

---

# Rechnernetze

## Eine (kurze) Einführung

Cluj, Wintersemester 2019/20

**Prof. Dr.-Ing. habil. Hartmut König**

---

## III.4

# Wegsuche in WAN

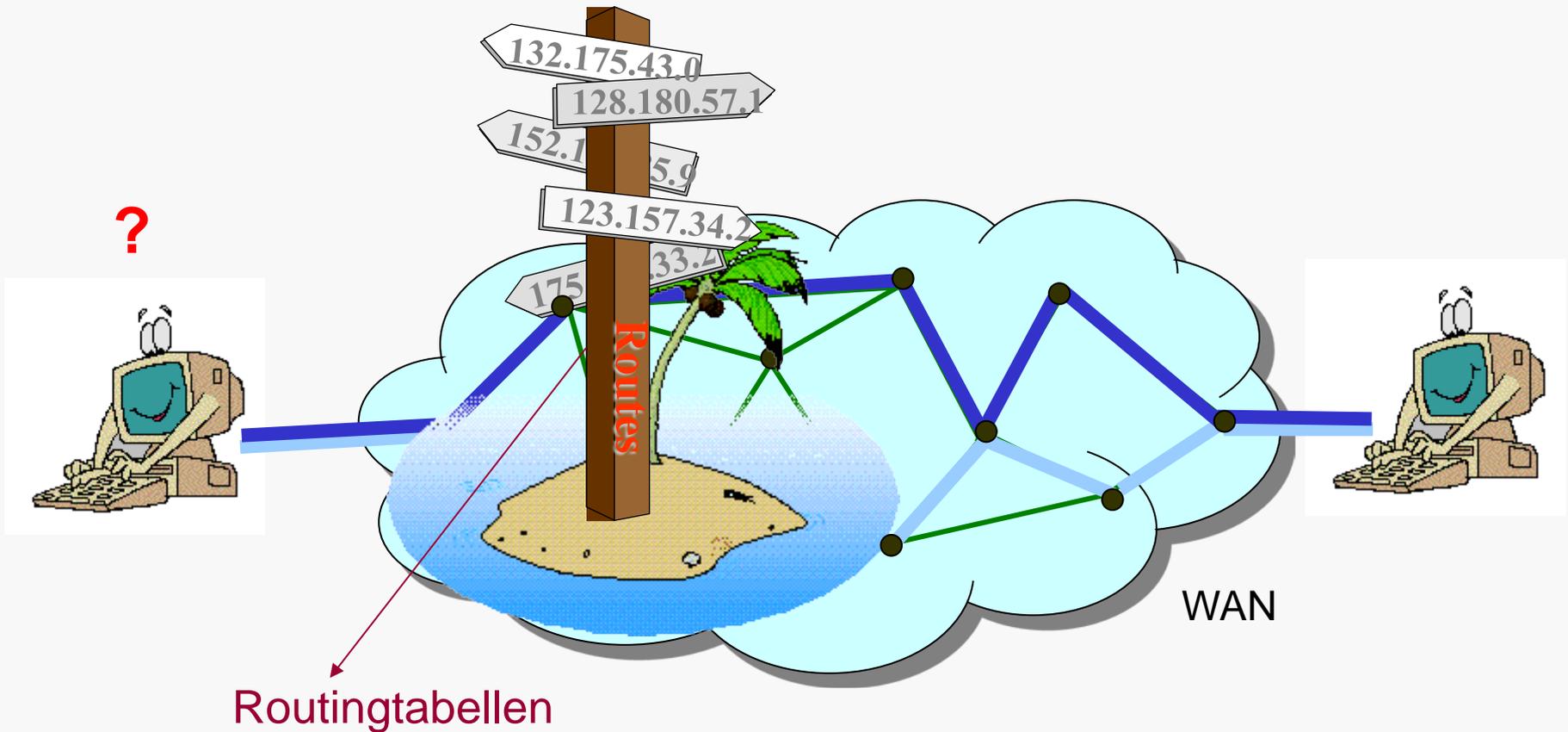
*(Routing)*



Tanenbaum / Wetherall 5.2  
Stallings 19.2



# Wie finde ich den optimalen Pfad zu meinem Partner?



# Routing-Tabelle

- **Routing-Tabellen** sind Einträge in den Netzknoten (Routern), die alle notwendigen Informationen zur Weiterleitung der PDUs/Datenpakete haben.

☞ Bei adaptiven Routing-Verfahren werden sie periodisch aktualisiert !!!

- **Beispiel:**

Zielnetz	Port	Metrik	Timer
A1	8	20	10
B3	12	75	60
...	...	...	...

Adressen erreich-  
barer Netze

Ausgangsleitung

Entfernung zum  
Zielnetz in Hops

Zeit seit letzter  
Aktualisierung



---

# Routing

Bestimmung des günstigsten Weges (*route*) für die Übertragung von Daten in Netzen mit vermaschten Topologien. Die günstigste Route wird nach vorgegebenen Routing-Metriken, z. B. Weglänge, Kosten, QoS, festgelegt.

## ● Arten des Routing:

### ● Hop-by-Hop Routing

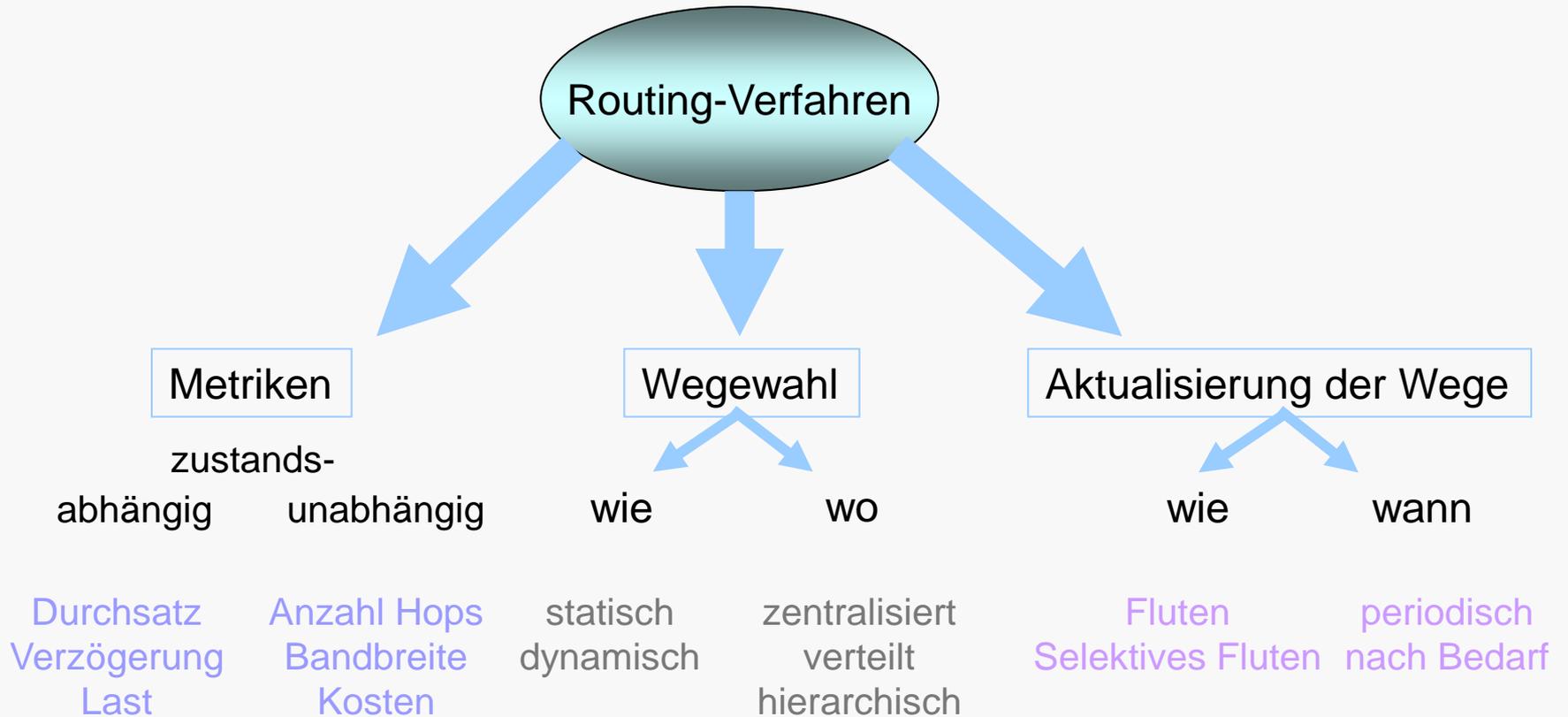
- ↪ Bestimmung des nächsten Wegabschnitts für ein Paket in jeden Netzknoten
- ↪ bevorzugte Anwendung in WAN

### ● Source Routing

- ↪ die Weginformation wird dem Paket mitgegeben

☞ Im Folgenden nur Betrachtung des Hop-by-Hop Routings !!!

# Routingverfahren



1) nach Badach, A; Knauer, O.: High Speed Internetworking. Addison Wesley, 1995

---

# Routing-Metrik

Maß für die Bewertung der Güte einer Route.

## ● Zustandsunabhängige Metriken

- Auswahl unabhängig vom aktuellen Netzzustand
  - ↳ Anzahl der Hops, Kosten, Bandbreite

## ● Zustandsabhängige Metriken

- Auswahl berücksichtigt aktuellen Netzzustand
  - ↳ unterbrochene Leitungen, Netzlast, Verzögerungen



---

# Routing-Verfahren

*Wo erfolgt die Wegewahl ?*

## ● Zentralisiertes Routing

- Berechnung der Route(n) an einen zentralen Punkt
- Verteilung des Ergebnisses an alle Router
- wurde z. B. in X.25-Datennetzen angewendet

## ● Verteiltes Routing

- Bestimmen der Route(n) dezentral in der Routern
  - ↳ heutzutage bevorzugt

## ● Hierarchisches Routing

- Bestimmen der Route(n) jeweils nur innerhalb einer bestimmten Region
  - ↳ Reduzierung der Größe der Routing-Tabellen
  - ↳ Anwendung im Internet



---

# Routing-Verfahren

*Wie erfolgt die Wegewahl ?*

## ● Statisches/nichtadaptives Routing

- Bestimmen einer Standard-Route (*default route*)
- feste Router-Konfiguration (Netzmanagement)
  - ↪ Änderungen des Netzzustands erfordern Neukonfigurationen

☞ Zweckmäßig nur bei kleinen Netzen mit festen Topologien !!!

## ● Dynamisches/adaptives Routing

- Bestimmen der Route(n) unter Berücksichtigung der aktuellen Netzsituation
  - ↪ ständige Aktualisierung der Routing-Tabellen
  - ↪ adaptive Anpassung an die Netzsituation

---

# III.4.1

## Statisches Routing



---

# A) Grundlagen



---

# Sink-Tree (1)

(*Senkenbaum*)

Ziel der Routing-Algorithmen ist der Aufbau des Senkenbaums bzw. des Sink-Trees.

## ● Sink-Tree:

- Baum, der für einen Knoten im Netz (Wurzel-Knoten) die optimalen Pfade zu allen anderen Knoten im Netz enthält.

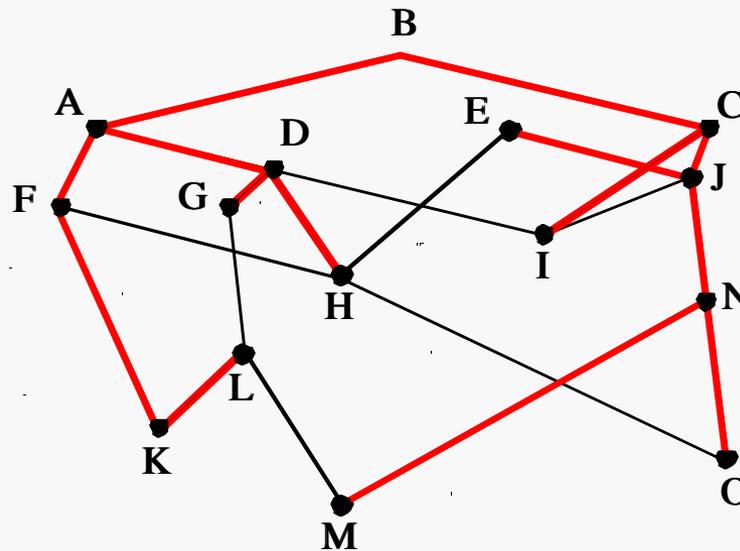
☞ **Routing-Algorithmen in Netzen müssen den Sink Tree für alle Endsysteme bestimmen !!!**

# Sink-Tree (2)<sup>1</sup>

(Senkenbaum)

● Beispiel:

Teilnetz



**Sink-Tree für B**

1) entnommen aus: Tanenbaum, A. S.; Wetherall, D.J.: Computernetzwerke. Pearson, 5. Auflage, 2012.

# Optimalitätsprinzip (1)

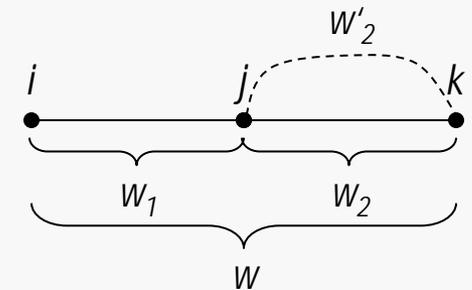
Routingalgorithmen wenden bei der Konstruktion des Sink Trees das Optimalitätsprinzip an.

## ● Optimalitätsprinzip

Wenn der aus den beiden Teilstrecken  $w_1$  von  $i$  nach  $j$  und  $w_2$  von  $j$  nach  $k$  zusammengesetzte Weg  $w$  ein optimaler Pfad von  $i$  zu  $k$  ist, dann gilt:

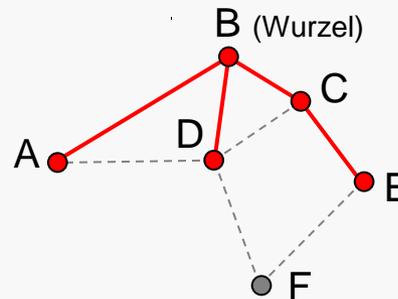
- (1)  $w_1$  ist ein optimaler Pfad von  $i$  nach  $j$ .
- (2) Jeder optimale Pfad  $w_2$  von  $j$  nach  $k$  ergibt verbunden mit  $w_1$  einen optimalen Pfad von  $i$  zu  $k$

→ z.B. auch  $w'_2$  könnte optimal sein



# Optimalitätsprinzip (2)

- **Anwendung:** Algorithmen zur Konstruktion des Sink-Trees
  - z. B. im Dijkstra-Algorithmus
  - kürzesten Wege von B zu A, C, D und E seien bereits bekannt
    - ↪ einer der beiden Wege  $B - D - F$  oder  $B - C - E - F$  muss optimal sein  
( $B - A - D - F$  kann nicht besser sein !)



---

# Häufig verwendete statische Routingalgorithmen

- Shortest Path Algorithmus
- Fluten



---

# A) Shortest Path Algorithmus



---

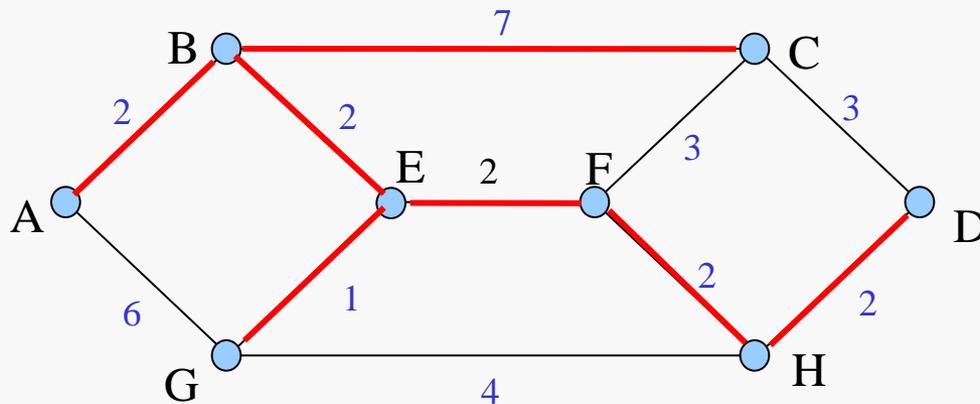
# Shortest Path Algorithmus (1)

- Dijkstra, 1959
- häufig genutzt im Netzrouting
- **Voraussetzung:** Graph der Netztopologie mit Wichtung der Teilstrecken
- **Mögliche Metriken:**
  - Zahl der Teilstrecken (Hops)
  - Bandbreite
  - durchschnittliche Verkehrsbelastung
  - Warteschlangenlängen
  - Übertragungsverzögerungen (Testmessungen)
  - Übertragungskosten

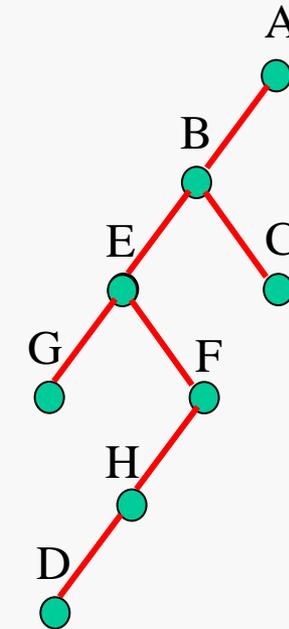
☞ **Durch Änderung der Wichtungsfunktion werden die Auswahlentscheidungen geändert !!!**

# Shortest Path Algorithmus (2)<sup>1</sup>

## Beispielnetz



 [Shortest Path Algorithmus](#)



Sink Tree für A

1) entnommen aus: Tanenbaum, A. S.; Wetherall, D.J.: Computernetzwerke. Pearson, 5. Auflage, 2012.

---

## B) Fluten



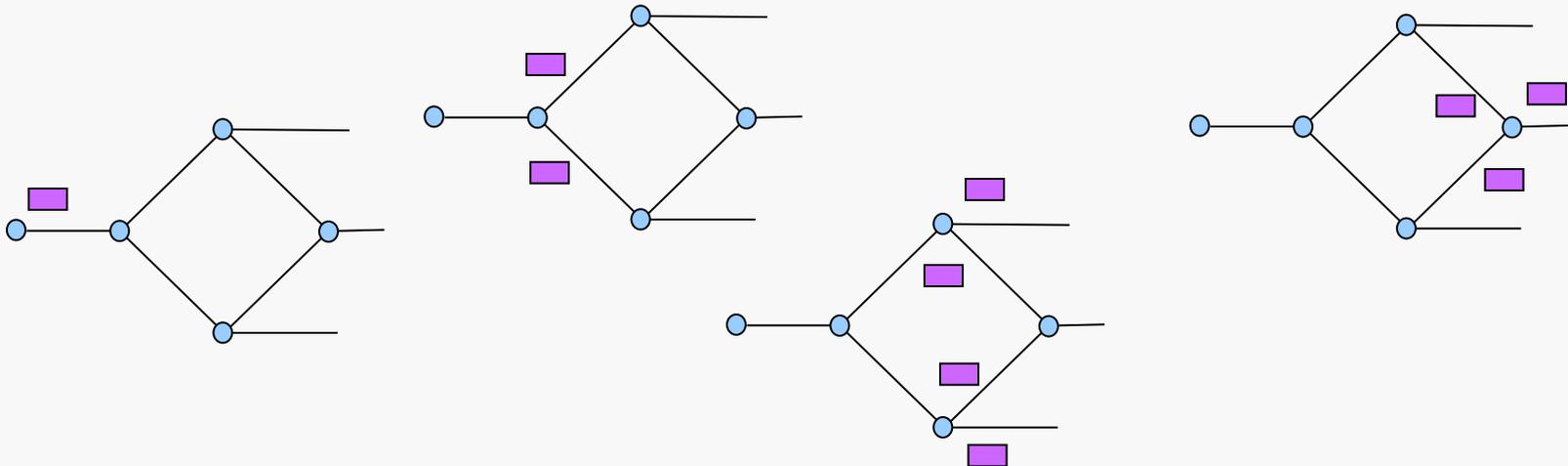
Tanenbaum / Wetherall 5.2.3

# Fluten (1)

(Flooding)

Fluten ist ein spezielles Routing-Verfahren, bei dem jedes ankommende Datenpaket auf alle Leitungen ausgegeben wird außer auf die Leitung, auf der es angekommen ist.

- ein Packet findet immer den kürzesten Pfad
- Pakete erreichen auch ihr Ziel bei Ausfall von Teilstrecken



---

# Fluten (2)

## (Flooding)

- **Vorteile:**
  - einfach
  - robust
- **Nachteile:**
  - Vervielfältigung von Paketen
  - Netzbelastung
- Gegenmaßnahmen
  - Time-to-live-Zähler
  - Selektives Fluten
- **Einsatzfelder:**
  - Austausch von Routing/Signalisierungsdaten
  - Ad hoc-Netze/Sensornetze



---

## III.4.3.2

# Adaptives Routing



---

# Adaptives Routing

Adaptive Routing-Verfahren passen sich dynamisch an die aktuelle Situation im Netz (Topologie, Netzlast) an.

Die wichtigsten Algorithmen für das adaptive Routing in WAN sind:

- **Distance Vector Routing**

  - ↳ auch Bellman-Ford- bzw. Ford-Fulkerson-Algorithmus (1957/1962)

- **Link State Routing**



---

# A) Distance Vector Routing



Tanenbaum / Wetherall 5.2.4

---

# Distance Vector Routing (1)

Jeder Router führt eine Routing-Tabelle, die für jeden im Teilnetz vorhandenen Router einen Eintrag enthält und nach diesen Einträgen indexiert wird.

- **Eintrag:** - bevorzugte Ausgangsleitung
  - geschätzte „Entfernung/Distanz“ zum Ziel
- **Angewandte Metrik:** - Zahl der Teilstrecken (Hops)
  - Zeitverzögerung in ms
    - ↳ Messung durch ECHO-Pakete
  - oder andere (vgl. Shortest Path)
    - ⇒ „Entfernung“ zu diesem Ziel

**Es wird angenommen, dass jeder Router die ‘Entfernung’ zu seinem Nachbarn kennt !!!**

---

# Distance Vector Routing (2)

## ● Algorithmus:

- Jeder Router sendet periodisch alle  $T$  ms eine Liste mit den geschätzten Entfernungen zu allen Knoten an seine Nachbarn.

↳ Distance Vector

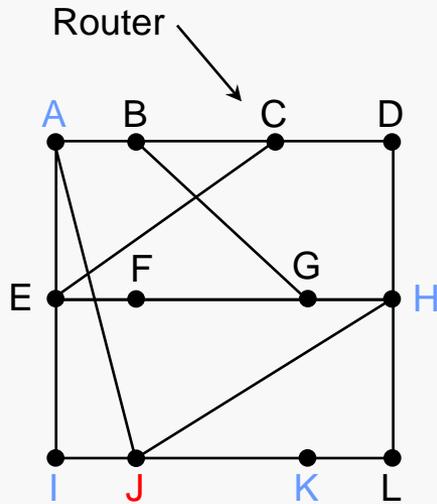
- Berechnung aller Wege und Auswahl des optimalen Weges
- Aktualisierung Routing-Tabelle

## ● Beispiel:

- Router Y erhält die Liste seines Nachbarn X
- $X_1$  sei die geschätzte „Entfernung“ von X zum Router I
- Entfernung  $Y \rightarrow X$ :  $k$  Einheiten
- geschätzte Entfernung zu I:  $X_1 + k$



# Distance Vector Routing<sup>1</sup>



(a)

An	A	I	H	K
A	0	24	20	21
B	12	36	31	28
C	25	18	19	36
D	40	27	8	24
E	14	7	30	22
F	23	20	19	40
G	18	31	6	31
H	17	20	0	19
I	21	0	14	22
J	9	11	7	10
K	24	22	22	0
L	29	33	9	9

	JA	JI	JH	JK
Distanz				
ist	8	10	12	6

Empfangene Vektoren von J's vier Nachbarn

Neu geschätzte Verzögerung von J

↓ Leitung

8	A
20	A
28	I
20	H
17	I
30	I
18	H
12	H
10	I
0	-
6	K
15	K

Neue Routing-Tabelle für J

(b)

 [Animation Distance Vector Routing](#)

1) entnommen aus: Tanenbaum, A. S.; Wetherall, D.J.: Computernetzwerke. Pearson, 5. Auflage, 2012.

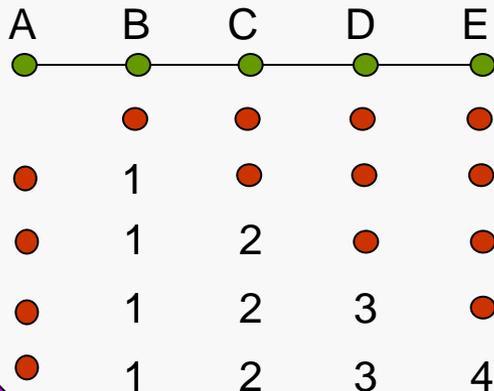
# Count-to-Infinity Problem

## ● Problem:

● Algorithmus gelangt zur richtigen Lösung, ist aber zu langsam

↪ reagiert schnell auf gute Nachrichten, aber zu langsam auf schlechte

☞ *Count-to-Infinity*



Verbindung zu A bricht zusammen

A	B	C	D	E
●	●	●	●	●
	1	2	3	4
	3	2	3	4
	3	4	3	4
	5	4	5	4
	5	6	5	6
	7	6	7	6
	7	8	7	8
	...	...	...	...

anfangs

nach 1. Austausch

nach 2. Austausch

nach 3. Austausch

nach 4. Austausch

A schaltet sich neu zum Netz

1) entnommen aus: Tanenbaum, A. S.; Wetherall, D.J.: Computernetzwerke. Pearson, 5. Auflage, 2012.

---

# Count-to-Infinity Problem

## ● Mögliche Gegenmaßnahmen

### ● Split Horizon

↪ Pfadinformation darf nicht über dasselbe Interface veröffentlicht werden, über das sie empfangen wurde

### ● Triggered Updates

↪ nicht nur periodische Updates, sondern auch wenn ein Router herausfindet, dass ein Nachbar nur noch schwer oder gar nicht mehr erreichbar ist  
→ schnelle Weitergabe neuer Informationen



---

## B) Link State Routing



Tanenbaum / Wetherall 5.2.5

---

# Link State Routing (1)

Das **Link State Routing** wurde 1979 als Alternative zum Distance Vector Routing in das ARPANET eingeführt.

- Das Link State Routing umfasst 5 Schritte, die sich grob in zwei Phasen unterteilen lassen:
  - Austausch von Daten zur Situation im Netz (Schritte (1) - (4))
    - ↳ Jeder Router baut ein aktuelles „Bild“ von der Netztopologie auf
  - Jeder Router berechnet seinen Sink Tree (Schritt (5))
    - ↳ Verteiltes Routing
- **Erfahrung:** gutes Anpassungsverhalten

---

# Link State Routing (2)

## ● Schritte:

### (1) Entdecken der Nachbarn und Feststellen ihrer Netzadressen

- ↪ Aussenden von Hello-Paketen auf jeder Leitung
- ↪ Nachbarn antworten (→ eindeutige Adressen erforderlich)
- ↪ spez. Problem: mehrere Router in einem LAN

### (2) Bestimmung der Verzögerung/Kosten zu seinen Nachbarn

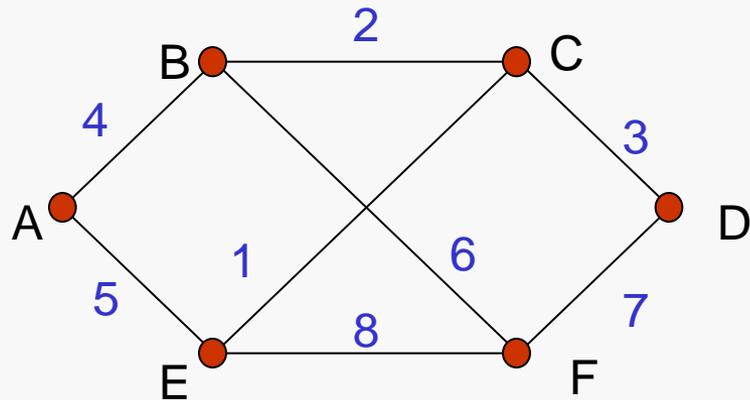
- ↪ Jeder Router muss die Verzögerung zu seinem Partner kennen
- ↪ z.B. Messung mittels ECHO-Paketen
- ↪ Varianten mit/ohne Berücksichtigung der Last

### (3) Zusammenstellen eines Link State Pakets

- ↪ enthält die Liste der Nachbarn mit Übertragungsverzögerungen
- ↪ Zeitpunkt der Erstellung
  - periodisch
  - ereignisorientiert (neue Leitung, Ausfall Router usw.)



# Link State Routing (3)



## Link State Pakete

A	
Sequ.nr	
TTL	
B	4
E	5

B	
Sequ.nr	
TTL	
A	4
C	2
F	6

C	
Sequ.nr	
TTL	
B	2
D	3
E	1

D	
Sequ.nr	
TTL	
C	3
F	7

E	
Sequ.nr	
TTL	
A	5
C	1
F	8

F	
Sequ.nr	
TTL	
B	6
D	7
E	8

1) entnommen aus: Tanenbaum, A. S.; Wetherall, D.J.: Computernetzwerke. Pearson, 5. Auflage, 2012.

---

# Link State Routing (3)

## (4) Senden des Pakets an alle Router

↪ Verteilen der Pakete durch Fluten

- ➔ Sequenznummern zeigen die Version/Aktualität
- ➔ TTL verhindert ein Weiterleiten veralteter Pakete
- ➔ kurze Verzögerungen beim Senden, um möglicherweise noch aktuellere Pakete weiterzuleiten

## (5) Berechnen des kürzesten Pfades zu allen anderen Routern

↪ Aufbau des Netzgraphen, wenn ein Router alle Link State Pakete gesammelt hat

- ➔ Verbindungen werden zweimal dargestellt (je 1x pro Richtung, unterschiedliche Wichtungen möglich)

↪ Anwendung [Shortest Path Algorithmus](#)

☞ **Link State Routing zeigt in der Praxis ein gutes Verhalten !!!**



---

# III.4.3.3

## Hierarchisches Routing



Tanenbaum / Wetherall 5.2.6

---

# Hierarchisches Routing

In großen Netzen spielen der Bedarf an Speicher und Berechnungszeit eine große Rolle.

## ● Problem:

- Größe der Routingtabellen
- Aktualisierungszeit
- Lösung: hierarchisches Routing

## ● Aufteilung in Regionen

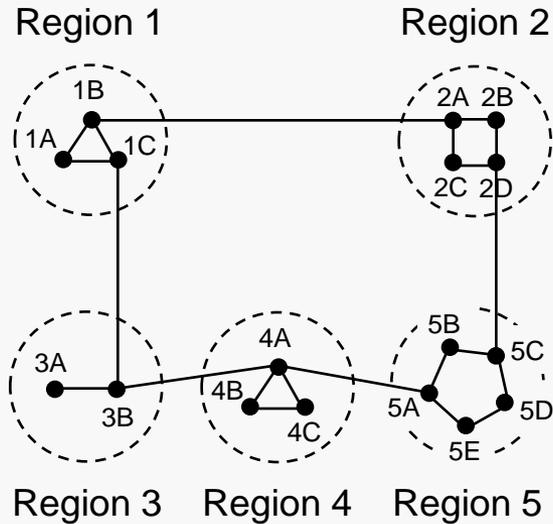
- Router kennt jeweils die Wege in seiner Region
- in jeder Region gibt es einen Router, der Grenz-Router/Border-Router, der überregionalen Datenaustausch bearbeitet
  - ↳ nicht immer der optimale Weg für alle Verbindungen

## ● Mehrstufige Hierarchie möglich

- Regionen → Cluster → Zonen
- optimale Anzahl von Stufen in einem Netz mit  $N$  Routern:  $\ln N$ 
  - ↳ Kamoun und Kleinrock



# Hierarchisches Routing<sup>1</sup>



(a)

Vollständige Tabelle für 1A

Ziel Leitung Teil-  
strecken

1A	-	-
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

Hierarchische Tabelle für 1A

Ziel Leitung Teil-  
strecken

1A	-	-
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

1. entnommen: Tanenbaum, A. S., Wetherall D.J.: Computernetzwerke, Pearson, 5. Auflage, 2012.

---

# III.5

## Multicast



Tanenbaum / Wetherall 5.2.8  
Stallings 19.1

---

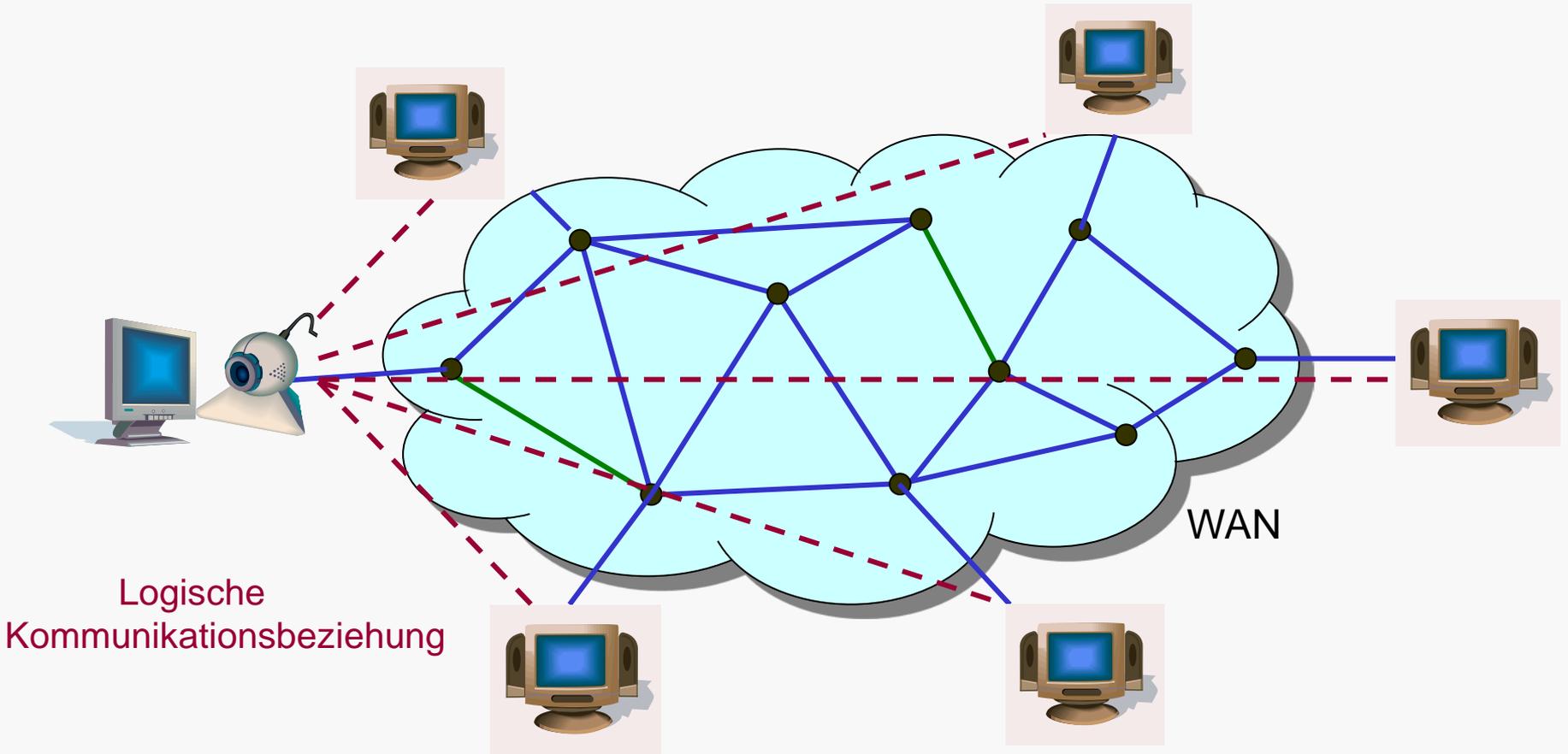
# 1:N-Kommunikationsbeziehung

Die traditionelle Datenkommunikation konzentrierte sich auf Punkt-zu-Punkt-Übertragungen. Einige Anwendungen, insbesondere in der multi-medialen Kommunikation, erfordern eine 1:N-Kommunikation.

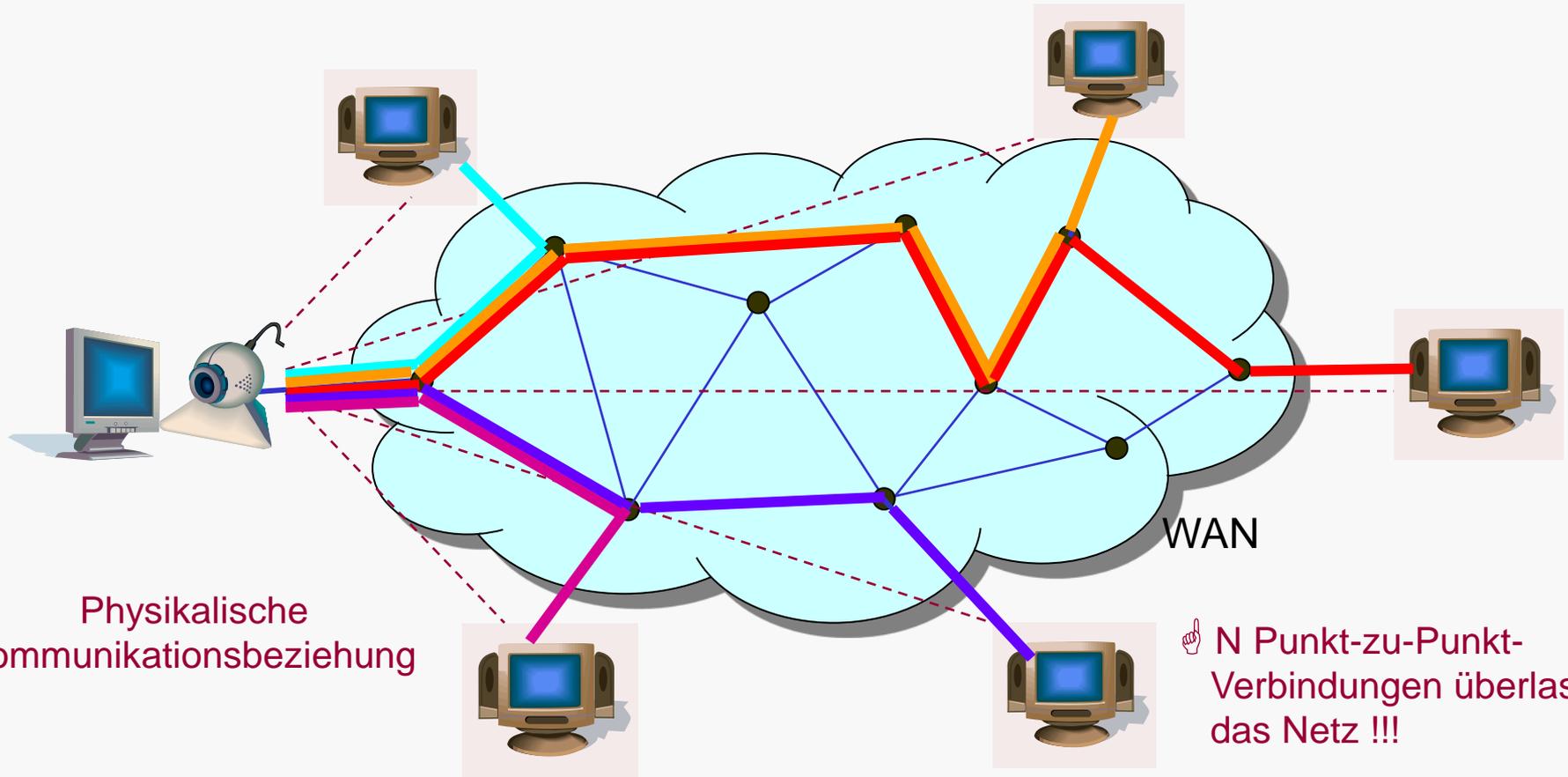
## ● Beispiele:

- Verteildienste
  - ↪ Audio/Videostreaming
    - z.B. Video on Demand, Internetradio
- (offene) Videokonferenzen
  - ↪ Teleteaching-Übertragungen
  - ↪ Konferenz/Vortrags-Übertragungen
- Telekooperations-Anwendungen

# 1:N-Kommunikationsbeziehung



# N Punkt-zu-Punkt-Verbindungen



Physikalische  
Kommunikationsbeziehung

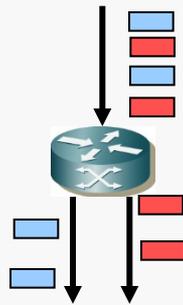
👉 N Punkt-zu-Punkt-  
Verbindungen überlasten  
das Netz !!!

---

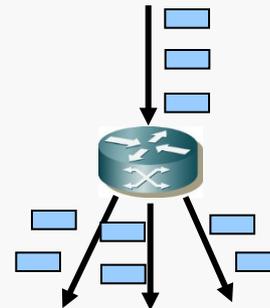
# Multicast

**Multicast** ist ein Kommunikationsparadigma in Netzen für eine 1:N-Kommunikationsbeziehung, bei dem die Daten auf gemeinsam durchlaufenen Wegen nur einmal ausgesandt werden und erst bei Trennung des Weges vervielfältigt werden.

👉 Voraussetzung: Multicastfähige Router !!!

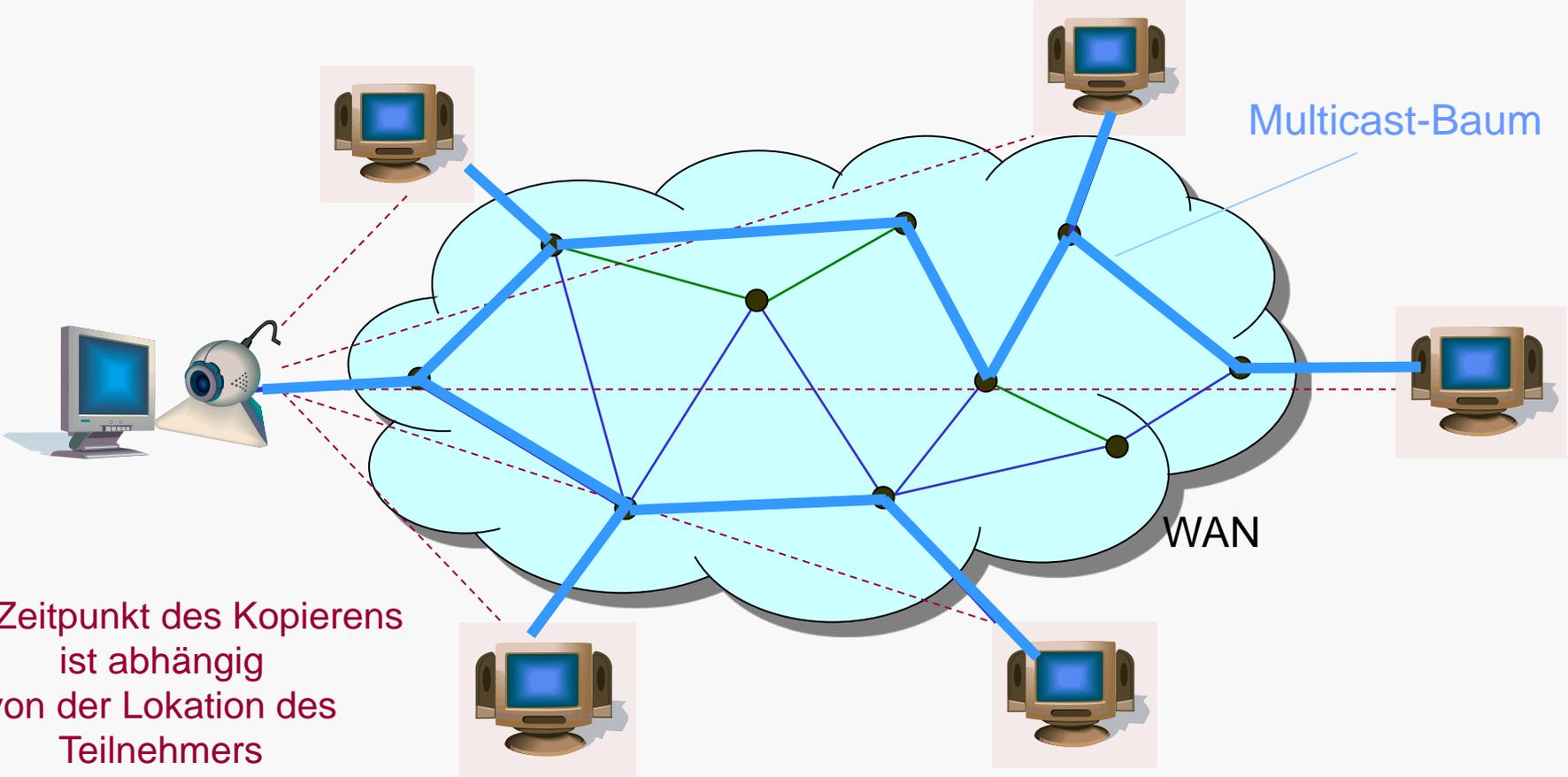


Router



Multicast-Router

# Multicast



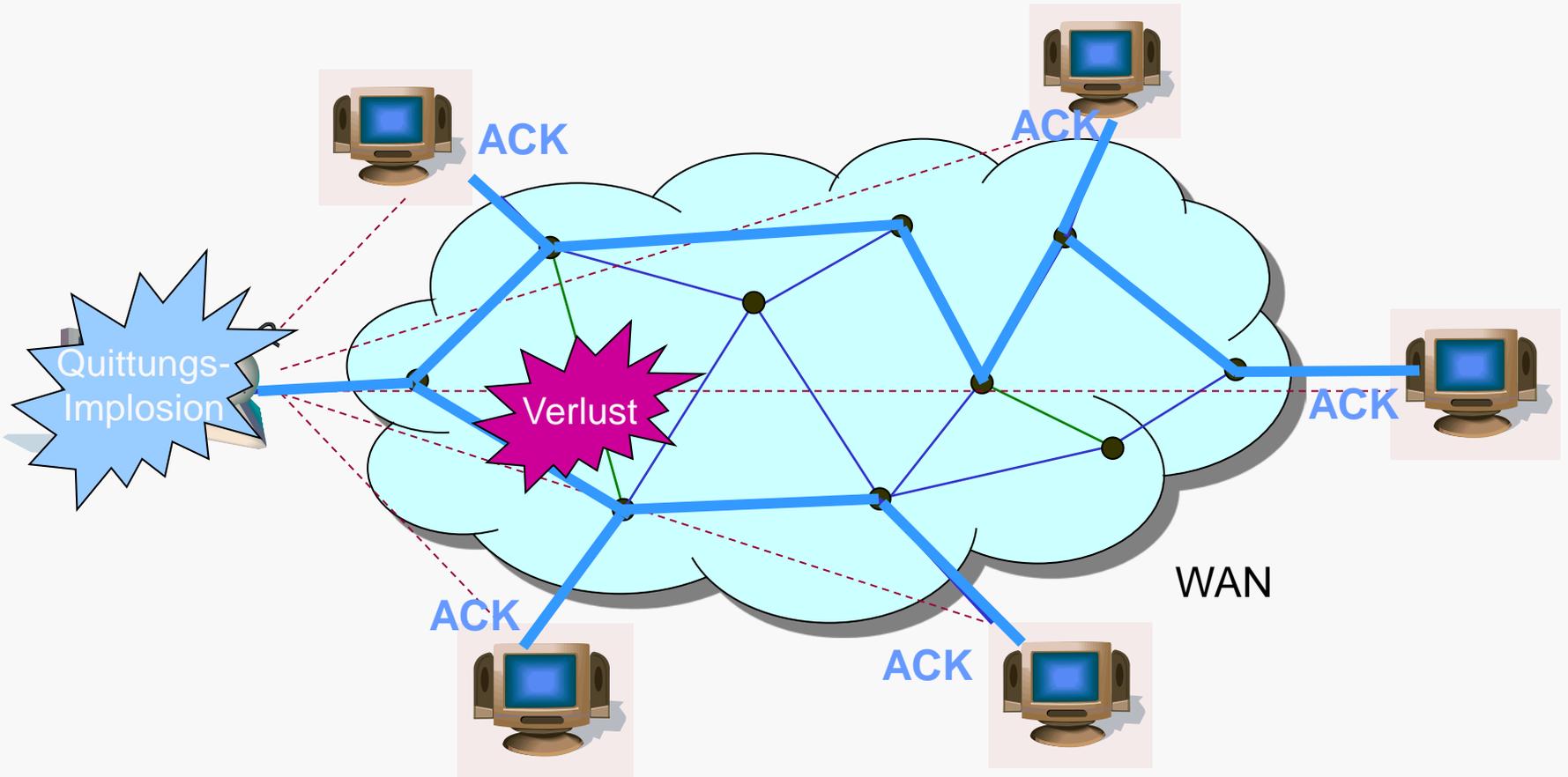
⏰ Zeitpunkt des Kopierens  
ist abhängig  
von der Lokation des  
Teilnehmers

---

# Probleme der Multicast-Übertragung

- Zuverlässigkeit der Übertragung
- geordnete Übertragung
- Multicast-Routing
  - ↪ hier nicht betrachtet
- QoS-Garantien

# Quittungs-Implosion und -Verlust



---

# Zuverlässigkeitsklassen

## ● Unzuverlässige Kommunikation

- keine Neuübertragung verlorener oder fehlerhafter PDUs

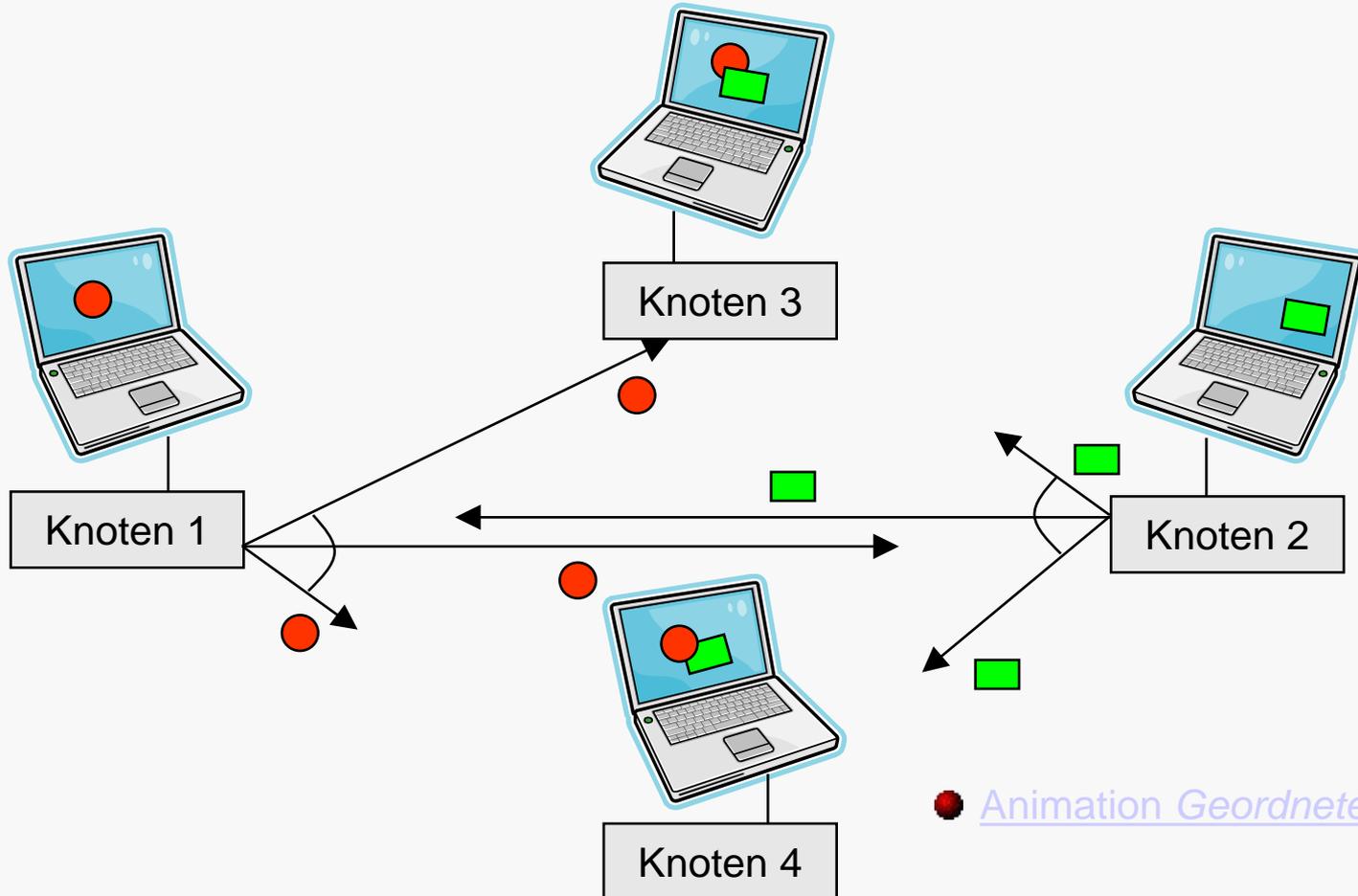
## ● Halb-zuverlässige Kommunikation

- *statistisch zuverlässiges Multicasting*
  - ↪ Neuübertragung in einem definierten Schwellwert
- *k-zuverlässiges Multicasting*
  - ↪ garantierte Übertragung für  $k (< m)$  Empfänger

## ● Zuverlässige Kommunikation

- garantierte Übertragung
  - ↪ aufwendig

# Problem: Geordnete Multicast-Übertragung



---

# Multicast-Nutzung

Multicast hat bisher nicht die hohen Erwartungen erfüllt, die an seine Entwicklung geknüpft wurden.

- **Gründe:** - Anpassung der Router (M-Router) erforderlich
  - Abrechnung der Dienste nicht geklärt (Geschäftsmodell)

☞ **Multicast wird von vielen ISPs nicht unterstützt !!!**

☞ **Kein breiter Einsatz in absehbarer Zeit zu erwarten !!!**

- **Alternative:** Verlagerung von Multicast in die Anwendungen
  - ☞ Overlay-Multicast