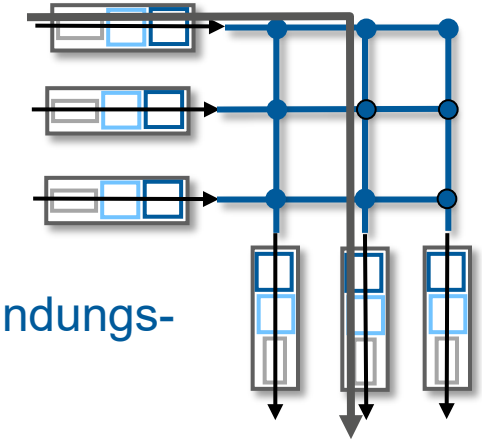
- Transferieren eines Pakets vom Eingangsport zum entsprechenden Ausgangsport
- **Switching Rate:** Rate, mit der Pakete von Eingangs- zu Ausgangsports transferiert werden können
  - Oft als Vielfaches der Eingangs/Ausgangsverbindungsrate angegeben
  - N Eingänge: gewünschte Switching Rate entspricht N Mal der Verbindungsrate
- Drei übergeordnete Arten von Switching Fabrics:



Speicher



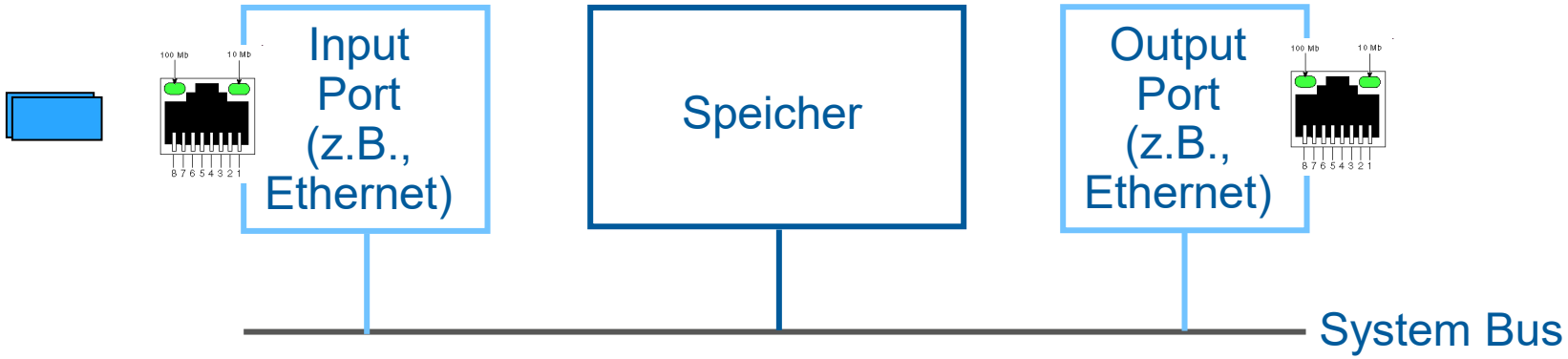
Bus



Verbindungs-
netz

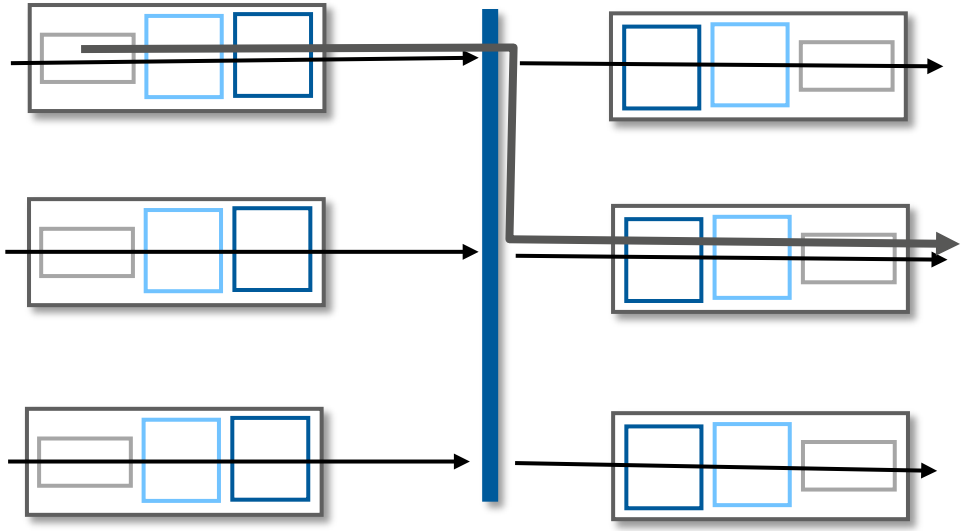
Router der 1. Generation:

- traditionelle Computer mit Switching unter direkter Kontrolle der CPU
- Paket wird in den Arbeitsspeicher kopiert
- Geschwindigkeitsbegrenzung durch Speicherbandbreite (2 Bus Durchläufe pro Datagramm)



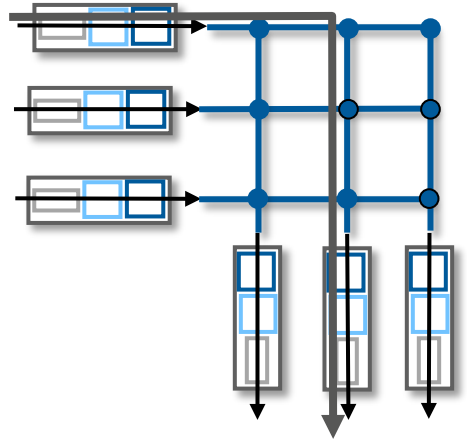
Switching über einen Bus

- Datagramm wird vom Eingangsport-Speicher über einen gemeinsamen Bus zum Ausgangsport-Speicher übertragen
- **Bus Konkurrenz:** Switching Geschwindigkeit durch Busbandbreite eingeschränkt
- 32 Gbit/s Bus, Cisco 5600: ausreichende Geschwindigkeit für Router im Zugangsbereich



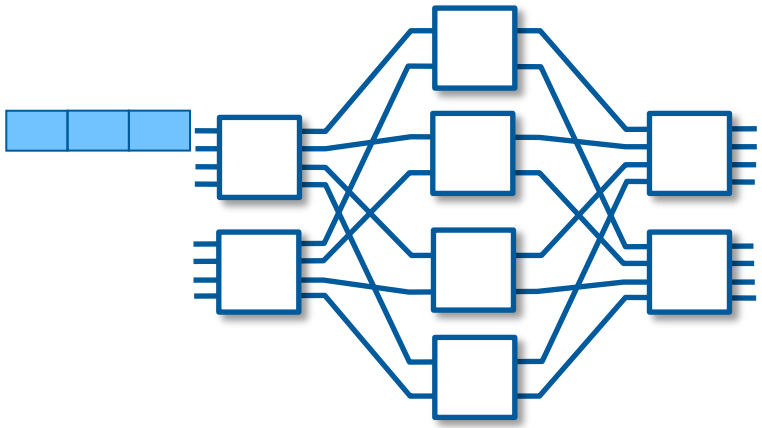
Switching über ein Verbindungsnetz

- Koppelnetze, Clos Netze, und andere Verbindungsnetze wurden ursprünglich entwickelt um mehrere Prozessoren in Multiprozessor-Systemen zu verbinden



3x3 Koppelnetz

- Mehrstufiger Switch: **nxn** Switch aus mehreren Stufen von kleineren Switches
- **Nutzen von Parallelisierung:**
  - Fragmentieren des Datagramms in Zellen fester Länge am Eingang
  - Switchen der Zellen durch das Fabric, zusammensetzen des Datagramms am Ausgang

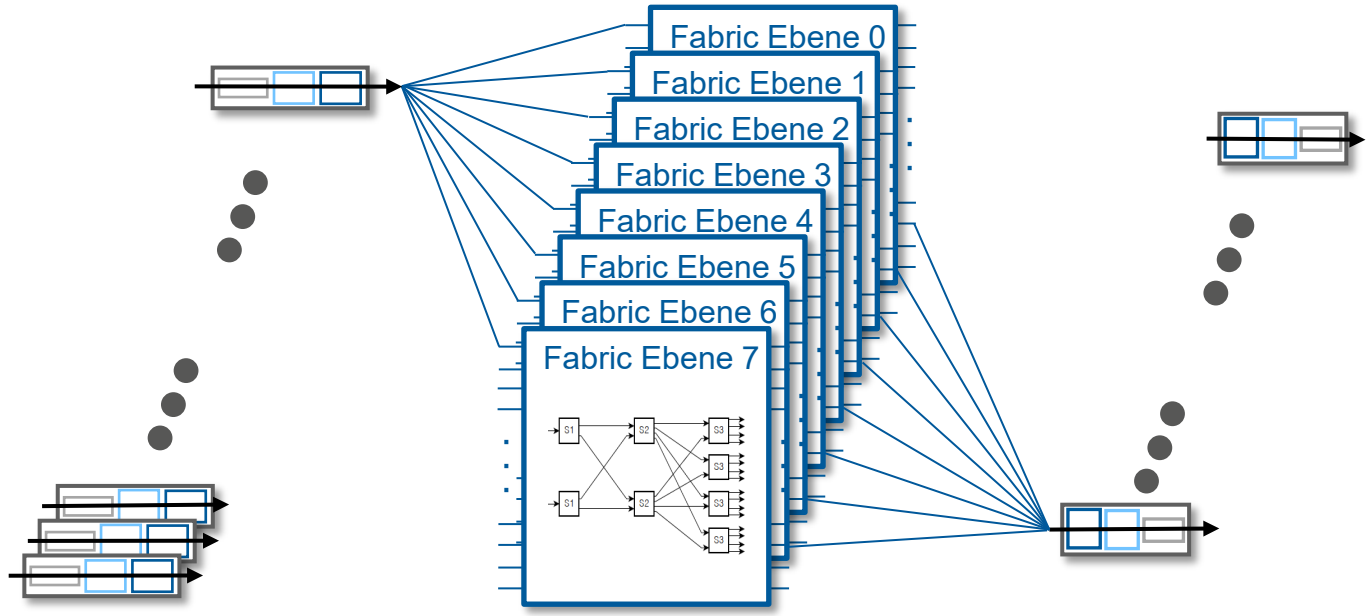


8x8 Mehrstufen-Switch
Bestehend aus kleineren Switches

Switching über ein Verbindungsnetz

- Skalieren durch paralleles nutzen mehrerer Switching “Ebenen” :
- Beschleunigung, Skalierung durch Parallelisierung

- Cisco CRS Router:
  - Standardmodell:
    8 Switching Ebenen
  - Jede Ebene: 3-stufiges Verbindungsnetz
  - Bis zu 100ten Tbit/s Switching Kapazität

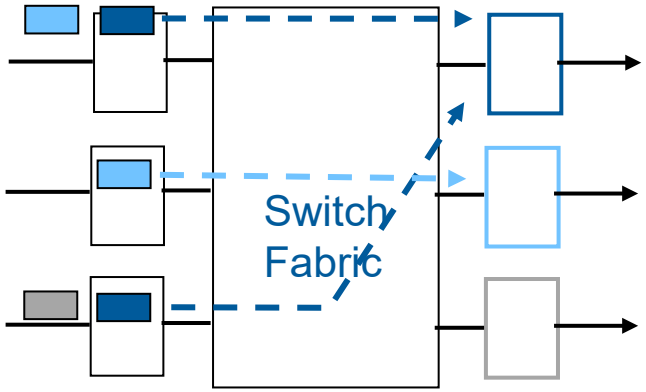


Eingangsport Queueing

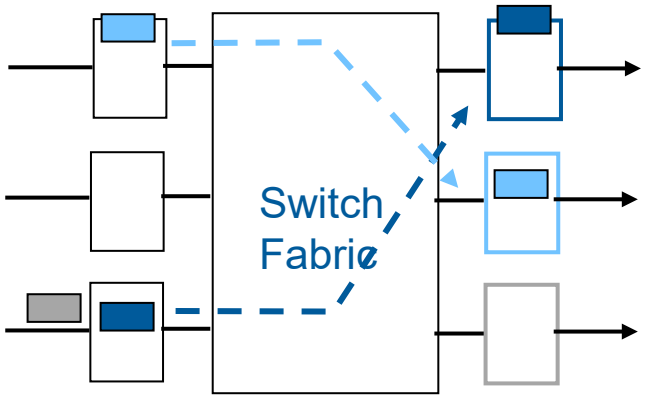
- Wenn das Switch Fabric langsamer ist als alle Eingangsports zusammen

  ➔ es kann zu Wartezeiten an den Eingangsport-Puffern kommen

  - Warteschlangenverzögerung und Paketverlust durch Pufferüberlauf
- **Head-of-the-Line (HOL) Blocking:** wartendes Datagramm am Anfang der Warteschlange blockiert das weiterleiten der anderen wartenden Pakete

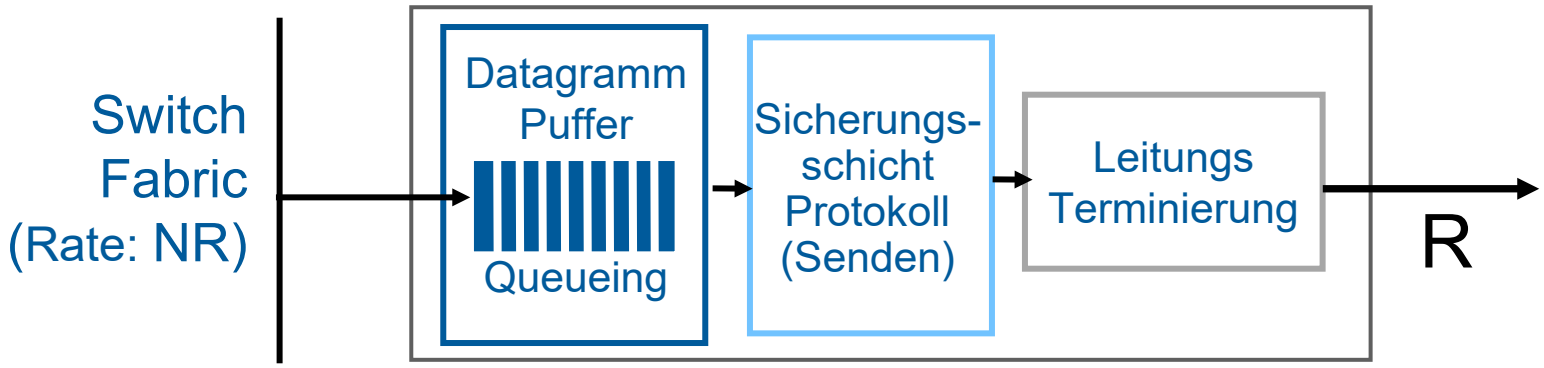


Ausgangsport Konkurrenz: nur ein dunkelblaues Paket kann übertragen werden. Das untere Paket wird *blockiert*



Einen Takt später: dem grauen Paket widerfährt HOL Blocking

Ausgangsport Queueing



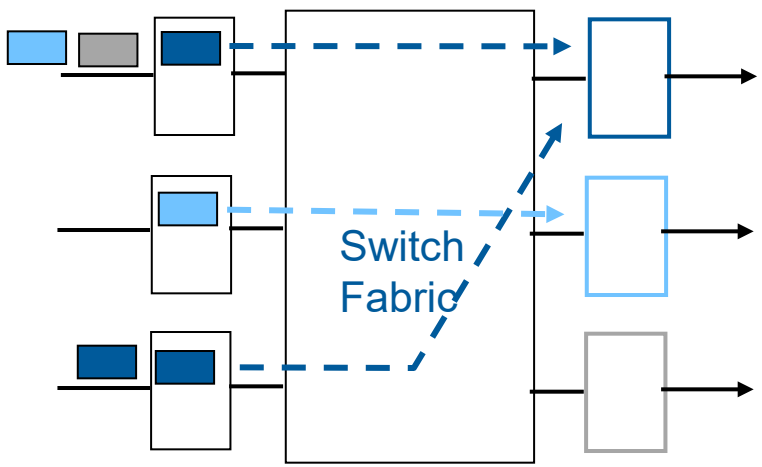
- **Puffern** erforderlich, wenn Datagramme mit einer höheren Rate aus dem Fabric ankommen als die Senderate des Links.
  Verwurfs-Policy: Welche Datagramme sollen verworfen werden, wenn kein Platz im Puffer frei ist?

→ Datagramme können auf Grund von Stau, fehlenden Pufferkapazitäten verloren gehen

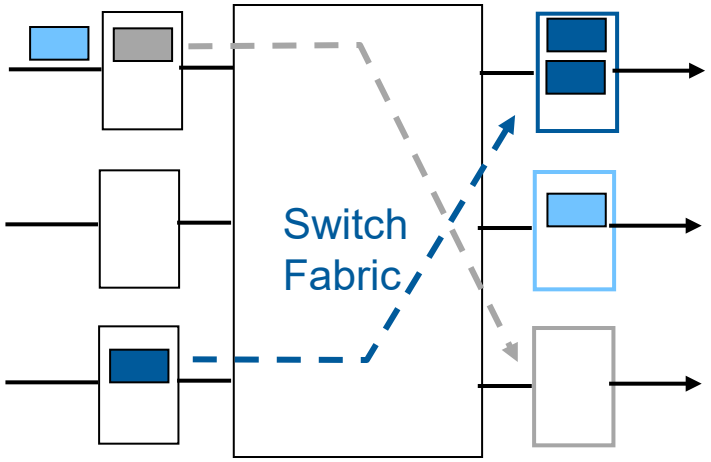
- **Scheduling Disziplin** Auswahl unter den zu versendenden Datagrammen

→ Prioritäts-Scheduling – wer bekommt die beste Leistung?
➔ Netzneutralität

Ausgangsport Queueing



zur Zeit *t,* mehr Pakete vom Eingang- zum Ausgangsport



einen Takt später

- Puffern wenn die Ankunftsrate, die Senderate des Links überschreitet
- **Warteschlangenverzögerung Verlust auf Grund von Überlaufen des Ausgangspuffers**

Wieviel soll gepuffert werden?

- RFC 3439 Faustregel: Durchschnittspuffergröße entspricht einer “typischen” RTT

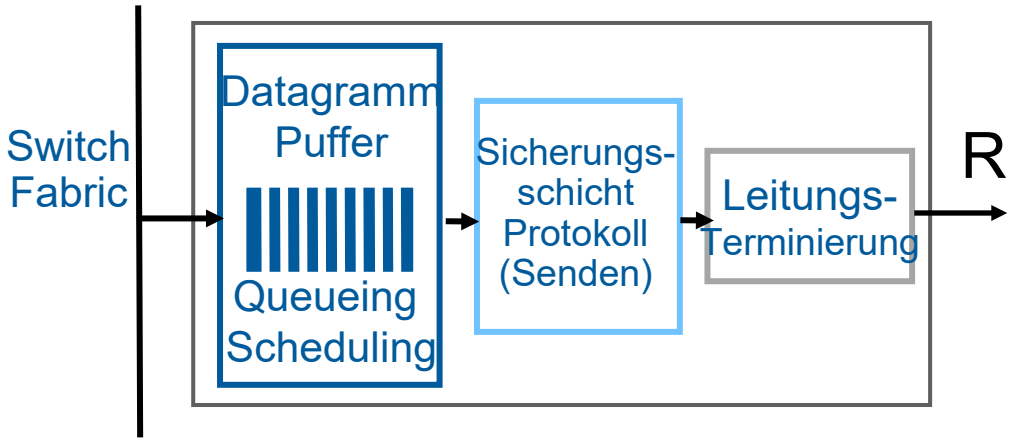
  (z.B. 250 ms) mal der Linkkapazität C

  - z.B., C = 10 Gbit/s Link: 2,5 Gbit Puffer
  - Neuere Empfehlung: bei *N* Flows, soll die Puffergröße folgender Formel entsprechen:

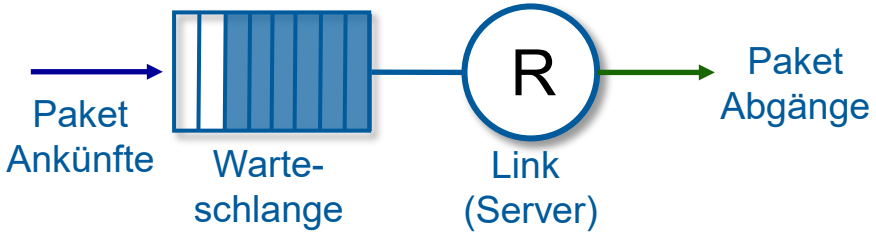
$$\frac{RTT \cdot C}{\sqrt{N}}$$

  - *zu viel* Puffern kann die Verzögerung erhöhen (besonders in Heimroutern)
    - lange RTTs: schlechte Performance für Echtzeitanwendungen

Puffer Management



Abstraktion: Warteschlange



Puffer Management:

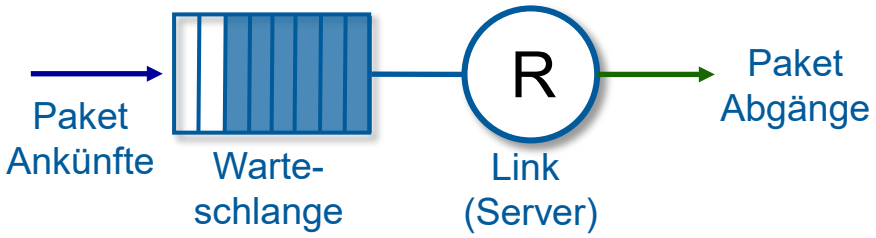
- **Verwerfen:** welches Paket kommt in die Warteschlange, welches wird verworfen, wenn der Puffer voll läuft?
  - **Tail Drop:** ankommende Paket wird verworfen
  - **Priorisierung:** Verwerfen von Paketen auf Basis von Prioritäten
- **Markieren:** welche Pakete sollen markiert werden, um Überlast zu signalisieren (ECN, RED)

Paket Scheduling: FCFS

Paket Scheduling: entscheiden, welches Paket als nächstes auf den Link kommt

- First Come, First Served
- Priorität
- Round Robin
- Weighted Fair Queueing

Abstraktion: Warteschlange



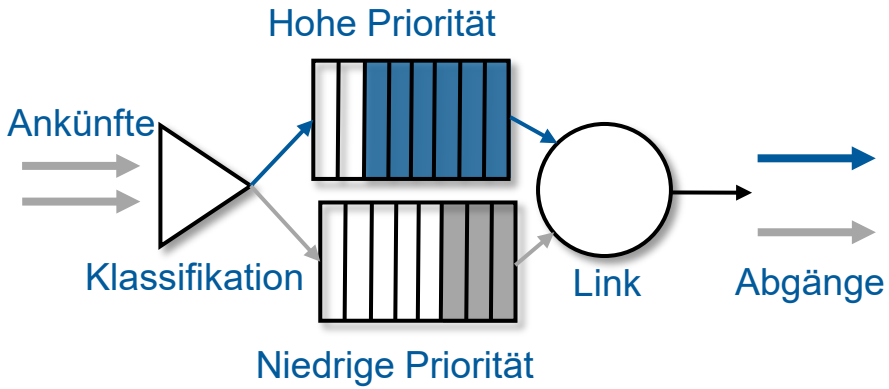
FCFS: Pakete werden in der Ankunftsreihenfolge zum Ausgangsport übertragen

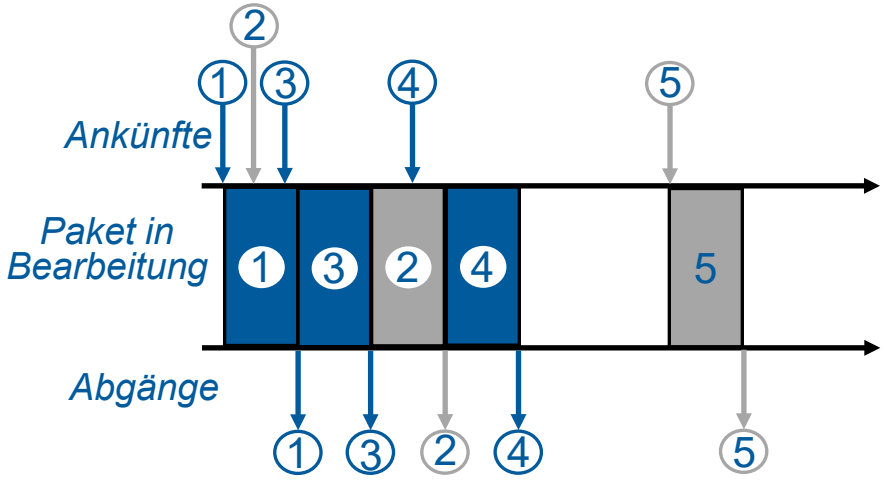
- Auch bekannt als: First-in-first-out (FIFO)
- Beispiele aus dem echten Leben?

Scheduling Policies: Priorität

Prioritäts-Scheduling:

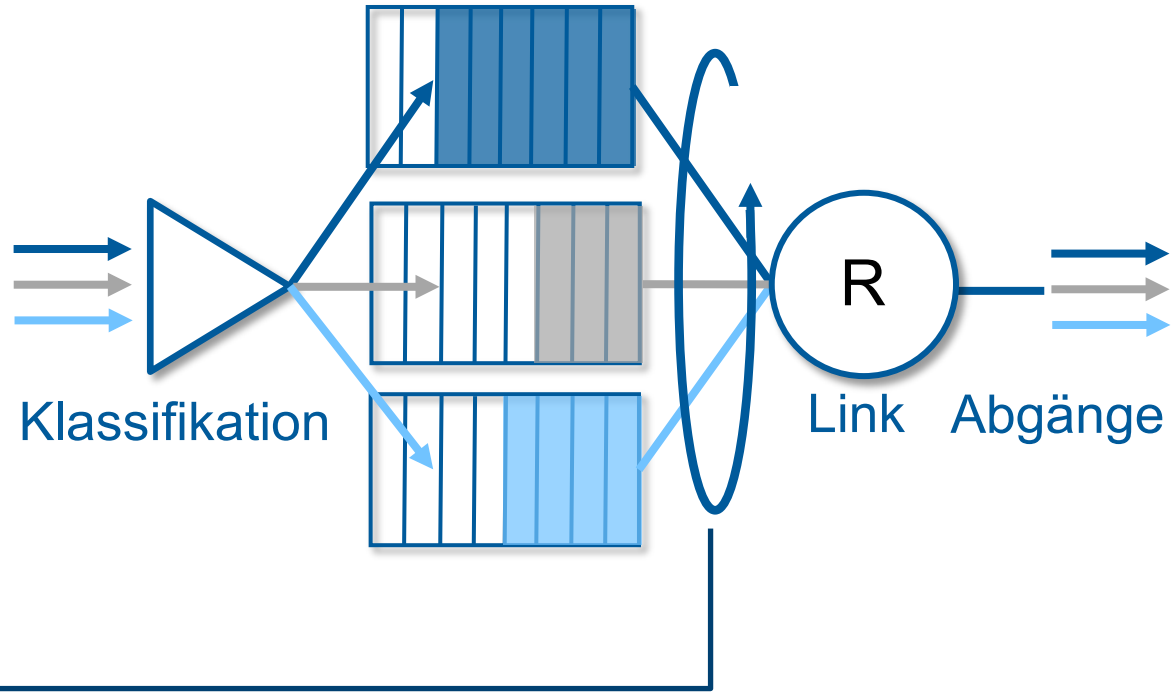
- Ankommender Verkehr wird klassifiziert und auf Warteschlangen verteilt
  - Beliebiger Header-Felder können für die Klassifikation benutzt werden
- Versenden eines Pakets aus Warteschlange mit der höchsten Priorität, in der Pakete warten
  - FCFS mit Prioritätsklassen





Round Robin (RR) Scheduling:

- Ankommender Verkehr wird klassifiziert und auf Warteschlangen verteilt
  - Beliebiger Header-Felder können für die Klassifikation benutzt werden



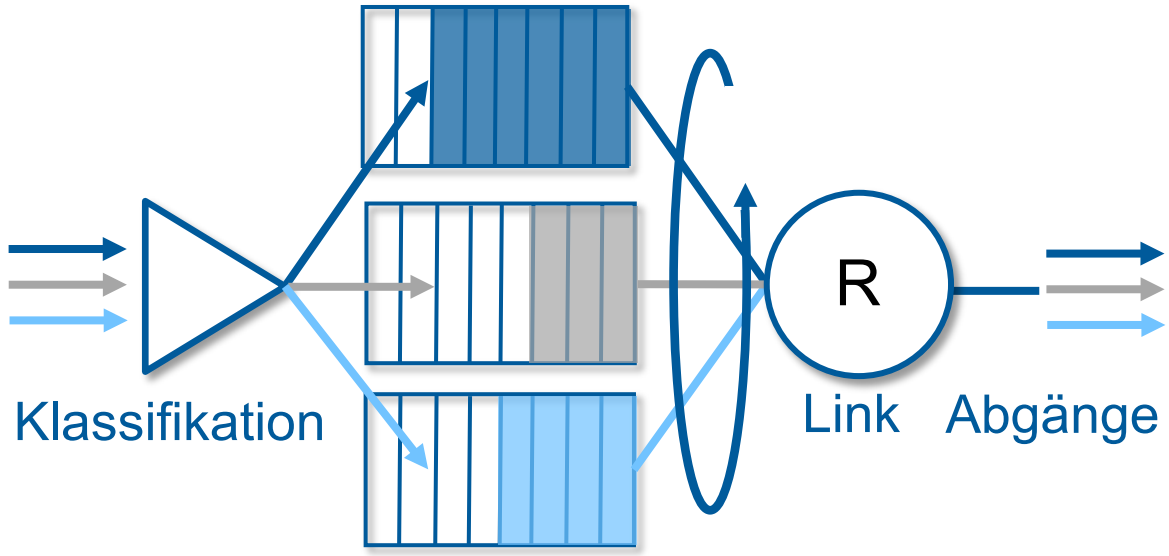
- Die Warteschlangen werden zyklisch betrachtet und jeweils ein Paket bei jedem Besuch versendet, falls eines vorhanden ist

Weighted Fair Queuing (WFQ):

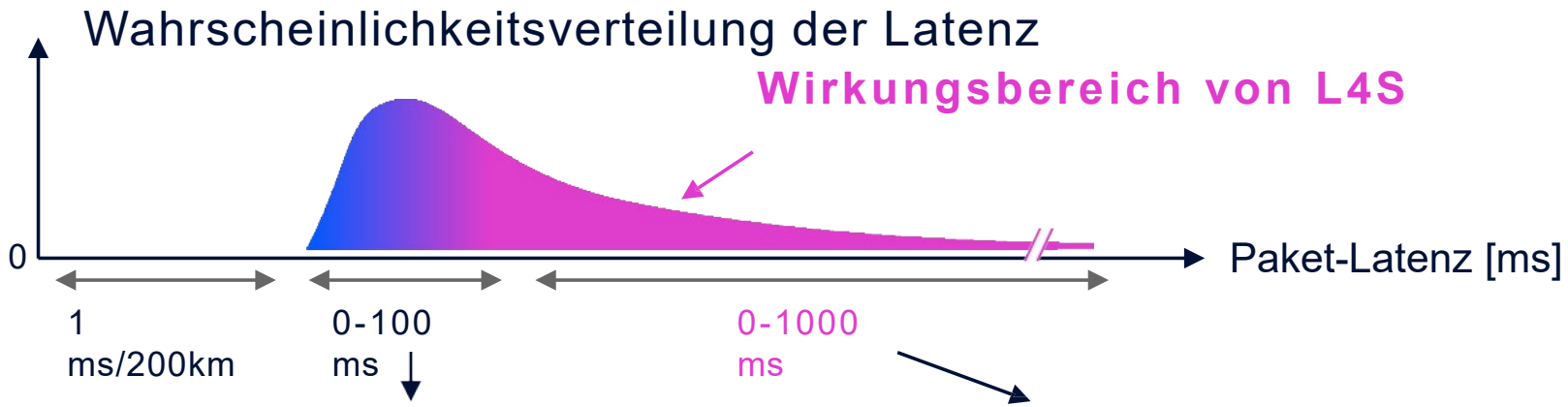
- verallgemeinertes Round Robin
- Jede Klasse, *i*, hat ein Gewicht, w_i, und kann in jedem Zyklus je nach Gewicht eine bestimmte Anzahl von Paketen versenden:

$$\frac{w_i}{\sum_j w_j}$$

- Minimale Bandbreitengarantien (pro Verkehrsklasse)



A new IETF internet protocol to reduce queuing delay to near-zero values



Ausbreitungsverzögerung

- Begrenzt durch Lichtgeschwindigkeit

Schnittstellenverzögerung

- Begrenzt durch Implementierung der unteren Schichten

Warteschlangenverzögerung

- Größte Ursache für Latenzabweichungen, die durch Warteschlangen in Netzwerkpuffern verursacht werden
- Adressierbar durch Vermeidung (GBR/Slicing) oder Verwaltung von Warteschlangenverzögerungen (AQM, IETF L4S)

Was bedeutet es, L4S in einem Netzwerkknoten zu unterstützen?

1. Entfernen von "In-Flight"-Jitter durch Priorisierung

2. Netzwerk-Feedback zur Anpassung der Quellenrate

Klassifikation

Identifizieren von L4S-Datenverkehr

Isolation

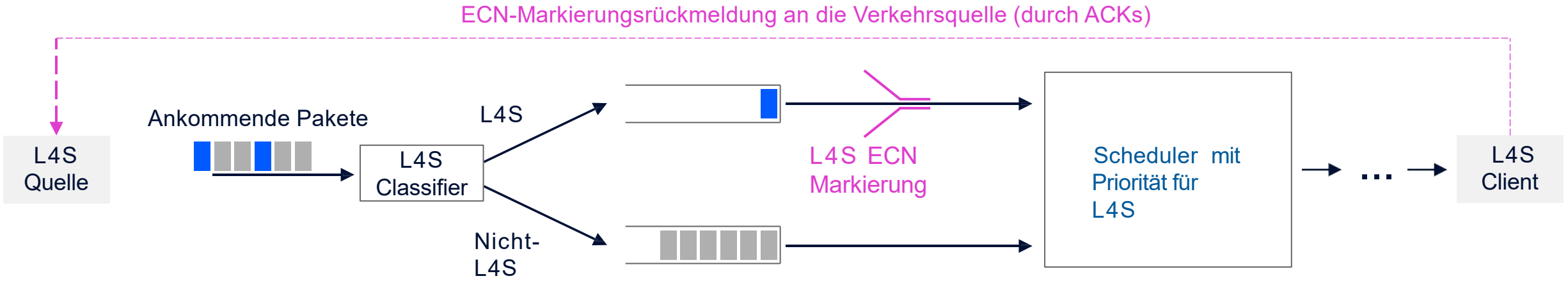
Verwenden separater Warteschlangen für L4S Pakete und Nicht-L4S-Pakete

Priorisierung

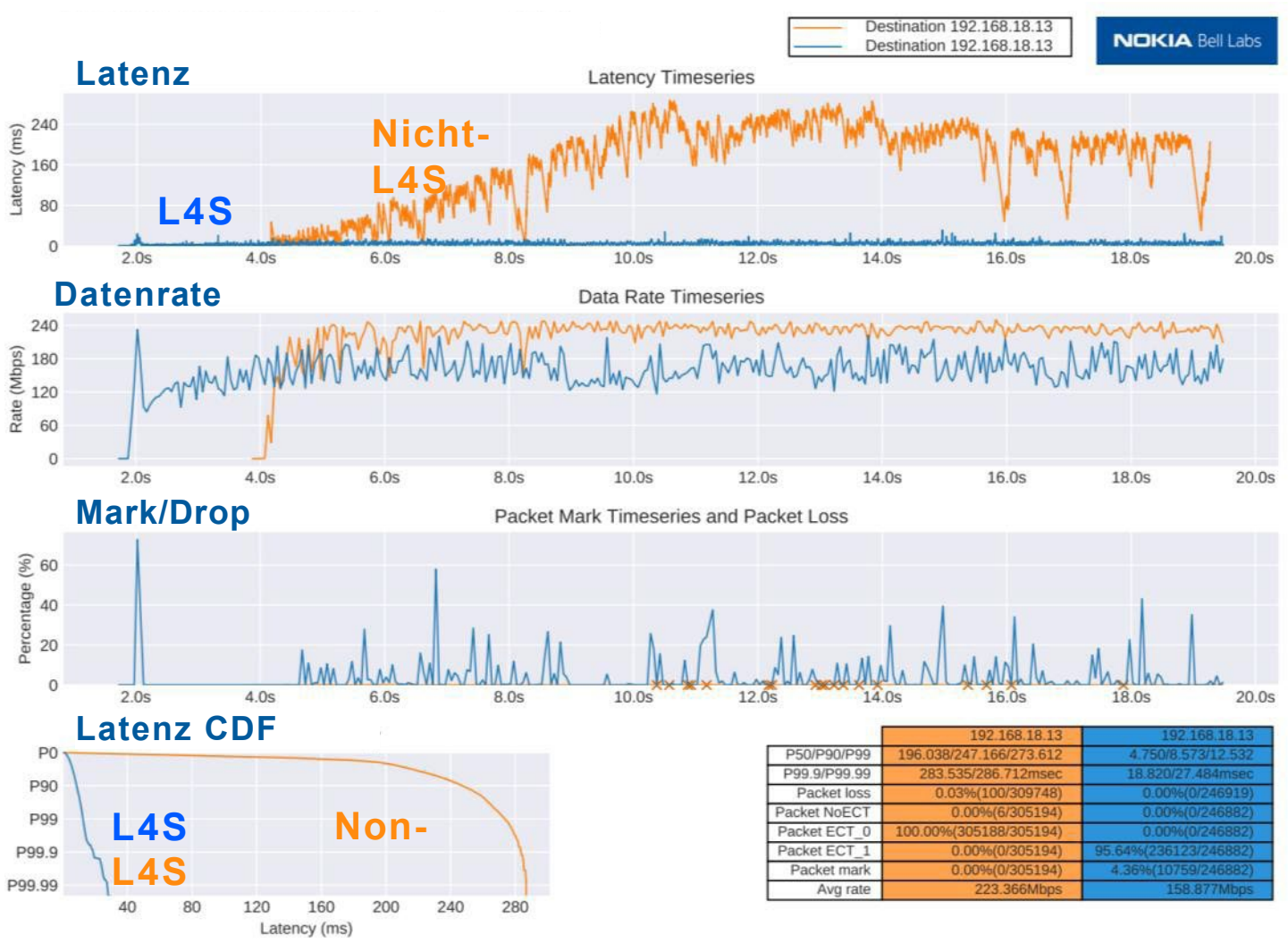
Überspringen von Nicht-L4S-Paketen durch die wenigen L4S-Pakete zulassen

ECN Markierung

Korrekte L4S-ECN-Kennzeichnung für Bewältigung von Ratenengpässen



L4S zeigt >10-fache Verringerung der Spitzenlatenz



	192.168.18.13	192.168.18.13
P50/P90/P99	196.038/247.166/273.612	4.750/8.573/12.532
P99.9/P99.99	283.535/286.712msec	18.820/27.484msec
Packet loss	0.03%(100/309748)	0.00%(0/246919)
Packet NoECT	0.00%(6/305194)	0.00%(0/246882)
Packet ECT_0	100.00%(305188/305194)	0.00%(0/246882)
Packet ECT_1	0.00%(0/305194)	95.64%(236123/246882)
Packet mark	0.00%(0/305194)	4.36%(10759/246882)
Avg rate	223.366Mbps	158.877Mbps

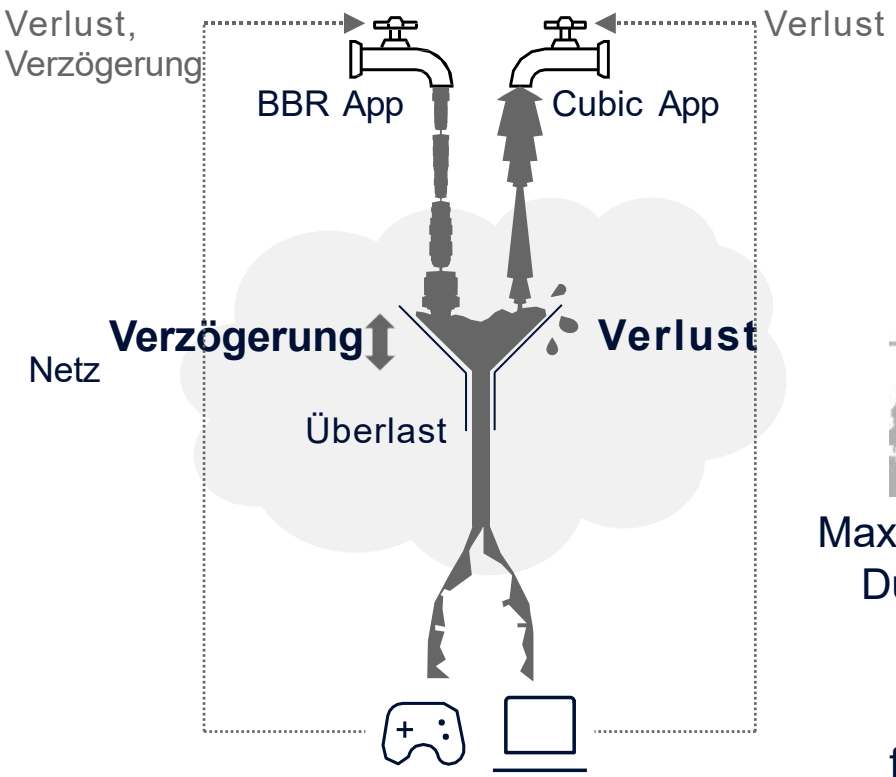
Latenz	P50	P90	P99
Kein AQM	196 ms	247 ms	273 ms
L4S	**4.7 ms**	**8.6 ms**	**12.5 ms**

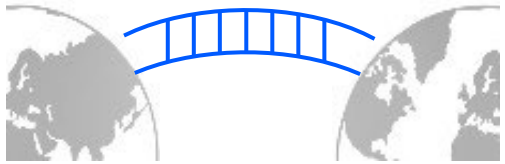
NOKIA BELL LABS

L4S ermöglicht es Anwendungen, zwischen 2 Arten von Datenverkehr zu wählen

Das Netz muss in der Mitte keine Kompromisse eingehen

Klassisch
Pufferung für maximalen Durchsatz



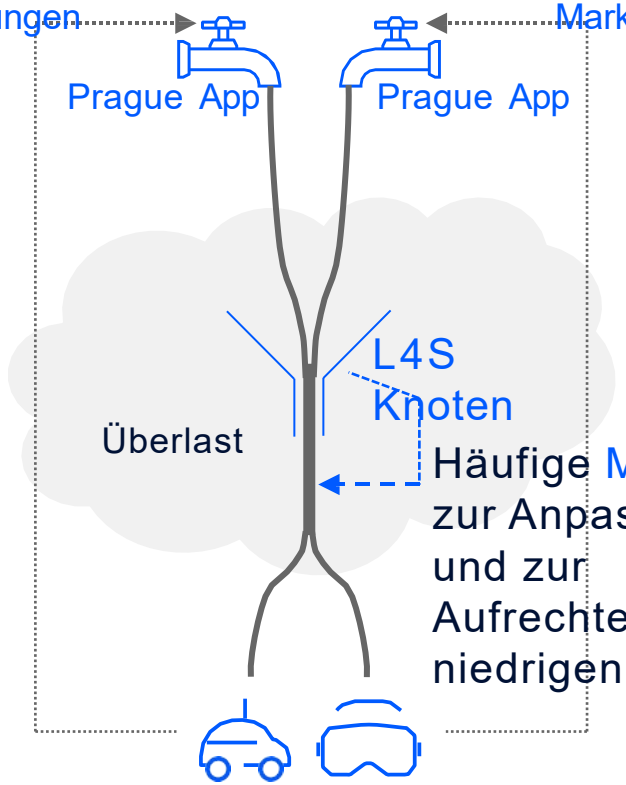


Maximierung des Durchsatzes

Minimierung der Latenz

L4S ermöglicht die friedliche Koexistenz dieser Welten [RFC9332]

L4S
Leere Puffer für **minimale Latenz** [RFC9331]



Häufige Markierungen zur Anpassung der Rate und zur Aufrechterhaltung einer niedrigen Latenz

Was ist Netzneutralität

- **Technisch:** wie ein ISP seine Ressourcen verwenden/teilen sollte
  - Paket Scheduling, Puffer Management sind die **Mechanismen** dafür
- **Soziale, wirtschaftliche** Prinzipien
  - Schutz der Meinungsfreiheit
  - Fördern von Innovation und Konkurrenz
- **Gesetzliche** Regeln und Politik

Verschiedene Länder haben verschieden "Interpretationen" von Netzneutralität.

2015 US FCC *Order on Protecting and Promoting an Open Internet:*
drei klare Regeln:

- **kein Blockieren** … “shall not block lawful content, applications, services, or non-harmful devices, subject to reasonable network management.”
- **kein Einbremsen** … “shall not impair or degrade lawful Internet traffic on the basis of Internet content, application, or service, or use of a non-harmful device, subject to reasonable network management.”
- **keine bezahlte Priorisierung** … “shall not engage in paid prioritization”

REGULATION (EU) 2015/2120

1. **End-users** shall have **the right to access and distribute information** and content, use and provide applications and services, and use terminal equipment of their choice, **irrespective of** the end-user's or provider's **location** or the location, **origin or destination** of the information, **content**, **application or service**, via their internet access service. This paragraph **is without prejudice to Union law**, or national law that complies with Union law, related to the lawfulness of the content, applications or services.

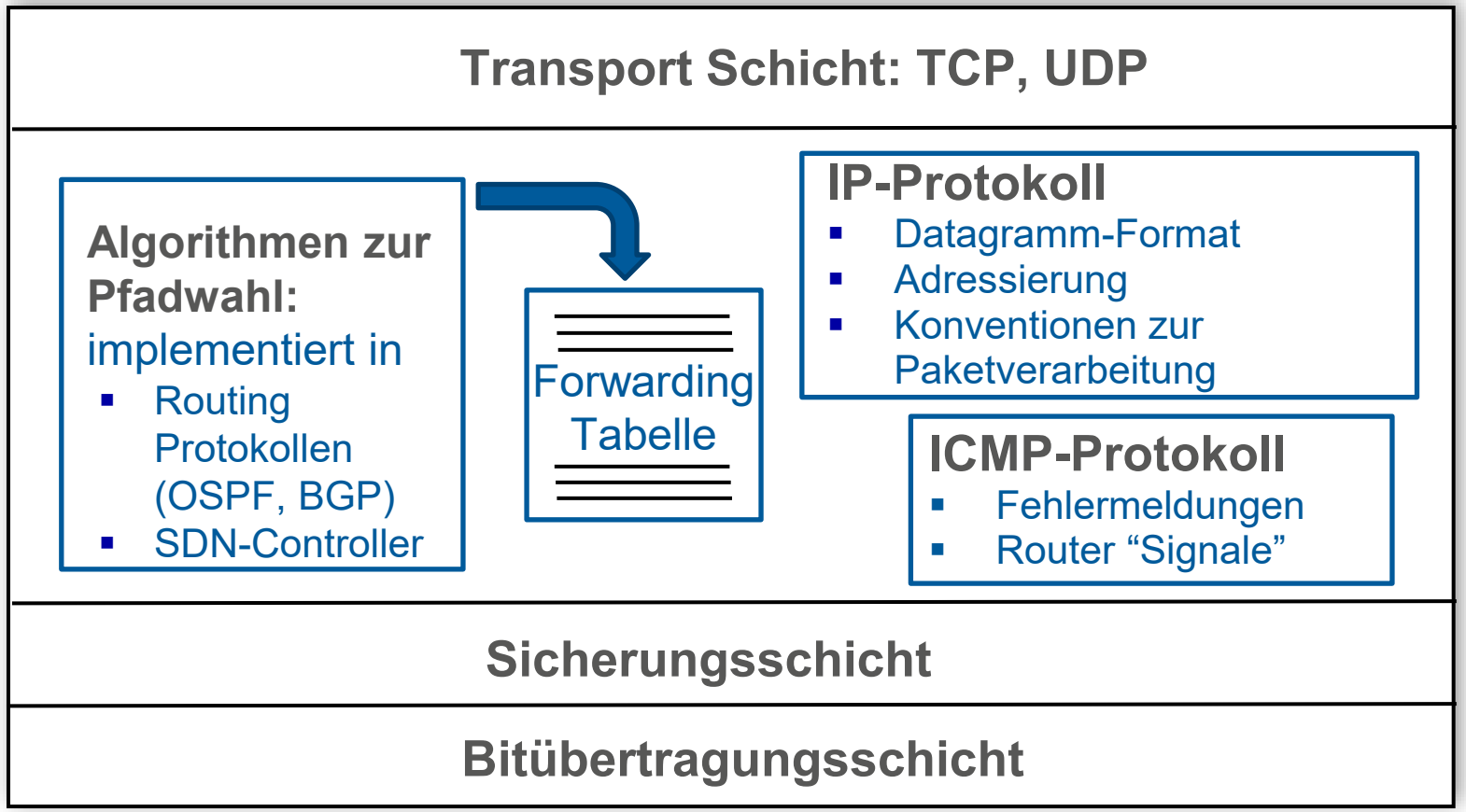
2. Agreements between providers of internet access services and end-users on commercial and technical conditions and the characteristics of internet access services such as price, data volumes or speed, and any commercial practices conducted by providers of internet access services, shall not limit the exercise of the rights of end-users laid down in paragraph 1.

3. **Providers of internet access services shall treat all traffic equally**, when providing internet access services, without discrimination, restriction or interference, and irrespective of the sender and receiver, the content accessed or distributed, the applications or services used or provided, or the terminal equipment used.

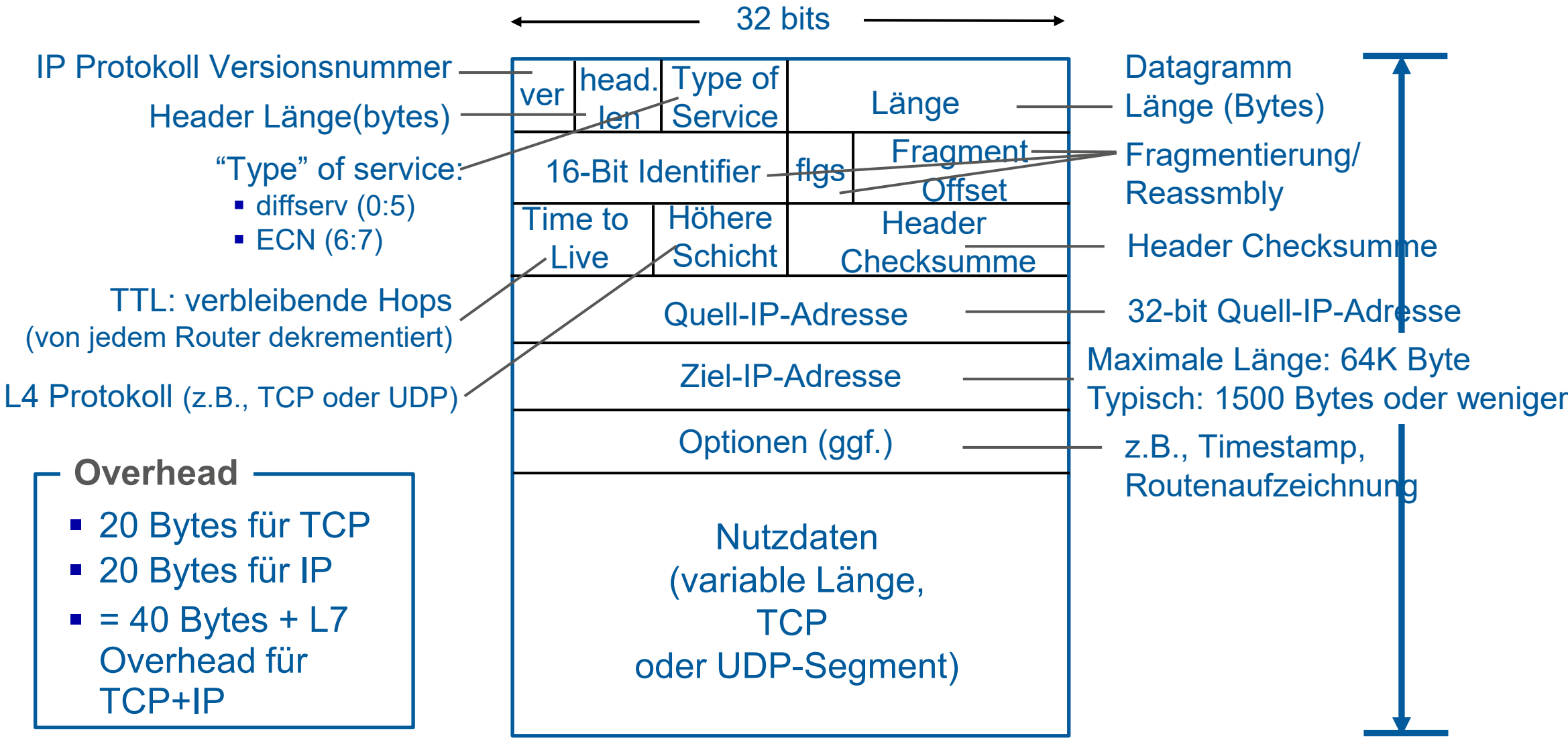
- **Vermittlungsschicht: Überblick**
  - Datenpfad
  - Kontrollebene
- **Was steckt in einem Router**
  - Eingangsports, Switching, Ausgangsports
  - Puffer Management, Scheduling
- **IP: das Internet Protokoll**
  - **Datagramm Format**
  - **Adressierung**
  - Network Address Translation
  - IPv6
- **Generalized Forwarding, SDN**
  - Match+Action
  - OpenFlow: Match+Action
- **Mittelboxen**
  - Funktionen von Mittelboxen
  - Evolution & Prinzipien der Internet Architektur

Funktionen der Vermittlungsschicht in Hosts & Routern:

Vermittlungs-schicht



IP-Datagramm Format

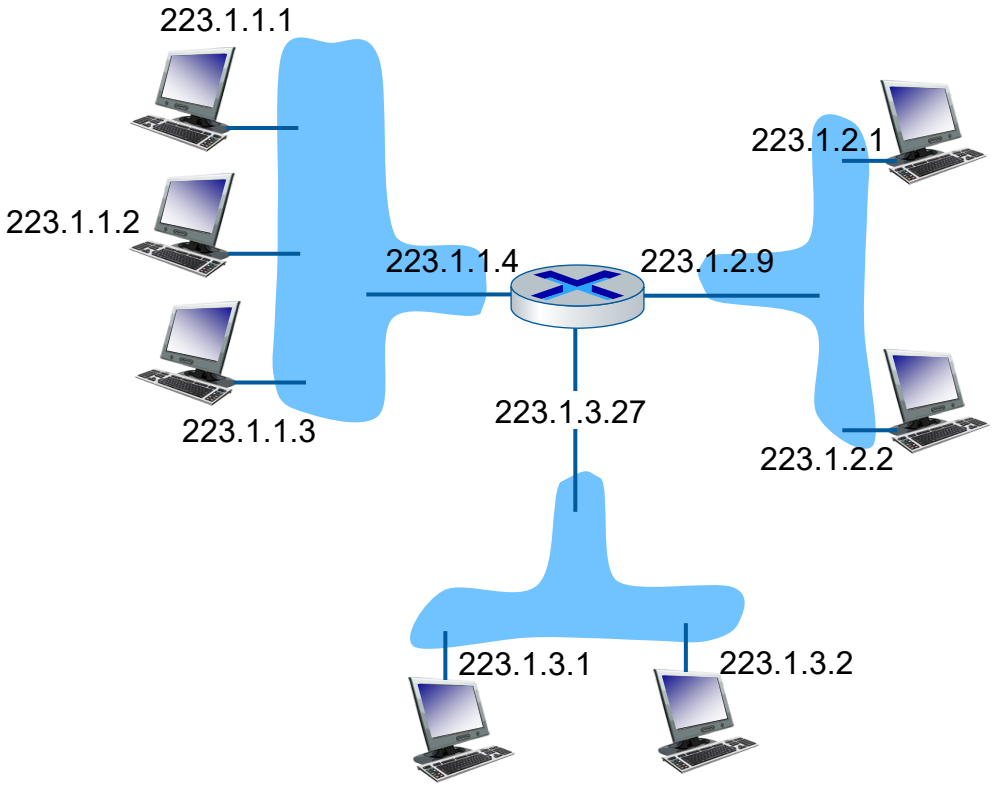


Overhead

- 20 Bytes für TCP
- 20 Bytes für IP
- = 40 Bytes + L7 Overhead für TCP+IP

IP-Adressierung: Einführung

- **IP-Adresse:** 32-bit Identifier assoziiert mit jedem **Interface** von Hosts oder Routern
- **Interface:** Verbindung zwischen Host/Router und physischem Link
  - Router haben normalerweise mehrere Interfaces
  - Ein Host hat normalerweise ein oder zwei Interfaces (z.B., Kabel-Ethernet, 802.11 WLAN)



IP-Adressen Schreibweise:

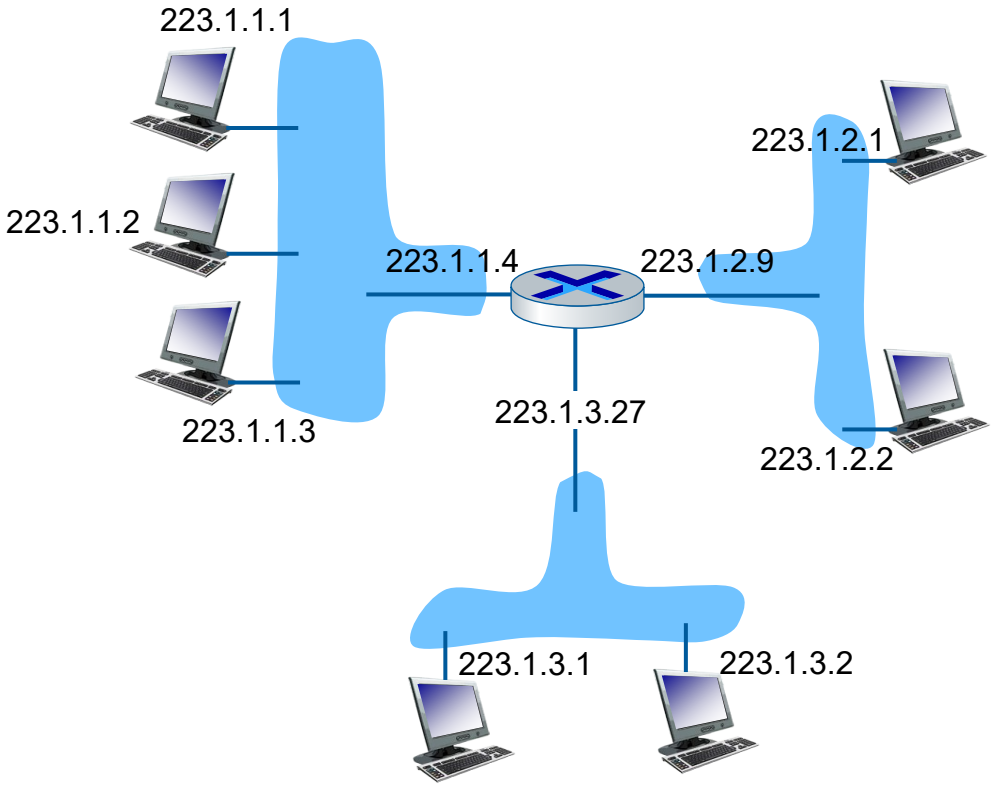
223.1.1.1 = 11011111 00000001 00000001 00000001

223 1 1 1

Subnetze

- **Was ist ein Subnetz?**
  - Geräte-Schnittstellen, die sich gegenseitig physisch erreichen können **ohne einen weiterleitenden Router zu passieren**

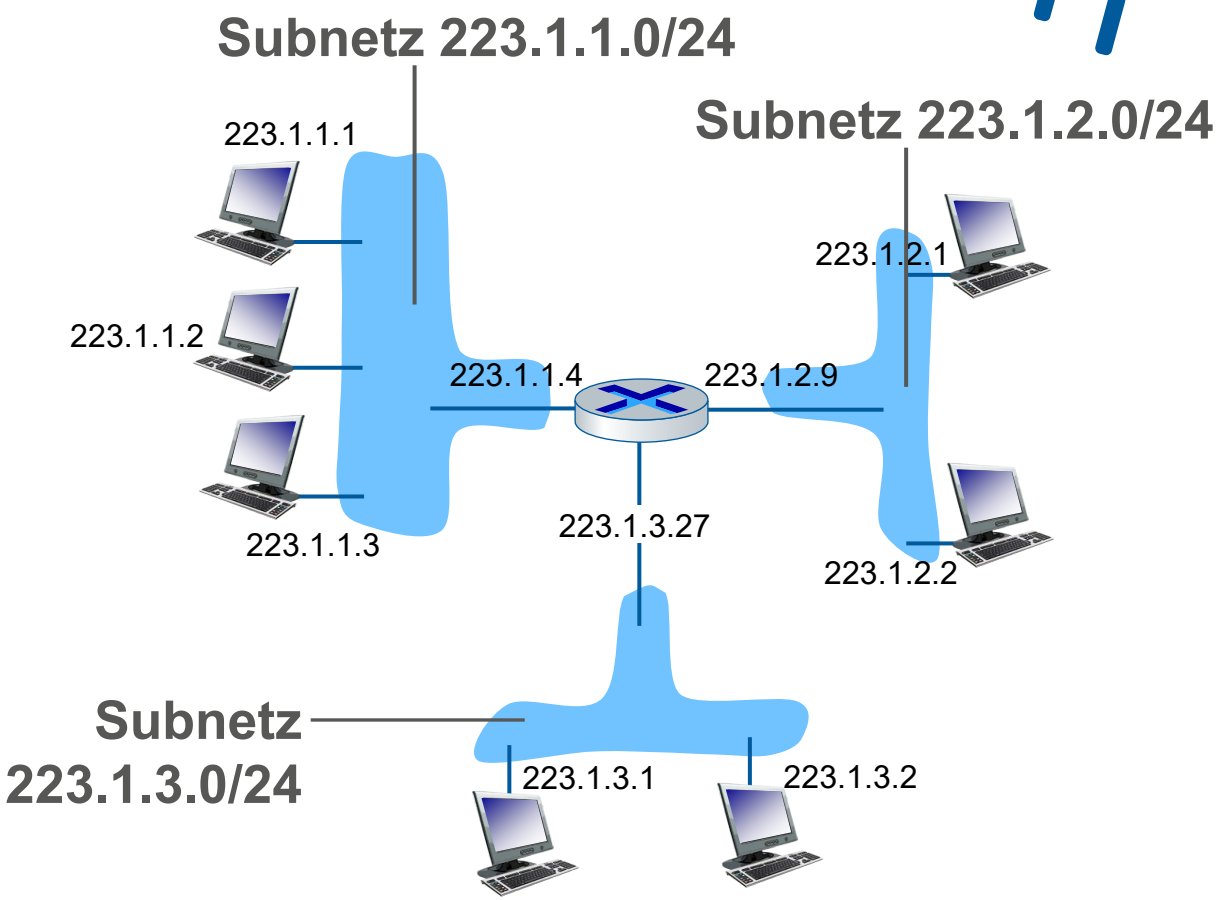
- **IP-Adressen haben Struktur:**
  - **Subnetz Anteil:** Geräte im selben Subnetz haben gemeinsame Bits höherer Ordnung
  - **Host Anteil: verbleibende** Bits niedrigerer Ordnung



Netz bestehend aus 3 Subnetzen

Vorgehen zur Subnetzdefinition

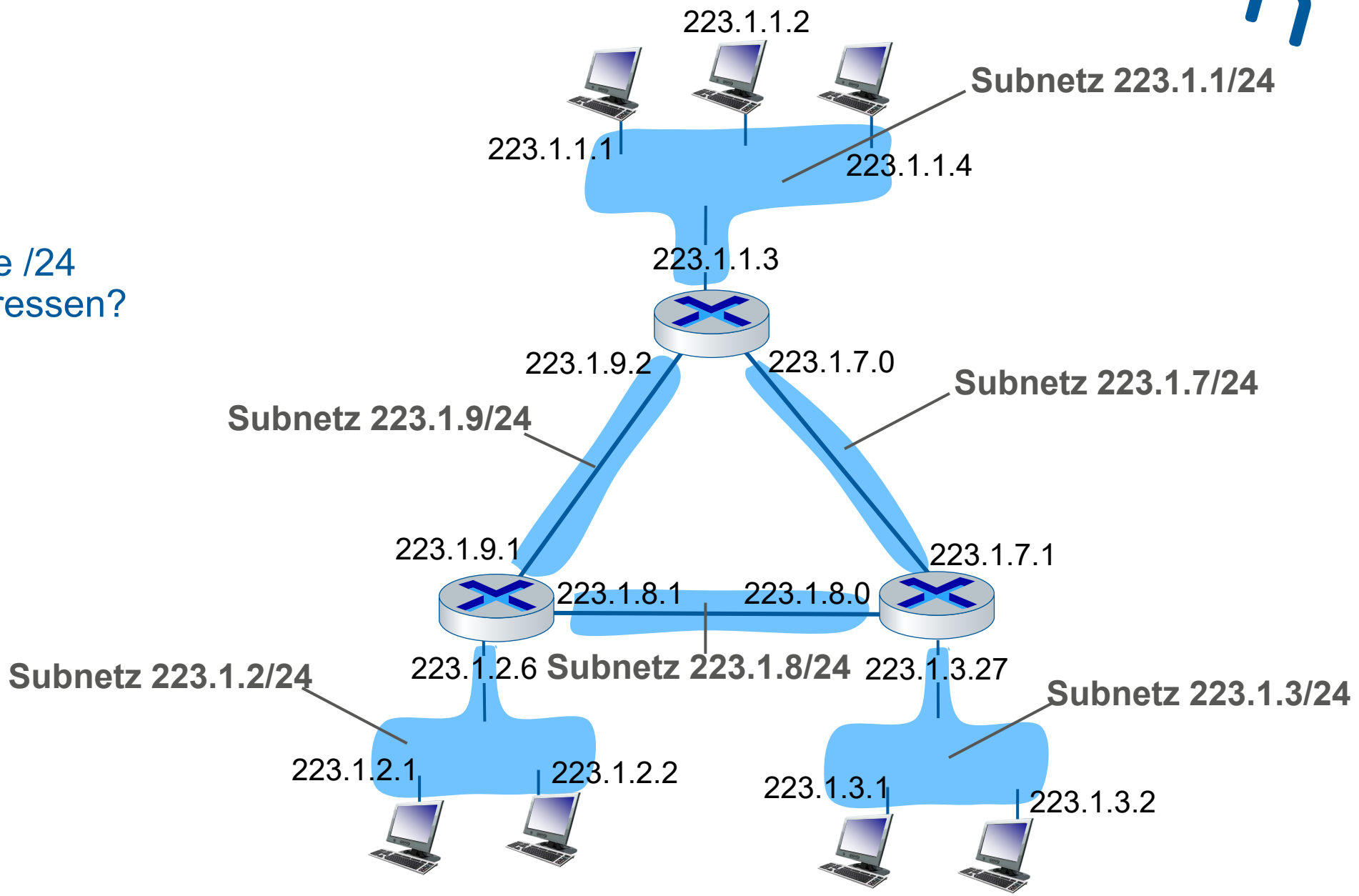
- Schaffen von isolierten “Inseln” durch abkoppeln jedes Interfaces von einem Host oder Router
- Jedes isolierte Netz wird **Subnetz** genannt



Subnetz Maske: /24
(24 Bits höhere Ordnung = Subnetz Anteil der IP-Adresse)

Subnetze

- Wo sind die Subnetze?
- Was sind die /24 Subnetz Adressen?



CIDR: Classless **I**nter**D**omain **R**outing (ausgesprochen “cider”)

- Subnetz Anteil einer Adresse mit “beliebiger” Länge
- Adress Format: **a.b.c.d/x**, x ist die Zahl Bits im Subnetz Anteil der Adresse

200.23.16.0/23

IP Addressen: Wie bekommt man eine?

Tatsächlich sind das **zwei** Fragen:

1. **Frage:** Wie bekommt ein **Host** eine IP-Adresse in seinem Netz (Host Anteil der Adresse)?
2. **Frage:** Wie bekommt ein **Netz** eine eigene IP-Adresse ((Sub)Netz Anteil der Adresse)?

Wie bekommt ein **Host** eine IP Adresse?

- Konfiguration durch Systemadmin (z.B., in /etc/network/interfaces in Linux)
- **DHCP: D**ynamic **H**ost **C**onfiguration **P**rotocol: dynamisches Anfragen einer Adresse von einem Server
  - “Plug-and-Play”

DHCP: Dynamic Host Configuration Protocol

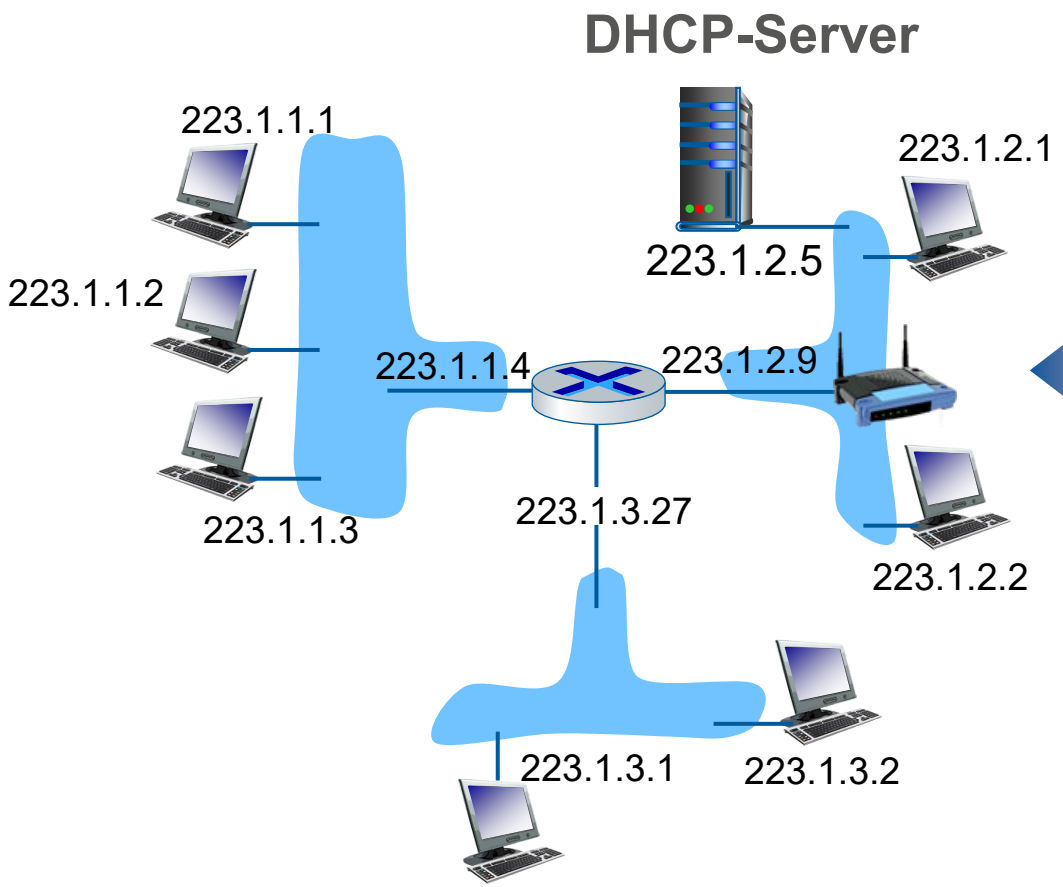
Ziel: Der Host erhält eine IP-Adresse **dynamisch** wenn er ein Netz “betritt”

- kann seinen “Lease” auf eine benutzte Adresse verlängern
- erlaubt die Mehrfachnutzung von Adressen (Adresse nur genutzt, wenn angeschaltet)
- Unterstützung für mobile Nutzer, die das Netz betreten/verlassen

DHCP Übersicht:

- Host broadcastet eine **DHCP Discover Nachricht** [optional]
- DHCP-Server antwortet mit einer **DHCP Offer Nachricht** [optional]
- Host fordert eine IP-Adresse an: **DHCP Request Nachricht**
- DHCP-Server sendet eine Adresse: **DHCP Ack Nachricht**

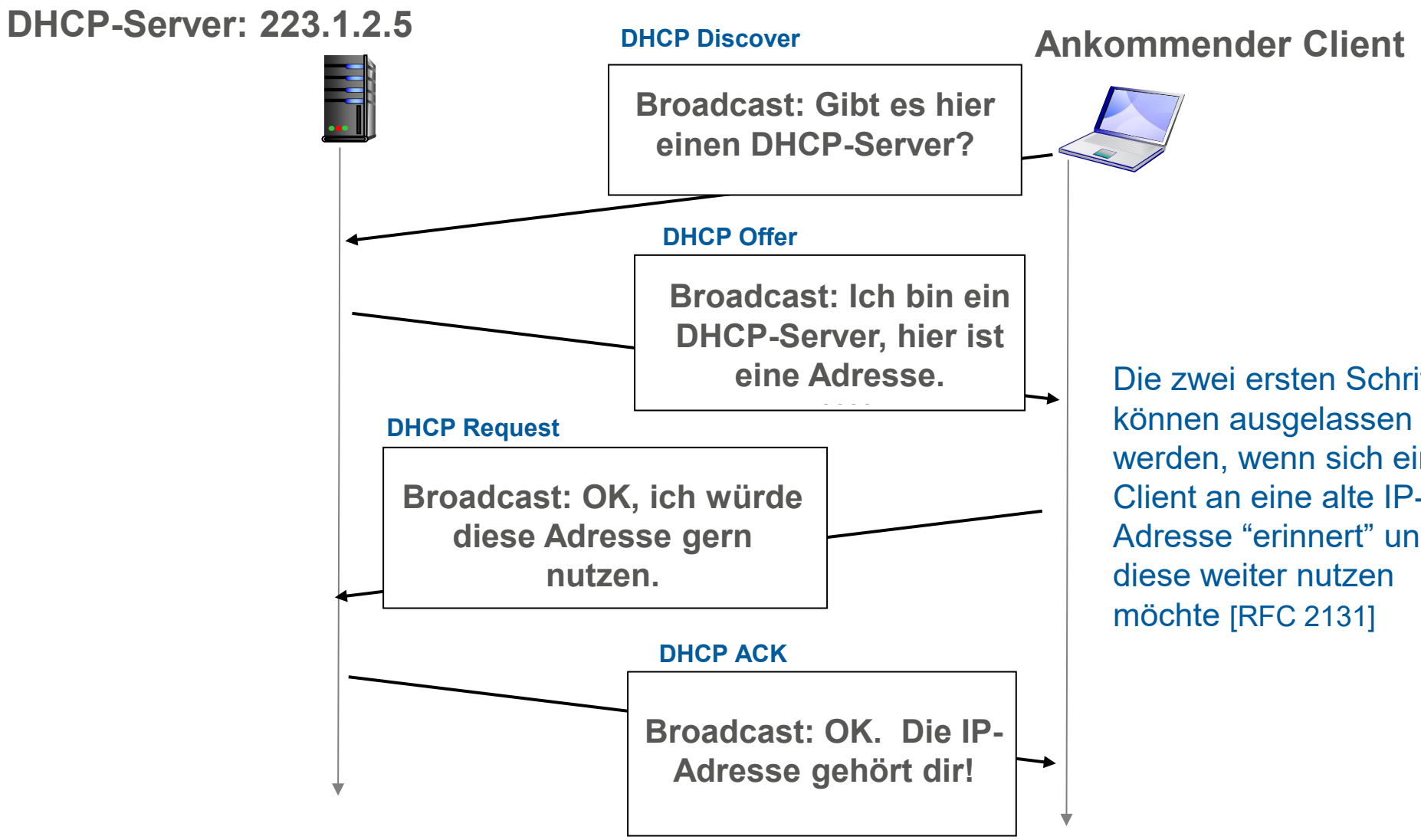
DHCP-Client+Server Szenario



Oft wird DHCP-Server auf dem Router ausgeführt, der alle Subnetze bedient, die am Router verbunden sind

ankommender **DHCP-Client** benötigt eine Adresse im Netz

DHCP-Client+Server Szenario



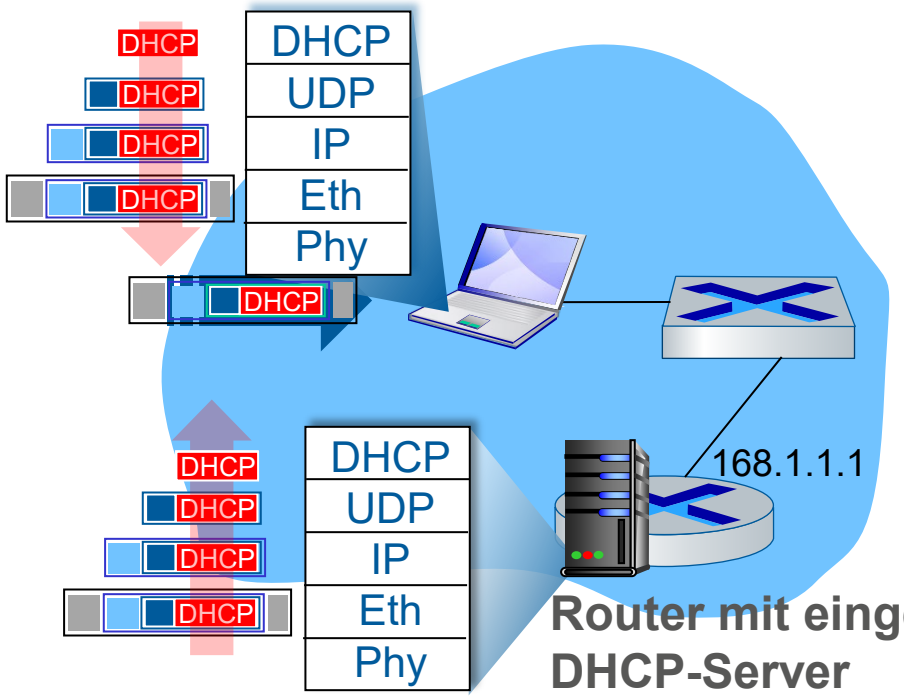
Die zwei ersten Schritte können ausgelassen werden, wenn sich ein Client an eine alte IP-Adresse “erinnert” und diese weiter nutzen möchte [RFC 2131]

DHCP: Mehr als nur IP Adressen

DHCP kann mehr liefern als nur IP-Adressen für ein Subnetz:

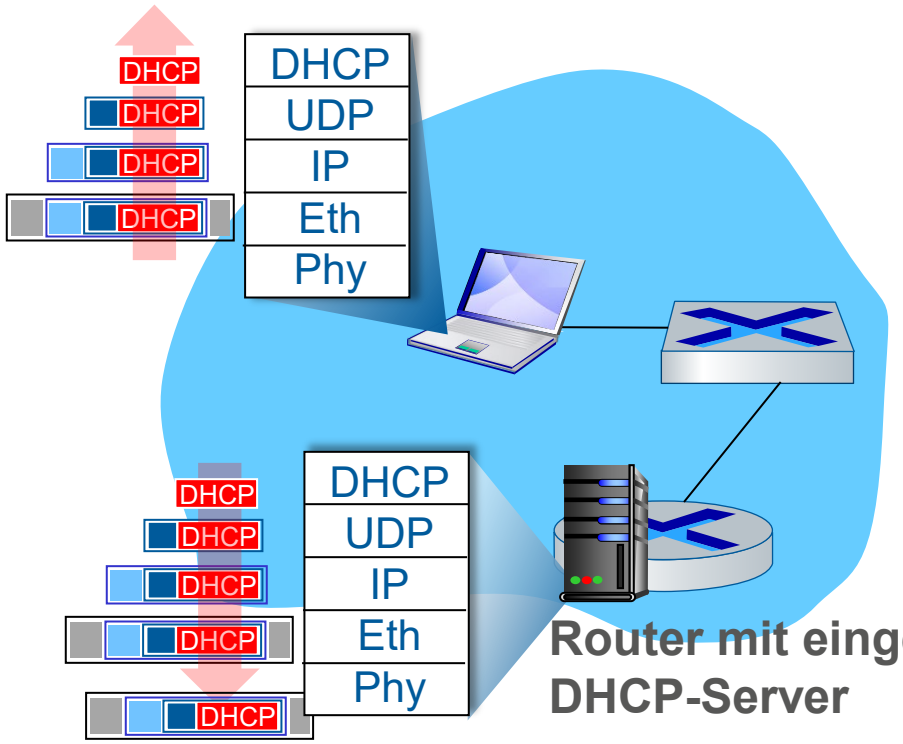
- Adresse des ersten Routers aus Sicht des Clients (Gateway)
- Name und IP-Adresse eines DNS-Servers
- Netzmaske (Netzanteil einer Adresse)

DHCP: Beispiel



- Der ankommende Laptop wird DHCP benutzen um eine IP-Adresse, die Gateway Adresse und die DNS Server Adresse zu erhalten.
- DHCP REQUEST Nachricht über UDP, IP, und schließlich Ethernet
- Ethernet Rahmen Broadcast (Ziel: `FFFFFFFFFFFF`) im LAN wird am Router mit DHCP-Server empfangen
- Ethernet wird ausgepackt zu IP, wird ausgepackt zu, UDP wird ausgepackt zu DHCP

DHCP: Beispiel



- DHCP-Server erstellt DHCP ACK, dass die Client IP Adresse, Gateway IP Adresse, Namen & IP-Adresse eines DNS-Servers beinhaltet
- “Verpackte” DHCP-Server Antwort wird zum Client weitergeleitet, der sie bis zu DHCP auspackt
- Der Client kennt nun seine IP-Adresse, Gateway IP Adresse, Namen & IP Adresse eines DNS-Servers

Frage: Wie bekommt das *Netz* den Subnetz Anteil einer Netzmaske?

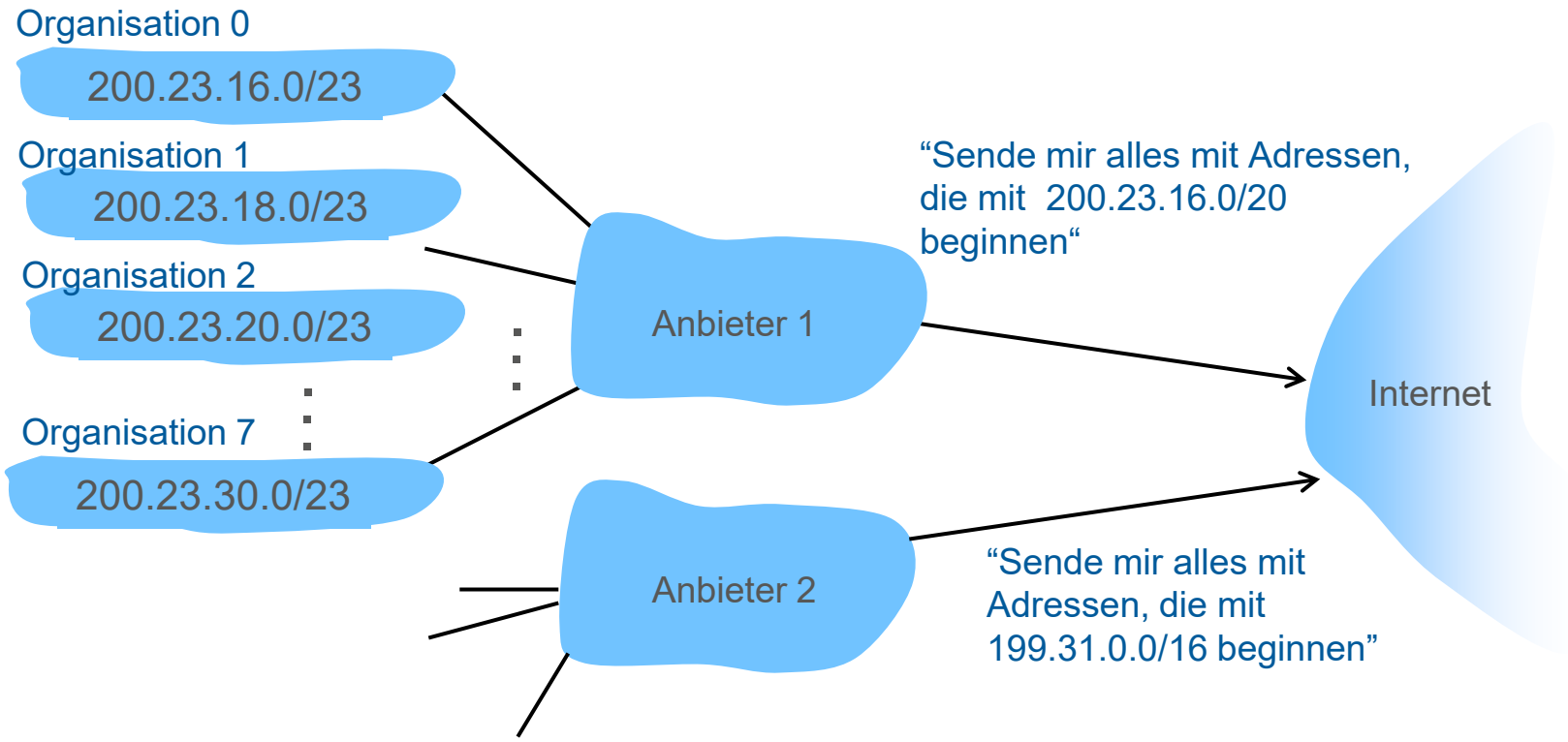
Antwort: es bekommt einen Teil des IP-Adressraumes seine ISPs zugeteilt

ISP-Block	**11001000 00010111 00010000**	**00000000**	**200.23.16.0/20**

Der ISP kann dann daraus z.B. 8 Blöcke bilden:

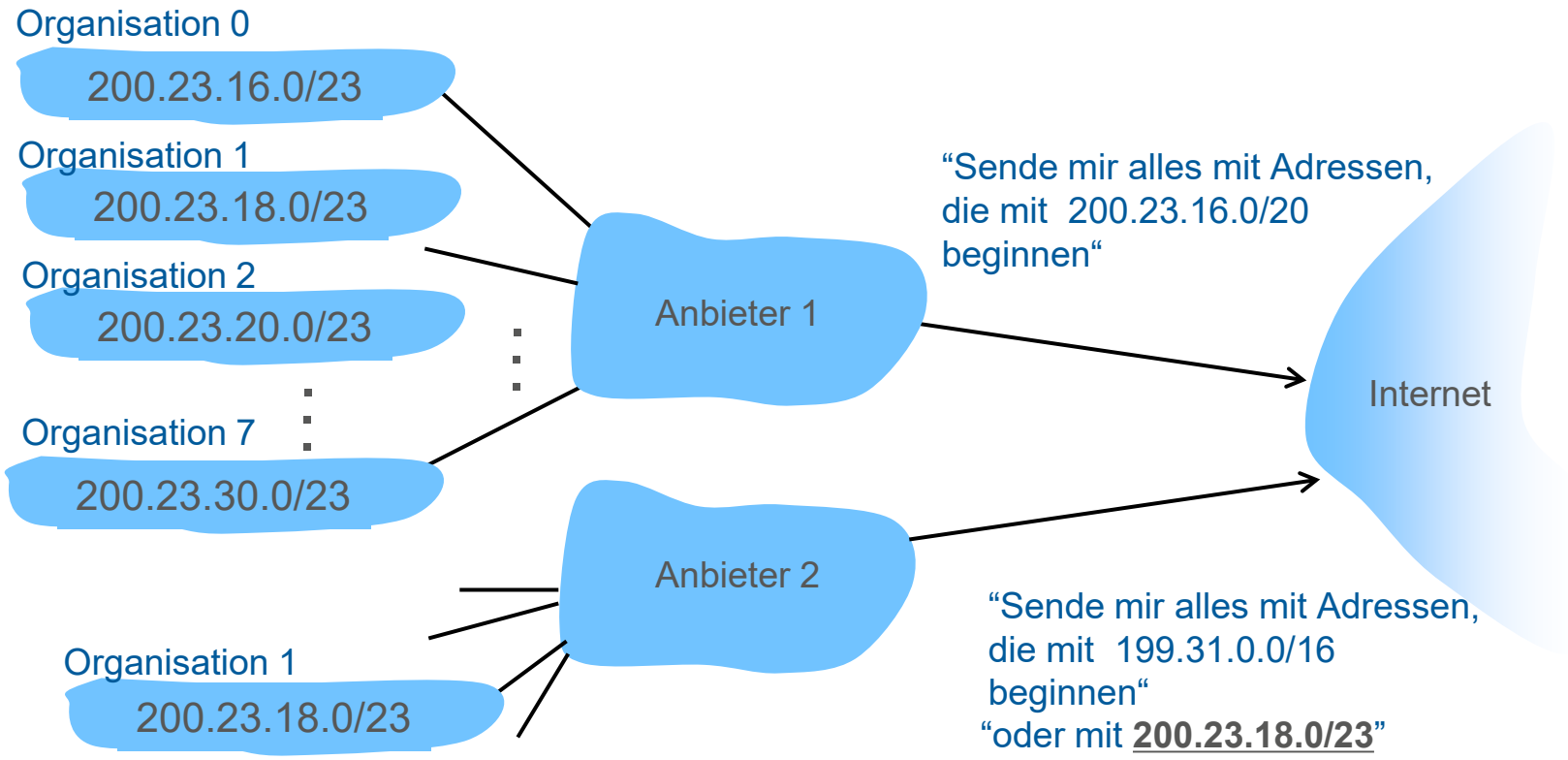
Organisation 0	11001000 00010111 00010000	00000000	200.23.16.0/23
Organisation 1	11001000 00010111 00010010	00000000	200.23.18.0/23
Organisation 2	11001000 00010111 00010100	00000000	200.23.20.0/23
...	.....	....	....
Organisation 7	11001000 00010111 00011110	00000000	200.23.30.0/23

Hierarchische Adressierung erlaubt das effiziente Verteilen von Routing Informationen:



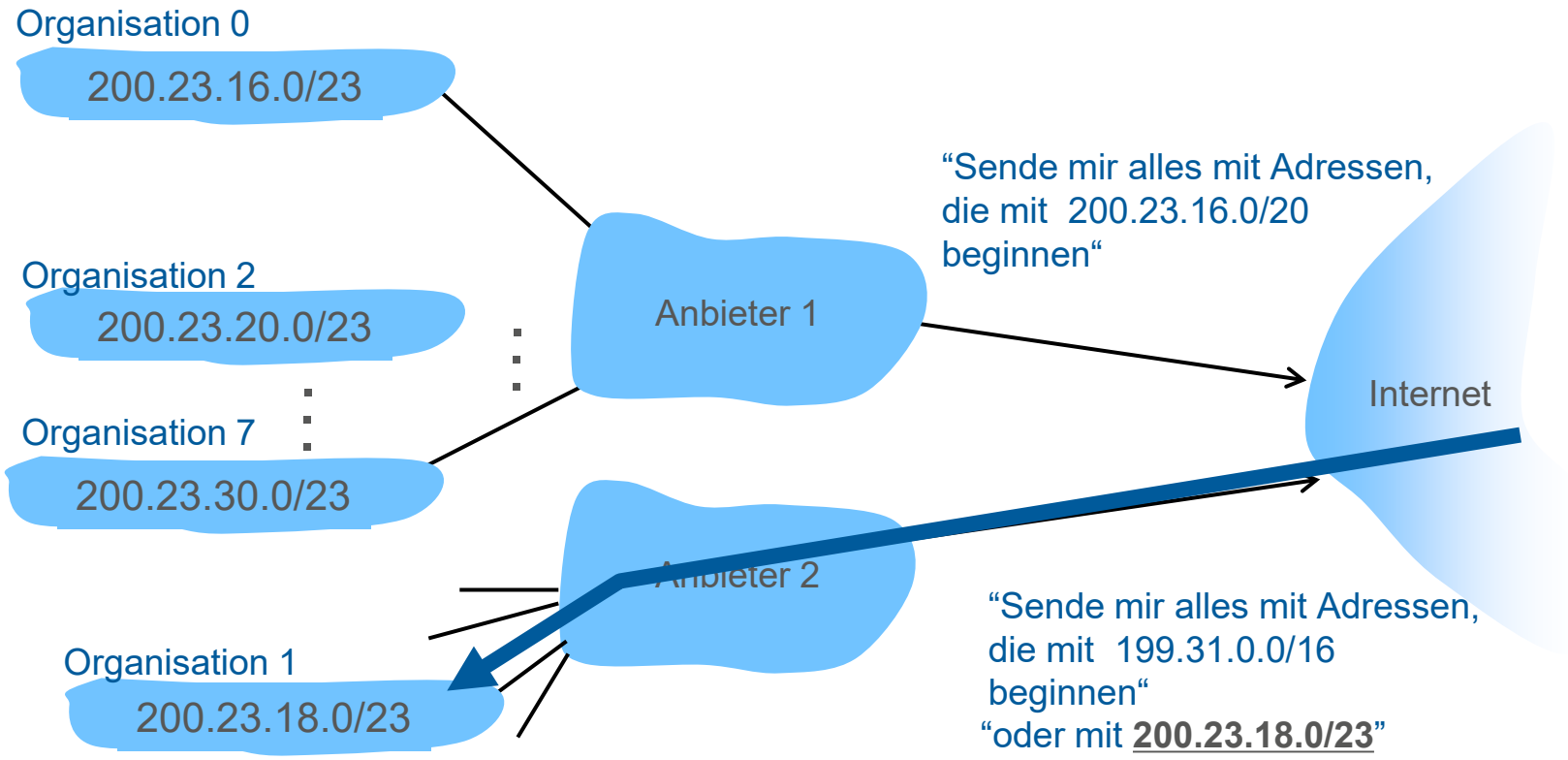
Hierarchische Adressierung: spezifischere Routen

- Organisation 1 wechselt von Anbieter 1 zu Anbieter 2
- Anbieter 2 verbreitet nun eine spezifischere Route zu Organisation 1



Hierarchische Adressierung: spezifischere Routen

- Organisation 1 wechselt von Anbieter 1 zu Anbieter 2
- Anbieter 2 verbreitet nun eine spezifischere Route zu Organisation 1



IP-Adressierung: abschließende Worte ...

Frage: Wie bekommt ein ISP einen Adressblock?

Antwort: ICANN: Internet Corporation for Assigned Names and Numbers http://www.icann.org/

- vergibt Adressen über 5 **regionale Registries (RRs)** (die ebenfall an lokale Registries weitervergeben können)
- verwaltet die DNS Root Zone, delegiert das Management einzelner TLDs (.com, .edu , …)

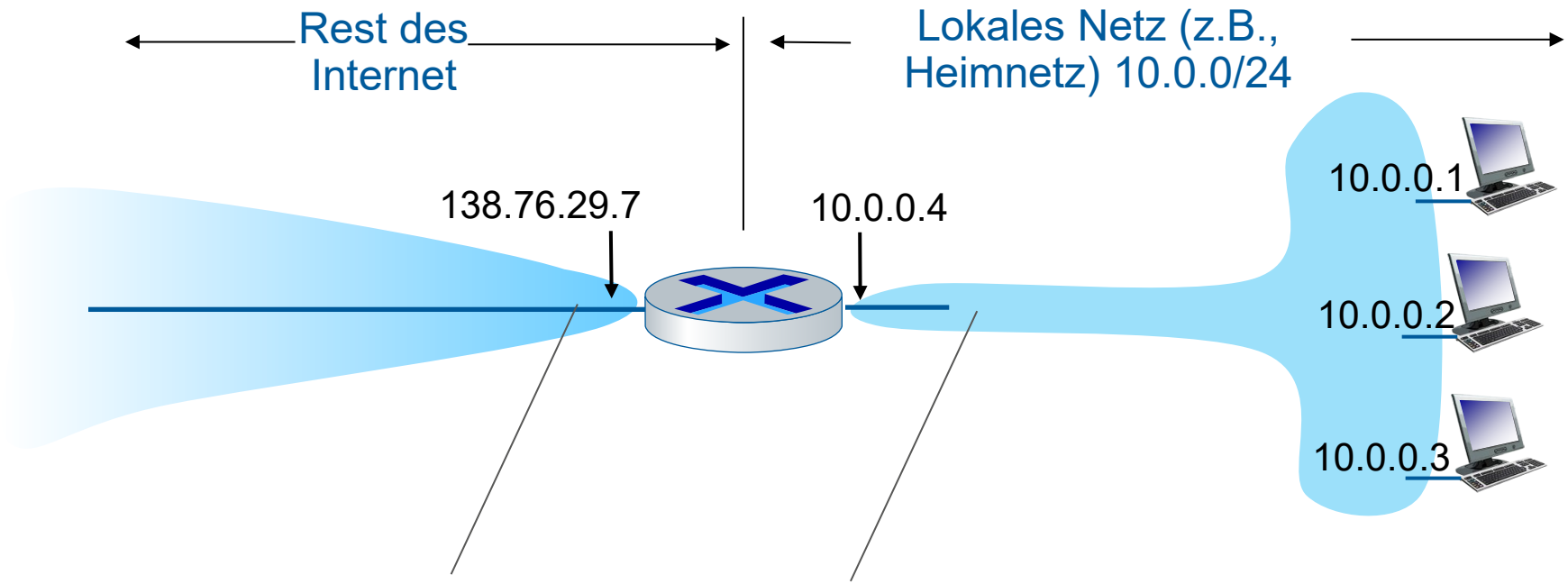
Frage: Gibt es genug 32-bit IP-Adressen?

- ICANN vergab 2011 den letzten IPv4 Adressbereich an die RRs
- NAT hilft beim Sparen von IPv4 Adressen
- IPv6 hat einen 128-bit Adressraum

"Who the hell knew how much address space we needed?"
Vint Cerf (gefragt nach der Entscheidung IPv4 Adressen 32 Bits lang zu machen)

- **Vermittlungsschicht: Überblick**
  - Datenpfad
  - Kontrollebene
- **Was steckt in einem Router**
  - Eingangsports, Switching, Ausgangsports
  - Puffer Management, Scheduling
- **IP: das Internet Protokoll**
  - Datagramm Format
  - Adressierung
  - **Network Address Translation**
  - **IPv6**
- **Generalized Forwarding, SDN**
  - Match+Action
  - OpenFlow: Match+Action
- **Mittelboxen**
  - Funktionen von Mittelboxen
  - Evolution & Prinzipien der Internet Architektur

NAT: Alle Geräte im lokalen Netz teilen sich nur **eine** IPv4 Adresse im Hinblick auf die Welt außen



Alle Datagramme, die das lokale Netz **verlassen** haben **dieselbe** Quell-NAT IP Adresse: 138.76.29.7, aber **verschiedene** Quell-Port-Nummern

Datagramme mit Quelle oder Ziel in diesem Netz haben die 10.0.0.0/24 Adresse als Quelle oder Ziel (wie sonst)

NAT: Network Address Translation

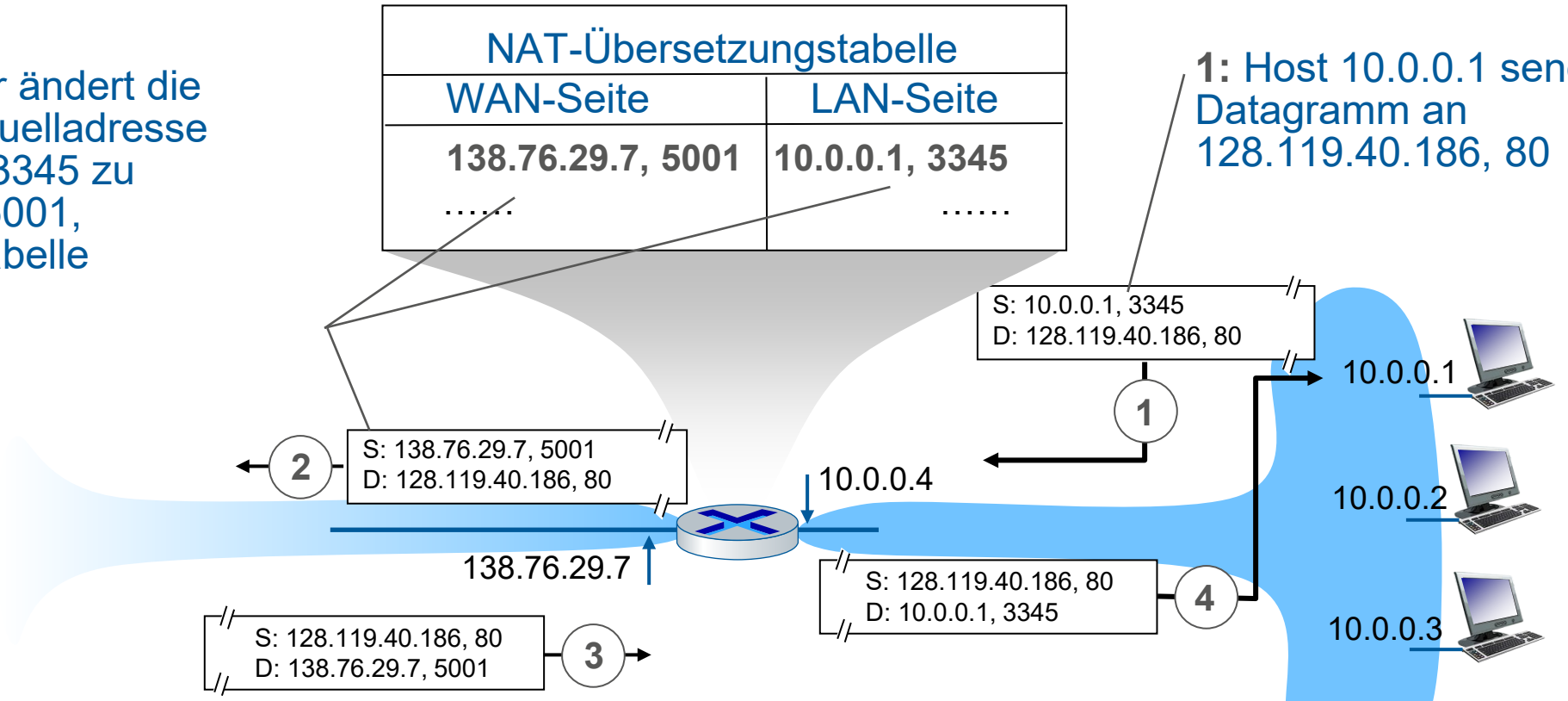
- Alle Geräte im lokalen Netz haben 32-Bit Adressen aus einem “privaten” IP Adress-Raum (10/8, 172.16/12, 192.168/16 Präfix), die nur in lokalen Netzen verwendet werden dürfen
- Vorteile:
  - nur **eine** IP-Adresse wird vom Anbieter ISP für **alle** Geräte benötigt
  - Adressen von Hosts im lokalen Netz können verändert werden, ohne das der äußeren Welt mitteilen zu müssen
  - ISP kann gewechselt werden, ohne die Adressen von Geräten im lokalen Netz verändern zu müssen
  - Sicherheit: Geräte innerhalb des lokalen Netzes sind nicht direkt adressierbar, sichtbar von der Außenwelt

Implementierung: NAT-Router muss (transparent):

- bei allen **ausgehenden Datagrammen** die Quell-IP Adresse & Port Nr. mit der NAT-IP Adresse & einer neuen Port Nr. ersetzen
  - entfernte Clients/Server werden mit der NAT-IP Adresse & der neuen Port Nr. als Zieladresse antworten
- sich alle (Quell-IP Adresse, Port Nr. & NAT-IP Adresse, Port Nr.) Paare **(in der NAT-Übersetzungstabelle) merken**
- bei **ankommenden Datagrammen** die NAT-IP Adresse & Port Nr. in den Zielfeldern mit der entsprechenden Quell-IP Adresse & Port Nr. ersetzen

NAT: Network Address Translation

2: NAT-Router ändert die Datagramm Quelladresse von 10.0.0.1, 3345 zu 138.76.29.7, 5001, updatet die Tabelle



NAT: Network Address Translation

- NAT ist umstritten:
  - Router “sollten” nur bis Layer 3 arbeiten
  - “Adressmangel” sollte durch IPv6 gelöst werden
  - bricht das Ende-zu-Ende Prinzip (Manipulation der Port Nr. durch Gerät der Vermittlungssicht)
  - NAT-Traversal: was, wenn sich ein Client zu einem Server hinter NAT verbinden möchte?
- aber NAT wird (vorerst) bleiben:
  - wird extensiv in Heim- und Firmennetze, sowie 4G/5G Mobilfunknetzen eingesetzt

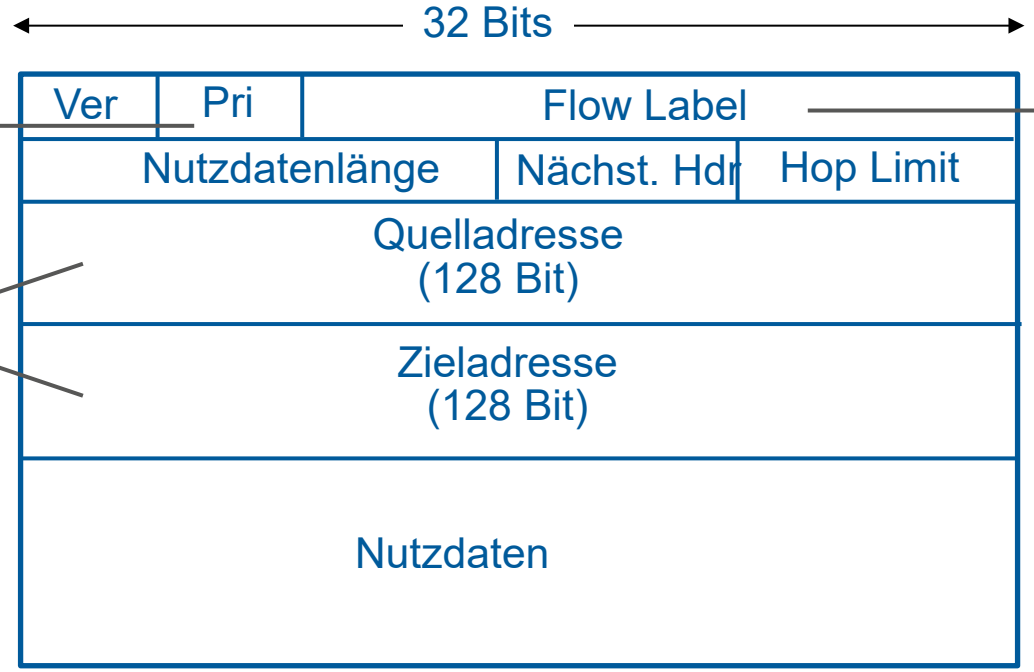
IPv6: Motivation

- **Ursprüngliche Motivation:** 32-Bit IPv4 Adressraum vollständig allokiert
- weitere Gründe:
  - Verarbeitungs-/Weiterleitungsgeschwindigkeit: feste 40-Byte Headerlänge
  - Ermöglichen von Andersbehandlung verschiedener “Flows” auf der Vermittlungsschicht

IPv6 Datagramm Format

Priorität: kennzeichnen von Prioritäten zwischen Datagrammen in einem Flow

128-Bit IPv6 Adressen



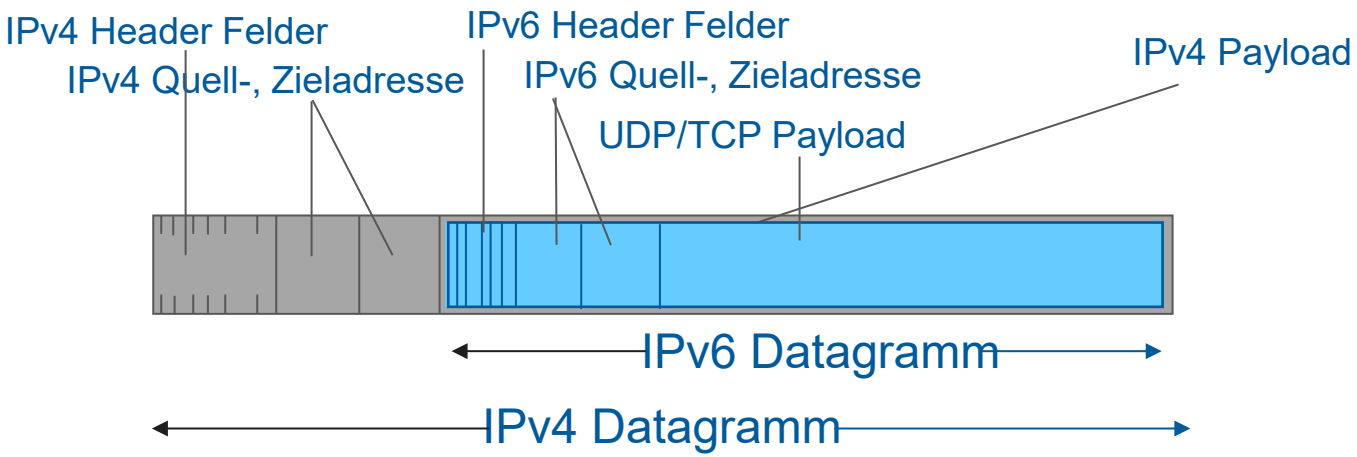
Flow Label: identifizieren von Datagrammen im selben “Flow.” (“Flow”-Konzept nur vage definiert).

Was fehlt (im Vergleich mit IPv4):

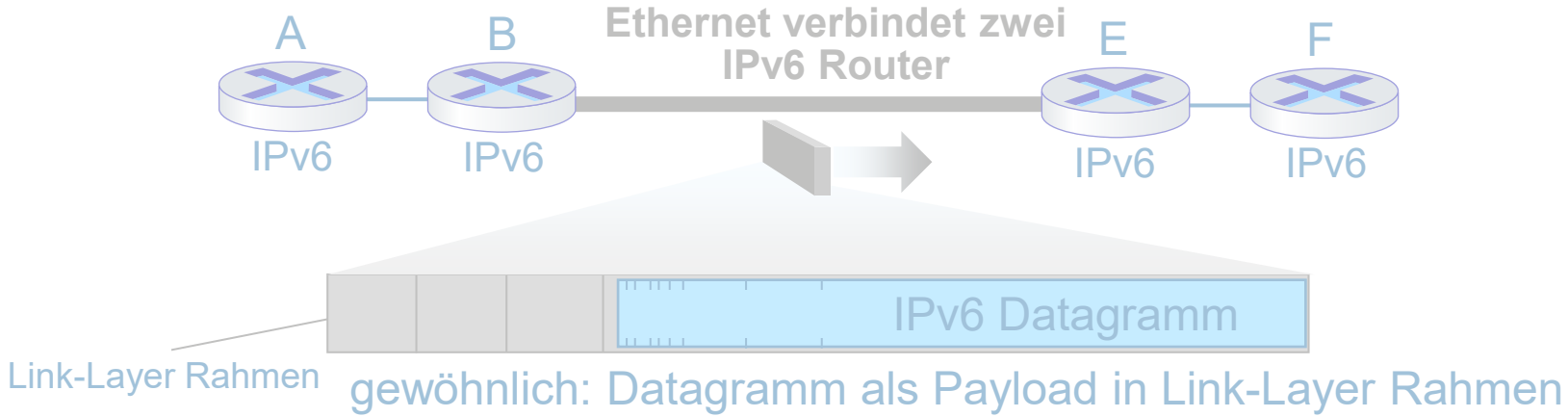
- keine Checksumme (Erhöhung der Verarbeitungsgeschwindigkeit)
- keine Fragmentierung/Neuzusammensetzung
- keine Optionen (verfügbar als höhere, Next-Header Protokolle am Router)

- nicht alle Router können gleichzeitig nachgerüstet werden
  - keine “Feiertage”
  - Wie kann ein Netz mit gemischten IPv4 und IPv6 Routern funktionieren?

- **Tunnel:** IPv6 Datagramm wird als *Payload* in IPv4 Datagramm zwischen IPv4 Routern transportiert (“Paket im Paket”)
  - Tunnel werden extensiv in anderen Zusammenhängen genutzt (Datenzentren, 4G/5G Kernnetze)

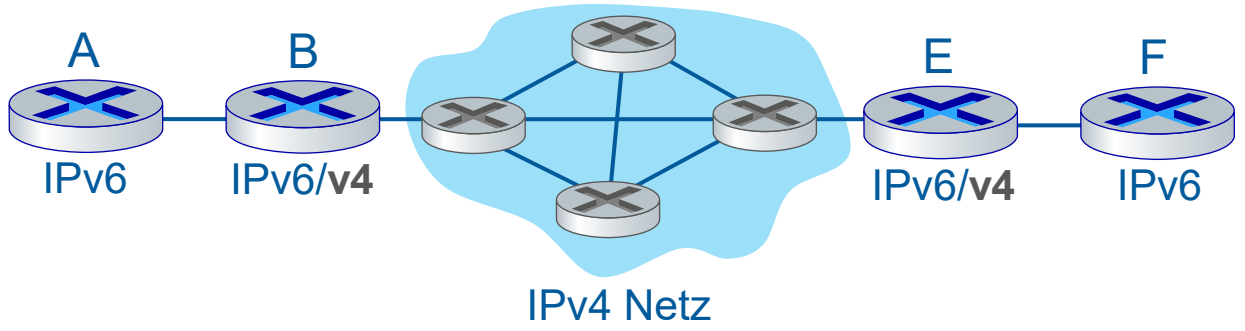


Ethernet verbindet zwei IPv6 Router:

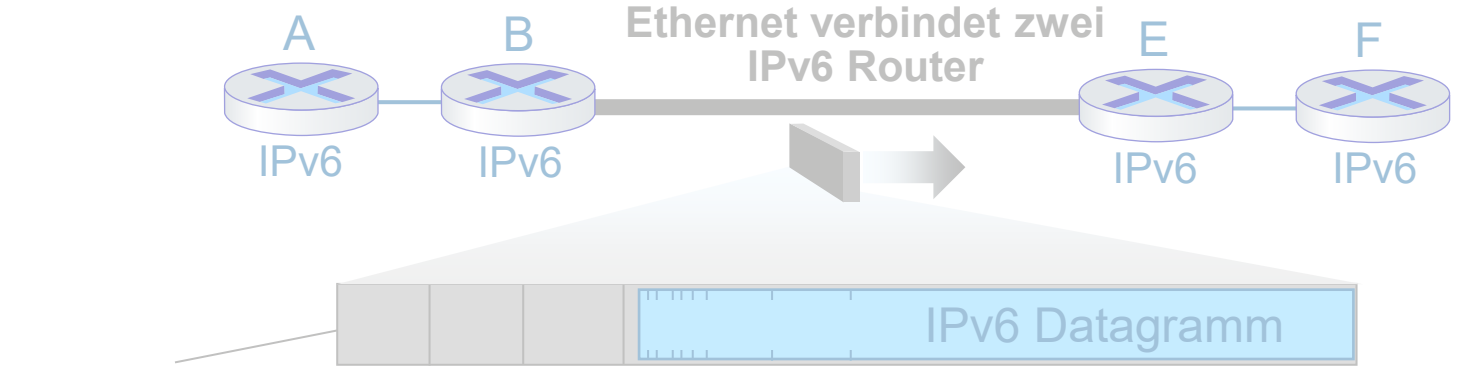


gewöhnlich: Datagramm als Payload in Link-Layer Rahmen

IPv4 Netz verbindet zwei IPv6 Router

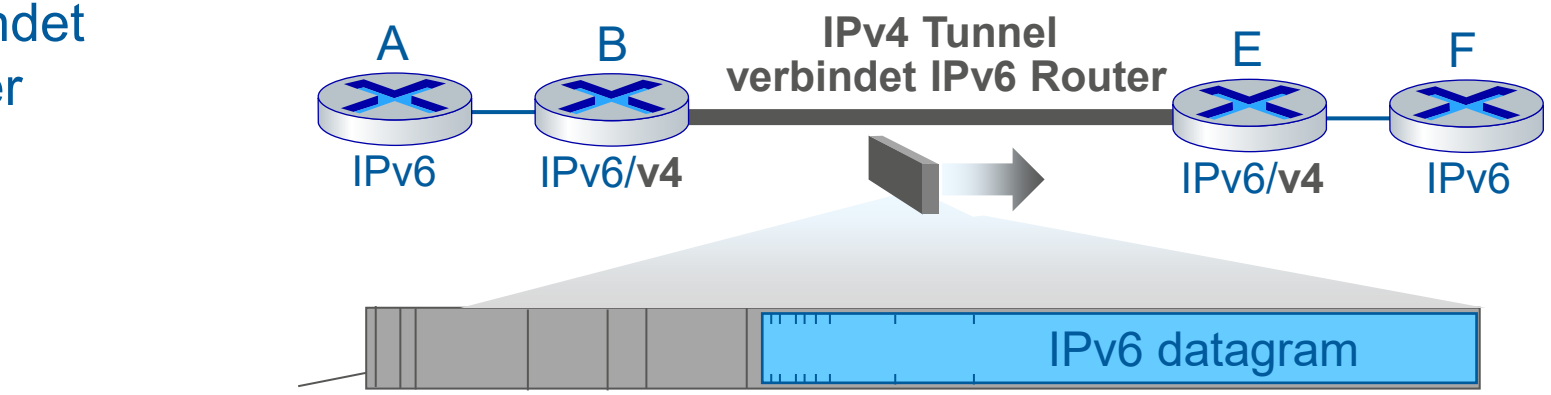


Ethernet verbindet zwei IPv6 Router:

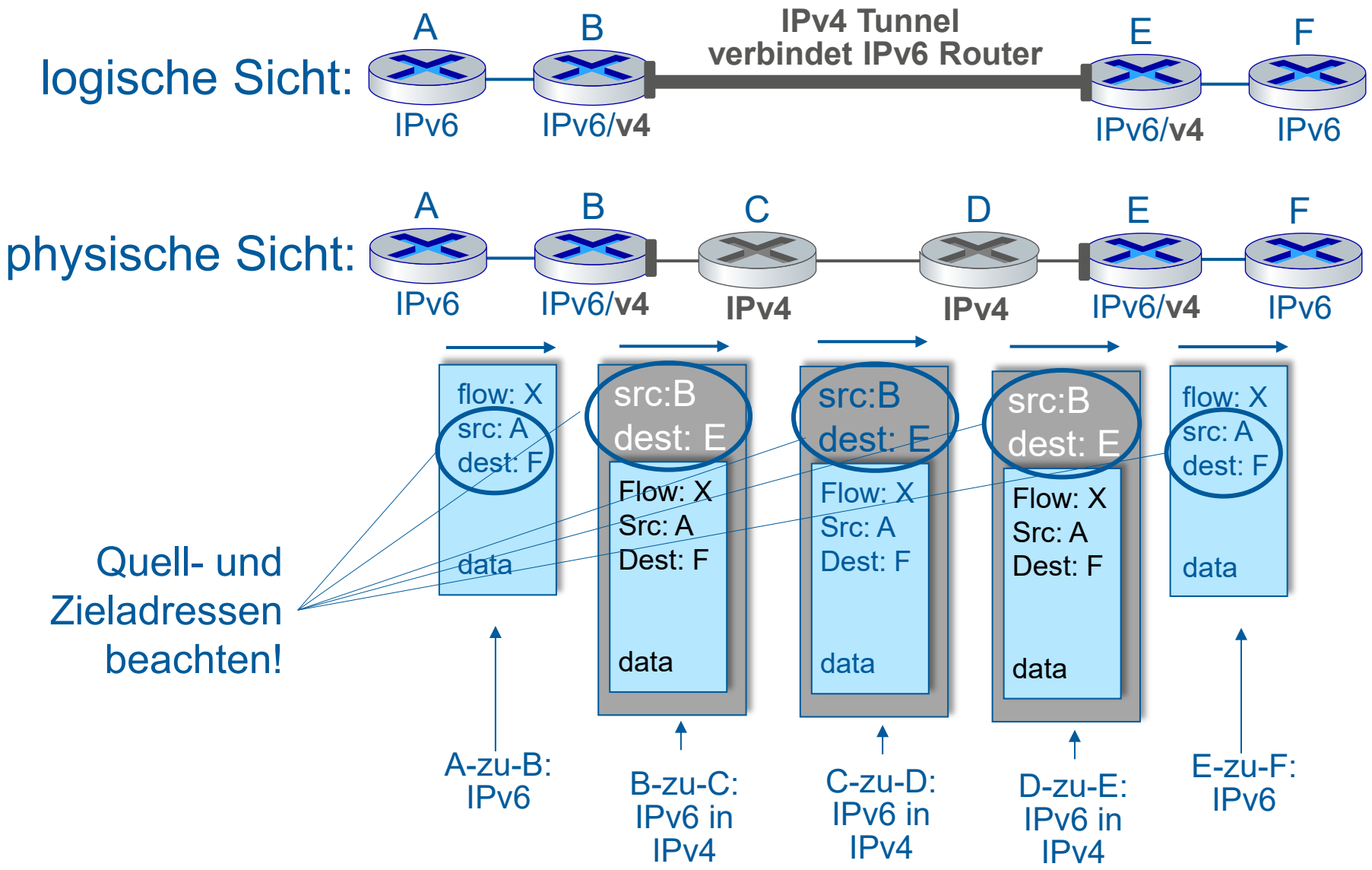


gewöhnlich: Datagramm als Payload in Link-Layer Rahmen

IPv4 Netz verbindet zwei IPv6 Router

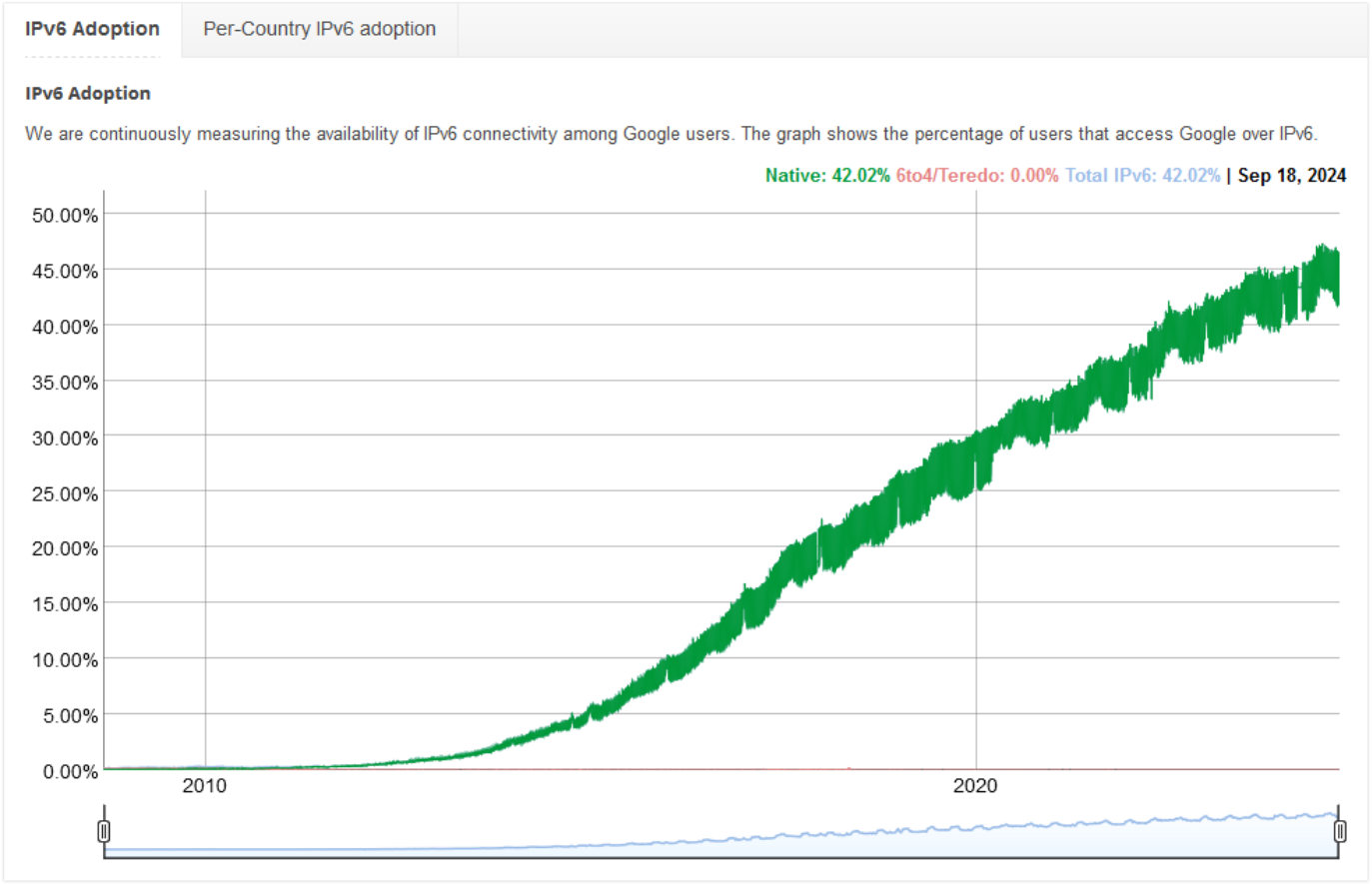


Tunnel: IPv6 Datagramm als Payload in einem IPv4 Datagramm


logische Sicht:
A
B
IPv4 Tunnel
verbindet IPv6 Router
E
F
IPv6
IPv6/v4
IPv6/v4
IPv6
physische Sicht:
A
B
C
D
E
F
IPv6
IPv6/v4
IPv4
IPv4
IPv6/v4
IPv6
Quell- und Zieladressen beachten!
flow: X
src: A
dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
src:B
dest: E
Flow: X
Src: A
Dest: F
data
flow: X
src: A
dest: F
data
A-zu-B: IPv6
B-zu-C: IPv6 in IPv4
C-zu-D: IPv6 in IPv4
D-zu-E: IPv6 in IPv4
E-zu-F: IPv6

IPv6: Verbreitung

- Google: ~ 42% aller Clients greifen auf Dienste mit IPv6 zu



https://www.google.com/intl/en/ipv6/statistics.html

IPv6: Verbreitung

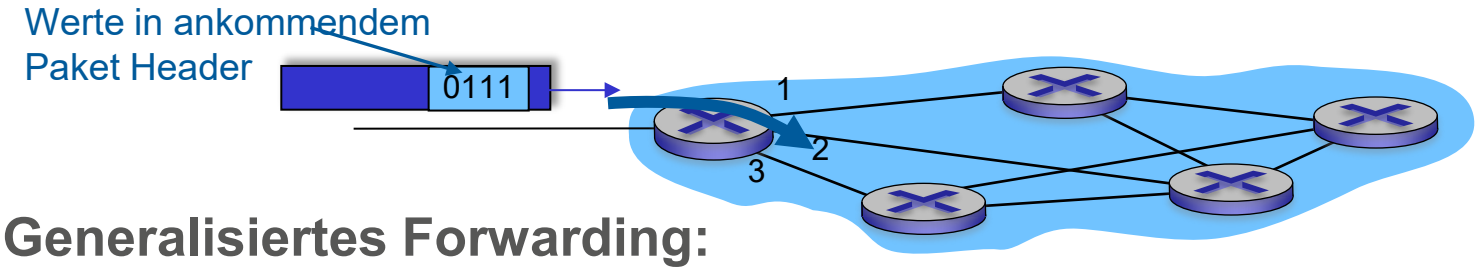
- Google: ~ 42% aller Clients greifen auf Dienste mit IPv6 zu
- Lange, lange Zeit für Ausrollen und Nutzung:
  - 29 Jahre bisher
  - Änderungen auf Applikationsschicht in den letzten 29 Jahren: Web, Soziale Medien, Streaming, Online Gaming, Telepräsenz, …
  - **Warum?**

- **Vermittlungsschicht: Überblick**
  - Datenpfad
  - Kontrollebene
- **Was steckt in einem Router**
  - Eingangsports, Switching, Ausgangsports
  - Puffer Management, Scheduling
- **IP: das Internet Protokoll**
  - Datagramm Format
  - Adressierung
  - Network Address Translation
  - IPv6
- **Generalized Forwarding, SDN**
  - Match+Action
  - OpenFlow: Match+Action
- **Mittelboxen**
  - Funktionen von Mittelboxen
  - Evolution & Prinzipien der Internet Architektur

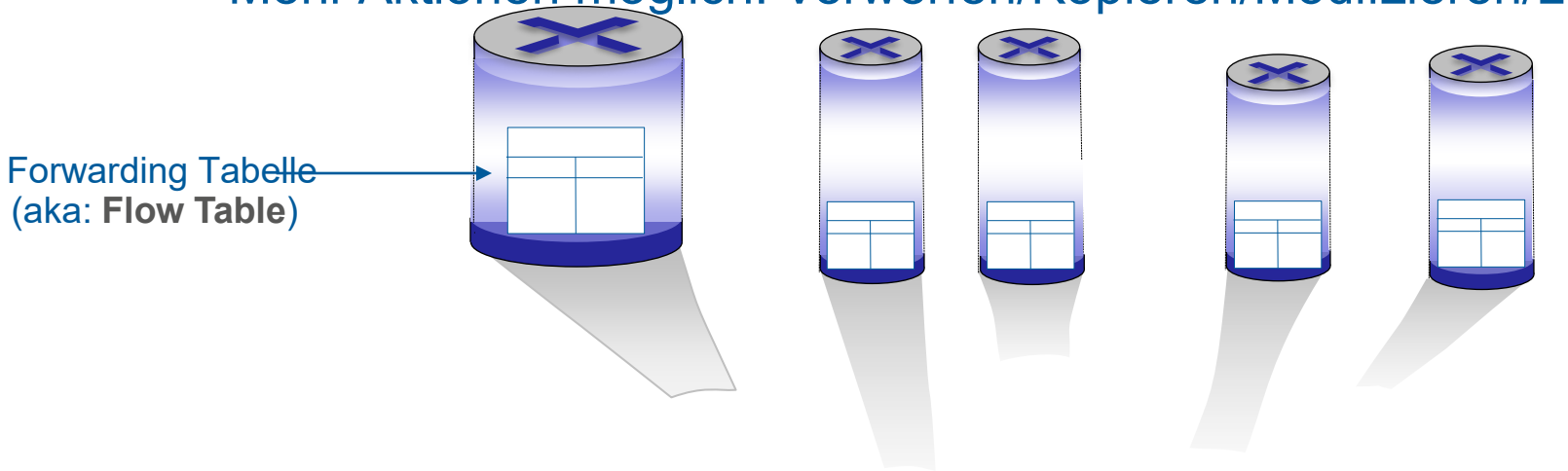
Generalisiertes Forwarding: Match + Action

Wiederholung: jeder Router hat eine **Forwarding Tabelle** (aka: **Flow Table**)

- **"Match + Action"** Abstraktion: Matchen von Bits in ankommendem Paket, Aktion durchführen
  - **Ziel-basiertes Forwarding:** Weiterleiten basierend auf Ziel-IP Adresse

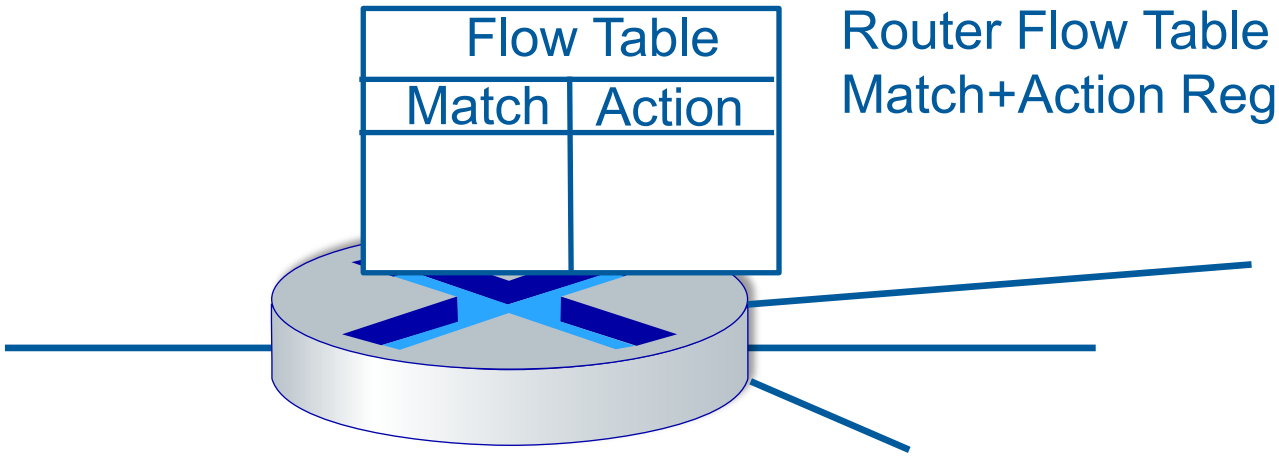


  - **Generalisiertes Forwarding:**
    - Viele Header können die Aktion bestimmen
    - Mehr Aktionen möglich: Verwerfen/Kopieren/Modifzieren/Loggen eines Pakets



Flow Table Abstraktion

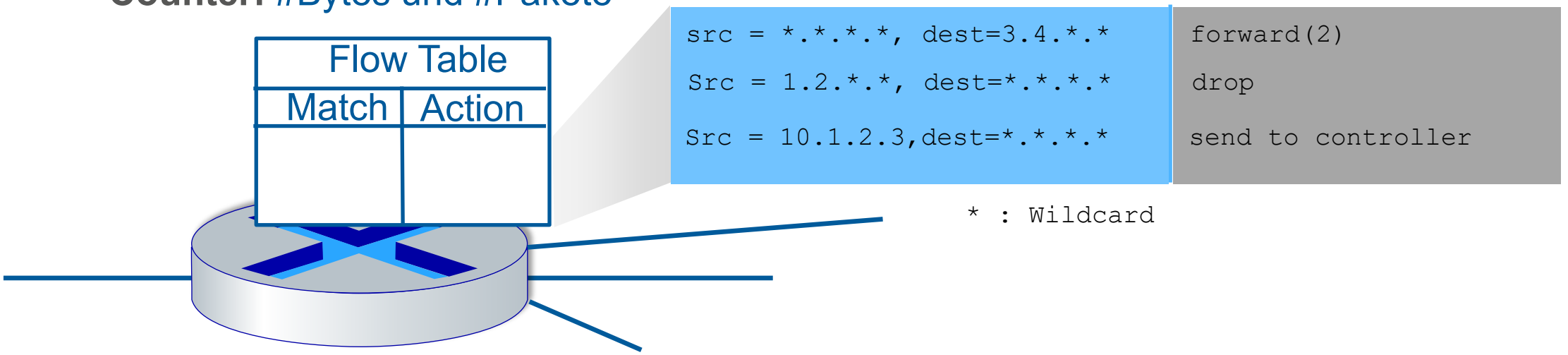
- **Flow:** definiert durch Header-Feld Werte (in Link-, Vermittlungs-, Transportschicht Feldern)
- **Generalisiertes Forwarding**: einfache Paket-Verarbeitungsregeln
  - **Match:** bestimmte Werte in Paket-Headerfeldern
  - **Aktionen:** für entsprechendes Paket: Verwerfen, Weiterleiten, Modifizieren des Pakets oder Senden an den Controller
  - **Priorität:** Zur Unterscheidung überlappender Matches
  - **Counter:** #Bytes und #Pakete



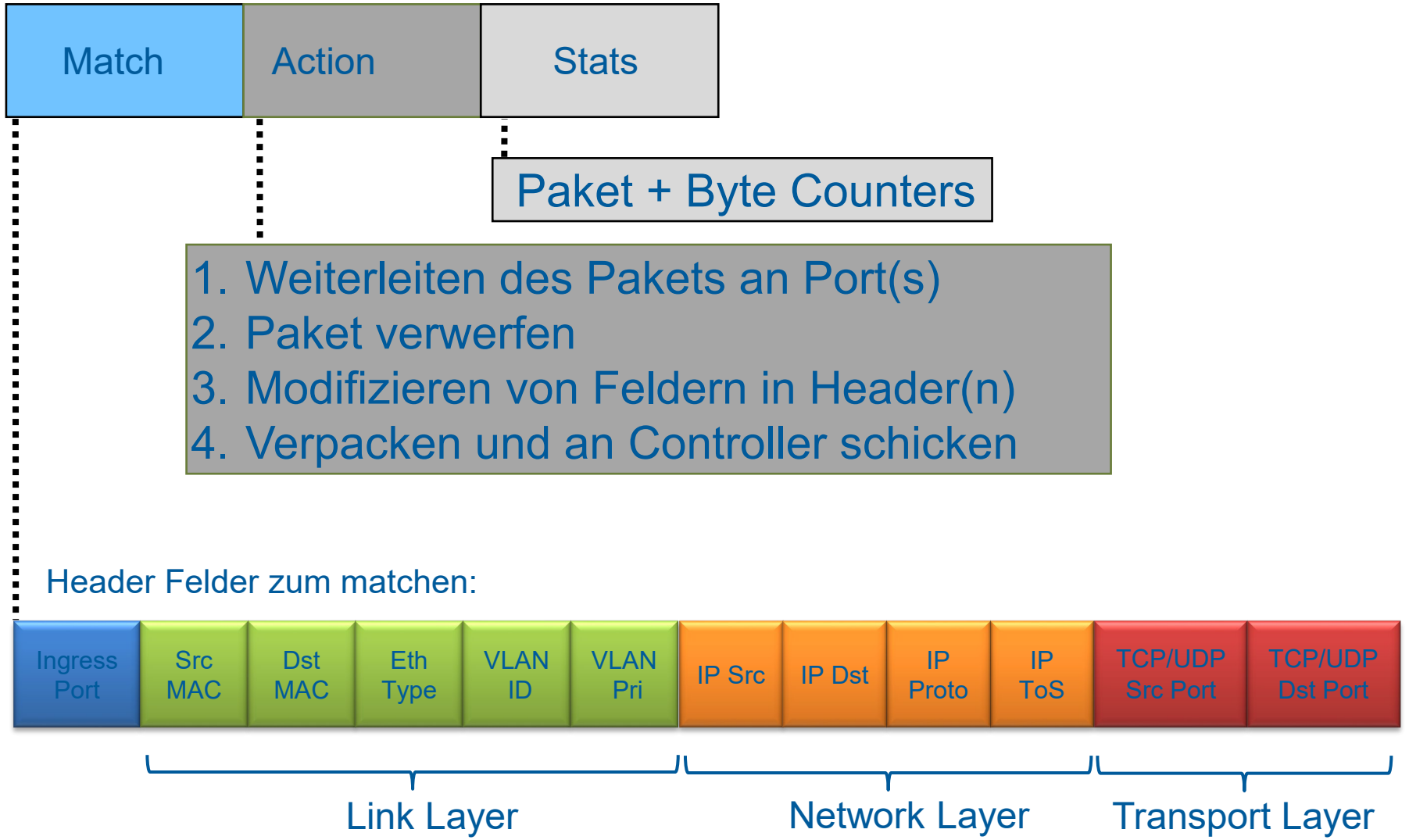
Router Flow Table definiert Match+Action Regeln

Flow Table Abstraktion

- **Flow:** definiert durch Header-Feld Werte (in Link-, Vermittlungs-, Transportschicht Feldern)
- **Generalisiertes Forwarding**: einfache Paket-Verarbeitungsregeln
  - **Match:** bestimmte Werte in Paket-Headerfeldern
  - **Aktionen:** für entsprechendes Paket: Verwerfen, Weiterleiten, Modifizieren des Pakets oder Senden an den Controller
  - **Priorität:** Zur Unterscheidung überlappender Matches
  - **Counter:** #Bytes und #Pakete



OpenFlow: Flow Table Einträge



Ziel-basiertes Forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	*	51.6.0.8	*	*	*	*	Port6

IP-Datagramme bestimmt für IP-Adresse 51.6.0.8 sollen an Router Port 6 weitergeleitet werden

Firewall:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	*	*	*	*	*	22	drop

Blockieren aller Datagramme an TCP-Port 22 (SSH Port Nr.)

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	*	*	*	*	128.119.1.1	*	*	*	*	*	drop

Blockieren aller Datagramme gesendet von Host 128.119.1.1

Layer 2 Ziel-basiertes Forwarding:

Switch Port	MAC src	MAC dst	Eth type	VLAN ID	VLAN Pri	IP Src	IP Dst	IP Prot	IP ToS	TCP s-port	TCP d-port	Action
*	*	22:A7:23: 11:E1:02	*	*	*	*	*	*	*	*	*	port3

Layer 2 Rahmen mit Ziel MAC-Adresse 22:A7:23:11:E1:02 sollen an Port 3 weitergeleitet werden

- **Match+Action:** Abstraktion vereinigt verschiedene Arten von Geräten

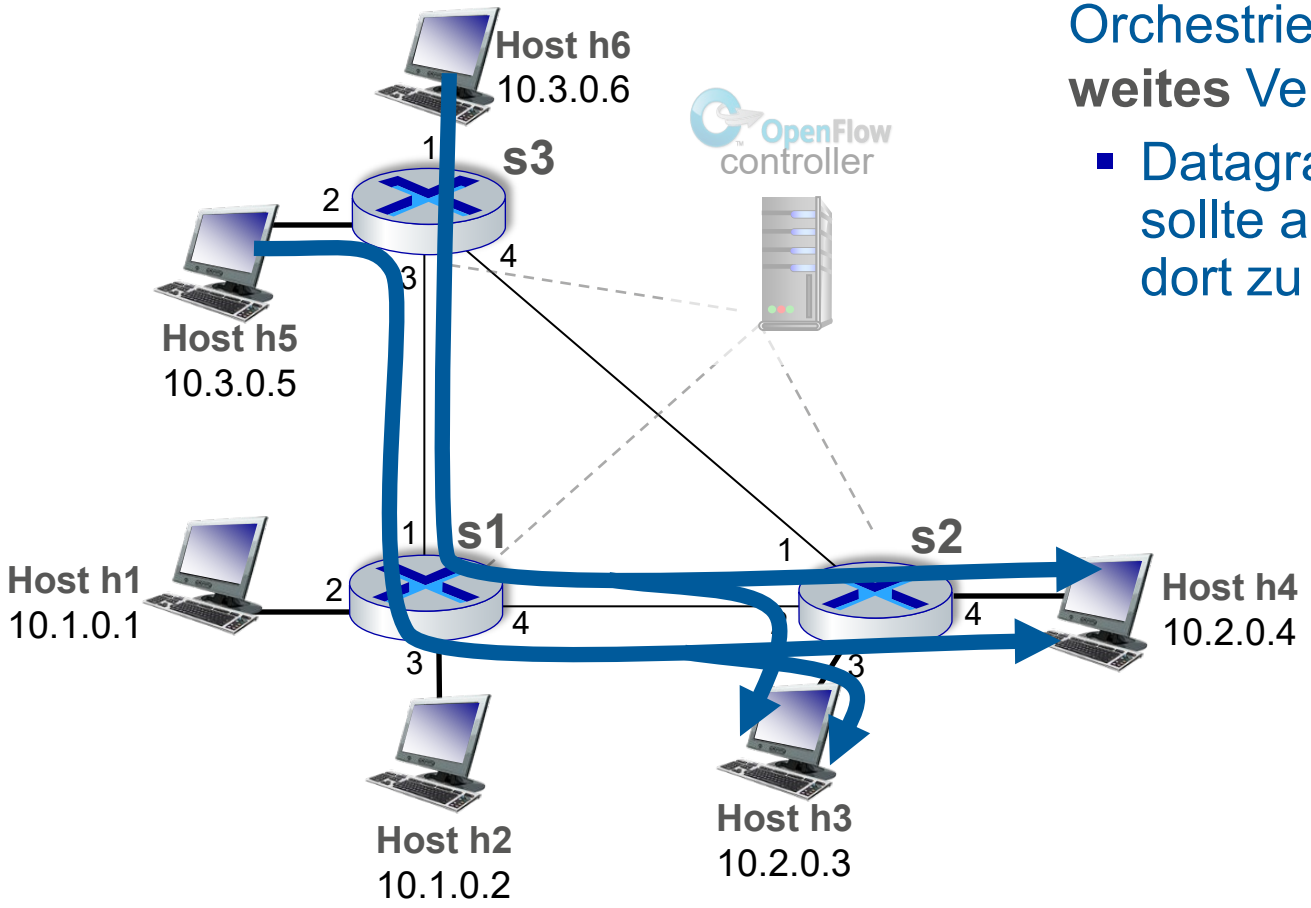
Router
- **Match:** längstes Ziel-IP Präfix
- **Action:** weiterleiten über Link

Switch
- **Match:** Ziel-MAC Adresse
- **Action:** weiterleiten oder fluten

Firewall
- **Match:** IP-Adressen und TCP/UDP Port-Nummern
- **Action:** erlauben oder blockieren

NAT
- **Match:** IP-Adresse und Port
- **Action:** ändern von Adresse und Port

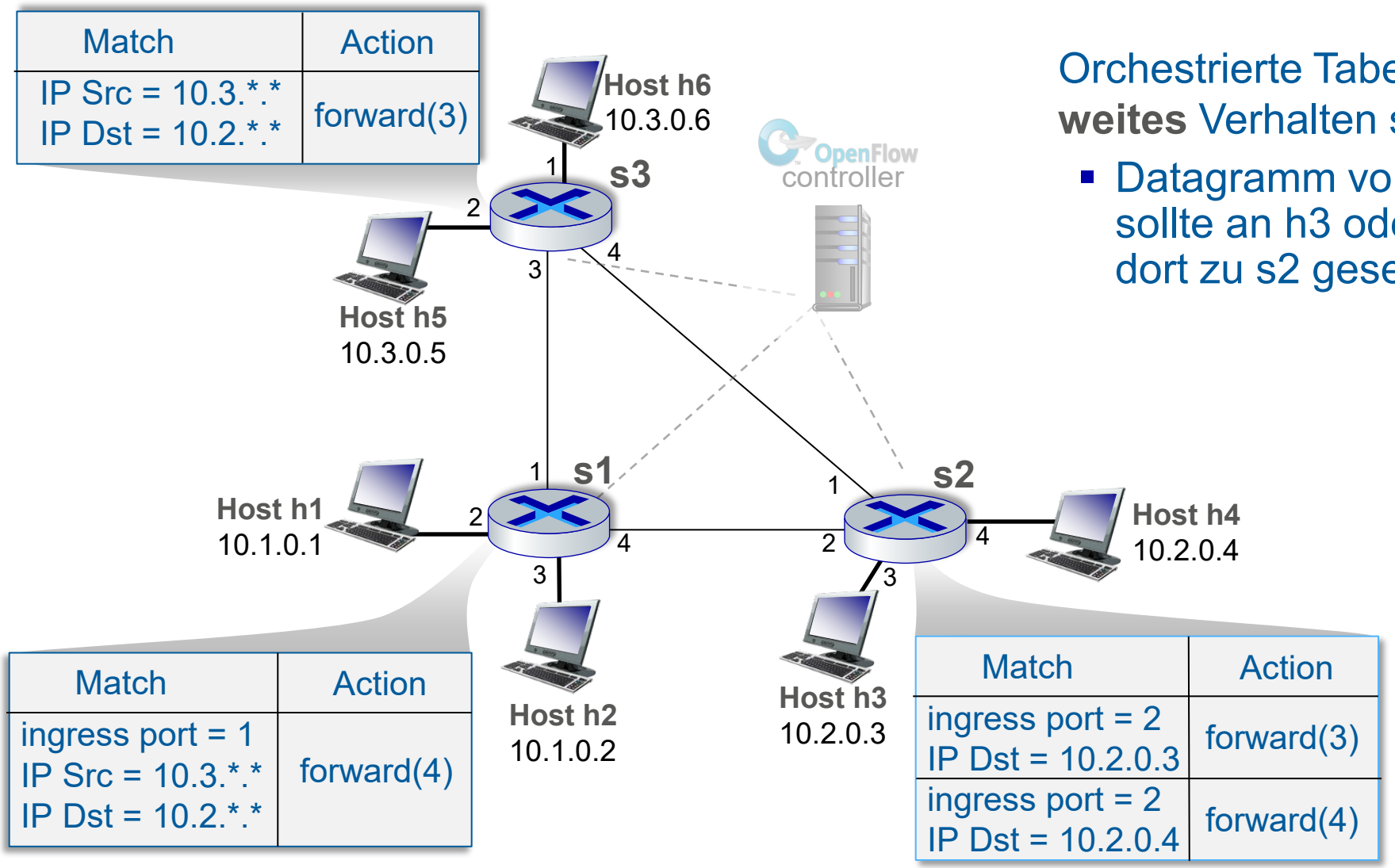
OpenFlow Beispiel



Orchestrierte Tabellen können **Netz-weites** Verhalten schaffen, z.B.,:

- Datagramm von Hosts h5 und h6 sollte an h3 oder h4, via s1 und von dort zu s2 gesendet werden

OpenFlow Beispiel



Orchestrierte Tabellen können **Netz-weites** Verhalten schaffen, z.B.,:

- Datagramm von Hosts h5 und h6 sollte an h3 oder h4, via s1 und von dort zu s2 gesendet werden

Generalisiertes Forwarding: Zusammenfassung

- **“Match + Action”** Abstraktion: Matchen von Bits in ankommenden Paket Header(n) aller Schichten, Aktion durchführen
  - Matchen über viele Felder hinweg (Link-, Vermittlungs-, Transport Schicht)
  - lokale Aktionen: Verwerfen, Weiterleiten, Modifizieren oder Paket an Controller schicken
  - “Programmieren” von **Netz-weiten** Verhalten
- Einfache Form von “Network Programmability”
  - Programmierbares, “per Paket”-Verarbeitung
  - **historische Wurzeln:** Active Networking
  - **heute:** generellere Programmierbarkeit: P4 (siehe p4.org).

- **Vermittlungsschicht: Überblick**
  - Datenpfad
  - Kontrollebene
- **Was steckt in einem Router**
  - Eingangsports, Switching, Ausgangsports
  - Puffer Management, Scheduling
- **IP: das Internet Protokoll**
  - Datagramm Format
  - Adressierung
  - Network Address Translation
  - IPv6
- **Generalized Forwarding, SDN**
  - Match+Action
  - OpenFlow: Match+Action
- **Mittelboxen**
  - Funktionen von Mittelboxen
  - Evolution & Prinzipien der Internet Architektur

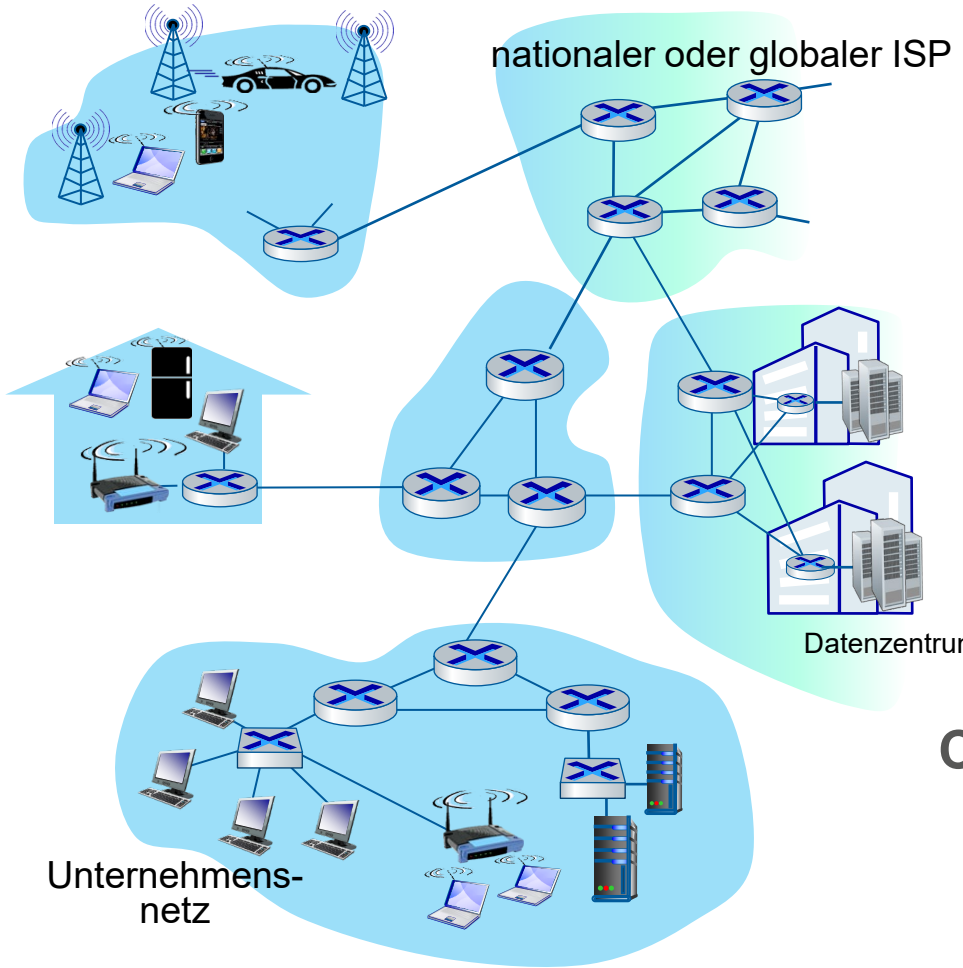
Middlebox (RFC 3234)

“any intermediary box performing functions apart from normal, standard functions of an IP router on the data path between a source host and destination host”

Mittelboxen sind überall!

Firewalls, IDS: Unternehmen, Dienstanbieter, ISPs

NAT: Daheim, Mobilfunk, Unternehmen



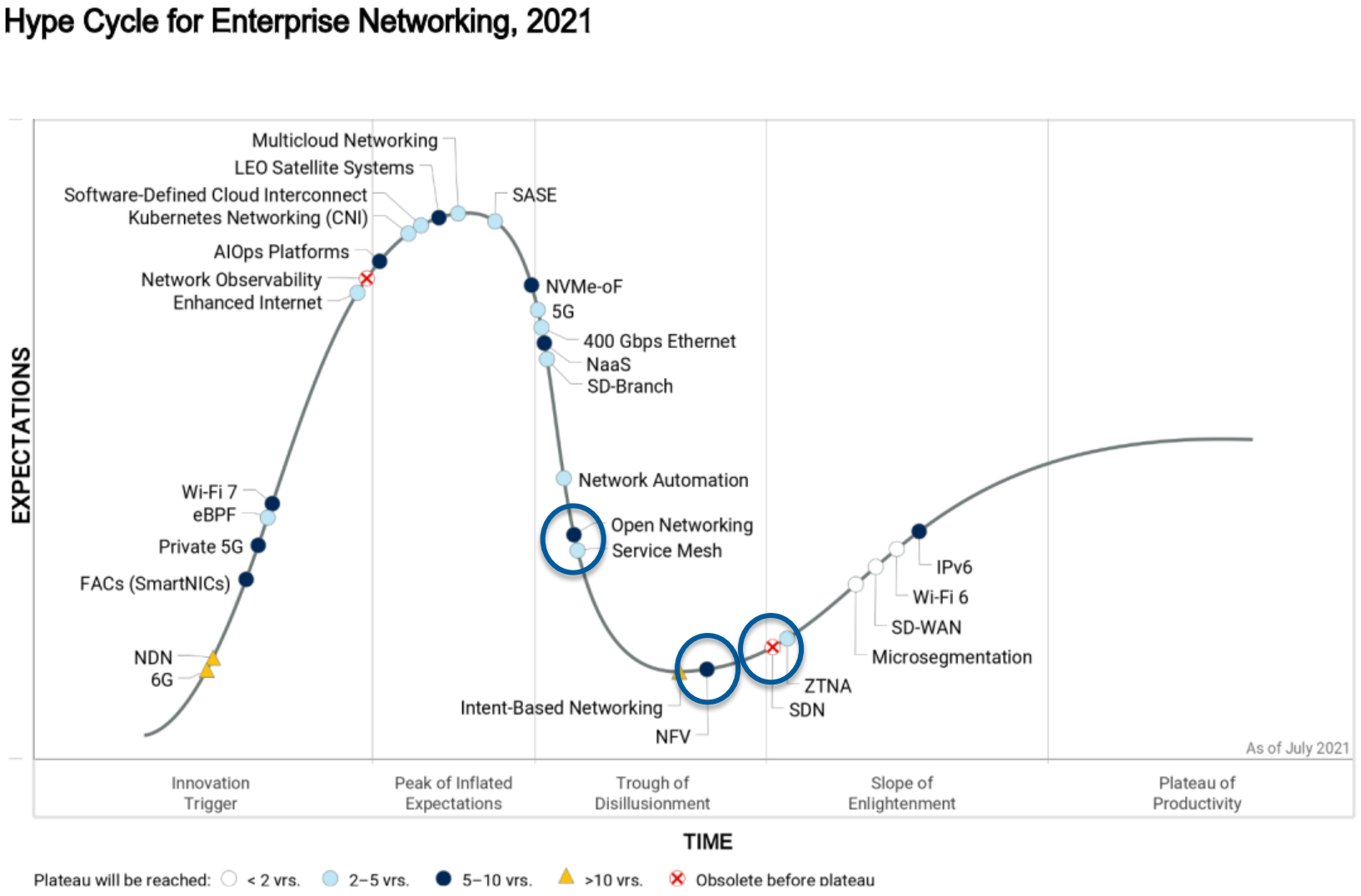
Load Balancer: Unternehmen, Dienstanbieter, Datenzentren, mobile Netze

Applikations-spezifisch: Dienstanbieter, Unternehmen, CDNs

Caches: Dienstanbieter, Datenzentren, mobile Netze, CDNs

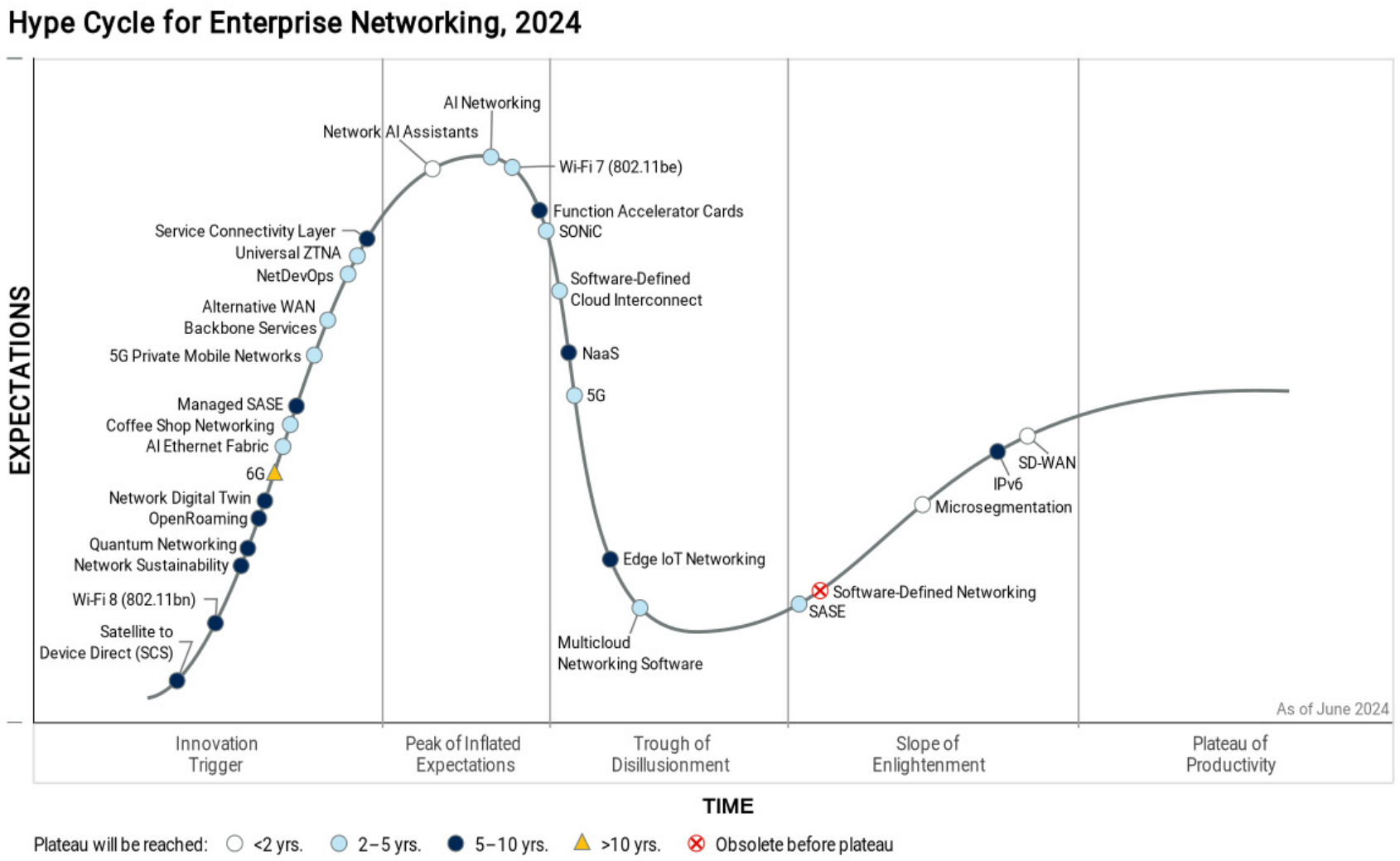
- ursprünglich: proprietäre (geschlossene) Hardware Lösungen
- Trend zu **“Whitebox” Hardware** mit offenen Schnittstellen (APIs)
  - weg von proprietären Hardware Lösungen
  - **Programmierbare, lokale Aktionen** via Match+Action
  - Trend zu Innovation/Differenzierung in Software
- **SDN:** (logisch) zentralisierte Kontrolle und Konfigurationsmanagement, oft in Private/Public Cloud
- **Network Functions Virtualization (NFV):** programmierbare Dienste über “Whitebox” Netz-, Compute- und Storagehardware

Gartner Hype Cycle Networking 2021



Gartner

Gartner Hype Cycle Networking 2024

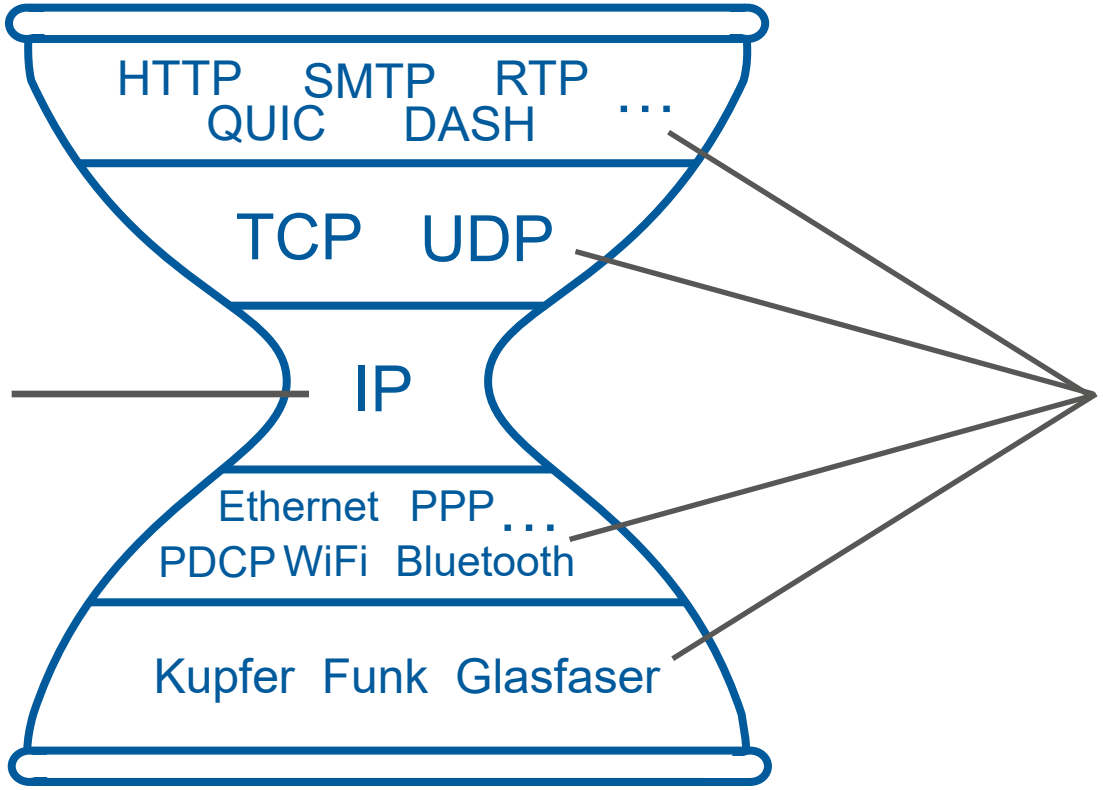


Gartner

Die IP-Sanduhr

Internet “enge Taille”:

- **ein** Vermittlungsschicht Protokoll: IP
- **muss** in allen (Milliarden) Internet-verbundenen Geräten implementiert sein

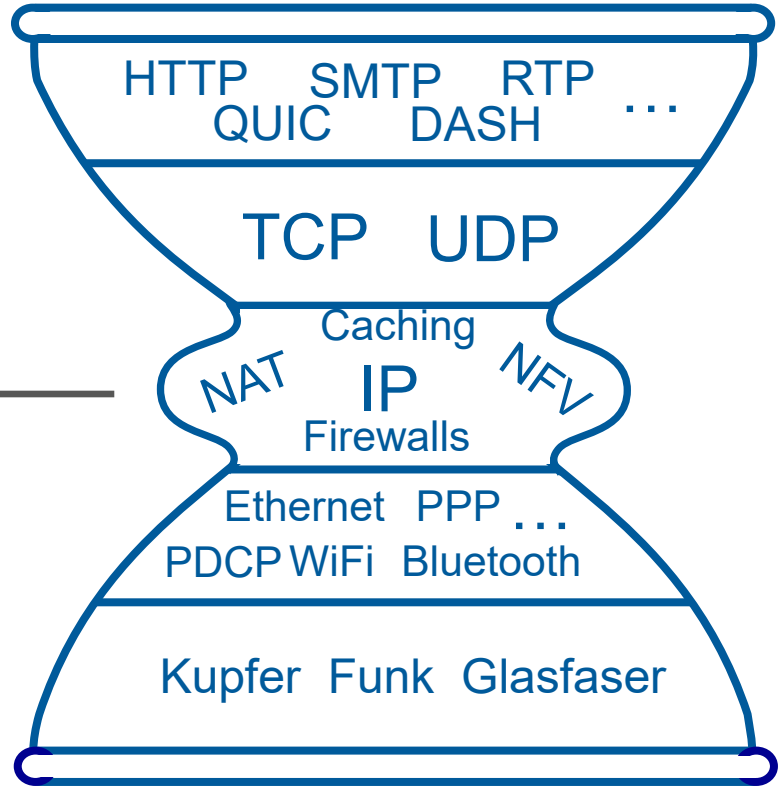


viele Protokolle in den Bitübertragungs-, Sicherungs-, Transport-, und Applikationsschichten

Die IP-Sanduhr, im mittleren Alter

Internet “Rettungsring” im mittleren Alter?

- Mittelboxen, im Netz operierend



RFC 1958

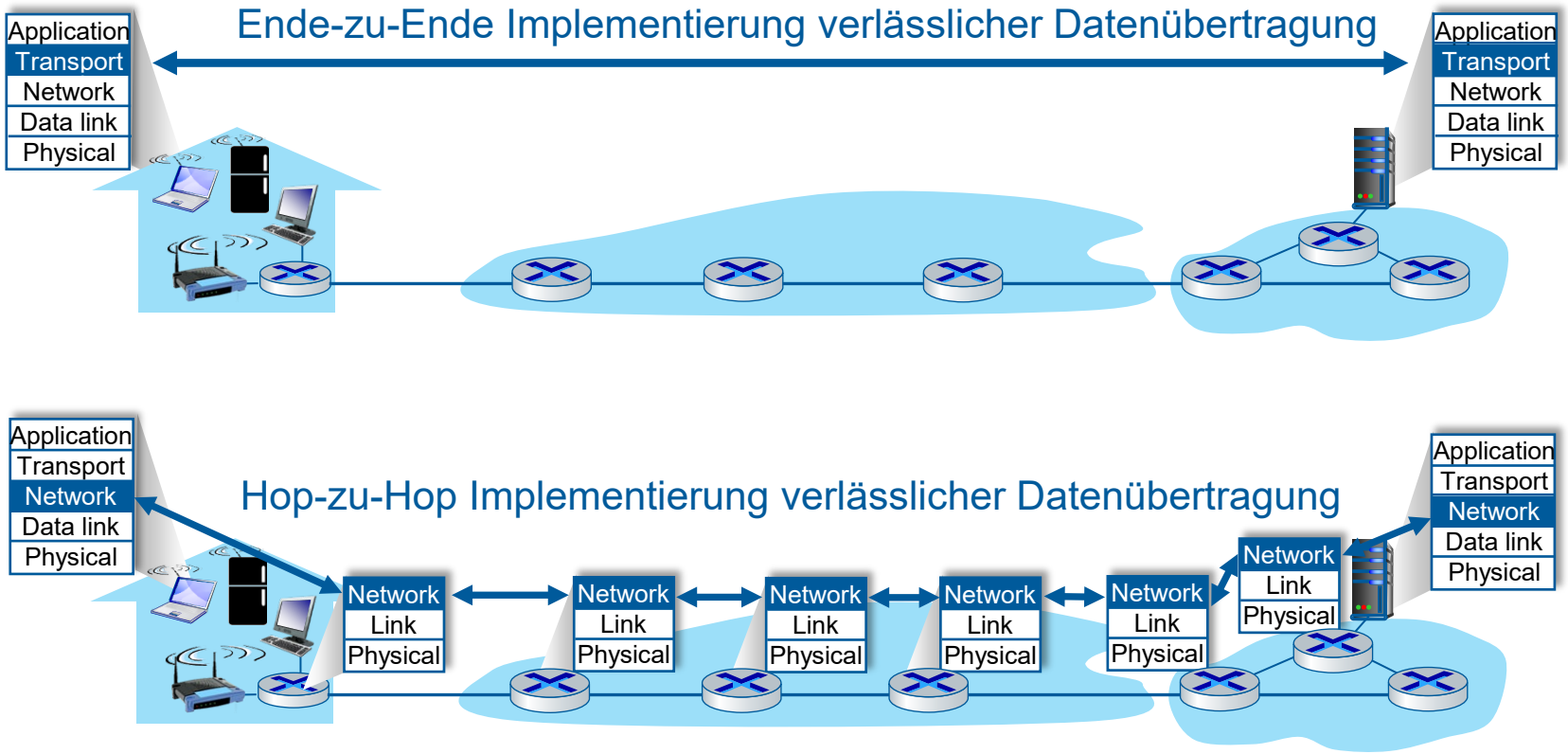
"Many members of the Internet community would argue that there is no architecture, but only a tradition, which was not written down for the first 25 years (or at least not by the IAB). However, in very general terms, the community believes that **the goal is connectivity, the tool is the Internet Protocol, and the intelligence is end to end rather than hidden in the network."**

Drei Schlüsselüberzeugungen:

- einfache Konnektivität
- IP-Protokoll: die enge Taille
- Intelligenz, Komplexität am Rand des Netzes

Das Ende-zu-Ende Prinzip

- Bestimmte Netzfunktionen (z.B., Verlässliche Datenübertragung, Staukontrolle) kann **im Netz** oder **am Rand des Netzes** implementiert werden

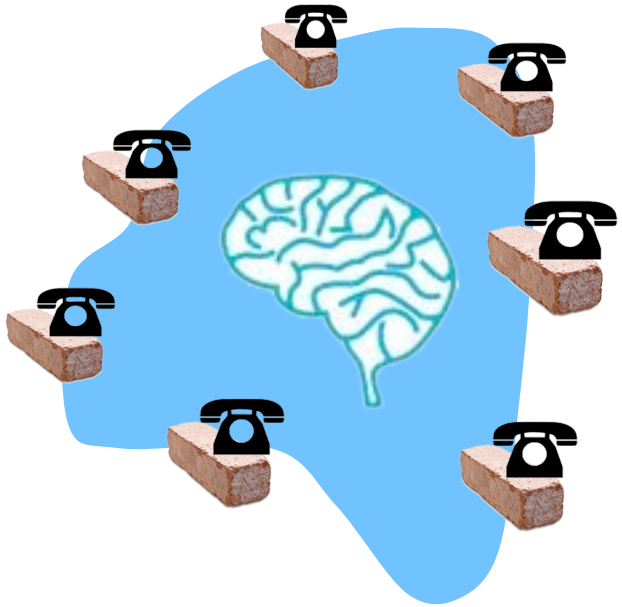


Das Ende-zu-Ende Prinzip

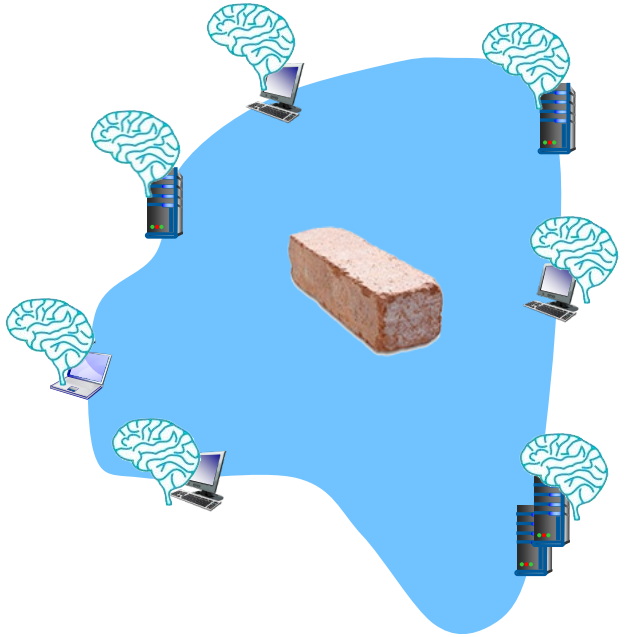
- Bestimmte Netzfunktionen (z.B., Verlässliche Datenübertragung, Staukontrolle) kann **im Netz** oder **am Rand des Netzes** implementiert werden

> "The function in question can completely and correctly be implemented only with the knowledge and help of the application standing at the end points of the communication system. Therefore, providing that questioned function as a feature of the communication system itself is not possible. (Sometimes an incomplete version of the function provided by the communication system may be useful as a performance enhancement.)
>
> We call this line of reasoning against low-level function implementation the "end-to-end argument."
>
> **Saltzer, Reed, Clark 1981**

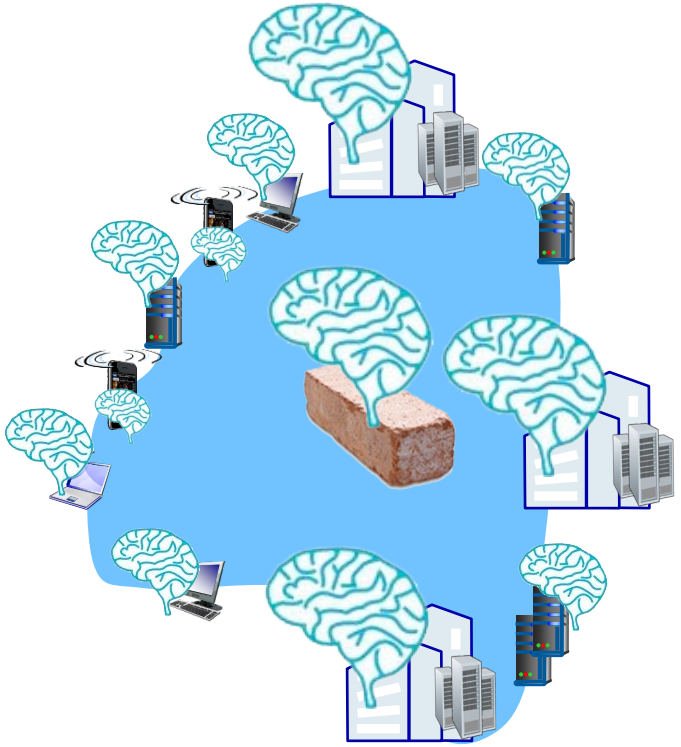
Wo liegt die Intelligenz?



20th Jhd. Telefonnetz:
- Intelligenz auf den Switches



Internet (vor 2005)
- Intelligenz am Rand



Internet (nach 2005)
- Programmierbare Netzgeräte
- Intelligenz, riesige Applikationsinfrastruktur am Rand