Rechnernetze und Telekommunikation

Routing

Übersicht

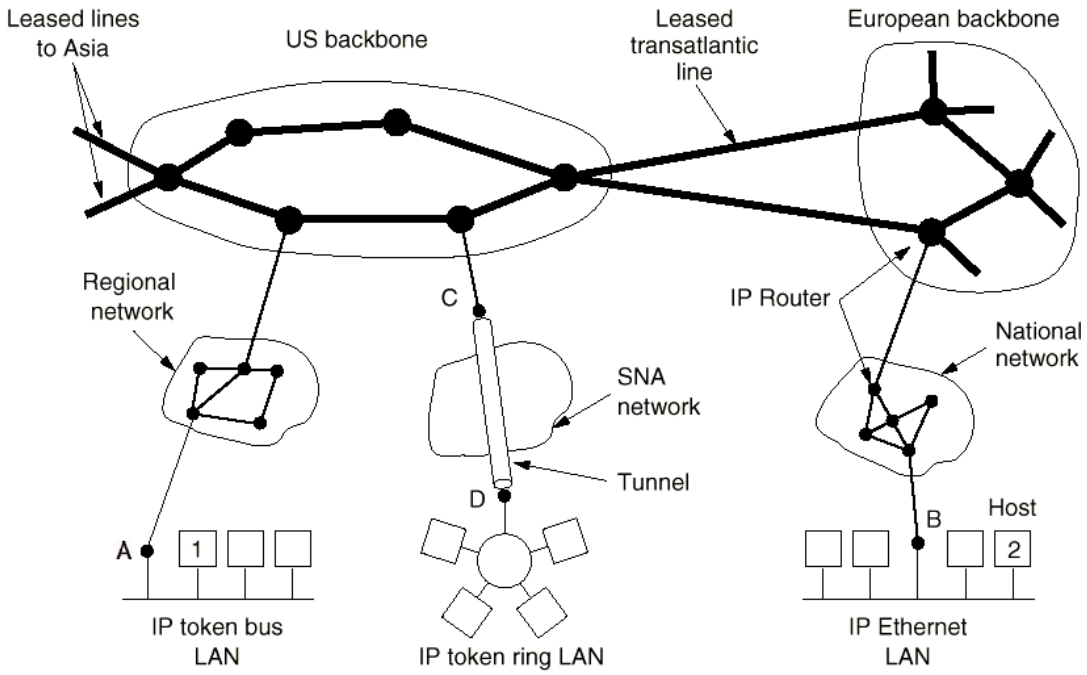
- Definition und Prinzipien des Routings
- IP-Netze und Subnetze
- Statisches und dynamisches Routen
- Distance Vector Routing
- Link State Routing
- Routing Protokolle und Software
- Network Address Translation

Wo findet überall Routing statt?

- Zwischen AS
  - zwischen Carriern
- Innerhalb eines AS
  - zwischen verschiedenen Standorten
  - zur Bildung von Subnetzten
- Innerhalb eines Subnetzes
  - zur Trennung von Netz-Segmenten
  - z.B. LAN-Segmente, VLANs
  - z.B. zur Anbindung von privaten Netzen (z.B. Home-Router mit NAT)
- Innerhalb eines Rechners
  - z.B. bei mehreren Netzwerk-Interfaces
  - Z.B. bei mehreren Virtual Machines auf einem Rechner

Wiederholung: “Struktur” des Internet

- Besteht aus zusammengefügten Netzen unterschiedlicher Organisationen
  - Sog. “Autonomen Systeme” (AS)
- IP, das “Internet Protocol” hält alles zusammen

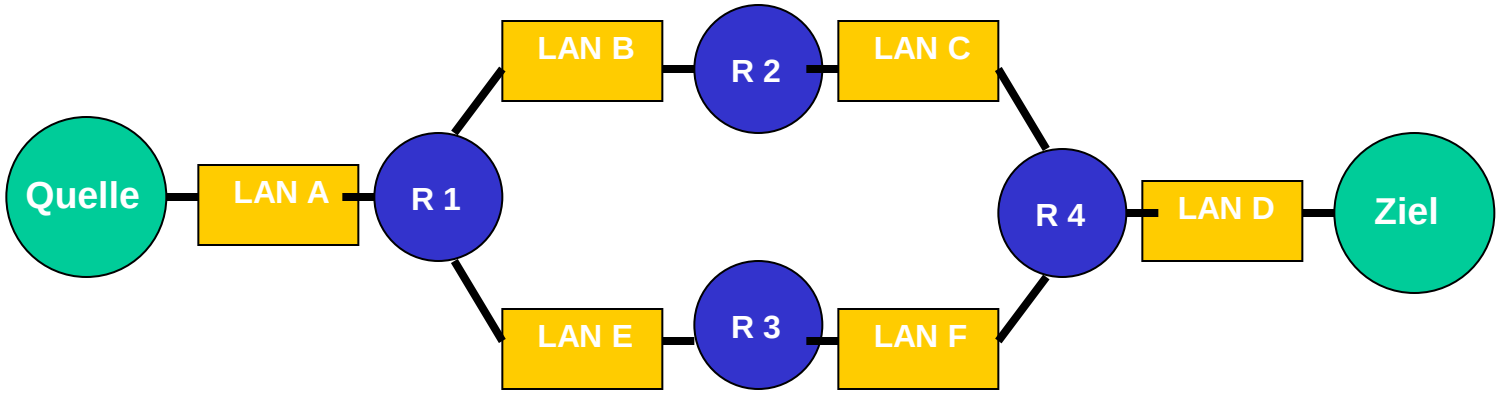


Begriff Autonome Systeme (AS)

- **AS = Administrativ „abgeschlossene“ Einheiten in einem Internet**
  - **d.h. abgeschlossen im Sinne der Verteilung von interner IP-Routinginformationen**
  - **typische AS: IP Provider Netzwerke**
  - **AS sind durch eine 16-bittige Nr. gekennzeichnet (*ASN*)**
    - öffentliche ASN, die im Internet benutzt werden dürfen:1 – 64511, z.B.:
      - AS3320 DTAG Deutsche Telekom
      - AS50595 Hochschule RheinMain
      - private ASN, nur innerhalb einer Organisation: 64512 - 65535
  - **Routen innerhalb eines AS werden nicht an andere AS propagiert**
  - **es gibt keine Default-Routen zwischen AS**

Prinzip des Routings

- Ein Router empfängt IP-Pakete von einem Netz und überträgt sie auf ein anderes
- Alle für das Routing erforderlichen Informationen sind in jedem IP-Paket enthalten.
- Der Router muss wissen:
  - welche Netze angeschlossen sind (Netznummer, Netzmaske)
  - welche anderen Netze über welche "Gateways" (nächster Router, der direkt erreichbar ist) erreichbar sind
  - wohin Pakete zu senden sind, für die kein direktes Routing möglich ist



IP-Netze (1)

- **Ein IP-Netz ist eindeutig definiert durch seine Netzadresse und die Netzmaske**
  - **Die IPv4 Netzadresse ist eine 32-bittige Zahl, i.d.R. geschrieben in der Punkt-Schreibweise mit Dezimalzahlen**
    - z.B.: 151.41.176.0 (= 10010111 00101001 10110000 00000000_2)
  - **Die IPv4 Netzmaske ist eine Zahl von 0-32, die angibt, wieviele Bitstellen der Netzadresse zur Netznummer gehören, i.d.R. geschrieben in der /-Notation oder auch in der Punkt-Schreibweise**
    - z.B.: /20 oder
      255.255.240.0 (= 11111111 11111111 1111 0000 00000000_2)
      20 x 1
  - **Die Bitstellen der Netzadresse, die nicht zur Netznummer gehören sind in der Netzadresse immer alle 0**
    - z.B.: 151.41.176.0/20 (= 10010111 00101001 1011 0000 00000000_2)
      20 Bit Netznummer | 12 Bit mit 0

IP-Netze (2)

- Die Bitstellen der Netzadresse, die nicht zur Netznummer gehören, werden für die Hostnummern genutzt
  - Ein IP-Netz hat also immer $2^{32\text{-Länge der Netzmake}}$ Hostnummern
  - z.B.: 151.41.176.0/20 hat 2^{12} Hostnummern
- Eine kompette IP-Adresse setzt sich somit aus der Netznummer und der Hostnummer zusammen
  - z.B.: Das Netz 151.41.176.0/20 hat also die Adressen von
  - 151.41.176.0 (= 10010111 00101001 10110000 00000000_2) bis
  - 151.41.191.255 (= 10010111 00101001 10111111 11111111_2)
- Die höchste IP-Adresse, also die mit nur 1en in der Hostnummer, ist die Broadcast-Adresse des Netzes
  - z.B. 151.41.191.255 (= 10010111 00101001 10111111 11111111_2) ist die Broadcast-Adresse von 151.41.176.0/20

IP-Netze (3)

- **Um zu testen, ob eine beliebigen IP-Adresse zu einem bestimmten Netzwerk gehört:**
  - **Testet man, ob: IP-Adresse & Netzmaske == Netzadresse**
    - **z.B. Test, ob IPv4-Adresse 151.41.181.201 in 151.41.176.0/20 liegt:**

151.41.181.201 = 10010111 00101001 10110110 11001001_2
151.41.176.0 = 10010111 00101001 10110000 00000000_2
255.255.240.0 („/20“) = 11111111 11111111 11110000 00000000_2

10010111 00101001 10110110 11001001_2 (= 151.41.181.201)
& 11111111 11111111 11110000 00000000_2 (= 255.255.240.0)
10010111 00101001 10110000 00000000_2 (= 151.41.176.0)

✓Ja

IP-Netze (4)

- **z.B. Test, ob IPv4-Adresse 151.41.193.201 in 151.41.176.0/20 liegt:**

151.41.193.201 = 10010111 00101001 11000001 11001001_2
151.41.176.0 = 10010111 00101001 10110000 00000000_2
255.255.240.0 („/20“) = 11111111 11111111 11110000 00000000_2

10010111 00101001 11000001 11001001_2 (= 151.41.193.201)
& 11111111 11111111 11110000 00000000_2 (= 255.255.240.0)
10010111 00101001 11000000 00000000_2 (= 151.41.192.0)

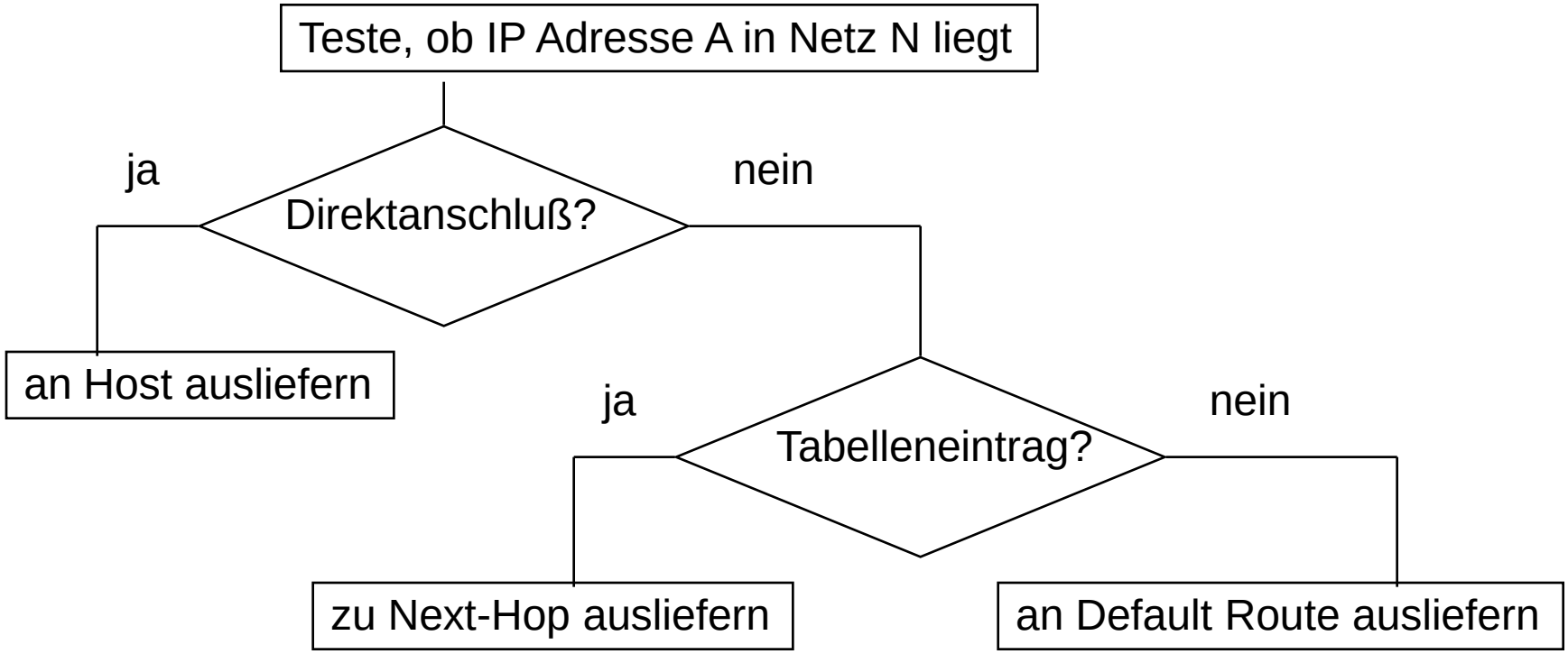
✘Nein

Subnetze

- **Um ein IP-Netz in 2^n Subnetze zu unterteilen**
  - **Verlängert man die Netzmaske um *n* und**
  - **Und vergibt auf den *n* neuen Bit der Netznummer alle 2^n möglichen Bit-Kombinationen**
  - **z.B. 151.41.176.0/20 (= 10010111 00101001 10110000 00000000_2) soll in $2^2 = 4$ Subnetze geteilt werden:**
    - Neue Netzmaske /22
    - Auf den 2 neuen Bit der Netznummer alle 2 Bit-Kombinationen

1. 10010111 00101001 10110000 00000000_2 = 151.41.176.0/22
2. 10010111 00101001 10110100 00000000_2 = 151.41.180.0/22
3. 10010111 00101001 10111000 00000000_2 = 151.41.184.0/22
4. 10010111 00101001 10111100 00000000_2 = 151.41.188.0/22

Prinzip des Routings
Algorithmus



- IP-Routing Tabelle enthält „Next-Hop“ Information, d.h.
- grundsätzlich Zeilen der Art (N, R) wobei:
  - **N: IP Adresse eines Zielnetzwerkes oder einer Zielstation**
  - **R: IP Adresse des „nächsten“ Routers entlang des Pfades zum Ziel der über ein direkt angeschlossenes Schicht 2 Netzwerk erreicht werden kann**

Beispiel für eine Routingtabelle

Netzadresse	Netzmaske	Ziel
193.174.12.0	255.255.255.240	eth0
194.121.202.160	255.255.255.248	sl0
194.174.11.176	255.255.255.240	gw 193.174.12.10
default		gw 193.174.12.1

Statisches Routing

Festlegungen:

- **IP Routing Tabelle wird manuell oder höchstens teilautomatisiert (ICMP-Redirect, SNMP) auf jedem System getrennt verwaltet**
- **Verwaltungsaufwand steigt mit Größe des Netzwerks, Zahl der Routen, Geschwindigkeit des Wachstums**
- **keine automatische Rekonfiguration, d.h. keine alternative Pfadwahl z.B. in Ausfallsituationen**

- **Vorteile:**
  - **in kleinen, sich seltenen ändernden Netzwerken ohne redundante Router etc. leicht wartbar**
  - **keine Sicherheitsprobleme durch Routingprotokolle**

Dynamisches Routing

- **IP Routing Tabellen werden mit Hilfe von Routing Protokollen automatisch zwischen den beteiligten Systemen aktualisiert**

- **Ziele:**
  - **Routing Tabellen in allen beteiligten Systemen möglichst zu jedem Zeitpunkt aktuell halten**
  - **Änderungen im Netzwerk (bei Ausfall, Wartungsarbeiten etc.) so schnell wie möglich an alle beteiligten Systeme verbreiten**
  - **d.h. Erhöhung der Zuverlässigkeit und Ausfallsicherheit, Verringerung des Wartungsaufwands**
  - **Lastverteilung (Diensttypen etc.)**

Routingprotokolle

Unterteilung der Routingprotokolle nach:

- **Einsatzgebiet (im administrativen Sinne)**
  - **Interior Gateway Protokolle (IGP‘s)**
    - für dynamisches Routing innerhalb eines AS
  - **Exterior Gateway Protokolle (EGP‘s)**
    - für dynamisches Routing zwischen verschiedenen AS

- **verwendeten Protokollalgorithmen**
  - **Distance Vector (Bellman-Ford)**
  - **Path Vector**
  - **Link State (Shortest Path First; SPF)**

- **Multicast oder Unicast Routing Protokoll**

Distance Vector Routing (1)
Grundlagen

- **Typischer verteilter Algorithmus**
  - **Kein zentrales Wissen oder eine globale Sicht des Netzes**

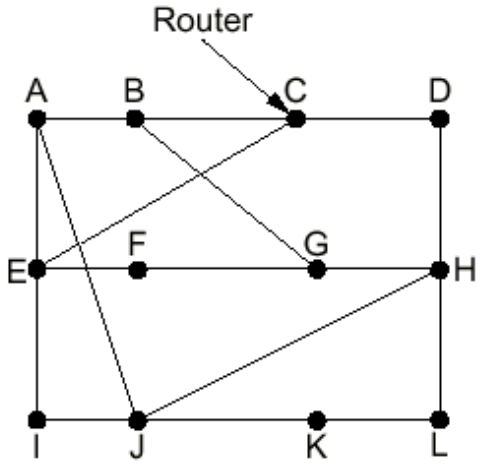
- **Idee Distance-Vector Routing (*DVA*)**
  - **Jeder Router unterhält eine Tabelle mit „Richtungen“ und „Distanzen“ für alle möglichen anderen Ziele**
  - **Alle bekannten Ziele/Distanzen werden zeitlich periodisch benachbarten Routern „angezeigt“**
    - per broad-, multi- oder unicast Nachrichten
  - **Aus den regelmäßig empfangenen Paketen konstruiert bzw. aktualisiert der DVA. die Einträge für die Routing Tabelle**
    - d.h. sucht die Richtung mit der kleinsten Distanz zu dem jeweiligen Ziel

Distance Vector Routing (2)
Details

- Routing Updates werden nur an benachbarte Router an direkt angeschlossenen Segmenten („direkte Links“) versendet
- jeder Router kennt immer nur den „Next Hop“ für jedes mögliche Ziel
- verwendete Distanzen in DVAs: „Metriken“
  - z.B. Zahl der „Hops“ zum Ziel, oder administrativ festgesetzte oder protokollspezifische Werte
- „beste“ Distanz (Metrik 0)
  - normalerweise direkte Links
- bei Änderungen werden sofort Updates gesendet
  - z.B. wenn ein Interface ausfällt
- Routen, für die eine bestimmte Zeitlang kein Update empfangen wurde, werden entfernt; optional abschaltbar

Distance Vector Routing (3)
Beispiel

- **Für Router J:**



(a)

New estimated delay from J

To	A	I	H	K	Line
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 delay is 8	JI delay is 10	JH delay is 12	JK delay is 6	New routing table for J

Vectors received from J's four neighbors

(b)

Distance Vector Routing (4)
Bewertung

Vorteile:

- **Wartung tyischerweise relativ leicht**
- **weit verbreitet, auf vielen Plattformen verfügbar (d.h. insbesondere RIP)**

Nachteile:

- **skalieren sich schlecht für große Netzwerke**
- **Updates pflanzen sich nur langsam fort**
- **es können daher Routing-Schleifen auftreten**
- **Routing-Update-Pakete können bei vielen Zielen sehr groß werden**

RIP (1)

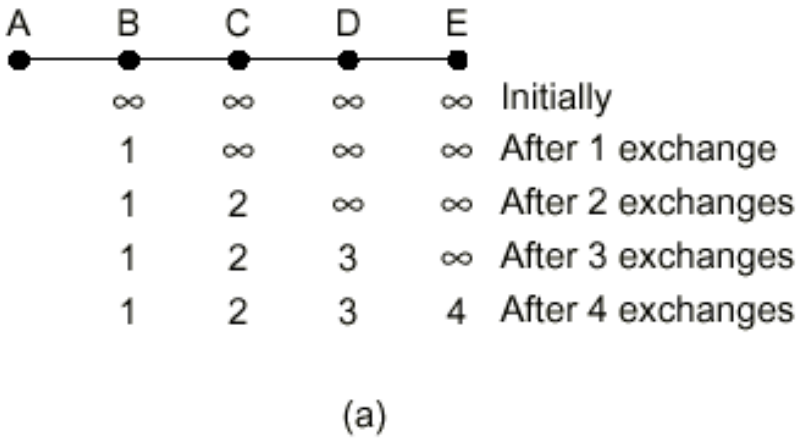
- ***Routing Information Protocol***
  - **RIP nutzt DVA**
  - **IGP (Verwendung innerhalb von AS)**
  - **war erstes verfügbares IGP**
  - **ursprünglich sehr weit verbreitet durch Distribution mit BSD Unix (*routed* Software)**
  - **entwickelt für relativ kleine Netzwerke**
  - **immer noch sehr weit verbreitet, nahezu auf jeder Plattform implementiert**
  - **Aktuell: RIP Version 2 (RFC 2453)**

RIP (2)
Parameter

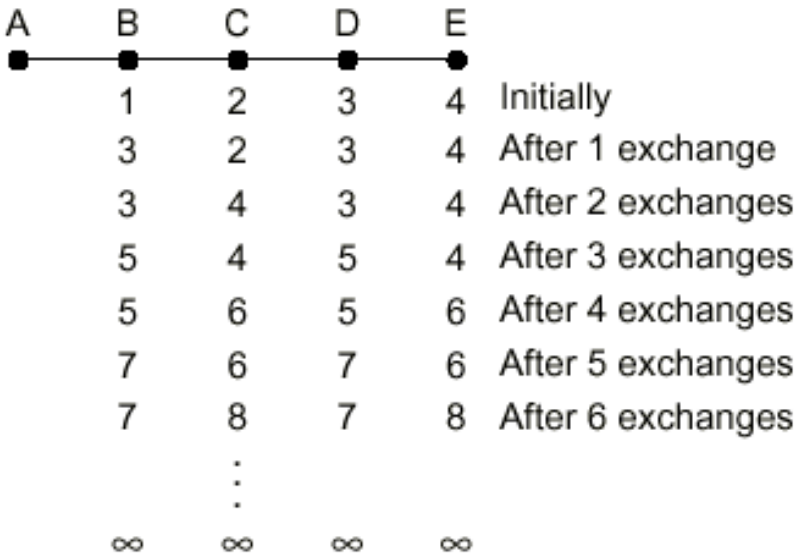
- als Distanz (Metrik):die Zahl der „Hops“ zum Ziel
  - Maximum: 15 „Hops“
  - Metrik 16 = „unendlich“, d.h. RIP ist nicht geeignet für Netzwerke mit mehr 15 Routern in einem Pfad
- RIP sendet standardmäßig alle 30 Sek. einen Update
- Wenn nach 180 Sekunden kein Update empfangen wurde
  - RIP sendet dann selber einen Request und fragt nach der Route
  - nach 270 Sekunden ohne Antwort wird die Route entfernt
- Wenn RIP lernt, dass eine Topologie-Änderung aufgetreten ist,
  - wartet es nicht bis zum nächsten periodischen Route-Updating-Zeitpunkt
  - sondern sendet sofort (Triggered Update)

Probleme des DVA (1)

- **Count-to-Infinity bei Ausfall einzelner Verbindungen:**
  - **Beispiel: Entfernung zu A**
  - **a) Im Normalfall (jeder Hop zählt 1)**
  - **b) Link A-B ist ausgefallen**



(a)



(b)

- **Problem: Schlechte Nachrichten verbreiten sich langsam**

Probleme des DVA (2)

- **Pfadschleifen**
  - Ein Paket wird im Kreis geroutet
  - Kann im stabilen Zustand eigentlich nicht auftreten
  - Können aber durch langsame Verbreitung temporär entstehen

- **Da im DVA kein Knoten eine globale Sicht hat, wird dies nicht bemerkt**
  - Kein Router schaut über den Horizont des nächsten Hops hinaus uns weiß, wie der Pfad weiter geht

Path Vector Routing
Grundlagen

- **Funktionsweise ähnelt sehr stark dem DVA**
- **Routingschleifen wird vorgebeugt**
  - **Router teilt dem Nachbarn nicht nur mit, *dass* er ein bestimmtes Netz zu bestimmten Kosten erreichen kann,**
  - **sondern auch den kompletten Pfad, den er nutzen würde**
- **Ein Router**
  - **kann so verschiedene Pfade zu einem Ziel kennen und unterscheiden**
- **Merkt ein Router nun, dass er bereits in diesem Pfad vorhanden ist, verwirft er das Update und vermeidet so, dass eine Routingschleife entsteht**

BGP

- ***Border Gateway Protocol***
  - **Nutzt Path Vector Routing**
  - **Aktuelle Version 4 definiert in RFC 4271**
  - **DAS EGP Protokoll zum Routing zwischen AS**
  - **Nutzt für die Verbindungen zwischen den Routern TCP**

Link-State Routing (1)
Grundlagen

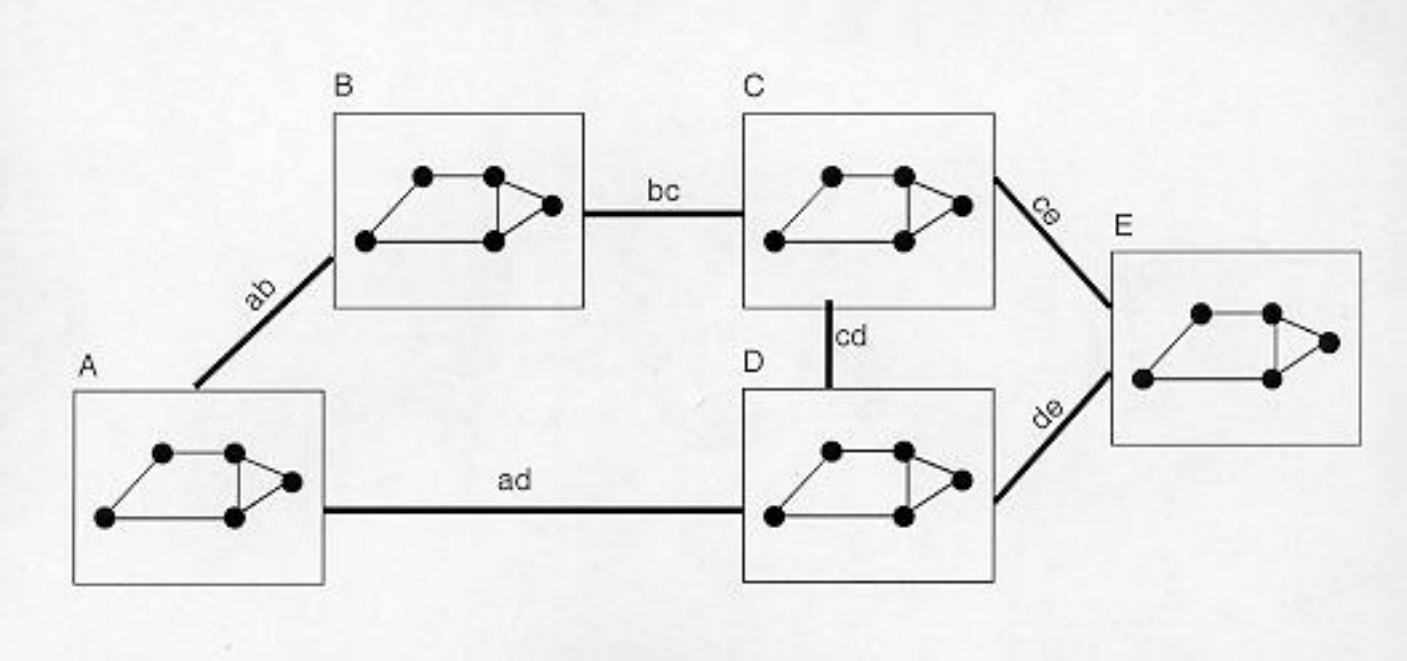
- Lokaler Algorithmus
- Jeder Router wird über alle verfügbaren Links (Link-State) informiert
  - mittels broad-, multi-, oder unicast Nachrichten
- Jeder Router unterhält eine komplette Topologieinformation über das Netzwerk
  - d.h. jeder Router kennt jeden anderen Router und die an diesen angeschlossenen Netzwerke (Link-Graph)
  - Auch „Full Topology Routing „ genannt

Link-State Routing (2)
Algorithmus

- Ermittlung aller direkten Nachbar-Router
- Jeder Router testet aktiv und zeitlich periodisch den Status aller benachbarten Router (d.h. direkte Links)
- Die gesamte Link-Status Information werden zeitlich periodisch *allen* anderen beteiligten Routern im Netzwerk mitgeteilt
- Router aktualisiert seine eigene Topologie Datenbasis, indem Links als „up“ bzw. „down“ markiert werden
- Werden Änderungen bei Links festgestellt, werden die betroffenen Routen neu berechnet und die eigene IP Routingtabelle aktualisiert
  - Berechnung mittels Shortest Path Algorithmus nach Dijkstra

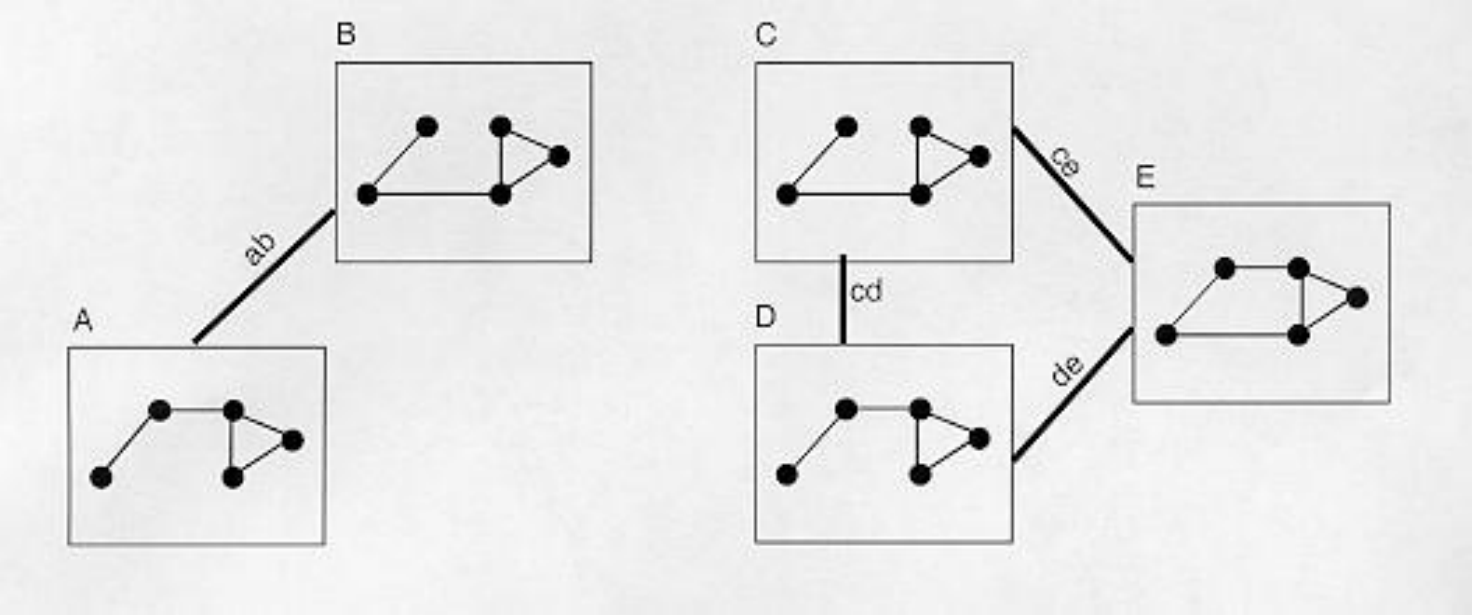
Link-State Routing (3)
Beispiel

Netz im stabilen Zustand



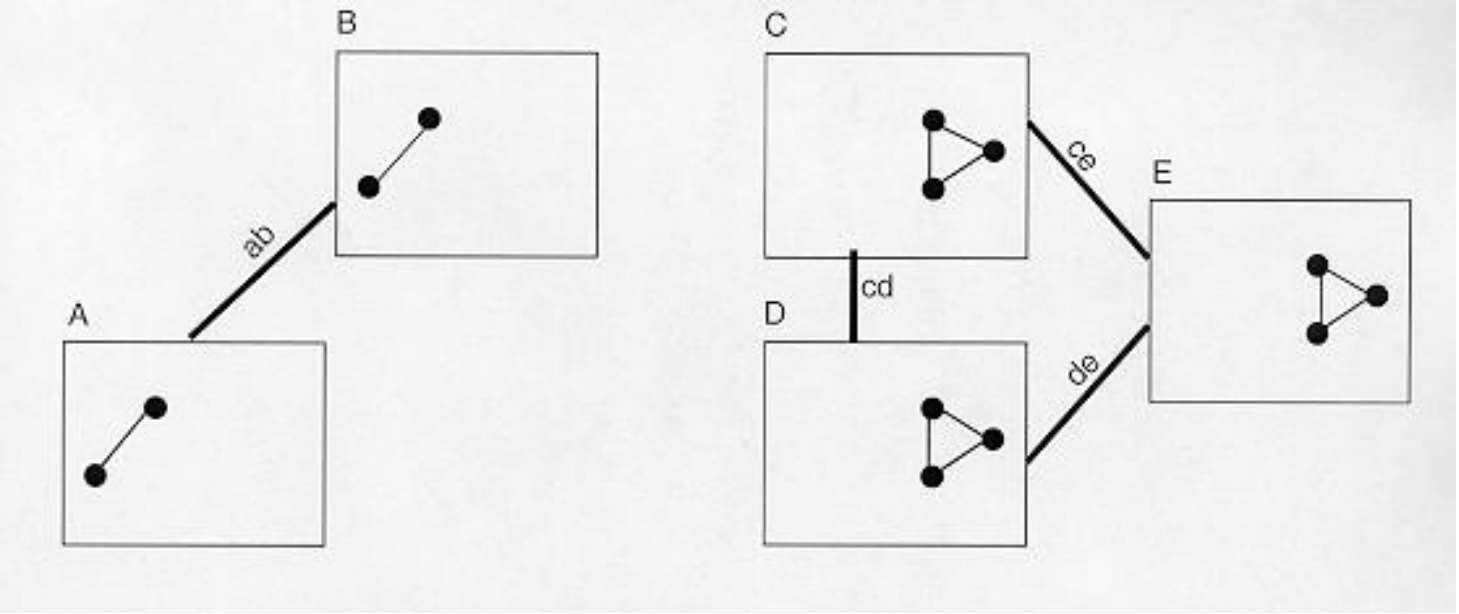
Link-State Routing (4)
Beispiel

Links bc und ad sind ausgefallen

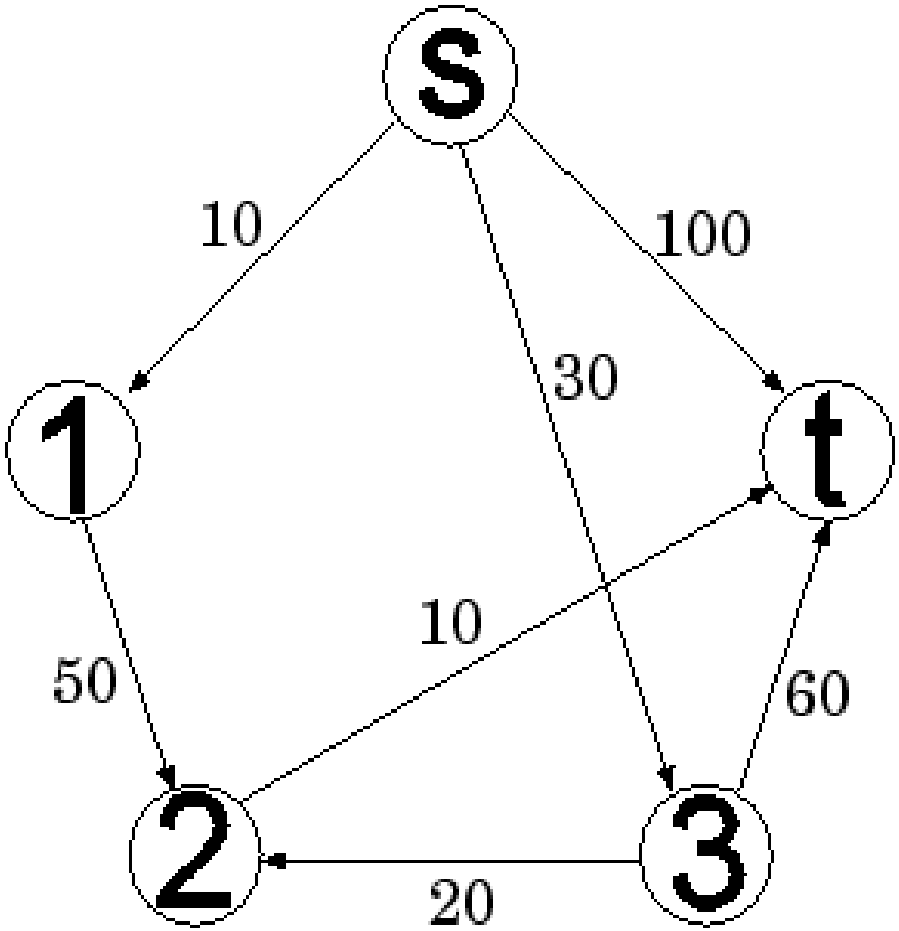


Link-State Routing (5)
Beispiel

Nach einer Nachrichtenrunde



Shortest Path: Algorithmus nach Dijkstra – Das Problem

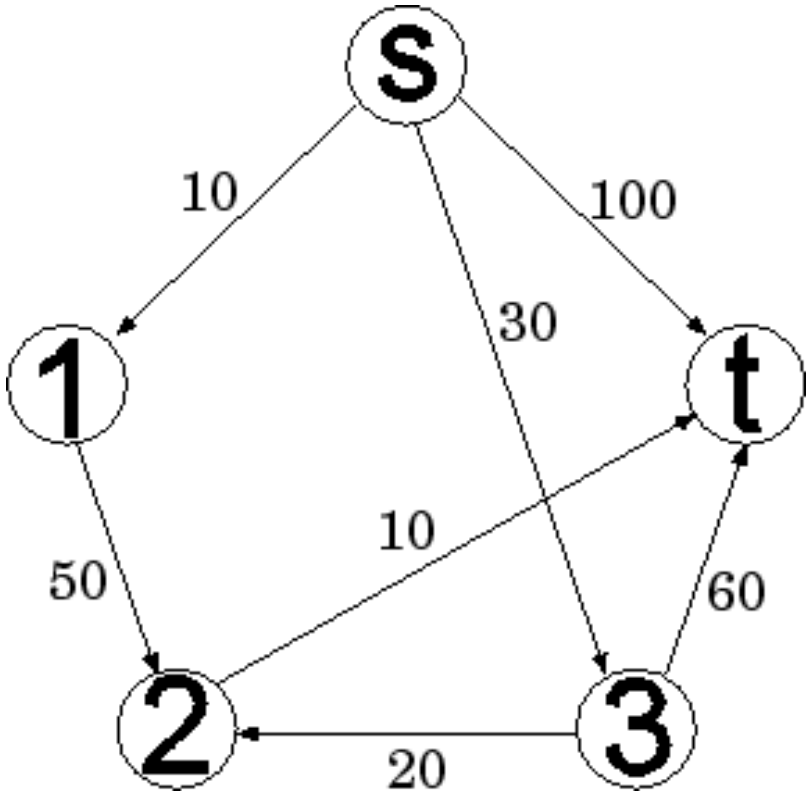


Shortest Path: Algorithmus

```
V = Menge aller Knoten;        E = Menge aller Kanten;
S = {s};      // Menge von Knoten
              // Invariante: kürzeste Weglänge zu jedem
              //                Knoten in S bekannt

while ( t ∉ S )  {
   füge zu S den Knoten aus (V \ S) hinzu, der mit kleinstem Aufwand zu erreichen ist
}
```

Shortest Path: Beispiel



S	d(1)	d(2)	d(3)	d(t)
{s}	**10**	∞	**30**	**100**
{s,1}	**10**	**60**	**30**	**100**
{s,1,3}	**10**	**50**	**30**	**90**
{s,1,3,2}	**10**	**50**	**30**	**60**
{s,1,3,2,t}	**10**	**50**	**30**	**60**

Link-State Routing (6)
Bewertung

Vorteile:

- **jeder Router berechnet seine Routingtabelle unabhängig von anderen, mit der originalen Link-Status-Info des „anzeigenden“ Routers**
- **d.h. keine Abhängigkeit von den Berechnungen von „Zwischen“-Routern**
- **Probleme leichter zu finden**
- **Größe der Pakete hängt nicht von der Zahl der gerouteten Netzwerke ab (d.h. L.S.A.‘s skalieren sich besser)**
- **Link-Status kann zusätzliche Information, wie Qualität des Links („cost“) enthalten, dadurch optimale Pfadwahl möglich**

Nachteile:

- **meist aufwendiger zu warten**
- **meist höhere Rechnerleistung auf dem Router erforderlich bei großen „Gebieten“**

OSPF (1)

- ***Open Shortest Path First***
  - **Link-State-Routing-Protokoll**
  - **Ist im RFC 2328 definiert**
  - **IGP (Verwendung innerhalb von AS), auch EGP möglich**
    - Unterteilt das Netz in separate *Areas*
    - Area 0 ist der Backbone
  - **Standard definiert nicht wie die Kosten zu berechnen sind**
    - Cisco nutzt z.B. die Bandbreite des Links als Kosten

OSPF (2)
Weitere Features

- **Route Aggregation**
  - Zusammenfassung von Routen mit gleichem Präfix (classless)
  - kann Größe der Routing Tabelle und Protokoll Traffic erheblich minimieren
- **Type of Service Routing**
  - mehrfache Routen zum gleichen Ziel installierbar, für verschiedene Servicetypen (Pfadwahl dann durch Felder IP-Header)
- **Load Balancing**
  - bei mehrfachen Routen zum gleichen Ziel mit gleicher „Cost“ kann OSPF Traffic über diese Pfade gleich verteilen
- **Authentication**
  - Routing Pakete können mit verschiedenen Verfahren authentifiziert werden

Übersicht Routingprotokolle

Internet Routingprotokolle

- **RIP, RIP2:** IGP, Distance Vector
- **IGRP:** IGP, Distance Vector
- **OSPF:** IGP/EGP, Link State
- **EIGRP:** IGP/EGP, Hybrid Distance Vector/Link State
- **BGP** EGP, Path Vector

Routingsoftware (1)

- Router können als dedizierte Geräte oder als Software auf Standard-Betriebssystemen laufen (typ. Linux)

- Moderne Routingsoftware gestattet üblicherweise sehr weitgehende administrative Eingriffe
  - z.B. das Erlauben oder Verbieten bestimmter Routen
  - das Festlegen von administrativen „Weights“, d.h. z.B. Bevorzugung bestimmter Routen
  - das Akzeptieren von Updates nur von bestimmten Nachbar Routern
  - peer-to-peer Betrieb (d.h. z.B. kein RIP-Broadcast) passiv (nur „Lernen“, kein aktives Anzeigen)
  - konfigurierbaren Routenaustausch zwischen allen auf dem gleichen System laufenden Routing Protokollen

Routingsoftware (2)

- **unterschiedlichste Routingsoftware für verschiedenste Plattformen am Markt, sehr weit verbreitet sind u.a.:**
  - **Cisco IOS**
    - Betriebssystem nahezu der gesamten Cisco Router und Switch Produktpalette, große Zahl von unterstützten Routingprotokollen
  - **gated Software**
    - auf nahezu allen Unix‘en vorhanden, unterstützt alle wichtigen Routingprotokolle, einschließlich RIP, OSPF und BGP
  - **Quagga**
    - unter der GPL lizenziertes Softwarepaket für unix-artige Betriebssysteme, unterstützt OSPF, RIP und BGP
- **Unterscheiden sich in Features, Performance, Skalierbarkeit**

NAT (Network Address Translation)

- **Sammelbegriff für Verfahren, die**
  - automatisiert Adressinformationen in Datenpaketen durch andere ersetzen
  - und zusätzlich zum Routen stattfinden können

- **Typisch Anwendung**
  - Betreiben von Netze mit vielen Rechnern in einem lokalen Netz (private IP-Adressen) mit nur wenigen (z.B. einer) öffentlichen IP-Adresse
  - Beispiele: Home-Router, Campus-WLAN, IP-Adressen in Mobilfunk-Netzen

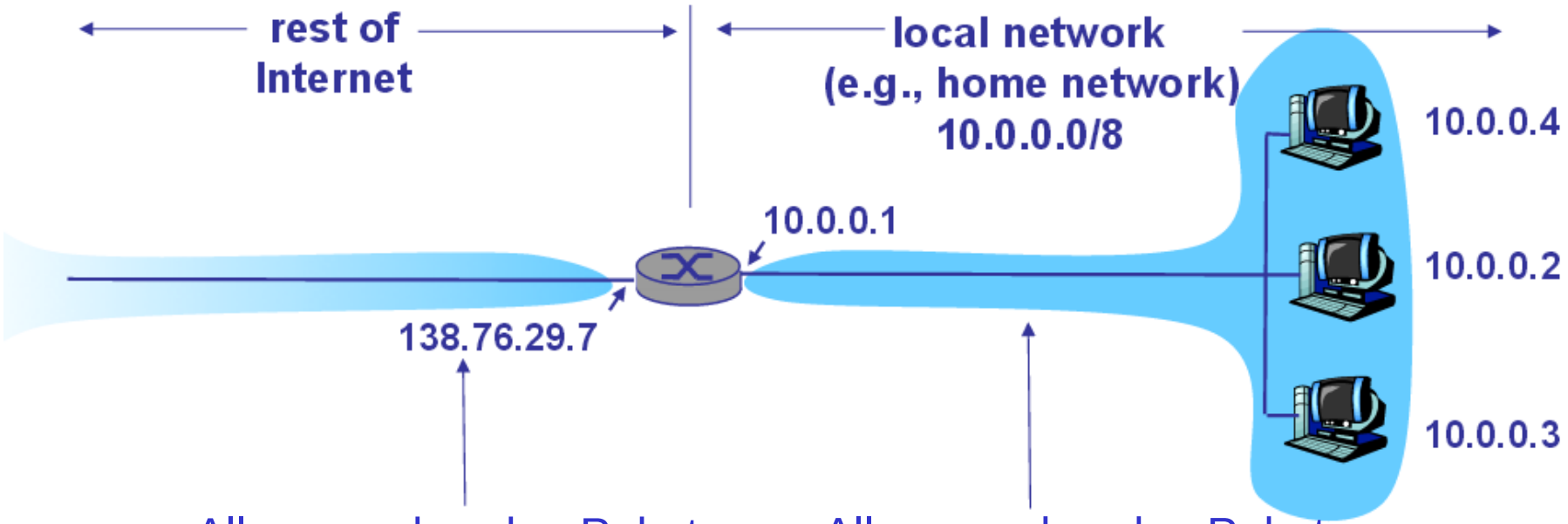
- **Vorteile**
  - Lindert massiv das Problem der knappen IPv4 Adressen
  - Schütz auch die lokalen Rechner vor Zugriffen aus dem Internet

PAT (Port Address Translation)

- **Auch Hiding NAT oder NAPT (Network Address Port Translation)**
  - **Meist aber auch einfach nur NAT genannt**
  - **Benötigt Ports, also nur für TCP/UDP (Layer 4!)**

- **Socket-Paar (IP-Addresse und Port-Nummer) wird umkodiert**
  - **NAT-Router führt eine Übersetzungstabelle:**
    - externe Adresse: externer Port – interne Adresse: interner Port
  - **Tabelle wird dynamisch erweitert und gelöscht**
    - TCP: Anlegen durch SYN-Pakete von innen, Löschen bei FIN
    - UDP: Anlegen beim ersten Paket von innen, Löschen durch Timeout

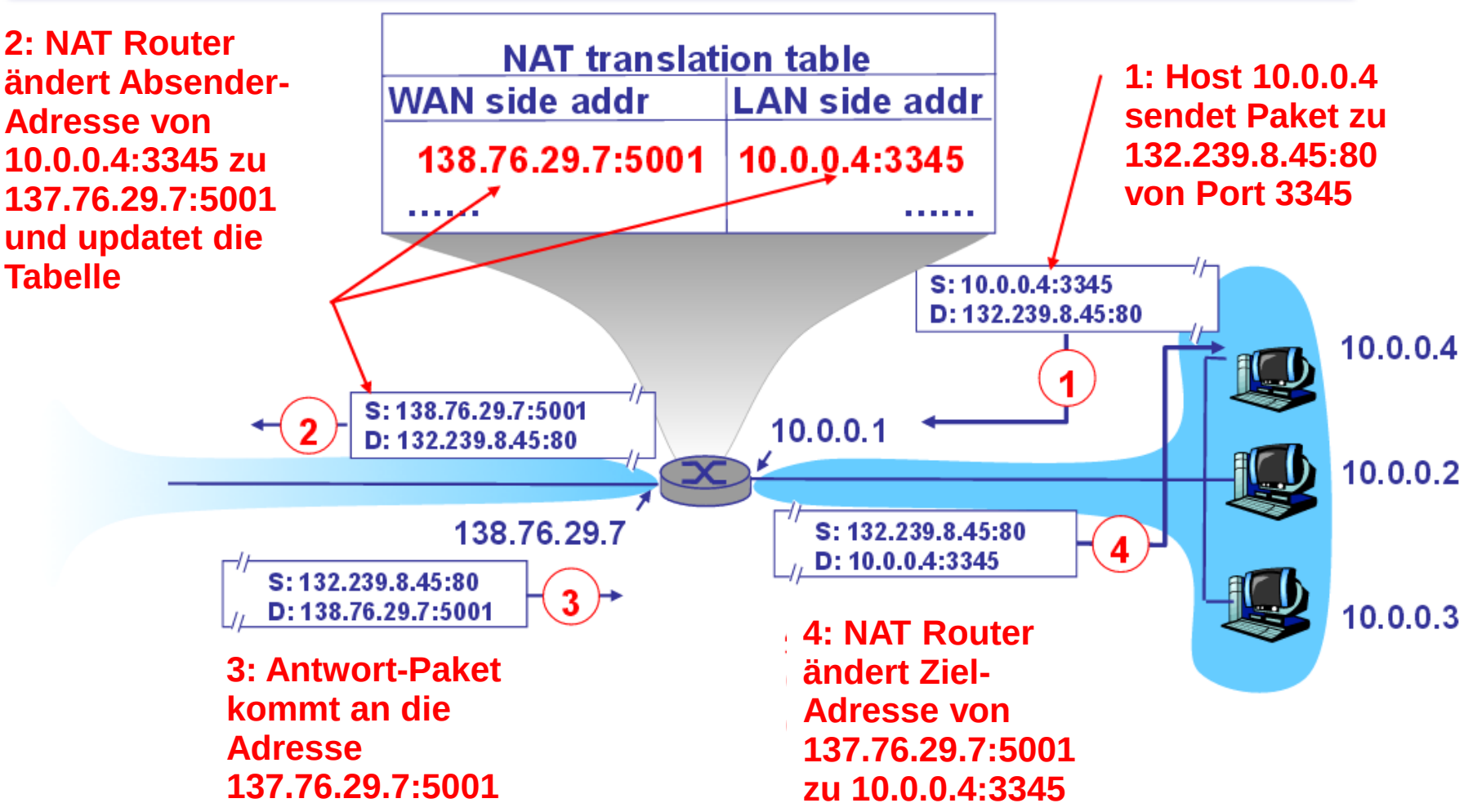
NAT – Beispiel (1)



Alle ausgehenden Pakete hier haben die gleiche Ansenderadresse 138.76.29.7, aber unterschiedliche Ports

Alle ausgehenden Pakete hier haben ihre lokale Absenderadresse

NAT – Beispiel (2)



Probleme bei NAT

- **Kann zu Problemen in Anwendungen führen, wenn IP-Adressen auch in den Nutzdaten (Layer 7) der Nachrichten verwendet werden**
  - **Adressen in den Paketen passen dann nicht mehr zu den im Protokoll sichtbaren Adressen**
    - Problem bei (sog. „active“) FTP und VoIP (SIP)

- **Ports von Server-Diensten mit lokalen Adressen sind von außen nicht unmittelbar erreichbar**
  - **Ein Eintrag in der Übersetzungstabelle existiert nur, wenn die Verbindung von INNEN aufgebaut wird**
  - **Lösung: statischer Eintrag in der Übersetzungstabelle („Port-Forwarding“)**
    - Manuell konfiguriert oder über UPnP (ggf. Security-Problem)

Zusammenfassung

- **Routing entscheidet, welchen Weg ein Paket auf dem Weg zum Ziel nimmt**
- **Statisches Routing nur machbar für kleine Netze, große Netze bestimmen Routen dynamisch mit Routing Protokollen**
- **Es gibt verschiedene Verfahren und Protokolle zum dyn. Routen:**
  - **Distance Vector Routing (z.B. RIP)**
  - **Path Vector Routing (z.B. BGP)**
  - **Link State Routing (z.B. OSPF)**
- **NAT ist ein Verfahren, um IP-Adressen beim Routing umzuschreiben**