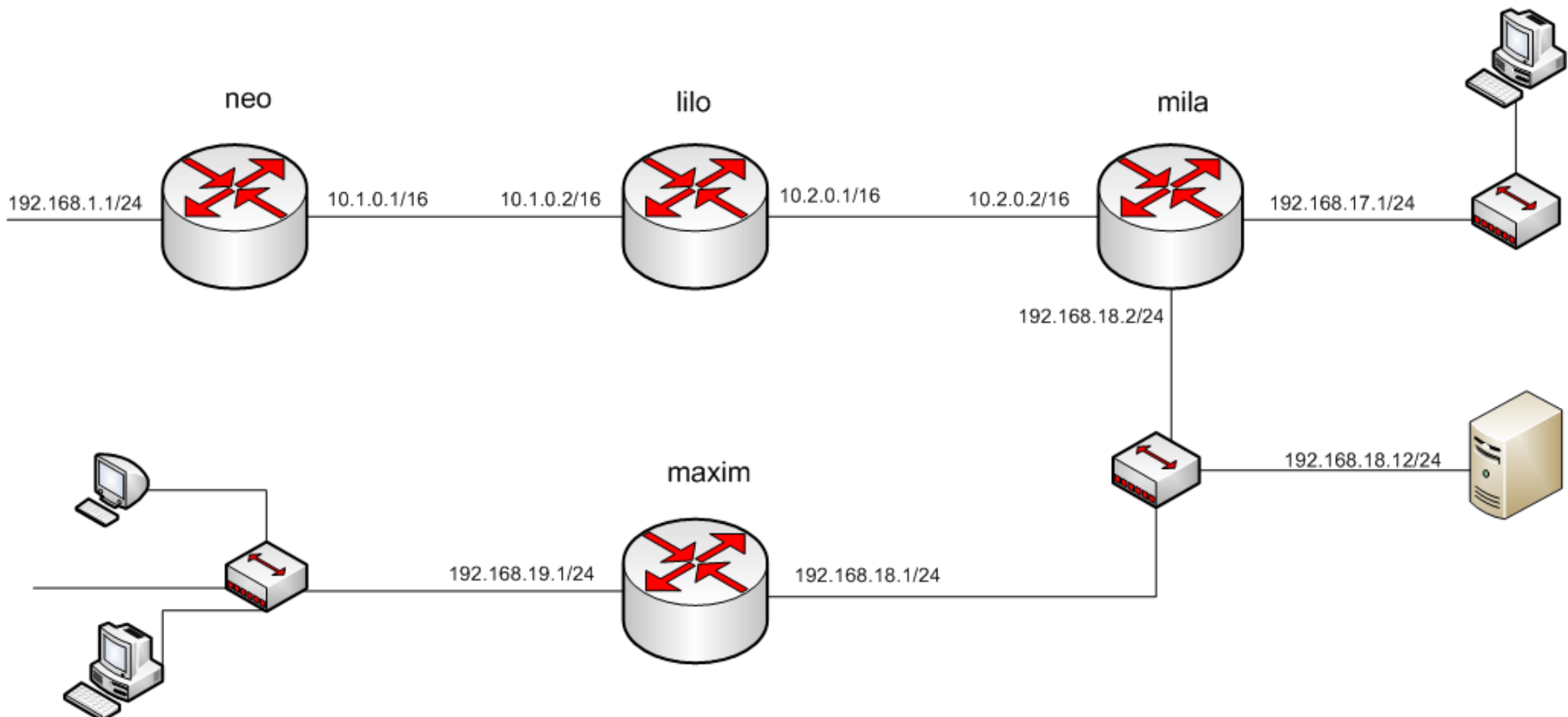


Kapitel 4: Netzwerkschicht

- 4.1 Einleitung
- 4.2 Aufbau eines Routers
- 4.3 IP: Internet Protocol
 - Datagrammformat
 - IPv4-Adressierung
 - ICMP
 - IPv6
- 4.4 **statisches Routen**
- 4.5 Routing-Algorithmen
 - Link State
 - Distance Vector
 - Hierarchisches Routing
- 4.6 Routing im Internet
 - RIP
 - OSPF
 - BGP
- 4.7 Broadcast- und Multicast-Routing

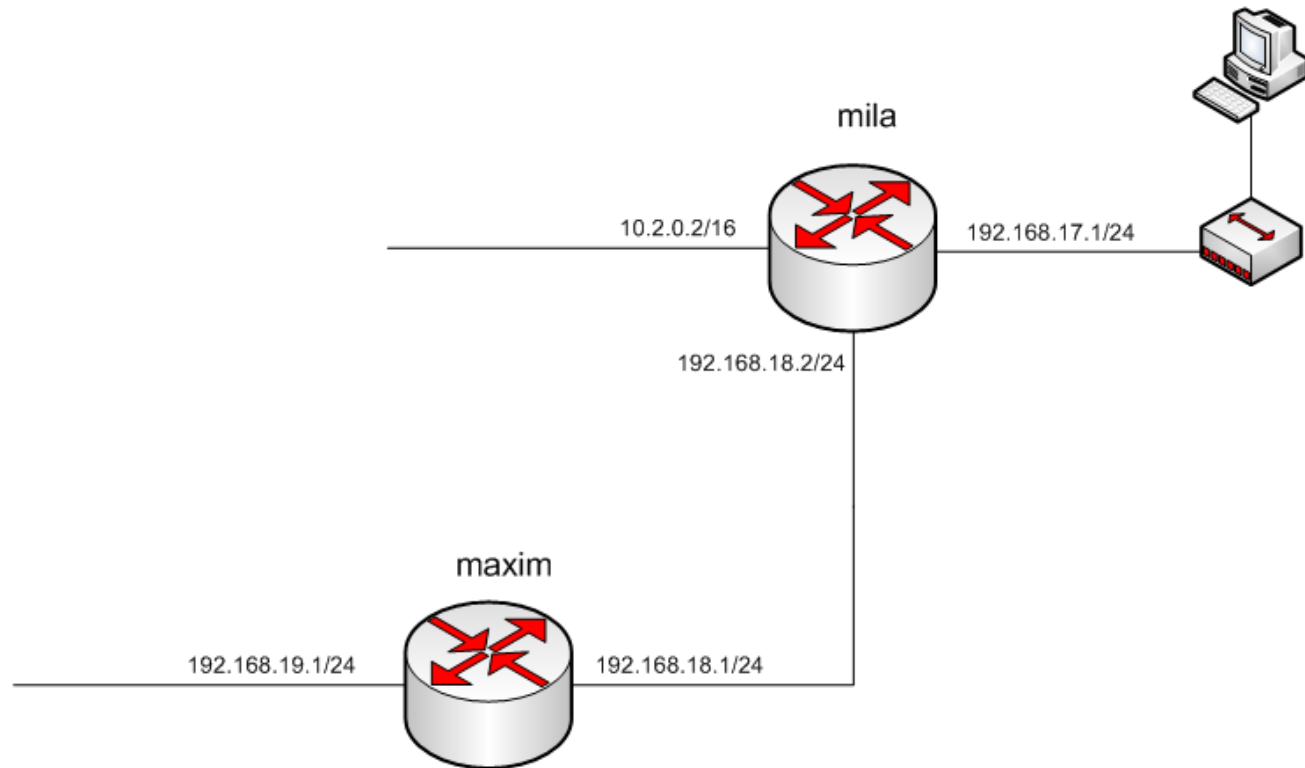
Fallbeispiel für statisches Routen(1)

- Wieviele Teilnetze hat das gezeigte Netzwerk?
- Wie lauten die jeweiligen Netzadressen und die zugehörigen Netzmasken?
- Wie lautet die Routingtabelle für den Router neo?



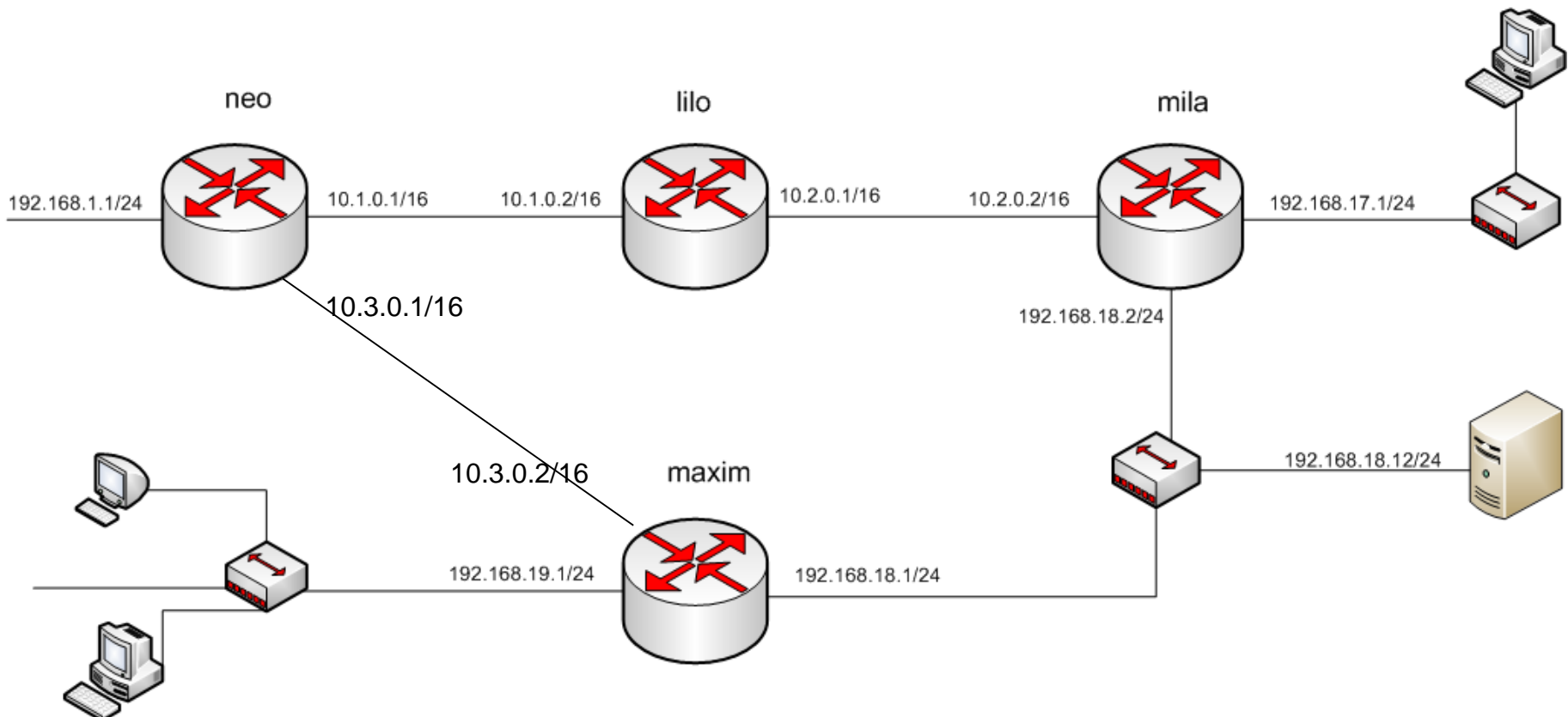
Fallbeispiel für statisches Routen(2)

- Supernetting bzw. route aggregation (CIDR)



Fallbeispiel für statisches Routen(3)

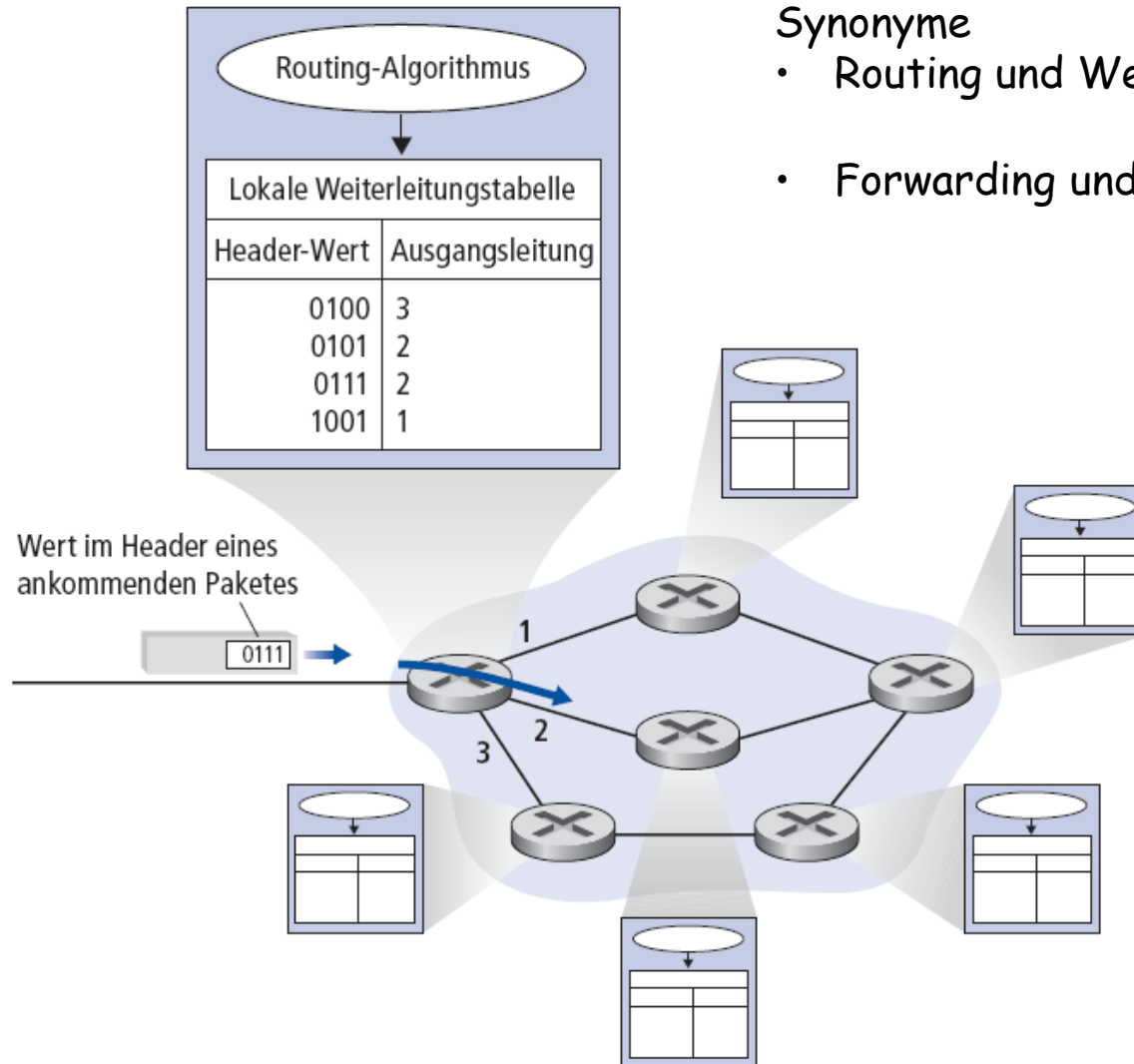
- Alternativroute von Router neo zu Router maxim



Kapitel 4: Netzwerkschicht

- 4.1 Einleitung
- 4.2 Aufbau eines Routers
- 4.3 IP: Internet Protocol
 - Datagrammformat
 - IPv4-Adressierung
 - ICMP
 - IPv6
- 4.4 statisches Routen
- 4.5 **Routing-Algorithmen**
 - Link State
 - Distance Vector
 - Hierarchisches Routing
- 4.6 Routing im Internet
 - RIP
 - OSPF
 - BGP
- 4.7 Broadcast- und Multicast-Routing

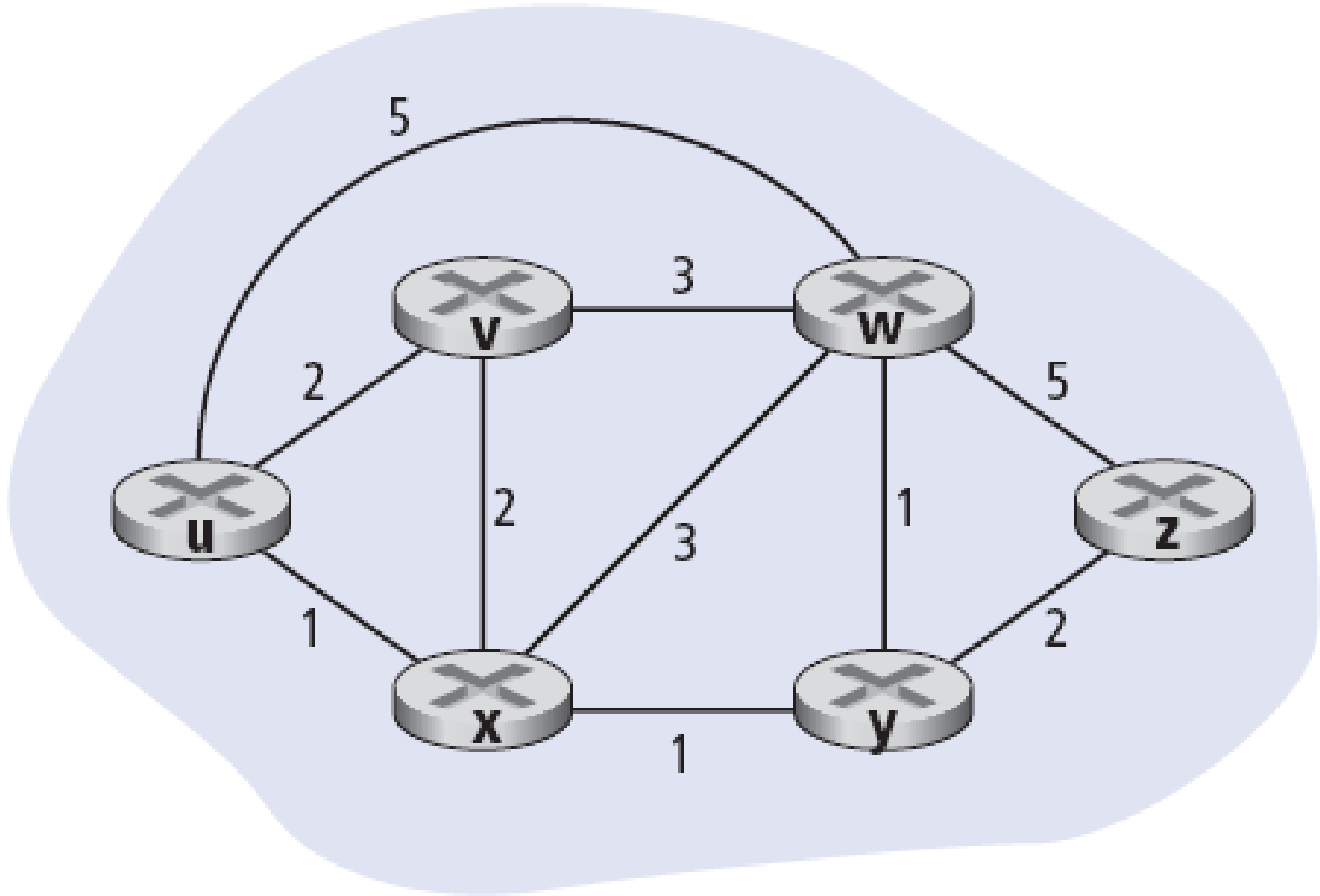
Zusammenspiel von Routing und Weiterleitung



Synonyme

- Routing und Wegewahl
- Forwarding und Weiterleitung.

Ein Netzwerk als Graph



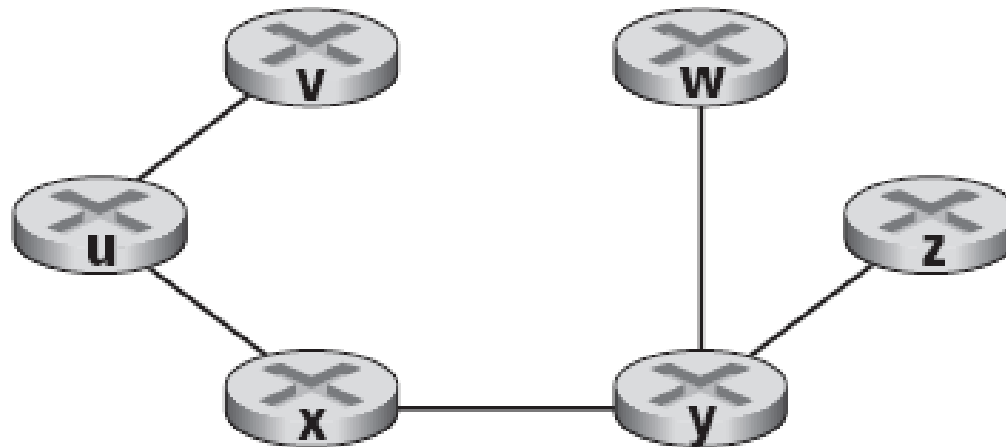
Kapitel 4: Netzwerkschicht

- 4.1 Einleitung
- 4.2 Aufbau eines Routers
- 4.3 IP: Internet Protocol
 - Datagrammformat
 - IPv4-Adressierung
 - ICMP
 - IPv6
- 4.4 statisches Routen
- 4.5 Routing-Algorithmen
 - **Link State**
 - Distance Vector
 - Hierarchisches Routing
- 4.6 Routing im Internet
 - RIP
 - OSPF
 - BGP
- 4.7 Broadcast- und Multicast-Routing

Dijsktras Algorithmus (u = Quelle)

```
1 initialisiere:
2   N' = {u}
3   für alle Knoten v aus N
4     wenn v ein Nachbar von u ist
5       dann  $D(v) = c(u,v)$ 
6       sonst  $D(v) = \infty$ 
7
8 wiederhole:
9   finde ein w aus N welches nicht in N' ist, so dass  $D(w)$  minimal ist
10  füge w zu N' hinzu
11  Berechne  $D(v)$  neu für jeden Nachbarn v von w der nicht in N' ist:
12     $D(v) = \min(D(v), D(w) + c(w,v))$ 
13  /* die neuen Kosten nach v sind entweder die alten Kosten
14  oder die Kosten nach w plus die Kosten von w nach v*/
15 bis  $N' = N$ 
```

Dijkstras Algorithmus: Beispiel (2)



Ziel	Leitung
v	(u, v)
w	(u, x)
x	(u, x)
y	(u, x)
z	(u, x)

Beispiel für ein Java Applet zur Demonstration:

http://

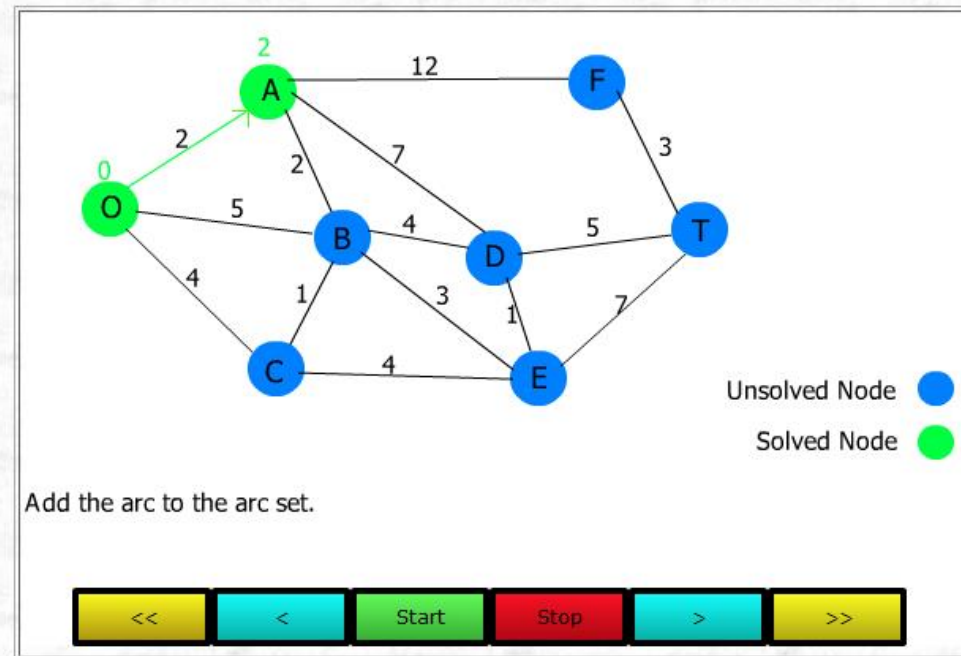
weierstrass.is.tokushima-u.ac.jp/ikeda/suuri/dijkstra/DijkstraApp.shtml?demo1

Dijkstras Algorithmus: Beispiel (3)

Example Networks1: Dijkstra's Algorithm for Shortest Route Problems

Below is a network with the arcs labeled with their lengths. The example will step through Dijkstra's Algorithm to find the shortest route from the origin O to the destination T.

Press the Start button twice to begin the example.



Kapitel 4: Netzwerkschicht

- 4.1 Einleitung
- 4.2 Datagramme und virtuelle Leitungen
- 4.3 Aufbau eines Routers
- 4.4 IP: Internet Protocol
 - Datagrammformat
 - IPv4-Adressierung
 - ICMP
 - IPv6
- 4.5 statisches Routen
- 4.6 Routing-Algorithmen
 - Link State
 - **Distance Vector**
 - Hierarchisches Routing
- 4.7 Routing im Internet
 - RIP
 - OSPF
 - BGP
- 4.8 Broadcast- und Multicast-Routing

Distance-Vector-Routing

- Prinzipielle Idee: Bellman-Ford-Gleichung
- Sei $d_x(y) :=$ billigster Pfad von x nach y
- Dann gilt:
 - $d_x(y) = \min \{c(x,v) + d_v(y)\}$
- Wobei das Minimum über alle Nachbarn v von x gebildet wird

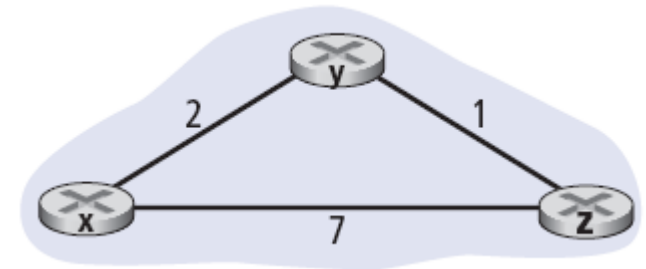
Distance-Vector-Routing

- Prinzipielles Vorgehen:
 - Wenn ein Knoten hochgefahren wird, dann sendet ein Knoten seinen eigenen Distanzvektor an alle seine Nachbarn
 - Immer wenn sich der Distanzvektor eines Knotens ändert, sendet er diesen an alle seine Nachbarn
- Wenn ein Knoten einen neuen Distanzvektor von einem Nachbarn erhält, überprüft er seinen eigenen Distanzvektor nach der Bellman-Ford-Gleichung:
$$d_x(y) = \min \{c(x,v) + d_v(y)\},$$
 Minimum wird über alle Nachbarn v gebildet
- Unter realistischen Annahmen konvergiert dieser Algorithmus zu einer Situation, in der jeder Knoten den Nachbarn auf dem günstigsten Weg zu jedem anderen Knoten kennt

Distance-Vector-Routing

- Iterativ und asynchron:
 - Jede Iteration wird ausgelöst durch:
 - Veränderung der Kosten eines Links zu einem direkten Nachbarn
 - Neuer Distanzvektor von einem Nachbarn
- Verteilt:
 - Keine globale Kenntnis der vollständigen Netzwerk-topologie notwendig
 - Stattdessen: lokales Verbreiten von Informationen

Beispiel - Notation



von

	nach		
	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

Tabelle von y

Distanzvektor von Nachbar x

Eigener Distanzvektor (von y)

Distanzvektor von Nachbar z

Eigentlich: 2 (über x) 0 (über y) 1 (über z)

Tabella von x

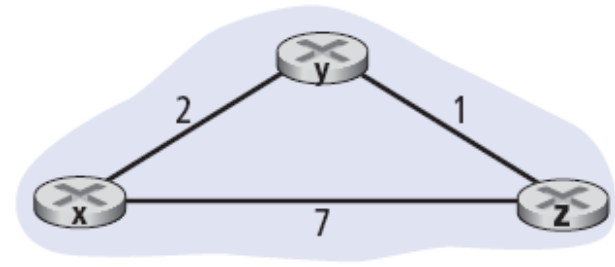
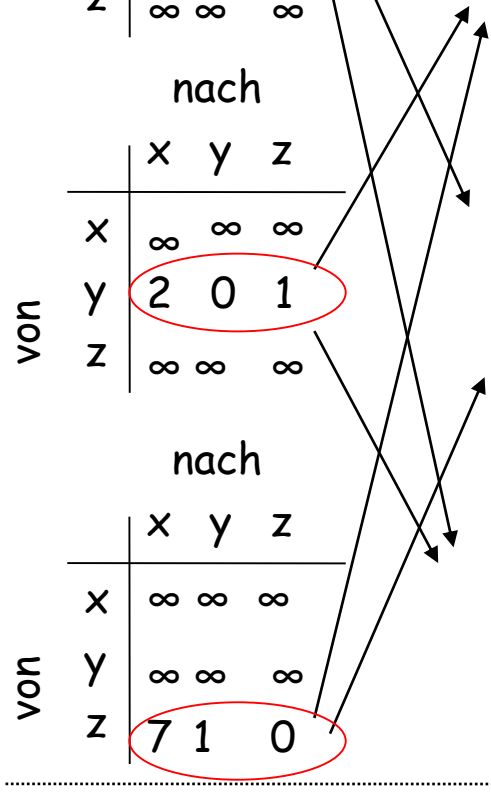
		nach		
		x	y	z
von	x	0	2	7
	y	∞	∞	∞
	z	∞	∞	∞

Tabella von y

		nach		
		x	y	z
von	x	∞	∞	∞
	y	2	0	1
	z	∞	∞	∞

Tabella von z

		nach		
		x	y	z
von	x	∞	∞	∞
	y	∞	∞	∞
	z	7	1	0



Zeit

$$D_x(y) = \min\{c(x,y)+D_y(y), c(x,z)+D_z(y)\}$$

$$= \min\{2+0, 7+1\} = 2$$

Tabelle von x

		nach		
		x	y	z
von	x	0	2	7
	y	∞	∞	∞
	z	∞	∞	∞

Tabelle von y

		nach		
		x	y	z
von	x	∞	∞	∞
	y	2	0	1
	z	∞	∞	∞

Tabelle von z

		nach		
		x	y	z
von	x	∞	∞	∞
	y	∞	∞	∞
	z	7	1	0

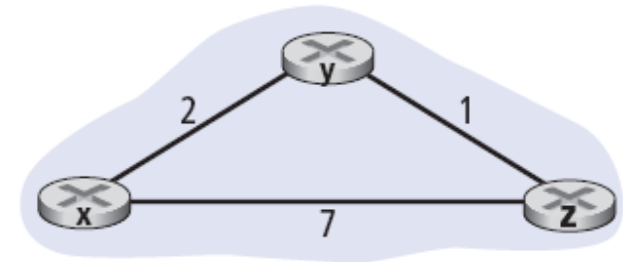
		nach		
		x	y	z
von	x	0	2	3
	y	2	0	1
	z	7	1	0

		nach		
		x	y	z
von	x	0	2	7
	y	2	0	1
	z	7	1	0

		nach		
		x	y	z
von	x	0	2	7
	y	2	0	1
	z	3	1	0

$$D_x(z) = \min\{c(x,y)+D_y(z), c(x,z)+D_z(z)\}$$

$$= \min\{2+1, 7+0\} = 3$$



Zeit

Tabelle von x

	nach		
von	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

Tabelle von y

	nach		
von	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

Tabelle von z

	nach		
von	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

	nach		
von	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

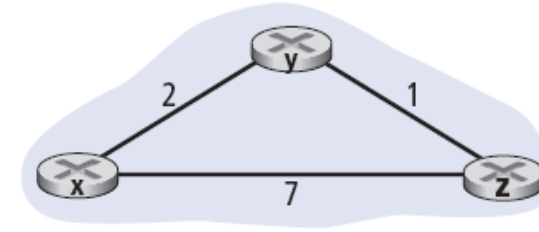
	nach		
von	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

	nach		
von	x	y	z
x	0	2	7
y	2	0	1
z	3	1	0

	nach		
von	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

	nach		
von	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

	nach		
von	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0



Zeit

Distance Vector: Beispiel

T=0

von A	via A	via B	via C	via D
zu A				
zu B		3		
zu C			23	
zu D				

von B	via A	via B	via C	via D
zu A	3			
zu B				
zu C			2	
zu D				

von C	via A	via B	via C	via D
zu A	23			
zu B		2		
zu C				
zu D				5

von D	via A	via B	via C	via D
zu A				
zu B				
zu C			5	
zu D				

T=1

von A	via A	via B	via C	via D
zu A				
zu B		3	25	
zu C		5	23	
zu D			28	

von B	via A	via B	via C	via D
zu A	3		25	
zu B				
zu C	26		2	
zu D			7	

von C	via A	via B	via C	via D
zu A	23	5		
zu B	26	2		
zu C				
zu D				5

von D	via A	via B	via C	via D
zu A			28	
zu B			7	
zu C			5	
zu D				

T=2

von A	via A	via B	via C	via D
zu A				
zu B		3	25	
zu C		5	23	
zu D		10	28	

von B	via A	via B	via C	via D
zu A	3		7	
zu B				
zu C	8		2	
zu D	31		7	

von C	via A	via B	via C	via D
zu A	23	5		33
zu B	26	2		12
zu C				
zu D	51	9		5

von D	via A	via B	via C	via D
zu A			10	
zu B			7	
zu C			5	
zu D				

T=3

von A	via A	via B	via C	via D
zu A				
zu B		3	25	
zu C		5	23	
zu D		10	28	

von B	via A	via B	via C	via D
zu A	3		7	
zu B				
zu C	8		2	
zu D	13		7	

von C	via A	via B	via C	via D
zu A	23	5		15
zu B	26	2		12
zu C				
zu D	33	9		5

von D	via A	via B	via C	via D
zu A			10	
zu B			7	
zu C			5	
zu D				

Vergleich von Link-State- und Distance-Vector-Routing

- Anzahl von Nachrichten:
 - LS: n Knoten, E Kanten:
 $O(nE)$ Nachrichten
 - DV: Nachrichten werden nur mit Nachbarn ausgetauscht
 - Zeit bis zur Konvergenz variiert
- Geschwindigkeit der Konvergenz
 - LS:
 - Fluten der Zustände der Links
 - Kann oszillieren
 - DV: variiert stark
 - Temporäre Routing-Schleifen sind möglich
 - Count-to-infinity-Problem
- Robustheit: Was passiert, wenn ein Router fehlerhaft ist?
 - LS:
 - Knoten kann falsche Kosten für einen Link fluten
 - Pfade möglicherweise nicht mehr optimal
 - DV:
 - Router kann falsche Kosten für einen ganzen Pfad ankündigen
 - Fehler propagiert durch das ganze Netzwerk, insbesondere wenn die (falschen) Kosten klein sind
 - u. U. katastrophal

Kapitel 4: Netzwerkschicht

- 4.1 Einleitung
- 4.2 Aufbau eines Routers
- 4.3 IP: Internet Protocol
 - Datagrammformat
 - IPv4-Adressierung
 - ICMP
 - IPv6
- 4.4 statisches Routen
- 4.5 Routing-Algorithmen
 - Link State
 - Distance Vector
 - Hierarchisches Routing
- 4.6 Routing im Internet
 - RIP
 - OSPF
 - BGP
- 4.7 Broadcast- und Multicast-Routing

Hierarchisches Routing: Idee

- Router werden zu Regionen zusammengefasst, diese nennt man autonome Systeme (AS)
- Router innerhalb eines AS verwenden ein Routing-Protokoll
 - "Intra-AS"-Routing-Protokoll
 - Router in verschiedenen AS können verschiedene Intra-AS-Routing-Protokolle verwenden
- Manchmal gilt:
 - Eine Organisation = ein AS
 - Es gibt aber auch Organisationen (z.B. einige ISPs), die aus mehreren AS bestehen
- Gateway-Router:
 - Ein Router in einem AS, der eine Verbindung zu einem Router in einem anderen AS hat
- Problem: Routing zwischen AS
 - „Inter-AS“-Routing-Protokoll

Kapitel 4: Netzwerkschicht

- 4.1 Einleitung
- 4.2 Aufbau eines Routers
- 4.3 IP: Internet Protocol
 - Datagrammformat
 - IPv4-Adressierung
 - ICMP
 - IPv6
- 4.5 statisches Routen
- 4.6 Routing-Algorithmen
 - Link State
 - Distance Vector
 - Hierarchisches Routing
- 4.7 Routing im Internet
 - RIP
 - OSPF
 - BGP
- 4.8 Broadcast- und Multicast-Routing

Kapitel 4:

Netzwerkschicht

- 4.1 Einleitung
- 4.2 Aufbau eines Routers
- 4.3 IP: Internet Protocol
 - Datagrammformat
 - IPv4-Adressierung
 - ICMP
 - IPv6
- 4.4 statisches Routen
- 4.5 Routing-Algorithmen
 - Link State
 - Distance Vector
 - Hierarchisches Routing
- 4.6 Routing im Internet
 - **RIP**
 - OSPF
 - BGP
- 4.7 Broadcast- und Multicast-Routing

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.0.1	255.255.255.255	RIPv1	66	Request
2	7.393000	10.0.0.1	255.255.255.255	RIPv1	66	Response
3	20.379000	10.0.0.2	255.255.255.255	RIPv1	66	Request
4	20.422000	10.0.0.1	10.0.0.2	RIPv1	66	Response
5	28.895000	10.0.0.1	255.255.255.255	RIPv1	66	Response
6	30.461000	10.0.0.2	255.255.255.255	RIPv1	66	Response
7	48.619000	10.0.0.2	255.255.255.255	RIPv1	66	Response
8	56.310000	10.0.0.1	255.255.255.255	RIPv1	66	Response
9	76.216000	10.0.0.2	255.255.255.255	RIPv1	66	Response
10	84.510000	10.0.0.1	255.255.255.255	RIPv1	66	Response
11	104.156000	10.0.0.2	255.255.255.255	RIPv1	66	Response
12	112.339000	10.0.0.1	255.255.255.255	RIPv1	66	Response
13	130.359000	10.0.0.2	255.255.255.255	RIPv1	66	Response
14	138.680000	10.0.0.1	255.255.255.255	RIPv1	66	Response
15	156.550000	10.0.0.2	255.255.255.255	RIPv1	66	Response
16	166.079000	10.0.0.1	255.255.255.255	RIPv1	66	Response
17	182.250000	10.0.0.2	255.255.255.255	RIPv1	66	Response
18	192.489000	10.0.0.1	255.255.255.255	RIPv1	66	Response

```

Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
Ethernet II, Src: ca:00:14:28:00:1c (ca:00:14:28:00:1c), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 10.0.0.1 (10.0.0.1), Dst: 255.255.255.255 (255.255.255.255)
User Datagram Protocol, Src Port: router (520), Dst Port: router (520)
Routing Information Protocol
  Command: Request (1)
  Version: RIPv1 (1)
  Address not specified, Metric: 16
    Address Family: Unspecified (0)
    Metric: 16

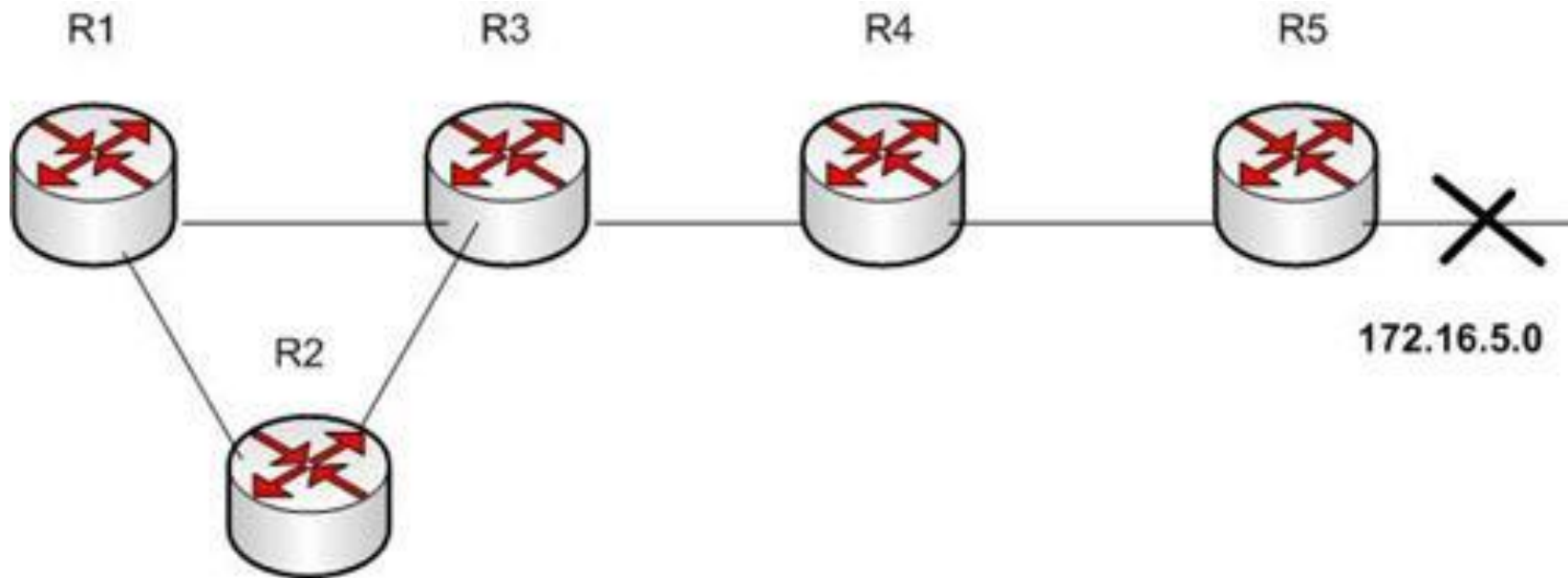
```

```

0000  ff ff ff ff ff ff ca 00 14 28 00 1c 08 00 45 c0  ..... .(....E.
0010  00 34 00 00 00 00 02 11 ad f9 0a 00 00 01 ff ff  .4.....
0020  ff ff 02 08 02 08 00 20 f0 8c 01 01 00 00 00 00  .....
0030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  .....
0040  00 10  ..

```

RIP-Count to Infinity



Kapitel 4:

Netzwerkschicht

- 4.1 Einleitung
- 4.2 Aufbau eines Routers
- 4.3 IP: Internet Protocol
 - Datagrammformat
 - IPv4-Adressierung
 - ICMP
 - IPv6
- 4.4 statisches Routen
- 4.5 Routing-Algorithmen
 - Link State
 - Distance Vector
 - Hierarchisches Routing
- 4.6 Routing im Internet
 - RIP
 - **OSPF**
 - BGP
- 4.7 Broadcast- und Multicast-Routing

OSPF (Open Shortest Path First)

- Version 2 spezifiziert in RFC 2328
- "Open": frei verfügbar
- „Shortest Path First“: verwendet Link-State-Routing-Algorithmus
 - Periodisches Fluten von Link-State-Paketen
 - Jeder Router kündigt seine Links an
 - Diese Ankündigungen werden geflutet
 - OSPF-Pakete werden direkt in IP-Pakete eingepackt
 - Topologie des Netzwerkes in jedem Router bekannt
 - Wird in einer sogenannten „Netzwerkkarte“ oder „Topology Map“ gespeichert
 - Routen werden mit Dijkstras Algorithmus berechnet

Eigenschaften von OSPF (nicht in RIP)

- Sicherheit: Alle OSPF-Nachrichten können authentifiziert werden
- Mehrere alternative Pfade auf Basis verschiedener Metriken können für jedes Ziel bestimmt werden
 - Für jeden Link können Gewichte mehrerer Metriken angegeben werden
 - Kosten für einen Satellitenlink gering (pro Paket), aber Latenzzeit hoch
- Integrierte Unterstützung für Multicast:
 - Multicast OSPF (MOSPF) verwendet die Netzwerkkarte, die auch OSPF in jedem Router benötigt
- Hierarchisches OSPF in größeren autonomen Systemen

Hierarchisches OSPF

- Zweistufige Hierarchie: Local Area, Backbone
 - Ein Backbone pro AS
 - Link-State-Advertisements werden nur in einer Local Area geflutet
 - Jeder Router in einer Local Area kennt deren detaillierte Topologie und die Richtung zu den Netzwerken der anderen Local Areas
- Area-Border-Router:
 - Zusammenfassen der Distanzen zu den Netzwerken in der eigenen Local Area
 - Ankündigen dieser Zusammenfassung an die anderen Area-Border-Router
 - Ankündigungen der Zusammenfassungen der anderen Area-Border-Router in der Local Area
- Backbone-Router: führe OSPF-Routing im Backbone durch
- Boundary-Router: stellt eine Verbindung zu anderen Autonomen Systemen her

Kapitel 4:

Netzwerkschicht

- 4.1 Einleitung
- 4.2 Aufbau eines Routers
- 4.3 IP: Internet Protocol
 - Datagrammformat
 - IPv4-Adressierung
 - ICMP
 - IPv6
- 4.4 statisches Routen
- 4.5 Routing-Algorithmen
 - Link State
 - Distance Vector
 - Hierarchisches Routing
- 4.6 Routing im Internet
 - RIP
 - OSPF
 - BGP
- 4.7 Broadcast- und Multicast-Routing

BGP (Border Gateway Protocol)

- Das Border Gateway Protocol, Version 4, ist der De-facto-Standard für Inter-AS-Routing im Internet
- Spezifiziert in RFC 4271
- BGP erlaubt es einem AS:
 - Informationen über die Erreichbarkeit von Netzen von seinen benachbarten Autonomen Systemen zu erhalten
 - Diese Informationen an die Router im Inneren des eigenen AS weiterzuleiten
 - „Gute“ Routen zu einem gewünschten Zielnetzwerk zu bestimmen, wobei die Qualität einer Route von den Informationen über die Erreichbarkeit und Politiken abhängig ist
- Außerdem ermöglicht BGP es einem AS, sein eigenes Netzwerk anzukündigen und so dessen Erreichbarkeit den anderen Autonomen Systemen im Internet mitzuteilen

Warum verschiedene Protokolle für Intra-AS- und Inter-AS-Routing?

- Politiken:
 - Inter-AS: Eine Organisation möchten kontrollieren, wie (und ob) der Verkehr anderer Organisationen durch das eigene Netzwerk geleitet wird
 - Intra-AS: eigener Verkehr, eigene Administration, hier sind keine Politiken nötig
- Größenordnung:
 - Hierarchisches Routing reduziert die Größe der Routing-Tabellen und reduziert den Netzwerkverkehr für Routing-Updates
 - Dringend notwendig für Inter-AS-Routing, nicht wichtig in Intra-AS-Routing
- Performance:
 - Intra-AS: kann sich auf Performance konzentrieren
 - Inter-AS: Politiken können wichtiger sein als Performance

Visualisierung BGP

The screenshot shows the RIPE NCC website interface. At the top, the browser address bar displays <https://stat.ripe.net/widget/bgplay>. The RIPE NCC logo and name are visible on the left. A search bar on the right contains the text "bgplay". Below the logo, there are navigation tabs: "Manage IPs and ASNs", "Analyse" (highlighted), "Participate", "Get Support", and "Publications". A breadcrumb trail reads: "You are here: Home > Analyse > Statistics > RIPEstat > bgplay". On the left side, there is a sidebar menu with items: "RIPEstat Home", "About RIPEstat", "Documentation", and "Use Cases". The main content area features a "BGPlay" widget. The widget has a title "BGPlay" and a message: "Reload this widget by entering a resource here". Below the message are buttons for "source data", "embed code", "permalink", and "info".