

# Kapitel 3

## Routing im Internet

2. Vorlesung  
18.04.2005  
Autor1  
(nicolas(at)upb.de)  
Autor2  
(mcclaus(at)upb.de)

### 3.1 IP-Paket

Das IP-Paket enthält die nötigen Informationen, um die enthaltenen Daten durch das Netzwerk zu seinem Ziel weiterleiten zu können. Somit findet man IP-Pakete in der dritten Schicht im ISO-OSI Modell, der Vermittlungsschicht. Ein IP-Paket besteht aus zwei Teilen, dem Header und der Nutzlast. Die Nutzlast ist der Bereich, in dem die Daten von höheren Schichten liegen. Im Header befinden sich alle protokollrelevanten Informationen. Bei IPv4 ist er so aufgebaut:

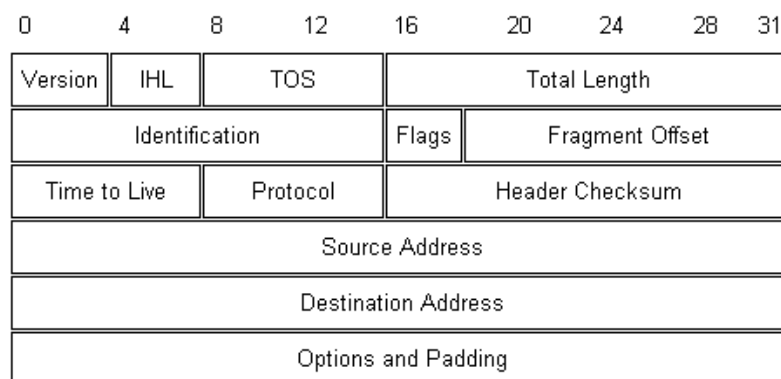


Abbildung 3.1: IP-Paket-Header

<b>Feld</b>	<b>Bedeutung</b>
Version	Gibt die Version an. Hier sind zur Zeit 4 und 6 möglich
IHL (Internet Header Length)	Gibt die Länge des Headers in Vielfachen von 32 Bit an
TOS (Type of service)	Dieses Feld kann für Priorisierung von Diensten gesetzt werden: Bits 0-2: Präzedenz Bit 3: 0/1 normale/geringe Verzögerung Bit 4: 0/1 normaler/hoher Durchsatz Bit 5: 0/1 normale/hohe Verlässlichkeit Bits 6 und 7: reserviert für zukünftige Anwendungen
Total Length	Gibt die Länge des gesamten Pakets an
Identification	Ermöglicht zusammen mit den Feldern Flags und Fragment Offset fragmentierte Pakete wieder zusammen zu setzen
Flags	Sind wie folgt gesetzt: Bit 0: reserviert, muss 0 sein Bit 1 (DF): 0/1 darf/darf nicht zerlegt (fragmentiert) werden Bit 2 (MF): 0/1 letztes Fragment/weitere Fragmente folgen
Fragment Offset	Bei fragmentierten Paketen gibt diese Nummer an, an welche Stelle im Fragment dieses Paket gehört
Time to Live	Hier wird angegeben, wie lange das Paket weiter geleitet werden darf, bevor es verworfen wird. Jede Station setzt diesen Wert herunter, bis er Null ist
Protocol	Dieses Feld gibt an, was das Folgeprotokoll ist. Z.B. 0x06 für TCP
Header Checksum	Eine Checksumme für den Header
Source Address	Die Zieladresse
Destination Address	Die Quelladresse
Options and Padding	Zusätzliche Optionen

IP wurde im RFC 791 definiert.

## 3.2 ICMP, Ping & Traceroute

Das Internet Control Message Protocol (ICMP) benutzt das Internet Protokoll zum Austausch von Fehlern und Informationen in Netzwerken. Obwohl es eigentlich eine Ebene über IP angeordnet ist, ist es in IP integriert, so dass von allen Routern erwartet wird, dass sie ICMP-Daten auswerten können. Sie werden in den Datenteil eines IP-Paketes geschrieben und der Servicetyp wird im Header auf 0 gesetzt. Die Protokollnummer ist immer 1. Ein Beispiel für Informationen, die mit ICMP verteilt werden, ist wenn ein Router den Sender benachrichtigt, dass er Pakete von ihm verwirft, weil das Ziel nicht erreichbar ist. Es gibt folgende ICMP-Nachrichten:

1. Echo Reply
- 2.
- 3.
4. Destination Unreachable
5. Source Quench
6. Redirect
- 7.
- 8.
9. Echo Request
10. Router Advertisement
11. Router Solicitation
12. Time Exceeded
13. Parameter Problem
14. Timestamp (erleichtert die Synchronisation von Uhren)
15. Timestamp Reply
16. Information Request
17. Information Reply
18. Address Mask Request
19. Address Mask Reply
20. Reserved (for Security)
21. Reserved (for Robustness Experiment)
22. Reserved (for Robustness Experiment)
23. Reserved (for Robustness Experiment)
24. Reserved (for Robustness Experiment)
25. Reserved (for Robustness Experiment)
26. Reserved (for Robustness Experiment)
27. Reserved (for Robustness Experiment)

- 28. Reserved (for Robustness Experiment)
- 29. Reserved (for Robustness Experiment)
- 30. Reserved (for Robustness Experiment)
- 31. Traceroute
- 32. Datagram Conversion Error
- 33. Mobile Host Redirect
- 34. IPv6 Where-Are-You
- 35. IPv6 I-Am-Here
- 36. Mobile Registration Request
- 37. Mobile Registration Reply
- 38. Domain Name Request
- 39. Domain Name Reply
- 40. SKIP
- 41. Photuris
- 42. ICMP messages utilized by experimental mobility protocols such as Seamoby
- 43. - 255. Reserved

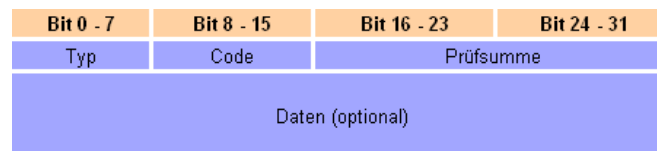


Abbildung 3.2: icmp-Paket-Header

Ping ist ein Programm, das ursprünglich 1983 von Mike Muus entwickelt wurde um die Erreichbarkeit von Computern zu testen. Es sendet ein ICMP-Echo-Request-Paket an den Empfänger, der darauf mit einem ICMP Echo-Reply antwortet. Der Name stammt aus dem Militärischen Bereich. Beim Einsatz von Sonar wird ein Schallsignal ausgesandt, das sich in einem U-Boot wie ein hohes Klopfgeräusch anhört. Lautmalerisch wird dieses Geräusch als *ping* bezeichnet. Beim Aufruf von Traceroute werden ICMP-Pakete zu einer Zieladresse gesendet. Jedes Paket erhält eine um 1 erhöhte TTL, beginnend mit 1. Anschließend werden die ICMP-Nachrichten von den einzelnen Routern ausgewertet, so dass am Ende eine Liste aller Router auf dem Weg zum Ziel zustande kommt.

## 3.3 Routingtabelle und Paketweiterleitung

### 3.3.1 IP-Routingtabelle

Sowohl Router als auch Computer benutzen die sogenannten Routingtabellen um Wege für Pakete zu finden. Dazu werden die Ziel-IP-Adressen in eine Host-IP-Adresse umgewandelt. Falls sich der Host im gleichen Subnetz befindet sind die beiden Adressen identisch. Andernfalls, ist die Host-IP-Adresse die IP-Adresse des nächsten Routers. Es gibt zwei mögliche Arten die Tabelle zu erzeugen: statisch oder dynamisch.

### 3.3.2 Packet Forwarding

Das Packet Forwarding wurde früher Packet Routing genannt. Gemeint ist damit der folgende Ablauf: Wenn die Adresse eines hereingekommenen Paketes die eigene ist, dann wird die Nachricht lokal weiterverarbeitet. Wenn für die Adresse oder das Subnetz des Paketes ein Eintrag in der Routingtabelle ist, sende es an die angegebene Adresse weiter. In allen anderen Fällen, sende das Paket an das Default Gateway. Wenn das Paket weitergeleitet werden soll, wird zunächst die time to live ausgewertet. Diese gibt an, wie lange das Paket maximal unterwegs sein darf, bis es verworfen wird. In diesem Fall wird eine ICMP-Nachricht erzeugt, die den Sender über das verworfene Paket informiert.

### 3.3.3 Statisches Routing

Beim statischen Routing wird die Routingtabelle von Hand erzeugt. Es wird also explizit eingetragen, welche Pakete wohin gehen sollen. Daher ist statisches Routing nur für kleine, stabile Netze sinnvoll.

### 3.3.4 Dynamisches Routing

Beim dynamischen Routing wird die Routingtabelle automatisch generiert und bei Bedarf auch automatisch an neuen Gegebenheiten angepasst.

#### Kürzeste Wege Problem

Gegeben ist ein gerichteter Graph  $G$ , ein Startknoten  $s$  und Kantengewichte  $w$  für alle Kanten. Gesucht wird der Pfad vom Startknoten zu allen anderen Knoten, der jeweils die geringste Summe an Kantengewichten aufweist. Als Lösung ergibt sich ein Baum, der den Startknoten als Wurzel enthält.

#### Dijkstra Algorithmus

Eine Möglichkeit kürzeste Wege für Graphen mit nicht negativen Kantengewichten zu berechnen ist der Dijkstra Algorithmus.

```

DIJKSTRA( $G, w, s$ )
01 für jedes  $v$  aus  $V$ 
02   Distanz( $v$ ) := unendlich, Vorgänger( $v$ ) := kein
03 Distanz( $s$ ) := 0, Vorgänger( $s$ ) :=  $s$ ,  $U$  :=  $V$ 
04
05 solange  $U$  nicht leer
06   wähle  $u$  aus  $U$  mit Distanz( $u$ ) minimal
07    $U$  :=  $U - \{u\}$ 
08   wenn  $u = z$  dann STOP #optional
09   für jedes  $(u,v)$  aus  $E$  mit  $v$  aus  $U$ 
10     wenn Distanz( $u$ ) + Kosten( $u,v$ ) < Distanz( $v$ ) dann
11       Distanz( $v$ ) := Distanz( $u$ ) + Kosten( $u,v$ )
12       Vorgänger( $v$ ) :=  $u$ 

```

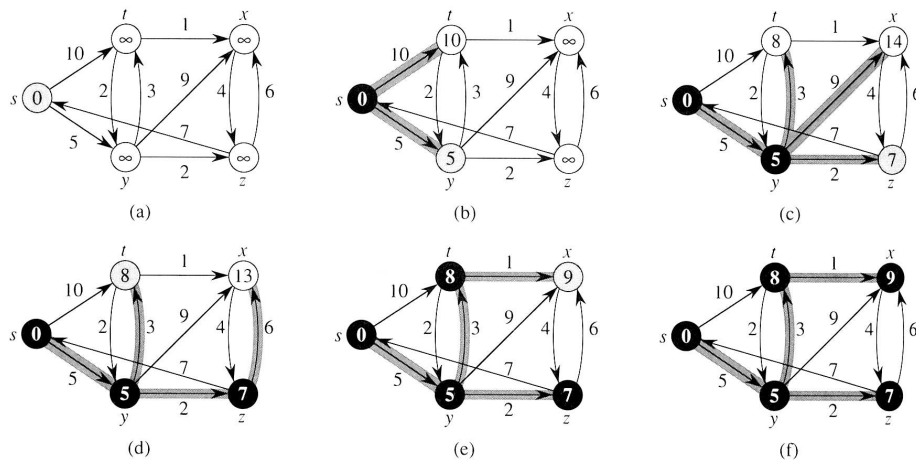


Abbildung 3.3: Verlauf des Dijkstra-Algorithmus aus [THCS01]

Der Algorithmus wählt als nächstes immer die Kante mit dem leichtesten Gewicht. Daher ist er Greedy-Algorithmus und hat eine Verwandtschaft zur Breitensuche. Fall der Graph nicht zusammenhängend oder gerichtet und nicht stark zusammenhängend ist, kann der Pfad zu einigen Knoten unendlich lang sein. Für die Bestimmung der Laufzeit sei  $m$  die Anzahl der Kanten und  $n$  die Anzahl der Knoten. Der Algorithmus muss  $n$  mal den nächsten minimalen Knoten bestimmen. Wenn die Knoten dafür in einem Heap gespeichert werden dauert ein Zugriff darauf  $n \cdot \log(n)$ . Somit ist die Gesamtlaufzeit  $O(m+n \cdot \log(n))$

### Bellman-Ford-Algorithmus

Falls der Graph in dem man den kürzesten Pfad bestimmen möchte negative Kantengewichte enthält, kann der Dijkstra-Algorithmus nicht angewendet werden. In diesem Fall können die kürzesten Wege mit dem Bellmann-Ford-Algorithmus bestimmt werden. In dem Graph dürfen keine Kreise mit negativer Länge vorhanden sein, da sonst zu jedem Knoten ein beliebig kurzer Pfad bestimmt werden kann.

#### Bellman-Ford-Algorithmus

```

01 für jedes  $v$  aus  $V$ 
02     Distanz( $v$ ) := unendlich, Vorgänger( $v$ ) := kein
03 Distanz( $s$ ) := 0, Vorgänger( $s$ ) :=  $s$ ,  $U$  :=  $V$ 
04
05 solange  $U$  nicht leer
06     wähle  $u$  aus  $U$ 
07      $U$  :=  $U - \{u\}$ 
08     fr jedes  $(u,v)$  aus  $E$  wiederhole  $V$ -mal
09         wenn Distanz( $u$ ) + Kosten( $u,v$ ) < Distanz( $v$ ) dann
10             Distanz( $v$ ) := Distanz( $u$ ) + Kosten( $u,v$ )
11             Vorgänger( $v$ ) :=  $u$ 
12              $U$  :=  $U - \{v\}$ 
13 für jedes  $(u,v)$  aus  $E'$ 
14     wenn Distanz( $u$ ) + Kosten( $u,v$ ) < Distanz( $v$ ) dann
15         STOP mit Ausgabe "Kreis negativer Länge gefunden"
```

Auch dieser Algorithmus ist ein Greedy-Algorithmus. Er prüft aber die Länge der Wege aller bereits bearbeiteten Knoten ständig auf ihre Optimalität hin. Die Laufzeit des Algorithmus beträgt  $O(m \cdot n)$

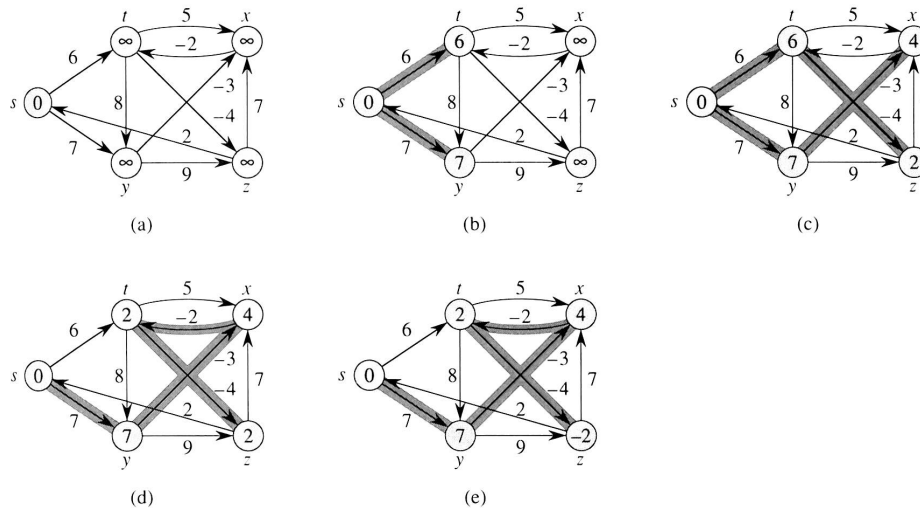


Abbildung 3.4: Verlauf des Bellman-Ford-Algorithmus aus [THCS01]

### Link State Routing

Um Algorithmen, wie den Dijkstra anwenden zu können benötigt ein Knoten die Informationen über das Netzwerk. Die Lösung für dieses Problem: Alle Knoten machen einen Broadcast an alle anderen Knoten, in dem diese Informationen enthalten sind. Das Link State Verfahren ist eine Möglichkeit die Routingtabelle eines Routers zu füllen. Bei diesem Verfahren werden, anders als bei Distant Vector, wenig Informationen über weite Strecken ausgetauscht. Um die Informationen über das Netzwerk auszutauschen, schickt ein Router lokale Änderungen per Multicast in das Netzwerk. Dadurch kann sehr schnell auf Änderungen des Netzwerks reagiert werden.

### Distance-Vector Routing

Bei diesem Verfahren werden im Gegensatz zum Link State Routing viele Informationen über kurze Strecken übertragen. Das Verfahren ist vergleichsweise einfach zu implementieren. Ein Nachteil gegenüber dem Link State Routing ist seine schlechte Skalierbarkeit. Das Verfahren hat folgenden Ablauf:

1. Erzeuge eine Kostenmatrix, in der alle Ziele mit ihren Kosten aufgeführt sind. Anfangs sind dies nur die direkten Nachbarn.
2. Suche zu allen Zielen den Weg mit den geringsten Kosten. Schicke die Matrix an alle direkten Nachbarn.
3. Pflege die Matrizen, die von Nachbarn kommen in die eigene Matrix ein.
4. Wenn sich die Kosten dadurch zu einem Ziel verringern fahre mit 2. fort, sonst 3.

Ein Problem, das beim Distant Vector Routing auftritt ist das Count to infinity Problem. Das Problem tritt zum Beispiel auf, wenn ein Router ausfällt. Sein direkter Nachbar A kann ihn nicht mehr erreichen. Ein Nachbar von A, zum Beispiel B meint aber er könne den ausgefallenen Router noch über A erreichen und teilt A mit, dass er dies kann. A meint darauf hin, es gibt noch eine Route zum ausgefallenen Router und pflegt die Entfernung+1 in seine Matrix ein. Diese schickt er an B. B weiß, dass der ausgefallenen Router über A erreichbar ist und erhöht die Kosten ebenfalls. Dies wird wieder an A geschickt, der die Kosten wieder erhöht usw. Hier wird deutlich, dass sich schlechte Nachrichten nur langsam verbreiten. Dagegen breiten sich gute Nachrichten sehr schnell aus. Ein Verfahren mit dem sich das Count to infinity Problem lösen lässt ist Poisoned Reverse oder Split Horizon.

Poisoned Reverse: Wenn ein Router die Verbindung zu einem anderen Router verliert, teilt er dies den anderen Routern mit, indem er die verlorene Verbindung mit Länge unendlich angibt. Somit spricht sich schnell herum, dass der fehlende Router nicht mehr erreichbar ist.

Split Horizon: Bei diesem Verfahren wird ein Router daran gehindert eine Route zu einem bestimmten Ziel zurück an den Router zu übermitteln, von dem er diese Route gelernt hat. Dadurch kann ein Router A, der die Verbindung zu einem anderen Router Z verloren hat, nicht seinen Nachbarn B fragen, wie man nach Z kommt, solange B davon ausgeht, dass dies über A geht.

Beide Verfahren haben das Problem, dass sie bei längeren Zyklen nicht mehr zuverlässig funktionieren.

Beispiel: Der beste Pfad ist grün dargestellt. Ein neuer bester Pfad, der im nächsten Schritt an die Nachbarn geschickt wird ist gelb.

T=0	<table border="1"> <thead> <tr><th>von</th><th>via</th><th>via</th><th>via</th><th>via</th></tr> <tr><th>A</th><th>A</th><th>B</th><th>C</th><th>D</th></tr> </thead> <tbody> <tr><td>zu A</td><td></td><td></td><td></td><td></td></tr> <tr><td>zu B</td><td></td><td>3</td><td></td><td></td></tr> <tr><td>zu C</td><td></td><td></td><td>23</td><td></td></tr> <tr><td>zu D</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	von	via	via	via	via	A	A	B	C	D	zu A					zu B		3			zu C			23		zu D					<table border="1"> <thead> <tr><th>von</th><th>via</th><th>via</th><th>via</th><th>via</th></tr> <tr><th>B</th><th>A</th><th>B</th><th>C</th><th>D</th></tr> </thead> <tbody> <tr><td>zu A</td><td>3</td><td></td><td></td><td></td></tr> <tr><td>zu B</td><td></td><td></td><td></td><td></td></tr> <tr><td>zu C</td><td></td><td></td><td>2</td><td></td></tr> <tr><td>zu D</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	von	via	via	via	via	B	A	B	C	D	zu A	3				zu B					zu C			2		zu D					<table border="1"> <thead> <tr><th>von</th><th>via</th><th>via</th><th>via</th><th>via</th></tr> <tr><th>C</th><th>A</th><th>B</th><th>C</th><th>D</th></tr> </thead> <tbody> <tr><td>zu A</td><td>23</td><td></td><td></td><td></td></tr> <tr><td>zu B</td><td></td><td>2</td><td></td><td></td></tr> <tr><td>zu C</td><td></td><td></td><td></td><td></td></tr> <tr><td>zu D</td><td></td><td></td><td></td><td>5</td></tr> </tbody> </table>	von	via	via	via	via	C	A	B	C	D	zu A	23				zu B		2			zu C					zu D				5	<table border="1"> <thead> <tr><th>von</th><th>via</th><th>via</th><th>via</th><th>via</th></tr> <tr><th>D</th><th>A</th><th>B</th><th>C</th><th>D</th></tr> </thead> <tbody> <tr><td>zu A</td><td></td><td></td><td></td><td></td></tr> <tr><td>zu B</td><td></td><td></td><td></td><td></td></tr> <tr><td>zu C</td><td></td><td></td><td>5</td><td></td></tr> <tr><td>zu D</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	von	via	via	via	via	D	A	B	C	D	zu A					zu B					zu C			5		zu D				
	von	via	via	via	via																																																																																																																							
	A	A	B	C	D																																																																																																																							
	zu A																																																																																																																											
zu B		3																																																																																																																										
zu C			23																																																																																																																									
zu D																																																																																																																												
von	via	via	via	via																																																																																																																								
B	A	B	C	D																																																																																																																								
zu A	3																																																																																																																											
zu B																																																																																																																												
zu C			2																																																																																																																									
zu D																																																																																																																												
von	via	via	via	via																																																																																																																								
C	A	B	C	D																																																																																																																								
zu A	23																																																																																																																											
zu B		2																																																																																																																										
zu C																																																																																																																												
zu D				5																																																																																																																								
von	via	via	via	via																																																																																																																								
D	A	B	C	D																																																																																																																								
zu A																																																																																																																												
zu B																																																																																																																												
zu C			5																																																																																																																									
zu D																																																																																																																												
T=1	<table border="1"> <thead> <tr><th>von</th><th>via</th><th>via</th><th>via</th><th>via</th></tr> <tr><th>A</th><th>A</th><th>B</th><th>C</th><th>D</th></tr> </thead> <tbody> <tr><td>zu A</td><td></td><td></td><td></td><td></td></tr> <tr><td>zu B</td><td></td><td>3</td><td>25</td><td></td></tr> <tr><td>zu C</td><td></td><td>5</td><td>23</td><td></td></tr> <tr><td>zu D</td><td></td><td></td><td>28</td><td></td></tr> </tbody> </table>	von	via	via	via	via	A	A	B	C	D	zu A					zu B		3	25		zu C		5	23		zu D			28		<table border="1"> <thead> <tr><th>von</th><th>via</th><th>via</th><th>via</th><th>via</th></tr> <tr><th>B</th><th>A</th><th>B</th><th>C</th><th>D</th></tr> </thead> <tbody> <tr><td>zu A</td><td>3</td><td></td><td>25</td><td></td></tr> <tr><td>zu B</td><td></td><td></td><td></td><td></td></tr> <tr><td>zu C</td><td>26</td><td></td><td>2</td><td></td></tr> <tr><td>zu D</td><td></td><td></td><td>7</td><td></td></tr> </tbody> </table>	von	via	via	via	via	B	A	B	C	D	zu A	3		25		zu B					zu C	26		2		zu D			7		<table border="1"> <thead> <tr><th>von</th><th>via</th><th>via</th><th>via</th><th>via</th></tr> <tr><th>C</th><th>A</th><th>B</th><th>C</th><th>D</th></tr> </thead> <tbody> <tr><td>zu A</td><td>23</td><td>5</td><td></td><td></td></tr> <tr><td>zu B</td><td>26</td><td>2</td><td></td><td></td></tr> <tr><td>zu C</td><td></td><td></td><td></td><td></td></tr> <tr><td>zu D</td><td></td><td></td><td></td><td>5</td></tr> </tbody> </table>	von	via	via	via	via	C	A	B	C	D	zu A	23	5			zu B	26	2			zu C					zu D				5	<table border="1"> <thead> <tr><th>von</th><th>via</th><th>via</th><th>via</th><th>via</th></tr> <tr><th>D</th><th>A</th><th>B</th><th>C</th><th>D</th></tr> </thead> <tbody> <tr><td>zu A</td><td></td><td></td><td>28</td><td></td></tr> <tr><td>zu B</td><td></td><td></td><td>7</td><td></td></tr> <tr><td>zu C</td><td></td><td></td><td>5</td><td></td></tr> <tr><td>zu D</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	von	via	via	via	via	D	A	B	C	D	zu A			28		zu B			7		zu C			5		zu D				
	von	via	via	via	via																																																																																																																							
	A	A	B	C	D																																																																																																																							
	zu A																																																																																																																											
zu B		3	25																																																																																																																									
zu C		5	23																																																																																																																									
zu D			28																																																																																																																									
von	via	via	via	via																																																																																																																								
B	A	B	C	D																																																																																																																								
zu A	3		25																																																																																																																									
zu B																																																																																																																												
zu C	26		2																																																																																																																									
zu D			7																																																																																																																									
von	via	via	via	via																																																																																																																								
C	A	B	C	D																																																																																																																								
zu A	23	5																																																																																																																										
zu B	26	2																																																																																																																										
zu C																																																																																																																												
zu D				5																																																																																																																								
von	via	via	via	via																																																																																																																								
D	A	B	C	D																																																																																																																								
zu A			28																																																																																																																									
zu B			7																																																																																																																									
zu C			5																																																																																																																									
zu D																																																																																																																												
T=2	<table border="1"> <thead> <tr><th>von</th><th>via</th><th>via</th><th>via</th><th>via</th></tr> <tr><th>A</th><th>A</th><th>B</th><th>C</th><th>D</th></tr> </thead> <tbody> <tr><td>zu A</td><td></td><td></td><td></td><td></td></tr> <tr><td>zu B</td><td></td><td>3</td><td>25</td><td></td></tr> <tr><td>zu C</td><td></td><td>5</td><td>23</td><td></td></tr> <tr><td>zu D</td><td></td><td>10</td><td>28</td><td></td></tr> </tbody> </table>	von	via	via	via	via	A	A	B	C	D	zu A					zu B		3	25		zu C		5	23		zu D		10	28		<table border="1"> <thead> <tr><th>von</th><th>via</th><th>via</th><th>via</th><th>via</th></tr> <tr><th>B</th><th>A</th><th>B</th><th>C</th><th>D</th></tr> </thead> <tbody> <tr><td>zu A</td><td>3</td><td></td><td>7</td><td></td></tr> <tr><td>zu B</td><td></td><td></td><td></td><td></td></tr> <tr><td>zu C</td><td>8</td><td></td><td>2</td><td></td></tr> <tr><td>zu D</td><td>31</td><td></td><td>7</td><td></td></tr> </tbody> </table>	von	via	via	via	via	B	A	B	C	D	zu A	3		7		zu B					zu C	8		2		zu D	31		7		<table border="1"> <thead> <tr><th>von</th><th>via</th><th>via</th><th>via</th><th>via</th></tr> <tr><th>C</th><th>A</th><th>B</th><th>C</th><th>D</th></tr> </thead> <tbody> <tr><td>zu A</td><td>23</td><td>5</td><td></td><td>33</td></tr> <tr><td>zu B</td><td>26</td><td>2</td><td></td><td>12</td></tr> <tr><td>zu C</td><td></td><td></td><td></td><td></td></tr> <tr><td>zu D</td><td>51</td><td>9</td><td></td><td>5</td></tr> </tbody> </table>	von	via	via	via	via	C	A	B	C	D	zu A	23	5		33	zu B	26	2		12	zu C					zu D	51	9		5	<table border="1"> <thead> <tr><th>von</th><th>via</th><th>via</th><th>via</th><th>via</th></tr> <tr><th>D</th><th>A</th><th>B</th><th>C</th><th>D</th></tr> </thead> <tbody> <tr><td>zu A</td><td></td><td></td><td>10</td><td></td></tr> <tr><td>zu B</td><td></td><td></td><td>7</td><td></td></tr> <tr><td>zu C</td><td></td><td></td><td>5</td><td></td></tr> <tr><td>zu D</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	von	via	via	via	via	D	A	B	C	D	zu A			10		zu B			7		zu C			5		zu D				
	von	via	via	via	via																																																																																																																							
	A	A	B	C	D																																																																																																																							
	zu A																																																																																																																											
zu B		3	25																																																																																																																									
zu C		5	23																																																																																																																									
zu D		10	28																																																																																																																									
von	via	via	via	via																																																																																																																								
B	A	B	C	D																																																																																																																								
zu A	3		7																																																																																																																									
zu B																																																																																																																												
zu C	8		2																																																																																																																									
zu D	31		7																																																																																																																									
von	via	via	via	via																																																																																																																								
C	A	B	C	D																																																																																																																								
zu A	23	5		33																																																																																																																								
zu B	26	2		12																																																																																																																								
zu C																																																																																																																												
zu D	51	9		5																																																																																																																								
von	via	via	via	via																																																																																																																								
D	A	B	C	D																																																																																																																								
zu A			10																																																																																																																									
zu B			7																																																																																																																									
zu C			5																																																																																																																									
zu D																																																																																																																												
T=3	<table border="1"> <thead> <tr><th>von</th><th>via</th><th>via</th><th>via</th><th>via</th></tr> <tr><th>A</th><th>A</th><th>B</th><th>C</th><th>D</th></tr> </thead> <tbody> <tr><td>zu A</td><td></td><td></td><td></td><td></td></tr> <tr><td>zu B</td><td></td><td>3</td><td>25</td><td></td></tr> <tr><td>zu C</td><td></td><td>5</td><td>23</td><td></td></tr> <tr><td>zu D</td><td></td><td>10</td><td>28</td><td></td></tr> </tbody> </table>	von	via	via	via	via	A	A	B	C	D	zu A					zu B		3	25		zu C		5	23		zu D		10	28		<table border="1"> <thead> <tr><th>von</th><th>via</th><th>via</th><th>via</th><th>via</th></tr> <tr><th>B</th><th>A</th><th>B</th><th>C</th><th>D</th></tr> </thead> <tbody> <tr><td>zu A</td><td>3</td><td></td><td>7</td><td></td></tr> <tr><td>zu B</td><td></td><td></td><td></td><td></td></tr> <tr><td>zu C</td><td>8</td><td></td><td>2</td><td></td></tr> <tr><td>zu D</td><td>13</td><td></td><td>7</td><td></td></tr> </tbody> </table>	von	via	via	via	via	B	A	B	C	D	zu A	3		7		zu B					zu C	8		2		zu D	13		7		<table border="1"> <thead> <tr><th>von</th><th>via</th><th>via</th><th>via</th><th>via</th></tr> <tr><th>C</th><th>A</th><th>B</th><th>C</th><th>D</th></tr> </thead> <tbody> <tr><td>zu A</td><td>23</td><td>5</td><td></td><td>15</td></tr> <tr><td>zu B</td><td>26</td><td>2</td><td></td><td>12</td></tr> <tr><td>zu C</td><td></td><td></td><td></td><td></td></tr> <tr><td>zu D</td><td>33</td><td>9</td><td></td><td>5</td></tr> </tbody> </table>	von	via	via	via	via	C	A	B	C	D	zu A	23	5		15	zu B	26	2		12	zu C					zu D	33	9		5	<table border="1"> <thead> <tr><th>von</th><th>via</th><th>via</th><th>via</th><th>via</th></tr> <tr><th>D</th><th>A</th><th>B</th><th>C</th><th>D</th></tr> </thead> <tbody> <tr><td>zu A</td><td></td><td></td><td>10</td><td></td></tr> <tr><td>zu B</td><td></td><td></td><td>7</td><td></td></tr> <tr><td>zu C</td><td></td><td></td><td>5</td><td></td></tr> <tr><td>zu D</td><td></td><td></td><td></td><td></td></tr> </tbody> </table>	von	via	via	via	via	D	A	B	C	D	zu A			10		zu B			7		zu C			5		zu D				
	von	via	via	via	via																																																																																																																							
	A	A	B	C	D																																																																																																																							
	zu A																																																																																																																											
zu B		3	25																																																																																																																									
zu C		5	23																																																																																																																									
zu D		10	28																																																																																																																									
von	via	via	via	via																																																																																																																								
B	A	B	C	D																																																																																																																								
zu A	3		7																																																																																																																									
zu B																																																																																																																												
zu C	8		2																																																																																																																									
zu D	13		7																																																																																																																									
von	via	via	via	via																																																																																																																								
C	A	B	C	D																																																																																																																								
zu A	23	5		15																																																																																																																								
zu B	26	2		12																																																																																																																								
zu C																																																																																																																												
zu D	33	9		5																																																																																																																								
von	via	via	via	via																																																																																																																								
D	A	B	C	D																																																																																																																								
zu A			10																																																																																																																									
zu B			7																																																																																																																									
zu C			5																																																																																																																									
zu D																																																																																																																												

Abbildung 3.5: Distance-Vector-Routing

Erläuterung der Vorgänge im Router A:

**T=0** Wir erzeugen die initiale Kostenmatrix. Sie enthält nur unsere direkten Nachbarn B und C mit den uns bekannten Kosten. Wir schicken daraufhin unsere neuen besten Pfade (B mit Kosten 3, C mit Kosten 23) an unsere direkten Nachbarn



- T=1** Wir haben von den Routern B und C Datenpakete erhalten und wissen jetzt, zu welchen Kosten wir D und wie wir C und B jeweils auch erreichen können. Im Fall der Zielrouter C und D ist das sogar ein neuer bester Pfad, den wir im nächsten Schritt an unsere Nachbarn übertragen werden
- T=2** Wir haben von Router B ein Datenpaket erhalten und wissen jetzt, dass B den Router D günstiger erreichen kann. Wir tragen die Kosten in unsere Matrix ein und werden diesen neuen besten Pfad wieder an unsere Nachbarn verbreiten
- T=3** Wir haben keine neuen Informationen mehr empfangen; unsere besten Pfade haben sich nicht geändert und wir senden keine neuen Informationen an unsere Nachbarn. Denen geht es genauso - der Algorithmus terminiert

### Probleme und Grenzen des flachen Routing

Sowohl Link State Routing als auch Distance Vector Routing benötigen  $O(gn)$  Einträge für  $n$  Router mit Grad  $g$ . Die Konvergenzzeit steigt bei zunehmender Größe an. Für einen einzelnen Router sind die Router, die „weit weg“ sind von ihm eher uninteressant. Für ihn reicht es, wenn er ein Paket in die „Nähe“ des Ziels bringt. Die Router in der Gegend werden das Ziel schon genauer kennen. Da es im Internet ca.  $10^6$  Router gibt müssen für das Routing Hierarchien eingeführt werden.

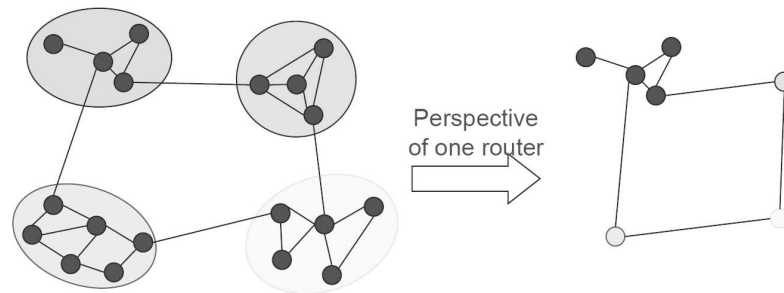


Abbildung 3.6: Hierarchien aus [Kar04]

Ein Router behandelt alle Ziele in einer fremden Region als ein Ziel. Pakete, die in die eigene Region sollen werden dagegen ganz normal geroutet. Der Preis, den man für ein hierarchisches Routing bezahlen muss ist, dass es vorkommen kann, dass die Routen nicht mehr optimal sind, da Details des Netzwerkes fehlen. Im IP gibt es Class A, B, C, D und E Netze. Diese unterscheiden sich hauptsächlich durch ihre Größe und Anzahl.

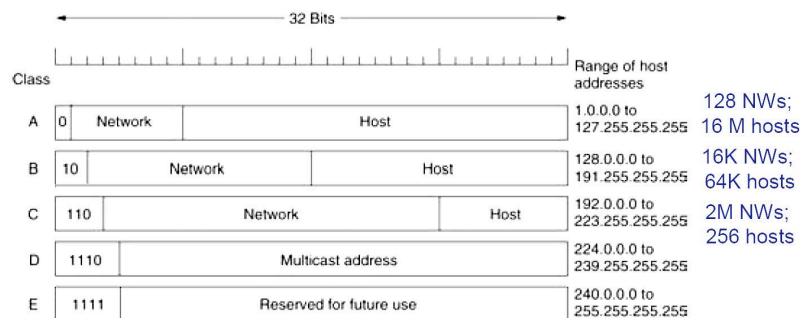


Abbildung 3.7: Aufteilung von IP-Adressen [Kar04]

Diese Form der Klassenbildung bringt zwei große Probleme mit sich. Zum einen sind Class A und Class B zu groß als das Router sie noch sinnvoll verwalten könnten. Zum anderen bringt diese Einteilung eine große Adressverschwendung mit sich: Eine Firma mit 2000 Hosts braucht ein Class B Netz und lässt darin ca. 62.000 Adressen unbenutzt. Um das erste Problem zu umgehen werden sogenannte Subnetze benutzt. Der Adressbereich wird in verschiedene Subnetze unterteilt. Der zentrale Router benutzt nun diese Subnetze um Pakete an sie auszuliefern. Dort nehmen weitere Router die Auslieferung an die einzelnen Hosts vor. Um die Subnetze zu erstellen wird ein Teil des Hostteils der Adresse als Subnetzadresse verwendet. Um die genaue Trennung anzugeben wird die sogenannte Subnetmask verwendet.

Die Subnetze sind außerhalb des „eigenen“ Netzwerkes nicht sichtbar. Der Router im eigenen Netzwerk teilt die Pakete auf die einzelnen Subnetze auf.

Das zweite Problem, dass die IP-Adressen ausgehen, wird mit einer ähnlichen Idee gelöst. Es werden auch wieder Teile des Host-teils der Adresse verwendet. Der Vorteil dieses Verfahrens ist, dass IP an sich

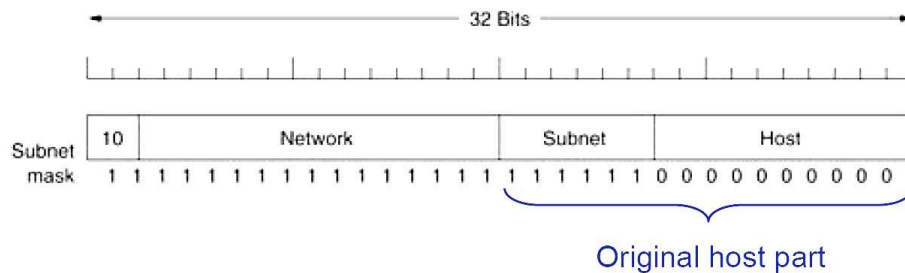


Abbildung 3.8: Subnetting [Kar04]

nicht erweitert werden musste. Die Router müssen nur die zusätzlichen Informationen in ihren Routingtabellen speichern.

### 3.4 Autonomous Systems

Ein Autonomes System (AS) ist im Internet ein IP-Netz, welches als Einheit verwaltet wird. Dieses Netz wiederum kann sich aus Teilnetzen zusammensetzen. Üblicherweise gehört ein autonomes System einem Internetdiensteanbieter, einer Universität z.B. uni-paderborn.de oder einer internationalen Firma, die redundant mit dem Rest des Internets verbunden ist.

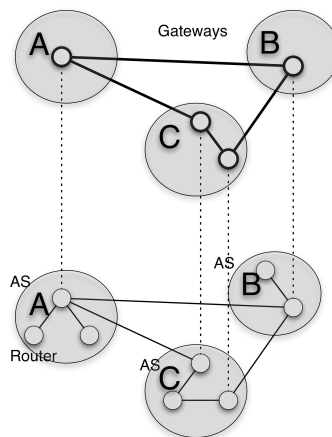


Abbildung 3.9: Autonomous Systems

Vorteile dieser Netzwerkstruktur sind:

- Größe der Routingtabellen ist abhängig von der Größe des autonomen Systems.
- Änderungen von Einträgen in den Routingtabellen werden nur innerhalb eines autonomen Systems weitergegeben.
- Routing kann im eigenen Netz kontrolliert werden
- Im administrativen System ein einheitliches Routingprotokoll
- Routingprotokolle der autonomen Systeme müssen nicht identisch sein

### 3.4.1 Intra-AS

Intra-AS bezeichnet das Routing innerhalb eines Autonomen Systems. Für dieses Routing stehen verschiedene Routingprotokolle zur Verfügung

#### **RIP: Routing Information Protocol**

Das Routing Information Protocol [Hed88] ist ein Routing-Protokoll auf Basis des Distanzvektor-Algorithmus. Beim Starten eines Routers kennt dieser nur seine direkt angeschlossenen Netzwerke und sendet diese Routingtabelle an die benachbarten Router. Gleichzeitig fordert er von seinen benachbarten Routern deren Routingtabelle an (Advertisement, max. 25). Mit diesen Informationen ergänzt der Router seine Routingtabelle und lernt somit, welche Netzwerke jeweils über welchen Router aus erreicht werden können und welche Kosten damit verbunden sind. Um Änderungen im Netzwerk (Ausfall oder Start eines Routers) zu erkennen, wird der Austausch der Routingtabellen regelmäßig wiederholt, dabei wird alle 30 Sekunden die gesamte Routingtabelle mit dem Nachbarn ausgetauscht, unabhängig davon ob sich Veränderungen ergeben haben oder nicht. Falls nach 180 Sekunden kein Advertisement eingetroffen ist, also keine Änderungen an einem Eintrag in der Routing-Tabelle vorgenommen wurde, wird dieser gelöscht. Damit wird die Route über den Nachbarn für ungültig erklärt und eine schnelle Reaktion auf Topologie-Änderungen gewährleistet. Zur Kontrolle einer doch noch möglichen Verbindung werden neue Advertisements zu dem Nachbarn geschickt, falls sich Änderungen an der Tabelle ergeben haben schickt dieser Nachbarn dann ebenfalls sein Advertisement und wird erneut in die Tabelle aufgenommen. Abschliessend muss bemerkt werden, dass durch das langsame Ausbreiten der Routing-Information im Netz, bei einer maximalen Ausdehnung von 15 Übertragungsabschnitten (Hops) beträgt diese bereits 7 Minuten, dieses Protokoll kaum für den Einsatz in großen Netzen geeignet ist.

#### **OSPF: Open Shortest Path First**

Das beim RIP bemängelte Verhalten in großen Netzen löst das Open Shortest Path First Protokoll [Moy91]. Kernstück von OSPF ist die LSD (Link State Database), die eine Liste aller benachbarten Router, zu denen eine bidirektionale Verbindung besteht, enthält. Sie spiegelt die Topologie des Netzes wieder, die in jedem Knotem abgebildet wird. Die Routenberechnung erfolgt auf diesen Daten mit dem Dijkstra-Algorithmus. Damit die Datenbank aufgebaut oder bei Topologie-Änderungen aktualisiert wird, ist der Austausch von Routing-Informationen notwendig. Diese werden mit Hilfe von, dem beim RIP - Protokoll beschriebenen, Advertisements durchs gesamte Autonome System geflutet.

Bei größeren Netzwerken wird zwischen 2 Ebenen des Netzes unterschieden. Eine Ebene bildet das lokale Gebiet und eine weitere das Rückgrat des Netzwerks, den so genannten Backbone. Spezielle Local Area Border Router fassen die Distanzen in dem eigenen lokalen Gebiet zusammen und bieten diese den anderen Area Border Routern per Advertisement an. Zur Verwaltung des gesamten Netzwerks findet ein Backbone Router Einsatz. Dieser verwendet gleichermaßen das OSPF - Protokoll beschränkt sich aber auf das Rückgrat (backbone).

#### **IGRP: Interior Gateway Routing Protocol**

[Gmb05] Das Interior Gateway Routing Protocol ist ein, 1980 von Cisco entwickeltes, proprietäres Distance-Vector-Routingprotokoll, das von Routern verwendet wird, um innerhalb eines autonomen Systems Routing-Informationen auszutauschen. Die Ziele bei der Entwicklung von IGRP waren vor allem eine Verbesserung der Skalierbarkeit sowie ein Überwinden der von RIP vorgegebenen maximalen Anzahl von 15 Netzwerkknoten die ein Zielnetzwerk entfernt sein darf bis das Netz als nicht erreichbar gilt. Bei IGRP ist der maximale Hop Count 255. IGRP kann, im gegensatz zu RIP und OSPF wo die Anzahl der Hops als Kostenmetrik gilt, die zur Verfügung stehende Bandbreite, die auf dem Pfad entstehende Verzögerung, die Leitungszuverlässigkeit und die Leitungsauslastung zur Erstellung von Kostenmetriken verwenden. Standardmäßig wird die Kostenmetrik einer Route aus der Bandbreite und der Leitungsverzögerung gebildet. Routing-Updates werden alle 90 Sekunden versendet. Zur Vermeidung von Routingschleifen kommen Holddown Timer, Split Horizon-Verfahren sowie Poison Reverse zum Einsatz.

### 3.4.2 Inter-AS

Die Autonomen Systeme werden über Exterior Gateway Protocols (EGPs) wie das Border Gateway Protocol verbunden. Dieses beschreibt, wie sich Router untereinander die Verfügbarkeit von Verbindungswegen zwischen den Systemen propagieren. Die BGP-Information enthält alle Daten über den kompletten Pfad zwischen den autonomen Systemen. Anhand dieser Information erstellt das Protokoll einen Graphen, der die Vernetzung der verschiedenen autonomen Systeme darstellt und eine Schleifenbildung des Routings ausschließt. Das Routing-Update, bei dem ein BGP-Router mit anderen BGP-Systemen in Verbindung steht, wird mittels TCP übertragen.

Dabei gilt: Falls Gateway  $X$  den Pfad zum Peer-Gateway  $W$  sendet

- dann kann  $W$  den Pfad wählen oder auch nicht
- Falls  $W$  den Pfad von  $X$  wählt, dann publiziert er  
 $Path(W, Z) = (W, Path(X, Z))$

Die vom BGP verwendete Metrik basiert auf Informationen, die der Netzverwalter den Routern bei deren Konfiguration zuweist, sowie auf den physikalischen und übertragungstechnischen Parametern. Da jeder BGP-Router über Routen-Informationen von anderen, insbesondere der benachbarten BGP-Routern verfügt, baut sich jeder BGP-Router eine Datenbank für die Routen zu allen erreichbaren autonomen Systemen auf.

## 3.5 Optimales Routing - Max Flow

Die Lastverteilung eines Netzwerks kann man, anhand des in der Vorlesung Datenstrukturen und Algorithmen vorgestellten Max - Flow - Algorithmus, optimieren. Hierbei ist ein gerichteter Graph  $G$  mit Kantenkapazitäten  $w(e) \geq 0$  und zwei ausgezeichneten Knoten  $s$  (Quelle) und  $t$  (Senke) aus  $V$  gegeben. Man kann sich nun diesen Graph als Röhrensystem vorstellen, in dem Wasser von  $s$  nach  $t$  gepumpt werden soll.

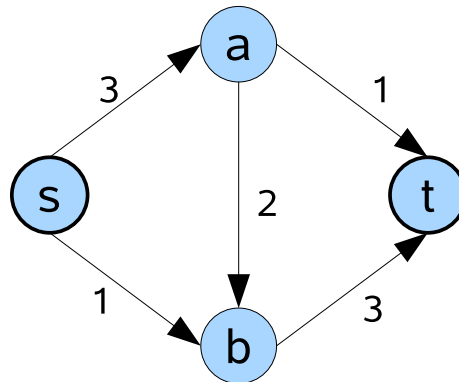


Abbildung 3.10: MaxFlow-Graph

Dieses Pumpen bewirkt einen Fluss der jeder Kante eine Zahl  $f(e)$  mit  $0 \leq f(e) \leq w(e)$  zuordnet. Damit gilt für alle Knoten  $x$   $\sum_{e \in I(x)} f(e) = \sum_{e \in O(x)} f(e)$  wobei  $I(x)$  die eingehenden Kanten und  $O(x)$  die ausgehenden Kanten symbolisiert. Daraus ergibt sich folgende Bedingung eines Max - Flow - Graphens die immer erfüllt sein muss: Abgesehen von der Quelle  $s$  und der Senke  $t$  muss in jeden Knoten genau so viel hineinfließen, wie herausfließen, dies wird auch Flussserhaltung genannt. Ziel ist es nun so viel Wasser wie möglich von  $s$  nach  $t$  zu pumpen, also den Fluss zu maximieren.

$$\max \sum_{e \in O(s)} f(e) = \sum_{e \in I(t)} f(e)$$

Der Wert eines Flusses von  $s$  nach  $t$  ist die Summe der eingehenden abzüglich der ausgehenden Belegungen der Senke  $s$  bzw. die ausgehenden Belegungen abzüglich der eingehenden Belegungen der Quelle  $t$ .

Eine Teilmenge der Knoten in einem Netzwerk, die  $s$  aber nicht  $t$  enthält, nennt man einen Schnitt. Feinstein, Ford, Fulkerson und Shannon formulierten hierzu das Max-Flow-Min-Cut Theorem [PDSOK05] welches besagt, dass die Kapazität eines Schnittes die Summe der Kapazitäten der aus dem Schnitt heraus führenden Kanten ist. Der Wert eines maximalen Flusses im Netzwerk kann damit nicht größer als die Kapazität eines beliebigen und somit auch eines minimalen Schnittes sein.

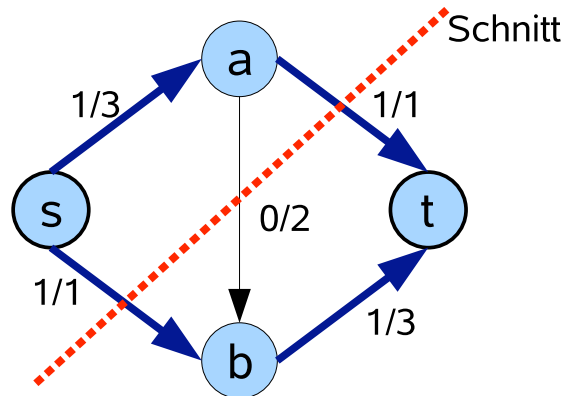


Abbildung 3.11: Minimaler Schnitt

Das Restnetzwerk (auch: Residualgraph) bezüglich eines zulässigen Flusses ist ein Netzwerk, das alle Kanten des ursprünglichen Netzwerkes enthält, mit um den jeweiligen Flusswert verminderten Kantenkapazitäten. Zusätzlich hat das Restnetzwerk noch genau dem Fluss entgegenlaufende Kanten.

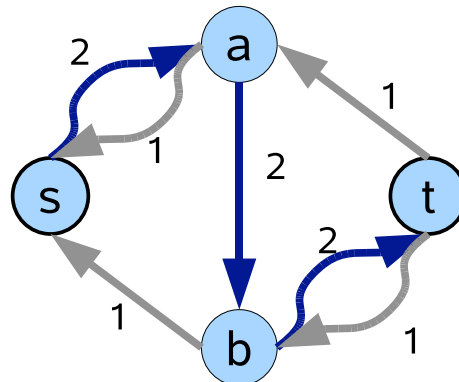


Abbildung 3.12: Residualgraph

Die Berechnung des maximalen Flusses lässt sich nun auf das Lösen eines linearen Programmes z.B. mit der Simplex - Methode / Ellipsoid - Methode oder der Ford - Fulkerson - Methode für ganze Zahlen reduzieren.

### 3.6 Nash Equilibrium

Das Nash Equilibrium oder auch Nash-Gleichgewicht ist ein zentraler Begriff der mathematischen Spieltheorie. Es beschreibt in Spielen einen Zustand eines strategischen Gleichgewichts. Im Nash Equilibrium



Nash Equilibrium [JFN50]: If there is a set of strategies with the property that no player can benefit by changing her strategy while the other players keep their strategies unchanged, then that set of strategies and the corresponding payoffs constitute the Nash Equilibrium.

Abbildung 3.13: John F. Nash, Jr. geb. 06.13. 1928

will kein Spieler seine Strategie ändern, solange die anderen Spieler bei ihrer gewählten Strategie bleiben. Definition und Existenzbeweis des Nash-Gleichgewichts gehen auf die 1950 veröffentlichte Dissertation des Mathematikers John Forbes Nash Jr. zurück, auf dessen Leben der Film „A Beautiful Mind“ basiert.

#### 3.6.1 Gefangenendilemma

Zwei Gefangene werden verdächtigt, gemeinsam eine Straftat begangen zu haben. Die Höchststrafe für das Verbrechen beträgt drei Jahre. Beiden Gefangenen wird nun ein Handel angeboten, der beiden bekannt ist. Wenn einer gesteht, und somit seinen Partner belastet, kommt er ohne Strafe davon - der andere muss die vollen drei Jahre absitzen. Entscheiden sich beide zu schweigen, bleiben nur Indizienbeweise, die aber ausreichen, um beide für ein Jahre einzusperren. Gestehen aber beide die Tat, erwartet jeden eine Gefängnisstrafe von drei Jahren. Nun werden die Gefangenen unabhängig voneinander befragt. Es besteht weder vor noch während der Befragung die Möglichkeit für die beiden sich untereinander abzusprechen.

Nach den Angaben kann folgende Tabelle aufgebaut werden. Die Pfeile symbolisieren die jeweilige optimale Strategie des jeweiligen Spielers bei entsprechender Entscheidung des Gegenspielers. Es ist sofort ersichtlich dass immer die Strategie „Aussagen“ die bessere Wahl ist für den einzelnen ist.

Paradox kann dieses Dilemma genannt werden, da die individuell vernünftigste Entscheidung der Gefan-

		Spieler 2	
		nicht aussagen	aussagen
Spieler 1	nicht aussagen	1 / 1 I	0 / 3 II
	aussagen	3 / 0 III	2 / 2 IV

Die Tabelle zeigt die optimalen Strategien für Spieler 1 (blau markiert) und Spieler 2 (rot markiert). Spieler 1 wählt 'aussagen' (III) und Spieler 2 wählt 'aussagen' (IV).

Abbildung 3.14: Gefangenendilemma

genen (aussagen) und die kollektiv vernünftigste Entscheidung (nicht aussagen) auseinander fallen.

Gemäß der klassischen Analyse des Spiels ist im nur einmal gespielten Gefangenendilemma die einzig rationale Strategie für einen am eigenen Wohl interessierten Spieler, nicht zu kooperieren und auszusagen, da er durch seine Entscheidung das Verhalten des Mitspielers nicht beeinflussen kann und er sich unabhängig

von der Entscheidung des Mitspielers immer besser stellt, wenn er selbst nicht kooperiert. Diese Analyse setzt voraus, dass die Spieler nur einmal aufeinander treffen, und ihre Entscheidungen keinen Einfluss auf spätere Interaktionen haben können. Die Situation ändert sich, wenn das Spiel mehrere Runden gespielt wird. Dann kann ein Vertrauensbruch im nächsten oder einem späteren Spiel geahndet werden (Vergeltung), Kooperation werden belohnt.

### 3.6.2 Das Paradoxon von Braess

Das Paradoxon von Braess [Bra68] beschreibt ein weiteres Beispiel bei dem die individuelle gewählte Strategie einzelner nicht das globale Optimum ergeben muss. Der Bochumer Dietrich Braess belegte damit, dass der Bau einer neuen Strasse häufig zu neuen Staus führen kann. Sein Modell lässt sich leicht auf ein Netzwerk übertragen bei dem von einem Knoten  $S$  nach  $T$  zwei Verbindungen, eine über Knoten  $A$  und eine über Knoten  $B$ , führen. Die Verzögerung auf den Kanten ist  $d(S, A) = d(B, T) = 1$  und  $d(S, B) = f(S, B)$ ,  $d(A, T) = f(A, T)$ , wobei  $f(e)$  die Anzahl der Pakete sind, welche die Kante  $e$  benutzen, geteilt durch die Gesamtanzahl  $n$  aller Pakete im Graphen. Die Verzögerung eines Pakets von der Quelle zur Senke ergibt sich aus der Summe der Verzögerungen der benutzten Kanten und diese ist von der Wegwahl der anderen Pakete abhängig.

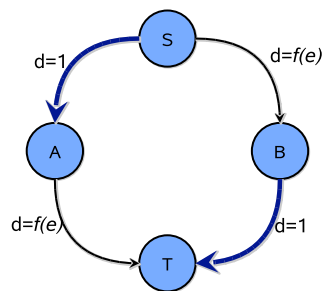


Abbildung 3.15: Braess Paradoxon Teil 1

In diesem Modell werden die Hälfte aller Pakete den Weg von  $S$  über  $A$  nach  $T$  und die andere Hälfte den Weg von  $S$  über  $B$  nach  $T$  wählen. Diese Wahl würde im Nash Equilibrium liegen, denn kein Paket will seinen Weg ändern da sich dann die Verzögerung von  $\frac{n}{2}$  zu  $\frac{n}{2} + 1$  erhöhen würde. Es ergibt sich somit ein stabiles System.

Fügt man dem Netzwerk eine weitere Kante mit Verzögerung 0 von  $B$  nach  $A$  hinzu verändert sich das Verhalten der einzelnen Pakete.

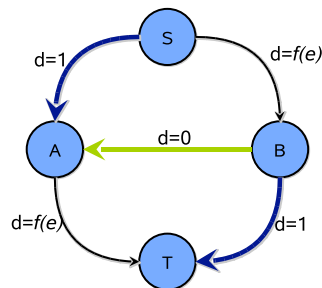


Abbildung 3.16: Braess Paradoxon Teil 2



Teilten sich vorher noch alle Pakete gleichmäßig auf wählen nun alle Pakete den Weg von  $S$  nach  $B$  über  $A$  nach  $T$ . Um dies zu zeigen seien folgende Strategien gegeben:

$$S_1 : S \Rightarrow A \Rightarrow T$$

$$S_2 : S \Rightarrow B \Rightarrow T$$

$$S_3 : S \Rightarrow B \Rightarrow A \Rightarrow T$$

Weiterhin seien folgende Fälle genauer betrachtet

1. Fall

Die Pakete teilen sich wie im oben betrachteten Beispiel genau zur Hälfte auf,  $1 \dots \frac{n}{2}$  wählen  $S_1$  und  $\frac{n}{2} + 1 \dots n$  wählen  $S_2$ , beide Strategien ergeben jeweils eine Verzögerung von  $\frac{1}{2} + 1$ . Mindestens ein Paket würde aber im nächsten Schritt  $S_3$  wählen, da sie im Vergleich zu den anderen Strategien weniger Verzögerung bietet, nämlich  $\frac{1}{2} + 0 + \frac{1}{2}$ . Die Strategiewahl  $S_1$  und  $S_2$  ist hier somit nicht im Nash Equilibrium.

2. Fall

Alle Pakete bis auf eins wählen Strategie  $S_3$ , das andere wählt  $S_1$ . Pakete die Strategie  $S_3$  wählten haben eine Verzögerung von  $\frac{n-1}{n} + 0 + \frac{n}{n}$ . Das einzelne Paket welches Strategie  $S_1$  wählte hat eine Verzögerung von  $\frac{n}{n} + \frac{n}{n}$ . Das einzelne Paket hat einen Geschwindigkeitsnachteil und wird seine Strategie zu  $S_3$  ändern. Damit ist auch Fall 2 nicht im Nash Equilibrium.

3. Fall

Alle Pakete wählen Strategie  $S_3$ . Alle Pakete besitzen eine Verzögerung von  $\frac{n}{n} + 0 + \frac{n}{n}$ . Bemerkenswerterweise ist dieser Fall im Nash Equilibrium, da keines der Pakete seine Strategie wechseln will, da es nur in Fall 2 zurückfallen würde, welches nicht für dieses Paket optimal wäre.

Global gesehen wäre der 1. Fall damit optimal, das Paradoxon aber besagt das der 3. Fall für die einzelnen Pakete, im Vergleich zu den anderen Paketen, optimaler ist.



# Literaturverzeichnis

- [Bra68] Dietrich Braess. Über ein paradoxon aus der verkehrsplanung, March 1968.
- [Gmb05] Core Networks GmbH. Wikipedia - interior gateway routing protocol, June 2005.
- [Hed88] C. Hedrick. Rfc 1058 - routing information protocol, June 1988.
- [JFN50] Jr. John F. Nash. Equilibrium points in n-person games. Master's thesis, Princeton, 1950.
- [Kar04] Holger Karl. Vorlesung rechnernetze. In *Folien zur Vorlesung Rechnernetze*, August 2004.
- [Moy91] J. Moy. Rfc 1247 - ospf version 2, July 1991.
- [PDSOK05] Hartmut Noltemeier Prof. Dr. Sven O. Krumke. Flüsse und strömungen. In *Graphentheoretische Konzepte und Algorithmen*, 2005.
- [THCS01] Ronald L. Rivest Thomas H. Cormen, Charles E. Leiserson and Clifford Stein. Introduction to algorithms, second edition. In *Introduction to Algorithms, Second Edition*, Fitzroy House, 11 Chenies Street, London WC1E 7EY, 2001. The MIT Press.