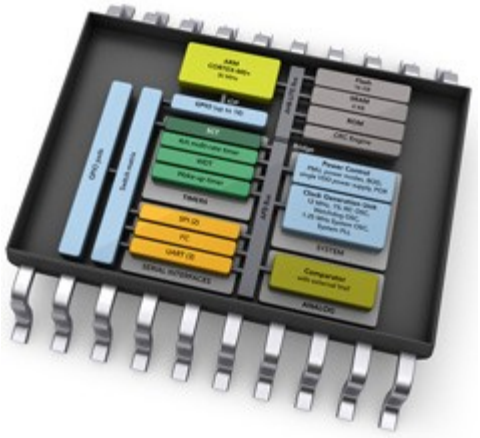
FH AACHEN
UNIVERSITY OF APPLIED SCIENCES

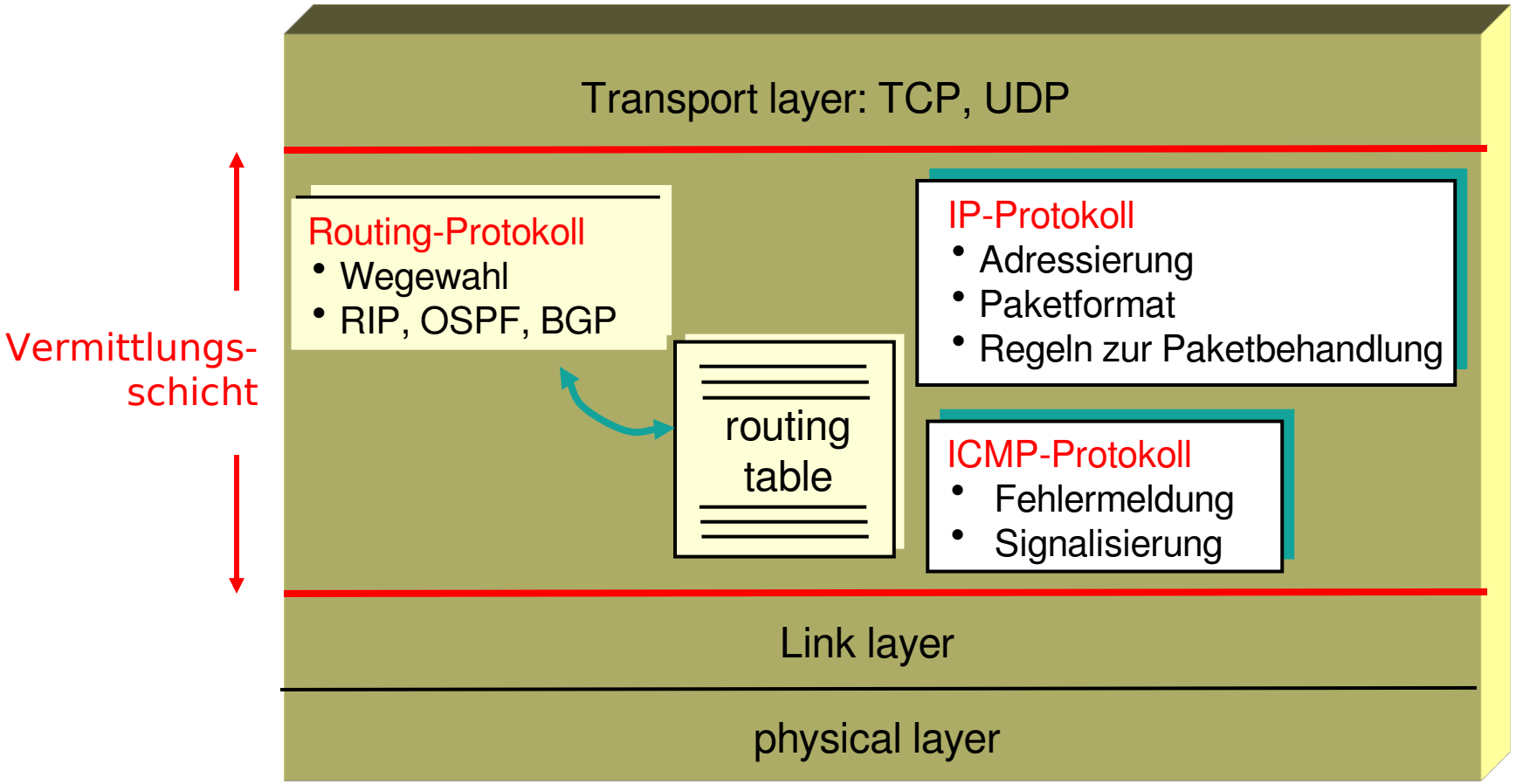
Kommunikationssysteme

(Modulcode 941306)

Prof. Dr. Andreas Terstegge



Die Vermittlungsschicht im Internet



Routing im Internet

„Das Weiterleiten (Routing) erfüllt die wichtige Aufgabe, einzelne Teilstrecken des Kommunikationsnetzes so zu verbinden, dass eine Ende-zu-Ende-Kommunikation zwischen den angeschlossenen Teilnehmern möglich wird.“

Das Routing kann in zwei Teilprozesse zerlegt werden:

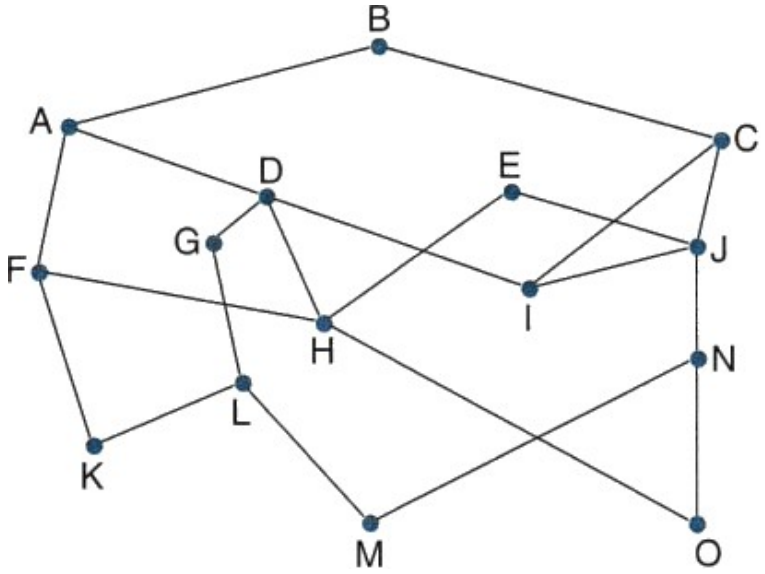
- Das eigentliche **Routing** (die Control-Plane) ist verantwortlich für Wegewahl, mit der die Ende-zu-Ende-Kommunikation erreicht wird

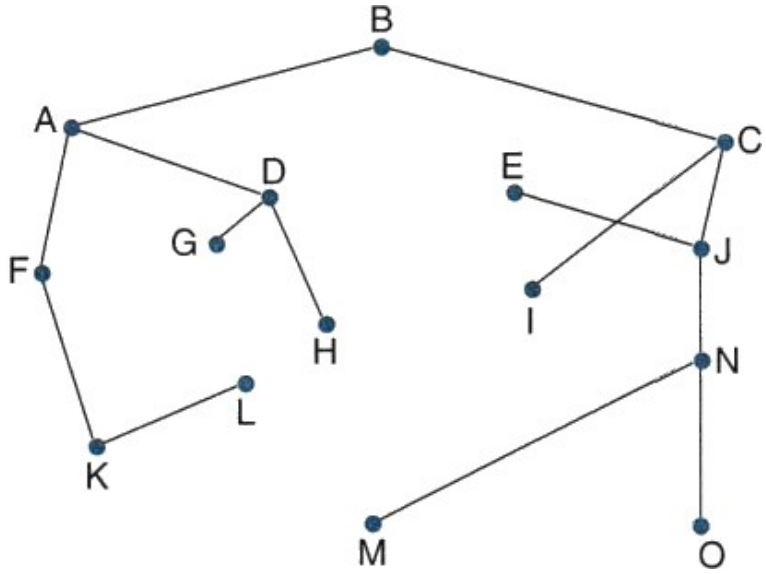
- Das **Weiterleiten** (engl. *forwarding*) der Pakete (die Data-Plane) gemäß den Vorgaben der Control-Plane

  Die Forwarding-Entscheidung wird für jedes Paket durchgeführt
  - Unabhängig von vorherigen Entscheidungen
  - Entscheidung auf Basis der Zieladresse

Das Optimalitätsprinzip

- Das Internet ist im weitesten Sinne ein Graph
  Knoten: Netzwerk-Elemente (Rechner/Router etc.)
  Kanten: Existierende physikalische Verbindungen
- Gesucht wird der (eindeutige) **Quelle-Senke-Baum** mit den ‚kürzesten' Verbindungen von beliebigen Quellen (Blätter) zum Ziel (Wurzel)





Routing – Kostenfunktion eines Weges

Was bedeutet eigentlich ‚kürzester Weg‘?

- Geringste Anzahl ‚hops‘ ?
- Schnellste Verbindung ?
- Physisch kürzeste Verbindung?
- Maximierung des Datendurchsatzes ?

Es gibt noch weitere Qualitätsmaße für ein ‚gutes‘ Routing, die sich ggf. widersprechen:

- korrekt, einfach, robust, stabil, fair, effizient

Für die weiteren Folien nehmen wir ein quantifizierbares **Gewicht** für jede Kante an, das sich i.d.R. nicht kurzfristig ändert.

Lösung des Optimierungsproblems: Dijkstra Algorithmus

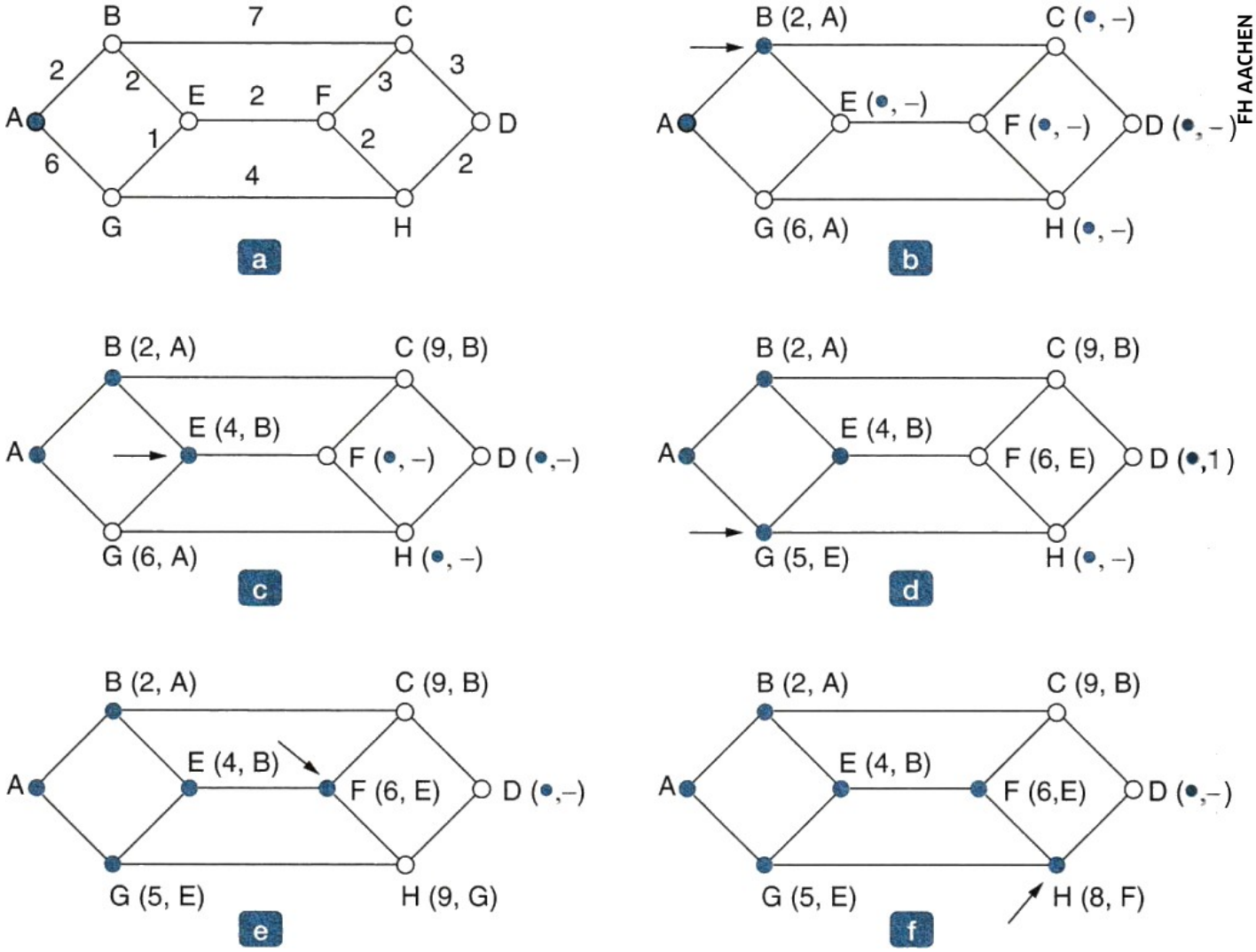
Ansatz:

- Ausgehend von der Quelle wird mit jedem Schritt ein neuer Knoten hinzugenommen
- Die Auswahl des Knotens geschieht so, dass der Knoten selektiert wird, der mit die günstigste Anbindung erlaubt
- Damit teilt sich der Graph in zwei Teile auf:
  - Eine Menge N‘ von Knoten für die der günstigste Weg schon ermittelt wurde
  - Eine Menge von Knoten, für die der Weg noch zu ermitteln ist
- Ordnung innerhalb der direkten Verbindungen zwischen diesen zwei Mengen bringt die Lösung
  - Auswahl des kleinsten Wertes
  - Es kann keinen günstigeren Weg geben

Lösung des Optimierungsproblems: Dijkstra Algorithmus (‚Shortest Path')

Der ‚kürzeste' Weg von A nach D soll berechnet werden.

Hier: Die ersten 6 Iterationen



Routing im realen Kontext

Verschicken von Paketen

- Jeder Rechner und jeder Router hat eigene Weiterleitungstabelle
  > Wird bei der Versendung jedes einzelnen Paketes konsultiert
  > Zieladresse im IP-Paket bestimmt, welcher Weg gewählt wird
- Kann statisch konfiguriert werden (z.B. im eigenen lokalen Netz via DHCP oder mittels Kommandos)
- Router im Backbone sollen auf Änderungen im Netz reagieren können
  > Statische Konfiguration unmöglich
  > **Wie kommen die Router dynamisch zu ihren Tabellen?**

→ **Routing-Protokolle**

- Router unterhalten sich und tauschen Informationen über Verbindungen im Netz aus!

Routing im realen Kontext

Zwei große Klassen von Routing-Verfahren:

- ohne Kenntnis der gesamten Netztopologie
  → **Distanzvektoralgorithmen**
- mit Kenntnis der gesamten (zumindest im AS) vorhandenen Netztopologie
  → **Link-State Routing**

- Innerhalb der AS werden heute Routing-Protokolle eingesetzt, die Link-State-Routing realisieren.
- Bei den Inter-AS Protokollen kommen auch noch Distanzvektoralgorithmen zum Einsatz (Verbergen der Struktur des AS).

Distanzvektoralgorithmen

Grundidee:

- Jeder Router schickt regelmäßig (z.B. alle 30s) seinen Nachbarn einen Distanzvektor, der die (geschätzten) Kosten zu allen anderen Knoten im Netz enthält.
- Jeder Router aktualisiert daraufhin seine eigene Liste mit den Kosten zu den anderen Knoten und wählt dabei den Weg mit den geringsten Kosten

→ **Bellman-Ford-Algorithmus**

- Grundlage des Routing im ARPANET und im Internet im Protokol RIP (Routing Information Protocol) verwendet

Bellman-Ford Algorithmus

Definiere

$c_x(v)$:= Kosten zum Nachbarn v von x aus

$d_x(y)$:= Kosten des günstigsten Weges von x nach y

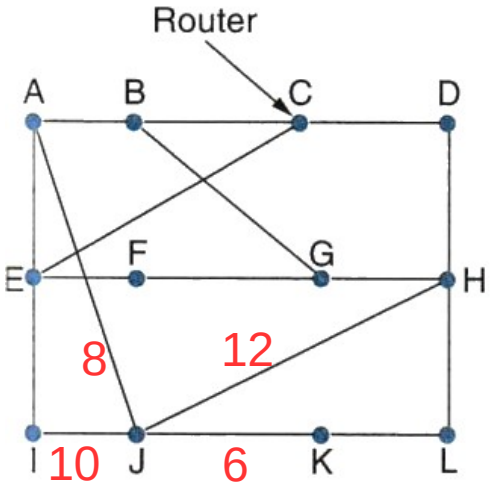
Gibt es einen Weg zwischen x und y und sind x und y nicht direkt verbunden, dann muss es einen Nachbarn v von x geben für den gilt:

$$d_x(y) = \min \{c_x(v) + d_v(y) \}$$

hierbei wird das Minimum über alle Nachbarn von x genommen.

Die Kosten der direkten Verbindung wird als bekannt angenommen

Distanzvektoralgorithmen



- Betrachtet wird Router J.
- Er kennt die Kosten (z.B. ms) zu den Routern A, I, H und K
- Er erhält Übertragungs-vektoren von A, I, H und K

zu	A	I	H	K	neu geschätzte Verzögerung von J: Leitung
A	0	24	20	21	8 A
B	12	36	31	28	20 A
C	25	18	19	36	28 I
D	40	27	8	24	20 H
E	14	7	30	22	17 I
F	23	20	19	40	30 I
G	18	31	6	31	18 H
H	17	20	0	19	12 H
I	21	0	14	22	10 I
J	9	11	7	10	0 -
K	24	22	22	0	6 K
L	29	33	9	9	15 K
	JA-Ver-zöge-rung ist 8	JI-Ver-zöge-rung ist 10	JH-Ver-zöge-rung ist 12	JK-Ver-zöge-rung ist 6	neue Routing-Tabelle für J

von den vier Nachbarn von J erhaltene Vektoren

Distanzvektoralgorithmen

Vor- und Nachteile des Distanzvektoralgorithmus

Vorteile:

- Leicht zu implementieren, einfache Berechnung
- Neue, bessere Routen werden im Netz schnell propagiert

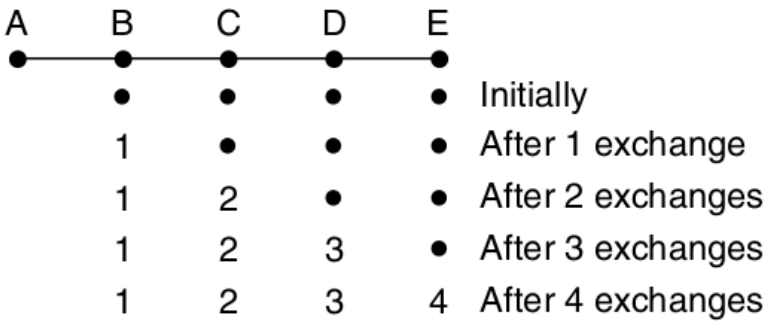
Nachteile:

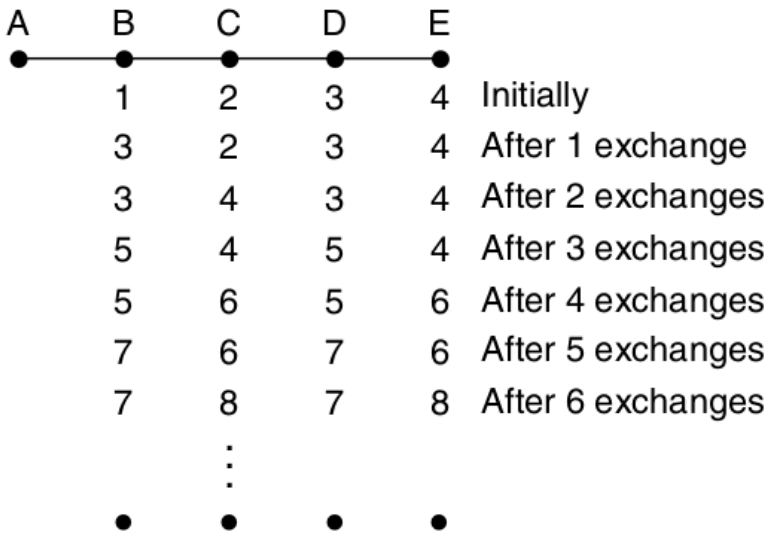
- Der Ausfall von Routen/Routern führt zum „*Count-to-Infinity*“ Problem:

  A – **B** – C

  Betrachte B: Router A fällt aus, bekommt aber von C gesagt, dass er einen Weg nach A kennt (der allerdings über B führt, was B nicht weiß…) → Die Kosten zum Weg nach A schaukeln sich langsam auf...

Distanzvektoralgorithmen: Count to Infinitiy





Router A ist initial ‚down‘,
und dann ‚up‘

→ Neue Routen werden schnell gelernt und propagiert.

Router A ist initial ‚up‘,
und dann ‚down‘

→ Count to Infinity-Problem

Link-State Routing

Grundidee:

- Jeder Router führt die folgenden Schritte durch:

  1) Die Nachbarn und deren Netzadressen ermitteln
  2) Die Kosten zu jedem seiner Nachbarn festlegen
  3) Ein Paket zusammenstellen, in dem alles steht was bisher gelernt wurde
  4) Dieses Paket an alle anderen Router senden und von allen anderen Routern derartige Pakete empfangen
  5) Den kürzesten Pfad zu allen anderen Routern berechnen

- Dadurch wird die gesamte Topologie in jedem Router abgebildet. Anwendung des Dijkstra-Algorithmus möglich!

Link-State Routing

1) Die Nachbarn und deren Netzadressen ermitteln

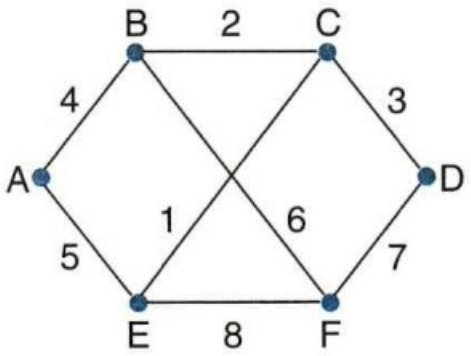
- Versenden von HELLO-Paketen

2) Die Kosten zu jedem seiner Nachbarn festlegen

- Z.B. über die reziproke Bandbreite der Verbindung
- Z.B. über spezielle ECHO-Pakete, die die Übertragungsdauer messen

Link-State Routing

3) Ein Paket zusammenstellen, in dem alles steht was bisher gelernt wurde



Link-State-Pakete

A	
Seq.-Nr.	
Alter	
B	4
E	5

B	
Seq.-Nr.	
Alter	
A	4
C	2
F	6

C	
Seq.-Nr.	
Alter	
B	2
D	3
E	1

D	
Seq.-Nr.	
Alter	
C	3
F	7

E	
Seq.-Nr.	
Alter	
A	5
C	1
F	8

F	
Seq.-Nr.	
Alter	
B	6
D	7
E	8

4) Dieses Paket an alle anderen Router senden und von allen anderen Routern derartige Pakete empfangen

- regelmäßig oder bei ‚Ereignissen‘
- Realisierung über Fluten (Broadcast)
- Versionierung durch Sequenznummern
- Altern der Pakete → keine unbegrenzte Lebensdauer

Link-State Routing

5) Den kürzesten Pfad zu allen anderen Routern berechnene

- z.B. über Dijkstra-Algorithmus

Vorteile:

- Optimale Lösung des Routings kann berechnet werden
- Gute Performance bei Änderungen des Netzwerkes

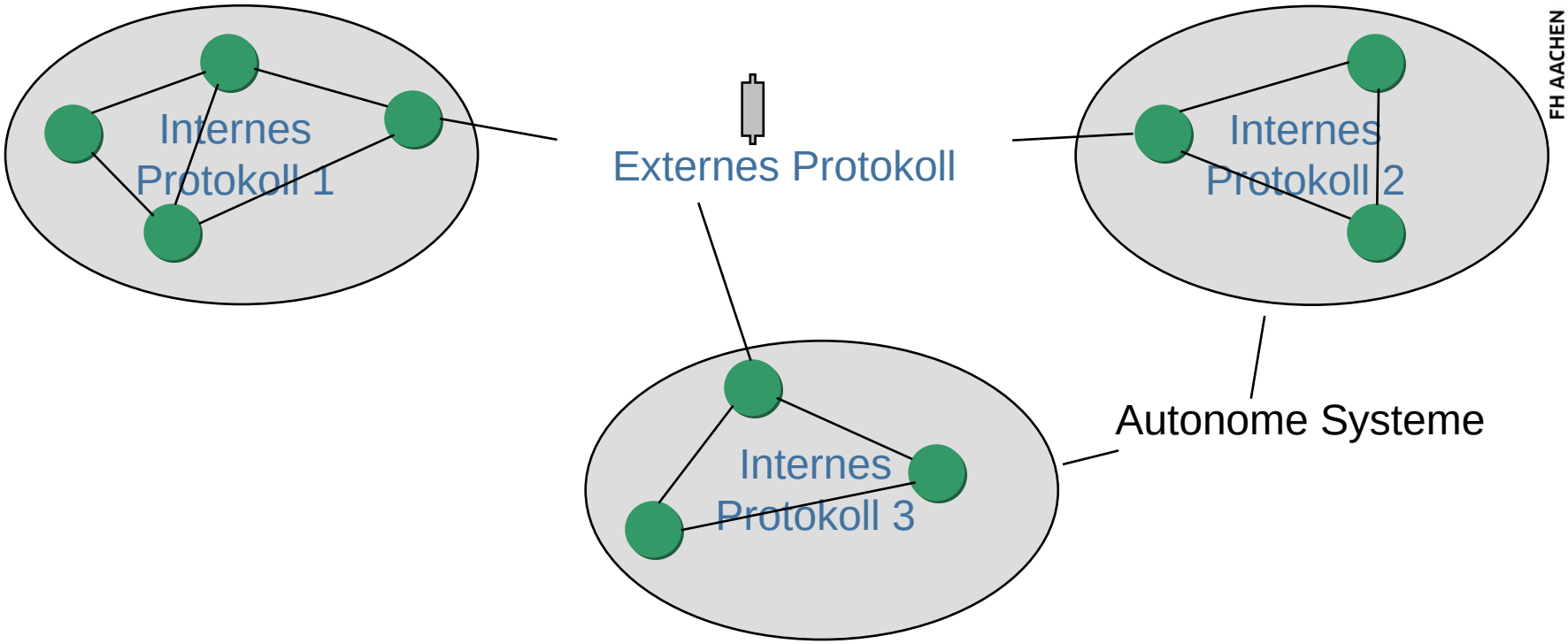
Nachteile:

- Höherer Rechenaufwand im Router
- Höhere Komplexität bei der Versendung der Link-State Pakete an alle andern Router und der Analyse der empfangenen Pakete
- Höhere Netzlast durch Routing Protokoll

Routing im Internet

- Routing im Internet muss höchst autonom und extrem dynamisch realisiert werden
- Das kann nur durch die bisher skizzierten Routing-Protokolle realisiert werden: Router unterhalten sich und tauschen Informationen aus
- Abgestuftes System:
  - Innerhalb eines Autonomen Systems kann man die gesamte Topologie wissen
  - Zwischen Autonomen Systemen sollte dies nicht Voraussetzung sein

Routing im Internet - Regionen



- Interne Protokolle – Intra-AS (OSPF, RIP, IS-IS)
- Externe Protokolle – Inter-AS (BGP)

Übersicht Routing-Protokolle

Internet Routing-Protokolle:

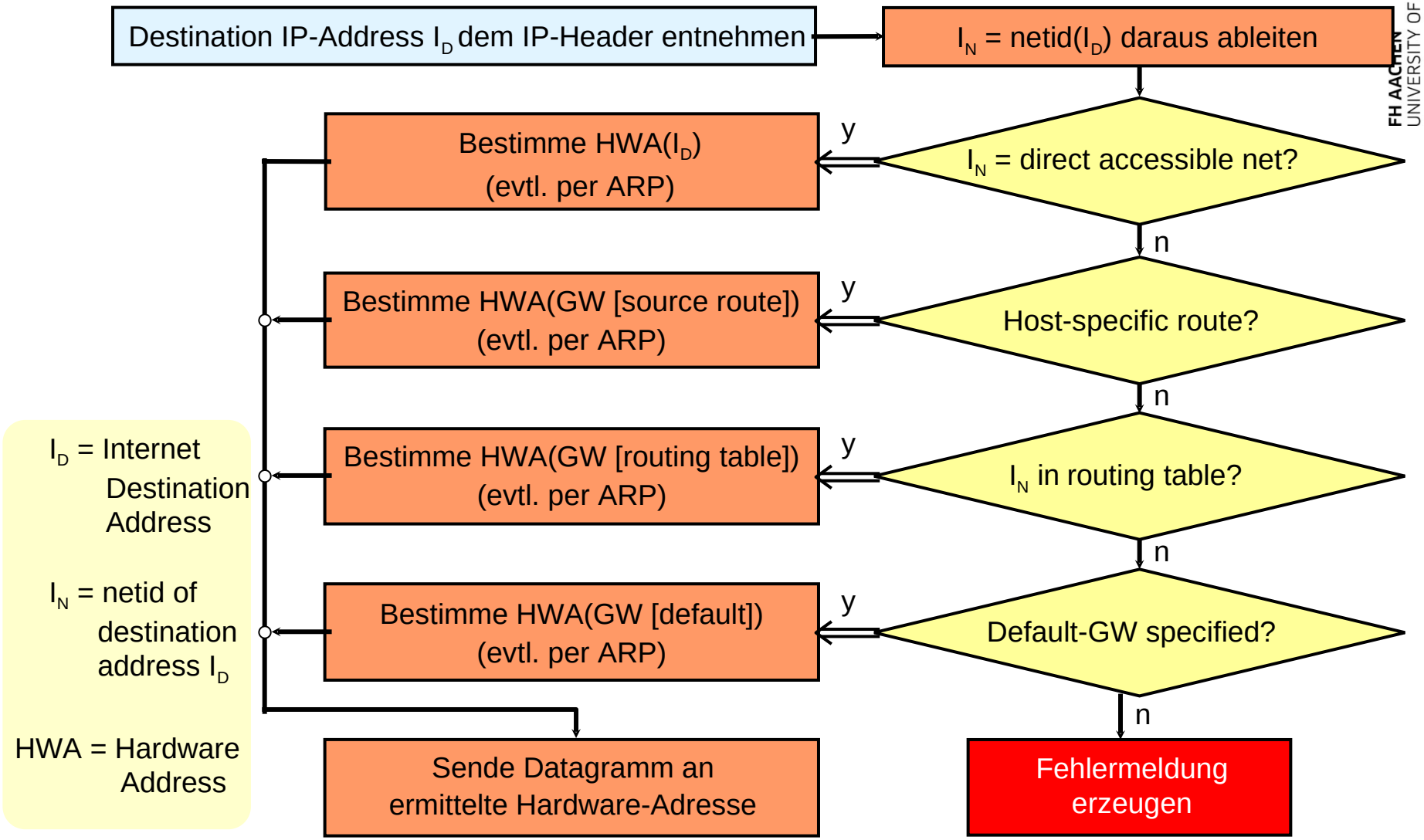
- **RIP, RIP2**: Intra-AS, Distance-Vector, veraltet
- **IS-IS:** Intra-AS, ursprünglich für DECnet, danach ISO Standard und Grundlage für OSPF, (Shortest Path First)
- **OSPF:** Intra-AS, Link-State, Unicast (Shortest Path First)
- **BGP:** Border Gateway Protokol Inter-AS, Distance-Vector

Routing: “lokal” vs. “global”

Jeder Rechner braucht eine Routing-Tabelle

- **Router im Internet**
  - > Meist Link-State-Verfahren (Dijkstra) wie OSPF **innerhalb** autonomer Systeme (Backbones)
  - > Distance-Vector-Verfahren (Bellmann-Ford) zur Kopplung autonomer Systeme
- **Router in Firmen / Stadtnetzes**
  - > Können Link-State- oder Distance-Vector-Verfahren nutzen
  - > Meist sinnvoller: statische Konfiguration
    - o Wenn es sowieso nur einen Pfad in den Backbone gibt, braucht man auch keine Informationen über mögliche Pfade auszutauschen
  - > Endrechner
    - o Statische Konfiguration oder per DHCP bezogene Informationen

Die Forwarding-Entscheidung im Detail



FH Aachen
Fachbereich 9 Medizintechnik und Technomathematik
Prof. Dr.-Ing. Andreas Terstegge
Straße Nr.
PLZ Ort
T +49. 241. 6009 53813
F +49. 241. 6009 53119
Terstegge@fh-aachen.de
www.fh-aachen.de