DHBW Duale Hochschule Baden-Württemberg Karlsruhe

Netztechnik I

T2INF4201.1

Vermittlungsschicht (Network Layer)

Markus Götzl
Dipl.-Inform. (FH)
mail@markusgoetzl.de

http://www.dhbw-karlsruhe.de

Vermittlungsschicht (Network Layer)

- Aufgaben der Vermittlungsschicht
- Routingalgorithmen
- Überlastkontrolle
- Vermittlungsschicht im Internet (IP)

Unterschied verbindungsorientierte vs -lose Verbindung

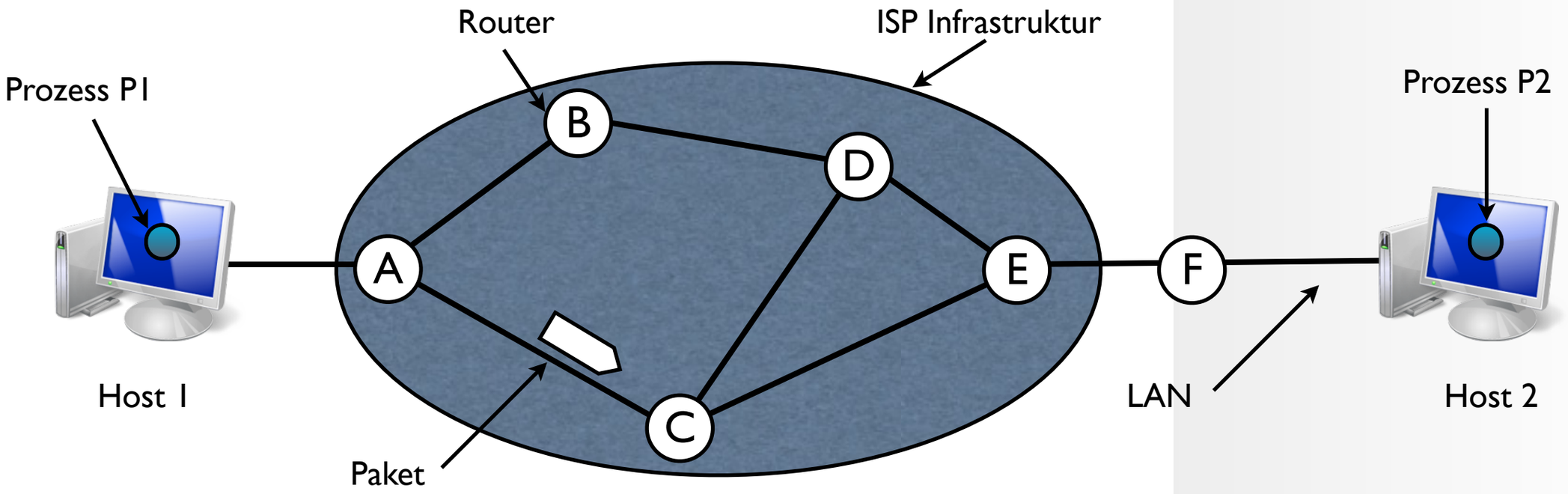
Dijkstra nicht in KA machen (nur wissen dass es für Netzwerke interessant ist)
wissen was Link State ist, Unterschied zu Distance Routing, aber auch nicht Distance Routing Verfahren wissen

Aufgaben der Vermittlungsschicht

- Zustellung von Paketen von der Quelle zum Ziel (Ende-zu-Ende)
  - Im Gegensatz zur Sicherungsschicht, welche für den Transport von Frames von einem Ende des Übertragungsmediums zum anderen verantwortlich ist.
- Finden geeigneter Routen.
- Überlastkontrolle der Routen.

Aufgaben der Vermittlungsschicht

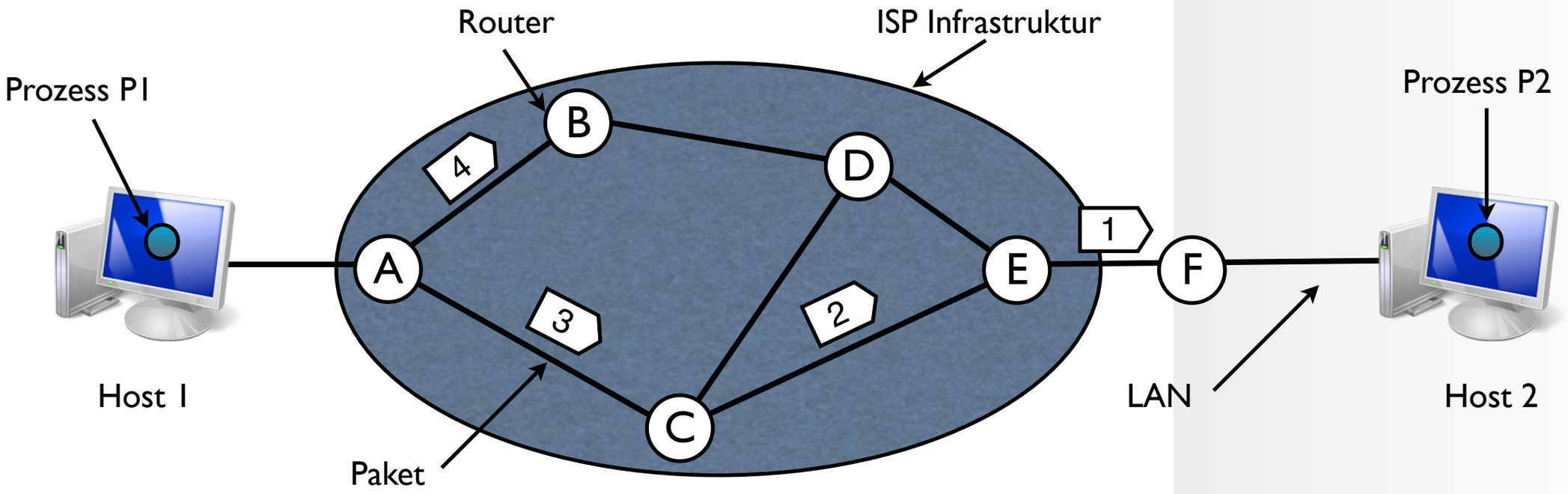
Kontext der Vermittlungsschicht



- Der größte Teil der Infrastruktur liegt beim ISP
- Ein Paket (unterwegs von P1 zu P2) folgt einer bestimmten Weg entlang mehrerer Router.

Aufgaben der Vermittlungsschicht

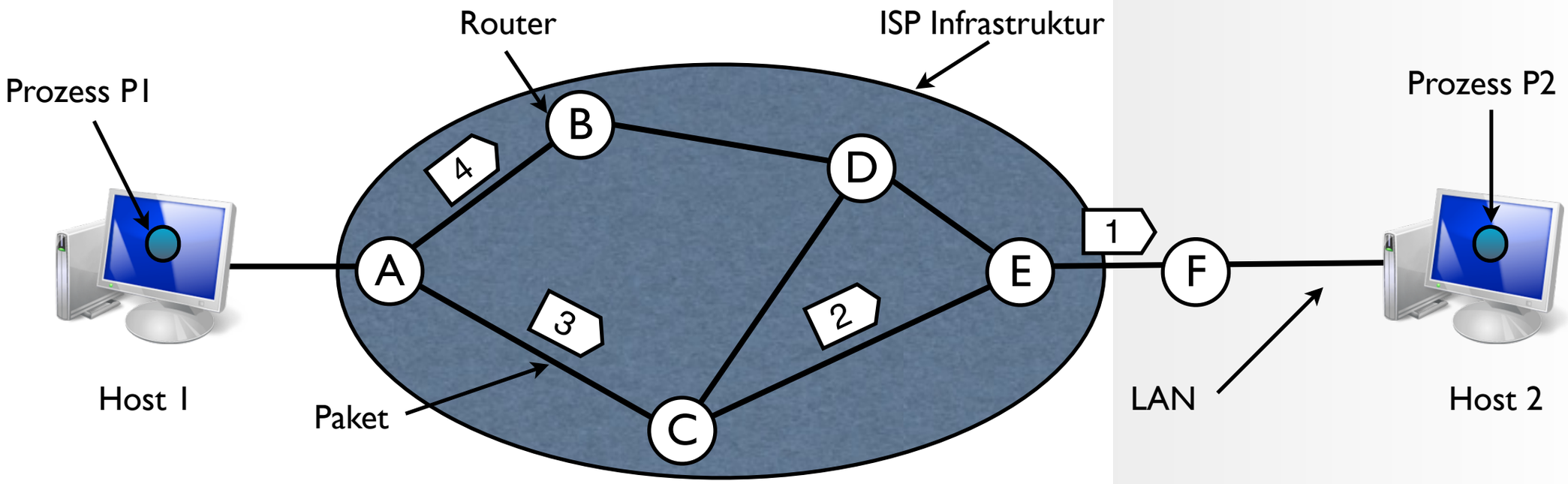
Verbindungsloser Dienst



- Eine Nachricht soll von P1 (Host 1) nach P2 (Host 2) gesendet werden.
- Die Größe der Nachricht verlangt eine Aufteilung in vier Pakete
  - Bei einem **verbindungslosen** Dienst werden die Pakete individuell und unabhängig geroutet.

Aufgaben der Vermittlungsschicht

Verbindungsloser Dienst



Routingtabelle A
(Ziel, Leitung)

A	-
B	b
C	c
D	b
E	c
F	c

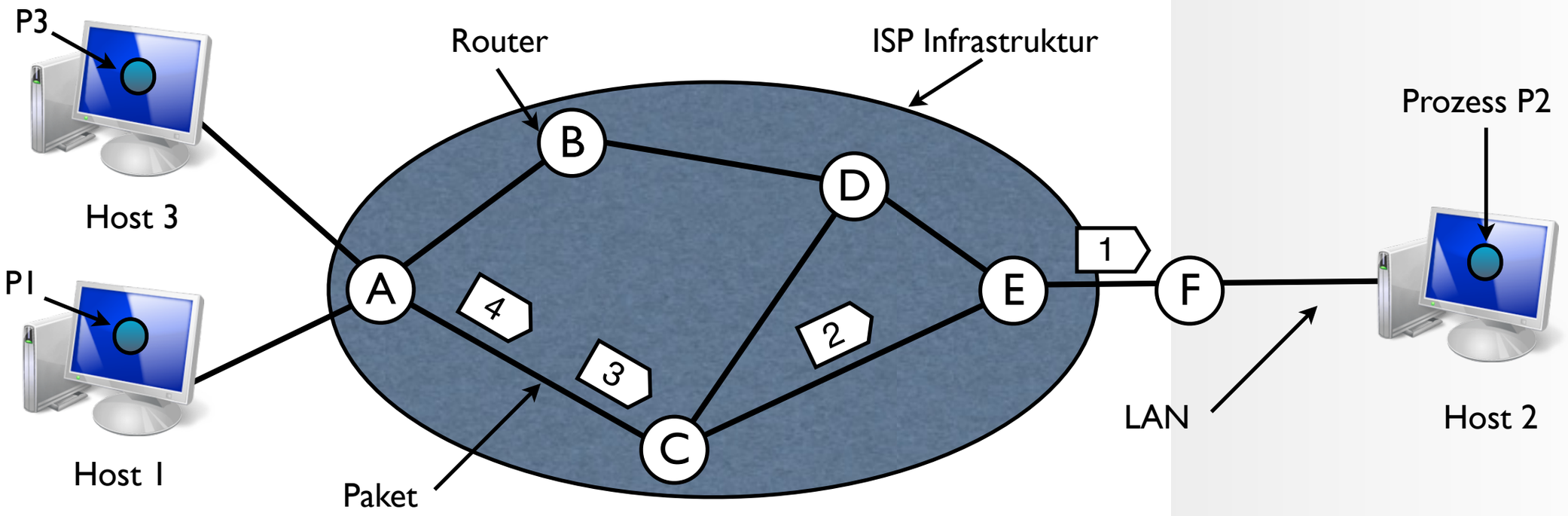
Routingtabelle A'
(Ziel, Leitung)

A	-
B	b
C	c
D	b
E	b
F	b

- Pakete 1,2,3 werden von A über C und E nach F gesendet. Ab F werden die Pakete in einem Frame (LAN) nach H2 und P2 weiter gesendet.
- Paket 4 wird allerdings von A nach B gesendet, da Routingtabelle geändert wurde (z.B. da die Route über C ausgelastet ist)

Aufgaben der Vermittlungsschicht

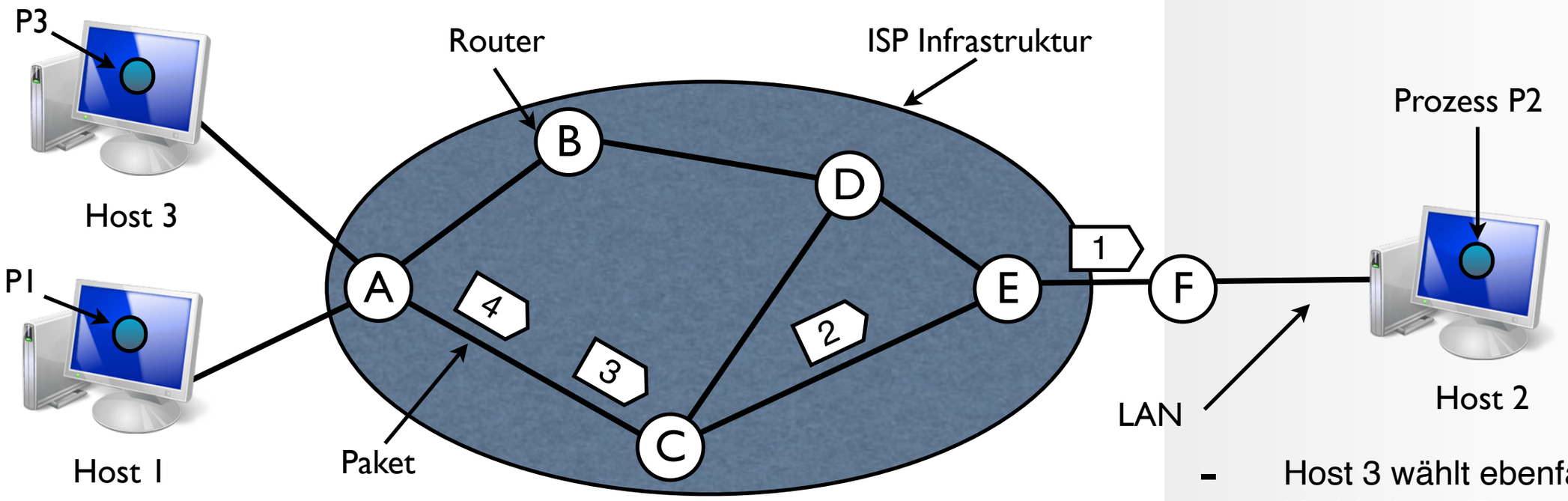
Verbindungsorientierter Dienst



- Eine Nachricht soll von P1 (Host 1) und nach P2 (Host 2) gesendet werden.
- Eine weitere Nachricht soll von P3 nach P2 gesendet werden
  - Bei einem **verbindungsorientierten** Dienst beim Aufbau der Verbindung ein Route festgelegt.
  - Beim Abbau der Verbindung wird die Route wieder gelöscht.

Aufgaben der Vermittlungsschicht

Verbindungsorientierter Dienst



Routingtabellen A (Quelle bzw. Ziel, Verbindung ID)

In		Out	
H1	1	C	1
H3	1	C	2

Routingtabellen C (Quelle bzw. Ziel, Verbindung ID)

In		Out	
A	1	E	1
A	2	E	2

- Host 3 wählt ebenfalls die Verbindung ID 1
- A kann die Verbindung unterscheiden, da die Quellen unterschiedlich sind.
- C kann das nicht, daher muss A eine neue Verbindung ID vergeben.

Routingalgorithmen

- Die Hauptaufgabe der Vermittlungsschicht ist das Routen von Paketen von einer Quelle zu einem Ziel.
- Mit der Hilfe von Routingalgorithmen kann in der Vermittlungsschicht entschieden werden auf welcher Leitung ein ankommendes Paket weitergeschickt werden soll.
  - Im Falle eines verbindungslosen Dienstes muss diese Entscheidung für jedes Paket getroffen werden (Optimierung ist möglich z.B. Entscheidung für Gruppen von Paketen!).
  - Im Falle eines verbindungsorientierten Dienstes wird diese Entscheidung beim Verbindungsaufbau für alle zu sendende Pakete getroffen (Session Routing).

Routingalgorithmen

Einteilung von Routingalgorithmen

- Statische Routingalgorithmen (nonadaptive algorithms)
  - Die Routingentscheidungen werden im Vorfeld berechnet und im Anschluss in den Netzwerkroutern gespeichert.
  - Diese Routingalgorithmen können nicht zur Laufzeit auf Veränderungen im Netzwerk reagieren und eignen sich am besten wenn keine, oder nur kleine, Änderungen in der Topologie oder dem Verkehrsaufkommen erwartet werden.
- Dynamische Routingalgorithmen (adaptive algorithms)
  - Die Routingentscheidungen werden in Abhängigkeit bestimmter Messwerte oder Metriken zur Laufzeit angepasst.
  - Diese Routingalgorithmen unterschieden sich nach
    - der Metrik, welche benutzt wird um Routen zu optimieren
    - woher sie die Daten/Messwerte beziehen (andere Router, lokal, von allen Routern, etc.)
    - wann sie Routen ändern (Änderungen an der Topologie, Verkehrsaufkommen, etc.)

Einteilung von Routingalgorithmen

- Interior Gateway Protokoll (IGP)
  - Diese Protokolle werden innerhalb von autonomen System eingesetzt (Routing Domäne). Der Betreiber kann dabei frei entscheiden, wie sein Routing intern aufgebaut sein soll.
- Exterior Gateway Protokolle (EGP)
  - Exterior Gateway Protokolle werden verwendet um unterschiedliche autonome Systeme miteinander zu verbinden.

Routingalgorithmen

Hierarchisches Routing

- Die Grundlage des hierarchischen Routings ist die Aufteilung großer Netze in Regionen.
- Die Knoten einer Region haben nur Routing-Informationen über ihre eigene Region.
- In jeder Region existiert ein oder mehrere ausgezeichnete Knoten, welche als Schnittstelle zu anderen Regionen dient.
- In sehr großen Netzen sind weitere Hierarchien aufgrund zunehmender Größe der Netze möglich (Regionen, Cluster, Zonen, Gruppen, ...).

Anforderungen an Routingalgorithmen

- Korrektheit
- Einfachheit
- Rubustheit
  - Ausfall unterschiedlichster Komponenten (Harware/Software) muss ohne Verlust der Gesamtfunktionalität überstanden werden.
- Stabilität
  - Nach Ausfällen oder Änderungen muss der Routingalgorithmus schnell in einen stabilen Zustand finden und darf auf keinen Fall in einen Schwingungszustand eintreten.

Anforderungen an Routingalgorithmen

- Fairness
  - Der Verkehr zwischen einzelnen Knoten sollte gleich behandelt werden.
  - Diese Anforderung kann in Wiederspruch zur Effizienz stehen.
    - Bei einer hohen Auslastung zwischen zwei Knoten ist es effizient, Paketen, die an dieser Kommunikation nicht beteiligt sind, eine weniger attraktive Route zuzuteilen. Dies würde aber diese Kommunikation benachteiligen.
- Effizienz
  - Effizienz kann kann auf unterschiedlichen Ebenen optimiert werden, (Verzögerungsschwankungen - Jitter oder Datendurchsatz) welche ebenfalls im Wiederspruch zueinander stehen können.

Grundlegende Klassen von Routing - Protokollen im Internet:

- Link State Routing
  - IGP: OSPF, IS-IS
- Distance Vector Routing
  - IGP: RIPv1, v2, IGRP, EIGRP
  - EGP: BGP-4

Link State Routingalgorithmus

- Der Link State Routingalgorithmus besteht aus fünf Schritten:

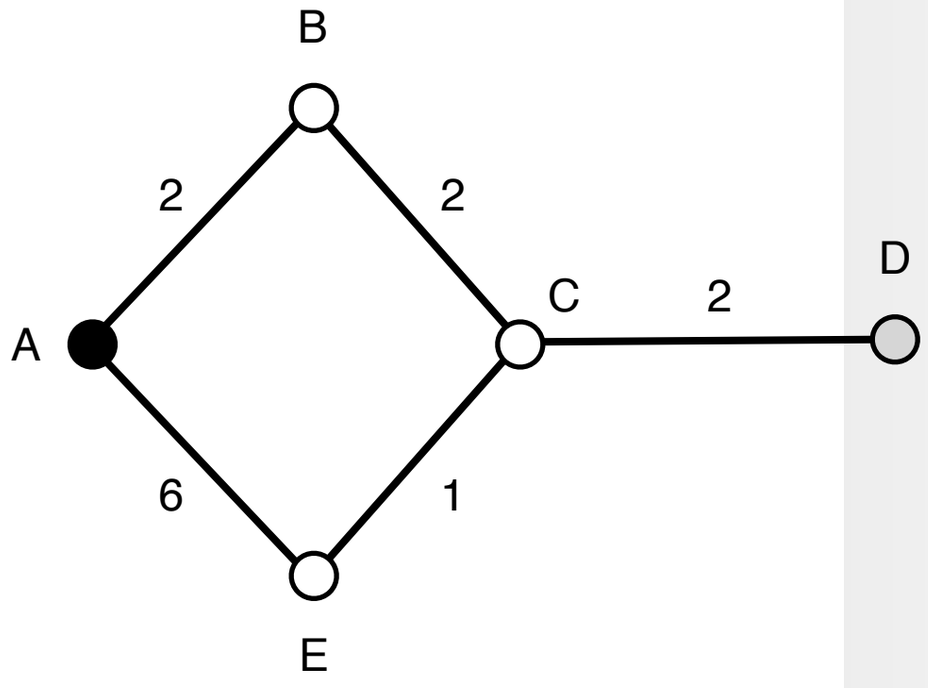
1. Finde alle Nachbarn (verbundene Router)
2. Ermittle die Distanz- oder Kostenmetrik zu allen Nachbarn
3. Erstelle ein Paket, welches diese Informationen enthält
4. Sende dieses Paket zu allen Nachbarn und empfange/speichere diese Pakete von Nachbarn
5. Berechne den kürzesten Pfad zu allen bekannten Routern

Link State Routingalgorithmus - kürzester Pfad (Dijkstra Algorithmus)

- Der kürzeste Pfad zwischen zwei Knoten in einem bekannten Graphen kann mit dem Dijkstra Algorithmus berechnet werden.
  - Der gesamte Graph (die Topologie) wird im Link State über die Schritte 1-4 ermittelt.
  - Der Dijkstra Algorithmus benötigt vier Schritte:
    1. Erstelle eine Tabelle mit drei Spalten: Knoten, Wert, Vorgänger
    2. Weise jedem Knoten den Wert ∞ zu (der Wert ist die Summe aller Kantengewichtungen).
    3. Setze den Wert des Startknotens A auf 0 und den Vorgänger auf A
    4. Für alle nicht besuchten Knoten:
       a) Alle nicht besuchten Nachbarn neu beschriften (mit Summe der Werte und dem Vorgänger), wenn die Summe größer als aktuell.
       b) Nicht besuchter Knoten mit der kleinsten Summe wird neuer permanenter Knoten
  - Der Algorithmus terminiert wenn alle Knoten besucht wurden.

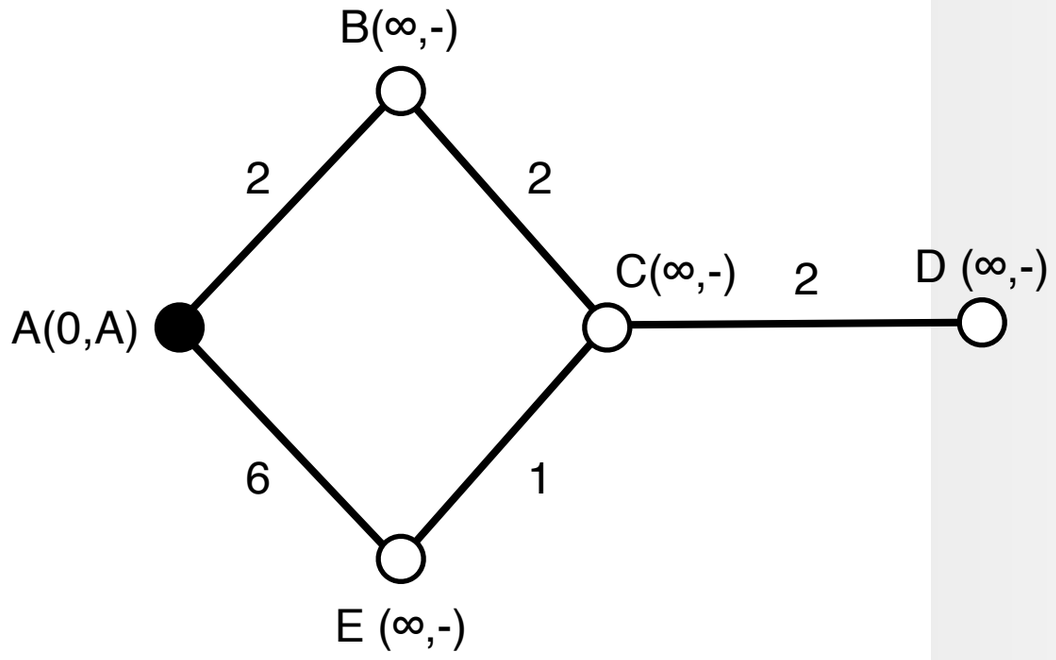
Dijkstra Algorithmus - Graph

Finde den kürzesten Pfad von A nach D



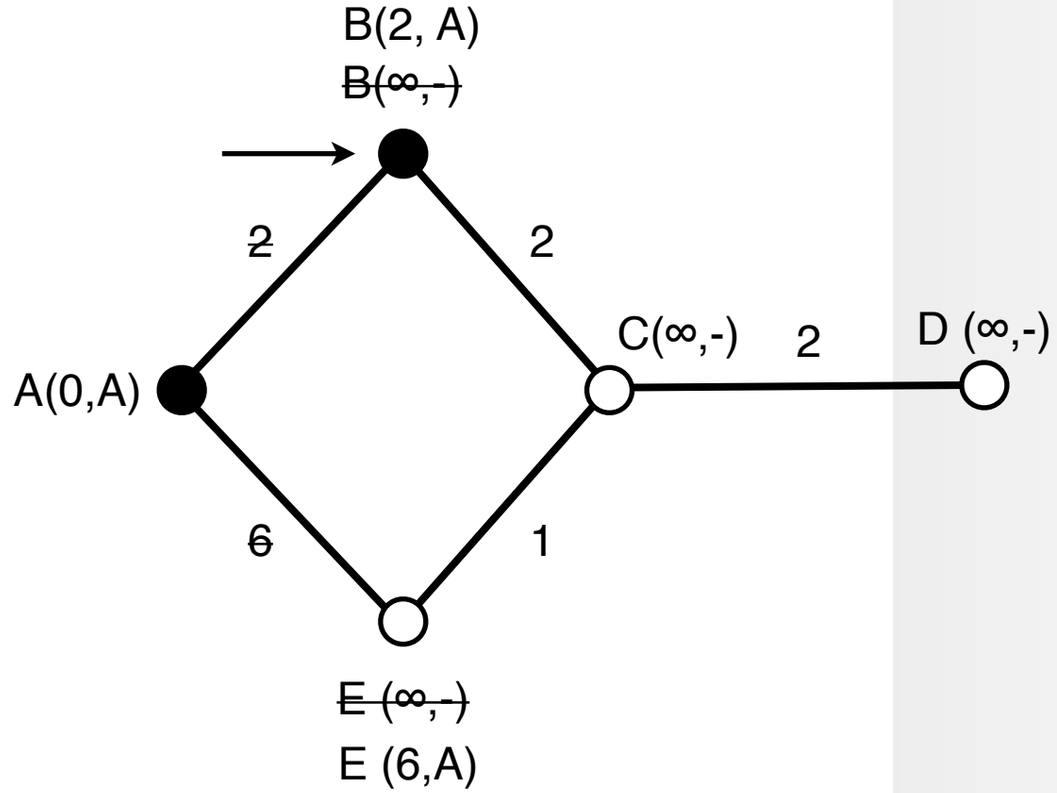
Dijkstra Algorithmus - Schritt 1,2,3

1. Initialisierung



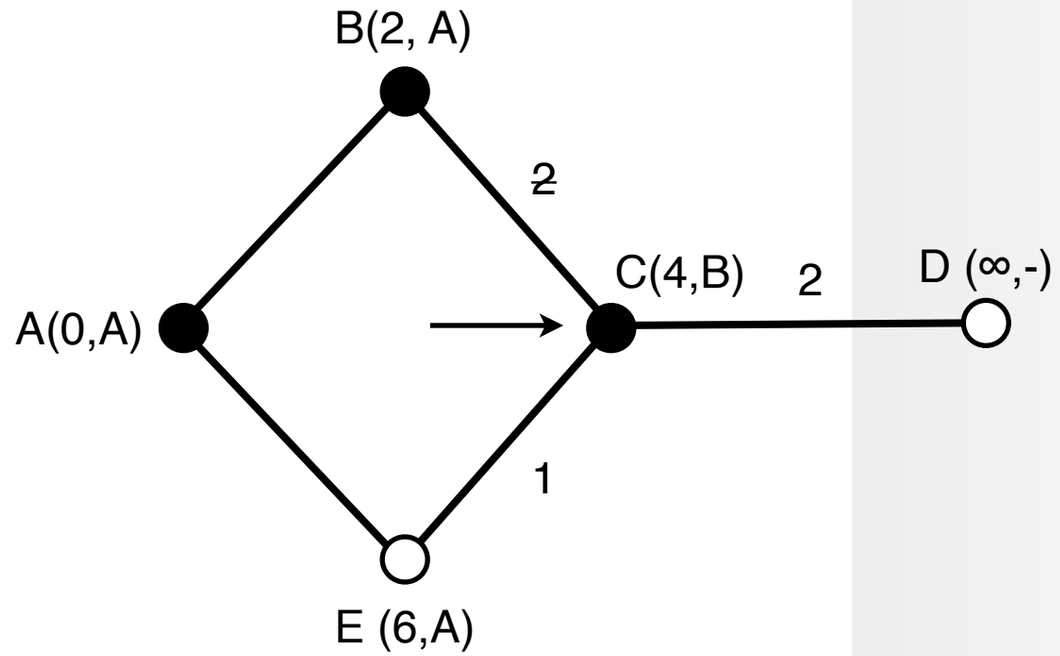
Dijkstra Algorithmus - Schritt 4:

1. A permanent
2. Alle nicht besuchten Nachbarn neu beschriften (mit Summe der Werte und dem Vorgänger), wenn die Summe größer als aktuell
3. Nicht besuchter Knoten mit der kleinsten Summe wird neuer permanenter Knoten



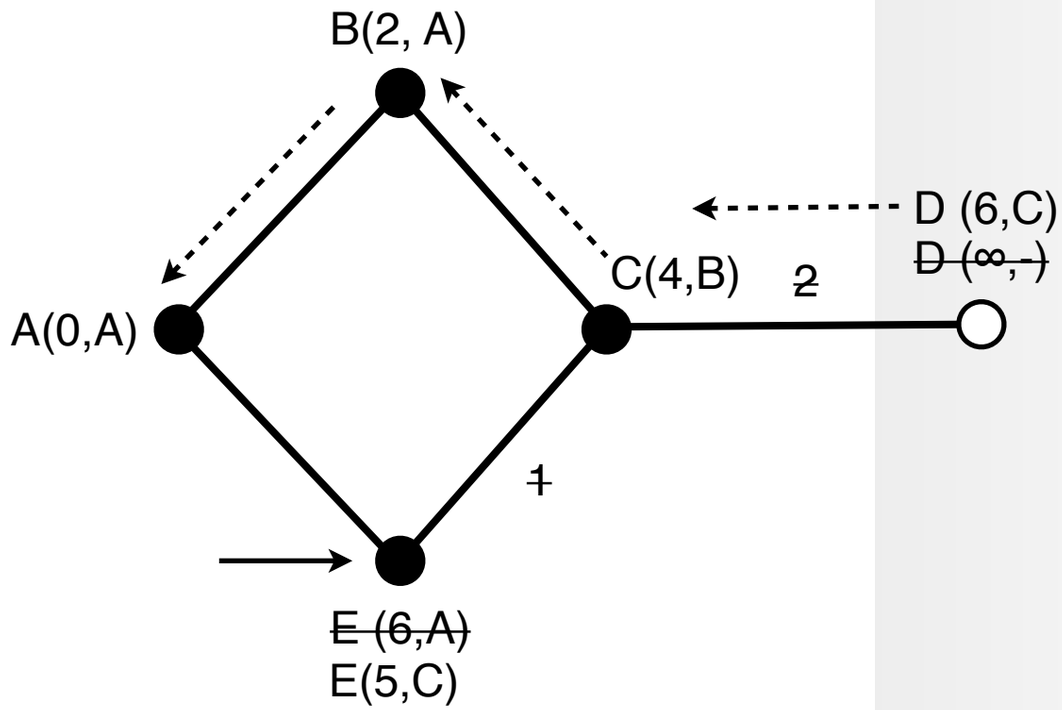
Dijkstra Algorithmus - Schritt 4:

1. B permanent
2. Alle nicht besuchen Nachbarn neu beschriften (mit Summe der Werte und dem Vorgänger), wenn die Summe größer als aktuell
3. Nicht besuchter Knoten mit der kleinsten Summe wird neuer permanenter Knoten



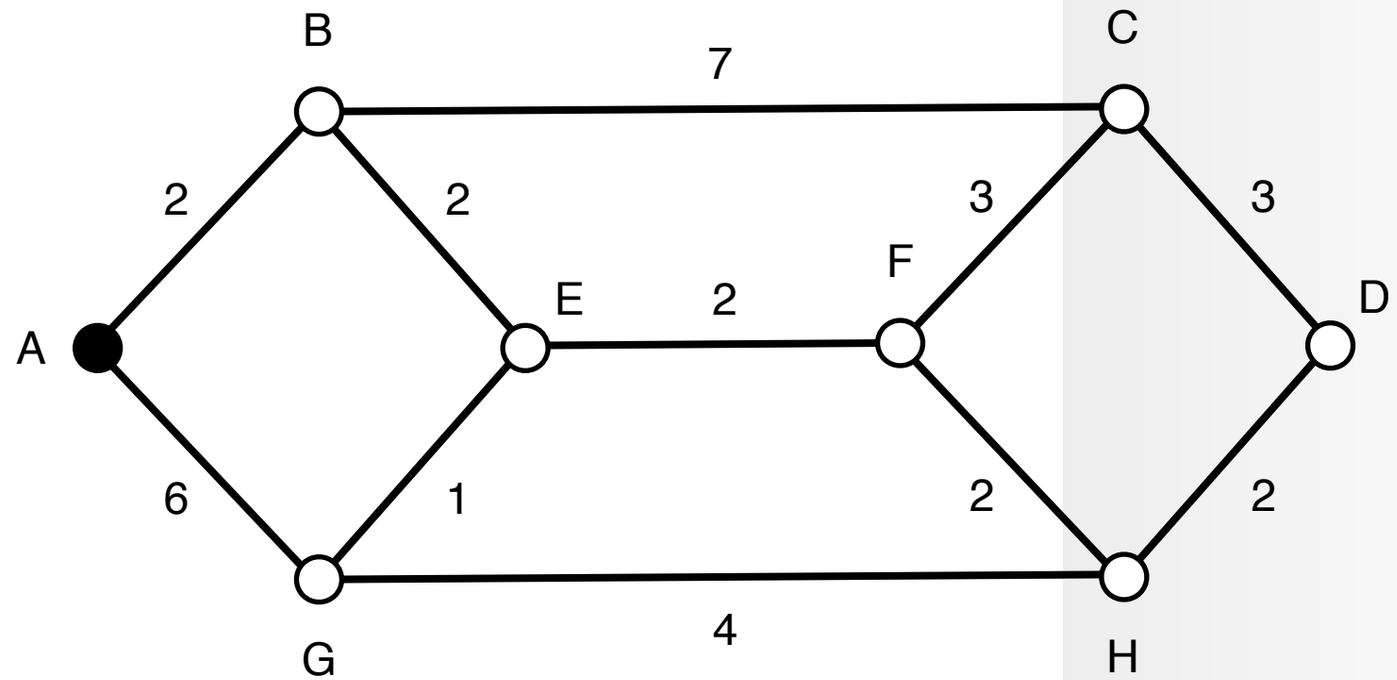
Dijkstra Algorithmus - Schritt 4:

1. C permanent
2. Alle nicht besuchen Nachbarn neu beschriften (mit Summe der Werte und dem Vorgänger), wenn die Summe größer als aktuell
3. Nicht besuchter Knoten mit der kleinsten Summe wird neuer permanenter Knoten



Dijkstra Algorithmus

Finde den kürzesten Pfad von A nach D



Link State Routingalgorithmus - Bellman

- Der Dijkstra Algorithmus findet die kürzesten Pfad auch für alle Router, die auf dem Pfad zum Ziel Router liegen.
  - **Optimalitätsprinzip von Bellman:**
    - Ein kürzester Weg P zwischen den Knoten A und B, der durch die Knoten X und Y führt, muss auch zwischen X und Y einen kürzesten Weg zwischen diesen beiden Knoten verwenden.
    - Wäre das nicht der Fall, könnte P verkürzt werden, indem zwischen X und Y ein kürzerer Teilweg verwendet wird, und dann wäre P kein kürzester Weg zwischen A und B gewesen, im Widerspruch zur Annahme.

Distance Vector Routing

- Der Distance Vector Algorithmus besteht aus vier Schritten.

1. Erzeuge eine Kostenmatrix: welche Router über welche Nachbarn und zu welchen Kosten erreichbar sind (im 1. Schritt nur die Kosten zu direkten Nachbarn).

2. Erzeuge eine Aufstellung mit Informationen, welche Router zu welchen Kosten am besten erreichbar sind. Schicke diese Informationen an alle Nachbarn.

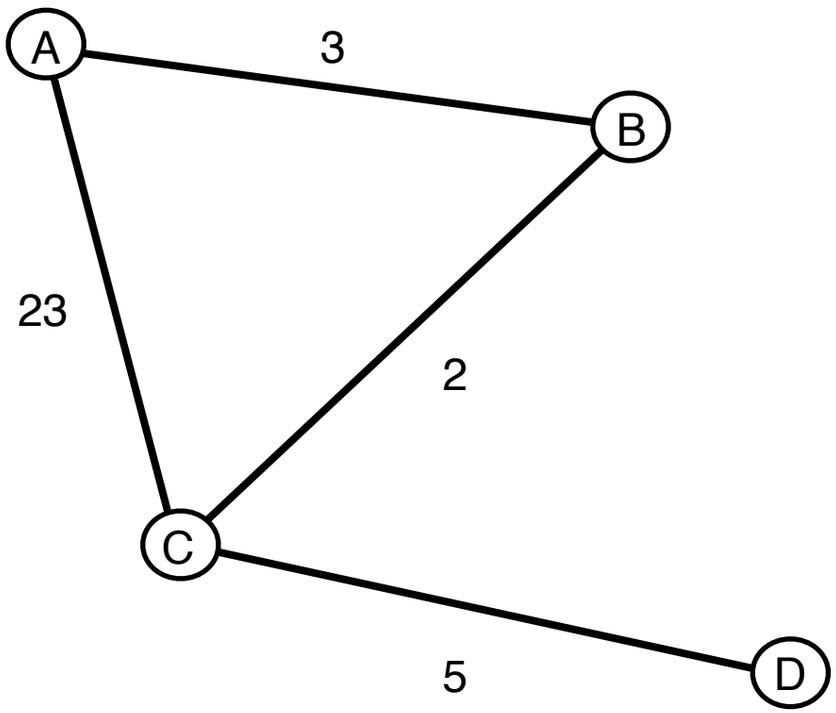
3. Warte auf Aufstellungen diese Informationen von anderen Routern und ergänze diese Informationen in der eigenen Kostenmatrix.

4. Ändern sich dadurch die minimalen Kosten, zu denen wir einen Router erreichen können:

   a) weiter mit Schritt 2

   b) sonst weiter mit Schritt 3.

Distance Vector Routing - Beispiel



T= 0:

1. Erzeuge eine Kostenmatrix: welche Router über welche Nachbarn und zu welchen Kosten erreichbar sind.

2. Erzeuge eine Aufstellung mit Informationen, welche Router zu welchen Kosten am besten erreichbar sind. Schicke diese Informationen an alle Nachbarn.

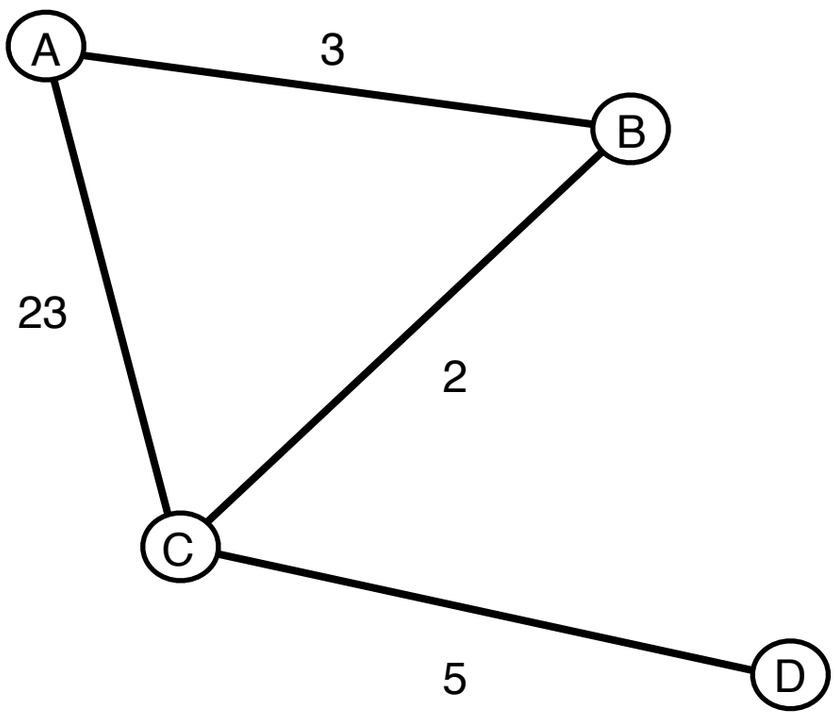
von A	via A	via B	via C	via D
zu A				
zu B		3		
zu C			23	
zu D				

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

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

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

Distance Vector Routing - Beispiel



T= 1:

3. Warte auf Aufstellungen diese Informationen von anderen Routern und ergänze diese Informationen in der eigenen Kostenmatrix. (Bekommen von B und C)

4.Ändern sich dadurch die minimalen Kosten, zu denen wir einen Router erreichen können: Zielrouter C und D - neuer bester Pfad -> Übertragung an Nachbarn (Grün)

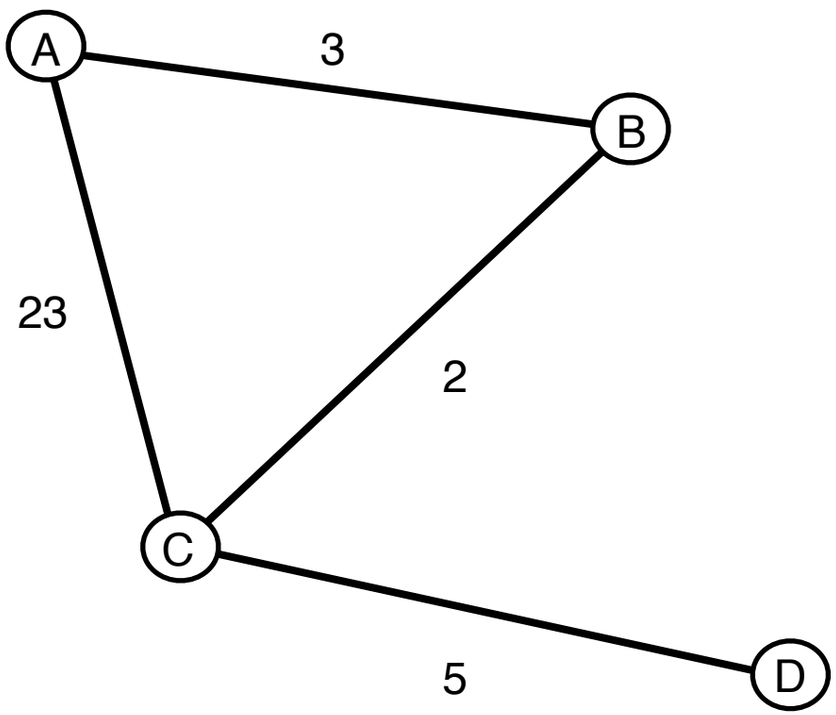
von A	via A	via B	via C	via D
zu A				
zu B		3	25	
zu C		5	23	
zu D			28	

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

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

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

Distance Vector Routing - Beispiel



T= 2:

3. Warte auf Aufstellungen diese Informationen von anderen Routern und ergänze diese Informationen in der eigenen Kostenmatrix. (Bekommen von B)

4.Ändern sich dadurch die minimalen Kosten, zu denen wir einen Router erreichen können: Ziel Router D von B günstiger - neuer bester Pfad -> Übertragung an Nachbarn (Grün)

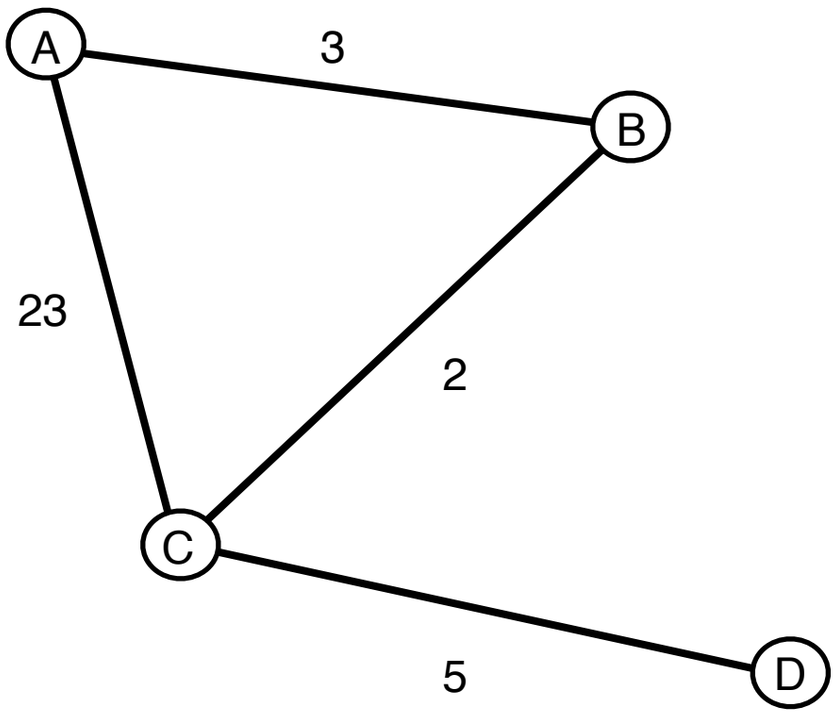
von A	via A	via B	via C	via D
zu A				
zu B		3	25	
zu C		5	23	
zu D		10	28	

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

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

Distance Vector Routing - Beispiel



T= 3:

3. Warte auf Aufstellungen diese Informationen von anderen Routern und ergänze diese Informationen in der eigenen Kostenmatrix. (Keine neuen Informationen)

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

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

Routingalgorithmen

Distance Vector Routing

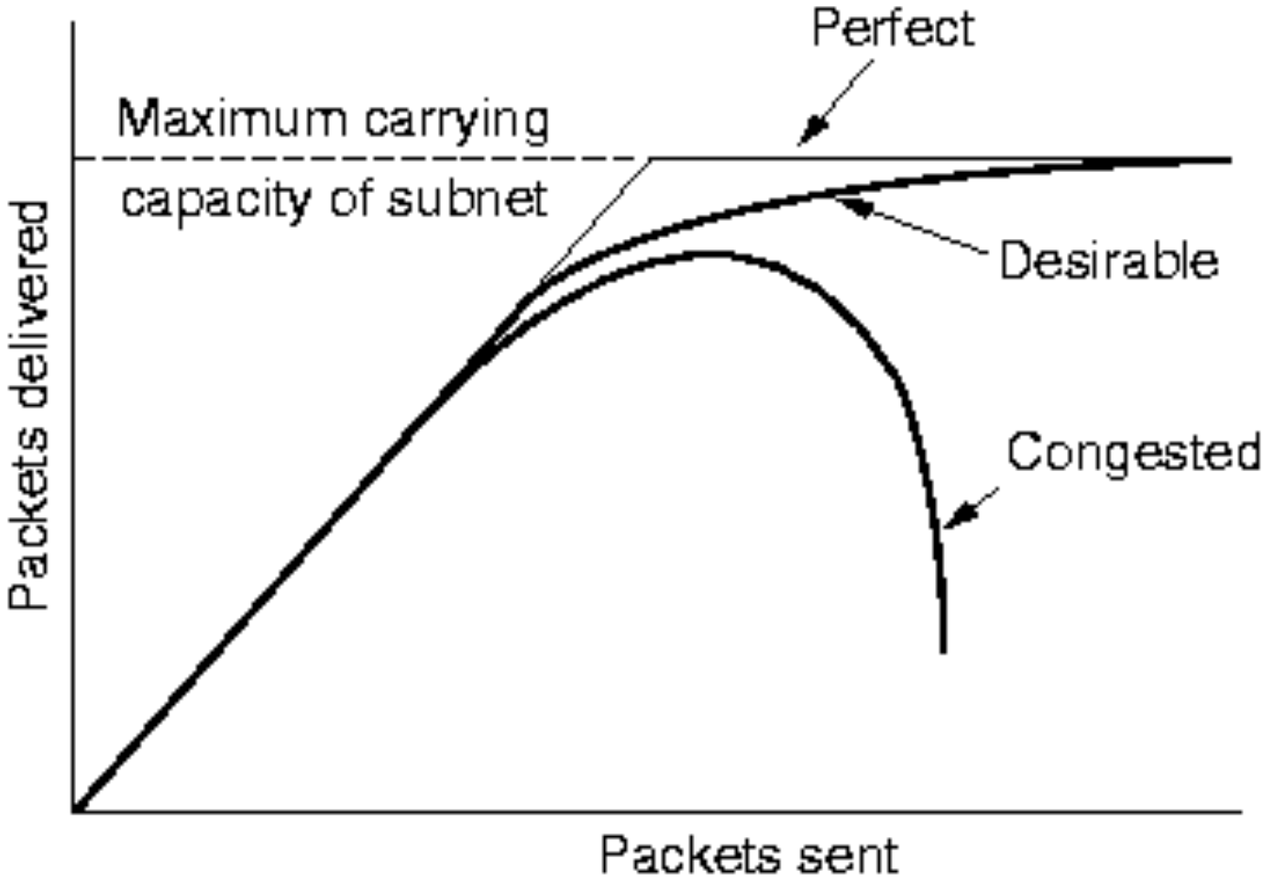
- Beim Distance Vector Routing tritt ein *Count To Infinity-Effekt* ein.
  - Hierbei schaukeln sich die Routing-Informationen hoch, was zu einer relativ langen Zeitdauer führt, bis ausgefallene Routen im Netz propagiert werden:
    - C meldet an A, dass D über C nur noch sehr schlecht zu erreichen ist -> keine Änderung da der bester Pfad über B führt.
    - B meldet an A, dass D über ihn nur noch sehr schlecht zu erreichen ist - Pfadkosten auf 13 gestiegen -> indirekte Route von B: B-A-B-C-D=13 + ursprüngliche Kosten von A zu D: A-b-C-D=10.
    - Da B den Router D nur noch zu Kosten von 13 erreichen kann, kann der Router A den Router D nur noch zu Kosten von 16 erreichen.
    - Dadurch Änderung des bester Pfad bei A, was an B propagiert wird - usw.
  - Dieses Problem lässt sich **für einfache Schleifen** mit dem *Split-Horizon-Verfahren* verhindern.
    - Eine Pfadinformation darf nicht über dasselbe Interface veröffentlicht werden, worüber sie empfangen wurde.

Routingalgorithmen

Distance Vector Routing

- Das Distance Vector Routing wurde im ARPANET verwendet und 1979 durch das Link State Routing ersetzt.
- In IPv4 wurde das Distance Vector Routing als RIPv1 und RIPv2 (Routing Information Protocol) implementiert.

Überlastkontrolle



- Solange sich die Anzahl der gesendeten Pakete **weit unterhalb der Kapazitätsgrenze** des Netzwerkes befindet, ist des Verhältnis zwischen **gesendeten Paketen und ausgelieferten Pakten proportional** (doppelte Menge gesendet - doppelte Menge ausgeliefert).
- Sobald sich allerdings die Anzahl der gesendeten Pakete der **Kapazitätsgrenze nähert**, füllen sich die Puffer der beteiligten Router. Wenn die **Puffer voll sind gehen die ersten Pakete verloren** und **die Anzahl der ausgelieferten Pakete sinkt** im Vergleich zu den gesendeten Paketen. **Das Netzwerk ist jetzt Überlastet.**
- Die Puffer der einzelnen Router zu vergrößern kann bis zu einer bestimmten Grenze hilfreich sein.
- Allerdings können **zu große Puffer** zu einem **Überlastungskollaps** führen:

  Pakete, die sehr lange in den Puffern verweilen werden **vom Empfänger nicht bestätigt und führen zu neu Übertragungen** (retransmits) bei den Sendern.

  Bei **großen Puffern sind mehr Pakete betroffen und führen zu mehr neu Übertragungen** (retransmits).

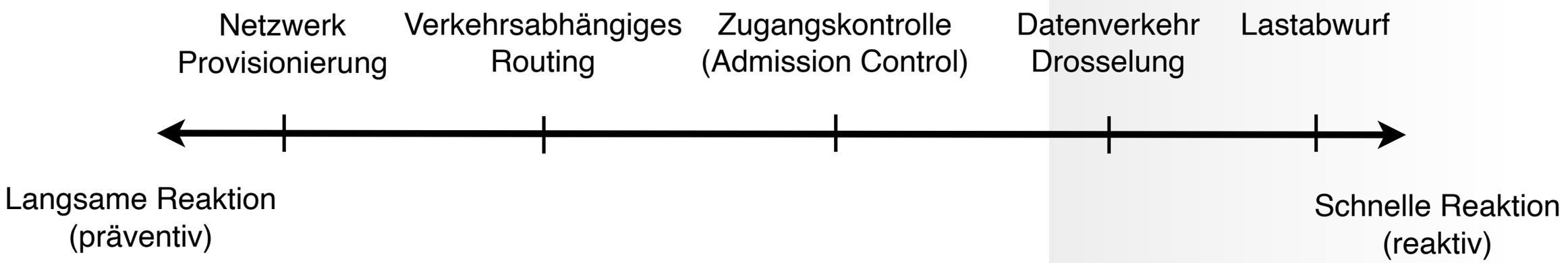
- **Langsame Router** oder Router welche über **langsame Anschlüsse verfügen** (langsamer als die Übertragungsrate des restlichen Netzes) können ebenfalls zu **Überlastungen führen**.

  Hier kann ein Ausweichen auf **alternative Routen** helfen.

Überlastkontrolle - Flusskontrolle

- Überlastkontrolle (Vermittlungsschicht):
  - Sicherstellen, dass ein Netzwerk in der Lage ist den aufkommenden Datenverkehr zu bewältigen.
  - Dies ist eine globale Aufgabe, welche alle beteiligten Rechner und Router einschließt.
- Flusskontrolle (Sicherungsschicht):
  - Regeln des Datenflusses zwischen einem Sender und einem Empfänger.
  - Verhindern, dass ein Sender durch zu schnelles senden einen Empfänger überfordert.

Maßnahmen zur Überlastkontrolle

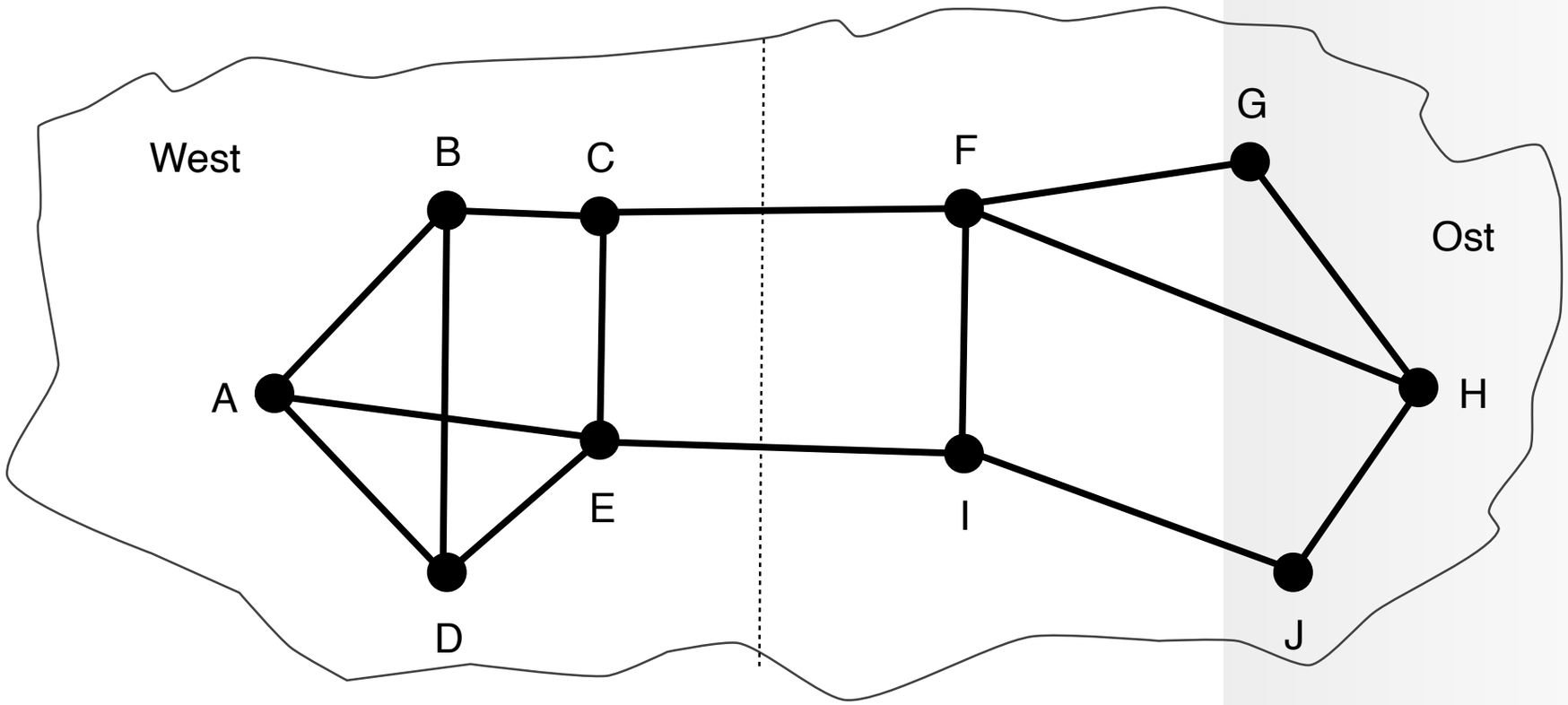


- Um eine (temporäre) Überlastsituation zu verhindern kann:
  1. Die Transportkapazität erhöht werden.
  2. Der Datenverkehr gemindert werden.
- Mögliche Maßnahmen lassen sich nach der Geschwindigkeit ihrer Wirkungsweise unterscheiden.

Verkehrsabhängiges Routing

- Berechnung der Kantengewichtung über:
  - Bandbreite
  - Paketverzögerung (durch Entfernung)
  - Verkehrsaufkommen oder Pufferverzögerung

Verkehrsabhängiges Routing - Schwingungszustand



- Hohes Verkehrsaufkommen von West nach Ost über C - F
  - Verkehrsabhängiges Routing -> E - I kleineres Kantengewicht
  - Hohes Datenverkehrsaufkommen nun über E - I -> C - F kleineres Kantengewicht
  - ...

Verkehrsabhängiges Routing - Schwingungszustand

- Ein solcher Schwingungszustand kann durch
  - Multipath-Routing (Mehrwegerouting) verhindert werden. Im Beispiel wird der Datenverkehr anteilig auf C-F und E-I verteilt.
  - langsames wechseln des Verkehrs verhindert werden. Damit hat der Routingalgorithmus Zeit zu konvergieren.

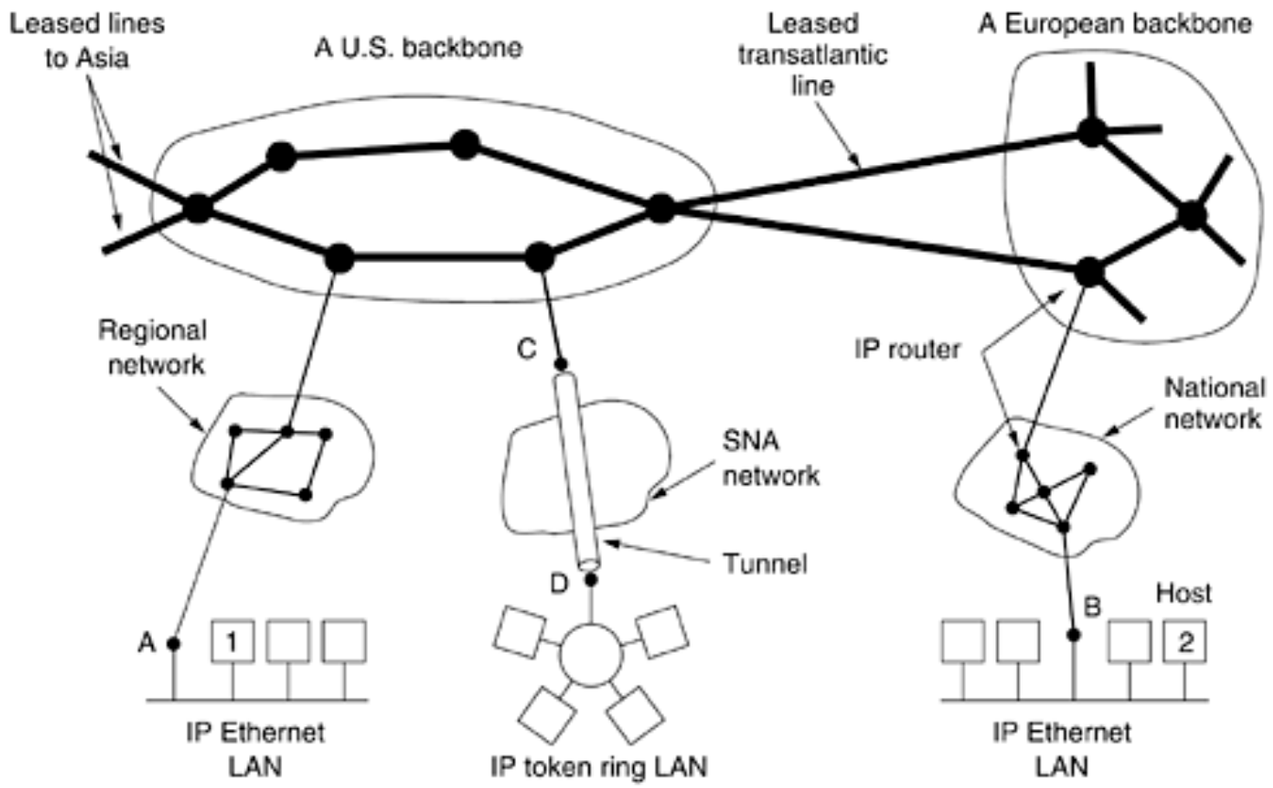
Überlastkontrolle

Zugangskontrolle (Admission Control)

- Leicht umzusetzen in virtuell verbindungsorientierten Netzwerken.
  - Wenn eine zusätzliche Verbindung zu einer Überlastsituation führen würde, werden keine Verbindungen zugelassen solange keine abgebaut wird.
- In Paketorientierten Netzwerken ist es Problematisch diese Grenze zu definieren.
  - Eine 100 MBit/s Leitung wird in 10 Segmente à 10 MBit/s aufgeteilt.
    - ineffiziente Bandbreitenauslastung da die 10MBit/s nicht immer erreicht werden.
  - Die Anzahl der virtuellen Verbindungen kann mit statistischen Werten über eine Heuristik berechnet werden.
- Eine Kombination aus Zugangskontrolle und verkehrsabhängigen Routing ist möglich.
  - Beim Verbindungsaufbau können Informationen aus dem verkehrsabhängigen Routing benutzt werden um potenziell von hohem Verkehr belastete Routen zu umgehen.

Vermittlungsschicht im Internet

Internet: Netzwerk von verschiedenen Netzwerken



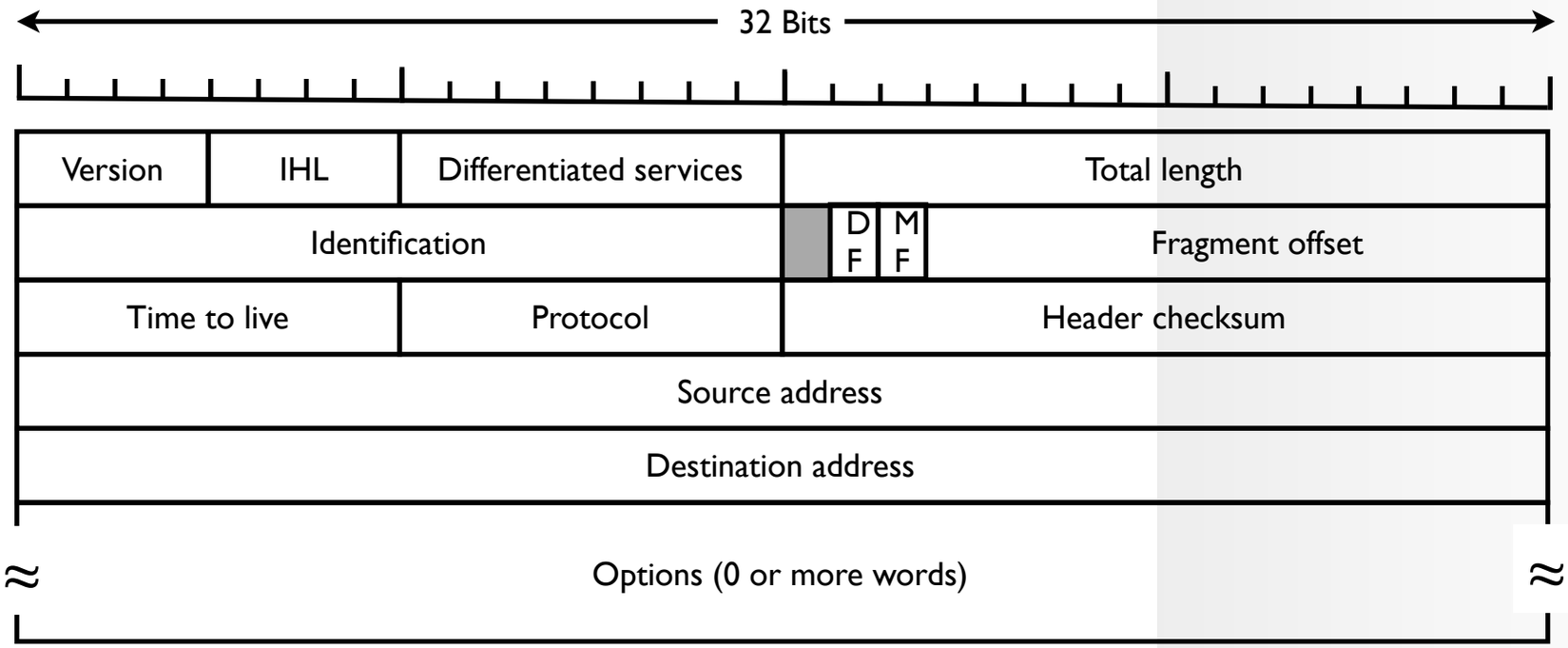
Computer Networks, Fifth Edition by Andrew Tanenbaum and David Wetherall, © Pearson Education-Prentice Hall, 2011

- Das Internet wird durch das Vermittlungsschicht Protokoll "**IP - Internet Protocol**" zusammengehalten.
- Im Gegensatz zu vielen anderen Netzwerken war das Internet von **Anfang an als Kopplung von Netzwerken** gedacht.
- Die Aufgabe der Vermittlungschicht (IP) ist es Pakete verbindungslos von einer Quelle zu einem Ziel zu vermitteln, **ohne dabei zu berücksichtigen in welchen Netzen sich die Quelle oder das Ziel befindet**.
- Eine beispielhafte Kommunikation könnte folgendermaßen aussehen:

1. Die Transportschicht erzeugt aus einen Datenstrom einzelne Datagramme. Diese Datagramme können max. 64 KBytes groß sein. meistens werden sie jedoch auf **1500 Bytes beschränkt - damit sie in einen Ethernet-Frame passen** (MTU - Maximum Transmission Unit).
2. Jedes Datagramm wird durch das Internet transportiert und dabei u.U. in weitere kleinere Einheiten zerlegt.
3. Am Ziel angekommen, werden die einzelnen Einheiten von der Vermittlungsschicht (IP) wieder zu dem Datagramm zusammengesetzt und an die Transportschicht übergeben.

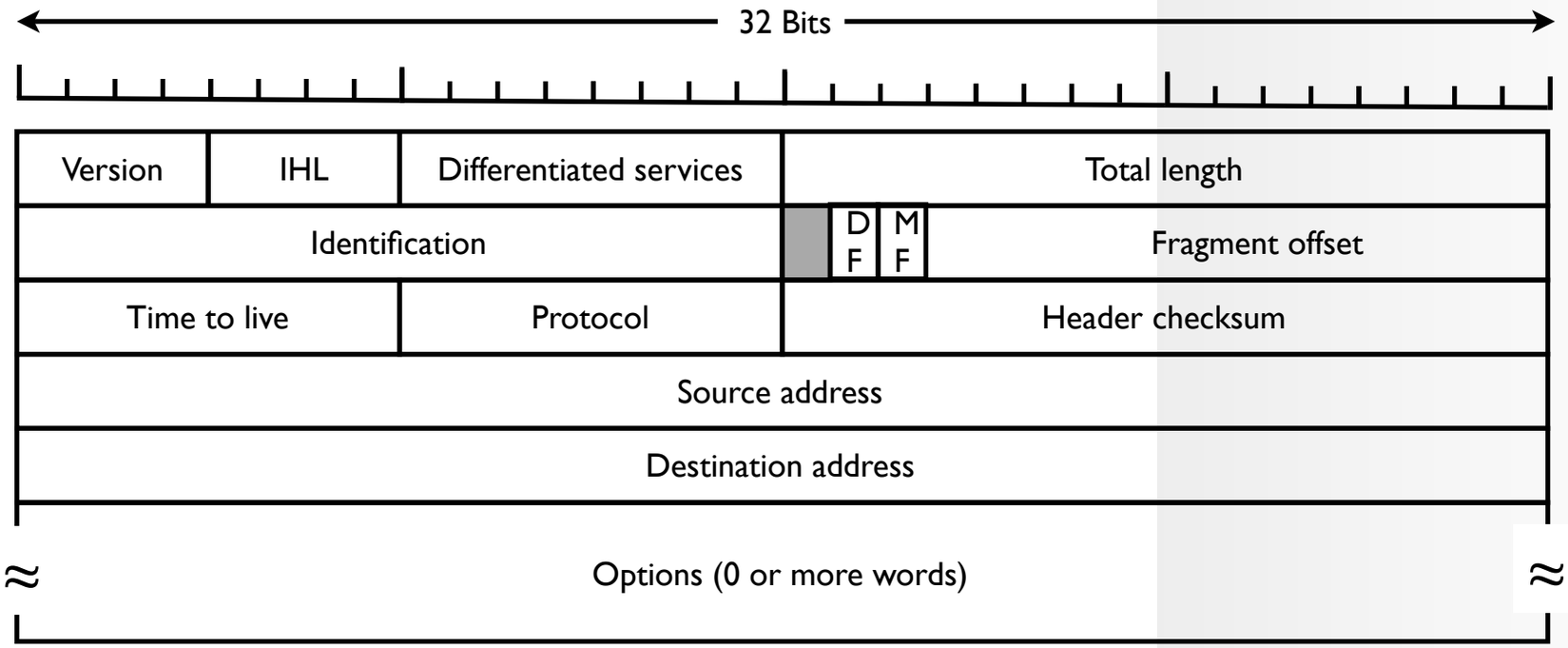
- Bei einer solchen Kommunikation kann sein, **dass ein Paket von der Quelle bis zum Ziel mehrere verschiedene Netzwerke durchläuft.**

Das IP Version 4 Protokoll (IPv4) - Format des IPv4 Datagramms



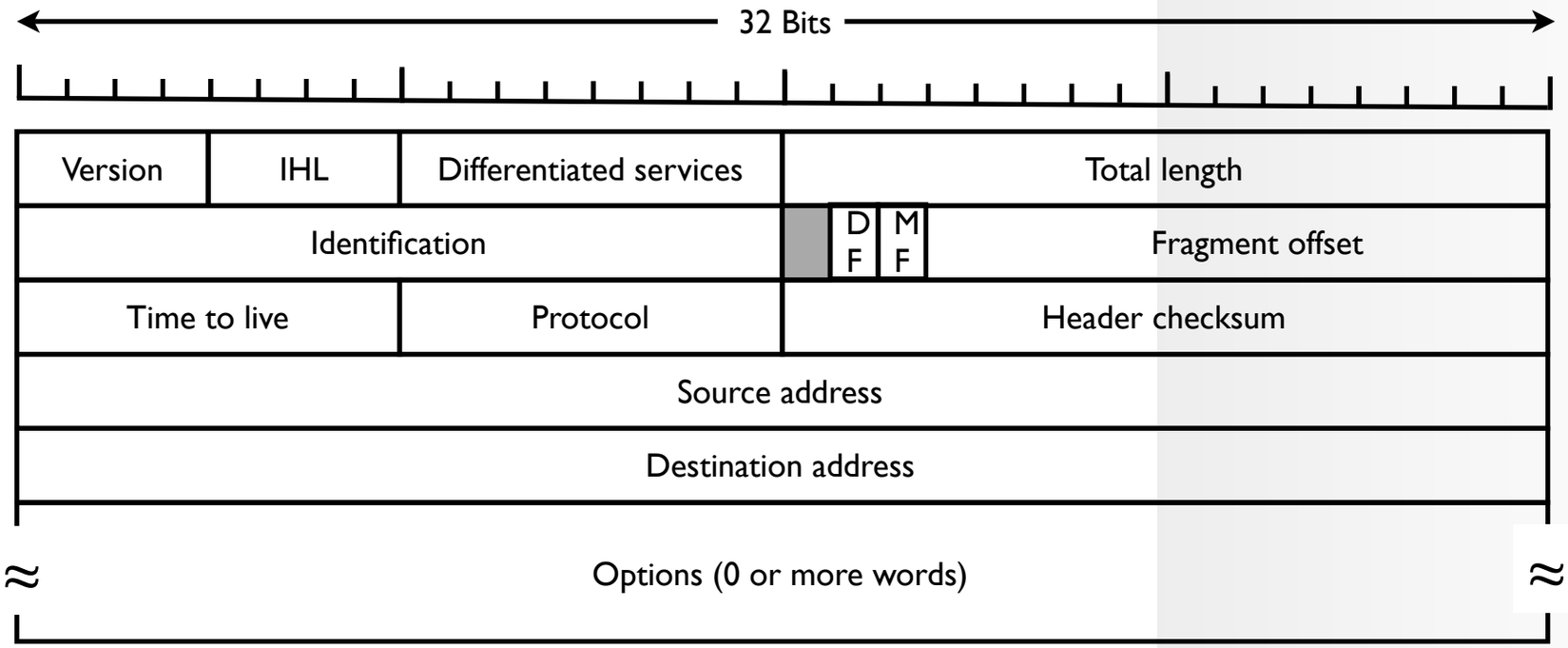
- Ein IPv4 Datagramm besteht aus einem Header und den Daten (Payload).
  - Der Header besteht aus einem festen 20 Byte (4 (32Bit) Byte x 5) großen Teil und einem variablen großen optionalen Teil.
  - Die Bits werden von links nach rechts (höchstes “Version” Bit zuerst) übermittelt (Big-Endian).

Das IP Version 4 Protokoll (IPv4) - Format des IPv4 Datagramms



- Version
  - Gibt die Version des Protokolls an (4 für IPv4)
- IHL
  - Länge der Headers in 32 Bit Wörtern (max. 2^4 = 15 -> 15*23 Bit = 480 Bit = 60 Bytes). Damit ist der optionale Teil (“Options”) max. 40 Byte lang (60 - 20 Byte)

Das IP Version 4 Protokoll (IPv4) - Format des IPv4 Datagramms

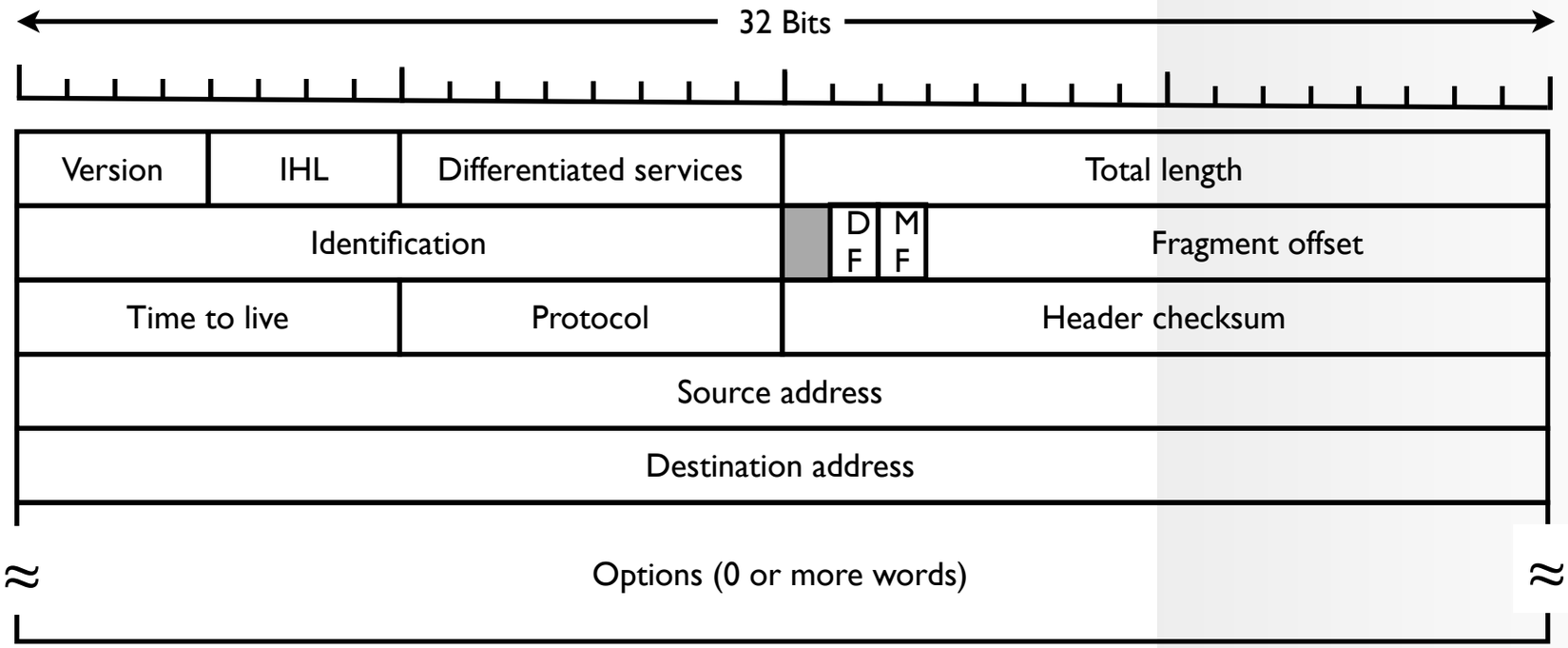


- Differentiated services

  - **Ursprünglich "Type of Service":** 3 Bits für Priorität und 3 Bits für Verzögerung, Datendurchsatz oder Zuverlässigkeit.

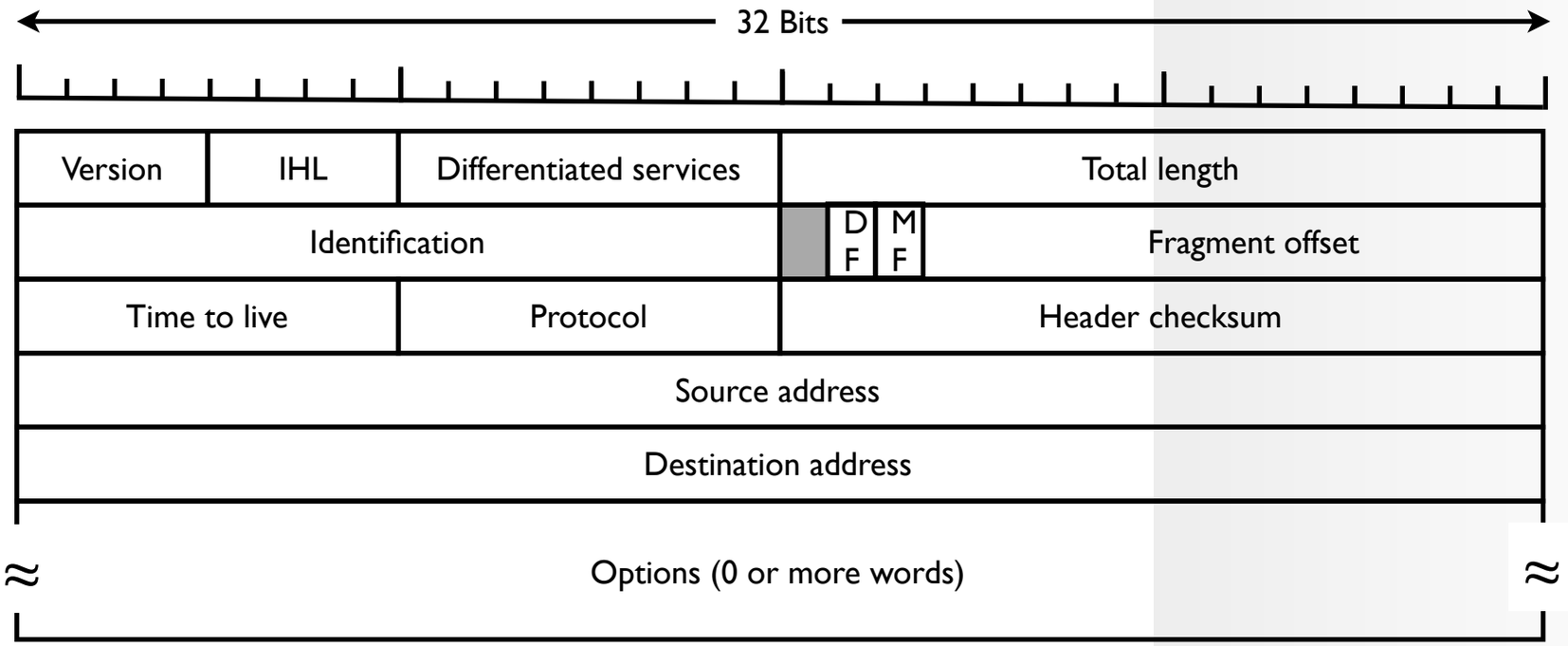
  - **Heute** werden die oberen 6 Bits für die Service Klasse (Class of Service) verwendet und die untern 2 Bits um Netzüberlastungen zu kennzeichnen.

Das IP Version 4 Protokoll (IPv4) - Format des IPv4 Datagramms



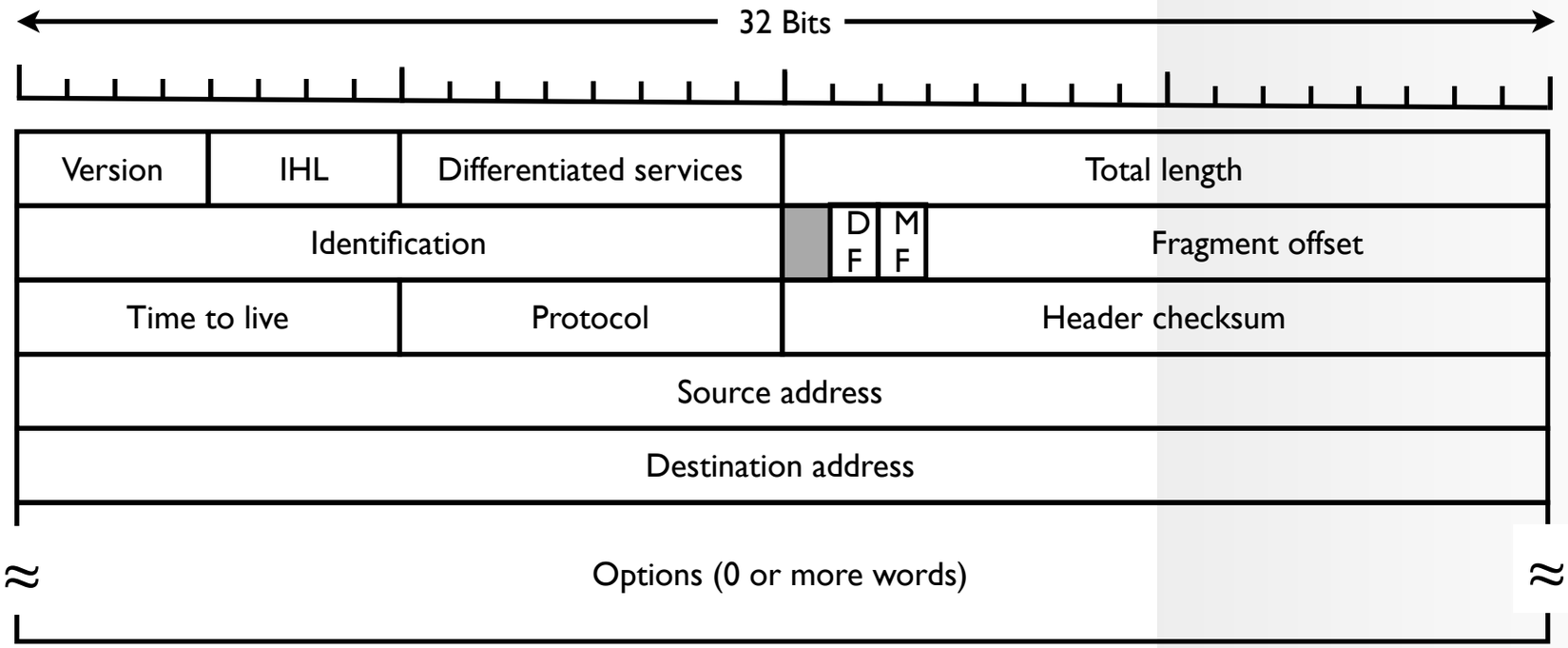
- Total Length
  - Die gesamte Länge des Datagramms, also Header + Daten (Payload)
  - Die maximale Länge ist 65535 Bytes

Das IP Version 4 Protokoll (IPv4) - Format des IPv4 Datagramms



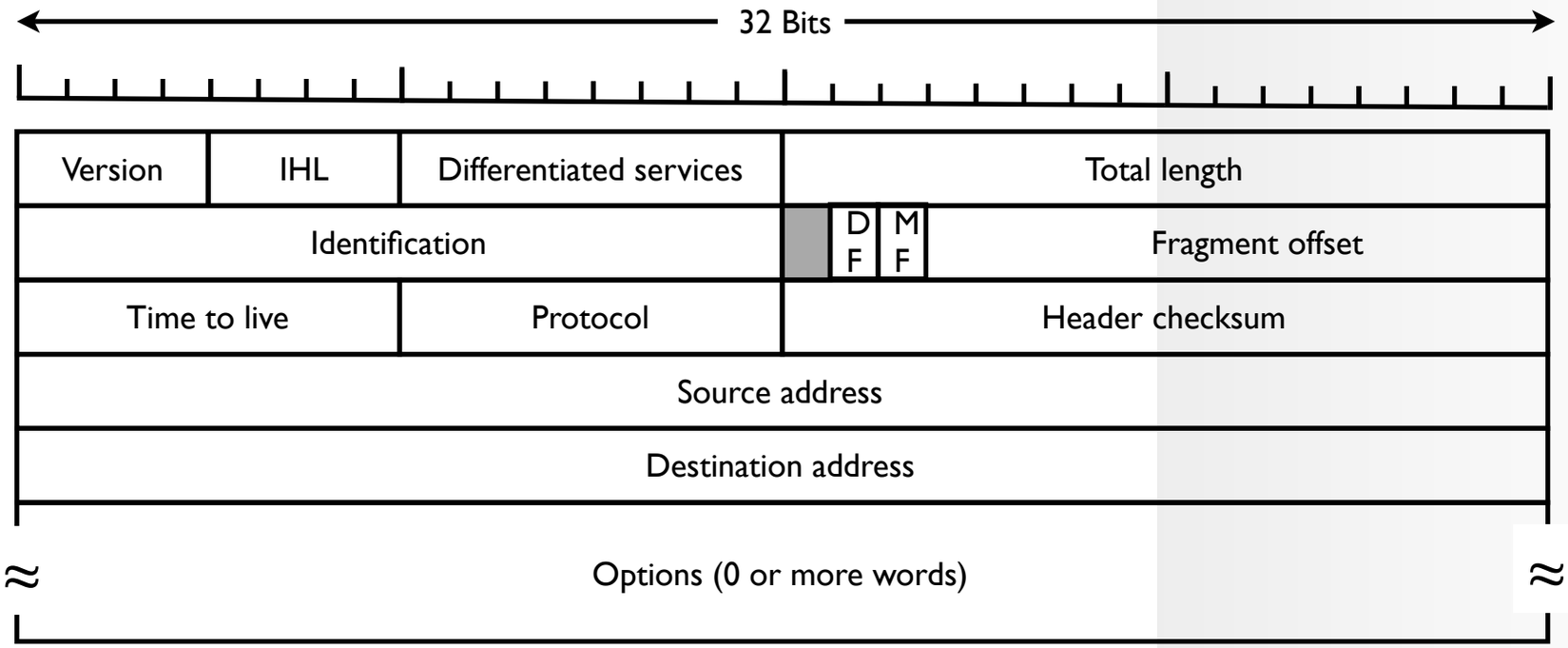
- Identification
  - Dieses Feld erlaubt eine eindeutige Identifizierung des Paketes zu dem ein Fragment gehört.
  - Jedes Fragment welches zu einem Paket gehört hat den selben “Identification” Wert.

Das IP Version 4 Protokoll (IPv4) - Format des IPv4 Datagramms



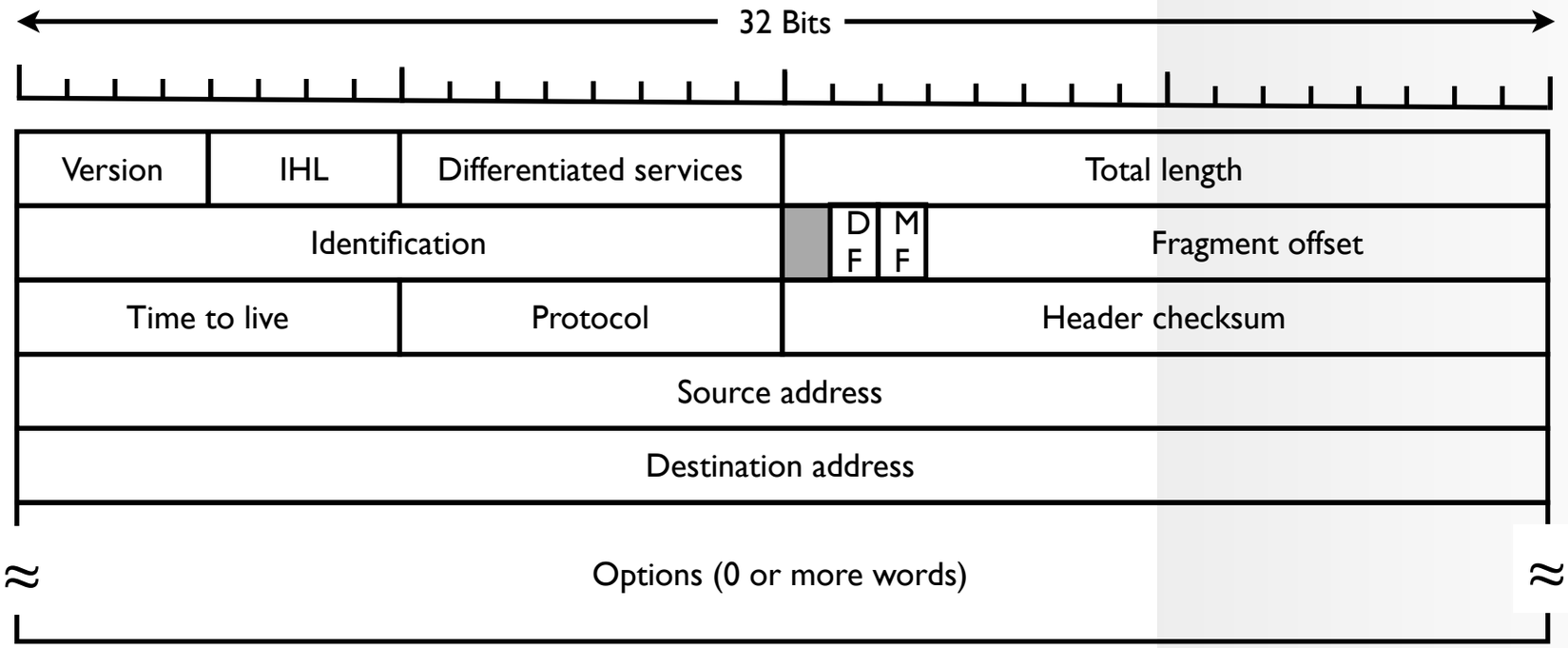
- “DF” - Don’t Fragment
  - Wenn diese Bit gesetzt ist darf das Paket nicht fragmentiert werden (Bestimmung der Pfad MTU - Maximum Transmission Unit).
- “MF” - More Fragments
  - Alle, außer das letze Fragment eines Paketes haben dieses Bit gesetzt und zeigen damit an, dass noch mehr Fragmente folgen.

Das IP Version 4 Protokoll (IPv4) - Format des IPv4 Datagramms



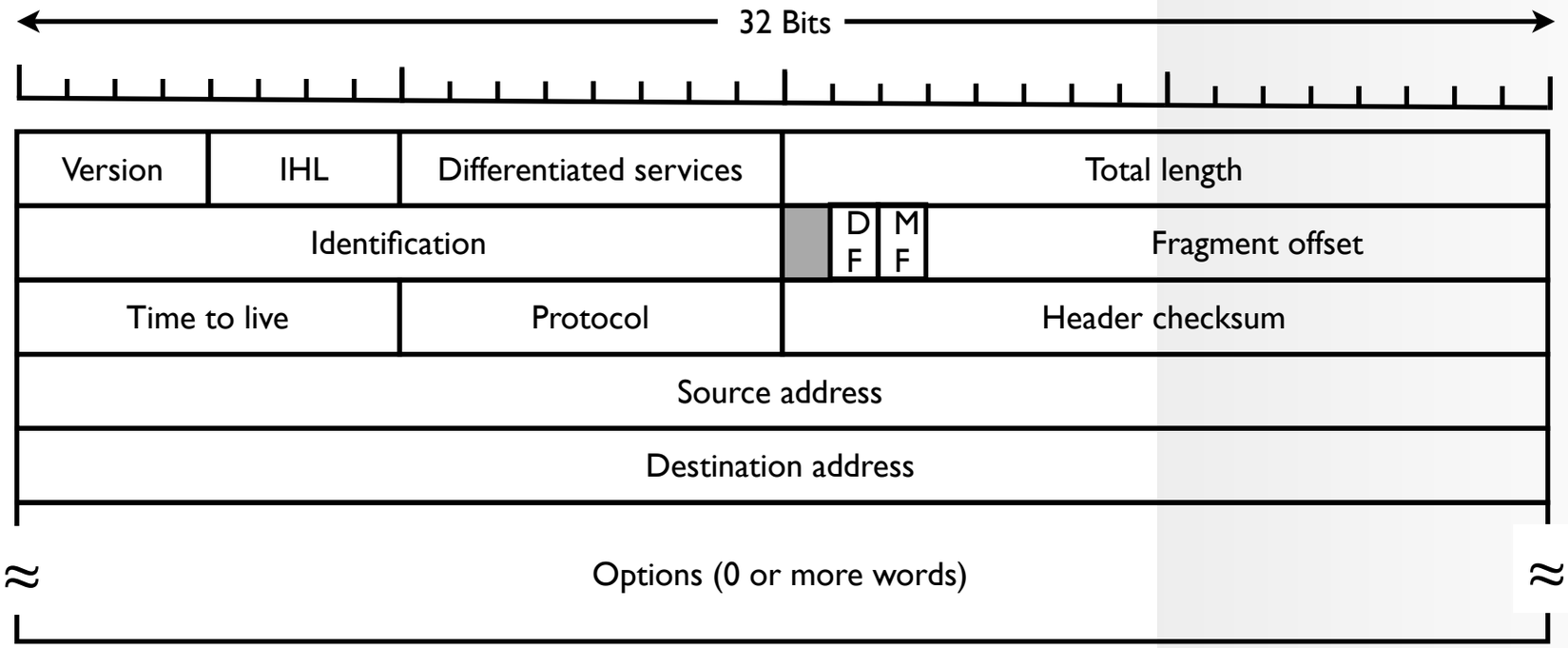
- Fragment offset
  - Bestimmt die Position eines Fragmentes innerhalb eines Datagramms.
- Time to live
  - Der Maximalwert (255) wird in jedem Router um eins erniedrigt, wenn der Wert Null erreicht, wird das Paket verworfen und der Sender darüber informiert.

Das IP Version 4 Protokoll (IPv4) - Format des IPv4 Datagramms



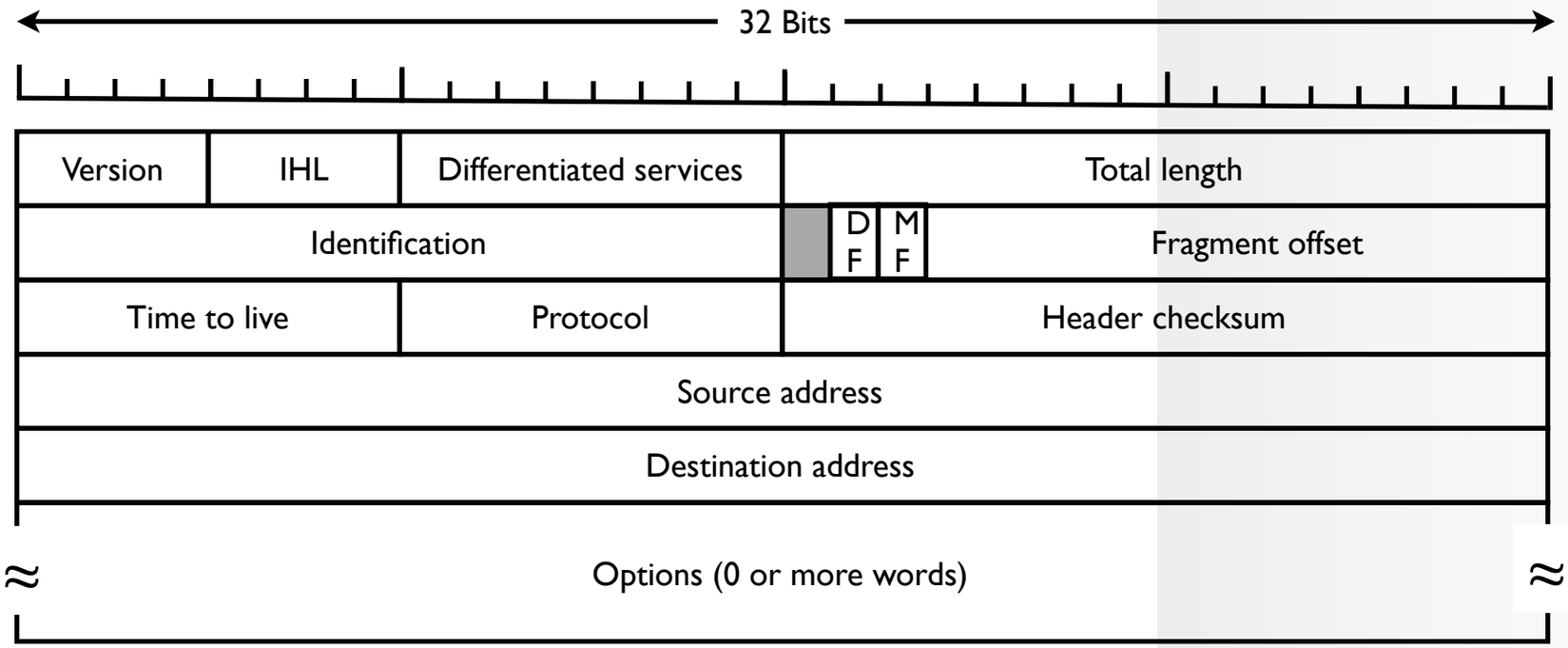
- Protocol
  - Bestimmt an welchen Prozess der Transportschicht das Paket übergeben werden soll. TCP, UDP oder andere Werte sind möglich.

Das IP Version 4 Protokoll (IPv4) - Format des IPv4 Datagramms



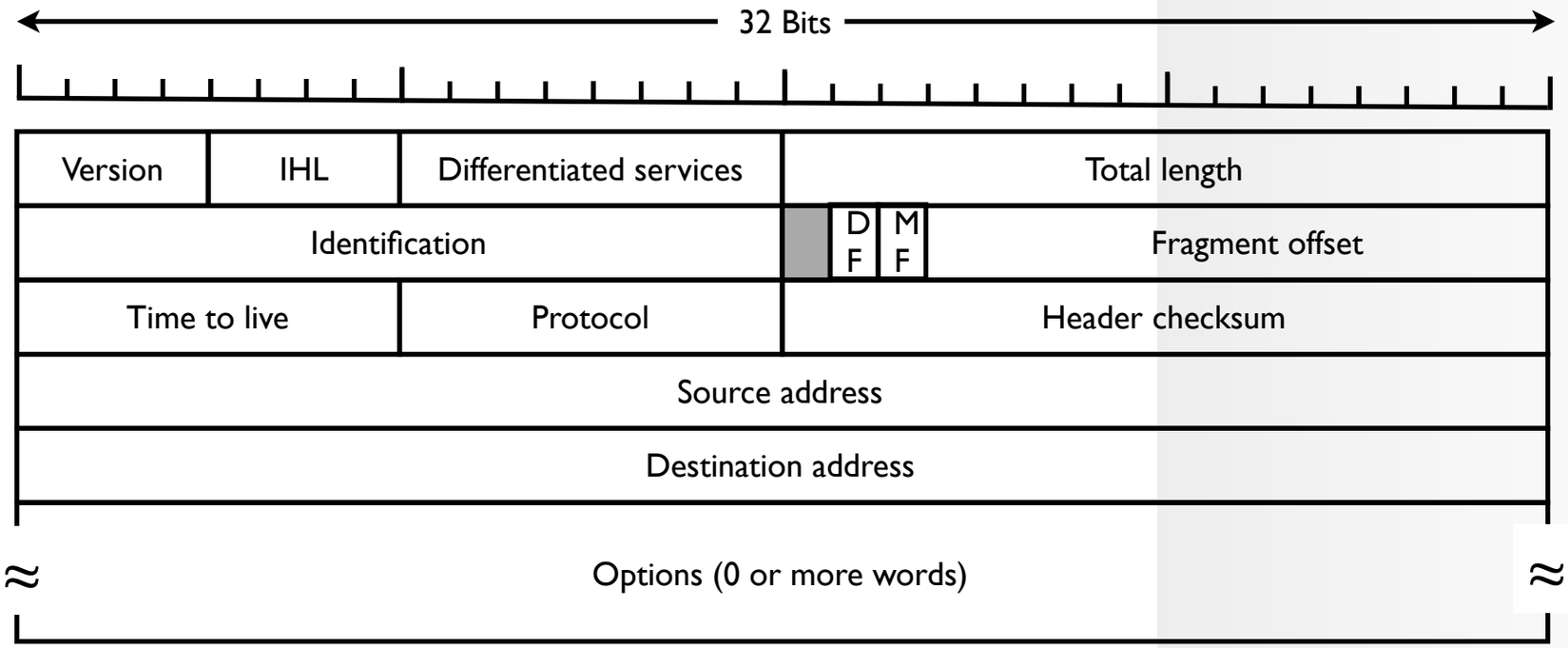
- Header checksum
  - Eine Prüfsumme für den Header (Einskomplement aus der Summe der 16Bit Header Halbwörter)
  - Wegen TTL muss die Prüfsumme in jedem Router neu berechnet werden (Prüfsummenalgorithmus erlaubt auch einfaches Inkrementieren)

Das IP Version 4 Protokoll (IPv4) - Format des IPv4 Datagramms



- Source address
  - Die Quelladresse des Paketes (32 Bit)
- Destination address
  - Die Zieladresse des Paketes (32 Bit)

Das IP Version 4 Protokoll (IPv4) - Format des IPv4 Datagramms



- Options
  - Folgende Zusatzoptionen sind möglich:
    - Strict Routing: Option gibt den kompletten Pfad an, welchen das Paket durchlaufen muss
    - Free Routing: Option gibt eine Liste von Routern an, die vom Paket nicht verfehlt werden dürfen
    - Record Route: Lässt die komplette Route aufzeichnen (Heute reicht die Größe des Option-Feldes meist nicht mehr dafür aus)
    - Time Stamp: Zeitstempel
    - Security: Bezeichnet, wie geheim das Paket ist

IP Adressen

- Jeder Rechner im Internet muss über mind. eine IP Adresse verfügen.
  - IP Adressen identifizieren nicht Rechner sondern Netzwerkkarten
- Jeder Router im Internet verfügt über eine IP Adresse, welche in den “Source address” bzw. “Destination address” Feldern des IP Headers benutzt werden können.
- Router verfügen über mehr Netzwerkkarten und damit über mehrere IP Adressen.

IP Adressen - Notation

- Die 32 Bit IP Adressen werden in 4 Blöcken zu je einem Byte dezimal und durch Punkte (.) getrennt angegeben.
  - Beispiel: `128.208.2.151`
  - binäre Darstellung: `10000000.11010000.00000010.10010111`
    `128 .208 .2 .151`
- IP Adressen teilen sich in zwei Bereiche auf:
  - Netzbereich: Bereich der zum Netz gehört
  - Hostbereich: Bereich der zu den einzelnen Geräten gehört

Vermittlungsschicht im Internet

Adressklassen

- Es existieren fünf verschiedene Adressklassen, wobei sich über die ersten 4 bits die jeweilige Adressklasse bestimmen lässt.

Klasse	Bereich	Netz-ID/Host-ID
A	0.0.0.0 - 127.255.255.255	0[7 bit] / [24 bit]
B	128.0.0.0 - 191.255.255.255	10[14 bit] / [16 bit]
C	192.0.0.0 - 223.255.255.255	110[21 bit] / [8 bit]
D	224.0.0.0 - 239.255.255.255	1110[Multicast ID - 28 bit]
E	240.0.0.0 - 255.255.255.255	11110[Reserviert 27 bit]

- D: Bildung von IP Multicast Gruppen (268 435 456)
- F: Reservierte Adressen für zukünftige Anwendungen

Vermittlungsschicht im Internet

Adressklassen

- Der Präfix gibt dabei die Nummer des Netzes an. Diese Nummer wird für Firmen und Einrichtungen global von der IANA – Internet Assigned Numbers Authority koordiniert und auf Antrag vergeben.

- Der Suffix bestimmt die einzelnen Rechner (Host) der antragstellenden Institution und wird durch diese festgelegt.

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Klasse A	0	Präfix							Suffix																							
Klasse B	1	0	Präfix														Suffix															
Klasse C	1	1	0	Präfix																					Suffix							

- Die maximale Anzahl der Rechner (Hosts) war damit $2^{Suffix} - 2$
  - die erste mögliche Adresse - alle Suffixbits haben den Wert Null, da sie das Netz bezeichnet
  - die letzte mögliche Adresse - alle Suffixbits haben den Wert Eins, da sie als Broadcast-Adresse reserviert ist.

Klassenlose Adressierung

- bei der klassenlosen Adressierung wird die starre Abgrenzung des Präfix vom Suffix aufgegeben und eine variable Bitgrenze zuzulassen.
- Dazu bedarf es einer weiteren 32-Bit-breiten Angabe, der Netzmaske oder Subnetzmaske.
- Die Netzadresse erhält man in dem man die IP-Adresse logisch UND-verknüpft.
- Zur Ermittlung der Hosts wird die Netzmaske logisch negiert (NICHT) und das Ergebnis mit der IP-Adresse logisch UND-verknüpft.

Klassenlose Adressierung - Beispiel

IP-Adresse 195.13.132.163, Subnetzmaske 255.255.255.224

Berechnung Netzadresse:

```
IP-Adresse     : 195.013.132.163   11000011 00001101 10000100 10100011
UND Netzmaske  : 255.255.255.224   11111111 11111111 11111111 11100000
Netzadresse    : 195.013.132.160   11000011 00001101 10000100 10100000
```

Berechung Hostadresse:

```
IP-Adresse         : 195.013.132.163   11000011 00001101 10000100 10100011
UND NOT Netzmaske  : 000.000.000.031   00000000 00000000 00000000 00011111
Hostnummer         :               3   00000000 00000000 00000000 00000011
```

CIDR-Notation (Classless Inter-Domain Routing)

- Die CIDR Notation ermöglicht eine kurze und prägnante Darstellung der IP-Adresse und Netzmaske.
- Dabei wird die IP-Adresse in üblicher Schreibweise dargestellt, gefolgt von der Anzahl der Bits, welche für die Netzmaske verwendet wird.
- Beispiel:
  - 195.13.132.163, Subnetzmaske 255.255.255.224: 195.13.132.163/27

CIDR-Notation (Classless Inter-Domain Routing) - Beispiel

- Anzahl der adressierbaren Hosts im zugeteilten Netz 195.13.132.160/27:
  - **30** adressierbare Hosts da: $\frac{2^{32}}{2^{27}} - 2 = 2^5 - 2 = 32 - 2 = 30$
- Adressmaske für Konfiguration in diesem Netz:
  - Letztes Byte - 5 Bits für Hosts, damit letztes Byte: $11100000_2 = 224$
  - Netzmaske: **255.255.255.224**

CIDR-Notation (Classless Inter-Domain Routing) - Übungsaufgabe

Gegeben ist ein Netz mit 194.95.66.16/28

Wieviele Hosts lassen sich im diesem Netz adressieren?

Mit Welcher Netzwerkmaske müssen die Hosts in diesem Netz konfiguriert werden?

Befindet sich der Host mit der IP Adresse 194.95.66.27 in diesem Netz?

NAT (Network Address Translation)

- NAT ist ein Verfahren, bei dem automatisiert Adressinformationen in Datenpaketen durch andere ersetzt bzw. ergänzt werden, um verschiedene Netze zu verbinden.
  - mehrere *private IP-Adressen/LAN Adressen* werden auf eine *öffentliche IP-Adresse/WAN Adresse* abgebildet.
- NAT wird in zwei Varianten verwendet, die sich durch die Kommunikationsrichtung unterscheiden:
  - **Source-NAT, Masquerading:** Änderung der Quell-Adresse
  - **Destination-NAT, Portforwarding:** Änderung der Ziel-Adresse
- SNAT und DNAT können auch in Kombination verwendet werden (statisches NAT).

Private IP Adressen

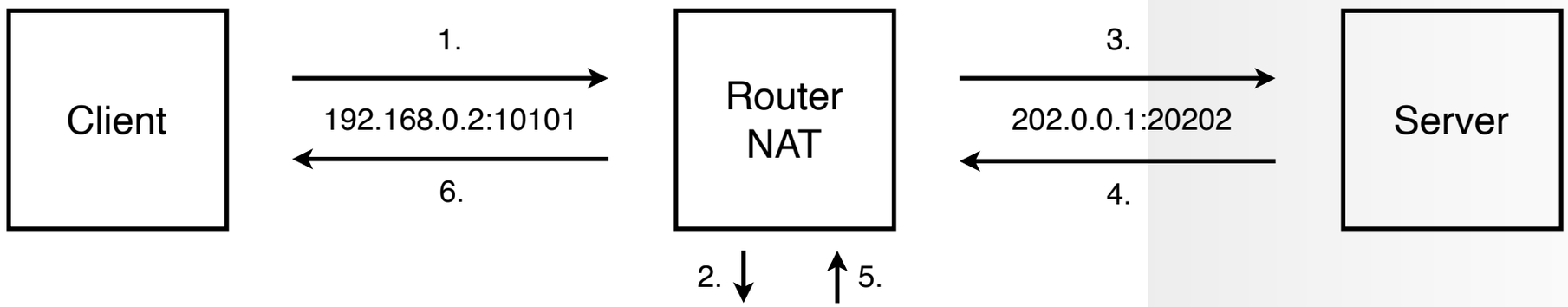
- Private IP Adressbereiche:

Klasse	Private Netzwerke	Netzmaske
A	10.0.0.0	255.0.0.0
B	172.16.0.0 - 172.31.0.0	255.240.0.0
C	192.168.0.0	255.255.0.0

- Diese können mit NAT mehrmals verwendet werden.
- Dürfen aber nicht als öffentliche IP Adressen (ohne NAT) verwendet werden!

NAT (Network Address Translation) - SNAT, Masquerading

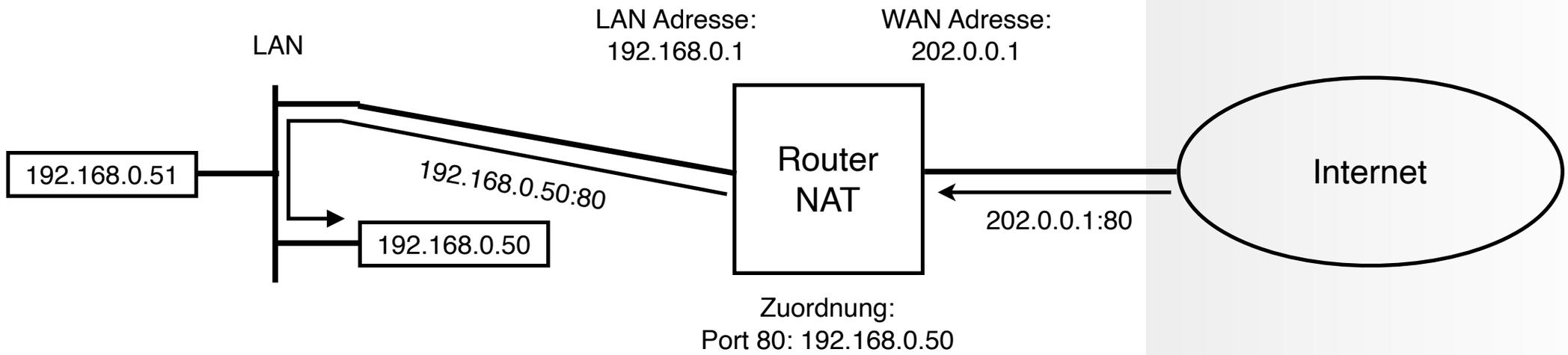
Verbindung kommt von innen => Router weiß von wem und wohin



LAN Port	LAN Adresse	WAN Port
10101	192.168.0.2	20202
:	:	:

NAT (Network Address Translation) - DNAT, Port Forwarding

Verbindung kommt von außen (Home Assistant Client zB) => Port muss geöffnet werden..
damit Router weiß, wohin das Paket geschickt wird

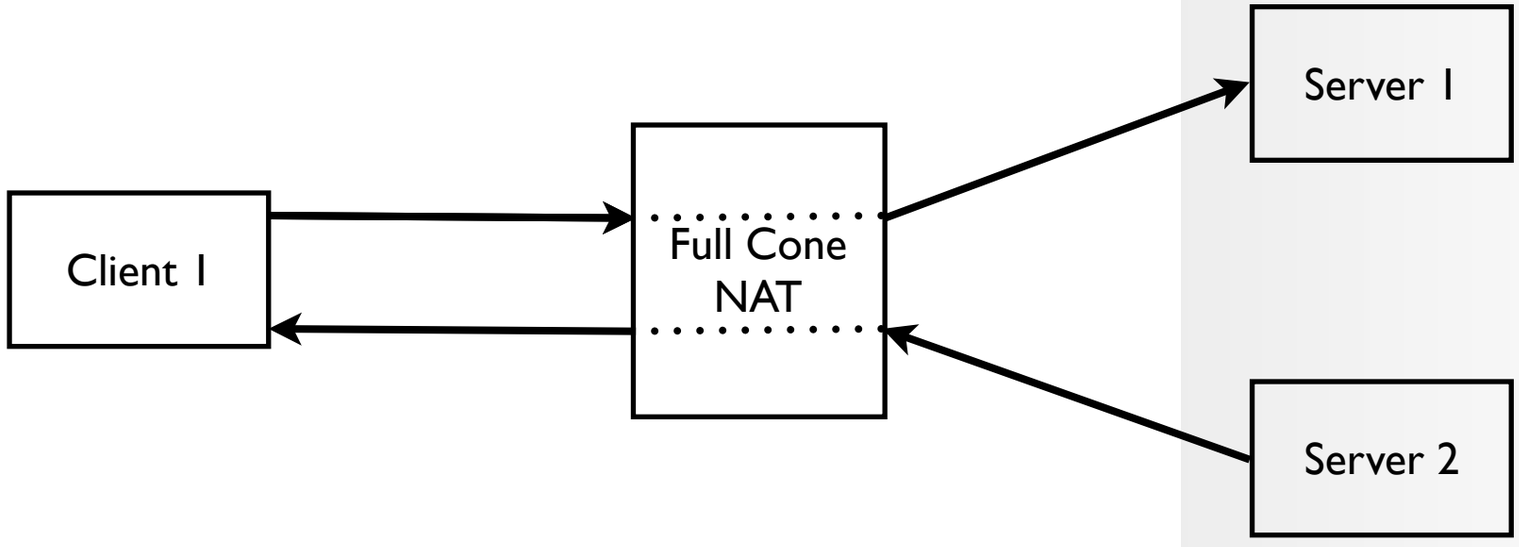


NAT (Network Address Translation) - Kategorien

- Full Cone NAT

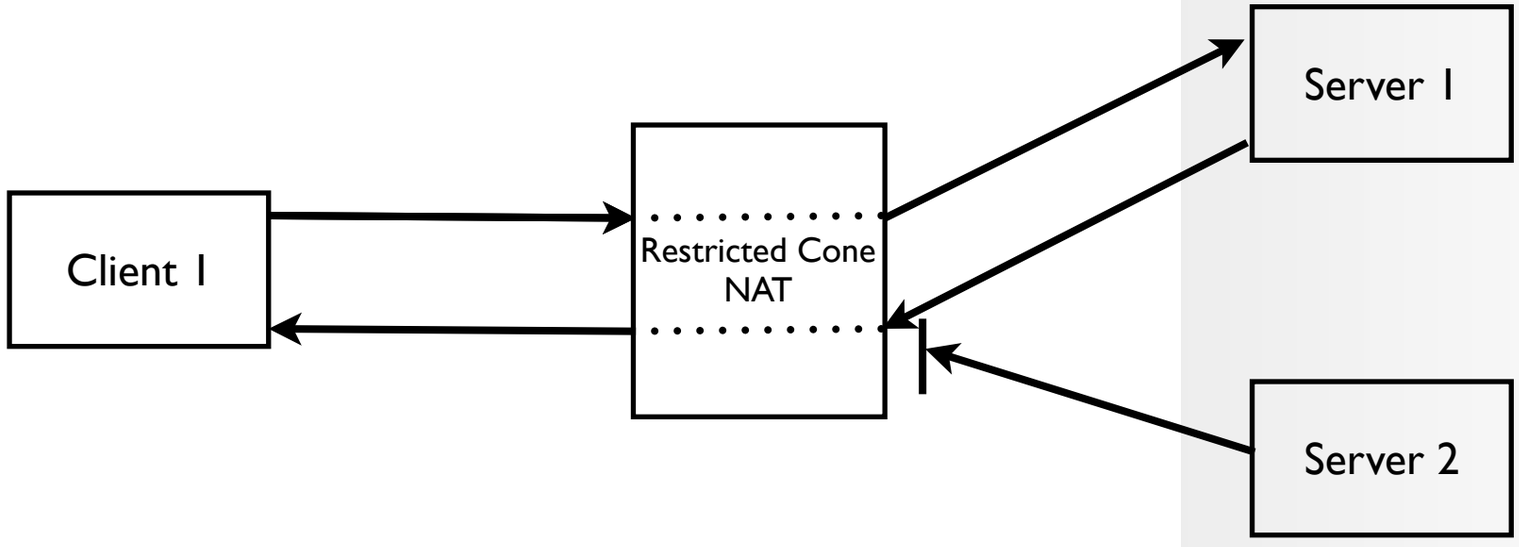
Kategorien nicht so wichtig (eher irrelevant für KA, Abwandlungen von SNAT und DNAT)

  - externe Hosts können über die externe Adresse des NAT-Gateways Verbindungen zu internen Hosts aufbauen.



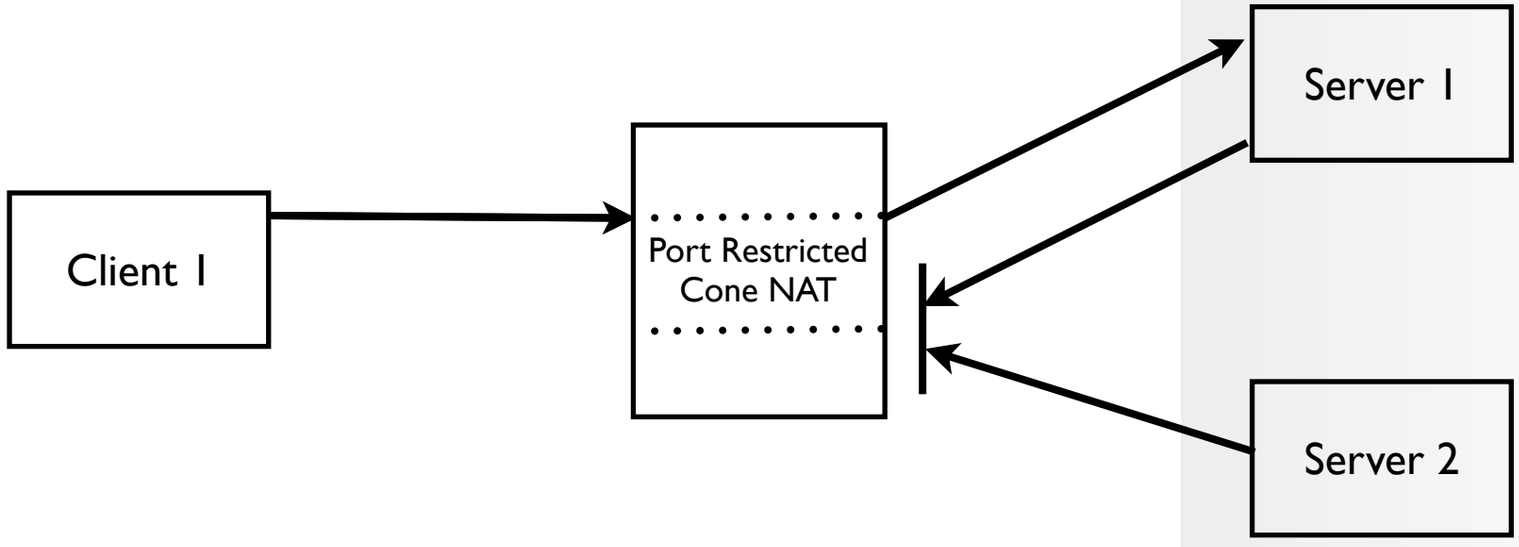
NAT (Network Address Translation) - Kategorien

- Restricted Cone NAT
  - externe Hosts können können nur mit internen Host Kontakt aufnehmen wenn diesem Verbindungsversuch eine Kontaktaufnahme des internen Hosts vorausging.



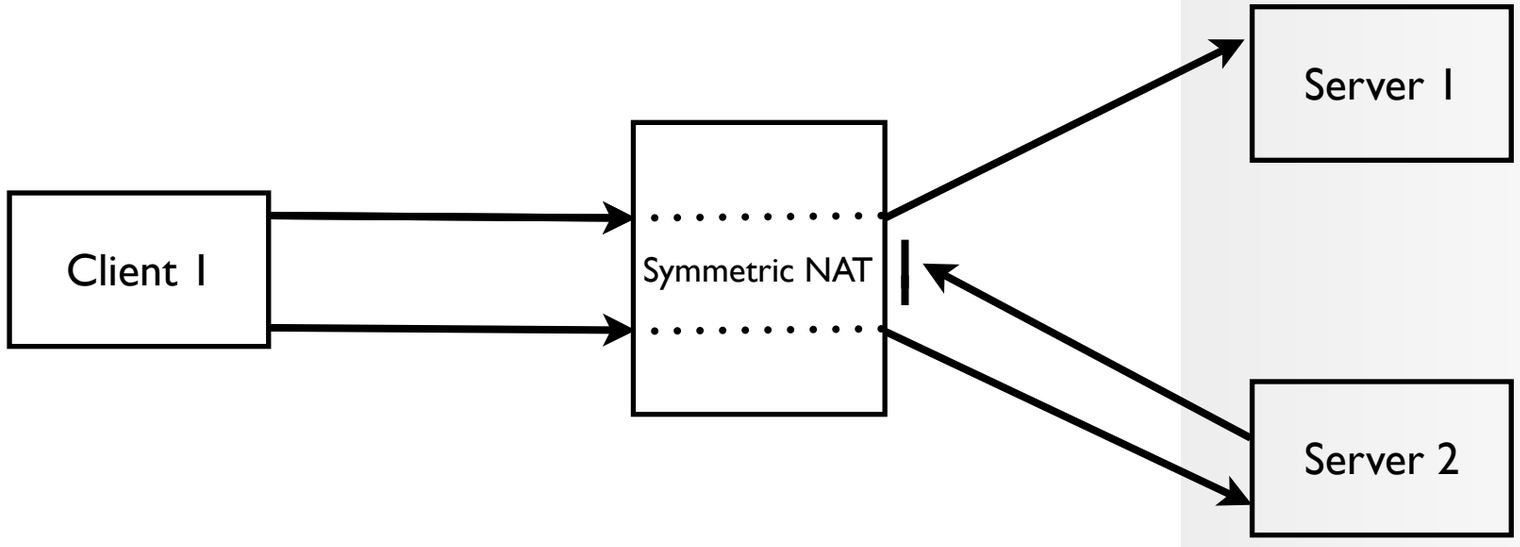
NAT (Network Address Translation) - Kategorien

- Port Restricted Cone NAT
  - externe Hosts können können nur über den selben externen Port mit internen Host Kontakt aufnehmen wenn diesem Verbindungsversuch eine Kontaktaufnahme des internen Hosts vorausging.



NAT (Network Address Translation) - Kategorien

- Symmetric NAT
  - externe Host können nur Pakete an interne Hosts senden (nach initialer Kontaktaufnahme durch den internen Host) wenn der selbe Port und die selben Ziel- und Quelladressen verwendet werden.



AB HIER VOR- UND NACHTEILE

NAT (Network Address Translation)

- Vorteile
  - IP-Adressen eines Netzes können vor einem anderen Netz verborgen werden. Somit kann NAT zur Verbesserung der Netzwerksicherheit eingesetzt werden.
  - NAT kann verwendet werden, um bei nicht dauerhaft verbundenen Netzinstallationen das Problem der knappen IPv4-Adressen zu umgehen.

- Nachteile
  - NAT-Gateways heben die strenge Trennung des OSI-Schichtenmodells auf.
    - Das Umschreiben ein- bzw. ausgehender Pakete führt insbesondere bei älteren Protokollen oder bei Verschlüsselungsverfahren auf Netzwerk- und Transportebene zu Problemen.
  - Ende-zu-Ende-Konnektivität wird gebrochen, da die NAT-Übergabestelle das Ziel eingehender Verbindungen nicht automatisch ermitteln kann.

ARP (Address Resolution Protocol)

- ARP setzt IP-Adressen in Hardware bzw. MAC-Adressen um.
- Ablauf:
  - Um an die Hardware-Adresse einer anderen Station zu kommen verschickt ARP z. B. einen Ethernet-Frame als Broadcast-Meldung mit der MAC-Adresse "FF FF FF FF FF FF". Diese Meldung wird von jedem Netzwerkinterface entgegengenommen und ausgewertet.
  - Der Ethernet-Frame enthält die IP-Adresse der gesuchten Station.
  - Findet einer der empfangenden Stationen diese IP-Adresse in der eigenen Netzwerkkonfiguration, schickt sie eine ARP-Antwort an den Sender zurück.
  - Die gemeldete MAC-Adresse wird dann im lokalen ARP-Cache des Senders gespeichert. Dieser Cache dient zur schnelleren ARP-Adressauflösung.

ICMP (Internet Control Message Protocol)

- Das Internet Control Message Protocol (ICMP) dient dem Austausch von Informations- und Fehlermeldungen über das Internet-Protokoll in der Version 4 (IPv4)
  - *“Destination network unreachable.”* (type-3 ICMP message) z.B. bei FTP, SFTP, Telnet, ssh oder HTTP(S) Sitzungen wenn ein IP Router keinen Pfad zu den spezifizierten Rechner (IP Adresse) finden konnte.
- ICMP ist Bestandteil von IPv4, wird aber wie ein eigenständiges Protokoll behandelt
  - ICMP liegt oberhalb von IP und ICMP-Nachrichten werden in IP-Paketen gekapselt.

ICMP (Internet Control Message Protocol)

- Die meisten ICMP-Pakete enthalten Diagnose-Informationen, sie werden vom Router zur Quelle zurückgeschickt.
- ICMP-Nachrichten werden nicht als Antwort auf Pakete an Zieladressen versendet, bei denen es sich um Multicast- oder Broadcast-Adressen handelt.
- ICMP Nachrichten enthalten ein *“Type”* und ein *”Code”* Feld sowie den Header und die ersten 8 Bytes des IP Paketes, welches für die Generierung der ICMP Nachricht verantwortlich war.

ICMP (Internet Control Message Protocol) - Nachrichten Typen

ICPM Type	Code	Description
0	0	echo reply (to ping)
3	0	destination network unreachable
3	1	destination host unreachable
3	2	destination protocol unreachable
3	3	destination port unreachable
3	4	Fragmentation required, and DF flag set
3	6	destination network unknown
3	7	destination host unknown
4	0	source quench (congestion control)
8	0	echo request
9	0	router advertisement
10	0	router discovery
11	0	TTL expired
12	0	IP header bad

Liste ist nicht vollständig! Eine vollständige Liste kann z.b. unter: https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol eingesehen werden.

ICMP - Ping

- *“Ping”* kann verwendet werden um die prinzipielle Erreichbarkeit einer IP Gegenstelle zu überprüfen.
  - *“Ping”* sendet eine ICMP Nachricht - Type 8, Code 0 an einen spezifizierten Rechner (Ziel Rechner - IP Adresse)
  - Der empfangende Rechner sendet eine ICMP Nachricht - Type 0, Code 0 zurück an den Absender.

ICMP - Traceroute

- *“Traceroute”* kann verwendet werden die Route von eine Quelle (eigener Rechner) zum Ziel aufzuzeigen.
  - *“Traceroute”* versendet hierzu IP Datagramme mit UDP Segmenten, welche eine unübliche Port-Adresse verwenden.
  - Die Datagamme erhalten dabei eine TTL, die der Sendereihenfolge der Datagramme entspricht (das erste Datagramm, eine TTL von eins, das zweite Datagramm eine TTL von zwei usw.).
  - Wenn das n’te Datagramm den n’ten Router erreicht sendet dieser eine ICMP Nachricht (type 11, code 0 - TTL expired) zurück zur Quelle. Diese ICMP Warnung enthält den Namen des Routers sowie dessen IP Adresse.

ICMP - Traceroute

- “Traceroute” wird aufhören IP Datagramme zu erzeugen wenn es eine ICMP Nachricht (type 3, code3 - destination port unreachable). Diese ICMP Nachricht wird von dem spezifizierten Ziel erzeugt, da das UDP Segment eine nicht existierende Port-Adresse enthält.

- Die Standardimplementierung des *“Traceroute”* Programms sendet drei zusätzliche Reihen con Datagrammen.

- “Traceroute” existiert in unterschiedlichen Implementierungen zwischen Unix/Linux und Windows. Die Windows Variante versendet statt UDP Datagramme ICMP Pakete.

Pfad MTU - PMTU (Path Maximum Transmission Unit)

- Um in IPv4-Netzen die maximale Größe zu bestimmen, die ein Datenpaket haben sollte, muss die Stelle des Pfades gefunden werden, die die kleinsten Datenpakete zulässt.

- Dazu wird ein IPv4-Paket versendet, bei dem das DF-Bit (Don't Fragment) gesetzt ist und das die Größe der lokal eingestellten Maximum Transmission Unit hat.

- Kommt das Paket an eine Stelle im Netz, an dem nur eine kleinere MTU verarbeitet werden kann, wird ein ICMP-Error type 3 code 4 (Destination Unreachable Fragmentation Needed, DF Set) zurückgeschickt, der auch die eigene MTU enthält.

- Der lokale Rechner erhält dieses ICMP-Paket, versendet nun die Nachrichten in der Größe der zurückgelieferten MTU und vermeidet somit die Fragmentierung des Paketes ab dem Flaschenhals über den gesamten Rest des Pfades.

OSPF (Open Shortest Path First) - Routing im Internet

- OSPF ist ein Interior Gateway Protokoll
  - Interior Gateway Protokoll: Diese Protokolle werden innerhalb von autonomen System eingesetzt. Der Betreiber kann dabei frei entscheiden, wie sein Routing intern aufgebaut sein soll.
- OSPF ist ein Link-State-Protokoll, basierend auf dem Dijkstra Algorithmus.
  - Bei OSPF konvergiert das Netz bei Änderungen schnell, da nur bei Änderungen inkrementelle Updates versendet werden.
- OSPF ist ein offener Standard, dies ermöglicht jedem das Protokoll zu implementieren ohne auf Patente oder ähnliches achten zu müssen.
- OSPF unterstützt hierarchische Strukturen.

BGP-4 (Border Gateway Protocol) - Routing im Internet

- BGP Version 4 (RFC1771)
- BGP ist ein Exterior Gateway Protokoll
  - Exterior Gateway Protokolle werden verwendet um unterschiedliche autonome Systeme miteinander zu verbinden.
  - Zwischen autonomen Systemen spielen andere Bedingungen eine Rolle als innerhalb von autonomen Systemen. Dies können politische, kommerzielle oder technische Gründe sein.
- BGP ist ein Distance-Vector-Protocol
  - Jeder Router speichert zusätzlich den verwendeten Pfad (BGP-Path). Daher bezeichnet man BGP als Path-Vector-Protocol.

BGP (Border Gateway Protocol) - Routing im Internet

- BGP führt ein Policy-Routing Konzept ein, dessen Ziel es ist das Standardverhalten des Routings zu verändern um z.B. politische oder kommerzielle Rahmenbedingungen zu erfüllen:
  - Kosten sparen
    - Bestimmte Peerings sollen bevorzugt werden, weil diese günstiger sind.
  - Lastverteilung
    - Da Peerings unterschiedliche Bandbreiten bieten kann es notwendig werden den anfallenden Traffic auf verschiedene Peerings zu verteilen.
  - Bedingtes Routing
    - Gerade im kommerziellen Umfeld kann es notwendig werden Netzteilnehmer nur über bestimmte Routen zu verbinden, da sie evt. nur bestimmte Peerings gebucht haben.