Gruppen Di-T14 / Mi-T25

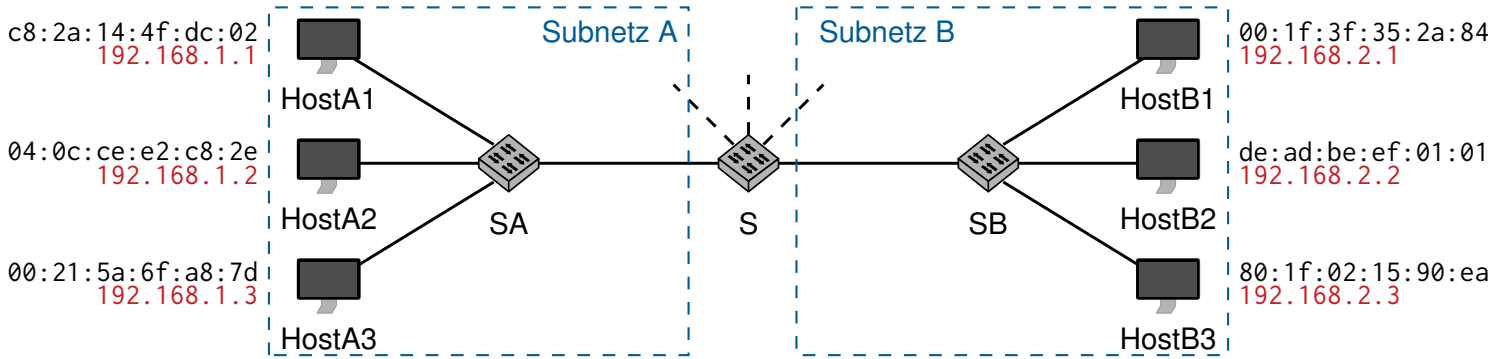
Tutorübung zu
Grundlagen: Rechnernetze und Verteilte Systeme
(SS 16)

Michael Schwarz

Institut für Informatik
Technische Universität München

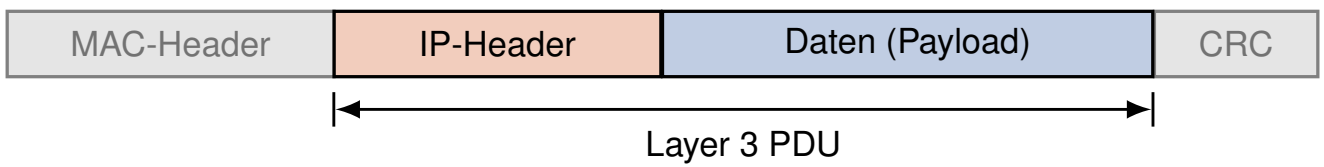
07.06 / 08.06.2016

Wir betrachten das Beispielnetz mit einem Router (R) in der Mitte:



- Jedem Host ist eine IP-Adresse zugewiesen. Jede IP-Adresse ist in vier Gruppen zu je einem Byte, durch Punkte getrennt, dargestellt (Dotted Decimal Notation).
- In diesem Beispiel identifiziert das 4. Oktett einen Host innerhalb eines Netzes.
- Die ersten drei Oktette identifizieren das Netzwerk, in dem sich der Host befindet.
- Der Router R trifft Weiterleitungsentscheidungen auf Basis der Ziel-IP-Adresse.

⇒ Jedes Paket muss mit einer Absender- und Ziel-IP-Adresse (im IP-Header) versehen werden:



IP-Header

- Der IP-Header beinhaltet nicht nur Quell- und Ziel-Adresse.
- Die Verwendung der wichtigsten Felder wird später in diesem Kapitel anhand von Beispielen genauer erläutert werden.

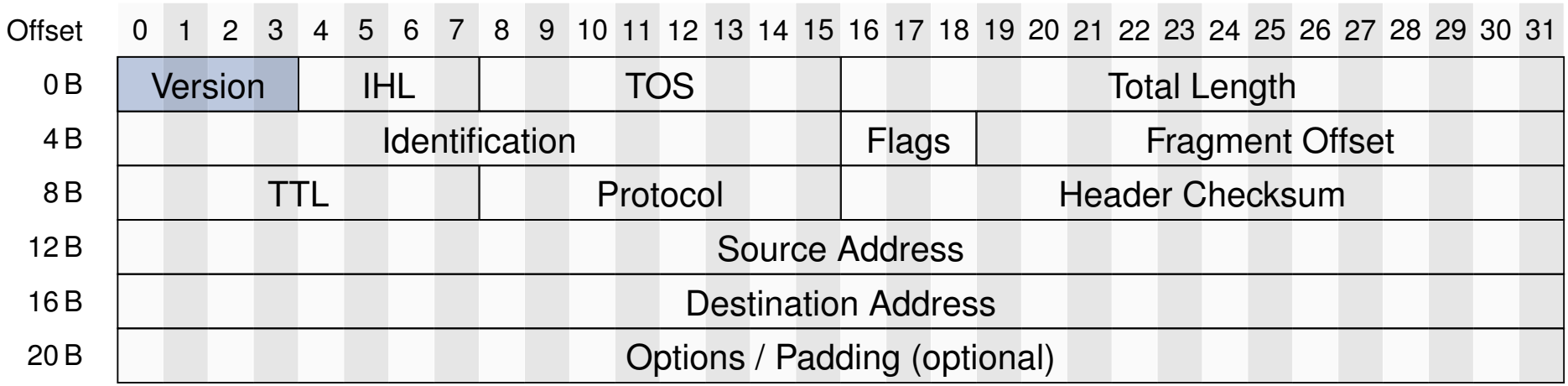


Abbildung: IPv4-Header (minimale Länge: 20 B)

Version

- Gibt die verwendete IP-Version an.
- Gültige Werte sind 4 (IPv4) und 6 (IPv6).

IP-Header

- Der IP-Header beinhaltet nicht nur Quell- und Ziel-Adresse.
- Die Verwendung der wichtigsten Felder wird später in diesem Kapitel anhand von Beispielen genauer erläutert werden.

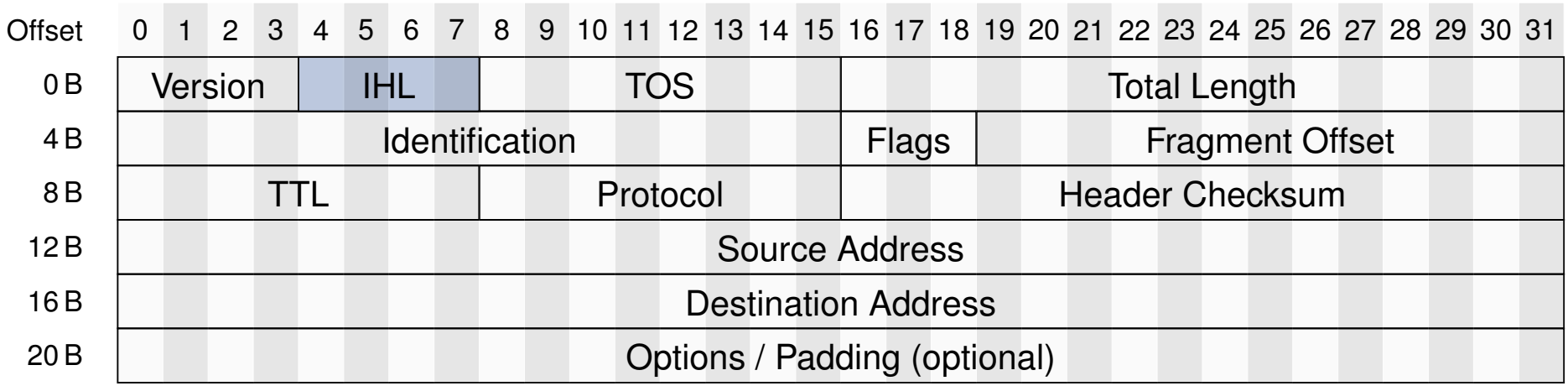


Abbildung: IPv4-Header (minimale Länge: 20 B)

IHL (IP Header Length)

- Gibt die Länge des IP Headers inkl. Optionen in Vielfachen von 32 bit an.
- Wichtig, da der IPv4-Header durch Optionsfelder variable Länge hat.

IP-Header

- Der IP-Header beinhaltet nicht nur Quell- und Ziel-Adresse.
- Die Verwendung der wichtigsten Felder wird später in diesem Kapitel anhand von Beispielen genauer erläutert werden.

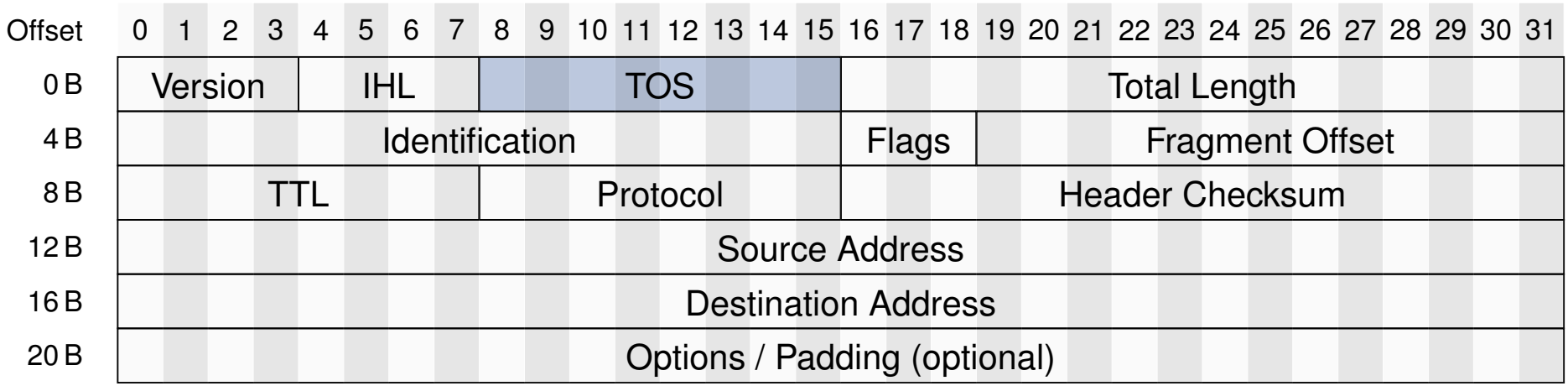


Abbildung: IPv4-Header (minimale Länge: 20 B)

TOS (Type of Service)

- Dient der Klassifizierung und Priorisierung von IP-Paketen (z. B. Hinweis auf zeitsensitive Daten wie Sprachübertragungen).
- Möglichkeit zur Staukontrolle (Explicit Congestion Notification) auf Schicht 3 (optional).

IP-Header

- Der IP-Header beinhaltet nicht nur Quell- und Ziel-Adresse.
- Die Verwendung der wichtigsten Felder wird später in diesem Kapitel anhand von Beispielen genauer erläutert werden.

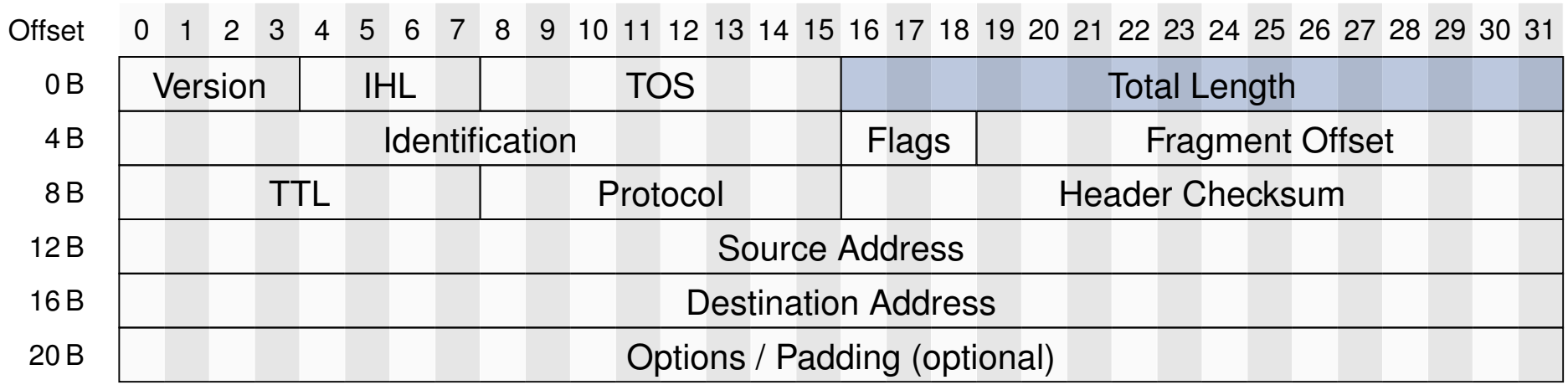


Abbildung: IPv4-Header (minimale Länge: 20 B)

Total Length

- Gibt die Gesamtlänge des IP-Pakets (Header + Daten) in Bytes an.
- Die Maximallänge eines IP-Pakets beträgt damit 65 535 B.
- Der Sender passt die Größe ggf. an, um Fragmentierung zu vermeiden.
- Die maximale Paketlänge, so dass keine Fragmentierung notwendig ist, bezeichnet man als Maximum Transmission Unit (MTU). Diese ist abhängig von Schicht 2/1 und beträgt bei FastEthernet 1500 B.

IP-Header

- Der IP-Header beinhaltet nicht nur Quell- und Ziel-Adresse.
- Die Verwendung der wichtigsten Felder wird später in diesem Kapitel anhand von Beispielen genauer erläutert werden.

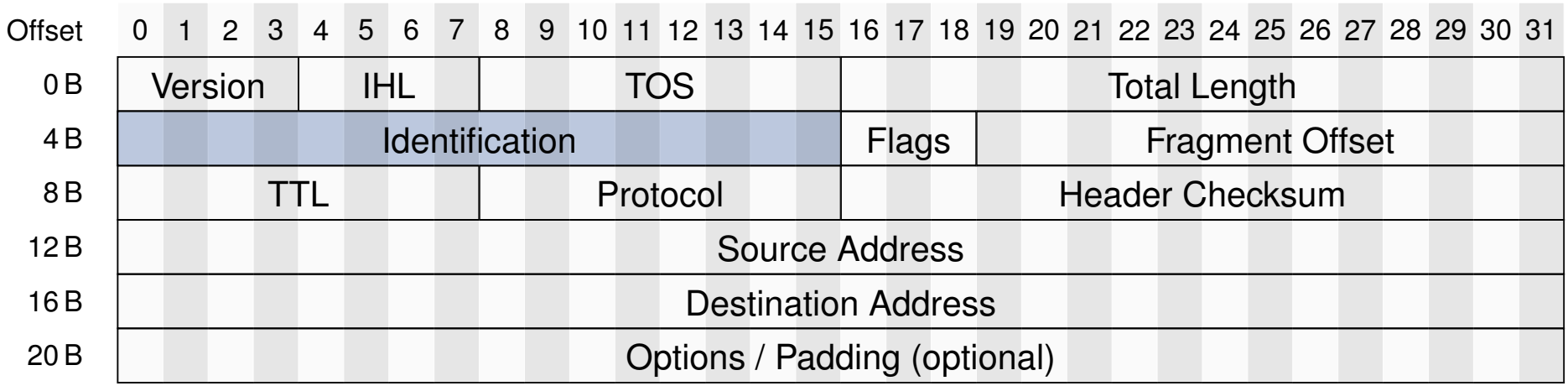


Abbildung: IPv4-Header (minimale Länge: 20 B)

Identification

- Für jedes IP-Paket (zufällig) gewählter 16 bit langer Wert.
- Dient der Identifikation zusammengehörender Fragmente (IP-Fragmentierung → später).

IP-Header

- Der IP-Header beinhaltet nicht nur Quell- und Ziel-Adresse.
- Die Verwendung der wichtigsten Felder wird später in diesem Kapitel anhand von Beispielen genauer erläutert werden.

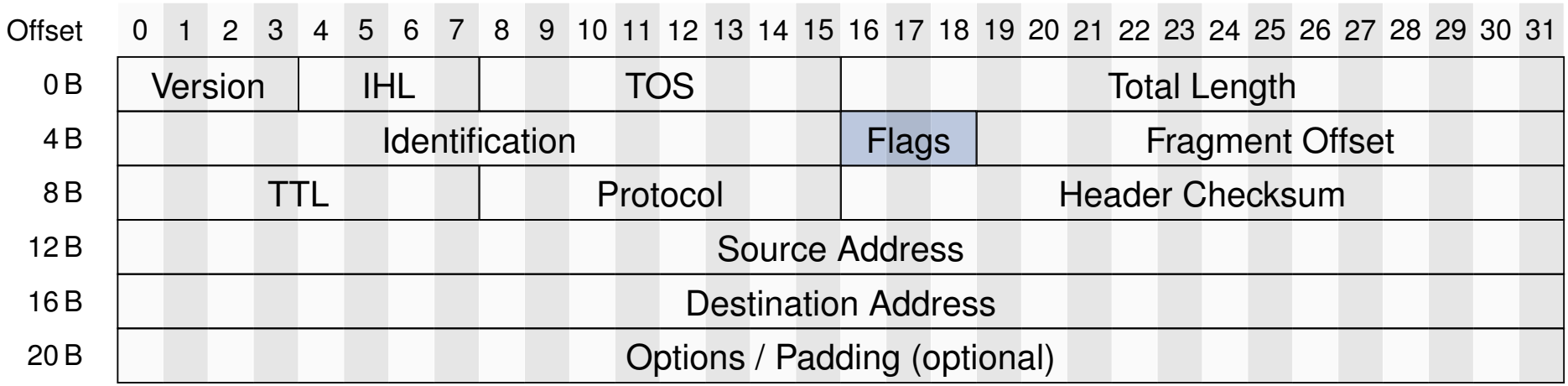


Abbildung: IPv4-Header (minimale Länge: 20 B)

Flags

- Bit 16: Reserviert und wird auf 0 gesetzt.
- Bit 17: Don't Fragment (DF). Ist dieses Bit 1, so darf das IP-Paket nicht fragmentiert werden.
- Bit 18: More Fragments (MF). Gibt an, ob weitere Fragmente folgen (1) oder dieses Paket das letzte Fragment ist (0). Wurde das Paket nicht fragmentiert, wird es ebenfalls auf 0 gesetzt.

IP-Header

- Der IP-Header beinhaltet nicht nur Quell- und Ziel-Adresse.
- Die Verwendung der wichtigsten Felder wird später in diesem Kapitel anhand von Beispielen genauer erläutert werden.

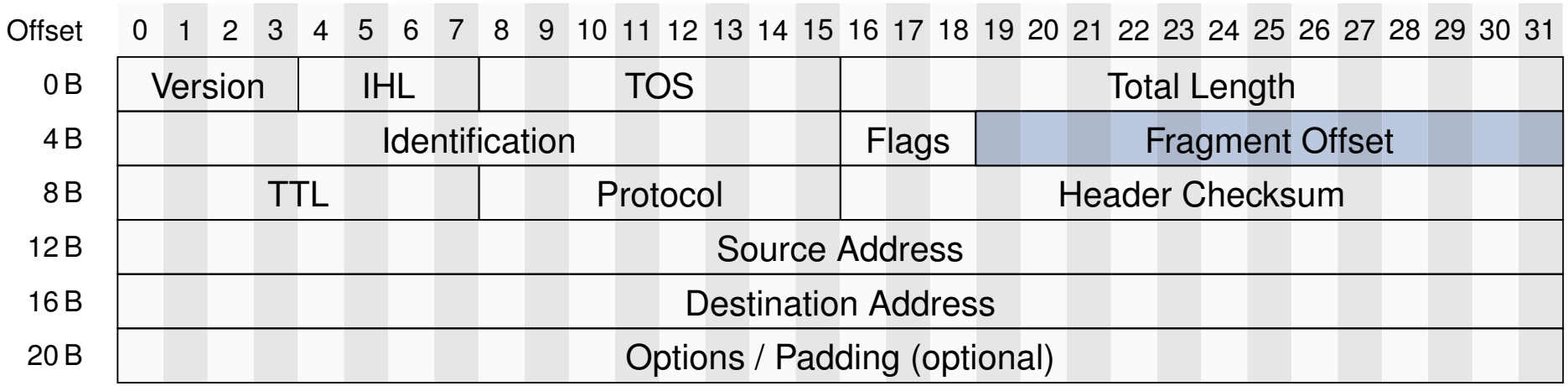


Abbildung: IPv4-Header (minimale Länge: 20 B)

Fragment Offset

- Gibt die absolute Position der Daten in diesem Fragment bezogen auf das unfragmentierte Paket in ganzzahligen Vielfachen von 8 B an.
- Ermöglicht zusammen mit dem Identifier und MF-Bit die Reassemblierung fragmentierter Pakete in der richtigen Reihenfolge.

IP-Header

- Der IP-Header beinhaltet nicht nur Quell- und Ziel-Adresse.
- Die Verwendung der wichtigsten Felder wird später in diesem Kapitel anhand von Beispielen genauer erläutert werden.

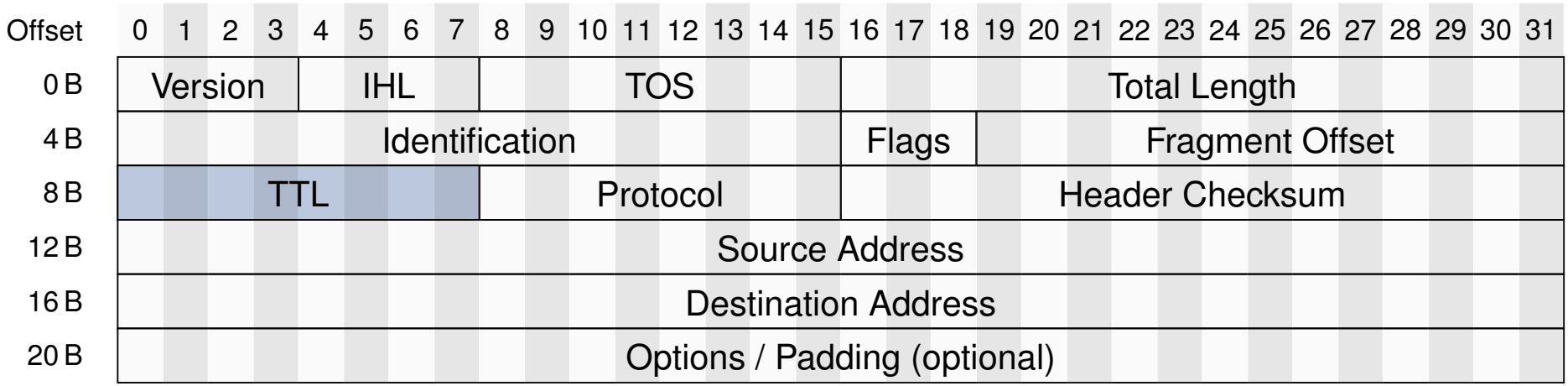


Abbildung: IPv4-Header (minimale Länge: 20 B)

TTL (Time to Live)

- Leitet ein Router ein IP-Paket weiter, so dekrementiert er das TTL-Feld um 1.
- Erreicht das TTL-Feld den Wert 0, so verwirft ein Router das Paket und sendet eine Benachrichtigung an den Absender (ICMP Time Exceeded $\rightarrow$ später).
- Dieser Mechanismus beschränkt die Pfadlänge im Internet und verhindert endlos kreisende Pakete infolge von Routing Loops.

IP-Header

- Der IP-Header beinhaltet nicht nur Quell- und Ziel-Adresse.
- Die Verwendung der wichtigsten Felder wird später in diesem Kapitel anhand von Beispielen genauer erläutert werden.

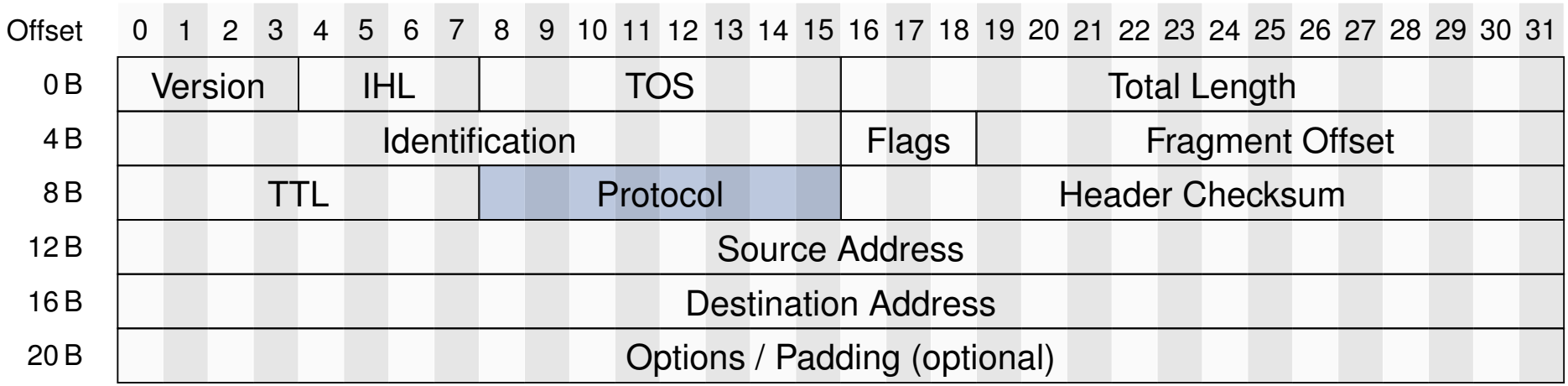


Abbildung: IPv4-Header (minimale Länge: 20 B)

Protocol

- Identifiziert das Protokoll auf Schicht 4, welches in der Payload (Daten) des IP-Pakets enthalten ist.
- Relevant u. a. für das Betriebssystem, um Pakete dem richtigen Prozess zuordnen zu können.
- Gültige Werte sind beispielsweise `0x06` (TCP) und `0x11` (UDP).

IP-Header

- Der IP-Header beinhaltet nicht nur Quell- und Ziel-Adresse.
- Die Verwendung der wichtigsten Felder wird später in diesem Kapitel anhand von Beispielen genauer erläutert werden.

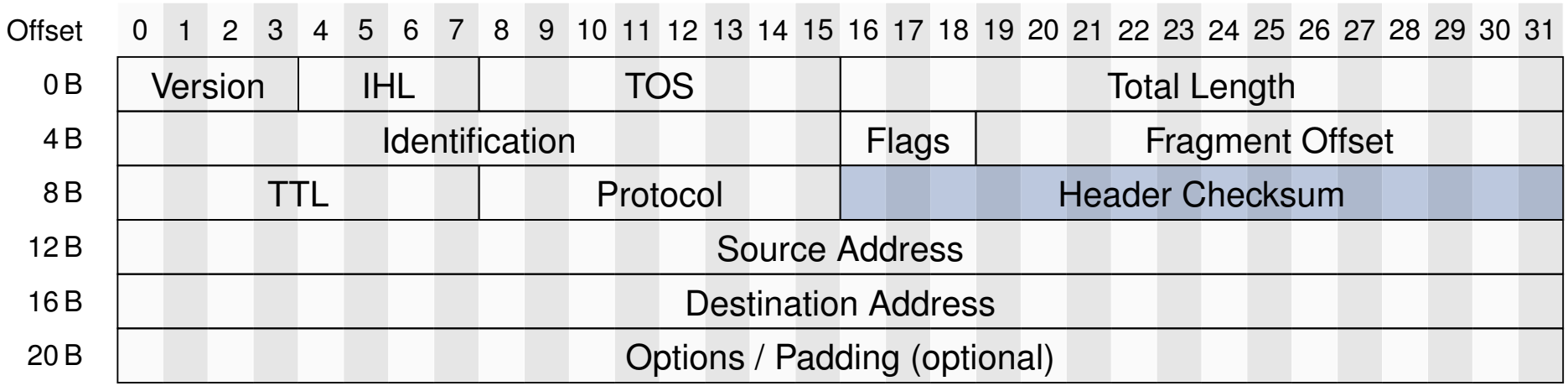


Abbildung: IPv4-Header (minimale Länge: 20 B)

Header Checksum

- Einfache, auf Geschwindigkeit optimierte Prüfsumme, welche nur den IP-Header (ohne Daten) schützt.
- Die Checksumme ist so ausgelegt, dass die Dekrementierung des TTL-Felds einer Inkrementierung der Checksumme entspricht. Es ist also keine Neuberechnung der Checksumme bei der Weiterleitung von Paketen notwendig.
- Es ist lediglich Fehlererkennung aber keine Korrektur möglich.

IP-Header

- Der IP-Header beinhaltet nicht nur Quell- und Ziel-Adresse.
- Die Verwendung der wichtigsten Felder wird später in diesem Kapitel anhand von Beispielen genauer erläutert werden.

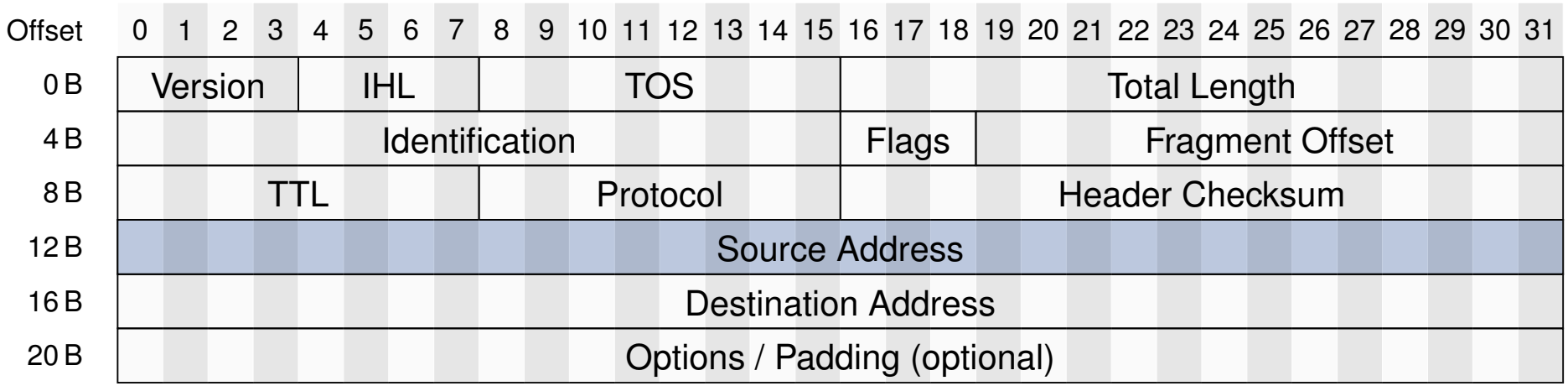


Abbildung: IPv4-Header (minimale Länge: 20 B)

Source Address

- IP-Adresse des Absenders.

IP-Header

- Der IP-Header beinhaltet nicht nur Quell- und Ziel-Adresse.
- Die Verwendung der wichtigsten Felder wird später in diesem Kapitel anhand von Beispielen genauer erläutert werden.

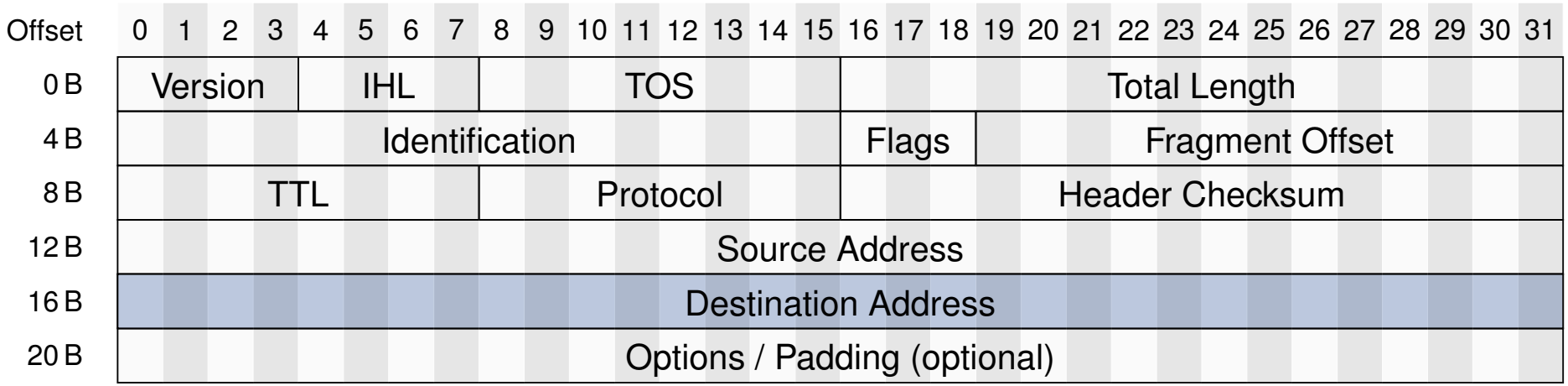


Abbildung: IPv4-Header (minimale Länge: 20 B)

Destination Address

- IP-Adresse des Empfängers.

IP-Header

- Der IP-Header beinhaltet nicht nur Quell- und Ziel-Adresse.
- Die Verwendung der wichtigsten Felder wird später in diesem Kapitel anhand von Beispielen genauer erläutert werden.

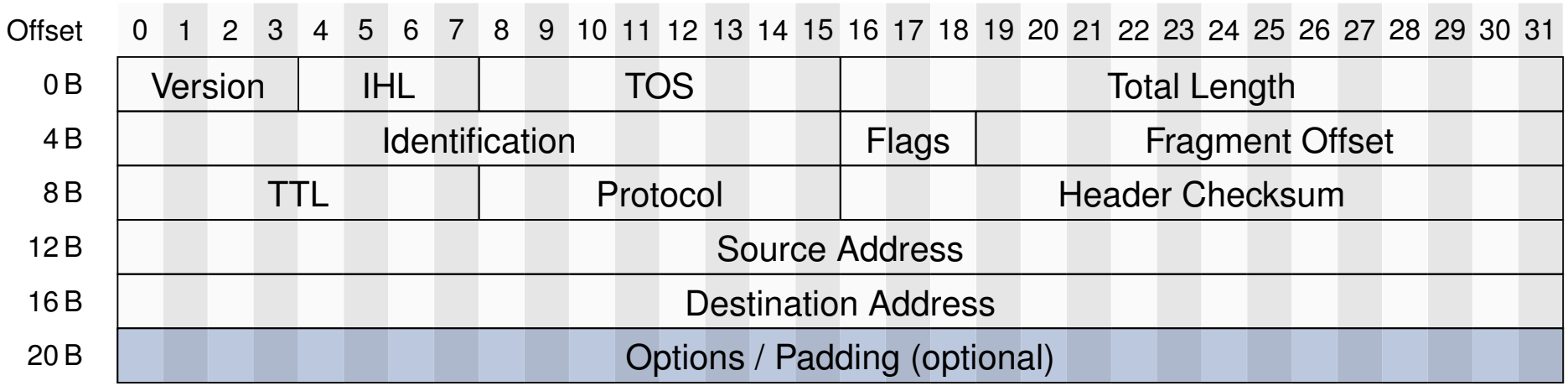


Abbildung: IPv4-Header (minimale Länge: 20 B)

Options / Padding

- IP unterstützt eine Reihe von Optionen (z. B. Zeitstempel, Route Recording, . . .), welche als optionale Felder an den IP-Header angefügt werden können.
- Nicht alle diese Optionen sind 4 B lang. Da die Länge des IP-Headers jedoch ein Vielfaches von 4 B betragen muss, werden kürzere Optionen ggf. durch Padding auf den nächsten gültigen Wert ergänzt.

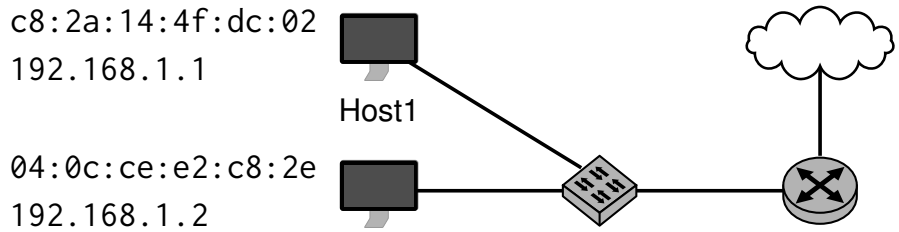
Adressauflösung

Situation

- Adressierung auf Layer 3 (IP) erfolgt mithilfe der IP-Adresse
- Darunter brauche ich aber für die Next-Hop-Adressierung die MAC-Adresse

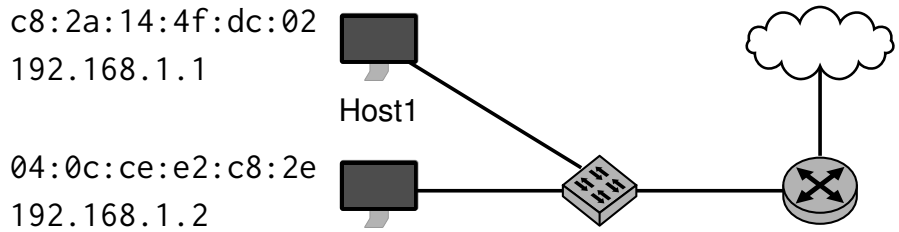
Adressauflösung [14]

- Host1 will eine Nachricht an Host2 senden
- Die IP-Adresse von Host2 (192.168.1.2) sei ihm bereits bekannt
- Wie erhält Host1 die zugehörige MAC-Adresse?



Adressauflösung [14]

- Host1 will eine Nachricht an Host2 senden
- Die IP-Adresse von Host2 (192.168.1.2) sei ihm bereits bekannt
- Wie erhält Host1 die zugehörige MAC-Adresse?



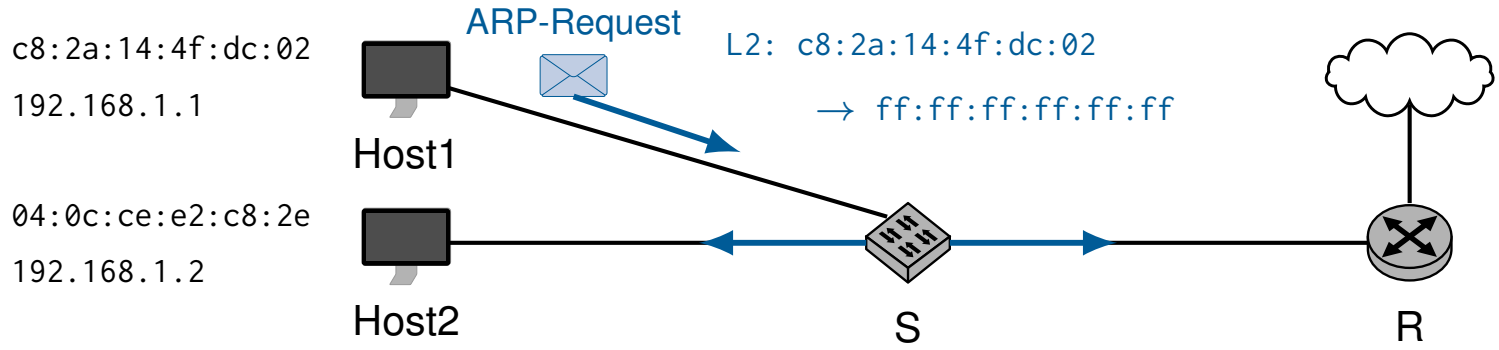
Address Resolution Protocol (ARP)

1. Host1 sendet einen ARP Request: „Who has 192.168.1.2? Tell 192.168.1.1 at `c8:2a:14:4f:dc:02`“
2. Host2 antwortet mit einem ARP Reply: „192.168.1.2 is at `04:0c:ce:e2:c8:2e`“

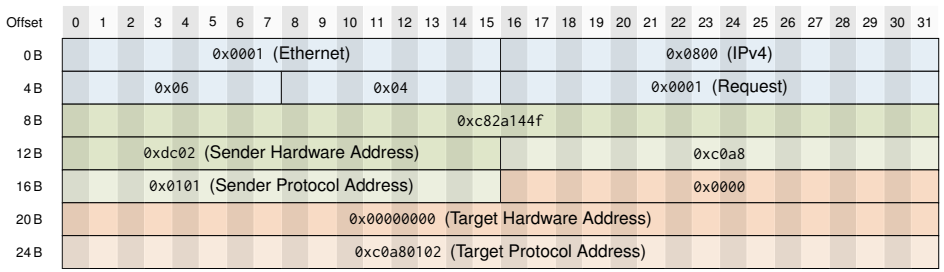
Offset	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0 B	Hardware Type																Protocol Type															
4 B	Hardware Addr. Length								Protocol Addr. Length								Operation															
8 B	Sender Hardware Address (first 32 bit)																															
12 B	Sender Hardware Address (last 16 bit)																Sender Protocol Address (first 16 bit)															
16 B	Sender Protocol Address (last 16 bit)																Target Hardware Address (first 16 bit)															
20 B	Target Hardware Address (last 32 bit)																															
24 B	Target Protocol Address																															

Abbildung: ARP-Paket für IPv4 über Ethernet

Beispiel:



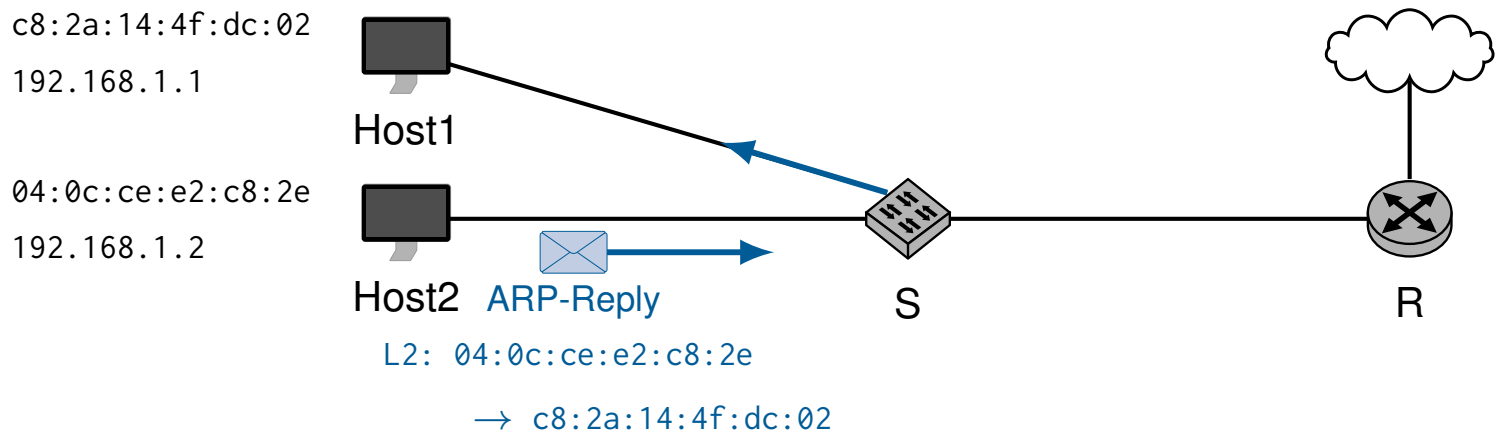
Hinweis: `L2: xx:xx:xx:xx:xx:xx → yy:yy:yy:yy:yy:yy` stellt Absender- und Ziel-MAC-Adresse auf Schicht 2 dar.



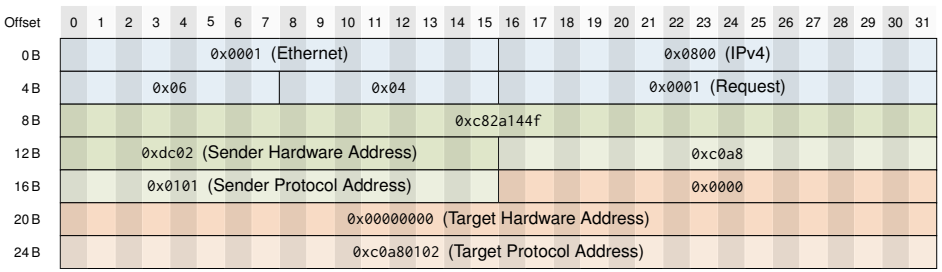
(a) ARP Request

- Der ARP-Request wird an die MAC-Broadcast-Adresse `ff:ff:ff:ff:ff:ff` geschickt, weswegen der Switch S den Rahmen an alle angeschlossenen Hosts weiterleitet.

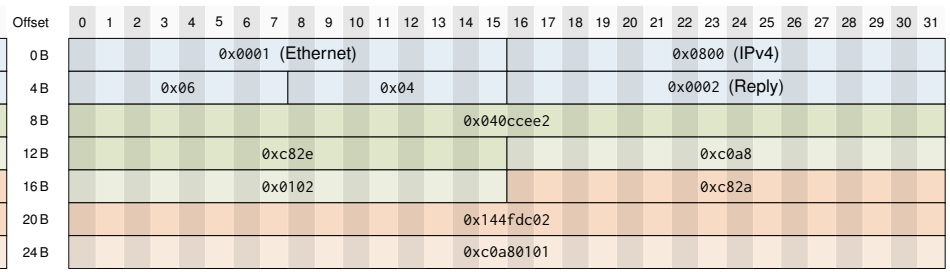
Beispiel:



Hinweis: `L2: xx:xx:xx:xx:xx:xx → yy:yy:yy:yy:yy:yy` stellt Absender- und Ziel-MAC-Adresse auf Schicht 2 dar.



(c) ARP Request

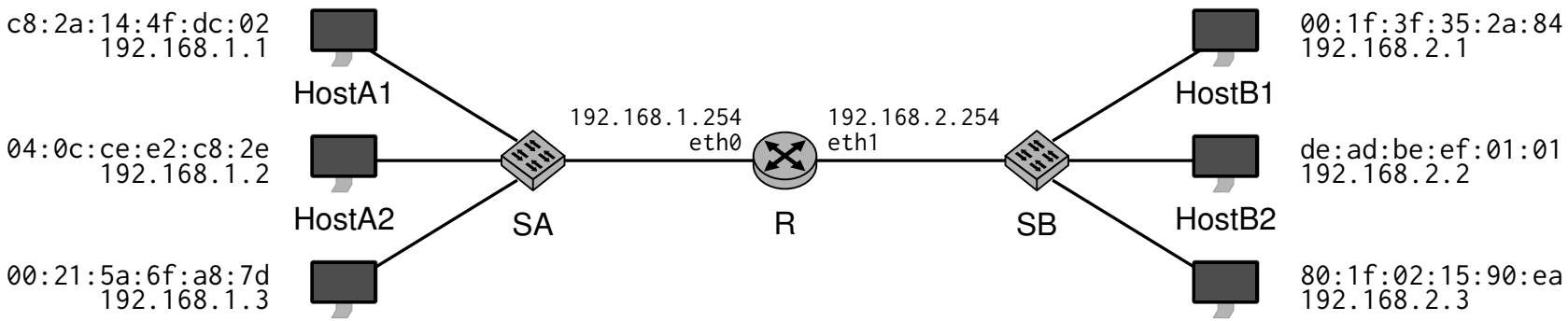


(d) ARP Reply

- Der ARP-Request wird an die MAC-Broadcast-Adresse `ff:ff:ff:ff:ff:ff` geschickt, weswegen der Switch S den Rahmen an alle angeschlossenen Hosts weiterleitet.
- Der ARP-Reply wird als MAC-Unicast versendet (adressiert an Host1).
- Die Rollen Sender / Target sind zwischen Request und Reply vertauscht (vgl. Inhalte der grünen und roten Felder).

Was ist nun, wenn das Ziel nicht im selben Netz liegt (z. B. HostA1 an HostB2)?

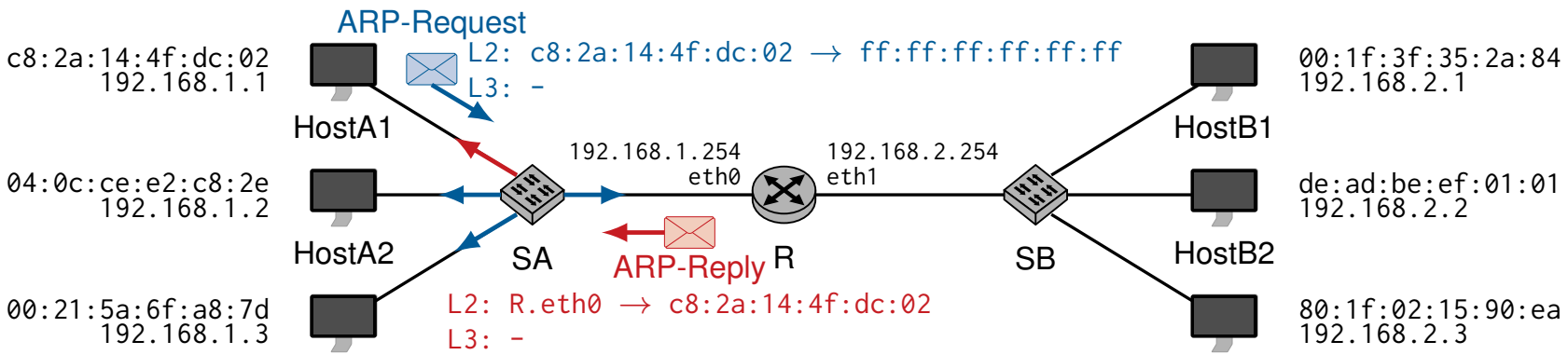
- Jeder Host sollte einen Router zum Internet, das sog. Default Gateway, kennen, an das er alle Pakete schickt, deren Zieladressen nicht im eigenen Netz liegen, und für die in seiner Routing-Tabelle nicht ein spezifisches Gateway eingetragen ist.
- Ob eine Zieladresse zum eigenen Netz gehört erkennt ein Host durch Vergleich der Zieladdresse mit der eigenen Netzadresse.
- Im Moment gehen wir noch davon aus, dass die ersten 3 Oktette einer IP-Adresse das Netz identifizieren
  $\Rightarrow$ 192.168.1.1 und 192.168.2.2 liegen in unterschiedlichen Netzen.



1. HostA1 erkennt, dass 192.168.2.2 nicht im eigenen Netz liegt. Sein Default-Gateway ist 192.168.1.254.

Was ist nun, wenn das Ziel nicht im selben Netz liegt (z. B. HostA1 an HostB2)?

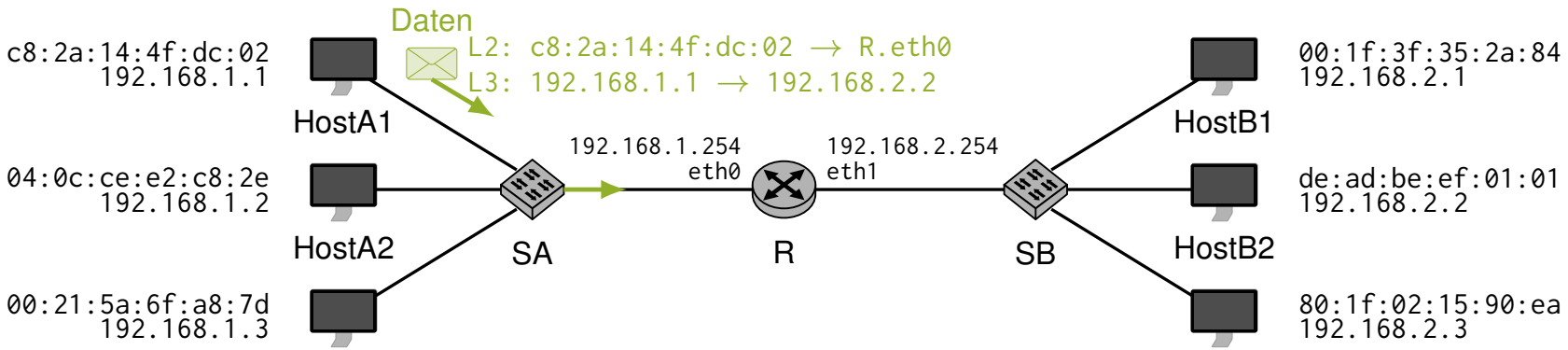
- Jeder Host sollte einen Router zum Internet, das sog. Default Gateway, kennen, an das er alle Pakete schickt, deren Zieladressen nicht im eigenen Netz liegen, und für die in seiner Routing-Tabelle nicht ein spezifisches Gateway eingetragen ist.
- Ob eine Zieladresse zum eigenen Netz gehört erkennt ein Host durch Vergleich der Zieladdresse mit der eigenen Netzadresse.
- Im Moment gehen wir noch davon aus, dass die ersten 3 Oktette einer IP-Adresse das Netz identifizieren $\Rightarrow$ 192.168.1.1 und 192.168.2.2 liegen in unterschiedlichen Netzen.



1. HostA1 erkennt, dass 192.168.2.2 nicht im eigenen Netz liegt. Sein Default-Gateway ist 192.168.1.254.
2. HostA1 löst die MAC-Adresse zu 192.168.1.254 auf.

Was ist nun, wenn das Ziel nicht im selben Netz liegt (z. B. HostA1 an HostB2)?

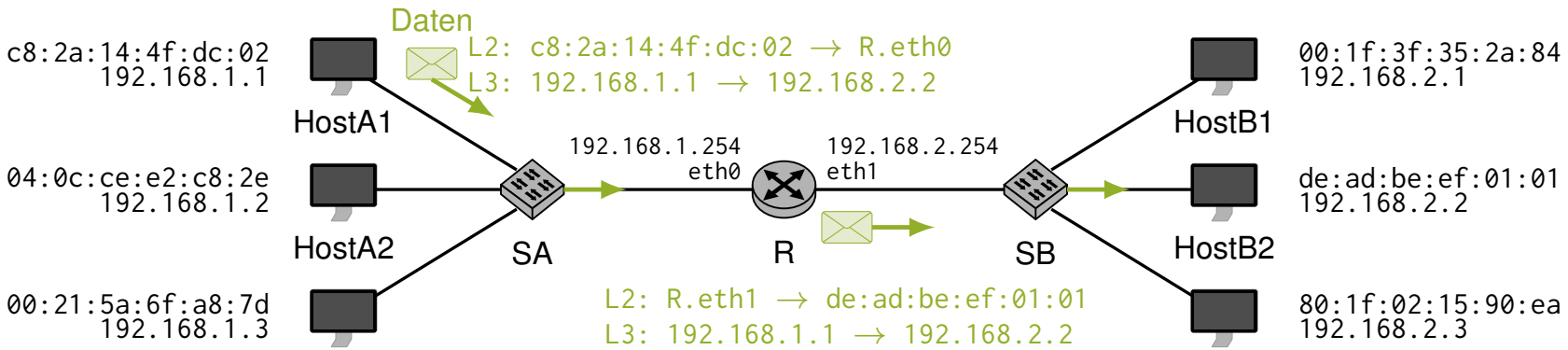
- Jeder Host sollte einen Router zum Internet, das sog. Default Gateway, kennen, an das er alle Pakete schickt, deren Zieladressen nicht im eigenen Netz liegen, und für die in seiner Routing-Tabelle nicht ein spezifisches Gateway eingetragen ist.
- Ob eine Zieladresse zum eigenen Netz gehört erkennt ein Host durch Vergleich der Zieladdresse mit der eigenen Netzadresse.
- Im Moment gehen wir noch davon aus, dass die ersten 3 Oktette einer IP-Adresse das Netz identifizieren $\Rightarrow$ 192.168.1.1 und 192.168.2.2 liegen in unterschiedlichen Netzen.



1. HostA1 erkennt, dass 192.168.2.2 nicht im eigenen Netz liegt. Sein Default-Gateway ist 192.168.1.254.
2. HostA1 löst die MAC-Adresse zu 192.168.1.254 auf.
3. HostA1 sendet das Datenpaket an R: Dabei adressiert er R mittels der eben bestimmten MAC-Adresse (Schicht 2). Als Ziel-IP-Adresse (Schicht 3) verwendet er die IP-Adresse von HostB2.

Was ist nun, wenn das Ziel nicht im selben Netz liegt (z. B. HostA1 an HostB2)?

- Jeder Host sollte einen Router zum Internet, das sog. Default Gateway, kennen, an das er alle Pakete schickt, deren Zieladressen nicht im eigenen Netz liegen, und für die in seiner Routing-Tabelle nicht ein spezifisches Gateway eingetragen ist.
- Ob eine Zieladresse zum eigenen Netz gehört erkennt ein Host durch Vergleich der Zieladdresse mit der eigenen Netzadresse.
- Im Moment gehen wir noch davon aus, dass die ersten 3 Oktette einer IP-Adresse das Netz identifizieren $\Rightarrow$ 192.168.1.1 und 192.168.2.2 liegen in unterschiedlichen Netzen.



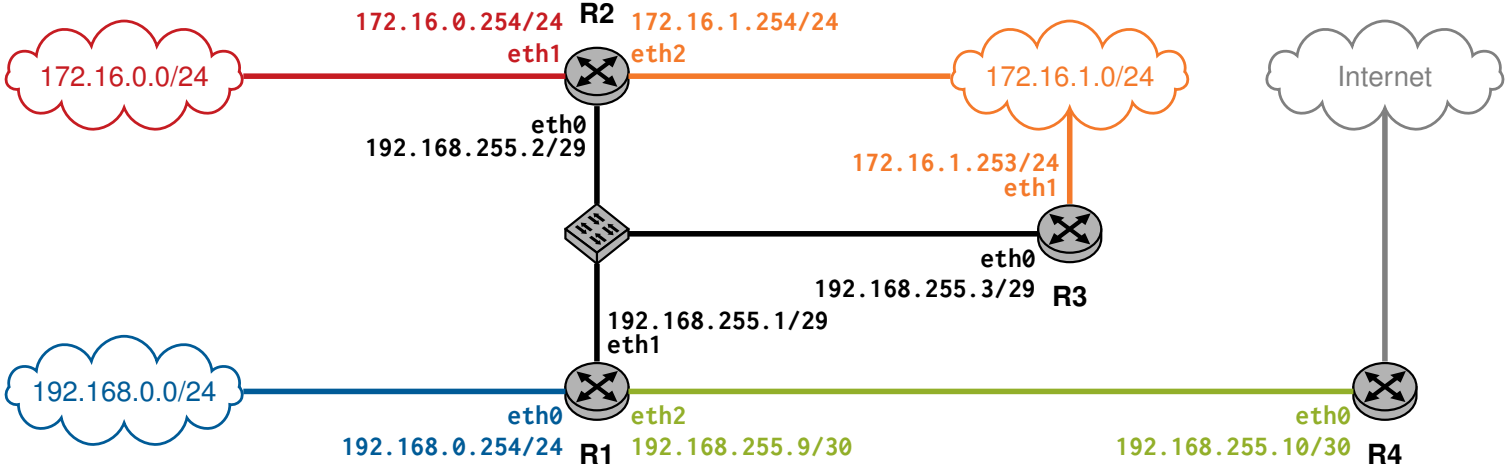
1. HostA1 erkennt, dass 192.168.2.2 nicht im eigenen Netz liegt. Sein Default-Gateway ist 192.168.1.254.
2. HostA1 löst die MAC-Adresse zu 192.168.1.254 auf.
3. HostA1 sendet das Datenpaket an R: Dabei adressiert er R mittels der eben bestimmten MAC-Adresse (Schicht 2). Als Ziel-IP-Adresse (Schicht 3) verwendet er die IP-Adresse von HostB2.
4. R akzeptiert das Paket, bestimmt das ausgehende Interface und leitet das Paket weiter an HostB2. Dabei adressiert R wiederum HostB2 anhand seiner MAC-Adresse (erfordert ggf. einen weiteren ARP-Schritt).

Situation

- Router hat ein Paket bekommen
  $\Rightarrow$ Wohin soll es weiter geschickt werden?

Statisches Routing

Wir betrachten im Folgenden das unten abgebildete Beispielnetzwerk:

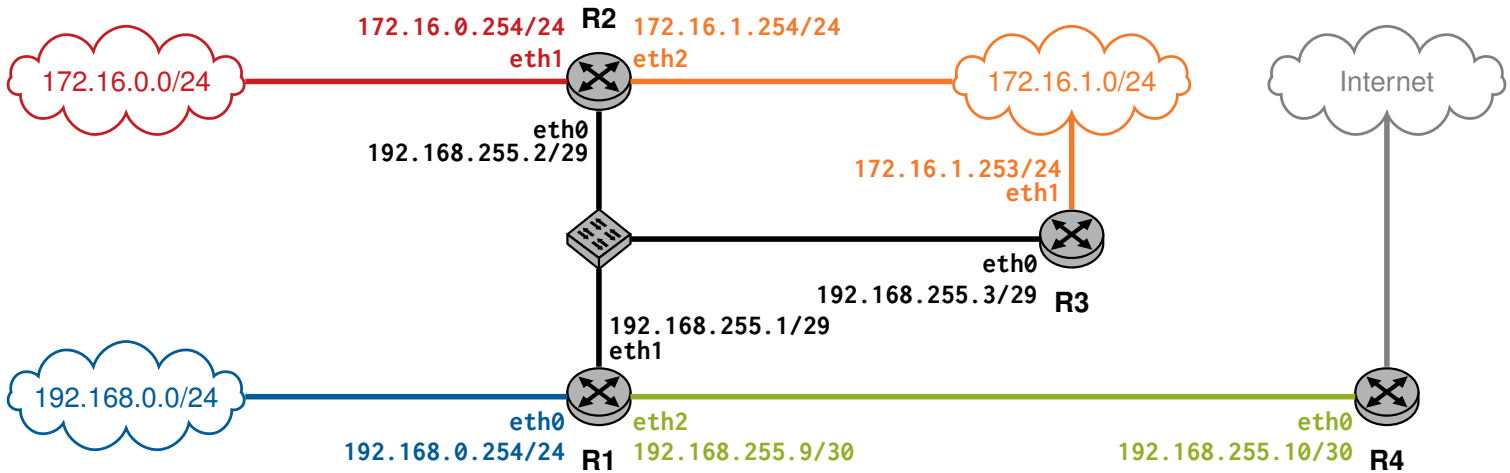


- Die Farben der Links und Interface-Adressen verdeutlichen die einzelnen Subnetze
- Das Netzwerk 192.168.255.0/29 (schwarz) verfügt über 6 nutzbare Hostadressen
- Das Netzwerk 192.168.255.8/30 (grün) ist ein Transportnetz mit nur 2 nutzbaren Hostadressen
- Die übrigen Netze sind /24 Netze mit jeweils 254 nutzbaren Hostadressen

Frage: Wie entscheidet R1, an welchen Next-Hop ein Paket weitergeleitet werden soll?

Routing Table und Longest Prefix Matching

Routing Table



Definition: Routing Table

In der Routing-Tabelle speichert ein Router (oder Host)

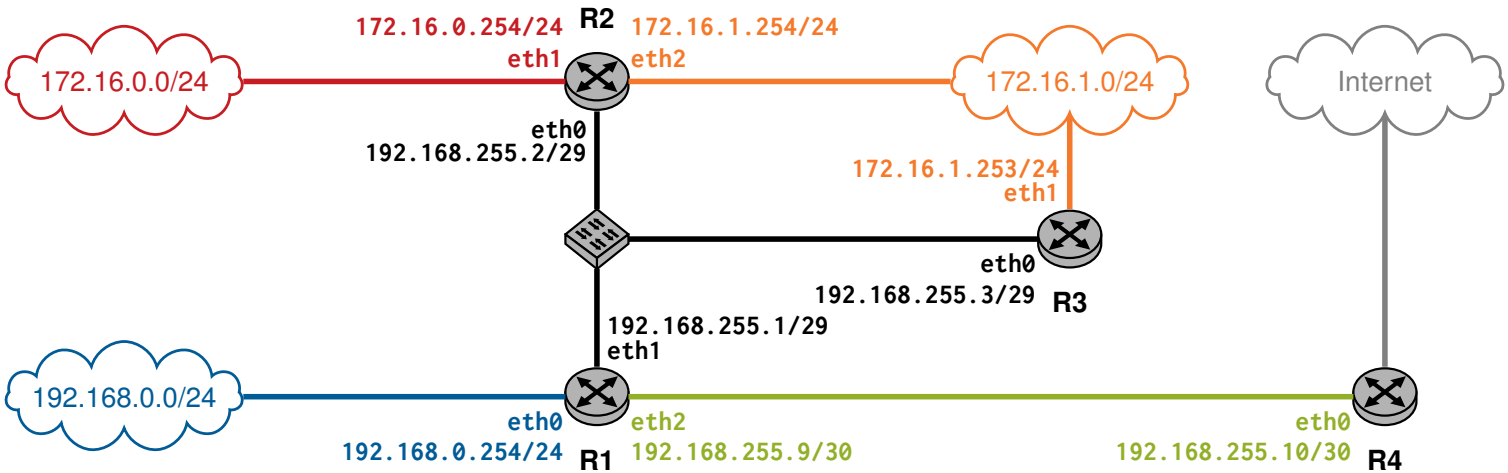
- die Netzadresse eines Ziels,
- die Länge des Präfixes,
- den zugehörigen Next-Hop (auch Gateway genannt),
- das Interface, über welches dieser Next-Hop erreichbar ist, und
- die Kosten bis zum Ziel.

Hinweise:

- Bei IPv4 wird häufig anstatt der Präfixlänge die Subnetzmaske (oder Genmask) angegeben. Bei einem Präfix von N bit handelt es sich bei IPv4 um einen Block von vier Oktetten, wobei genau die erste N bit dieses Blocks 1 und alle übrigen 0 sind.
- Die Kosten werden fälschlicherweise auch als Metrik bezeichnet. Die Metrik hingegen ist, woraus die Kosten berechnet werden (z.B. Hop-Count, Bandbreite, Verzögerung etc.).

Routing Table und Longest Prefix Matching

Routing Table



Beispiel: Routing-Tabelle für R1

Destination	NextHop	Costs	Iface
192.168.255.8/30	0.0.0.0	0	eth2
192.168.255.0/29	0.0.0.0	0	eth1
192.168.0.0/24	0.0.0.0	0	eth0
172.16.1.0/24	192.168.255.3	1	eth1
172.16.0.0/23	192.168.255.2	1	eth1
0.0.0.0/0	192.168.255.10	0	eth2

- Die Netze 172.16.{0.1}.0/24 wurden zusammengefasst
- Die Route 0.0.0.0 wird auch als Default Route bezeichnet
- Interessant: R1 kennt zwei (eigentlich sogar drei) Routen zum Netz 172.16.1.0/24 !

Longest Prefix Matching

1. R1 berechnet das logische AND aus der Zieladresse des Pakets und den Subnetzmasken (welche aus der Präfixlänge hervorgehen) in seiner Routingtabelle.
2. Das Ergebnis wird mit dem Eintrag in der Spalte „ Destination“ verglichen.
3. Stimmt das Ergebnis damit überein, werden Gateway und zugehöriges Interface bestimmt.
4. Nachdem die MAC-Adresse des Gateways ggf. via ARP aufgelöst wurde, wird das Paket mit einem neuen Ethernet-Header versehen und weitergeleitet.

Beispiel: R1 erhalte ein Paket mit der Zieladresse 172.16.1.23.

Destination	NextHop	Costs	Iface
192.168.255.8/30	0.0.0.0	0	eth2
192.168.255.0/29	0.0.0.0	0	eth1
192.168.0.0/24	0.0.0.0	0	eth0
172.16.1.0/24	192.168.255.3	1	eth1
172.16.0.0/23	192.168.255.2	1	eth1
0.0.0.0/0	192.168.255.10	0	eth2

Longest Prefix Matching

1. R1 berechnet das logische AND aus der Zieladresse des Pakets und den Subnetzmasken (welche aus der Präfixlänge hervorgehen) in seiner Routingtabelle.
2. Das Ergebnis wird mit dem Eintrag in der Spalte „ Destination“ verglichen.
3. Stimmt das Ergebnis damit überein, werden Gateway und zugehöriges Interface bestimmt.
4. Nachdem die MAC-Adresse des Gateways ggf. via ARP aufgelöst wurde, wird das Paket mit einem neuen Ethernet-Header versehen und weitergeleitet.

Beispiel: R1 erhalte ein Paket mit der Zieladresse 172.16.1.23.

	Destination	NextHop	Costs	Iface
→	192.168.255.8/30	0.0.0.0	0	eth2
	192.168.255.0/29	0.0.0.0	0	eth1
	192.168.0.0/24	0.0.0.0	0	eth0
	172.16.1.0/24	192.168.255.3	1	eth1
	172.16.0.0/23	192.168.255.2	1	eth1
	0.0.0.0/0	192.168.255.10	0	eth2

IP-Adresse	10101100 . 00010000 . 00000001 . 00010111	172.16.1.23
Subnetz Maske	11111111 . 11111111 . 11111111 . 11111100	255.255.255.252
Netzadresse	10101100 . 00010000 . 00000001 . 00010100	172.16.1.20

$\Rightarrow$ kein Match, da 172.16.1.20 $\neq$ 192.168.255.8

Routing Table und Longest Prefix Matching

Longest Prefix Matching

1. R1 berechnet das logische AND aus der Zieladresse des Pakets und den Subnetzmasken (welche aus der Präfixlänge hervorgehen) in seiner Routingtabelle.
2. Das Ergebnis wird mit dem Eintrag in der Spalte „ Destination“ verglichen.
3. Stimmt das Ergebnis damit überein, werden Gateway und zugehöriges Interface bestimmt.
4. Nachdem die MAC-Adresse des Gateways ggf. via ARP aufgelöst wurde, wird das Paket mit einem neuen Ethernet-Header versehen und weitergeleitet.

Beispiel: R1 erhalte ein Paket mit der Zieladresse 172.16.1.23.

	Destination	NextHop	Costs	Iface
	192.168.255.8/30	0.0.0.0	0	eth2
→	192.168.255.0/29	0.0.0.0	0	eth1
	192.168.0.0/24	0.0.0.0	0	eth0
	172.16.1.0/24	192.168.255.3	1	eth1
	172.16.0.0/23	192.168.255.2	1	eth1
	0.0.0.0/0	192.168.255.10	0	eth2

IP-Adresse	10101100 . 00010000 . 00000001 . 00010111	172.16.1.23
Subnetz Maske	11111111 . 11111111 . 11111111 . 11111000	255.255.255.248
Netzadresse	10101100 . 00010000 . 00000001 . 00010000	172.16.1.16

$\Rightarrow$ kein Match, da 172.16.1.16 $\neq$ 192.168.255.0

Routing Table und Longest Prefix Matching

Longest Prefix Matching

1. R1 berechnet das logische AND aus der Zieladresse des Pakets und den Subnetzmasken (welche aus der Präfixlänge hervorgehen) in seiner Routingtabelle.
2. Das Ergebnis wird mit dem Eintrag in der Spalte „ Destination“ verglichen.
3. Stimmt das Ergebnis damit überein, werden Gateway und zugehöriges Interface bestimmt.
4. Nachdem die MAC-Adresse des Gateways ggf. via ARP aufgelöst wurde, wird das Paket mit einem neuen Ethernet-Header versehen und weitergeleitet.

Beispiel: R1 erhalte ein Paket mit der Zieladresse 172.16.1.23.

	Destination	NextHop	Costs	Iface
	192.168.255.8/30	0.0.0.0	0	eth2
	192.168.255.0/29	0.0.0.0	0	eth1
→	192.168.0.0/24	0.0.0.0	0	eth0
	172.16.1.0/24	192.168.255.3	1	eth1
	172.16.0.0/23	192.168.255.2	1	eth1
	0.0.0.0/0	192.168.255.10	0	eth2

IP-Adresse	10101100 . 00010000 . 00000001 . 00010111	172.16.1.23
Subnetz Maske	11111111 . 11111111 . 11111111 . 00000000	255.255.255.0
Netzadresse	10101100 . 00010000 . 00000001 . 00000000	172.16.1.0

$\Rightarrow$ kein Match, da 172.16.1.0 $\neq$ 192.168.0.0

Routing Table und Longest Prefix Matching

Longest Prefix Matching

1. R1 berechnet das logische AND aus der Zieladresse des Pakets und den Subnetzmasken (welche aus der Präfixlänge hervorgehen) in seiner Routingtabelle.
2. Das Ergebnis wird mit dem Eintrag in der Spalte „ Destination“ verglichen.
3. Stimmt das Ergebnis damit überein, werden Gateway und zugehöriges Interface bestimmt.
4. Nachdem die MAC-Adresse des Gateways ggf. via ARP aufgelöst wurde, wird das Paket mit einem neuen Ethernet-Header versehen und weitergeleitet.

Beispiel: R1 erhalte ein Paket mit der Zieladresse 172.16.1.23.

	Destination	NextHop	Costs	Iface
	192.168.255.8/30	0.0.0.0	0	eth2
	192.168.255.0/29	0.0.0.0	0	eth1
	192.168.0.0/24	0.0.0.0	0	eth0
→	172.16.1.0/24	192.168.255.3	1	eth1
	172.16.0.0/23	192.168.255.2	1	eth1
	0.0.0.0/0	192.168.255.10	0	eth2

IP-Adresse	10101100 . 00010000 . 00000001 . 00010111	172.16.1.23
Subnetz Maske	11111111 . 11111111 . 11111111 . 00000000	255.255.255.0
Netzadresse	10101100 . 00010000 . 00000001 . 00000000	172.16.1.0

⇒ Match, da 172.16.1.0 = 172.16.1.0 ⇒ Gateway ist 192.168.255.3

Routing Table und Longest Prefix Matching

Definition: Longest Prefix Matching

Die Routingtabelle wird von längeren Präfixen (spezifischeren Routen) hin zu kürzeren Präfixen (weniger spezifische Routen) durchsucht. Der erste passende Eintrag liefert das Gateway (Next-Hop) eines Pakets. Diesen Prozess bezeichnet man als Longest Prefix Matching.

Beachte:

Destination	NextHop	Costs	Iface
192.168.255.8/30	0.0.0.0	0	eth2
192.168.255.0/29	0.0.0.0	0	eth1
192.168.0.0/24	0.0.0.0	0	eth0
172.16.1.0/24	192.168.255.3	1	eth1
172.16.0.0/23	192.168.255.2	1	eth1
0.0.0.0/0	192.168.255.10	0	eth2

- Der Eintrag für 172.16.0.0/23 liefert ebenfalls einen Match, ist aber weniger spezifisch als der für 172.16.1.0/24 (1 bit kürzeres Präfix).
- Die Default Route 0.0.0.0/0 liefert immer einen Match (logisches AND mit der Maske 0.0.0.0).
- Es ist nicht garantiert, dass das Gateway/Next-Hop der Default Route („Gateway of last resort“) eine Route zum Ziel kennt ($\rightarrow$ ICMP Destination Unreachable / Host Unreachable).
- Routen zu direkt verbundenen Netzen (also solchen, zu denen ein Router selbst gehört) können automatisch erzeugt werden. Der NextHop ist in diesem Fall die unspezifizierte Adresse.
- Routen zu entfernten Netzen müssen „gelernt“ werden – entweder durch händisches Eintragen (statisches Routing) oder durch Routing Protokolle (dynamisches Routing).