



# Systeme II

**7. Vorlesungswoche  
09.06. – 13.06.2008**

Albert-Ludwigs-Universität Freiburg  
Institut für Informatik  
Rechnernetze und Telematik  
Christian Schindelhauer  
Sommer 2008

Die Vermittlungsschicht  
The Network Layer

# Kapitel V

# Circuit Switching oder Packet Switching

## ▶ **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

# Circuit Switching oder Packet Switching

## ▶ 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

# Taktik der Schichten

## ▶ Transport

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

## ▶ Vermittlung

- Quality of Service (virtuelles Circuit Switching)

## ▶ Sicherung

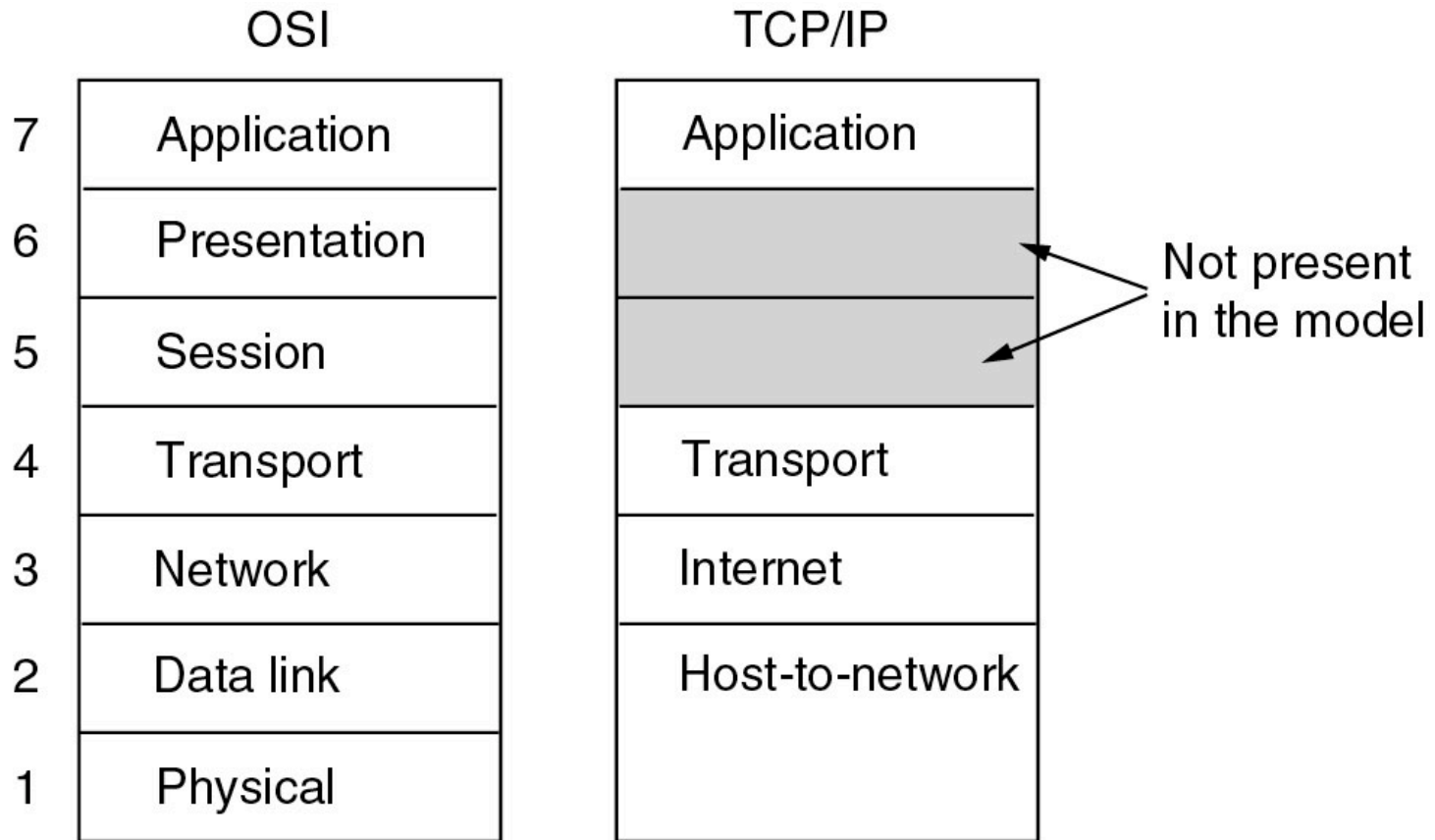
- Flusskontrolle zur Auslastung des Kanals

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

# 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.)

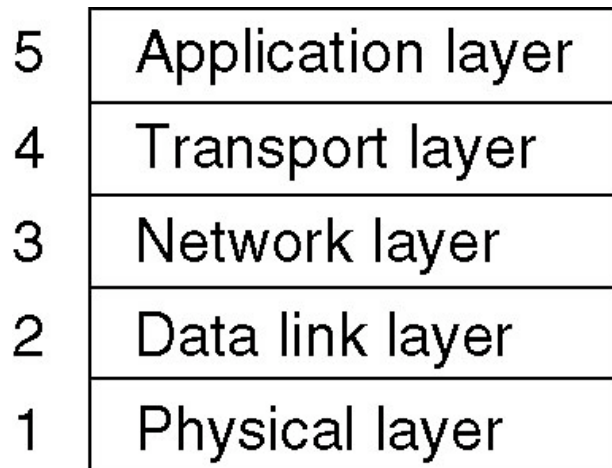
# OSI versus TCP/IP



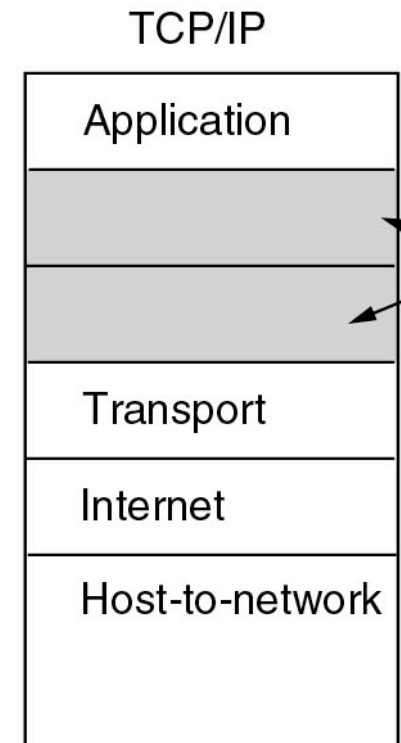
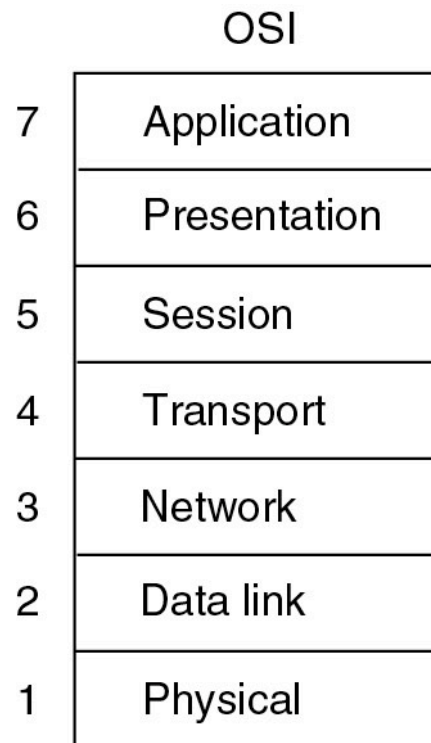
(Aus Tanenbaum)

# Hybrides Modell

- ▶ Wir verwenden hier Tanenbaums hybrides Modell



**Tanenbaum**



(Aus Tanenbaum)

Rechnernetze und Telematik  
Albert-Ludwigs-Universität Freiburg  
Christian Schindelhauer



# Warum eine Vermittlungsschicht

- ▶ **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 10 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**

# Routing-Tabelle und Paket-Weiterleitung

## ▶ 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

# Paket-Weiterleitung im Internet Protokoll

- ▶ **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

# Statisches und Dynamisches Routing

- ▶ **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

# Das Kürzeste-Wege-Problem

- ▶ **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

# Kürzeste Wege mit Edsger Wybe Dijkstra

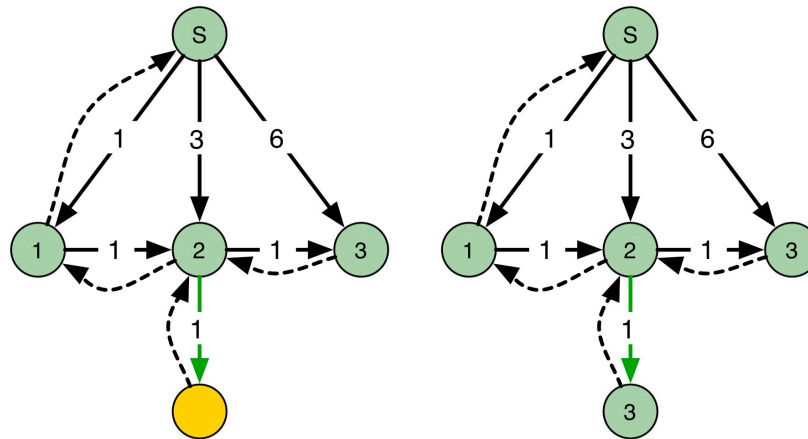
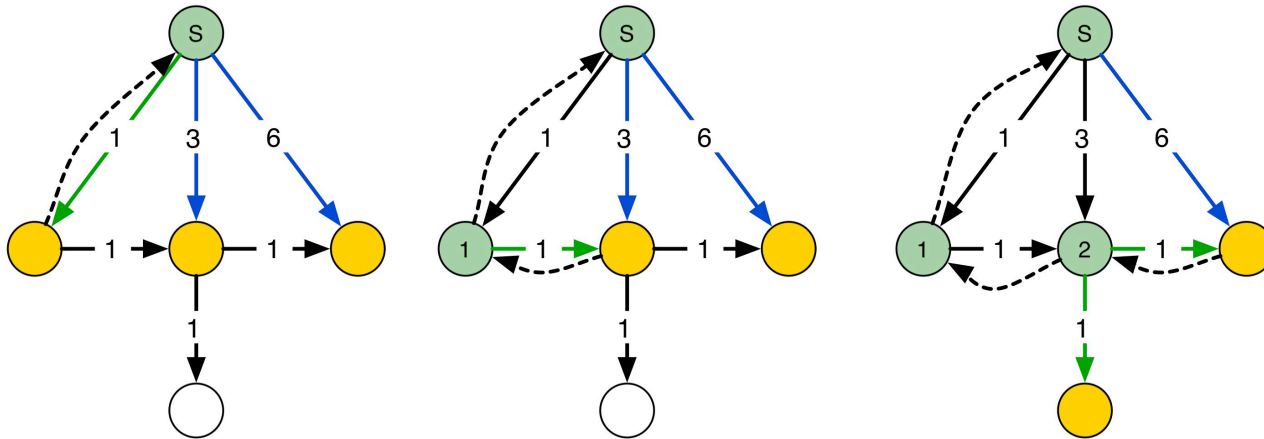
- ▶ Dijkstras Kürzeste-Wege-Algorithmus kann mit Laufzeit  $\Theta(|E| + |V| \log |V|)$  implementiert werden.

```
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
```

```
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
```

```
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



# Bellman-Ford

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

```
Bellman-Ford( $G, w, s$ )
begin
  Init-Source( $G, w$ )
  loop  $|V| - 1$  times do
    for all  $(u, v) \in E$  do
      Relax( $u, v$ )
    od
  od
  for all  $(u, v) \in E$  do
    if  $d(v) > d(u) + w(u, v)$  then return false
  od
  return true
end
```

```
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
```

```
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
```



# Distance Vector Routing Protocol

## ▶ Distance Table Datenstruktur

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

## ▶ Verteilter Algorithmus

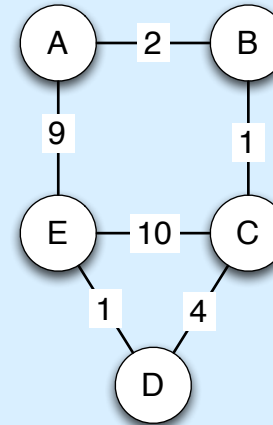
- Jeder Knoten kommuniziert nur mit seinem Nachbarn

## ▶ 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	E

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

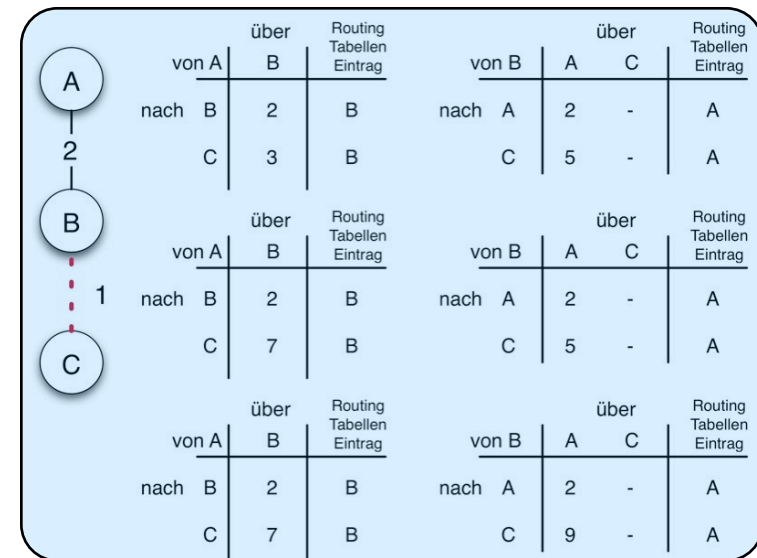
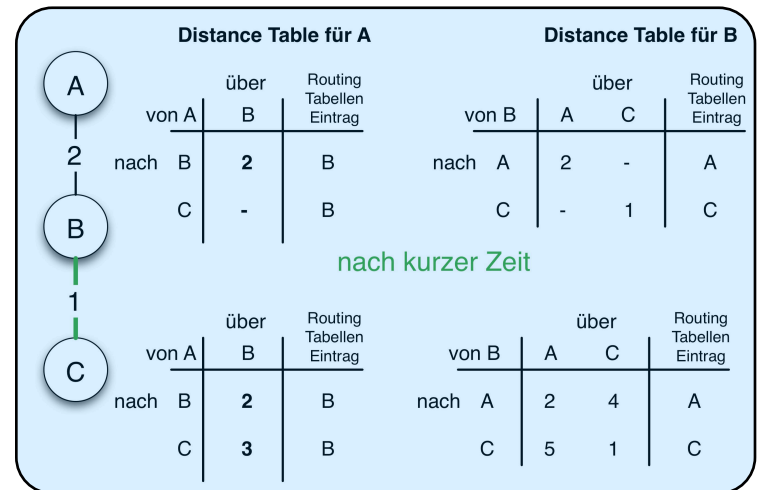
# Das "Count to Infinity" - Problem

▶ **Gute Nachrichten verbreiten sich schnell**

- Neue Verbindung wird schnell veröffentlicht

▶ **Schlechte Nachrichten verbreiten sich langsam**

- Verbindung fällt aus
- Nachbarn erhöhen wechselseitig ihre Entfernung
- "Count to Infinity"-Problem



# Link-State Protocol

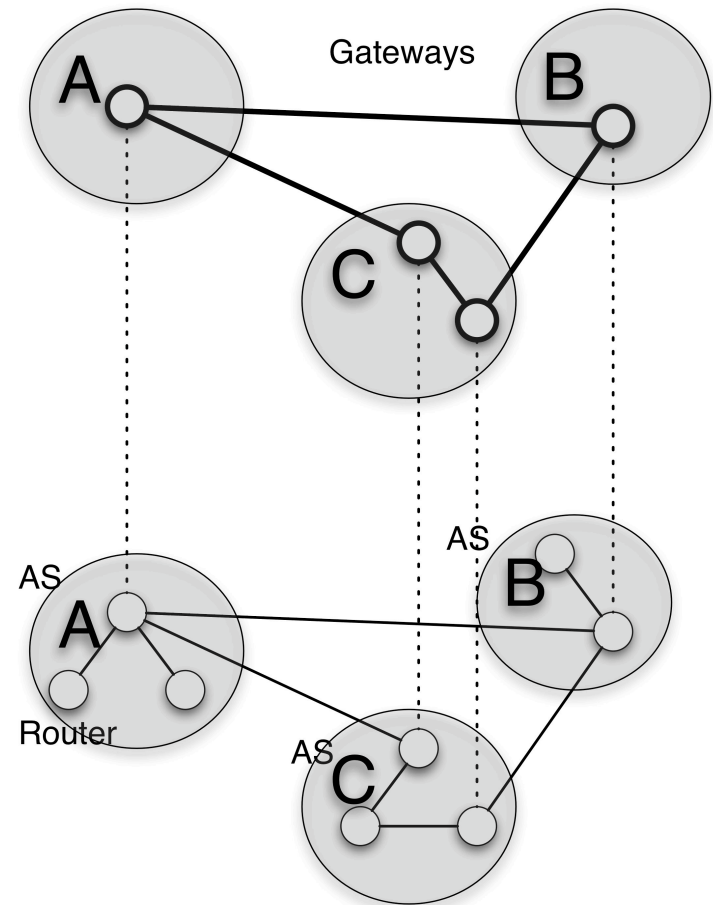
- ▶ **Link State Router**
  - tauschen Information mittels Link State Packets (LSP) aus
  - Jeder verwendet einen eigenen Kürzeste-Wege-Algorithmus zu Anpassung der Routing-Tabelle
- ▶ **LSP enthält**
  - ID des LSP erzeugenden Knotens
  - Kosten dieses Knotens zu jedem direkten Nachbarn
  - Sequenznr. (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

# Die Grenzen des flachen Routing

- ▶ **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

# AS, Intra-AS und Inter-AS

- ▶ **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



# Typen autonomer Systeme

## ▶ Stub-AS

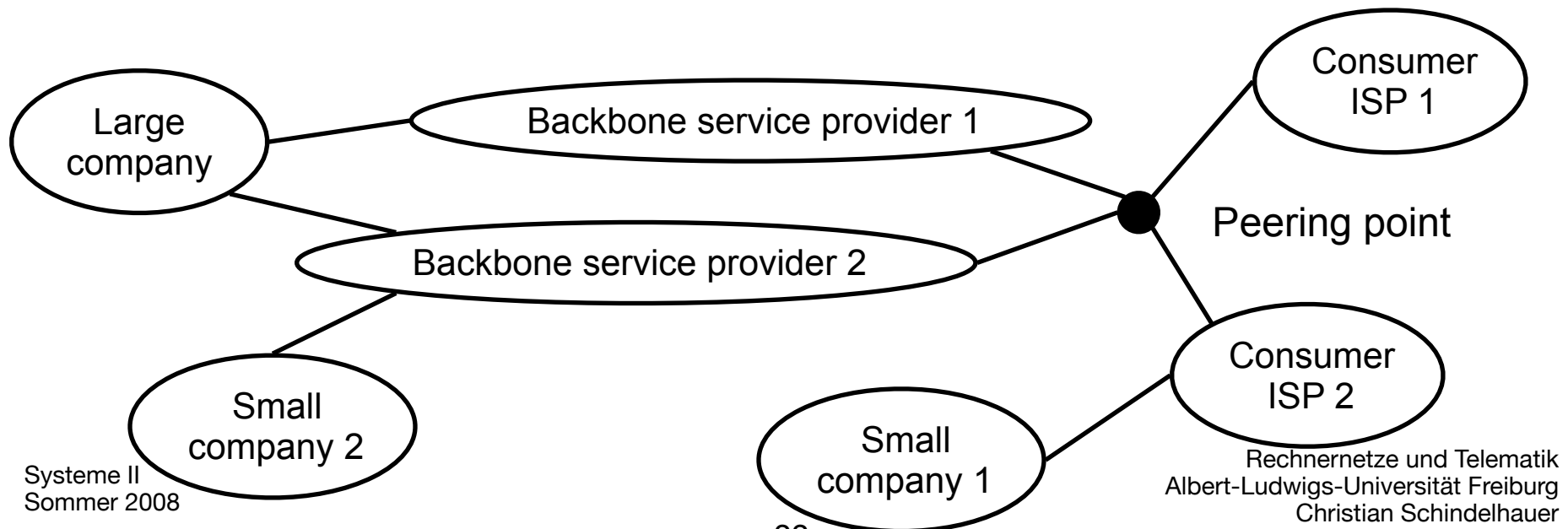
- Nur eine Verbindung zu anderen AS

## ▶ 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)



# Intra-AS: RIP

## Routing Information Protocol

- ▶ **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-Advertisement**
  - per TCP, erhöht Sicherheit (security)
  - werden in die gesamte AS geflutet
  - Mehrere Wege gleicher Kosten möglich



# Intra-AS

## Hierarchisches OSPF

- ▶ **Für große Netzwerke zwei Ebenen:**
  - Lokales Gebiet und Rückgrat (backbone)
    - Lokal: Link-state advertisement
    - Jeder Knoten berechnet nur 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

# Intra-AS: IGRP (Interior Gateway Routing Protocol)

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

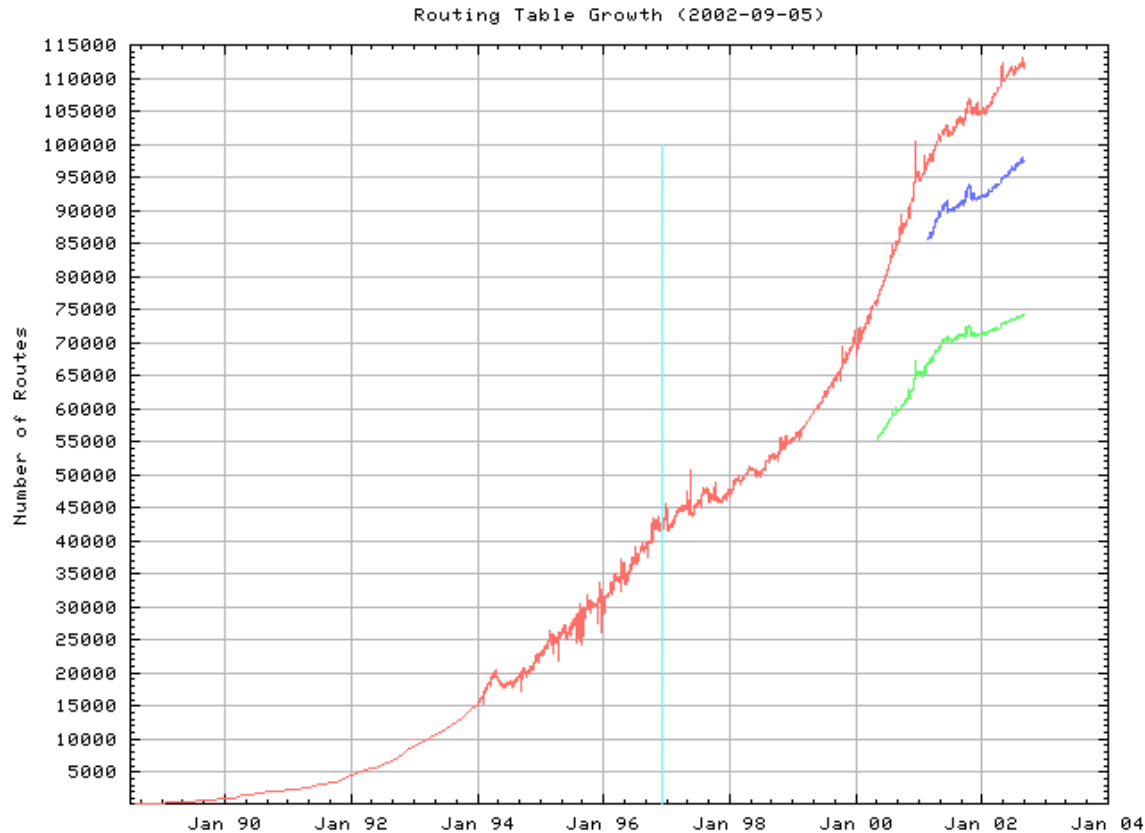
# Inter-AS-Routing

- ▶ **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 140.000 Netzwerke kennen

# Inter-AS: BGPv4 (Border Gateway Protocol)

- ▶ **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 1990-2002



<http://www.mcvax.org/~jhma/routing/bgp-hist.html>

# Broadcast & Multicast

## ▶ **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

# IP Multicast

## ► Motivation

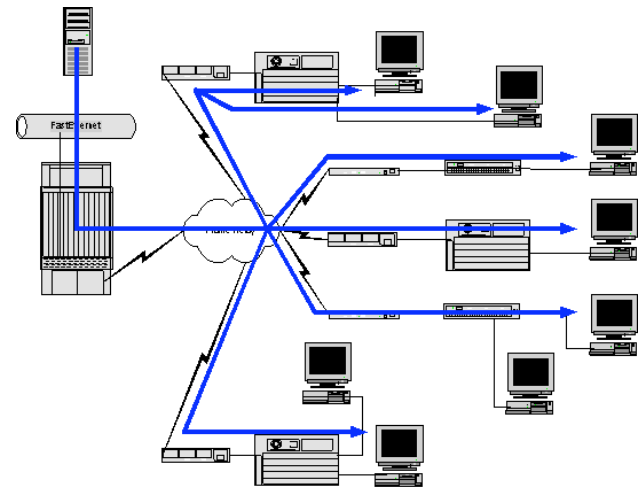
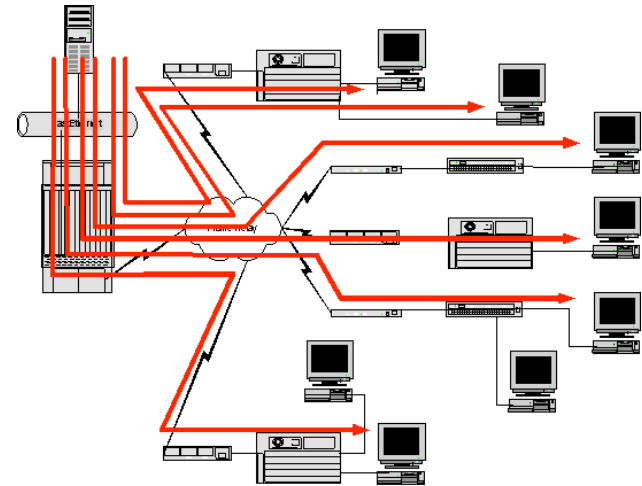
- Übertragung eines Stroms an viele Empfänger

## ► Unicast

- Strom muss mehrfach einzeln übertragen werden
- Bottleneck am Sender

## ► Multicast

- Strom wird über die Router vervielfältigt
- Kein Bottleneck mehr



# Funktionsprinzip

- ▶ **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



# Routing Protokolle

- ▶ **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-Attacken**
  - 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!)



# Systeme II

**Ende der 7. Vorlesungswoche**

Albert-Ludwigs-Universität Freiburg  
Institut für Informatik  
Rechnernetze und Telematik  
Christian Schindelhauer  
Sommer 2008