TECHNISCHE HOCHSCHULE DEGGENDORF

Netzwerktechnik und IT-Netze

Bachelor Angewandte Informatik
Wintersemester 2018/2019

Prof. Dr. Andreas Fischer
andreas.fischer@th-deg.de

Inhalt

Die Netzwerkschicht

Überblick

- Aufgaben der Netzwerkschicht
- Das Internet Protokoll
- IP Adressen und Adressvergabe
- Funktionsweise eines Routers
- Routing Protokolle
  - Link-State: OSPF
  - Distance Vector: RIP
  - Hierarchisch: BGP
- Flexibilisierung der Netzwerkschicht
  - Software-defined Networking
  - Network Function Virtualization

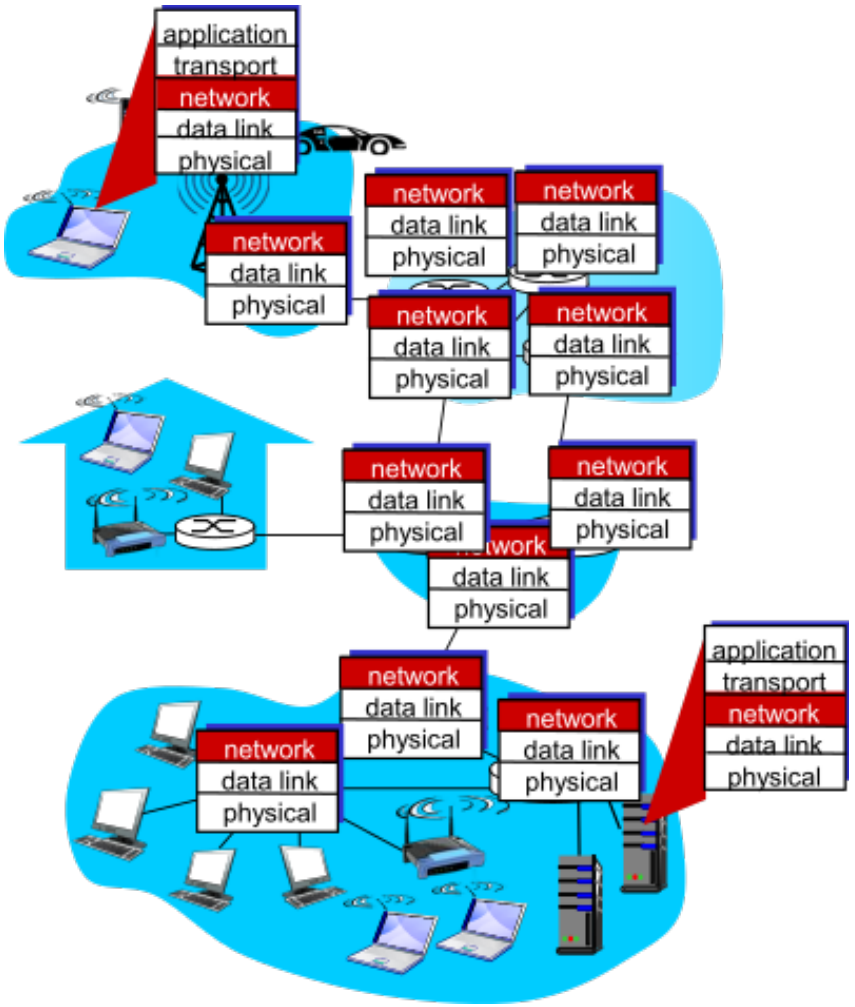
Die Netzwerkschicht

Überblick

- Aufgaben der Netzwerkschicht
- Das Internet Protokoll
- IP Adressen und Adressvergabe
- Funktionsweise eines Routers
- Routing Protokolle
  - Link-State: OSPF
  - Distance Vector: RIP
  - Hierarchisch: BGP
- Flexibilisierung der Netzwerkschicht
  - Software-defined Networking
  - Network Function Virtualization

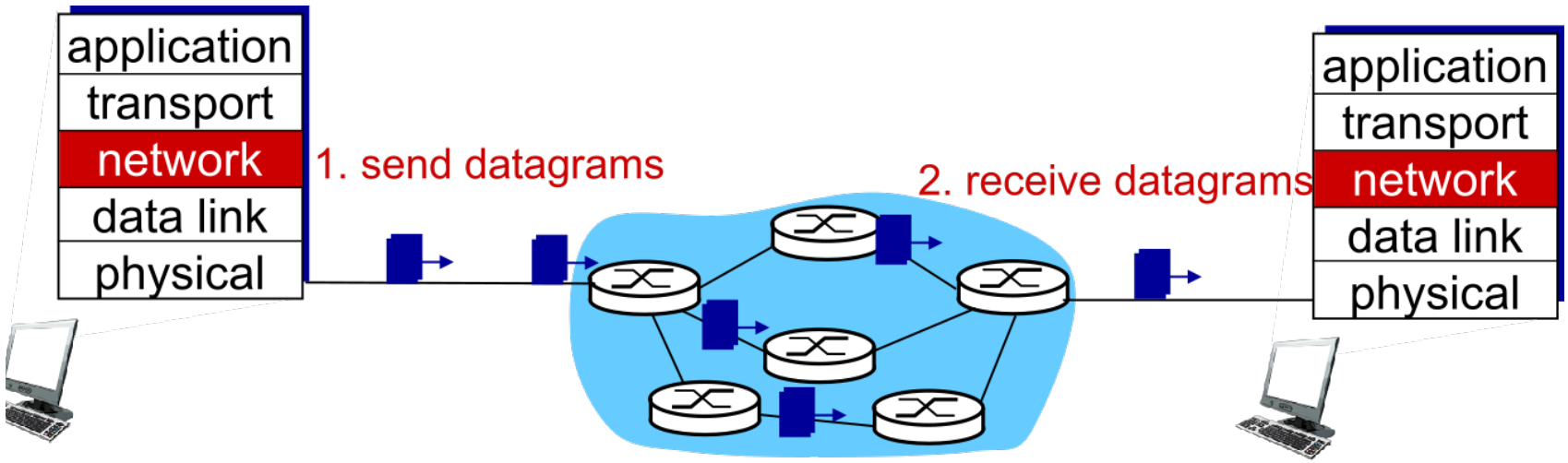
Aufgaben der Netzwerkschicht

- Segmente von Sender an Empfänger übertragen
  - Sender: Segmente in Datagramme packen und versenden
  - Empfänger: Segmente an Transportschicht weitergeben
- Weiterleitung von Paketen
  - Datagramme von Rechner zu Rechner weitergeben
  - Mittels Header-Infos nächsten Hop bestimmen



Paketorientierte Netzwerke

- Kein Verbindungsaufbau auf Netzwerkschicht
- Router kennen Zustand der Ende-zu-Ende Kommunikation nicht
- Pakete werden individuell nur anhand der Zieladresse weitergeleitet



Die Netzwerkschicht

Überblick

- Aufgaben der Netzwerkschicht
- Das Internet Protokoll
- IP Adressen und Adressvergabe
- Funktionsweise eines Routers
- Routing Protokolle
  - Link-State: OSPF
  - Distance Vector: RIP
  - Hierarchisch: BGP
- Flexibilisierung der Netzwerkschicht
  - Software-defined Networking
  - Network Function Virtualization

Das Internet Protokoll (IP)

Aufgaben:

- Adressierung von Rechnern
- Verwaltungsinformationen für Pakete
  - Quality of Service (QoS): Sollen Pakete bevorzugt behandelt werden?
  - Fragmentierung: Müssen Pakete aufgeteilt werden?
  - Time to live (TTL): Wann soll ein Paket verworfen werden?
- Interoperabilität zwischen unterschiedlichen physikalischen Netzwerken

Umsetzung:

- Paket wird zum Datagramm: Ein weiterer Header
- Abstraktionsschicht oberhalb der Vermittlungsschicht

Entwicklung des Internet Protokolls

1974 TCP als allgemeines Übertragungsprotokoll
- Übernimmt Aufgaben der Netzwerk- und der Transportschicht
- 20 Bit Adressen (4 Bit Netzwerk + 16 Bit Host)

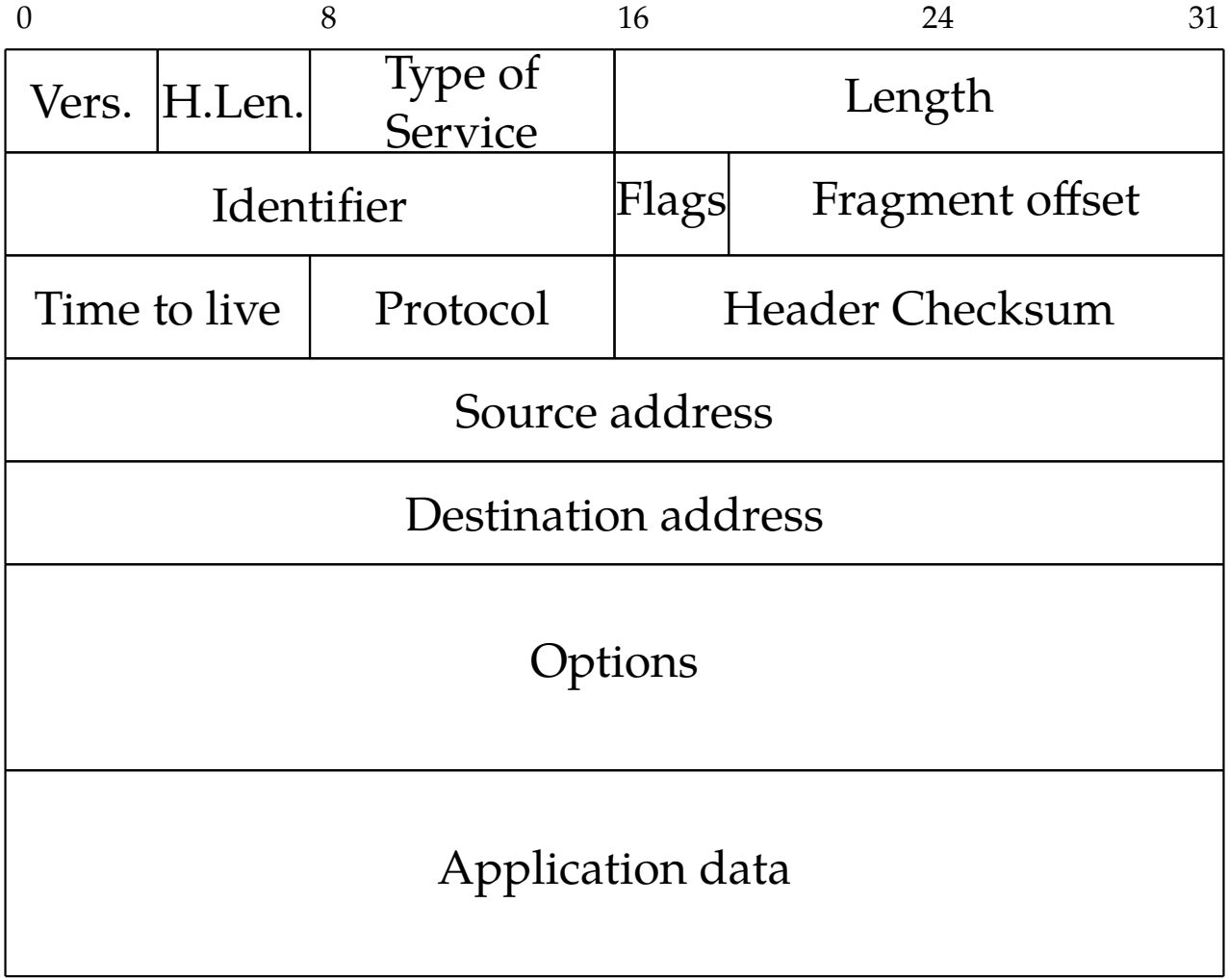
1978 Aufteilung in Transportprotokoll und Netzwerkprotokoll: TCP/IP
- IP (ohne Version) $\rightarrow$ IPv2 $\rightarrow$ IPv4
- 32 Bit Adressen (8 Bit Netzwerk + 24 Bit Host)

1979 IPv5: Internet Stream Protocol (kaum genutzt)

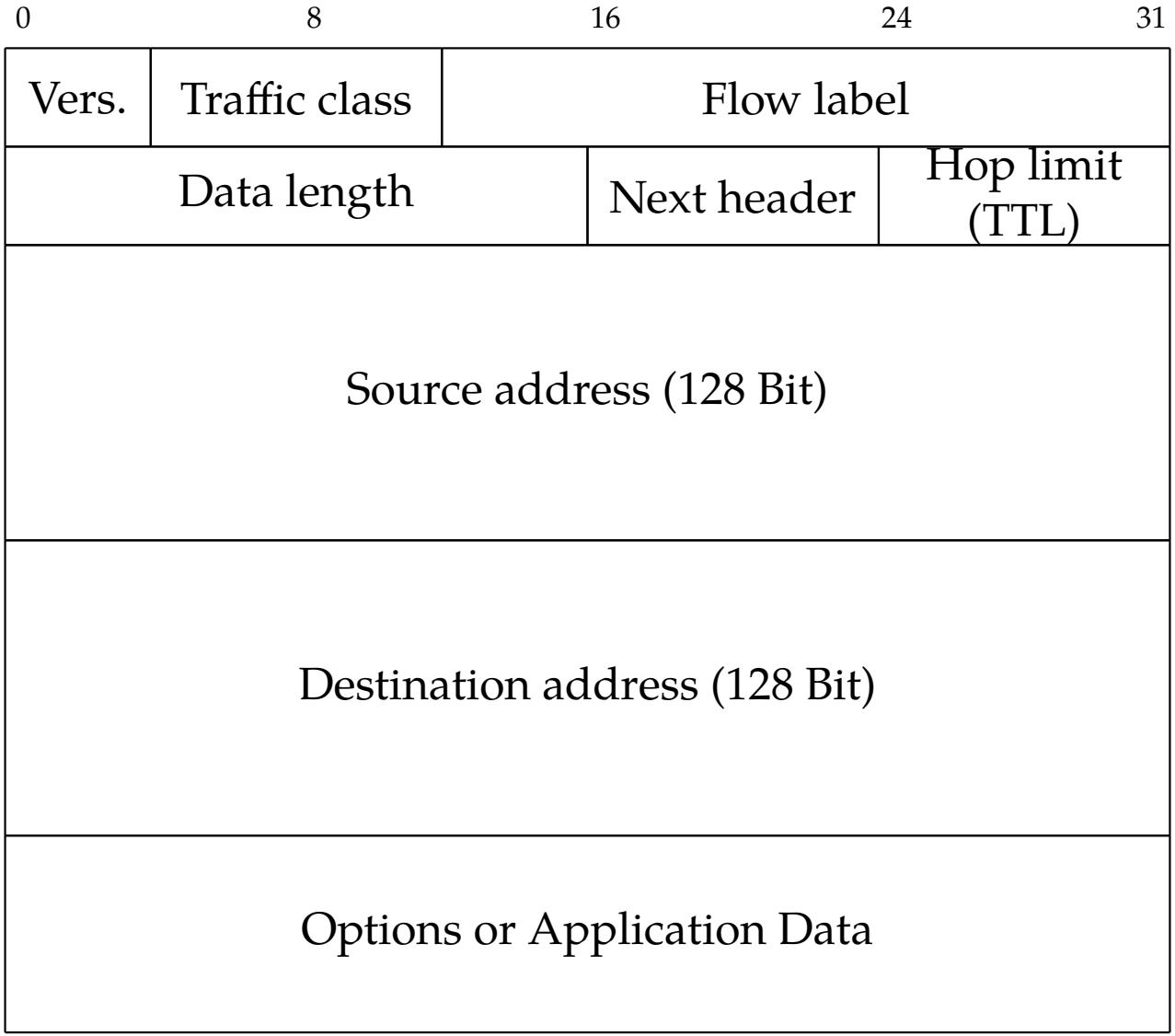
1998 IPv6: Draft Standard

2017 IPv6: Verbindlicher Standard

Aufbau eines IPv4 Datagramms



Aufbau eines IPv6 Datagramms



Wichtige Änderungen in IPv6

Mehr Adressen verfügbar

- 32 Bit: $2^{32} \simeq 4 \cdot 10^9$ (Weltbevölkerung: $7,6 \cdot 10^9$)
- 128 Bit: $2^{128} \simeq 3 \cdot 10^{38}$ (Erdoberfläche: $5.1 \cdot 10^{20} mm^2$)

Schnellere Verarbeitung

- Keine Checksumme mehr (muss an jedem Router berechnet werden)
- Feste Headerlänge

Aufräumen

- Fragmentierung fällt weg (Aufgabe der Endpunkte)
- Flow labels statt vordefinierter Klassen

IPv6 als Beispiel für die Verknöcherung des Internets

- IPv6 bereits seit 20 (!) Jahren spezifiziert
- Henne-Ei Problem bei der Umsetzung:

  ISP: "Die Anwender nutzen es nicht"

  Anwender: "Mein ISP unterstützt es nicht"
- Änderung am Netzwerkprotokoll zieht viele weitere Änderungen nach sich
  - DNS: Neue AAAA Einträge
  - Ping, Traceroute: Neue Implementierungen
  - Lokale Software
    - Jede Serversoftware muss eigene IP Adresse kennen/parsen
    - Jedes UDP Programm muss IP Adresse des Kommunikationspartners kennen
  - TCP und UDP in Details stark mit IPv4 verbandelt

Lösung ???

Network Address Translation (NAT)

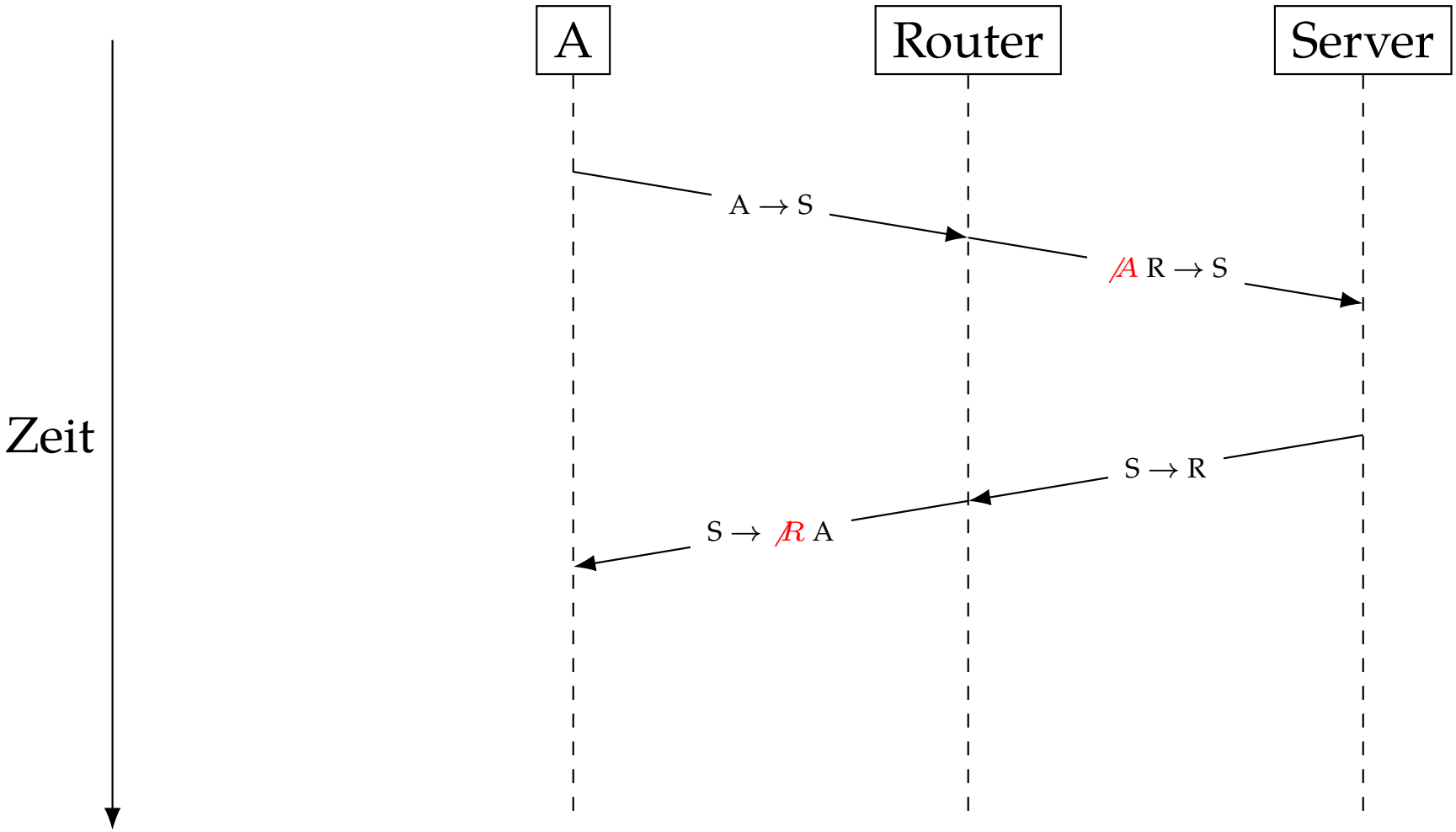
Idee:

- Drei Adressbereiche sind privat deklariert:
  - 10.0.0.0/8
  - 172.16.0.0/12
  - 192.168.0.0/16
  - Dürfen mehrfach vergeben werden
  - Allerdings nicht von extern erreichbar
- Ein Router ("NAT-Box") könnte als "Vermittler" agieren und die Kommunikation mit der Aussenwelt organisieren

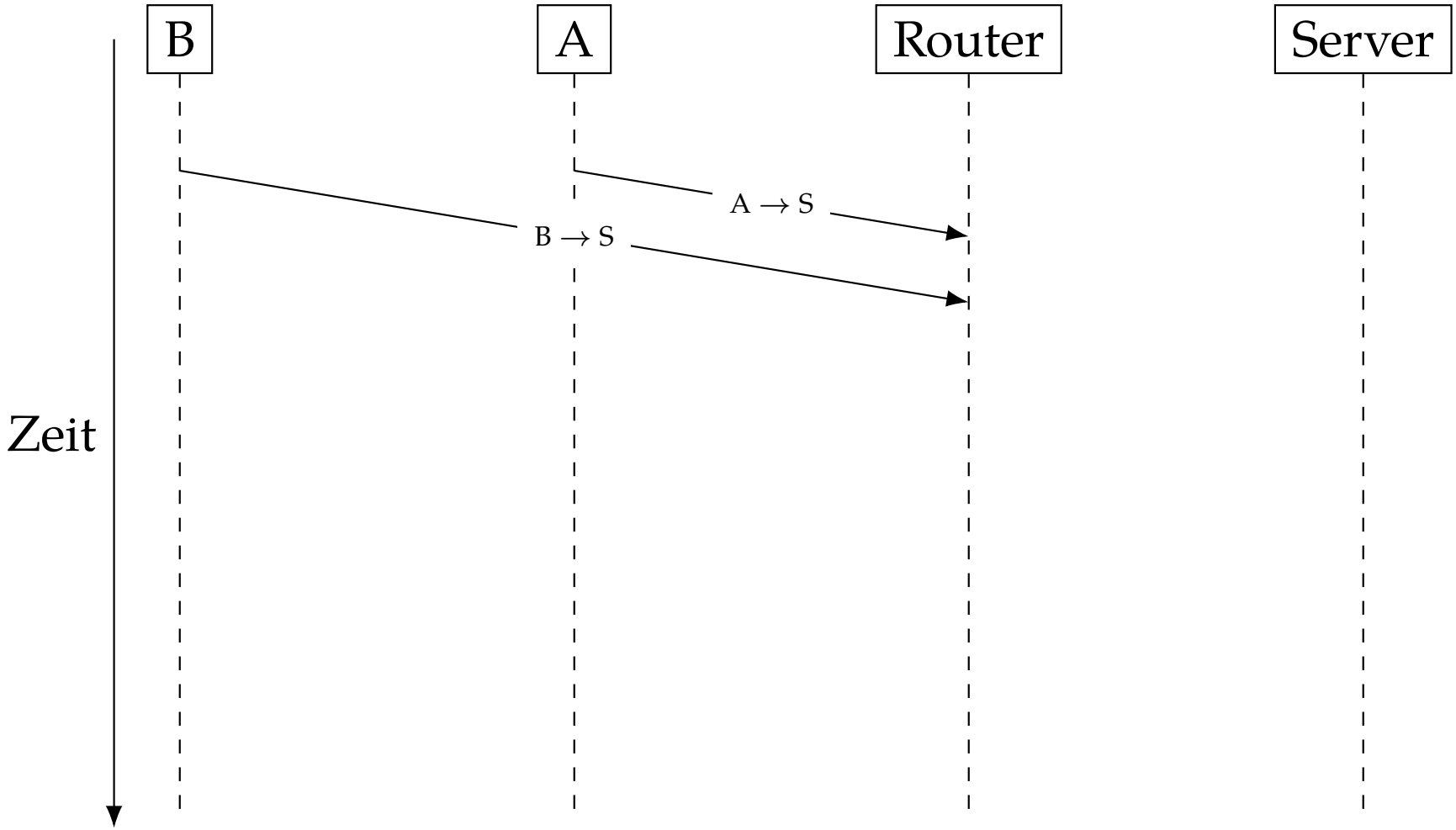
Umsetzung:

- Intern: Privater IP Adressbereich
- Extern: Eine IP Adresse für die NAT-Box
- NAT-Box ersetzt IP Adresse bei allen Paketen

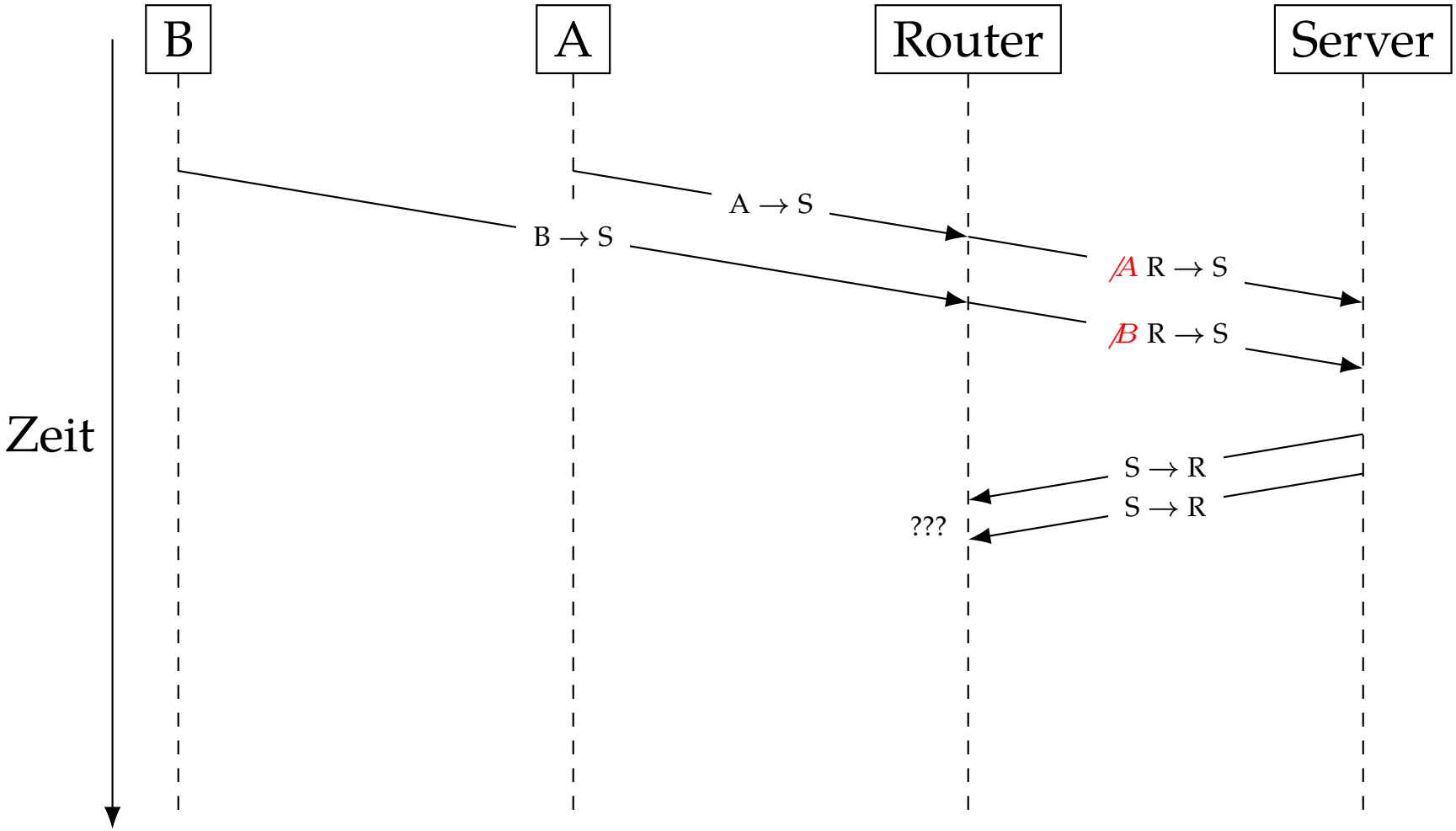
NAT: Problem



NAT: Problem



NAT: Problem



NAT: Lösung

Die NAT-Box muss Antworten zuordnen können

- Internes Mapping notwendig: Antwort X gehört zu Host A
- Die Information muss in der Antwort stecken …
  - …ohne das Protokoll zu verändern (sonst: IPv6)
  - …ohne Beteiligung des Servers (sonst nicht mehr transparent)

Umsetzung: Verletzung des Schichtenmodells

- Daten sind überwiegend entweder TCP oder UDP
- TCP und UDP Ports sind frei wählbar
- Router ersetzt Quellport durch zufälligen eigenen Port
- Internes Mapping: Port $\rightarrow$ lokaler Host

NAT: Lösung

Die NAT-Box muss Antworten zuordnen können

- Internes Mapping notwendig: Antwort X gehört zu Host A
- Die Information muss in der Antwort stecken …
  - …ohne das Protokoll zu verändern (sonst: IPv6)
  - …ohne Beteiligung des Servers (sonst nicht mehr transparent)

Umsetzung: Verletzung des Schichtenmodells

- Daten sind überwiegend entweder TCP oder UDP
- TCP und UDP Ports sind frei wählbar
- Router ersetzt Quellport durch zufälligen eigenen Port
- Internes Mapping: Port → lokaler Host

Das ist ein Hack!!! (aber funktioniert)

Die Netzwerkschicht

Überblick

- Aufgaben der Netzwerkschicht
- Das Internet Protokoll
- IP Adressen und Adressvergabe
- Funktionsweise eines Routers
- Routing Protokolle
  - Link-State: OSPF
  - Distance Vector: RIP
  - Hierarchisch: BGP
- Flexibilisierung der Netzwerkschicht
  - Software-defined Networking
  - Network Function Virtualization

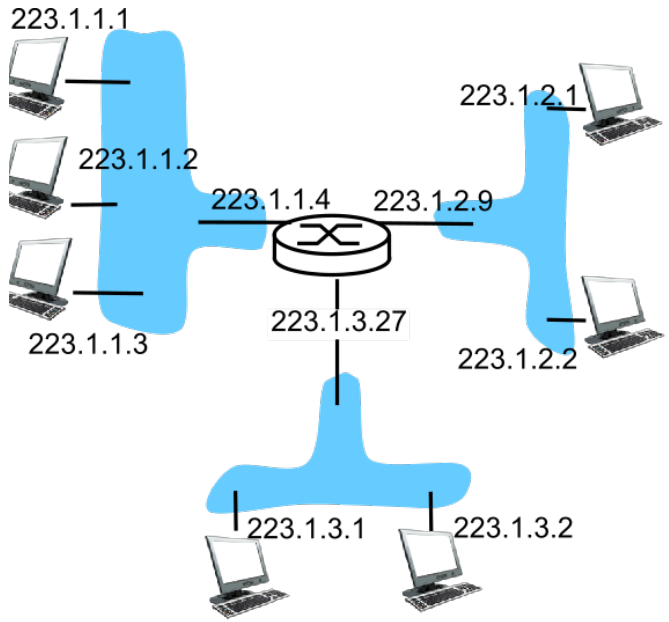
IP Adressen

Eine IP Adresse für jedes Netzwerkinterface

- Interface: “Ende eines Links”
  - Endgeräte haben oft mehrere Interfaces
  - Router haben immer mehrere Interfaces
- IP adressiert nicht den Rechner sondern das Interface
- Adresslänge und -notation abhängig von v4/v6

  IPv4: 223.1.1.1

  IPv6: fd9e:21a7:a92c::1



223.1.1.1: 32 Bit Adresse:

11011111	00000001	00000001	00000001
223	1	1	1

fd9e:21a7:a92c::1: 128 Bit Adresse:

1111	1101	1001	1110	…
f	d	9	e	…

IP Adressen: Notation

IPv4 Adressen

- 32 Bit = 4 Byte
- "Dotted decimal notation": A.B.C.D

11011111	00000001	00000001	00000001
223	1	1	1

IPv6 Adressen

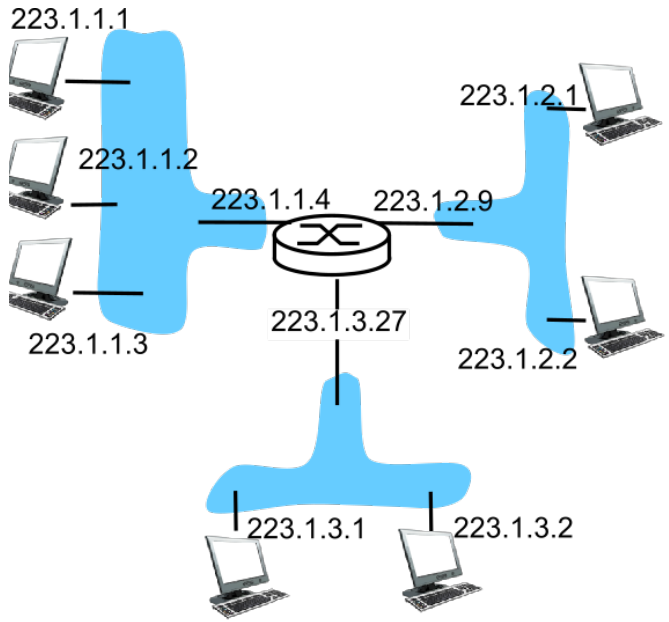
- 128 Bit = 16 Byte
- Notation als Hexadezimalzahlen (je 2 Byte):
  fd9e:21a7:a92c:0:0:0:0:1
- 0-Sequenzen dürfen *einmal* durch "::" abgekürzt werden:
  fd9e:21a7:a92c:0:0:0:0:1 $\rightarrow$ fd9e:21a7:a92c::1

1111	1101	1001	1110	…
f	d	9	e	…

Subnetze

Topologische Information

- In welchem Teil des Internets befindet man sich?
  - Adresse nicht nur für Rechner, sondern auch für Netze
  - Ursprünglich: Explizit ausgewiesen (8Bit/24Bit)
  - Heute: Dynamischer Anteil der IP Adresse
- Subnetz: Kleinstes Netzwerk (atomar)
  - Keine weiteren Router dazwischen
  - Rechner haben "ähnliche" IP Adressen



Dieses Netzwerk enthält drei Subnetze

Adressierung von Netzwerken: CIDR

Classless Inter-Domain Routing (CIDR):

- Vorderer Teil der Adresse spezifiziert das Netzwerk
- Angabe des Netzwerks durch Netzwerkmaske
  - Welche Bits adressieren das Netzwerk?
  - Entweder als Bytes: 255.255.255.0
  - Oder als Bitanzahl: /24
- Hierarchische Organisation von Netzwerken
  - Ein /16 Netz kann mehrere /24 Netze enthalten
  - Hohe Flexibilität gegeben

Beispiel: 200.23.16.0/23 (IPv6 äquivalent)

IP Adressvergabe

Problem: Wie erhält man eine Adresse?

- Adressen müssen weltweit eindeutig sein (sonst funktioniert die Zustellung nicht)
- Adresse sollte topologische Information beinhalten
  - Hinweis: Welcher Teil des Internets?
  - Sonst muss jede Adresse speziell behandelt werden

Zwei Varianten der Adressvergabe:

- Statisch vom Administrator konfiguriert
  $\rightarrow$ Gängig für Server mit festen Adressen
- Dynamisch per DHCP erhalten
  $\rightarrow$ Gängig für Clients mit häufig wechselnden Adressen

Das Dynamic Host Configuration Protocol (DHCP)

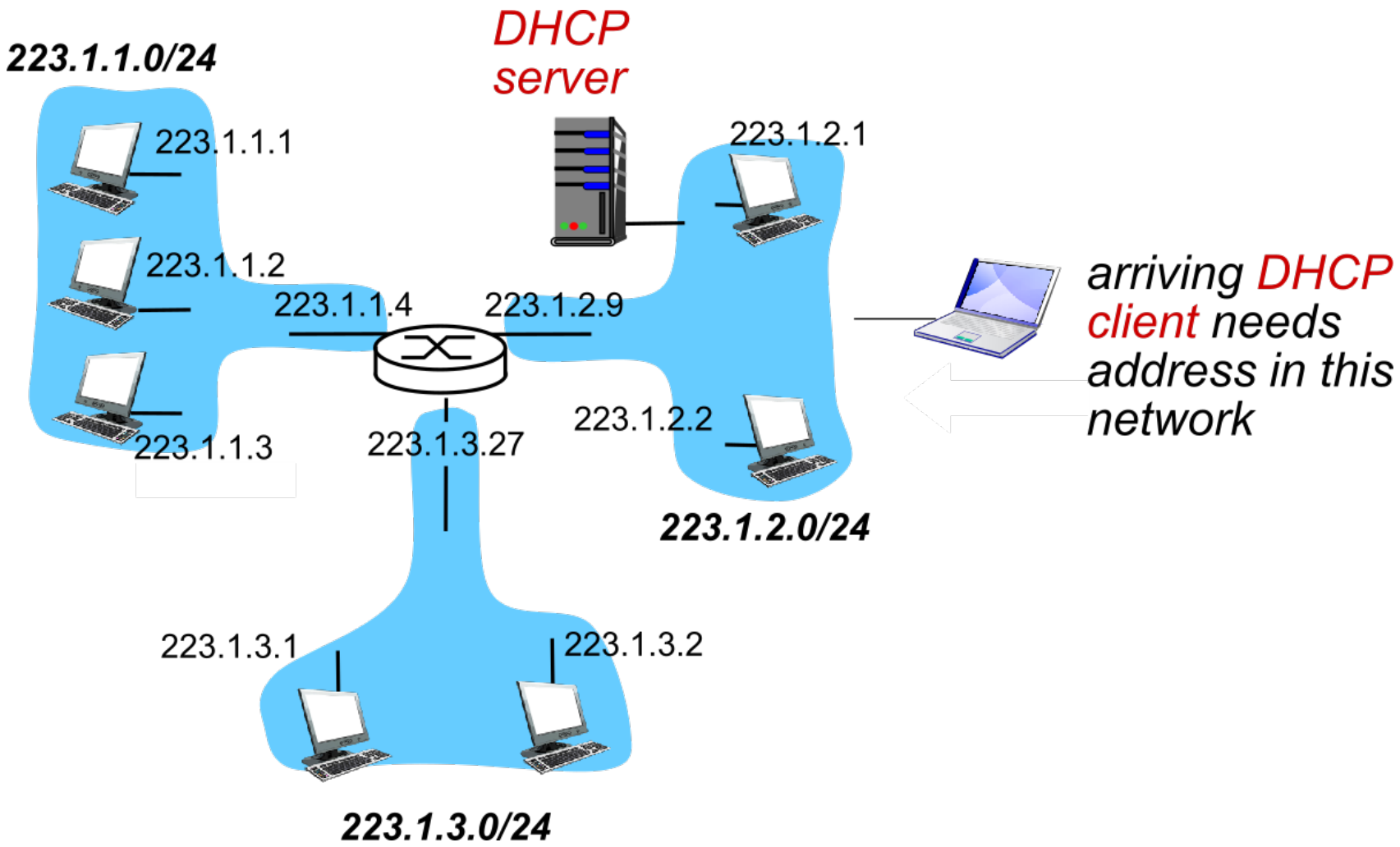
Ziel: Rechner erhält dynamisch Adresse aus Adresspool zugewiesen

- Keine Interaktion durch Menschen notwendig
- Mehr Clients als Adressen möglich
  - Voraussetzung: Nicht alle gleichzeitig aktiv
  - Unterstützt Mobilität: eine Adresse pro Netz

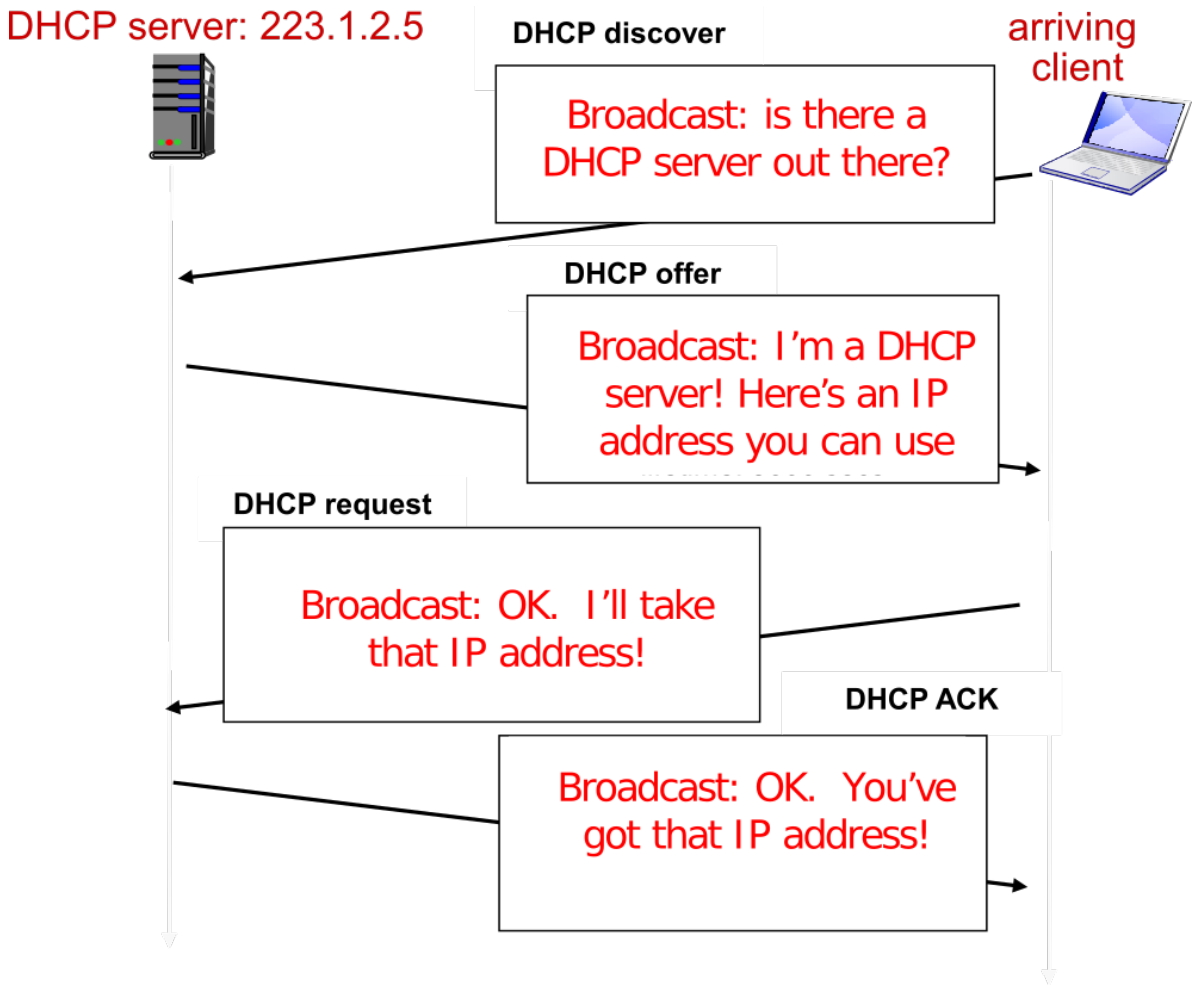
DHCP Überblick:

1. Hosts senden "DHCP discover" an gesamtes Subnetz (Broadcast)
2. DHCP Server im Subnetz antwortet mit "DHCP offer"
3. Host bittet um IP Adresse mit "DHCP request"
4. DHCP Server bestätigt mit "DHCP ACK"

DHCP Szenario



DHCP Ablauf


DHCP server: 223.1.2.5
DHCP discover
arriving client
Broadcast: is there a DHCP server out there?
DHCP offer
Broadcast: I'm a DHCP server! Here's an IP address you can use
DHCP request
Broadcast: OK. I'll take that IP address!
DHCP ACK
Broadcast: OK. You've got that IP address!

DHCP: Nicht nur IP Adressen

Problem:

- Neuer Client benötigt viele Informationen um kommunizieren zu können
- Je ein eigenes Protokoll wäre sinnlos

Über DHCP verfügbare Informationen

- IP Adresse
- Netzwerkadresse (Netzwerkmaske)
- Default Gateway (erster Router)
- Lokaler DNS Server
- …

Adressvergabe für Netzwerke

Problem:

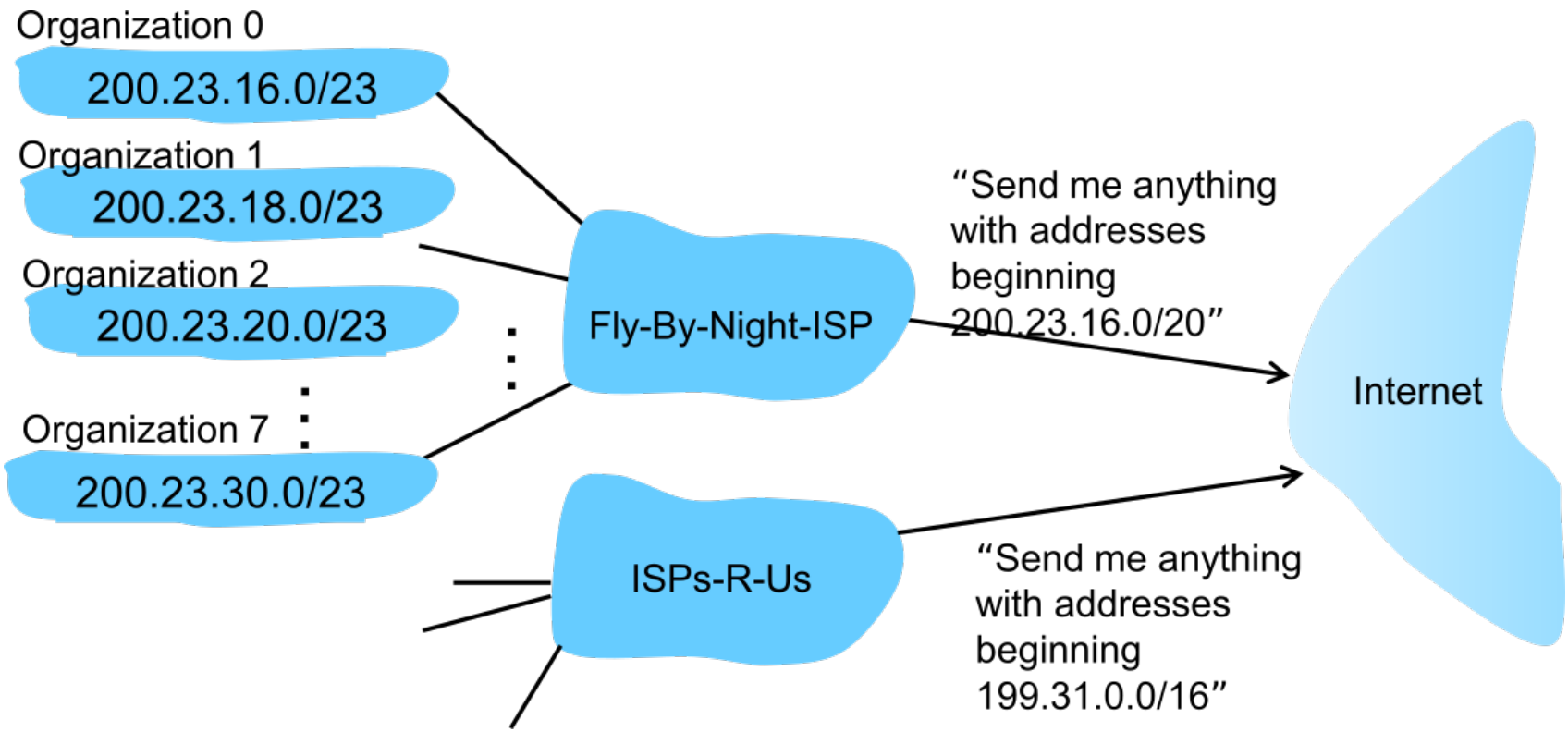
- Woher kennt der Administrator die richtigen Adressen?
- Woher kennt der DHCP Server den richtigen Adresspool?

Antwort: Vom ISP zugewiesen

ISP's block	11001000 00010111 00010000	00000000	200.23.16.0/20
Organization 0	11001000 00010111 00010000	00000000	200.23.16.0/23
Organization 1	11001000 00010111 00010010	00000000	200.23.18.0/23
Organization 2	11001000 00010111 00010100	00000000	200.23.20.0/23
...	.....	....	....
Organization 7	11001000 00010111 00011110	00000000	200.23.30.0/23

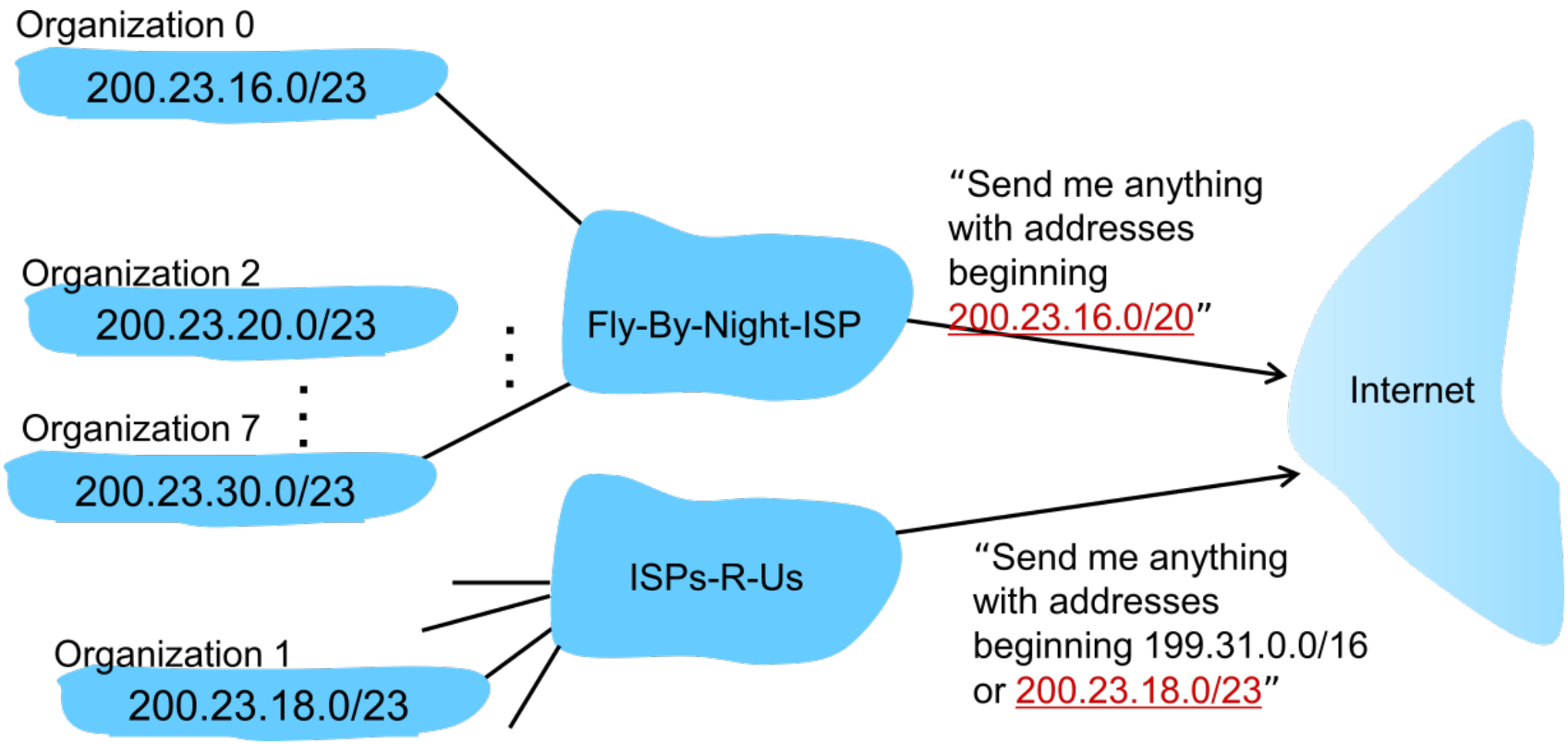
Hierarchische Adressierung

- Hierarchische Adressierung ermöglicht effizientes Routing
- Logisches Adressschema entspricht organisatorischer Aufteilung



Hierarchische Adressierung: Speziellere Routen

- Problem: Organisationsschema kann sich ändern
- Lösung: Speziellere Route "gewinnt" (Longest-Prefix)



Die Netzwerkschicht

Überblick

- Aufgaben der Netzwerkschicht
- Das Internet Protokoll
- IP Adressen und Adressvergabe
- Funktionsweise eines Routers
- Routing Protokolle
  - Link-State: OSPF
  - Distance Vector: RIP
  - Hierarchisch: BGP
- Flexibilisierung der Netzwerkschicht
  - Software-defined Networking
  - Network Function Virtualization

Endgeräte vs. Router

Endgeräte

- Nehmen Daten von der Transportschicht entgegen
- Müssen Pakete mit Adressinformation versehen
- Müssen Pakete empfangen, dekodieren und Daten an die Transportschicht weiterreichen

Router

- Reichen Pakete von Gerät zu Gerät weiter
- Tauschen untereinander Informationen über verfügbare Links aus
- Ermitteln den aktuell “besten” Pfad durch das Netzwerk

Die zwei Kernfunktionen eines Routers

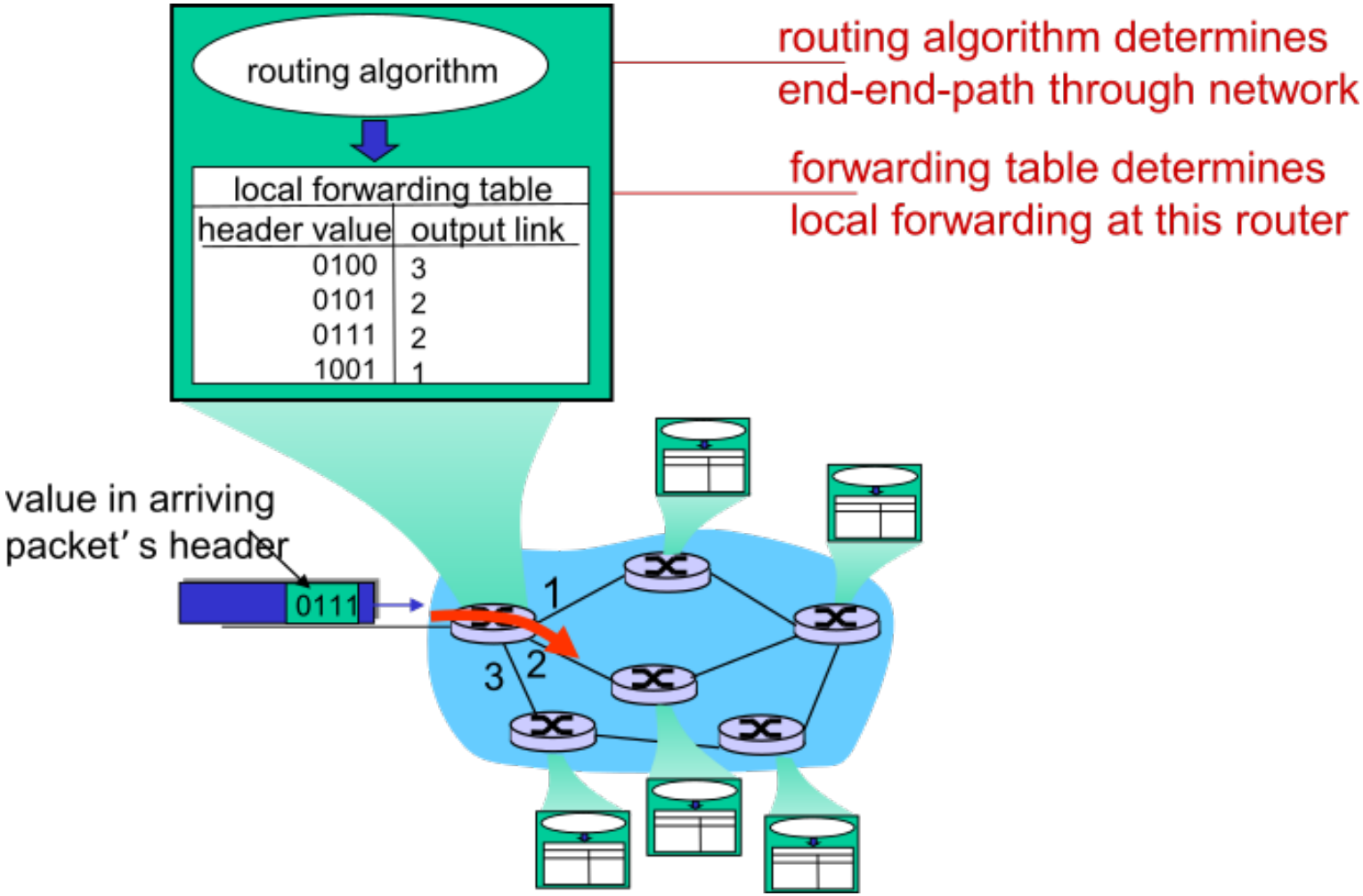
Routing

- Beste Route zwischen zwei Punkten bestimmen
- Regeln für Weiterleitung aufstellen
- Algorithmus, der Tabelle mit Regeln erstellt (Forwarding table)

Forwarding

- Eingehende Pakete analysieren
- An richtigen Ausgang weiterleiten
- Nutzt Forwarding table um richtigen Ausgang zu ermitteln

Zusammenspiel zwischen Routing und Forwarding



Die Forwarding Tabelle

Aufbau

Drei Informationen: Zielnetzwerk, Netzwerkinterface, Nächster Router

Zielnetzwerk	Interface	Router
192.168.178.0/24	Ethernet 1	0.0.0.0
195.37.0.0/16	Ethernet 1	192.168.178.1

Longest-Prefix-Matching

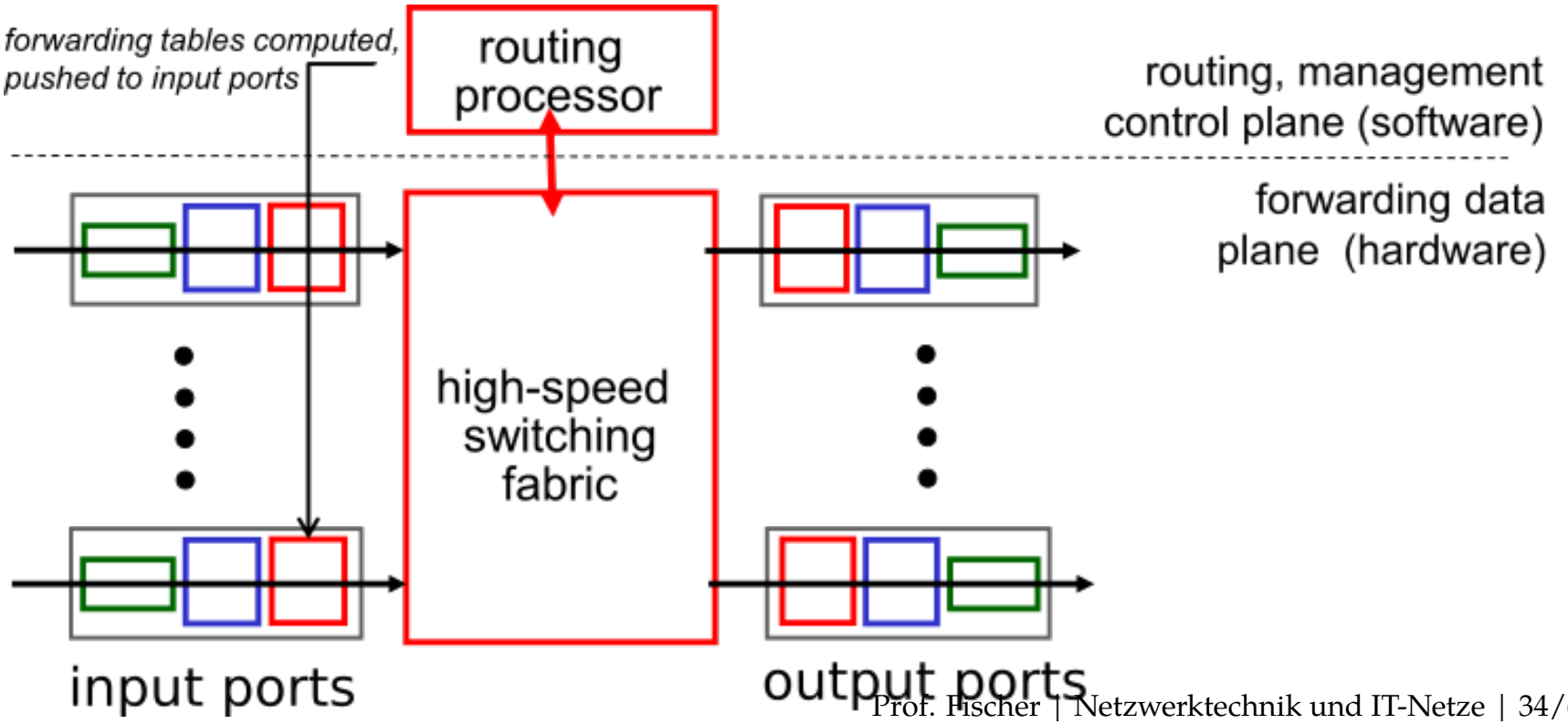
Die “genaueste” Route wird verwendet

Zielnetzwerk	Interface	Router
195.37.0.0/16	Ethernet 1	192.168.178.1
195.37.242.0/24	VPN 1	10.0.0.1

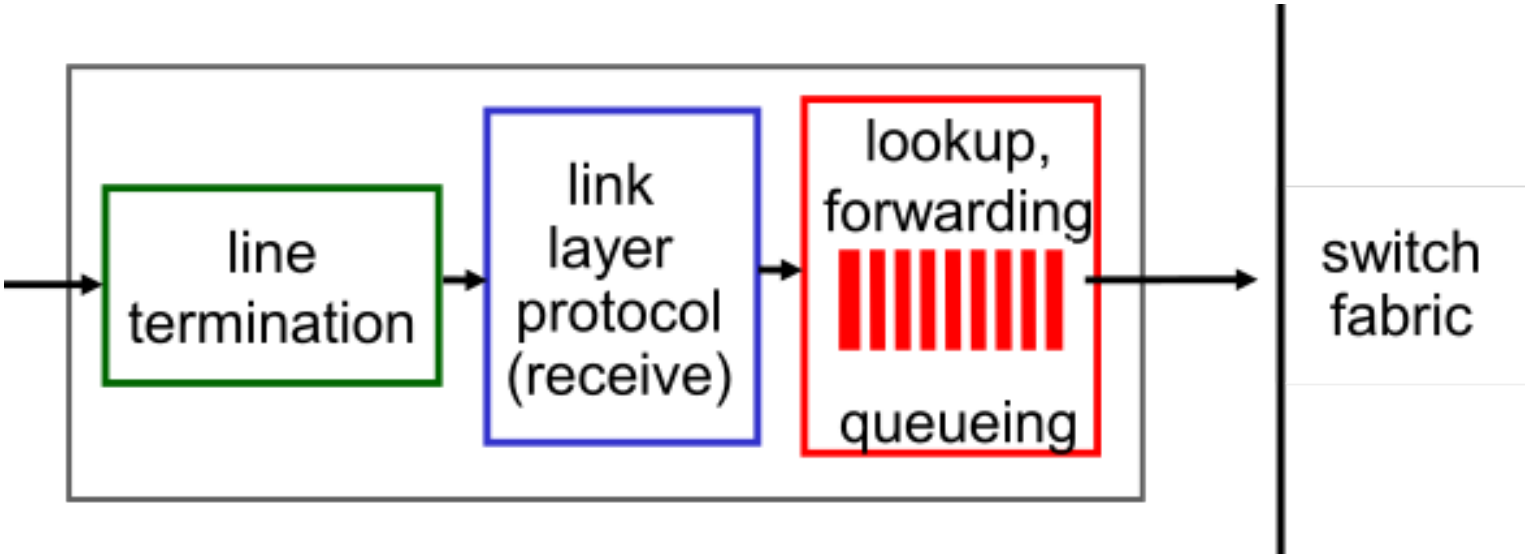
Die Architektur eines Routers

Trennung von Routing und Forwarding

- Control Plane: Pfadfindung mit Algorithmen
- Data Plane: Weiterleitung von Paketen



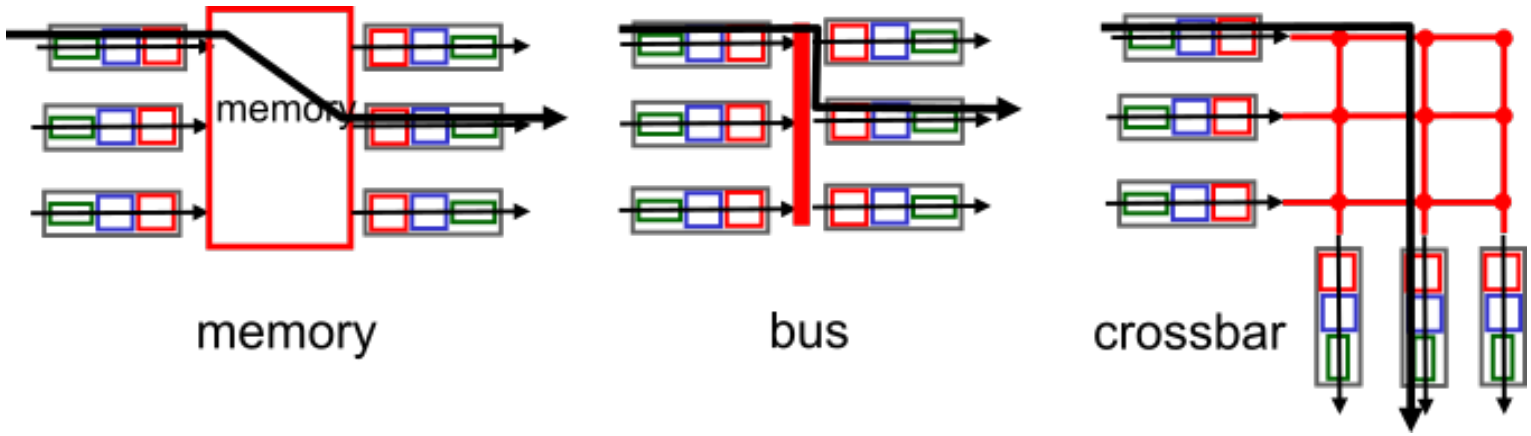
Input ports



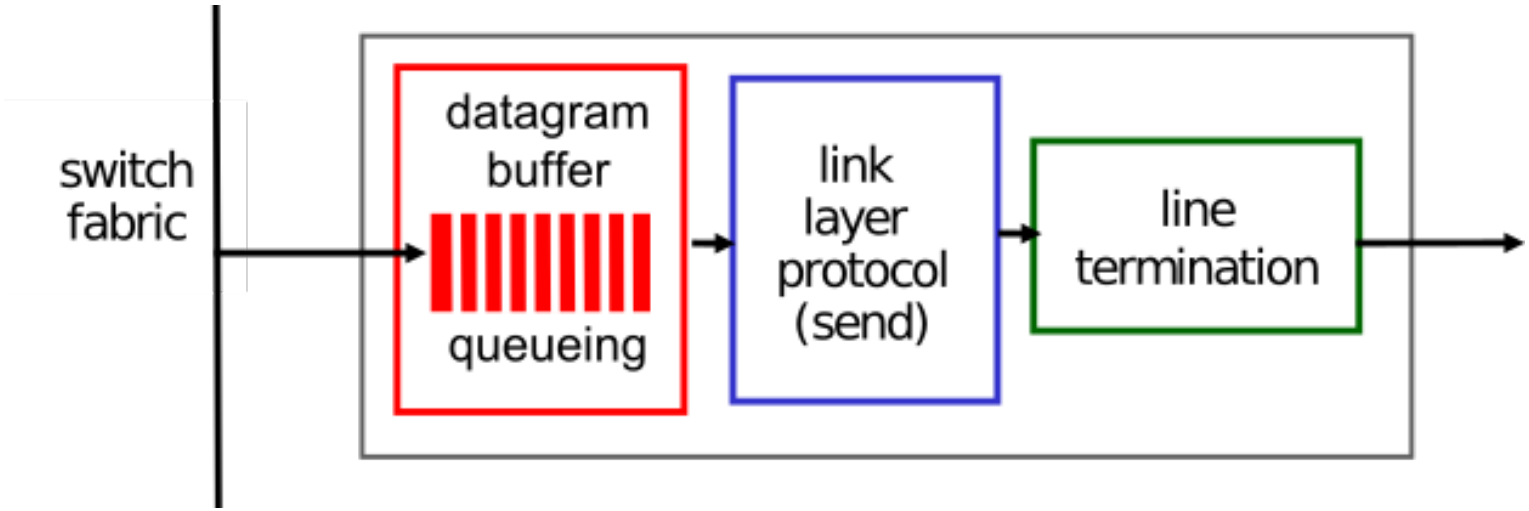
- Physische Schicht + Vermittlungsschicht greifen Paket an der Leitung ab
- Paket wird lokal gepuffert
- Entscheidung über Weiterleitung wird hier getroffen
  - Match: Welche Regel der Forwarding Tabelle trifft zu?
  - Action: Über die Switch Fabric an den richtigen Output Port

Switch fabrics

- Pakete müssen von Input zu Output Port weitergeleitet werden
- Transfer muss zügig erfolgen
  - Oft ein vielfaches der Datenrate einer Leitungen
  - Bei N Leitungen mit Rate R: $N \cdot R$ wünschenswert
- Switch fabric ist eigenes Netzwerk innerhalb eines Routers
- Drei Arten von Switch fabrics: Speicher; Bus; Crossbar



Output ports



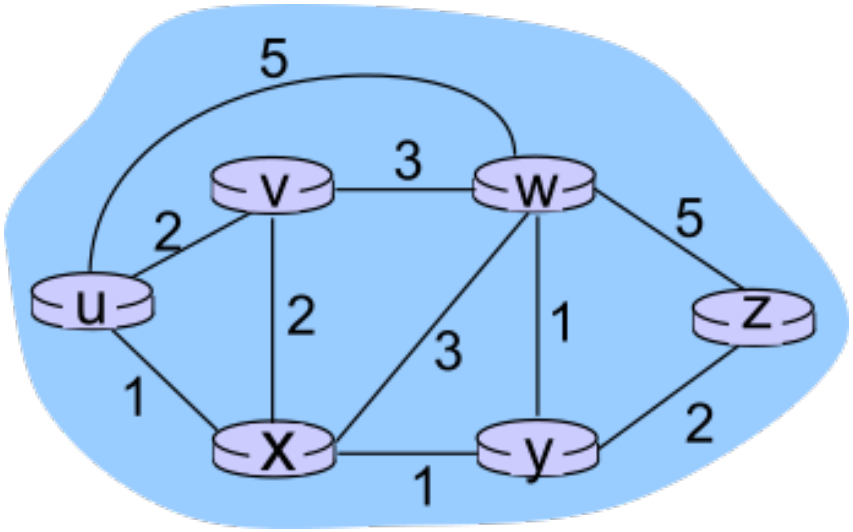
- Pakete müssen in Warteschlange gepuffert werden
  - Warteschlange wächst wenn Pakete schneller ankommen als sie gesendet werden
  - Hier können Pakete verloren gehen!
- Auswahlstrategie entscheidet, welches Paket als nächstes gesendet wird
  - FIFO: First in, first out
  - Priority-Queueing: Bevorzugte Warteschlange

Die Netzwerkschicht

Überblick

- Aufgaben der Netzwerkschicht
- Das Internet Protokoll
- IP Adressen und Adressvergabe
- Funktionsweise eines Routers
- Routing Protokolle
  - Link-State: OSPF
  - Distance Vector: RIP
  - Hierarchisch: BGP
- Flexibilisierung der Netzwerkschicht
  - Software-defined Networking
  - Network Function Virtualization

Das Netzwerk als Graph



- Gewichteter Graph $G = (V, E)$, bestehend aus
  - Knoten $v \in V$
  - Kanten $e \in E \subseteq V \times V$
  - Kantengewichtsfunktion $c : E \to \mathbb{R}$
- Kosten eines Pfades: Summe der Kantengewichte
- Was ist der “billigste” Pfad zwischen zwei Punkten?

Routing Protokolle

Frage: Wie wird die Forwarding Tabelle gefüllt?

Option 1: Vom Netzwerkadministrator per Hand
→ Kommando `route`

Option 2: Von einem Algorithmus

- Graph-Problem: Kürzeste Pfade bestimmen
- Verteilte Systeme: Information muss konsistent sein → Protokoll notwendig

Zwei Varianten (plus eine):

- Innerhalb des Provider-Netzwerks
  - Mit globaler Information: Link-State, OSPF
  - Mit lokaler Information: Distance-Vector, RIP
- Zwischen Providern: Hierarchisch, BGP

Die Netzwerkschicht

Überblick

- Aufgaben der Netzwerkschicht
- Das Internet Protokoll
- IP Adressen und Adressvergabe
- Funktionsweise eines Routers
- Routing Protokolle
  - Link-State: OSPF
  - Distance Vector: RIP
  - Hierarchisch: BGP
- Flexibilisierung der Netzwerkschicht
  - Software-defined Networking
  - Network Function Virtualization

Link-State Routing

Konzept:

- Netzwerktopologie ist allen Routern vollständig bekannt
- Jeder Router berechnet lokal kürzeste Wege
  - Typisches Graph-Problem
  - Algorithmus: z.B. Dijkstra, Floyd-Warshall, Bellman-Ford

Umsetzung: Open Shortest Path First (OSPF)

- Link-State Protokoll
- Jeder Router verwaltet eigene Kopie der Topologie
- Lokale Verbindungsinformationen werden an alle Router gesendet
- Kürzeste Wege werden mit Dijkstra berechnet

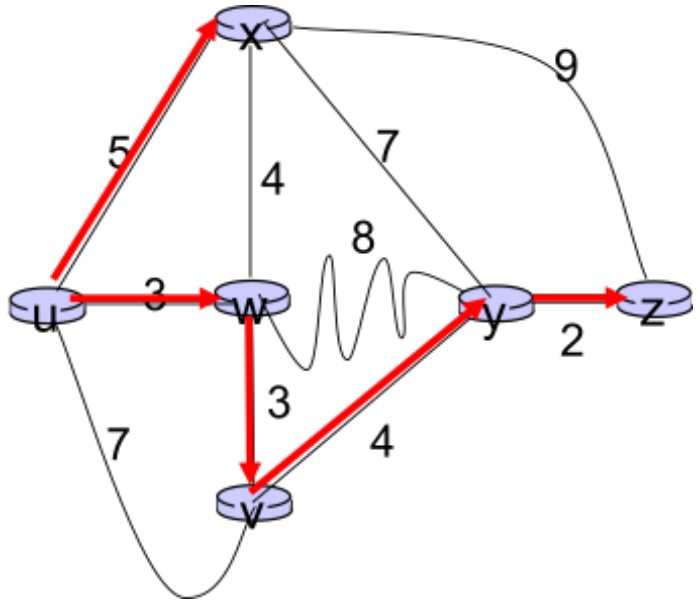
Der Dijkstra-Algorithmus

- Pfade mit geringsten Kosten zu allen anderen Knoten finden
- Benötigt zwei Funktionen
  - Kosten $c : V \times V \to \mathbb{R}$
  - Distanz $d : V \to \mathbb{R} \cup \infty$
- Iterativ: In jedem Schritt ein neuer, kürzester Weg bekannt

Data: Vertices V
Data: $V' = \{self\}$
for *each* $v \in V$ **do**
  if v *adajcent to* $self$ **then**
    $d(v) = c(u, v)$;
  else
    $d(v) = \infty$;
  end
end
while $|V'| < |V|$ **do**
  Find $w \notin V' : d(w)$ is minimal;
  $V' := V' \cup \{w\}$;
  for *each* $v : v$ *adjacent to* w; $v \notin V'$ **do**
    $d(v) = \min(d(v), d(w) + c(w, v))$;
  end
end

Der Dijkstra-Algorithmus: Ein Beispiel

Step	N'	D(v) p(v)	D(w) p(w)	D(x) p(x)	D(y) p(y)	D(z) p(z)
0	u	7,u	3,u	5,u	∞	∞
1	uw	6,w		5,u	11,w	∞
2	uwx	6,w			11,w	14,x
3	uwxv				10,v	14,x
4	uwxvy					12,y
5	uwxvyz					



- Kürzeste Wege werden rekonstruiert in dem man sich den Vorgänger merkt
- Falls Minimum in einem Schritt nicht eindeutig: Zufällige Wahl

Die Netzwerkschicht

Überblick

- Aufgaben der Netzwerkschicht
- Das Internet Protokoll
- IP Adressen und Adressvergabe
- Funktionsweise eines Routers
- Routing Protokolle
  - Link-State: OSPF
  - Distance Vector: RIP
  - Hierarchisch: BGP
- Flexibilisierung der Netzwerkschicht
  - Software-defined Networking
  - Network Function Virtualization

Dynamische Programmierung

Bellmans Optimalitätsprinzip:

Wenn eine Lösung insgesamt optimal ist, dann sind auch die Teillösungen aus denen sie zusammengesetzt ist optimal

- Sei $d_x(y)$ der Pfad von x nach y mit den geringsten Kosten
- Sei Z die Menge der zu x direkt adjazenten Knoten
- Dann gilt: $d_x(y) = \min_{z \in Z}\{c(x,z) + d_z(y)\}$

Prinzip der dynamischen Programmierung

- Konstruiere optimale Teillösungen
- Speichere die Teillösungen in einer Tabelle
- Die optimale Gesamtlösung ergibt sich schließlich aus den Teillösungen

Distance-Vector Algorithmus

Voraussetzungen:

Jeder Knoten x...

- ...kennt seine Nachbarn Z, sowie die jeweiligen Kosten $c(x, z); z \in Z$
- ...verwaltet einen Vektor der geschätzten Distanzen zu allen anderen Knoten: $D_x = [D_x(y)]_{y \in V}$
- ...speichert Distanzvektoren seiner Nachbarn $D_z; z \in Z$

Vorgehensweise:

Jeder Knoten x...

- ...sendet regelmäßig seinen Distanzvektor D_x an alle Nachbarn
- ...aktualisiert bei Erhalt eines neuen Distanzvektors seine eigene Abschätzung

Distance-Vector Szenario

node x table

cost to	x	y	z
from x	0	2	7
from y	∞	∞	∞
from z	∞	∞	∞

cost to	x	y	z
from x	0	2	3
from y	2	0	1
from z	7	1	0

$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$
$= \min\{2+0 , 7+1\} = 2$

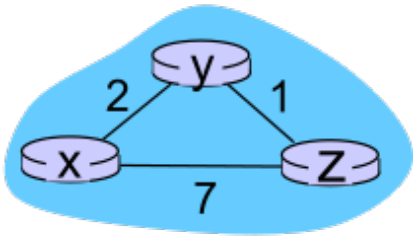
node y table

cost to	x	y	z
from x	∞	∞	∞
from y	2	0	1
from z	∞	∞	∞

node z table

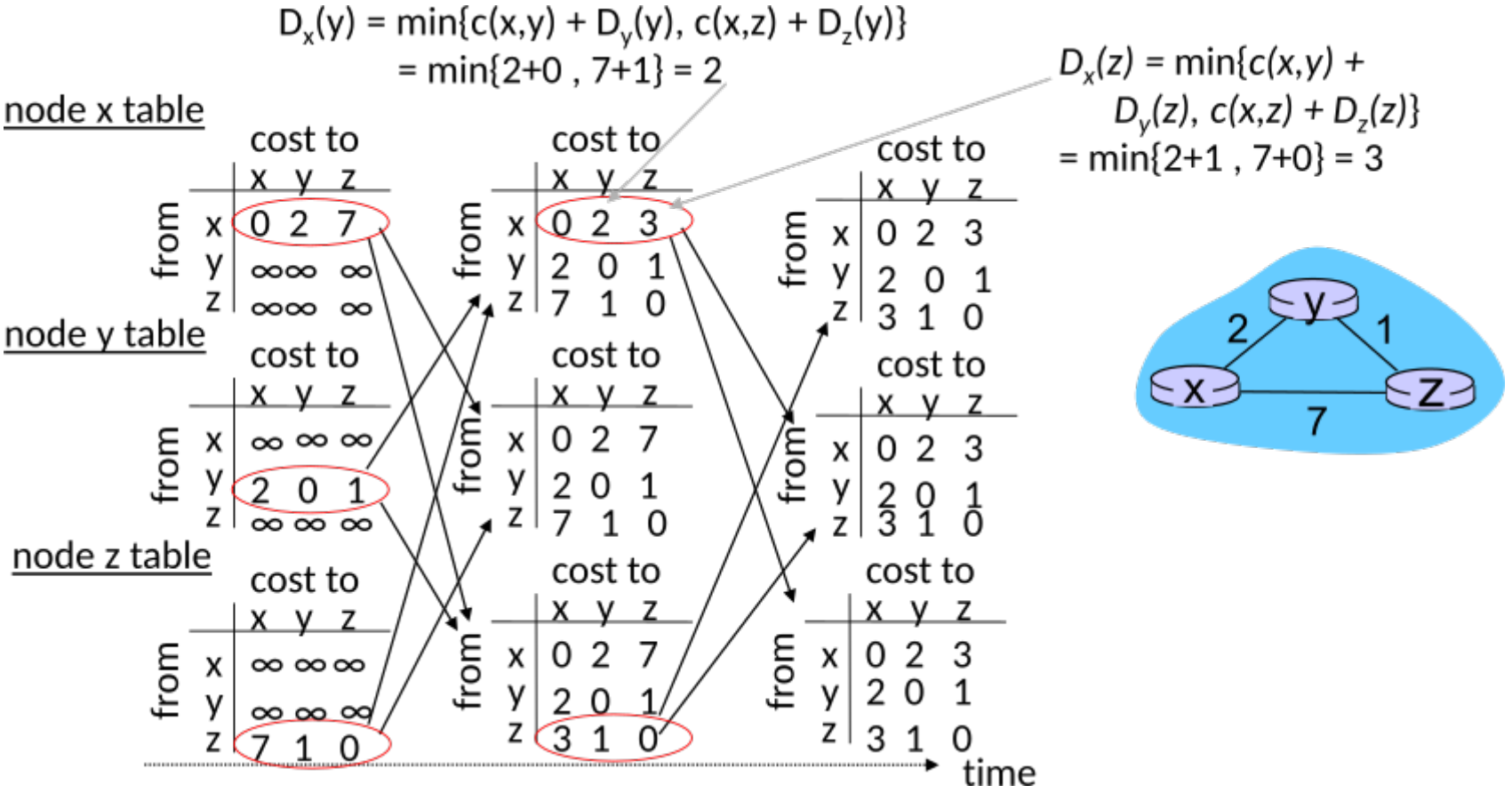
cost to	x	y	z
from x	∞	∞	∞
from y	∞	∞	∞
from z	7	1	0

$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$
$= \min\{2+1 , 7+0\} = 3$



time

Distance-Vector Szenario



Adaption an Kostenänderungen

Allgemein:

- Nachbarknoten stellen höhere Kosten fest
- Geänderte Distanzvektoren werden durchs Netzwerk verschickt

Kosten sinken

- Aktualisierung propagiert schnell durchs Netz: “Good news travels fast”
- Bereits informierte Knoten werden nicht nocheinmal aktualisiert

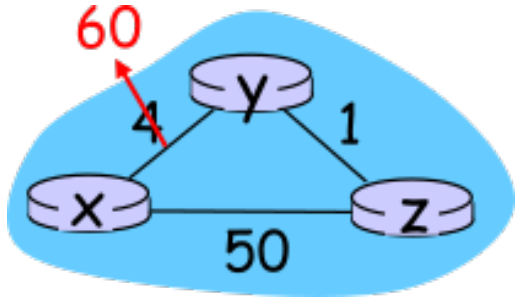
Kosten steigen

- Knoten aktualisieren sich wechselseitig: “Count-to-infinity” Problem
- Stabilisierung dauert lange

Count-to-Infinity und Poisoned Reverse

Das Count-to-Infinity Problem:

- Kosten zwischen x und y steigen deutlich
- y und z verweisen wechselseitig aufeinander
- Kosten werden hochgezählt bis Pfad $y \to z \to x$ erkannt wird



Poisoned Reverse:

- Wenn der Pfad von y nach x durch z führt:
  - y sendet modifizierten Distanzvektor D'_y an z
  - In diesem ist $D'_y(x) = \infty$
  - Dann löscht z auch seine Route über y
- Löst leider das Problem nicht ganz $\to$ Warum?

Das Routing Information Protocol

Implementierung eines Distance-Vector Protokolls:

- Entwickelt 1982
- Distanzmetrik: Anzahl an Hops
- Nachrichtenaustausch alle 30 Sekunden
- Distanzvektoren enthalten bis zu 25 Subnetze (keine Knoten)

Fehlerbehandlung in RIP:

- Falls kein Update nach 180 Sekunden: Link ist tot
- Poisoned Reverse vermeidet Ping-Pong Routen
- Kleine maximale Distanz: 16 Hops ($\rightarrow$ Small-World Argument!)

Die Netzwerkschicht

Überblick

- Aufgaben der Netzwerkschicht
- Das Internet Protokoll
- IP Adressen und Adressvergabe
- Funktionsweise eines Routers
- Routing Protokolle
  - Link-State: OSPF
  - Distance Vector: RIP
  - Hierarchisch: BGP
- Flexibilisierung der Netzwerkschicht
  - Software-defined Networking
  - Network Function Virtualization

Hierarchisches Routing

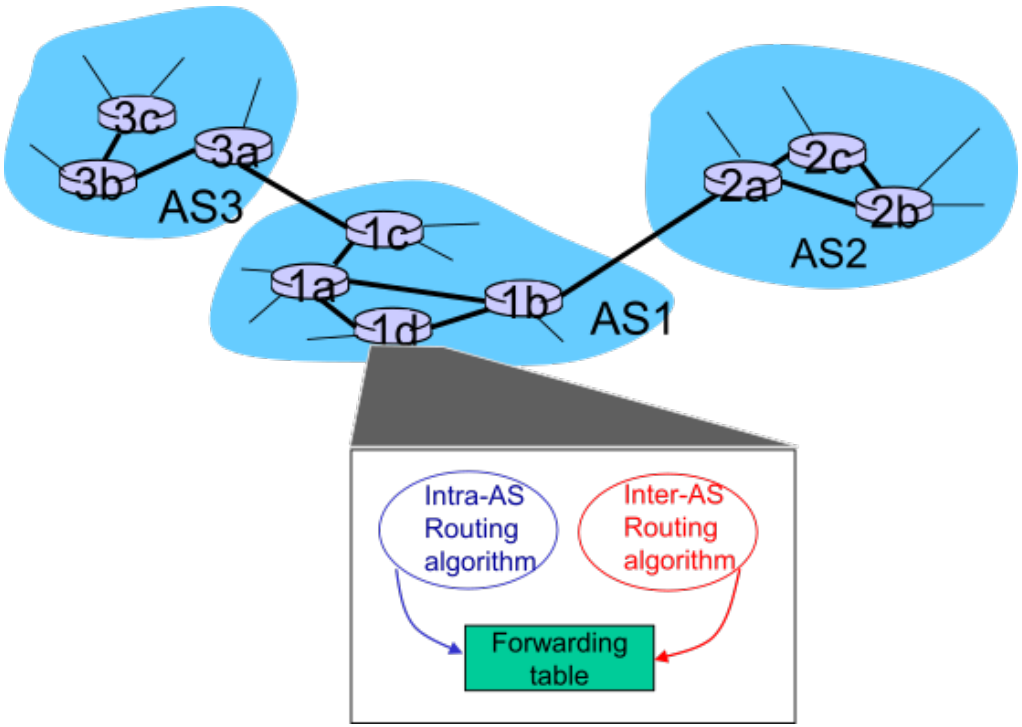
Bisher: Naïve Annahmen

- Lauter identische Router
- Eine Organisation – ein Netzwerk

Tatsächliche Situation:

- Internet ist Netzwerk von Netzwerken
- Vielzahl Autonomer Systeme (AS)
- Provider haben eigene Bedürfnisse:
  - Technisch: Eigene Hardware, eigene Protokolle
  - Politisch: Vereinbarungen mit anderen Providern

Vermaschte Autonome Systeme



- Forwarding-Tabelle von zwei Routingalgorithmen befüllt
  - Inter-AS Routing
  - Intra-AS Routing
- AS durch Gateway Router miteinander verbunden
- Intra-AS Routing Protokoll kann individuell sein
- Inter-AS Routing: BGP

Das Border Gateway Protocol (BGP)

- Standardisiert 1989 in RFC 1105
- De-facto Standard für Inter-AS Routing
- "Hält das Internet zusammen"
- Zwei Teile:

  eBGP: Externe Kommunikation mit anderen AS: Erreichbarkeit fremder Subnetze

  iBGP: Propagierung der erhaltenen Information an interne Router
- Ermöglicht einem ISP (und seinen Kunden) die Teilnahme am weltweiten Internet

Die Netzwerkschicht

Überblick

- Aufgaben der Netzwerkschicht
- Das Internet Protokoll
- IP Adressen und Adressvergabe
- Funktionsweise eines Routers
- Routing Protokolle
  - Link-State: OSPF
  - Distance Vector: RIP
  - Hierarchisch: BGP
- Flexibilisierung der Netzwerkschicht
  - Software-defined Networking
  - Network Function Virtualization

Ansätze zur Flexibilisierung der Netzwerkschicht

Problem:

- Verknöcherung des Internets
- IP ist so zentral, dass es nicht einfach auszutauschen ist

Lösungsansätze:

- Programmierbare Netzwerke – ein verteiltes Betriebssystem für das Netzwerk
- Netzwerkoverlays: Peer-to-peer Netze, MPLS
- Aktuell: Software-defined Networking & Network Function Virtualization

Die Netzwerkschicht

Überblick

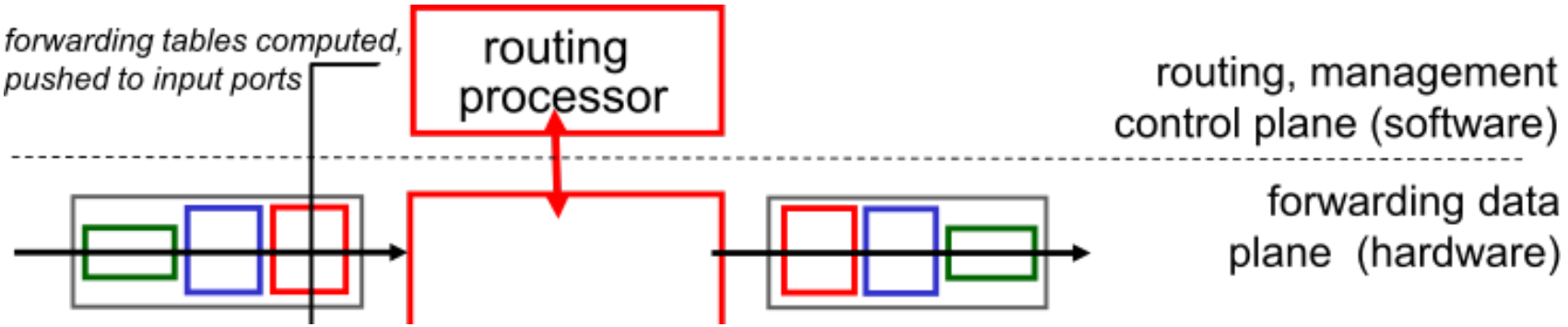
- Aufgaben der Netzwerkschicht
- Das Internet Protokoll
- IP Adressen und Adressvergabe
- Funktionsweise eines Routers
- Routing Protokolle
  - Link-State: OSPF
  - Distance Vector: RIP
  - Hierarchisch: BGP
- Flexibilisierung der Netzwerkschicht
  - Software-defined Networking
  - Network Function Virtualization

Software-defined Networking

Trennung von Control Plane und Data Plane

Control Plane: Routing, Erstellen der Forwarding Tabelle

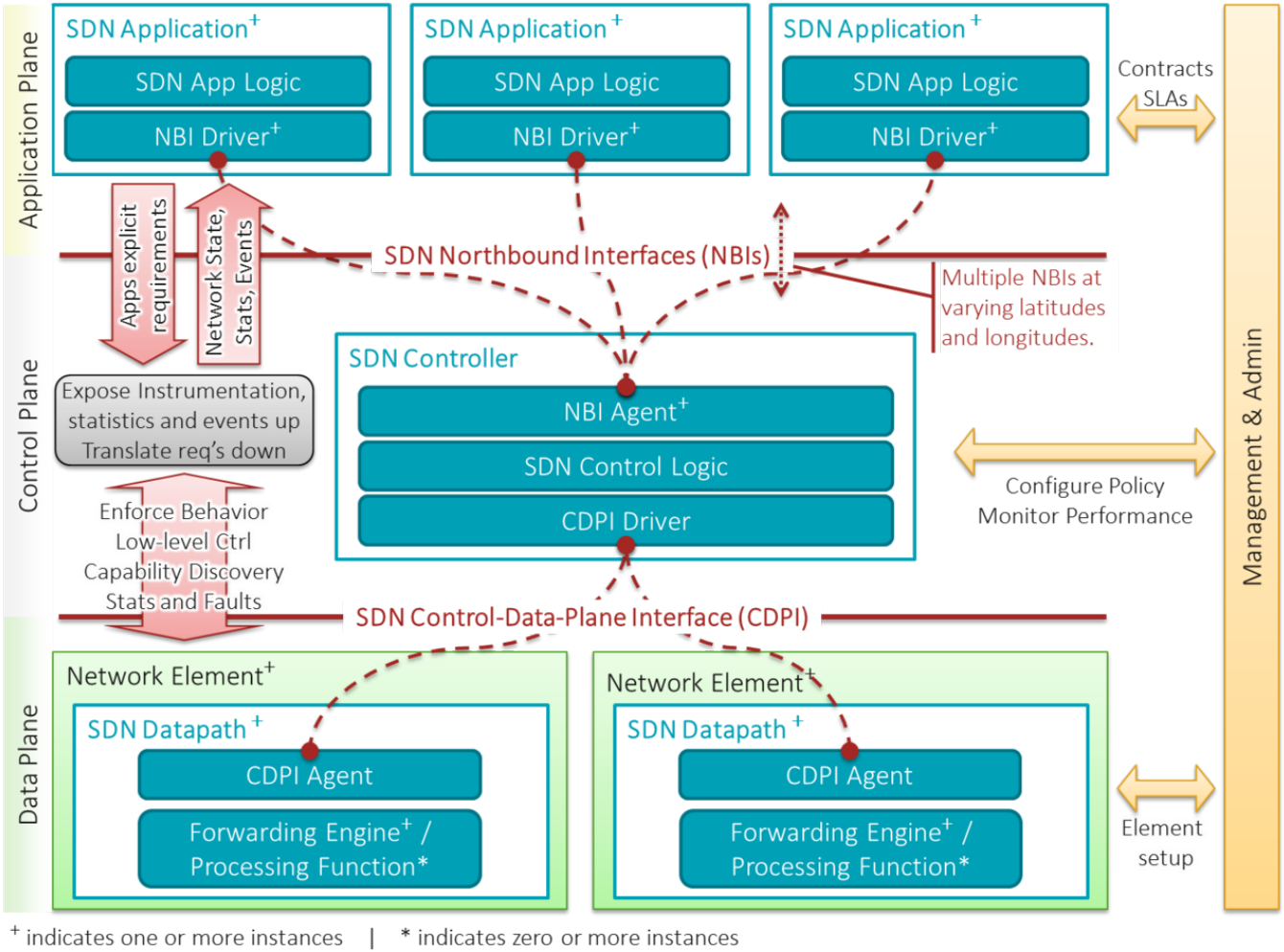
Data Plane: Forwarding, hochspezialisierte Hardware



Ziele:

- Netzwerkmanagement automatisieren
- Programmierbare Netzwerkfunktionalität
- Verteilte Funktionalität, aber zentral gewartet

Architektur eines SDN Systems



By Open Networking Foundation (ONF) - SDN Architecture Overview (PDF), Version 1.0, December 12, 2013.
CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=36034296

SDN Komponenten

Control Plane: SDN Controller

- Steuert zentral das Routing
- Erstellt dynamisch neue Forwardingregeln und verteilt diese an alle betroffenen Netzwerkelemente

Data Plane: SDN Datapath

- Teil eines SDN-fähigen Netzwerkgeräts
- Realisiert das Forwarding von Paketen

Flexibilisierung: SDN Applications

- Geben vor, wie neue Forwardingregeln erstellt werden
- Ermöglichen es, “das Netzwerk zu programmieren”

Die beiden zentralen SDN Schnittstellen

Northbound Interface

- Ermöglicht es, neue Funktionalität über SDN Applications flexibel zu realisieren
- Quasi ein “Plug-in” Mechanismus für das zentrale Netzwerk

Southbound/Control-Data Plane Interface (CDPI)

- Interface über das der Controller den Netzwerkelementen neue Forwardingregeln mitteilt
- Netzwerkelemente leiten unbekannte Pakete an den Controller zur weiteren Verarbeitung

Das OpenFlow Protokoll

- Realisiert das Control-Data Plane Interface (CDPI)
  - Zugriff auf die Data Plane eines Geräts über das Netzwerk
  - Kommunikation über TCP/TLS
- Unterscheidet nicht zwischen Switch und Router
  - Verwaltet Netzwerkschnittstellen (Ports) auf jedem Gerät
  - Spezifiziert Forwarding-Regeln

Forwarding bisher:

Match:	Zielnetz in CIDR Notation mit Forwardingtabelle abgleichen
Action:	Weiterleitung an angegebener Netzwerkschnittstelle

OpenFlow Forwarding:

Match:	Mit beliebigen Feldern in Paketheadern (L2-L4) abgleichen
Action:	Weiterleiten oder Weiterverarbeiten oder an Controller

Die Netzwerkschicht

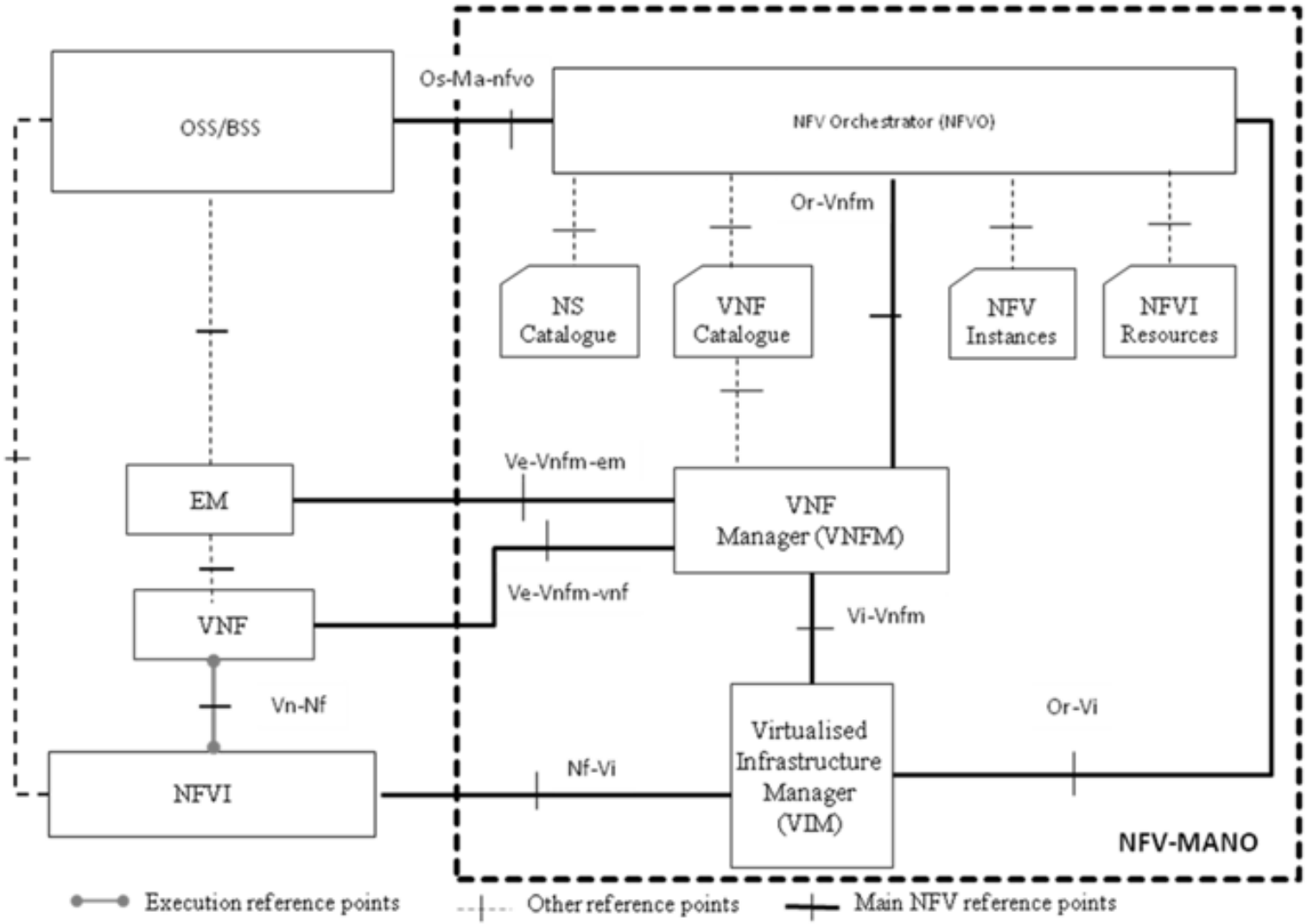
Überblick

- Aufgaben der Netzwerkschicht
- Das Internet Protokoll
- IP Adressen und Adressvergabe
- Funktionsweise eines Routers
- Routing Protokolle
  - Link-State: OSPF
  - Distance Vector: RIP
  - Hierarchisch: BGP
- Flexibilisierung der Netzwerkschicht
  - Software-defined Networking
  - Network Function Virtualization

Network Function Virtualization

- Initiative von Telekommunikationsunternehmen
- Virtualisierung als Einsparmöglichkeit
  - Mehrere Netzwerkfunktionen auf einem physischen Gerät
  - Ermöglicht Skalierbarkeit
  - Ermöglicht Energieeffizienz
- Beispiele für Netzwerkfunktionen:
  - Sicherheit: Firewall, Deep Packet Inspection, Intrusion Detection/Prevention
  - Datenverkehrsabrechnung (z.B. Mobilfunk)
  - Caching, Datenkomprimierung
- Management und Orchestrierung der Services notwendig
  - MANO: Management and Orchestration
  - Analog zu Cloud-Computing

Die NFV MANO Architektur



Service Function Chains

Beobachtung:

- Netzwerkfunktionen stehen nicht isoliert da
- Komplexe Dienste setzen sich aus atomaren Diensten zusammen
- Dienste müssen entsprechend verknüpft werden

Beispiel:

- Funktionalität: Verschlüsselung, Komprimierung, Virenscan
- Netzwerkfunktionen sind frei im Netzwerk platzierbar (unter Beachtung der verfügbaren Ressourcen)
- Aber: Reihenfolge spielt eine wichtige Rolle! → Warum?

Zusammenfassung und Ausblick

Netzwerkschicht kennengelernt

- Aufgaben der Netzwerkschicht: Routing, Forwarding
- Funktionsweise des IP Protokolls
- Funktionsweise von Routingprotokollen

Was bleibt?

- "Reinschnuppern" in die Vermittlungsschicht
- Übergreifendes Thema: Security