

Systeme II

7. Die Vermittlungsschicht

Christian Schindelhauer

Technische Fakultät

Rechnernetze und Telematik

Albert-Ludwigs-Universität Freiburg

Version 13.06.2012

- Circuit Switching
 - Etablierung einer Verbindung zwischen lokalen Benutzern durch Schaltstellen
 - mit expliziter Zuordnung von realen Schaltkreisen
 - oder expliziter Zuordnung von virtuellen Ressourcen, z.B. Slots
 - Quality of Service einfach, außer bei
 - Leitungsaufbau
 - Leitungsdauer
 - Problem
 - Statische Zuordnung
 - Ineffiziente Ausnutzung des Kommunikationsmedium bei dynamischer Last
 - Anwendung
 - Telefon
 - Telegraf
 - Funkverbindung

■ Packet Switching

- Grundprinzip von IP
 - Daten werden in Pakete aufgeteilt und mit Absender/Ziel-Information unabhängig versandt
- Problem: Quality of Service
 - Die Qualität der Verbindung hängt von einzelnen Paketen ab
 - Entweder Zwischenspeichern oder Paketverlust
- Vorteil:
 - Effiziente Ausnutzung des Mediums bei dynamischer Last

■ Resümee

- Packet Switching hat Circuit Switching in praktisch allen Anwendungen abgelöst
- Grund:
 - Effiziente Ausnutzung des Mediums

■ Transport

- muss gewisse Flusskontrolle gewährleisten
- z.B. Fairness zwischen gleichzeitigen Datenströmen

■ Vermittlung

- Quality of Service (virtuelles Circuit Switching)

■ Sicherung

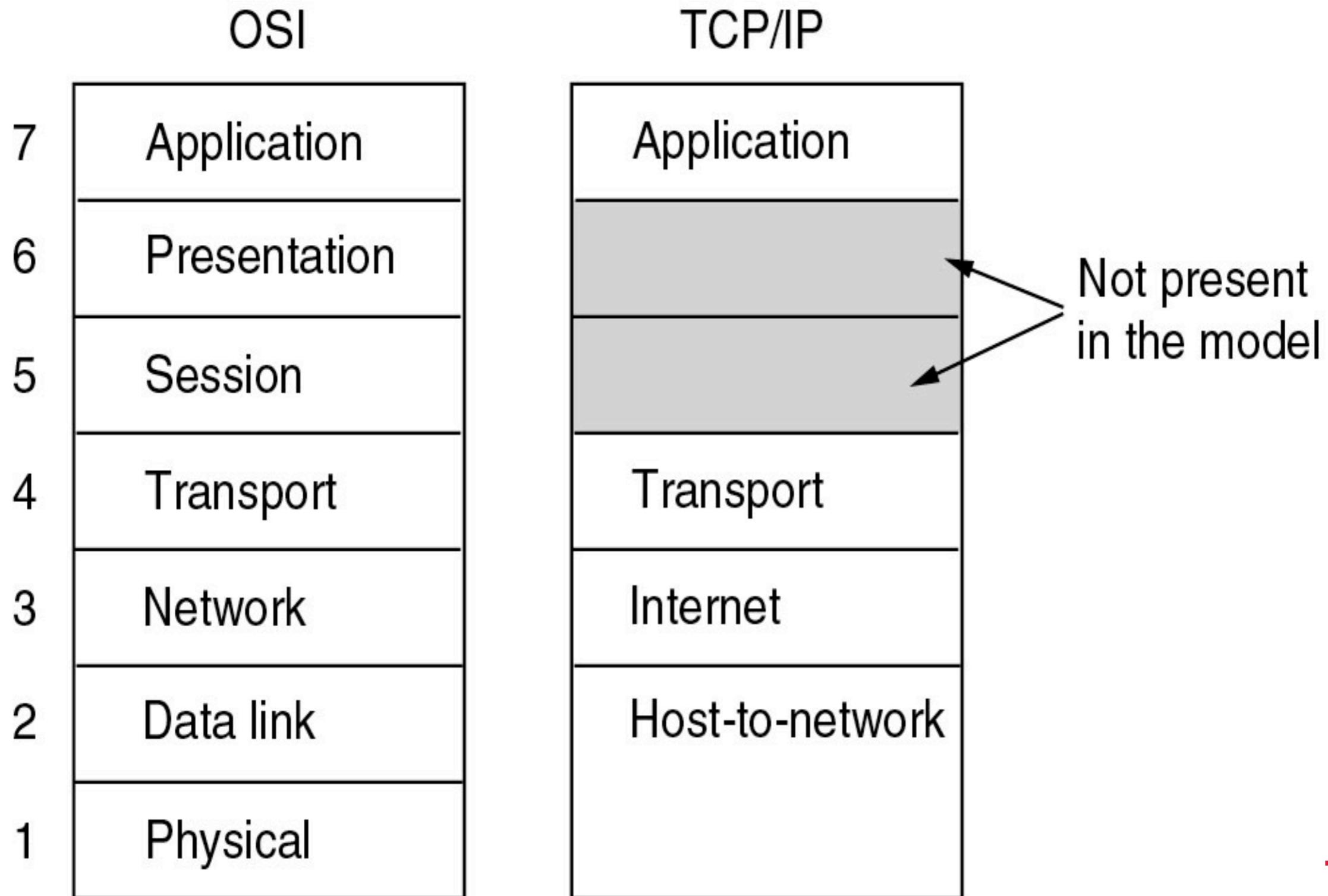
- Flusskontrolle zur Auslastung des Kanals

Layer	Policies
Transport	<ul style="list-style-type: none"> • Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy • Timeout determination
Network	<ul style="list-style-type: none"> • Virtual circuits versus datagram inside the subnet • Packet queueing and service policy • Packet discard policy • Routing algorithm • Packet lifetime management
Data link	<ul style="list-style-type: none"> • Retransmission policy • Out-of-order caching policy • Acknowledgement policy • Flow control policy

Die Schichtung des Internets - TCP/IP-Layer

Anwendung	Application	Telnet, FTP, HTTP, SMTP (E-Mail), ...
Transport	Transport	TCP (Transmission Control Protocol) UDP (User Datagram Protocol)
Vermittlung	Network	IP (Internet Protocol) + ICMP (Internet Control Message Protocol) + IGMP (Internet Group Management Protocol)
Verbindung	Host-to-network	LAN (z.B. Ethernet, Token Ring etc.)

OSI versus TCP/IP



- Lokale Netzwerke können nicht nur über Hubs, Switches oder Bridges verknüpft werden
 - Hubs: Kollisionen nehmen überhand
 - Switches:
 - Routen-Information durch Beobachtung der Daten ineffizient
 - Broadcast aller Nachrichten schafft Probleme
 - Es gibt über 100 Mio. lokale Netzwerke im Internet...
- Zur Beförderung von Paketen in großen Netzwerken braucht man Routeninformationen
 - Wie baut man diese auf?
 - Wie leitet man Pakete weiter?
- Das Internet-Protokoll ist im wesentlichen ein Vermittlungsschichtprotokoll

- IP-Routing-Tabelle
 - enthält für Ziel (Destination) die Adresse des nächsten Rechners (Gateway)
 - Destination kann einen Rechner oder ganze Sub-nets beschreiben
 - Zusätzlich wird ein Default-Gateway angegeben
- Packet Forwarding
 - früher Packet Routing genannt
 - IP-Paket (datagram) enthält Start-IP-Adresse und Ziel-IP-Adresse
 - Ist Ziel-IP-Adresse = eigene Rechneradresse dann Nachricht ausgeliefert
 - Ist Ziel-IP-Adresse in Routing-Tabelle dann leite Paket zum angegebenen Gateway
 - Ist Ziel-IP-Subnetz in Routing-Tabelle dann leite Paket zum angegebenen Gateway
 - Ansonsten leite zum Default-Gateway

- IP-Paket (datagram) enthält unter anderen
 - TTL (Time-to-Live): Anzahl der Hops
 - Start-IP-Adresse
 - Ziel-IP-Adresse
- Behandlung eines Pakets
 - Verringere TTL (Time to Live) um 1
 - Falls $TTL \neq 0$ dann Packet-Forwarding aufgrund der Routing-Tabelle
 - Falls $TTL = 0$ oder bei Problemen in Packet-Forwarding:
 - Lösche Paket
 - Falls Paket ist kein ICMP-Paket dann
 - Sende ICMP-Paket mit
 - Start= aktuelle IP-Adresse und
 - Ziel = alte Start-IP-Adresse

- Forwarding:
 - Weiterleiten von Paketen
- Routing:
 - Erstellen Routen, d.h.
 - Erstellen der Routing-Tabelle
- Statisches Routing
 - Tabelle wird manuell erstellt
 - sinnvoll für kleine und stabile LANs
- Dynamisches Routing
 - Tabellen werden durch Routing-Algorithmus erstellt
 - Zentraler Algorithmus, z.B. Link State
 - Einer/jeder kennt alle Information, muss diese erfahren
 - Dezentraler Algorithmus, z.B. Distance Vector
 - arbeitet lokal in jedem Router
 - verbreitet lokale Information im Netzwerk

- Gegeben:
 - Ein gerichteter Graph $G=(V,E)$
 - Startknoten
 - mit Kantengewichtungen $w : E \rightarrow \mathbb{R}$
- Definiere Gewicht des kürzesten Pfades
 - $\delta(u,v)$ = minimales Gewicht $w(p)$ eines Pfades p von u nach v
 - $w(p)$ = Summe aller Kantengewichte $w(e)$ der Kanten e des Pfades
- Gesucht:
 - Die kürzesten Wege vom Startknoten s zu allen Knoten in G
 - also jeweils ein Pfad mit dem geringsten Gewicht zu jedem anderen Knoten
- Lösungsmenge:
 - wird beschrieben durch einen Baum mit Wurzel s
 - Jeder Knoten zeigt in Richtung der Wurzel

- Dijkstras Kürzeste-Wege-Algorithmus kann mit Laufzeit $\Theta(|E| + |V| \log |V|)$ implementiert werden.

```

Init-Source( $G, w, s$ )
begin
  for all  $v \in V$  do
     $d(v) \leftarrow \infty$ 
     $\pi(v) \leftarrow v$ 
  od
   $d(s) \leftarrow 0$ 
end
  
```

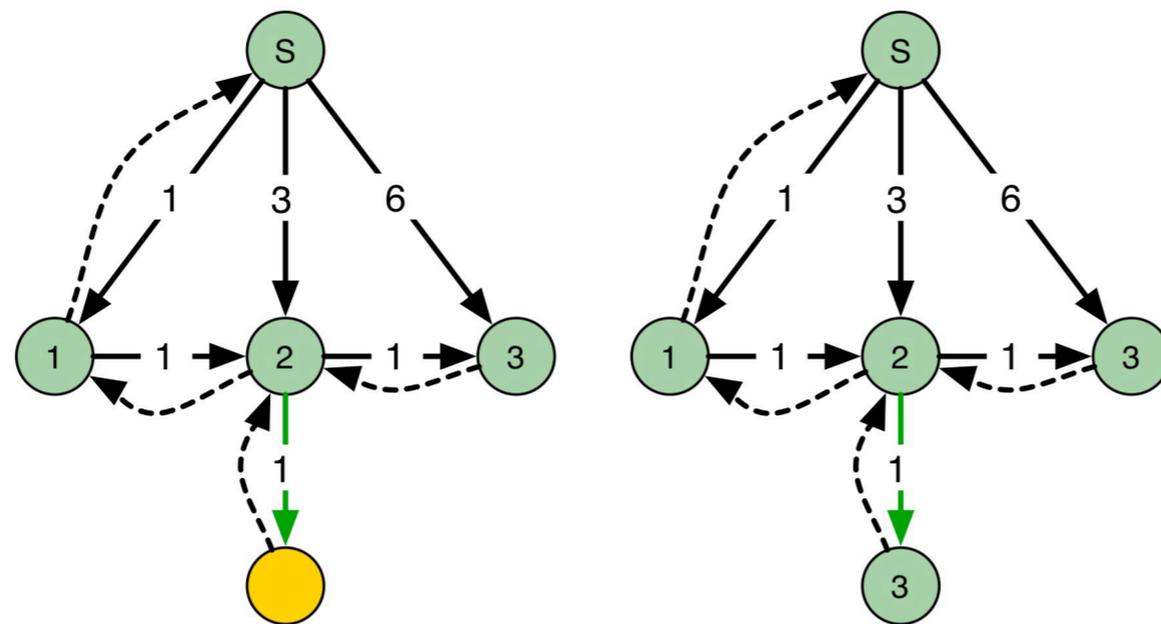
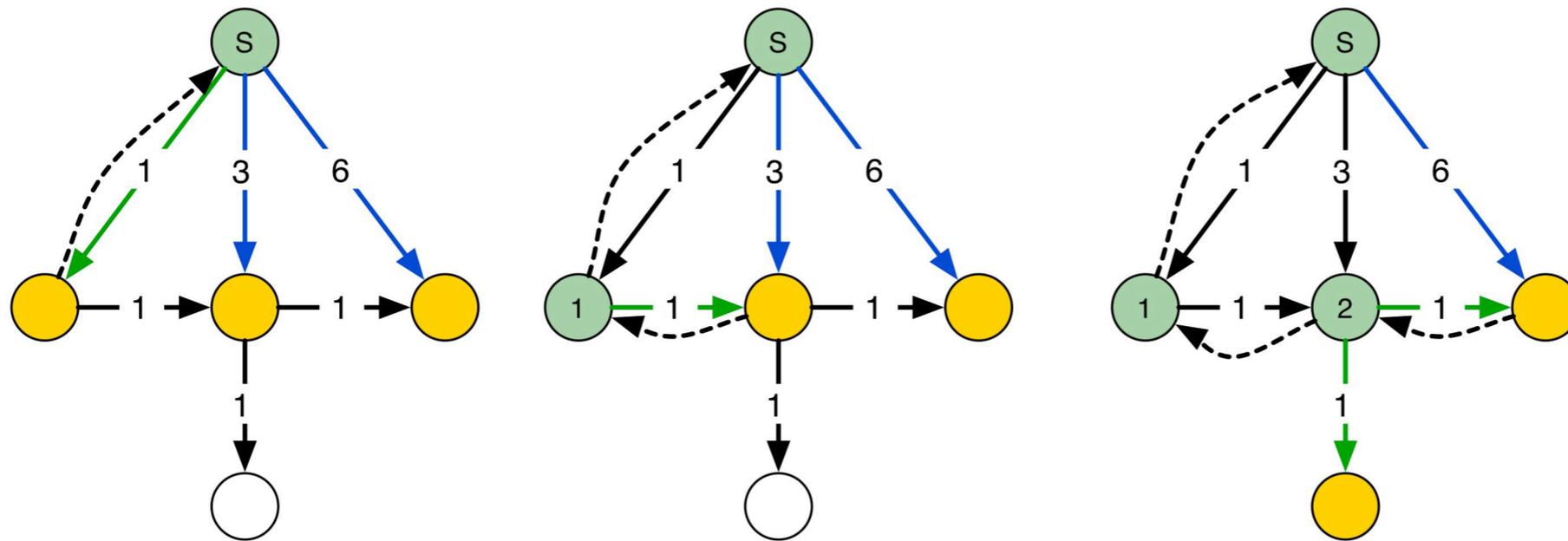
```

Dijkstra( $G, w, s$ )
begin
  Init-Source( $G, w$ )
   $S \leftarrow \emptyset$ 
   $Q \leftarrow V$ 
  while  $Q \neq \emptyset$  do
     $u \leftarrow$  Element aus  $Q$  mit minimalen Wert  $d(u)$ 
     $S \leftarrow S \cup \{u\}$ 
     $Q \leftarrow Q \setminus \{u\}$ 
    for all  $v \in \text{Adj}(u)$  do
      Relax( $u, v$ )
    od
  od
end
  
```

```

Relax( $u, v$ )
begin
  if  $d(v) > d(u) + w(u, v)$  then
     $d(v) \leftarrow d(u) + w(u, v)$ 
     $\pi(v) \leftarrow u$ 
  fi
end
  
```

Dijkstra: Beispiel



- Bei negativen Kantengewichten versagt Dijkstras Algorithmus
- Bellman-Ford
 - löst dies in Laufzeit $O(|V| |E|)$.

Bellman-Ford(G, w, s)

- **Init-Target**(G, w)
- loop $|V| - 1$ times:
 - for all $(u, v) \in E$ do
 - **Relax**(u, v)
- for all $(u, v) \in E$ do
 - if $d(u) > d(v) + w(u, v)$ then return false

Init-Target(G, w, t)

- **Init-Target**(G, w)
- for all $v \in V$ do
 - $d(v) \leftarrow \infty$
 - $\pi(v) \leftarrow v$
- $d(t) \leftarrow 0$

Relax(u, v)

- **Relax**(u, v)
- if $d(u) > w(u, v) + d(v)$ then
 - $d(u) \leftarrow w(u, v) + d(v)$
 - $\pi(u) \leftarrow v$

Distance Table Datenstruktur

- Jeder Knoten besitzt eine
 - Zeile für jedes mögliches Ziel
 - Spalte für jeden direkten Nachbarn

Verteilter Algorithmus

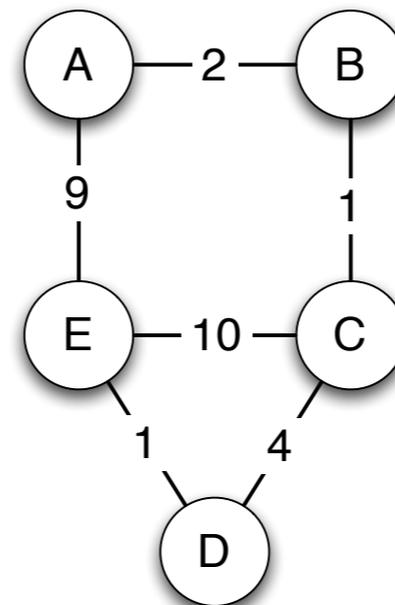
- Jeder Knoten kommuniziert nur mit seinem Nachbarn

Asynchroner Betrieb

- Knoten müssen nicht Informationen austauschen in einer Runde

Selbst Terminierend

- läuft bis die Knoten keine Informationen mehr austauschen



Distance Table für A

von A	über		Routing Tabellen Eintrag
	B	E	
nach B	2	11	B
C	3	19	B
D	7	10	B
E	8	9	B B

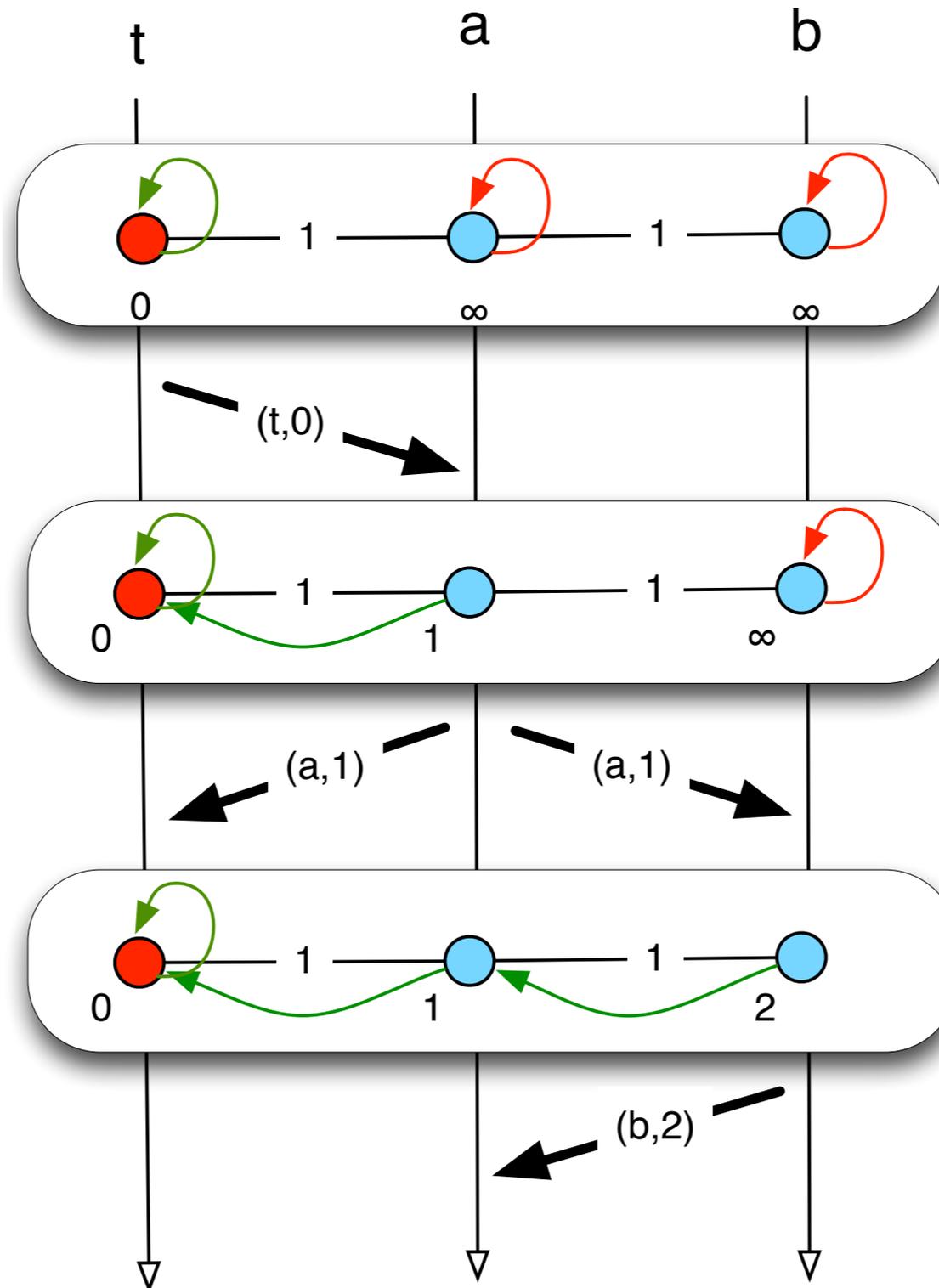
Distance Table für C

von C	über			Routing Tabellen Eintrag
	B	D	E	
nach A	3	14	18	B
B	1	9	9	B
D	6	4	11	D
E	7	5	10	D

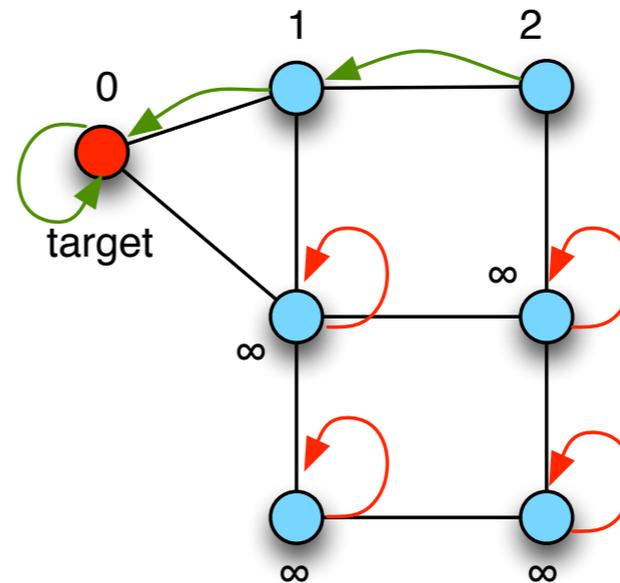
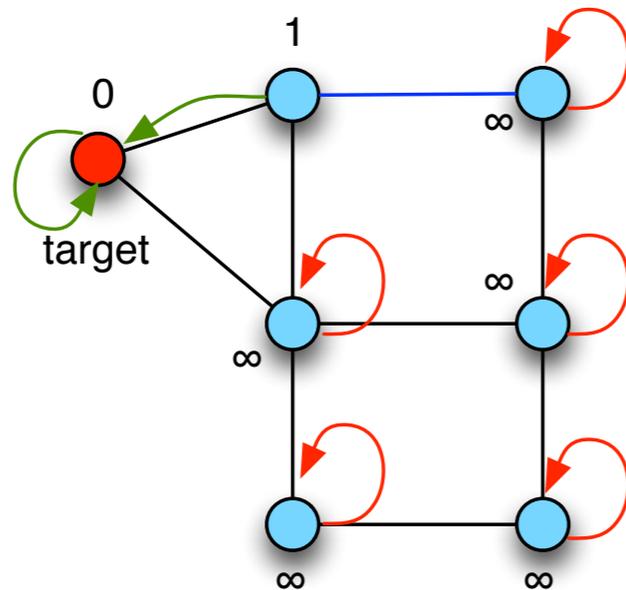
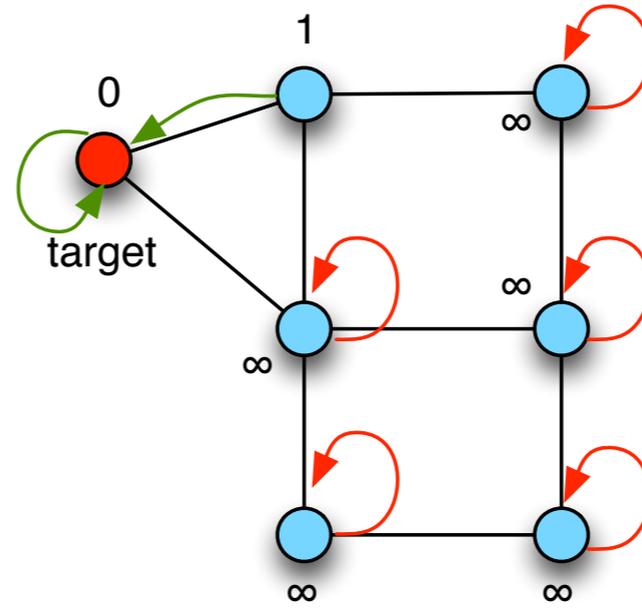
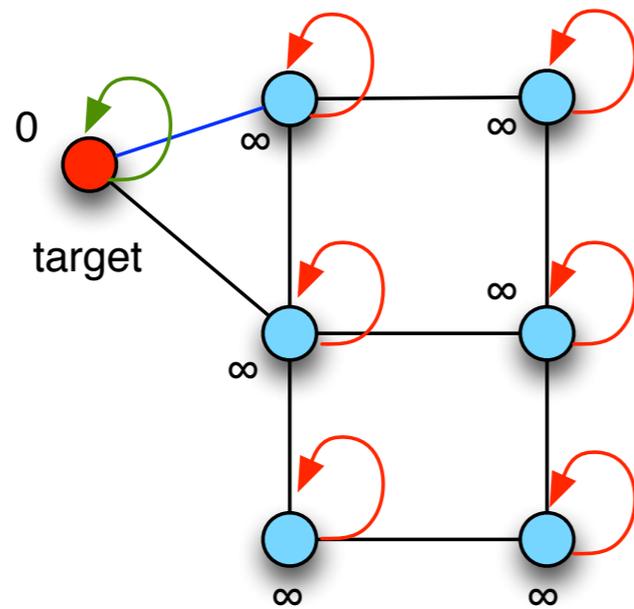
Distributed Bellman Ford for target t (Distance-Vector Routing)

- If node is t then $d(t) \leftarrow 0$; $\pi(t) \leftarrow t$
- If a message from u to $\pi(u)$ fails then
 - $d(u) \leftarrow \infty$
- If u detects a new neighbor v then
 - send $(u, d(u))$ to v
- If u receives $(v, d(v))$ from v
 - if $d(u) > d(v) + w(u, v)$ or $v = \pi(u)$ then
 - $d(u) \leftarrow d(v) + w(u, v)$
 - $\pi(u) \leftarrow v$
- if $d(u) = \infty$ then $\pi(u) \leftarrow u$
- Every time $d(u)$ or $\pi(u)$ has changed u sends $(u, d(u))$ to all neighbors

Beispiel für Distance-Vector für Ziel t

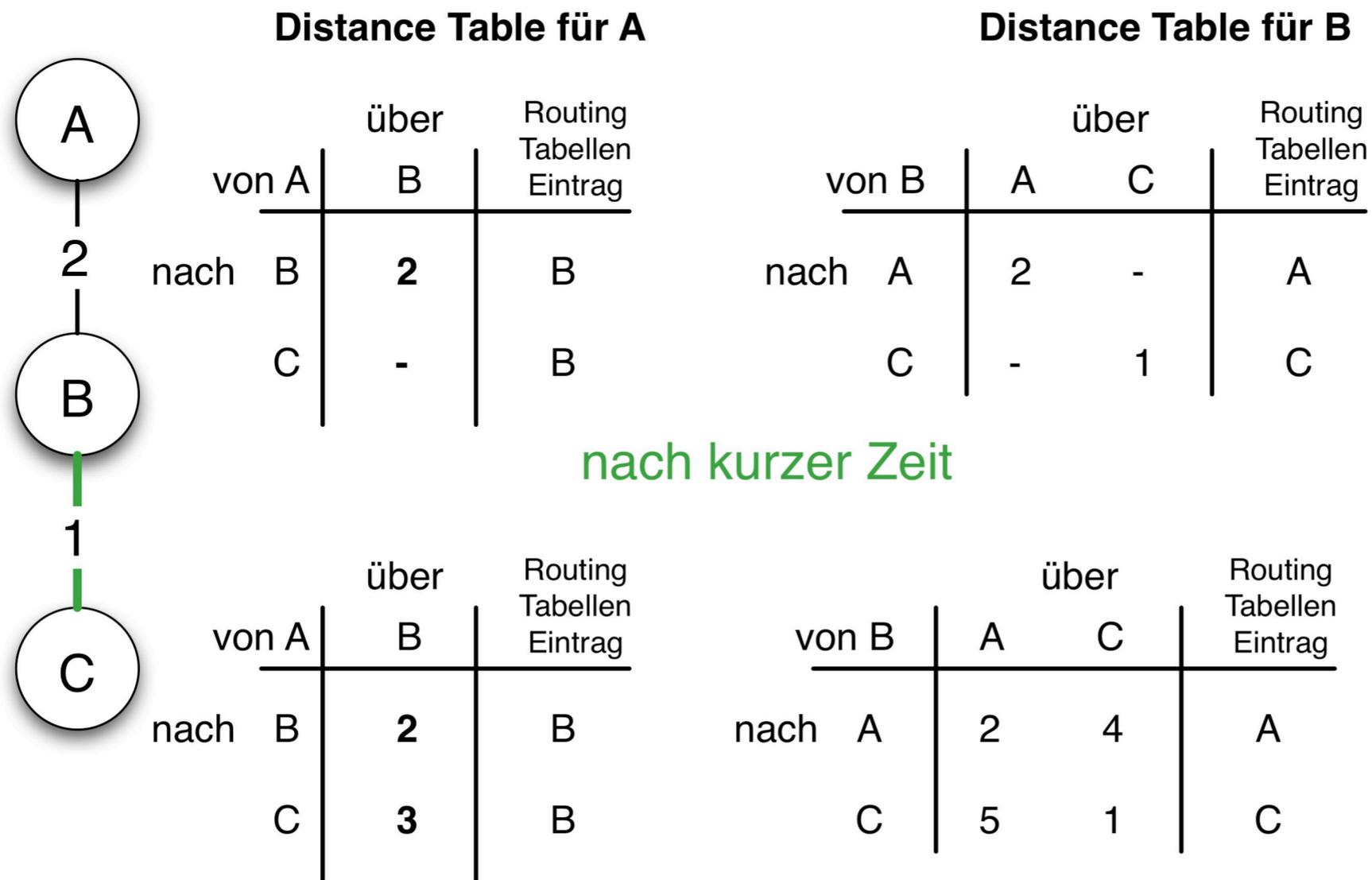


Distance-Vector für ein Ziel



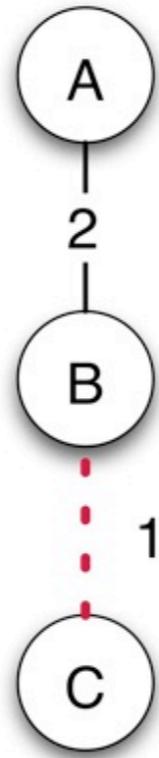
Das "Count to Infinity" - Problem

- Gute Nachrichten verbreiten sich schnell
 - Neue Verbindung wird schnell veröffentlicht



Das “Count to Infinity” - Problem

- Schlechte Nachrichten verbreiten sich langsam
 - Verbindung fällt aus
 - Nachbarn erhöhen wechselseitig ihre Entfernung
 - “Count to Infinity”-Problem



		über		Routing Tabellen Eintrag
		von A	B	
nach	B		2	B
	C		3	B

		über		Routing Tabellen Eintrag
		von A	B	
nach	B		2	B
	C		7	B

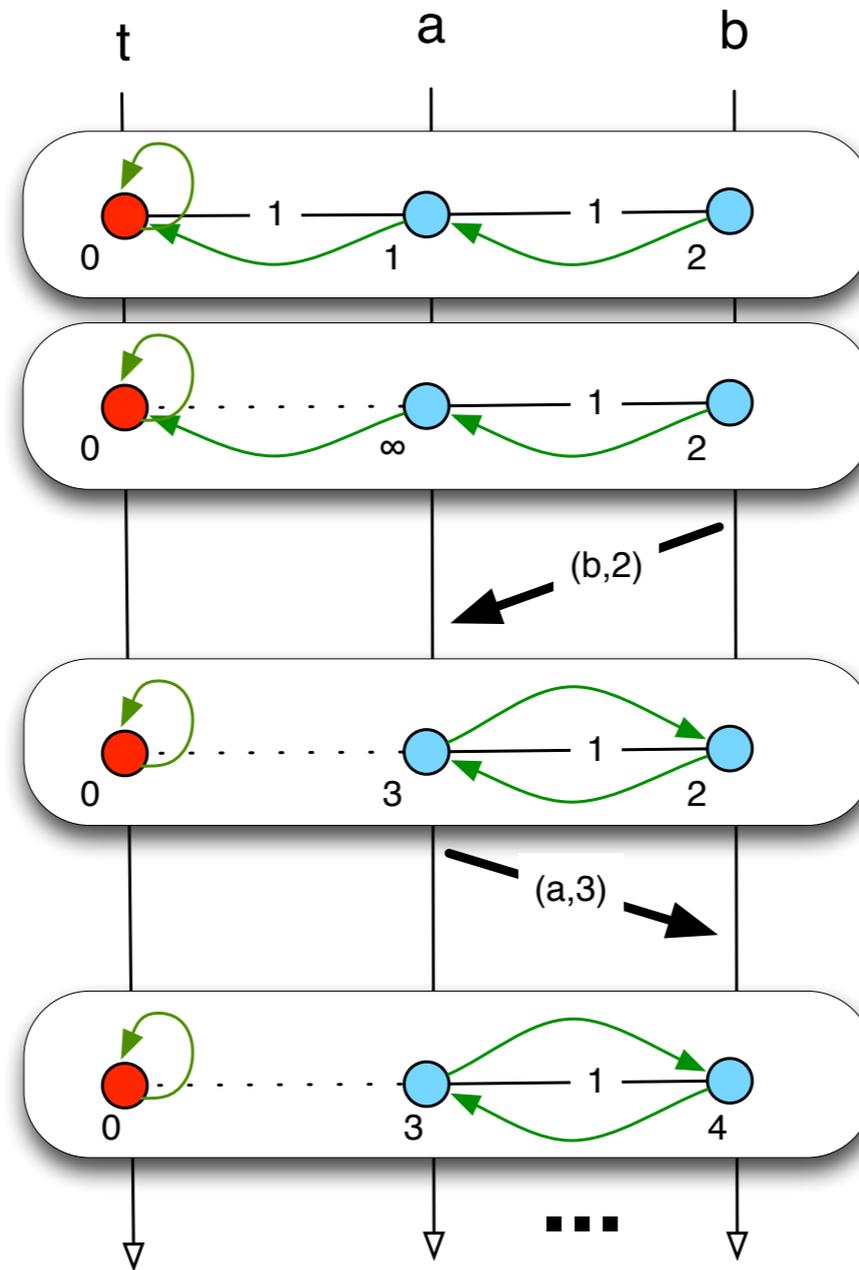
		über		Routing Tabellen Eintrag
		von A	B	
nach	B		2	B
	C		7	B

		über			Routing Tabellen Eintrag
		von B	A	C	
nach	A		2	-	A
	C		5	-	A

		über			Routing Tabellen Eintrag
		von B	A	C	
nach	A		2	-	A
	C		5	-	A

		über			Routing Tabellen Eintrag
		von B	A	C	
nach	A		2	-	A
	C		9	-	A

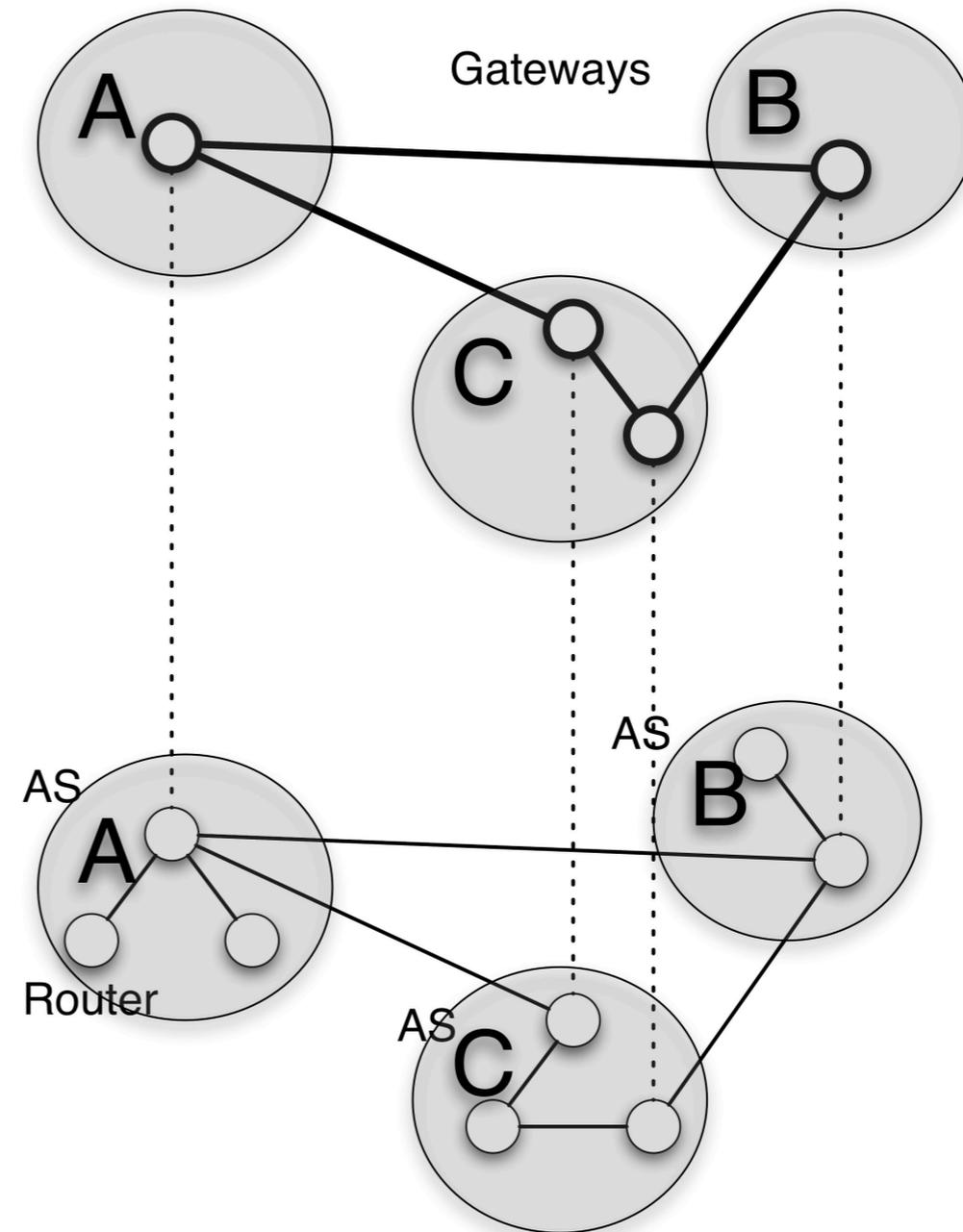
Das “Count to Infinity” - Problem für Ziel t



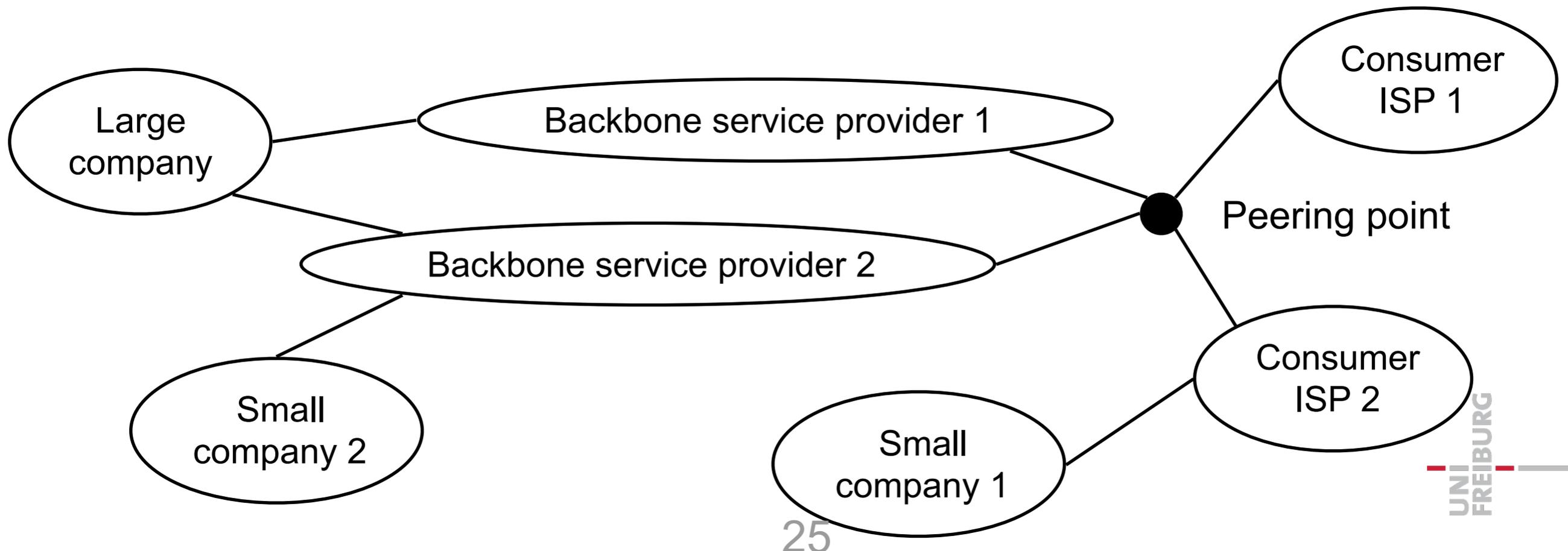
- Link State Router
 - tauschen Information mittels Link State Packets (LSP) aus
 - Jeder verwendet einen eigenen Kürzeste-Wege-Algorithmus zu Anpassung der Routing-Tabelle
- LSP enthält
 - ID des LSP erzeugenden Knotens
 - Kosten dieses Knotens zu jedem direkten Nachbarn
 - Sequenznr. (SEQNO)
 - TTL-Feld für dieses Feld (time to live)
- Verlässliches Fluten (Reliable Flooding)
 - Die aktuellen LSP jedes Knoten werden gespeichert
 - Weiterleitung der LSP zu allen Nachbarn
 - bis auf den Knoten der diese ausgeliefert hat
 - Periodisches Erzeugen neuer LSPs
 - mit steigender SEQNOs
 - Verringern der TTL bei jedem Weiterleiten

- Link State Routing
 - benötigt $O(g \cdot n)$ Einträge für n Router mit maximalen Grad g
 - Jeder Knoten muss an jeden anderen seine Informationen senden
- Distance Vector
 - benötigt $O(g \cdot n)$ Einträge
 - kann Schleifen einrichten
 - Konvergenzzeit steigt mit Netzwerkgröße
- Im Internet gibt es mehr als 10^6 Router
 - damit sind diese so genannten flachen Verfahren nicht einsetzbar
- Lösung:
 - Hierarchisches Routing

- Autonomous System (AS)
 - liefert ein zwei Schichten-Modell des Routing im Internet
 - Beispiele für AS:
 - uni-freiburg.de
- Intra-AS-Routing (Interior Gateway Protocol)
 - ist Routing innerhalb der AS
 - z.B. RIP, OSPF, IGRP, ...
- Inter-AS-Routing (Exterior Gateway Protocol)
 - Übergabepunkte sind Gateways
 - ist vollkommen dezentrales Routing
 - Jeder kann seine Optimierungskriterien vorgeben
 - z.B. EGP (früher), BGP



- Stub-AS
 - Nur eine Verbindung zu anderen AS
- Multihomed AS
 - Verbindungen zu anderen ASen
 - weigert sich aber Verkehr für andere zu befördern
- Transit AS
 - Mehrere Verbindungen
 - Leitet fremde Nachrichten durch (z.B. ISP)



- Distance Vector Algorithmus
 - Distanzmetrik = Hop-Anzahl
- Distanzvektoren
 - werden alle 30s durch Response-Nachricht (advertisement) ausgetauscht
- Für jedes Advertisement
 - Für bis zu 25 Zielnetze werden Routen veröffentlicht per UDP
- Falls kein Advertisement nach 180s empfangen wurde
 - Routen über Nachbarn werden für ungültig erklärt
 - Neue Advertisements werden zu den Nachbarn geschickt
 - Diese antworten auch mit neuen Advertisements
 - falls die Tabellen sich ändern
 - Rückverbindungen werden unterdrückt um Ping-Pong-Schleifen zu verhindern (poison reverse) gegen Count-to-Infinity-Problem
 - Unendliche Distanz = 16 Hops

Intra-AS OSPF (Open Shortest Path First)

- “open” = öffentlich verfügbar
- Link-State-Algorithmus
 - LS Paket-Verbreitung
 - Topologie wird in jedem Knoten abgebildet
 - Routenberechnung mit Dijkstras Algorithmus
- OSPF-Advertisment
 - per TCP, erhöht Sicherheit (security)
 - werden in die gesamte AS geflutet
 - Mehrere Wege gleicher Kosten möglich

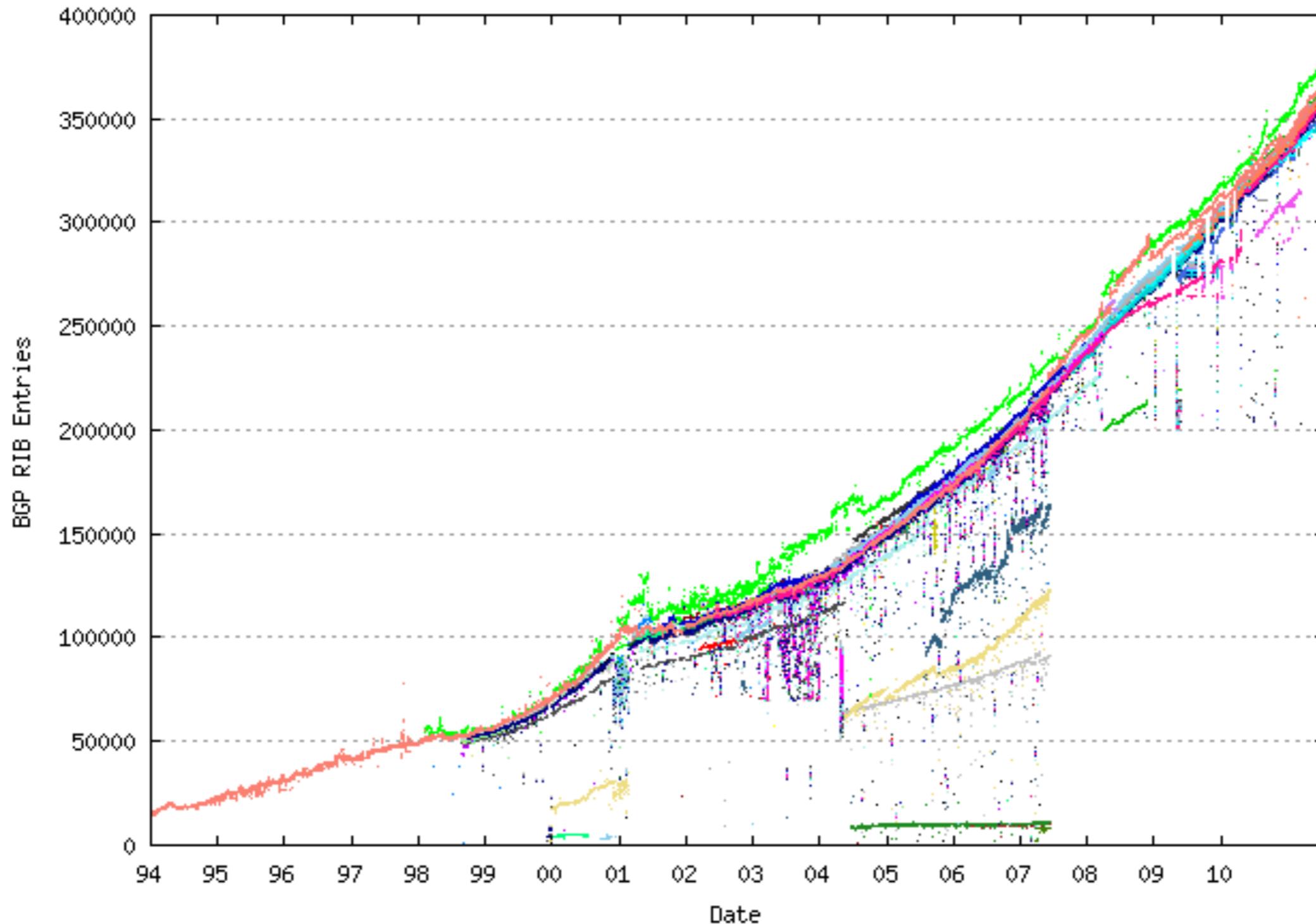
- Für große Netzwerke zwei Ebenen:
 - Lokales Gebiet und Rückgrat (backbone)
 - Lokal: Link-state advertisement
 - Jeder Knoten berechnet nur in Richtung zu den Netzen in anderen lokalen Gebieten
- Local Area Border Router:
 - Fassen die Distanzen in das eigene lokale Gebiet zusammen
 - Bieten diese den anderen Area Border Routern an (per Advertisement)
- Backbone Routers
 - verwenden OSPF beschränkt auf das Rückgrat (backbone)
- Boundary Routers:
 - verbinden zu anderen AS

- CISCO-Protokoll, Nachfolger von RIP (1980er)
- Distance-Vector-Protokoll, wie RIP
 - Hold Down
 - Split Horizon
 - Poison Reverse
- Verschiedene Kostenmetriken
 - Delay, Bandwidth, Reliability, Load etc.
- Verwendet TCP für den Austausch von Routing Updates

- Inter-AS-Routing ist schwierig...
 - Organisationen können Durchleitung von Nachrichten verweigern
 - Politische Anforderungen
 - Weiterleitung durch andere Länder?
 - Routing-Metriken der verschiedenen autonomen Systeme sind oftmals unvergleichbar
 - Wegeoptimierung unmöglich!
 - Inter-AS-Routing versucht wenigstens Erreichbarkeit der Knoten zu ermöglichen
 - Größe: momentan müssen Inter-Domain-Router mehr als 300.000 Einträge verwalten

- Ist faktisch der Standard
- Path-Vector-Protocol
 - ähnlich wie Distance Vector Protocol
 - es werden aber ganze Pfade zum Ziel gespeichert
 - jeder Border Gateway teilt all seinen Nachbarn (peers) den gesamten Pfad (Folge von ASen) zum Ziel mit (advertisement) (per TCP)
- Falls Gateway X den Pfad zum Peer-Gateway W sendet
 - dann kann W den Pfad wählen oder auch nicht
 - Optimierungskriterien:
 - Kosten, Politik, etc.
 - Falls W den Pfad von X wählt, dann publiziert er
 - $\text{Path}(W,Z) = (W, \text{Path}(X,Z))$
- Anmerkung
 - X kann den eingehenden Verkehr kontrollieren durch Senden von Advertisements
 - Sehr kompliziertes Protokoll

BGP-Routing Tabellengröße 1994-2011



- Broadcast routing
 - Ein Paket soll (in Kopie) an alle ausgeliefert werden
 - Lösungen:
 - Fluten des Netzwerks
 - Besser: Konstruktion eines minimalen Spannbaums
- Multicast routing
 - Ein Paket soll an eine gegebene Teilmenge der Knoten ausgeliefert werden (in Kopie)
 - Lösung:
 - Optimal: Steiner Baum Problem (bis heute nicht lösbar)
 - Andere (nicht-optimale) Baum-konstruktionen

- Motivation

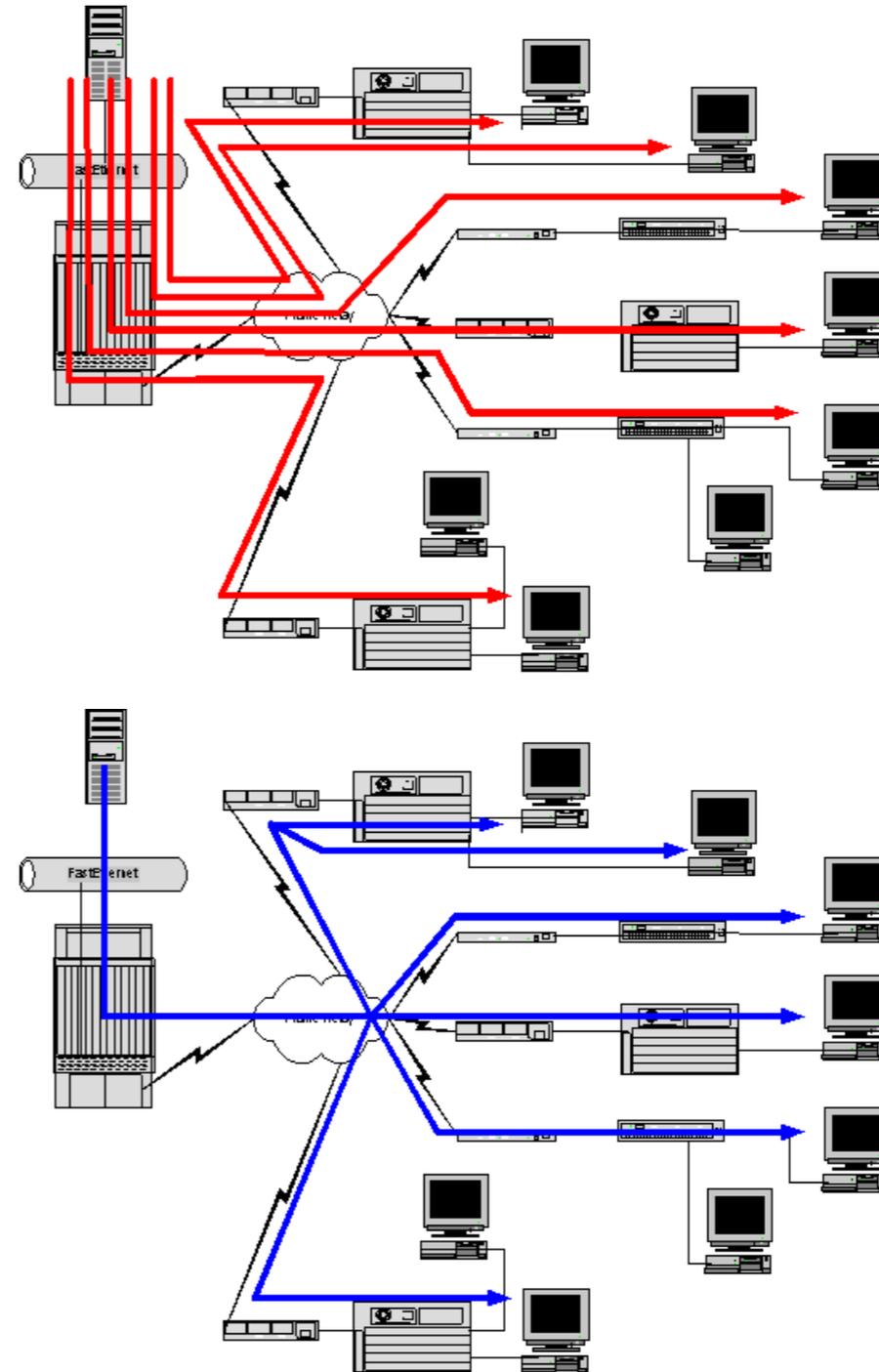
- Übertragung eines Stroms an viele Empfänger

- Unicast

- Strom muss mehrfach einzeln übertragen werden
- Bottleneck am Sender

- Multicast

- Strom wird über die Router vervielfältigt
- Kein Bottleneck mehr



Bilder von Peter J. Welcher

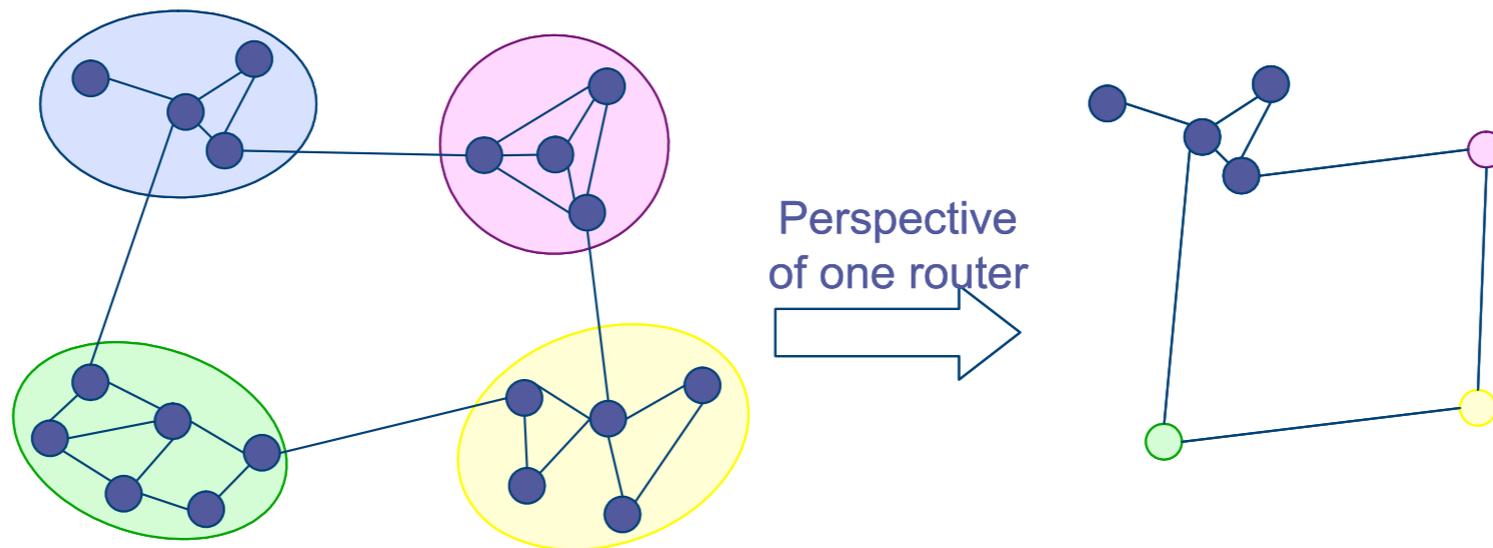
- IPv4 Multicast-Adressen
 - in der Klasse D (außerhalb des CIDR - Classless Interdomain Routings)
 - 224.0.0.0 - 239.255.255.255
- Hosts melden sich per IGMP bei der Adresse an
 - IGMP = Internet Group Management Protocol
 - Nach der Anmeldung wird der Multicast-Tree aktualisiert
- Source sendet an die Multicast-Adresse
 - Router duplizieren die Nachrichten an den Routern
 - und verteilen sie in die Bäume
- Angemeldete Hosts erhalten diese Nachrichten
 - bis zu einem Time-Out
 - oder bis sie sich abmelden
- Achtung:
 - Kein TCP, nur UDP
 - Viele Router lehnen die Beförderung von Multicast-Nachrichten ab
 - Lösung: Tunneln

- Distance Vector Multicast Routing Protocol (DVMRP)
 - jahrelang eingesetzt in MBONE (insbesondere in Freiburg)
 - Eigene Routing-Tabelle für Multicast
- Protocol Independent Multicast (PIM)
 - im Sparse Mode (PIM-SM)
 - aktueller Standard
 - beschneidet den Multicast Baum
 - benutzt Unicast-Routing-Tabellen
 - ist damit weitestgehend protokollunabhängig
- Voraussetzung PIM-SM:
 - benötigt Rendezvous-Point (RP) in ein-Hop-Entfernung
 - RP muss PIM-SM unterstützen
 - oder Tunneling zu einem Proxy in der Nähe eines RP

Warum so wenig IP Multicast?

- Trotz erfolgreichen Einsatz
 - in Video-Übertragung von IETF-Meetings
 - MBONE (Multicast Backbone)
- gibt es wenig ISP welche IP Multicast in den Routern unterstützen
- Zusätzlicher Wartungsaufwand
 - Schwierig zu konfigurieren
 - Verschiedene Protokolle
- Gefahr von Denial-of-Service-Attacks
 - Implikationen größer als bei Unicast
- Transport-Protokoll
 - Nur UDP einsetzbar
 - Zuverlässige Protokolle
 - Vorwärtsfehlerkorrektur
 - Oder proprietäre Protokolle in den Routern (z.B. CISCO)
- Marktsituation
 - Endkunden fragen kaum Multicast nach (benutzen lieber P2P-Netzwerke)
 - Wegen einzelner Dateien und weniger Abnehmer erscheint ein Multicast wenig erstrebenswert (Adressenknappheit!)

- Flache (MAC-) Adressen haben keine Strukturinformation

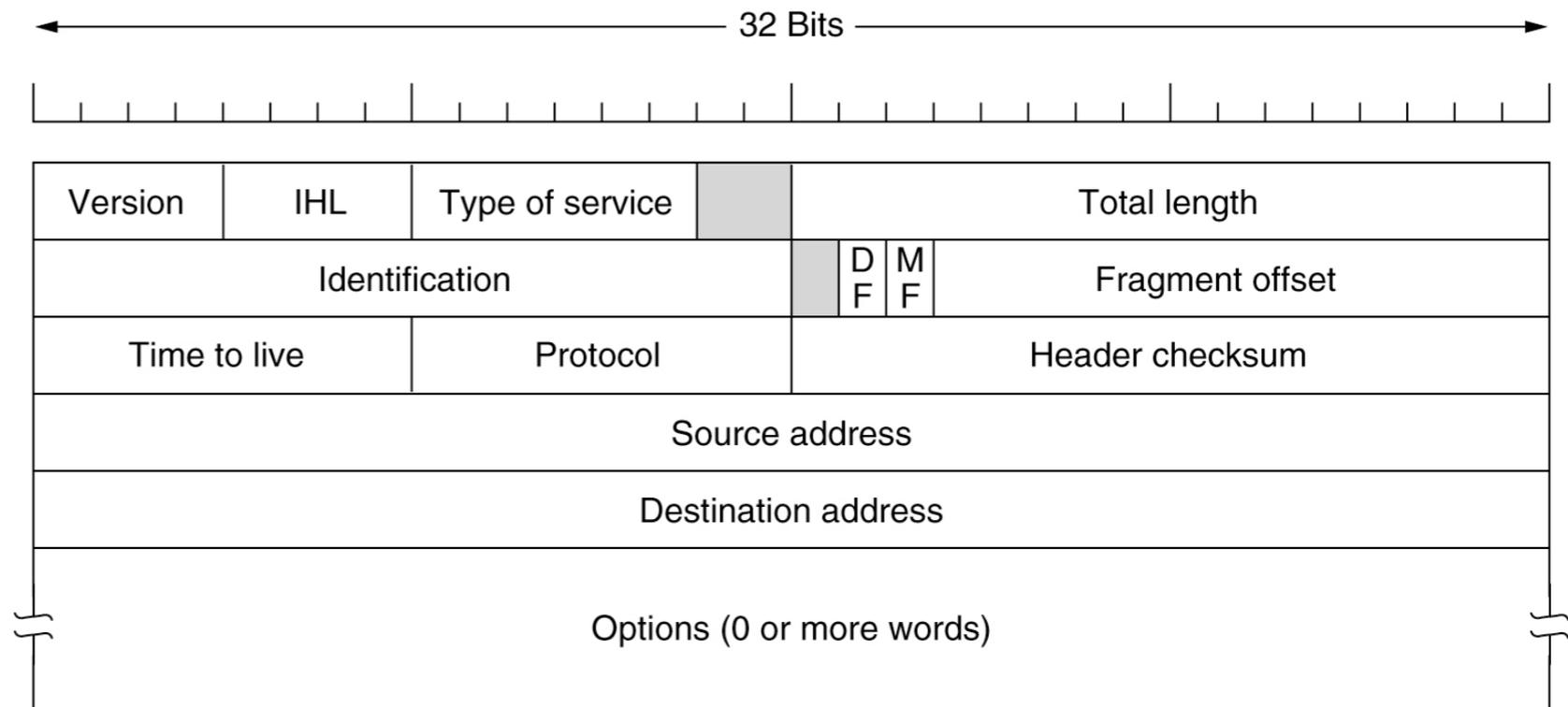


- Hierarchische Adressen
 - Routing wird vereinfacht wenn Adressen hierarchische Routing-Struktur abbilden
 - $\text{Group-ID}_n:\text{Group-ID}_{n-1}:\dots:\text{Group-ID}_1:\text{Device-ID}$

- IP-Adressen
 - Jedes Interface in einem Netzwerk hat weltweit eindeutige IP-Adresse
 - 32 Bits unterteilt in Net-ID und Host-ID
 - Net-ID vergeben durch Internet Network Information Center
 - Host-ID durch lokale Netzwerkadministration
- Domain Name System (DNS)
 - Ersetzt IP-Adressen wie z.B. 132.230.167.230 durch Namen wie z.B. falcon.informatik.uni-freiburg.de und umgekehrt
 - Verteilte robuste Datenbank

IPv4-Header (RFC 791)

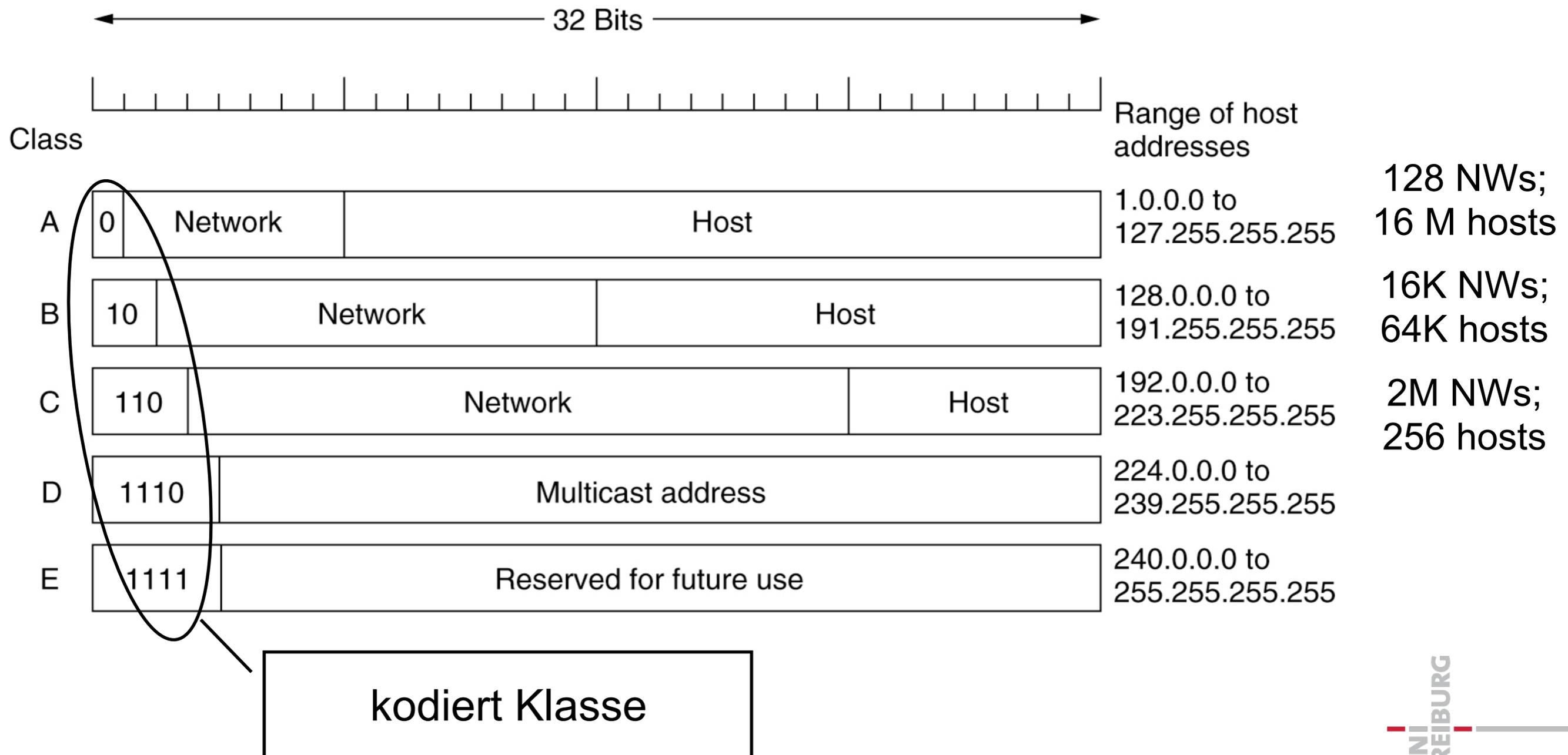
- Version: 4 = IPv4
- IHL: IP Headerlänge
 - in 32 Bit-Wörtern (>5)
- Type of Service
 - Optimiere delay, throughput, reliability, monetary cost
- Checksum (nur für IP-Header)
- Source and destination IP-address
- Protocol, identifiziert passendes Protokoll
 - Z.B. TCP, UDP, ICMP, IGMP
- Time to Live:
 - maximale Anzahl Hops



- IP-Adressen unterscheiden zwei Hierarchien
 - Netzwerk-Interfaces
 - Netzwerke
 - Verschiedene Netzwerkgrößen
 - Netzwerkklassen:
 - Groß - mittel - klein
(Klasse A, B, and C)
- Eine IP-Adresse hat 32 Bits
 - Erster Teil: Netzwerkadresse
 - Zweiter Teil: Interface

IP-Klassen bis 1993

- Klassen A, B, and C
- D für multicast; E: "reserved"



- Bis 1993 (heutzutage veraltet)
 - 5 Klassen gekennzeichnet durch Präfix
 - Dann Subnetzpräfix fester Länge und Host-ID (Geräteteil)
- Seit 1993
 - Classless Inter-Domain-Routing (CIDR)
 - Die Netzwerk-Adresse und die Host-ID (Geräteteil) werden variabel durch die Netzwerkmaske aufgeteilt.
 - Z.B.:
 - Die Netzwerkmaske 11111111.11111111.11111111.00000000
 - Besagt, dass die IP-Adresse
 - 10000100. 11100110. 10010110. 11110011
 - Aus dem Netzwerk 10000100. 11100110. 10010110
 - den Host 11110011 bezeichnet
- Route aggregation
 - Die Routing-Protokolle BGP, RIP v2 und OSPF können verschiedene Netzwerke unter einer ID anbieten
 - Z.B. alle Netzwerke mit Präfix 10010101010* werden über Host X erreicht

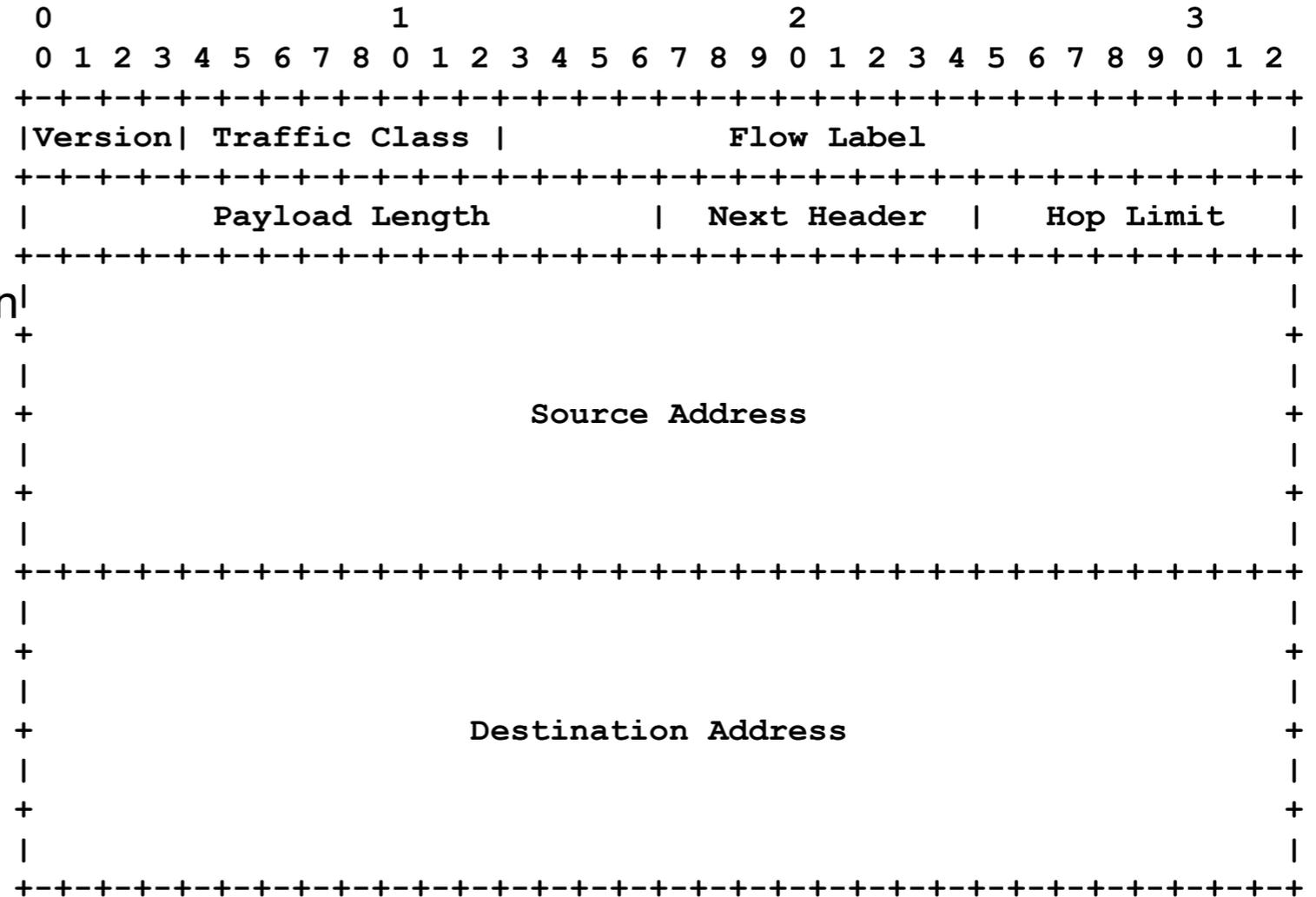
- Address Resolution Protocol (ARP)
- Umwandlung: IP-Adresse in MAC-Adresse
 - Broadcast im LAN, um nach Rechner mit passender IP-Adresse zu fragen
 - Knoten antwortet mit MAC-Adresse
 - Router kann dann das Paket dorthin ausliefern

- Wozu IPv6:
- IP-Adressen sind knapp
 - Zwar gibt es 4 Milliarden in IPv4 (32 Bit)
 - Diese sind aber statisch organisiert in Netzwerk- und Rechnerteil
 - Adressen für Funktelefone, Kühlschränke, Autos, Tastaturen, etc...
- Autokonfiguration
 - DHCP, Mobile IP, Umnummerierung
- Neue Dienste
 - Sicherheit (IPSec)
 - Qualitätssicherung (QoS)
 - Multicast
- Vereinfachungen für Router
 - keine IP-Prüfsummen
 - Keine Partitionierung von IP-Paketen

- DHCP (Dynamic Host Configuration Protocol)
 - Manuelle Zuordnung (Bindung an die MAC-Adresse, z.B. für Server)
 - Automatische Zuordnung (feste Zuordnung, nicht voreingestellt)
 - Dynamische Zuordnung (Neuvergabe möglich)
- Einbindung neuer Rechner ohne Konfiguration
 - Rechner „holt“ sich die IP-Adresse von einem DHCP-Server
 - Dieser weist dem Rechner die IP-Adressen dynamisch zu
 - Nachdem der Rechner das Netzwerk verlässt, kann die IP-Adresse wieder vergeben werden
 - Bei dynamischer Zuordnung, müssen IP-Adressen auch „aufgefrischt“ werden
 - Versucht ein Rechner eine alte IP-Adresse zu verwenden,
 - die abgelaufen ist oder
 - schon neu vergeben ist
 - Dann werden entsprechende Anfragen zurückgewiesen
 - Problem: Stehlen von IP-Adressen

IPv6-Header (RFC 2460)

- Version: 6 = IPv6
- Traffic Class
 - Für QoS (Prioritätsvergabe)
- Flow Label
 - Für QoS oder Echtzeitanwendungen
- Payload Length
 - Größe des Rests des IP-Pakets (Datagramms)
- Next Header (wie bei IPv4: protocol)
 - Z.B. ICMP, IGMP, TCP, EGP, UDP, Multiplexing, ...
- Hop Limit (Time to Live)
 - maximale Anzahl Hops
- Source Address
- Destination Address
 - 128 Bit IPv6-Adresse



- Schutz vor Replay-Attacken
- IKE (Internet Key Exchange) Protokoll
 - Vereinbarung einer Security Association
 - Identifikation, Festlegung von Schlüsseln, Netzwerke, Erneuerungszeiträume für Authentifizierung und IPsec Schlüssel
 - Erzeugung einer SA im Schnellmodus (nach Etablierung)
- Encapsulating Security Payload (ESP)
 - IP-Kopf unverschlüsselt, Nutzdaten verschlüsselt, mit Authentifizierung
- IPsec im Transportmodus (für direkte Verbindungen)
 - IPsec Header zwischen IP-Header und Nutzdaten
 - Überprüfung in den IP-Routern (dort muss IPsec vorhanden sein)
- IPsec im Tunnelmodus (falls mindestens ein Router dazwischen ist)
 - Das komplette IP-Paket wird verschlüsselt und mit dem IPsec-Header in einen neuen IP-Header verpackt
 - Nur an den Enden muss IPsec vorhanden sein.
- IPsec ist Bestandteil von IPv6
- Rückportierungen nach IPv4 existieren

- Typen von Firewalls
 - Host-Firewall
 - Netzwerk-Firewall
- Netzwerk-Firewall
 - unterscheidet
 - Externes Netz
(Internet - feindselig)
 - Internes Netz
(LAN - vertrauenswürdig)
 - Demilitarisierte Zone
(vom externen Netz erreichbare Server)
- Host-Firewall
 - z.B. Personal Firewall
 - kontrolliert den gesamten Datenverkehr eines Rechners
 - Schutz vor Attacken von außerhalb und von innen (Trojanern)

- Paketfilter
 - Sperren von Ports oder IP-Adressen
 - Content-Filter
 - Filtern von SPAM-Mails, Viren, ActiveX oder JavaScript aus HTML-Seiten
- Proxy
 - Transparente (extern sichtbare) Hosts
 - Kanalisierung der Kommunikation und möglicher Attacken auf gesicherte Rechner
- NAT, PAT
 - Network Address Translation
 - Port Address Translation
- Bastion Host
- Proxy

- (Network) Firewall
 - beschränkt den Zugriff auf ein geschütztes Netzwerk aus dem Internet
- Paket-Filter
 - wählen Pakete aus dem Datenfluss in oder aus dem Netzwerk aus
 - Zweck des Eingangsfilters:
 - z.B. Verletzung der Zugriffskontrolle
 - Zweck des Ausgangsfilters:
 - z.B. Trojaner
- Bastion Host
 - ist ein Rechner an der Peripherie, der besonderen Gefahren ausgesetzt ist
 - und daher besonders geschützt ist
- Dual-homed host
 - Normaler Rechner mit zwei Interfaces (verbindet zwei Netzwerke)

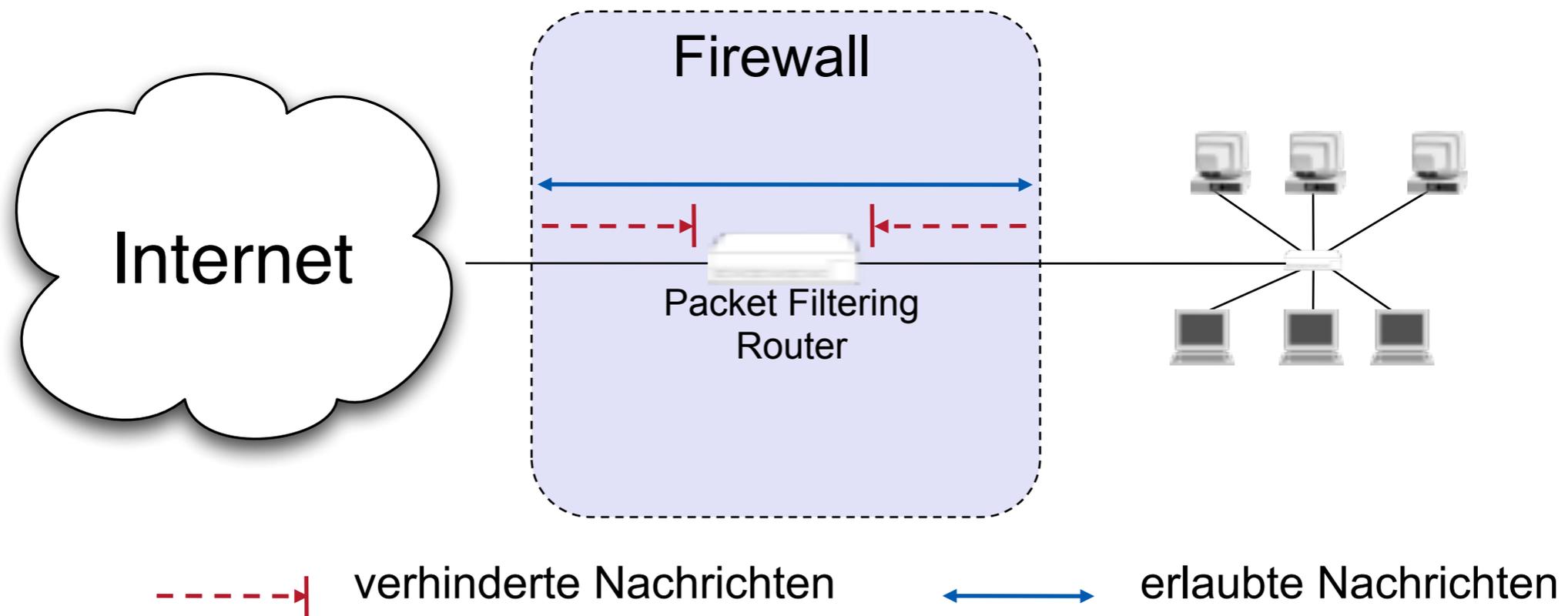
- Proxy (Stellvertreter)
 - Spezieller Rechner, über den Anfragen umgeleitet werden
 - Anfragen und Antworten werden über den Proxy geleitet
 - Vorteil
 - Nur dort müssen Abwehrmaßnahmen getroffen werden
- Perimeter Network:
 - Ein Teilnetzwerk, das zwischen gesicherter und ungesicherter Zone eine zusätzliche Schutzschicht bietet
 - Synonym demilitarisierte Zone (DMZ)

- NAT (Network Address Translation)
- Basic NAT (Static NAT)
 - Jede interne IP wird durch eine externe IP ersetzt
- Hiding NAT = PAT (Port Address Translation) = NAPT (Network Address Port Translation)
 - Das Socket-Paar (IP-Adresse und Port-Nummer) wird umkodiert

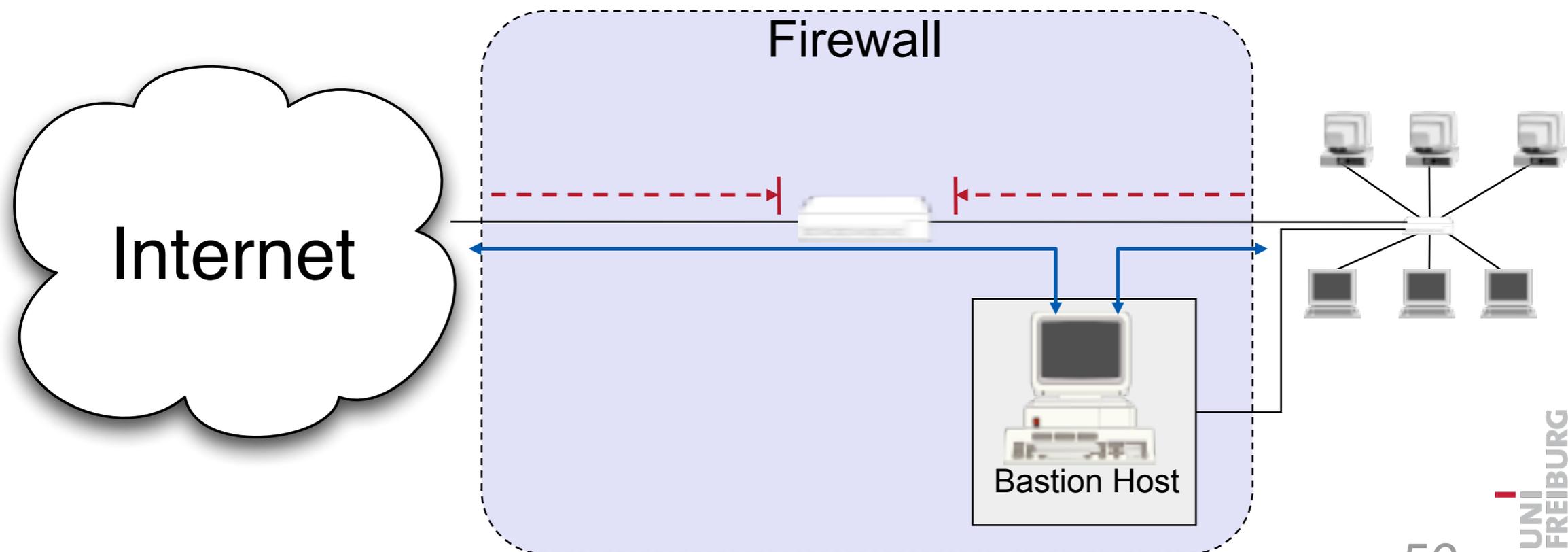
- Verfahren
 - Die verschiedenen lokalen Rechner werden in den Ports kodiert
 - Diese werden im Router an der Verbindung zum WAN dann geeignet kodiert
 - Bei ausgehenden Paketen wird die LAN-IP-Adresse und ein kodierter Port als Quelle angegeben
 - Bei eingehenden Paketen (mit der LAN-IP-Adresse als Ziel), kann dann aus dem kodierten Port der lokale Rechner und der passende Port aus einer Tabelle zurückgerechnet werden
- Sicherheitsvorteile
 - Rechner im lokalen Netzwerk können nicht direkt angesprochen werden
 - Löst auch das Problem knapper IPv4-Adressen
 - Lokale Rechner können nicht als Server dienen
- DHCP (Dynamic Host Configuration Protocol)
 - bringt ähnliche Vorteile

Firewall-Architektur Einfacher Paketfilter

- Realisiert durch
 - Eine Standard-Workstation (e.g. Linux PC) mit zwei Netzwerk-Interfaces und Filter-Software oder
 - Spezielles Router-Gerät mit Filterfähigkeiten

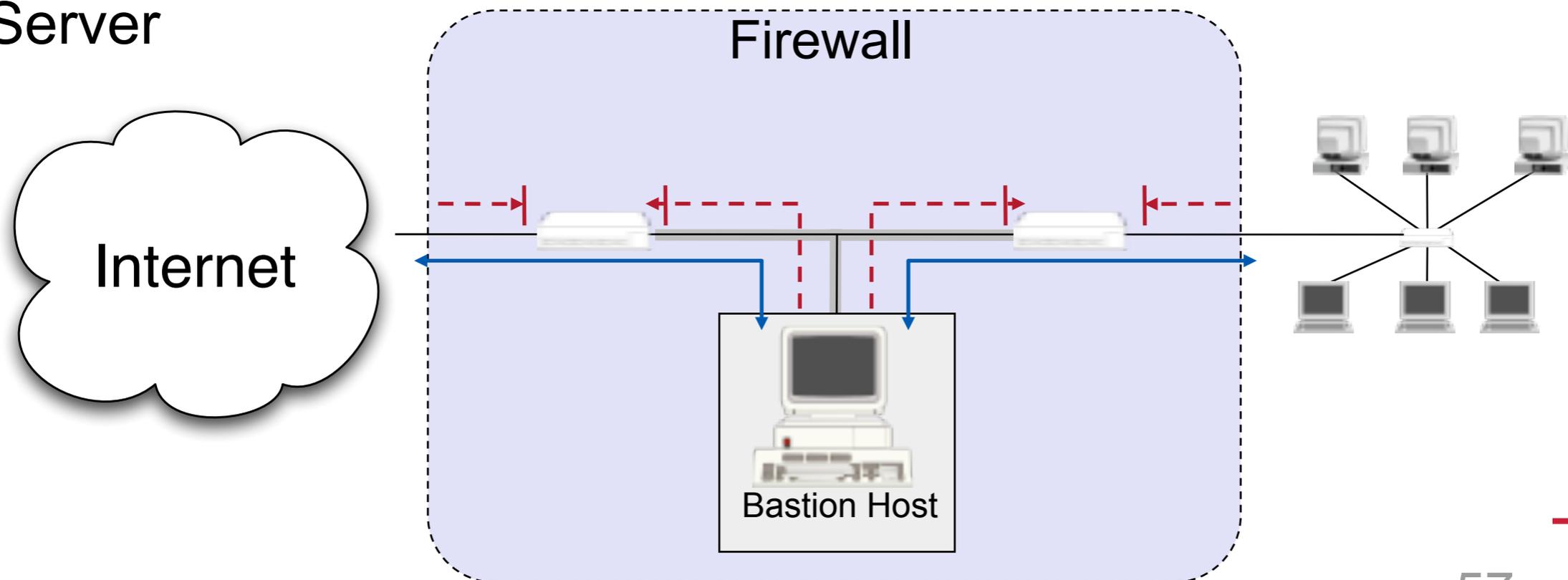


- Screened Host
- Der Paketfilter
 - erlaubt nur Verkehr zwischen Internet und dem Bastion Host und
 - Bastion Host und geschützten Netzwerk
- Der Screened Host bietet sich als Proxy an
 - Der Proxy Host hat die Fähigkeiten selbst Angriffe abzuwehren



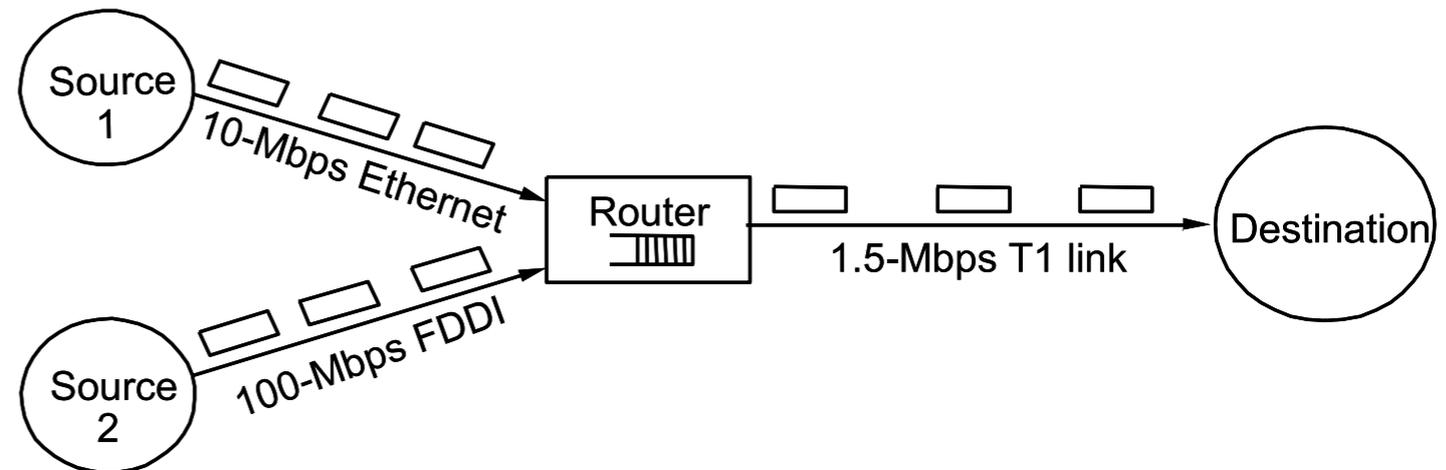
Firewall-Architektur Screened Subnet

- Perimeter network zwischen Paketfiltern
- Der innere Paketfilter schützt das innere Netzwerk, falls das Perimeter-Network in Schwierigkeiten kommt
 - Ein gehackter Bastion Host kann so das Netzwerk nicht ausspionieren
- Perimeter Netzwerke sind besonders geeignet für die Bereitstellung öffentlicher Dienste, z.B. FTP, oder WWW-Server



- Fähigkeiten von Paketfilter
 - Erkennung von Typ möglich (Demultiplexing-Information)
- Verkehrskontrolle durch
 - Source IP Address
 - Destination IP Address
 - Transport protocol
 - Source/destination application port
- Grenzen von Paketfiltern (und Firewalls)
 - Tunnel-Algorithmen sind aber mitunter nicht erkennbar
 - Möglich ist aber auch Eindringen über andere Verbindungen
 - z.B. Laptops, UMTS, GSM, Memory Sticks

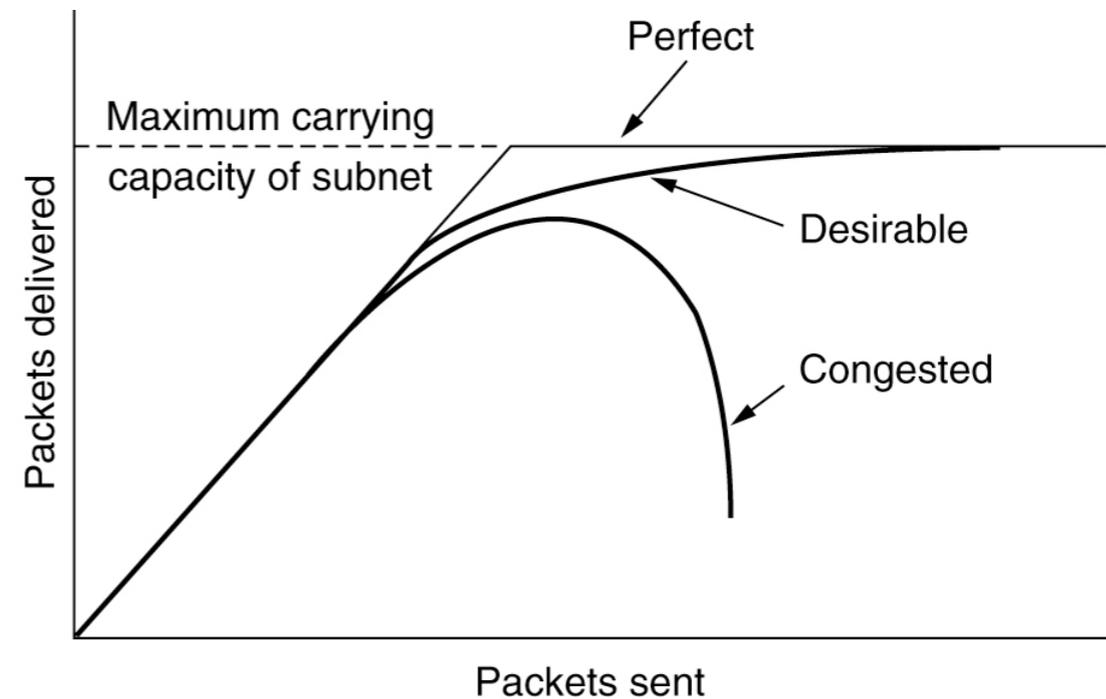
- Jedes Netzwerk hat eine eingeschränkte Übertragungs-Bandbreite



- Wenn mehr Daten in das Netzwerk eingeleitet werden, führt das zum

- Datenstau (congestion) oder gar
- Netzwerkzusammenbruch (congestive collapse)

- Folge: Datenpakete werden nicht ausgeliefert



- Congestion control soll Schneeballeffekte vermeiden
 - Netzwerküberlast führt zu Paketverlust (Pufferüberlauf, ...)
 - Paketverlust führt zu Neuversand
 - Neuversand erhöht Netzwerklast
 - Höherer Paketverlust
 - Mehr neu versandte Pakete
 - ...

- **Effizienz**
 - Verzögerung klein
 - Durchsatz hoch

- **Fairness**
 - Jeder Fluss bekommt einen fairen Anteil
 - Priorisierung möglich
 - gemäß Anwendung
 - und Bedarf

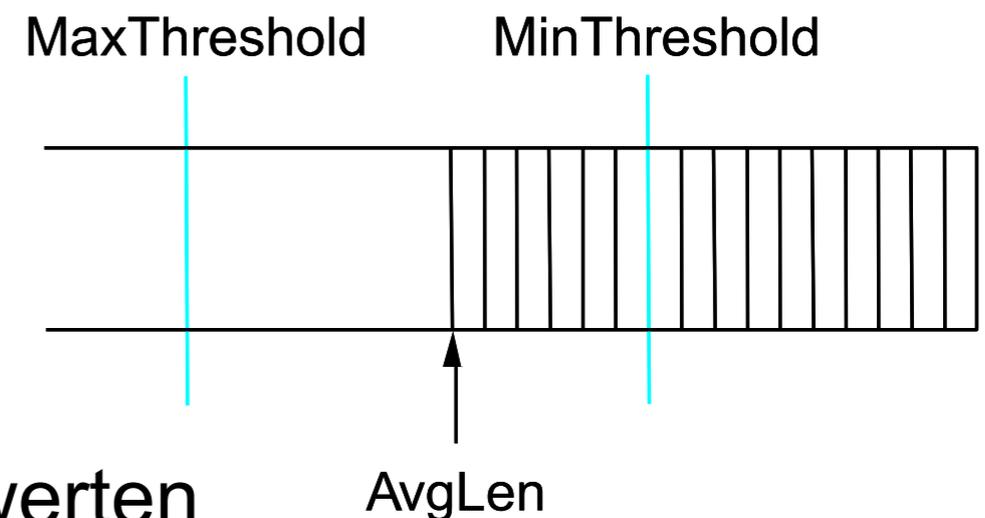
- Erhöhung der Kapazität
 - Aktivierung weiterer Verbindungen, Router
 - Benötigt Zeit und in der Regel den Eingriff der Systemadministration
- Reservierung und Zugangskontrolle
 - Verhinderung neuen Verkehrs an der Kapazitätsgrenze
 - Typisch für (Virtual) Circuit Switching
- Verringerung und Steuerung der Last
 - (Dezentrale) Verringerung der angeforderten Last bestehender Verbindungen
 - Benötigt Feedback aus dem Netzwerk
 - Typisch für Packet Switching
 - wird in TCP verwendet

- Router- oder Host-orientiert
 - Messpunkt (wo wird der Stau bemerkt)
 - Steuerung (wo werden die Entscheidungen gefällt)
 - Aktion (wo werden Maßnahmen ergriffen)
- Fenster-basiert oder Raten-basiert
 - Rate: x Bytes pro Sekunde
 - Fenster: siehe Fenstermechanismen in der Sicherungsschicht
 - wird im Internet verwendet

- Bei Pufferüberlauf im Router
 - muss (mindestens) ein Paket gelöscht werden
- Das zuletzt angekommene Paket löschen (*drop-tail queue*)
 - Intuition: “Alte” Pakete sind wichtiger als neue (Wein)
 - z.B. für go-back-n-Strategie
- Ein älteres Paket im Puffer löschen
 - Intuition: Für Multimedia-Verkehr sind neue Pakete wichtiger als alte (Milch)

- Paketverlust durch Pufferüberlauf im Router erzeugt Feedback in der Transportschicht beim Sender durch ausstehende Bestätigungen
 - Internet
- Annahme:
 - Paketverlust wird hauptsächlich durch Stau ausgelöst
- Maßnahme:
 - Transport-Protokoll passt Senderate an die neue Situation an

- Pufferüberlauf deutet auf Netzwerküberlast hin
- Idee: Proaktives Feedback = Stauvermeidung (Congestion avoidance)



- Aktion bereits bei kritischen Anzeigewerten
- z.B. bei Überschreitung einer Puffergröße
- z.B. wenn kontinuierlich mehr Verkehr eingeht als ausgeliefert werden kann
- ...
- Router ist dann in einem Warn-Zustand

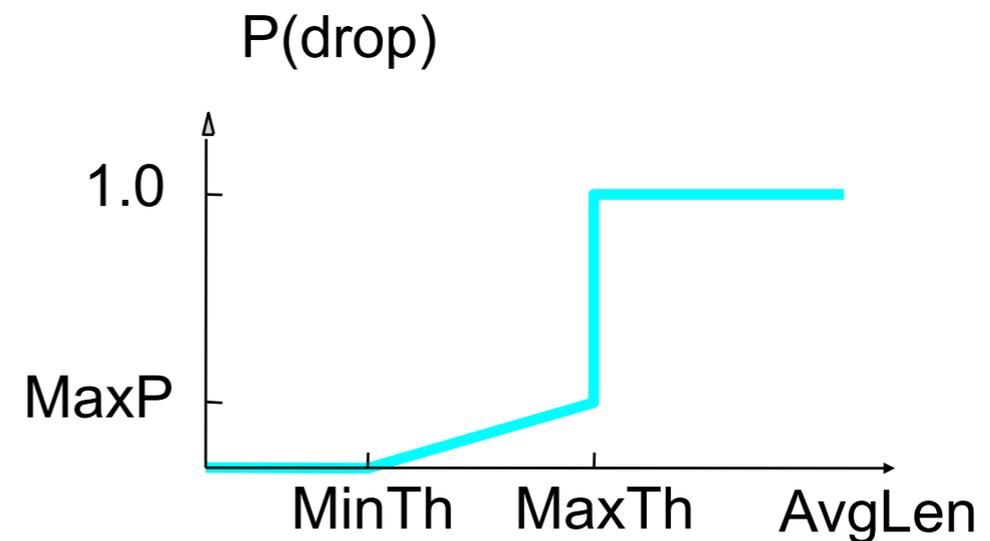
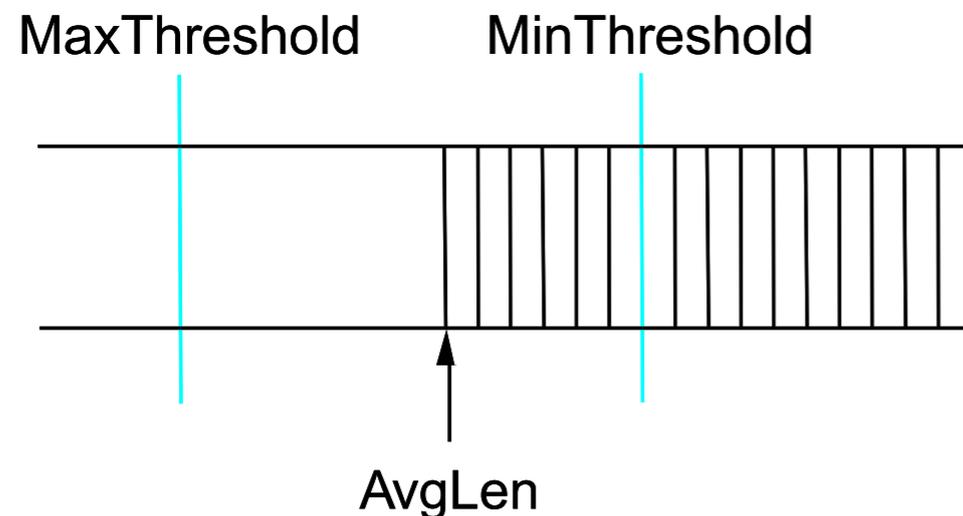
Proactive Aktion: Pakete drosseln (Choke packets)

- Wenn der Router in dem Warnzustand ist:
 - Sendet er Choke-Pakete (Drossel-Pakete) zum Sender
- Choke-Pakete fordern den Sender auf die Senderate zu verringern
- Problem:
 - Im kritischen Zustand werden noch mehr Pakete erzeugt
 - Bis zur Reaktion beim Sender vergrößert sich das Problem

- Wenn der Router in dem Warnzustand ist:
 - Sendet er Warn-Bits in allen Paketen zum Ziel-Host
- Ziel-Host sendet diese Warn-Bits in den Bestätigungs-Bits zurück zum Sender
 - Quelle erhält Warnung und reduziert Sende-Rate

Proaktive Aktion: Random early detection (RED)

- Verlorene Pakete werden als Indiz aufgefasst
- Router löschen Pakete willkürlich im Warnzustand
- Löschrage kann mit der Puffergröße steigen



- Raten-basierte Protokolle
 - Reduzierung der Sende-Rate
 - Problem: Um wieviel?
- Fenster-basierte Protokolle:
 - Verringerung des Congestion-Fensters
 - z.B. mit AIMD (additive increase, multiplicative decrease)

Systeme II

7. Die Vermittlungsschicht

Christian Schindelhauer

Technische Fakultät

Rechnernetze und Telematik

Albert-Ludwigs-Universität Freiburg