Grundlagen

-

Internetprotokoll

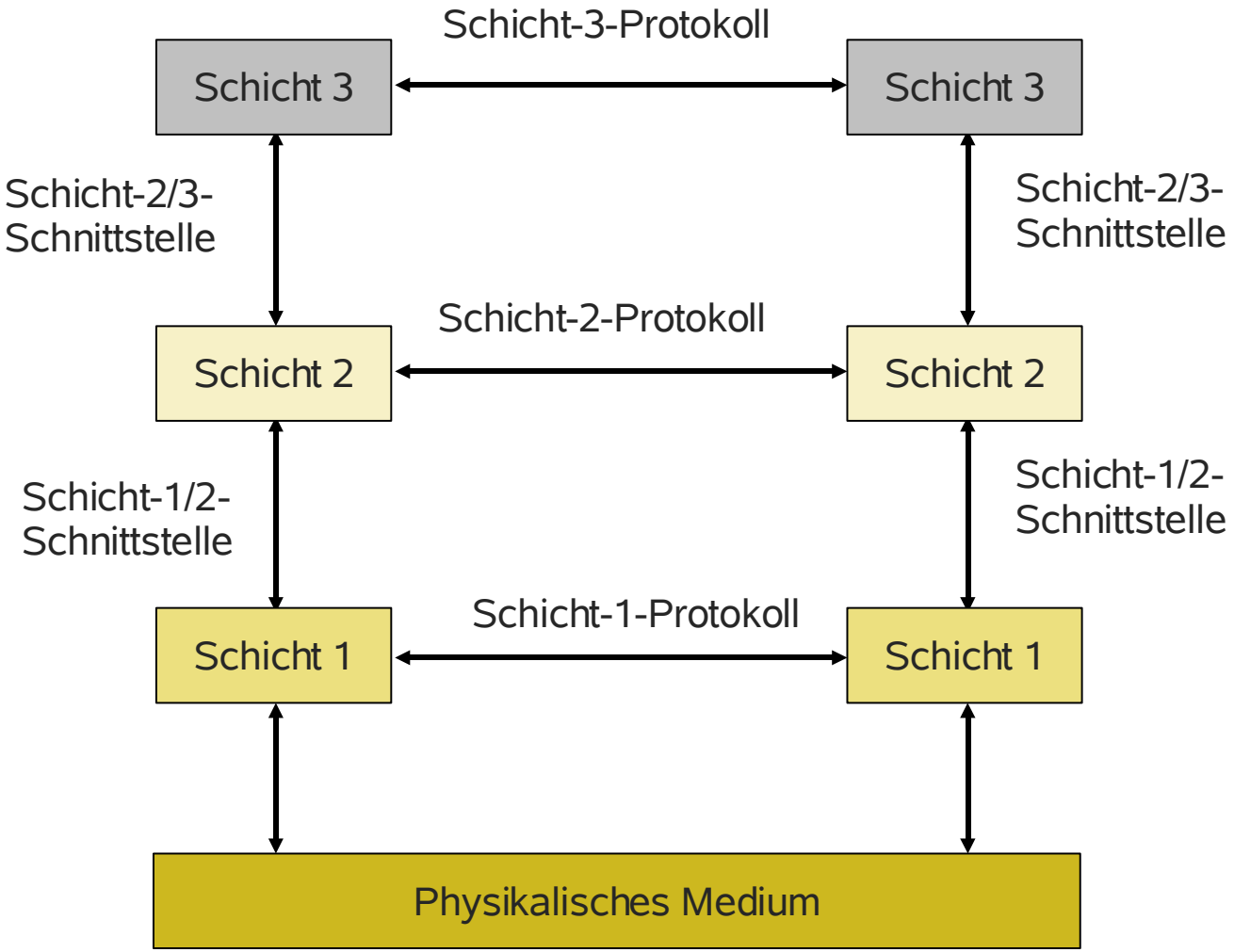
Schichten

- Hierarchische Organisation
  - Vorgesetzte verlassen sich auf die Leistung der Untergebenen
  - präsentieren ihrem Vorgesetzten wiederum ein funktionierendes System mit geringerer Komplexität
  - Dadurch können höherwertige Funktionen realisiert werden
- Beispiel
  - Vorgesetzter beauftragt den Bauherrn, für Unterkünfte zu sorgen
  - Bauherr beauftragt die Baugesellschaft, ein Haus zur Verfügung zu stellen
  - Baugesellschaft beauftragt die Bauleiter, das Haus zu bauen
  - Bauleiter beauftragt seine Vorarbeiter, die Zimmer zu errichten
  - Vorarbeiter beauftragt die Maurer, die Wand zu bauen
  - Maurer bauen die Wand aus Ziegelsteinen

Protokollhierarchien in der IT

- Reihe von übereinandergestapelten Schichten verringern die Komplexität
- Jede Schicht abstrahiert von der darunter liegenden Schicht und bietet durch eine Schnittstelle der jeweils nächsthöheren Schicht Dienste an
- Ein **Protokoll** legt die Regeln und Konventionen fest, wie die Schicht n der Maschine A mit Schicht n der Maschine B kommuniziert.
- Eine Protokollhierarchie nennt man auch Protokollstapel (Protokollstack, Protocol stack)

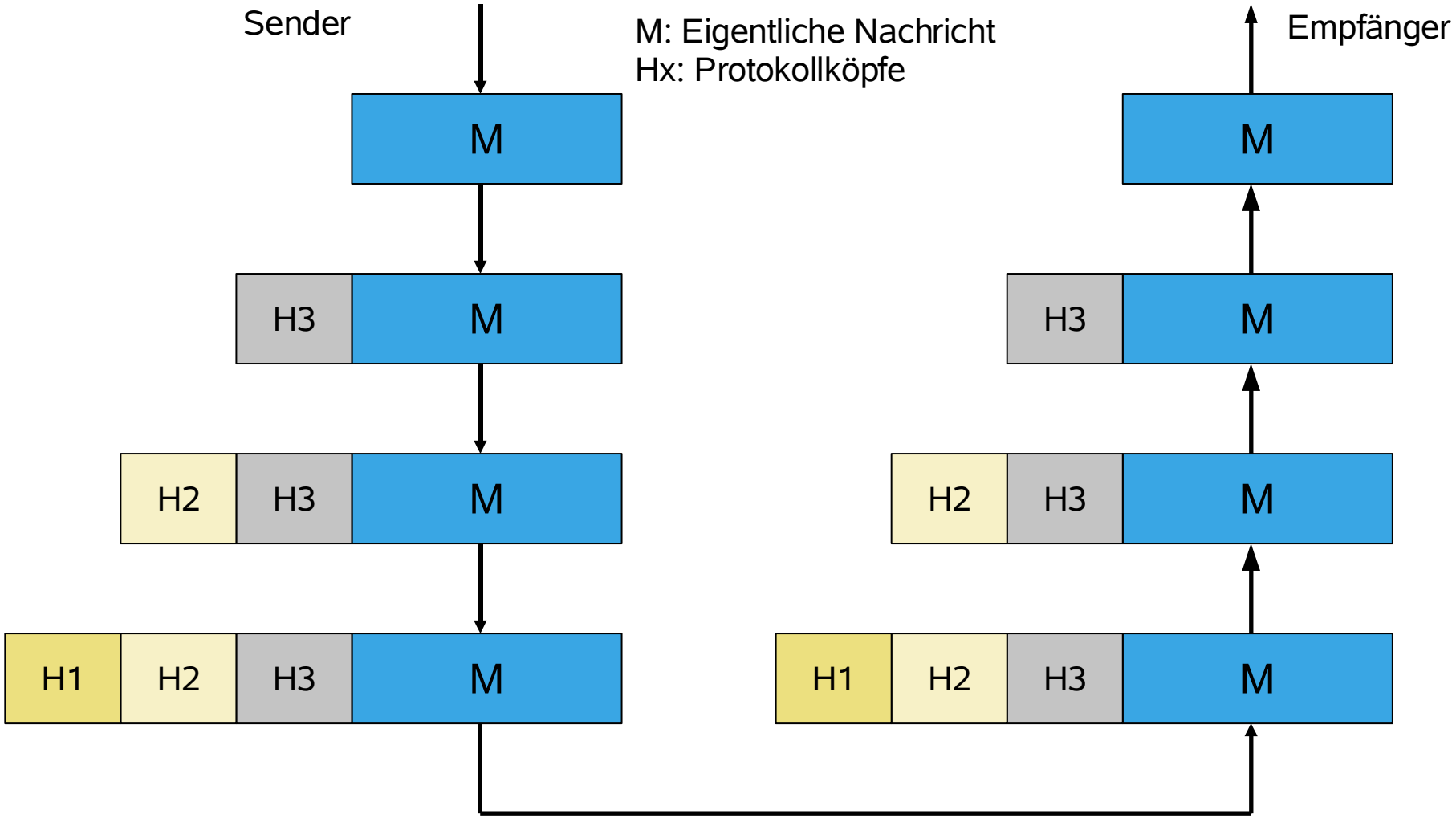
Protokollhierarchien



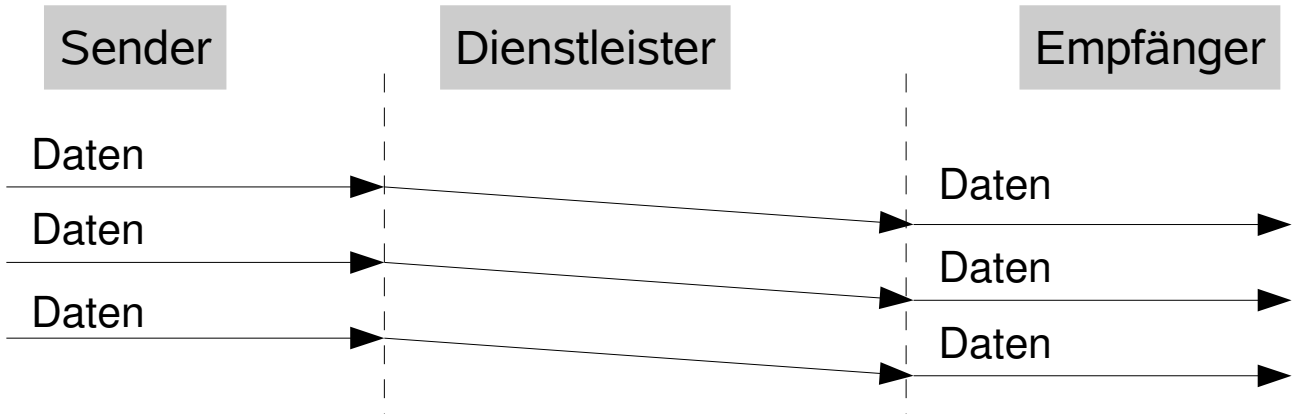
Protokollhierarchien in der IT

- Jede Schicht überträgt die Nachricht in einer Weise, die ihr Ziel versteht
  - Nachricht muss entsprechend um Informationen ergänzt werden, die in oberen Schichten nicht sichtbar sind.
  - Informationen der empfangenen Nachricht müssen entsprechend verarbeitet und wieder entfernt werden, bevor die Nachricht in die nächsthöhere Schicht geleitet wird.
- Nachrichten bestehen aus einem Inhaltsteil (Nutzlast) und einem verarbeitungsbezogenen Teil (Kopf)

Protokollhierarchien

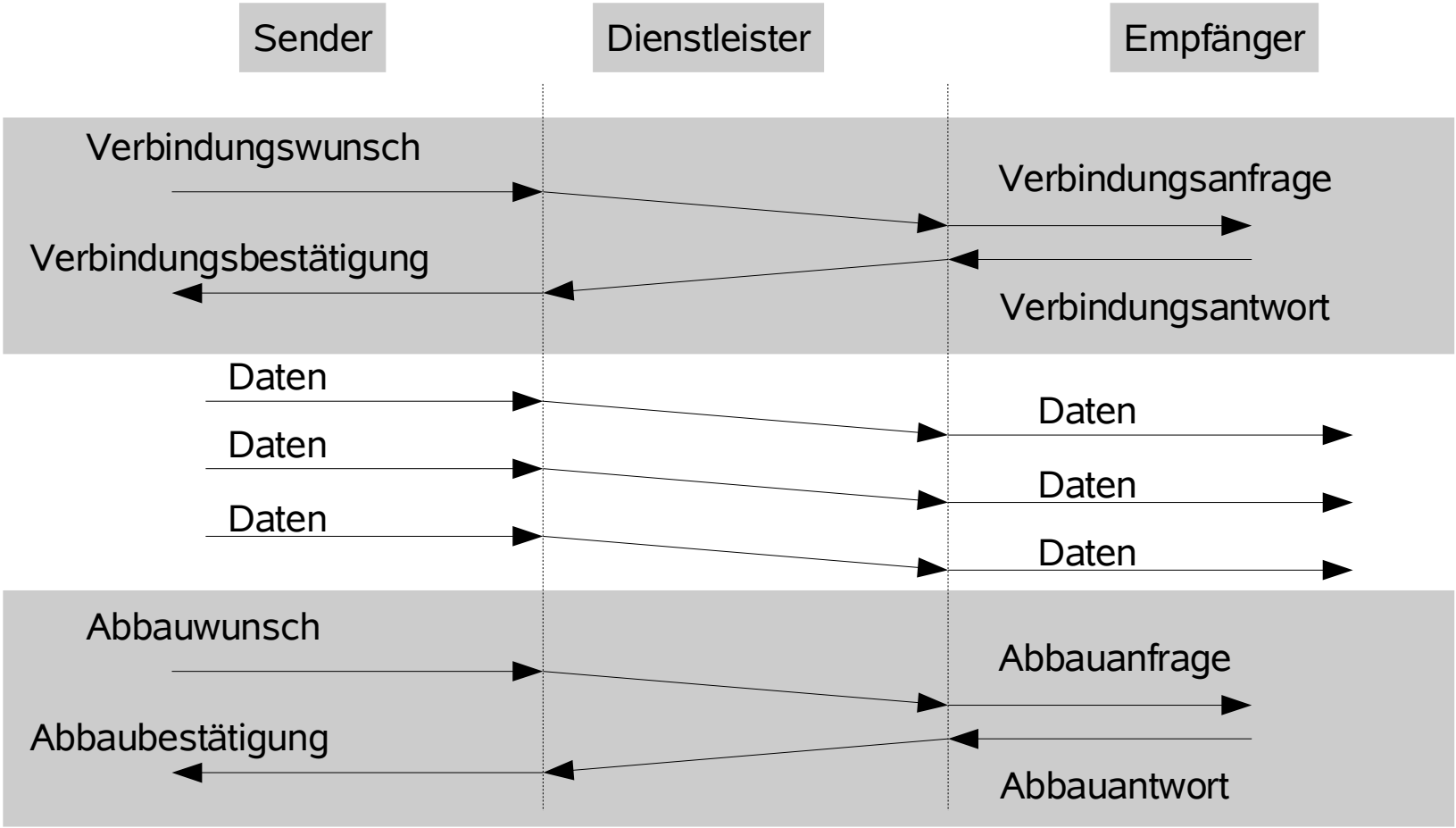


Verbindungsloser Dienst



- Ähnelt der Postzustellung
- Unidirektionale Kommunikation ausreichend
- Pakete sind unabhängig voneinander
  - Empfangsreihenfolge der Pakte kann variieren (je nach unterliegendem Netzwerk)
  - Pakete können verloren gehen (es gibt keinen Beleg für das Fehlen eines Pakets außer über Applikationsdaten)

Verbindungsorientierter Dienst



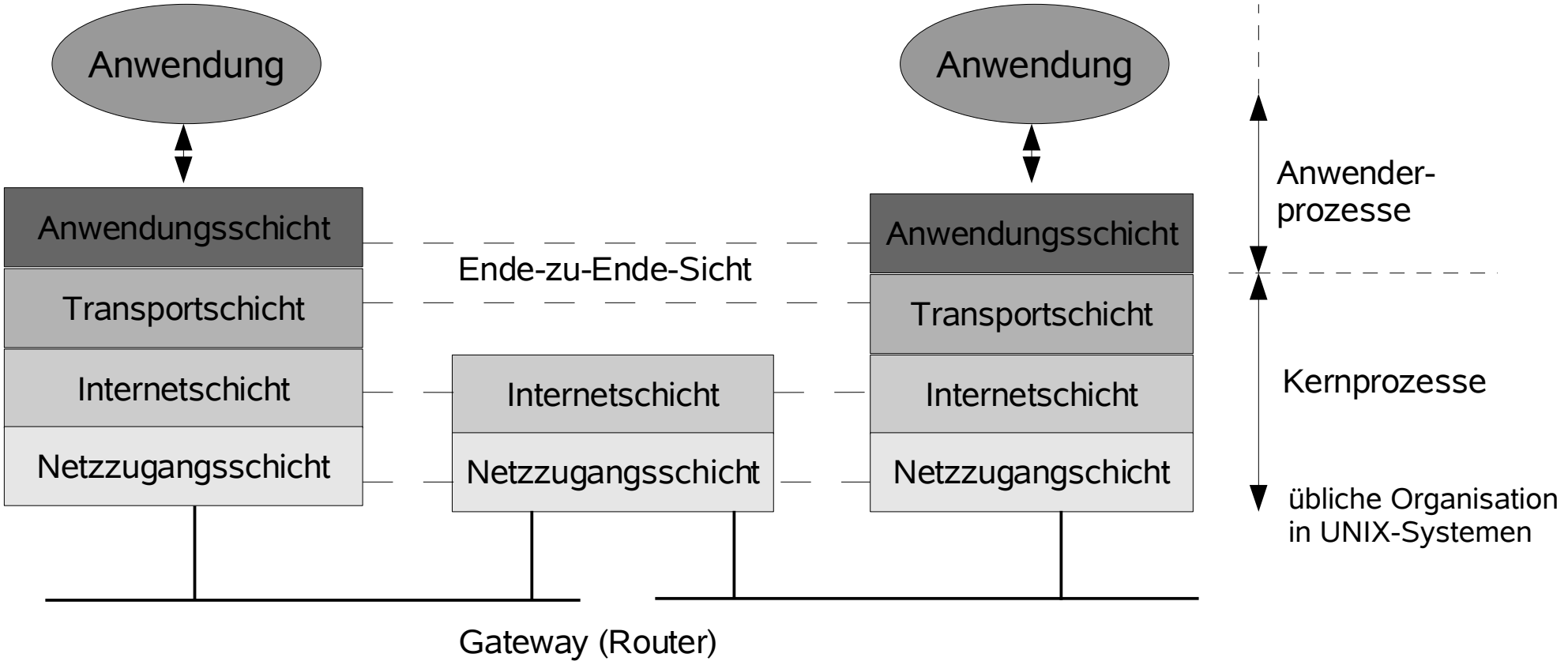
Verbindungsorientierter Dienst

- Ähnelt dem Telefon
- Kommunikation geht durch drei Phasen
  - Verbindungsaufbau (Wählen, klingeln, abnehmen)
  - Austausch von Daten (Sprechen)
  - Verbindungsabbau (Auflegen)
- Ermöglicht Verhandeln von Übertragungsparametern
- Erfordert bidirektionale Kommunikation

Verlässlich – unzuverlässig

- Von der Verbindungseigenschaft zu trennen!
  - Es gibt (verlässliche | unzuverlässige) Verbindungs-(orientierte | lose) Kommunikation.
- Verlässlich
  - Das Kommunikationsystem garantiert die Vollständigkeit und Korrektheit der überlieferten Daten.
  - Unzuverlässig: Diese Garantie ist entsprechend nicht gegeben
- Höhere Schichten können Verlässlichkeit ermöglichen
- Verbindungsorientierte Kommunikation erleichtert die Implementierung von Verlässlichkeit
  - Es kann z.B. verlangt werden, dass bis zum Abbau der Kommunikation alle Pakete angekommen sind.

TCP/IP-Referenzmodell



TCP/IP-Schichten

- Netzzugang
  - Senden, weiterleiten und empfangen einzelner Pakete zwischen direkt miteinander kommunizierenden Rechnern
- Internet
  - Pakete werden gemäß Routing-Tabellen weitergeleitet; zwischen Start und Ziel können weitere Router liegen
- Transport
  - Ende-zu-Ende-Sicht
  - Ermöglicht die Kommunikation zwischen Anwendungen
  - TCP verbindungsorientiert und zuverlässig, UDP ermöglicht unzuverlässige, verbindungslose Kommunikation von Nutzdatenpaketen
- Anwendung
  - Stellt Applikationsprotokolle wie HTTP, FTP oder SMTP zur Verfügung

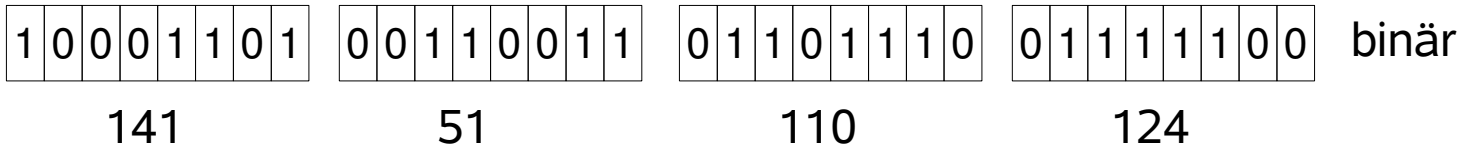
Das Internet-Protokoll

IP-Dienstmodell

- Einheitliches Adressierungsschema
  - Jeder Rechner erhält eindeutige Kennung
  - Aufteilung in „Netz“ und „Rechner“
- Verbindungslose Auslieferung von Datenpaketen
  - „Best-effort“-Kommunikation
    - Das Netz kann Pakete verlieren, duplizieren, verzögern, Reihenfolge ändern, ...
  - Beliebig heterogene Netze
    - IP ist „kleinster gemeinsamer Nenner“ für alle Netze
  - Keine Ende-zu-Ende-Eigenschaft!
    - Es müssen Zwischenstopps eingeplant werden.

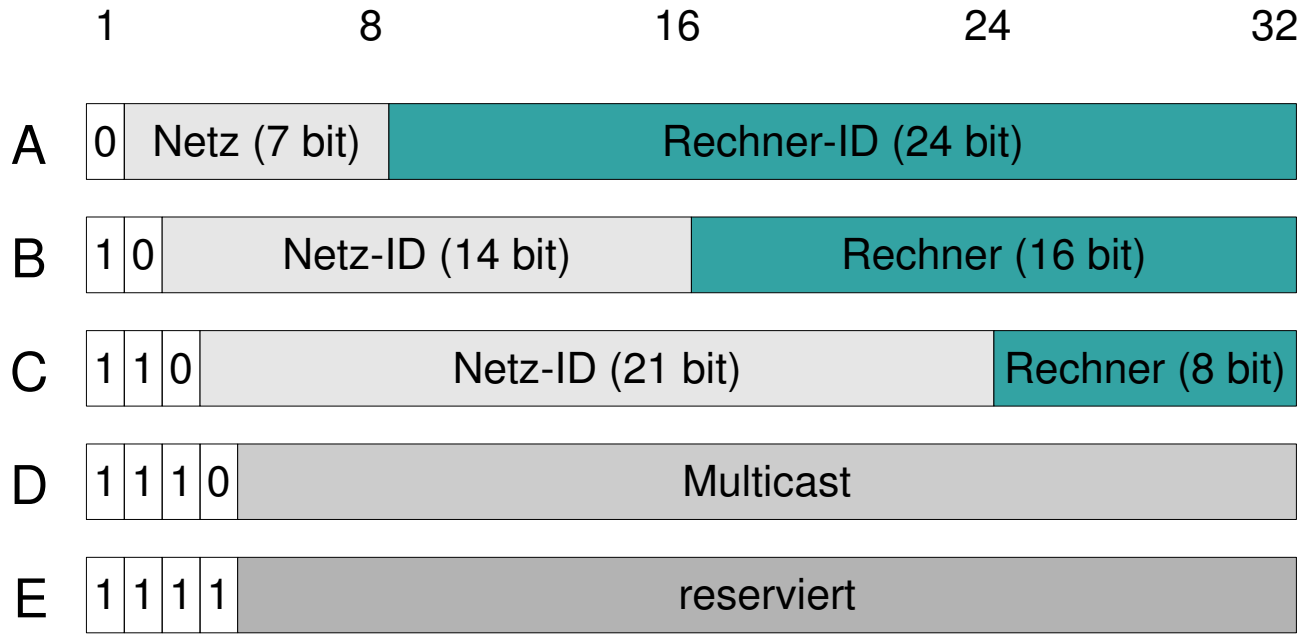
IP-Adressen (IP Version 4)

- Jeder Anschluss bekommt eigene IP-Adresse
  - Jede (erreichbare) IP-Adresse darf nur einem Gerät zugeordnet sein.
- 32 Bits ⇒ 4.294.967.295 Adressen
  - Reicht für 8 Adressen pro km²
- Notation: Folge von vier 8-Bit-Dezimalzahlen (Bytes)
  - durch Punkte voneinander getrennt
  - Beispiel: 141.51.110.124 (unser Webserver)



Netzklassen (v4)

- IP-Adressen sind unterteilt in „Netz“ und „Rechner“
  - Bei der Weiterleitung braucht der Router nur das „Netz“ zu beachten
  - Zielnetz sorgt für Weiterleitung an Zielrechner



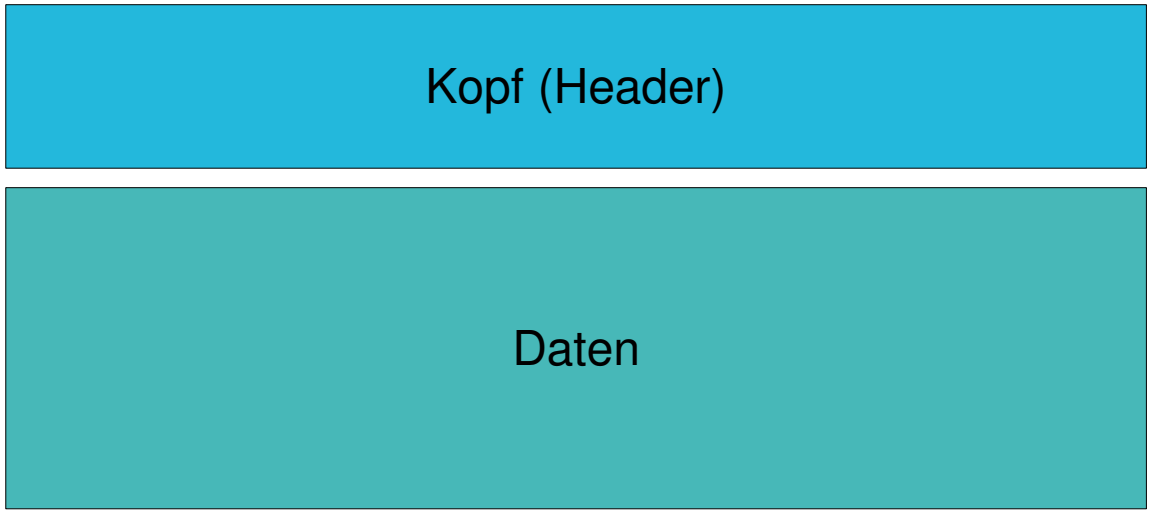
Netzklasse

IP-Adressen (IP Version 4)

- Adressbereiche
  - Klasse A: 1.0.0.0 bis 127.255.255.255
  - Klasse B: 128.0.0.0 bis 191.255.255.255
  - Klasse C: 192.0.0.0 bis 223.255.255.255
  - Klasse D: 224.0.0.0 bis 239.255.255.255 („Multicast“)
  - Klasse E: 240.0.0.0 bis 255.255.255.255 (reserviert)
- Reserviert für lokale Netze
  - 10.0.0.0 bis 10.255.255.255; 172.16.0.0 bis 172.31.255.255; 192.168.0.0 bis 192.168.255.255
- Besondere Bedeutung
  - 0.0.0.0: dieser Rechner
  - 127.x.x.x: „Loopback“, derselbe Rechner (meist 127.0.0.1)
  - Nur Einsen für Rechner (z.B. 192.168.0.255): Broadcast (d.h. Sendung an alle)

Aufbau eines IPv4-Pakets

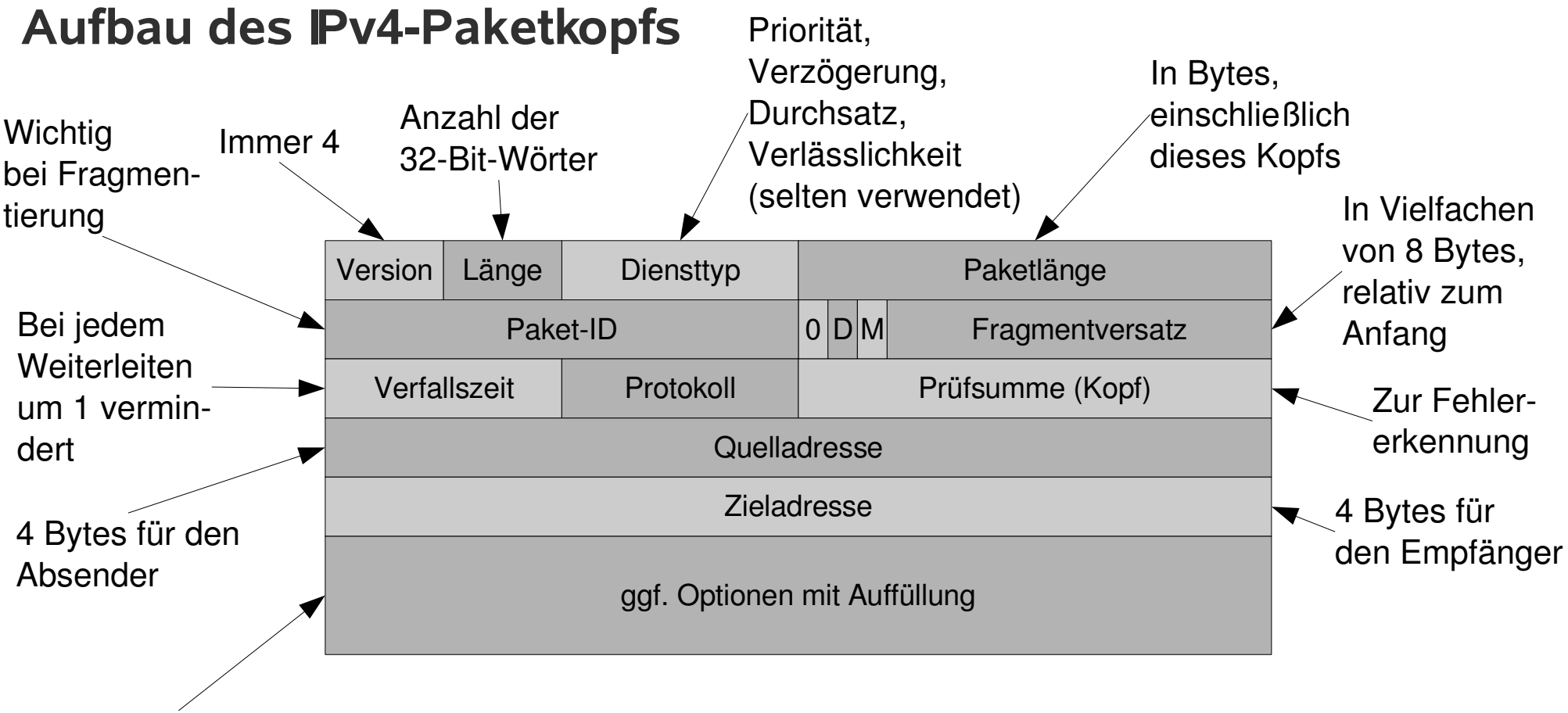
- Verbindungslose Kommunikation
  - Jedes Paket trägt die notwendige Information, um von der Quelle zum Ziel zu gelangen



Einziger IP-Kopf

Daten können protokollspezifische Köpfe beinhalten (z.B. für TCP, darüber wiederum HTTP usw.)

Aufbau des IPv4-Paketkopfs

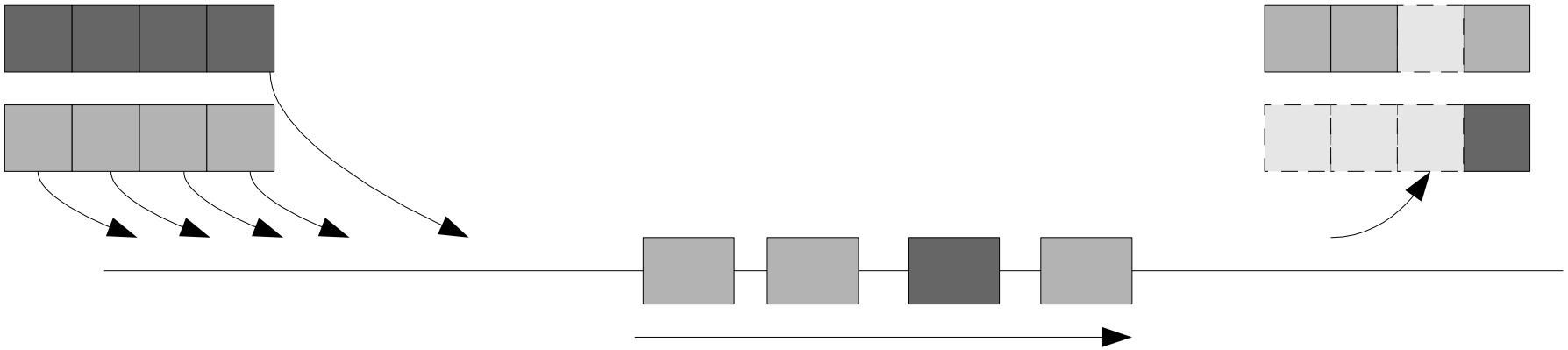


Flags
D: „Don't fragment“
M: „more fragments“

Protokoll
Kennzeichnung des darüber liegenden Transportprotokolls (z.B. TCP, UDP, ICMP,...)

Fragmentierung

- Längenfeld: Pakete können prinzipiell bis zu 64KiB groß werden
  - Jedoch können manche Übertragungsmedien keine so langen Pakete senden
- Lösung
  - Pakete müssen für die jeweilige Übertragung zerteilt und später wieder zusammengesetzt werden („fragment / reassemble“)
  - Jeder Router darf zerlegen, zusammensetzen kann nur das Ziel
  - Orientiert sich an Paket-ID und Fragmentversatz
    - Paket-ID gleich für alle Fragmente desselben Pakets



Wie gelangen IP-Pakete durch mehrere Netze zum Ziel?

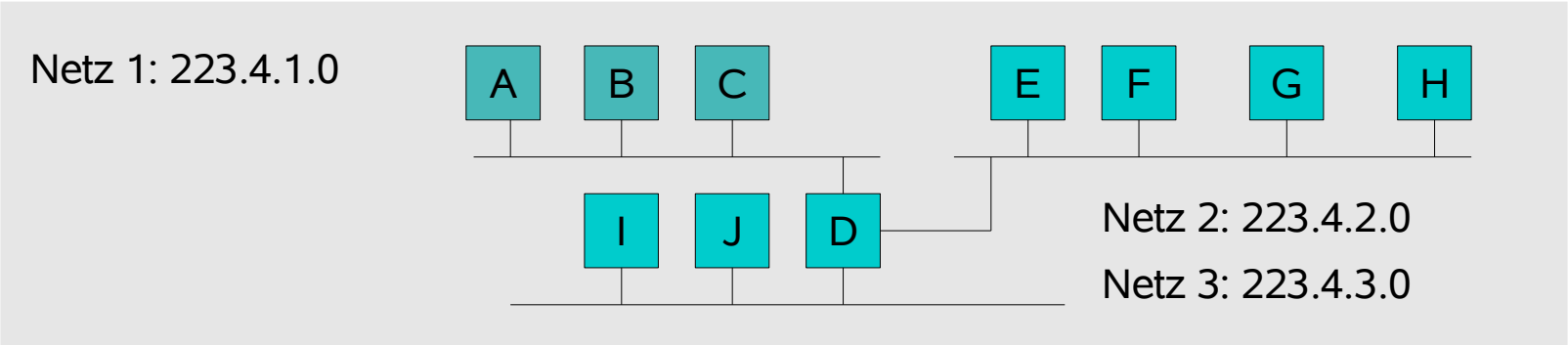
- Notwendige Voraussetzungen
  - Jedes IP-Paket enthält eine Zieladresse
  - Der „Netz“-Teil der IP-Adresse bezeichnet eindeutig ein einziges physisches Netz im Internet
  - Alle Rechner und Router, die die gleiche Netzadresse haben, können direkt miteinander kommunizieren
  - Jedes Netz im Internet hat mindestens einen Router, der mit mindestens einem anderen Netz im Internet verbunden ist
- „Routing“
  - Falls der Empfänger des Paket im selben Netz ist, schicke es direkt.
  - Falls nicht, schicke das Paket an einen geeigneten Router.

IP-Routing

- IP-Routing ändert nicht die Adressen von Quelle und Ziel
- Paket wird „schrittweise“ von Router zu Router weitergereicht, bis es sein Ziel erreicht
- Jeder Router (und der absendende Rechner) beinhaltet eine „Routingtabelle“
  - beinhaltet Rechner (Hosts) oder Netze als Ziele
- Routingtabelle ordnet IP-(Ziel)-Adressen die nächste zu wählende Teilstrecke zu
  - Direktes Routing (im selben Netz)
  - Indirektes Routing (über Router, auch Gateway genannt)
  - Standardrouter („default“), wenn keine speziellere Angabe in der Tabelle passt

Priorität: Rechnereintrag vor Netzeintrag vor default

Routingtabellen



Netzwerk	direkt/indirekt	Gateway	Schnittstelle
223.4.1	direkt	-	1
223.4.2	indirekt	D	1
223.4.3	indirekt	D	1
default	indirekt	D	1

Routing-Tabelle in A

Netzwerk	direkt/indirekt	Gateway	Schnittstelle
223.4.1	direkt	-	1
223.4.2	direkt	-	2
223.4.3	direkt	-	3

Routing-Tabelle in D

Praktische Beispiele

dsl0 Protokoll:Punkt-zu-Punkt Verbindung
inet Adresse:217.83.121.37 P-z-P:217.5.98.78 Maske:255.255.255.255
UP PUNKTZUPUNKT RUNNING NOARP MULTICAST MTU:1492 Metric:1

eth0 Protokoll:Ethernet Hardware Adresse 00:0E:7B:76:F2:0B
inet Adresse:192.168.111.121 Bcast:192.168.111.255 Maske:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

lo Protokoll:Lokale Schleife
inet Adresse:127.0.0.1 Maske:255.0.0.0

```
$ netstat -rn
Kernel IP Routentabelle
Ziel            Router          Genmask         Flags   ... Iface
217.5.98.78     0.0.0.0         255.255.255.255 UH      ... dsl0
192.168.111.0   0.0.0.0         255.255.255.0   U       ... eth0
169.254.0.0     0.0.0.0         255.255.0.0     U       ... eth0
127.0.0.0       0.0.0.0         255.0.0.0       U       ... lo
0.0.0.0         217.5.98.78     0.0.0.0         UG      ... dsl0
```

Standardroute („default“)

U=up (aktiv), H=Host, G=Gateway

Erläuterung des Beispiels

- Flags
  - U=Up: Dieser Eintrag ist zu verwenden; er verweist auf ein Ziel, das direkt angeschlossen ist oder über ein Gateway erreicht wird
  - G=Gateway: Der Eintrag verweist auf ein Ziel, das über ein Gateway (Router) erreicht wird (Rechner oder Netz)
  - H=Host: Der Eintrag verweist auf einen Rechner als Ziel
  - D=Dynamic: Der Eintrag fand automatisch statt (z.B. über RIP, „Routing Information Protocol“), sonst statisch, also per Konfiguration
  - M=Modified: Der Eintrag wurde automatisch modifiziert
- Localhost
  - Lokale Schleife; liegt im Netz 127.0.0.0/8, das direkt angeschlossen ist (also „U“)
  - Manche Systeme definieren sich weitere „localhosts“, etwa 127.0.0.2

Erläuterung des Beispiels

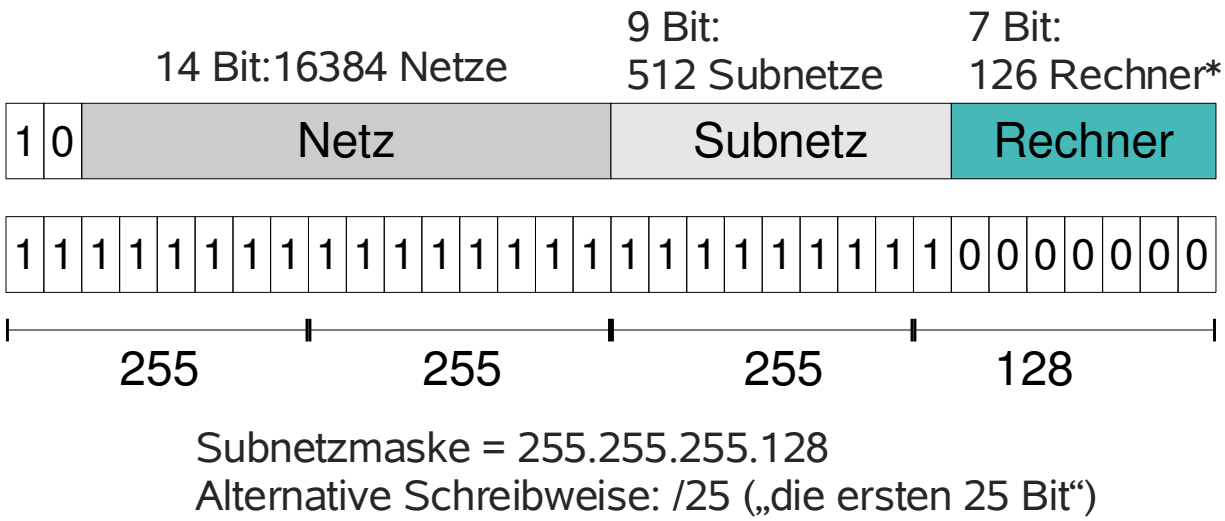
- Default: Netzmaske des Ziels ist unbekannt; daher 0.0.0.0
- Host-Eintrag („UH“)
  - Eintrag „255.255.255.255“, wenn es sich um einen Rechner handelt
  - UGH: Host, aber nur über ein Gateway erreichbar (eher selten)
  - Bei Punkt-zu-Punkt-Verbindungen wird häufig die eigene Punkt-zu-Punkt-Adresse als Gateway angegeben (wäre hier 217.83.121.37), manchmal auch 0.0.0.0 („dieser Rechner“)

IP-Subnetze

- Verwaltung der Netze ist einfacher, je weniger Netze verwaltet werden müssen
  - Für jedes Netz muss ein Eintrag in die Routingtabelle (insbesondere bei wichtigen Routern)
  - Problem: Es gibt rechnerisch 2,1 Millionen Klasse-C-Netze!
- Andererseits Adressverschwendung
  - Klasse-B-Netze könnten 65534 Rechner aufnehmen (was sie selten tun).
- Konzept des „Subnetzes“
  - Teil der Rechneradresse wird als eigenes Netz (im B-Netz) aufgefasst
  - Von „außen“ nicht zu sehen; es wird einfach zum B-Netz geroutet.
  - Vermeidet überlaufende Weiterleitungstabellen
    - Routing des Subnetzes kann dem zuständigen Router des Netzes überlassen werden

IP-Subnetze

- Beispiel einer Subnetzmaske anhand einer Klasse-B-Adresse



Subnetzmaske = 255.255.255.128
Alternative Schreibweise: /25 („die ersten 25 Bit“)

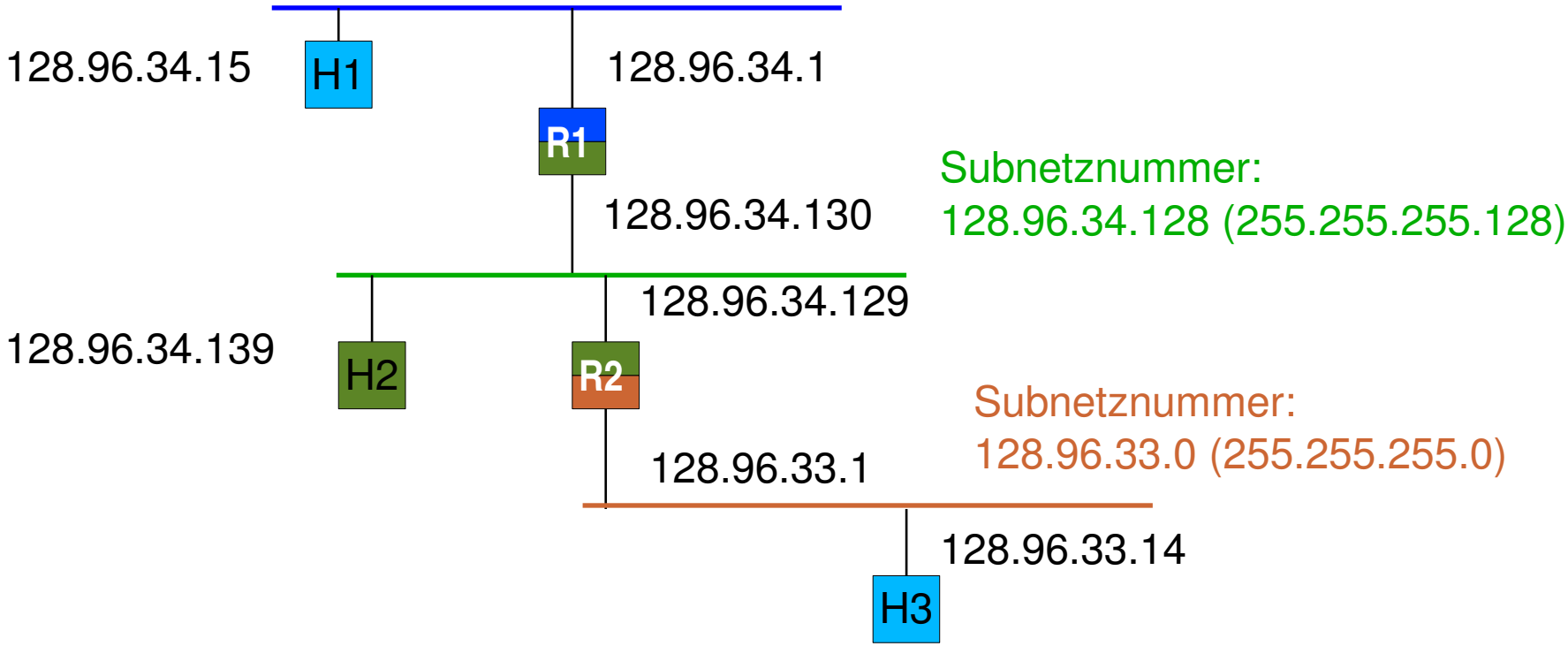
*: nur Nullen oder nur Einsen nicht für Rechneradresse zulässig

IP-Subnetze

- Rechner kennen eigene IP-Adresse und Subnetzmaske
- Alle Rechner im gleichen Subnetz haben dieselbe Maske
- Beim Senden eines Datagramms wird ein logisches AND von Zieladresse und Subnetzmaske berechnet
  - Ist das Ergebnis gleich der eigenen Subnetznummer, dann direktes Senden, sonst indirektes Senden über Router
- Routingtabelle muss für den nächsten Hop auch immer neben der Subnetznummer die Subnetzmaske bereithalten

IP-Subnetze

- Beispiel: Klasse-B-Adresse



IP-Subnetze

1. Fall: H1 schickt an H2

   (Adresse von H2) AND (Maske von H1)
   =128.96.34.139 AND 255.255.255.128 = 128.96.34.128

   → also nicht Subnetz von H1, d.h. über R1 schicken

2. Fall: H1 an R1

   (Adresse von R1) AND (Maske von H1)
   = 128.96.34.1 AND 255.255.255.128 = 128.96.34.0

   → gleiches Subnetz, d.h. direkt schicken

Klassenloses Routing

- Starre Netzklassen sind lange überholt
  - Zuordnung der Netzgrößen anhand der IP-Adresse ist **praktisch nicht mehr möglich**
  - Routingtabellen müssen verpflichtend immer die Subnetzmaske verwenden
  - Es können sogar größere Netze vergeben werden: 194.76.48.0/20
- CIDR = Classless Interdomain Routing
  - Keine Unterscheidung Netz – Subnetz, nur die angegebene Netzmaske entscheidet
  - Netzmaske wird meist nur noch in der Form „/Bits“ angegeben
- Zusammenfassung von Routen (Routenaggregation)
  - Routing zu den Netzen 141.51.12.0/24 und 141.51.13.0/24 kann auch durch den gemeinsamen Eintrag 141.51.12/23 erfolgen

10001101 00110011 00001100 xxxxxxxx
10001101 00110011 00001101 xxxxxxxx

10001101 00110011 0000110x xxxxxxxx

IPv6 – das neue Internetprotokoll

IP Next Generation (IPv6)

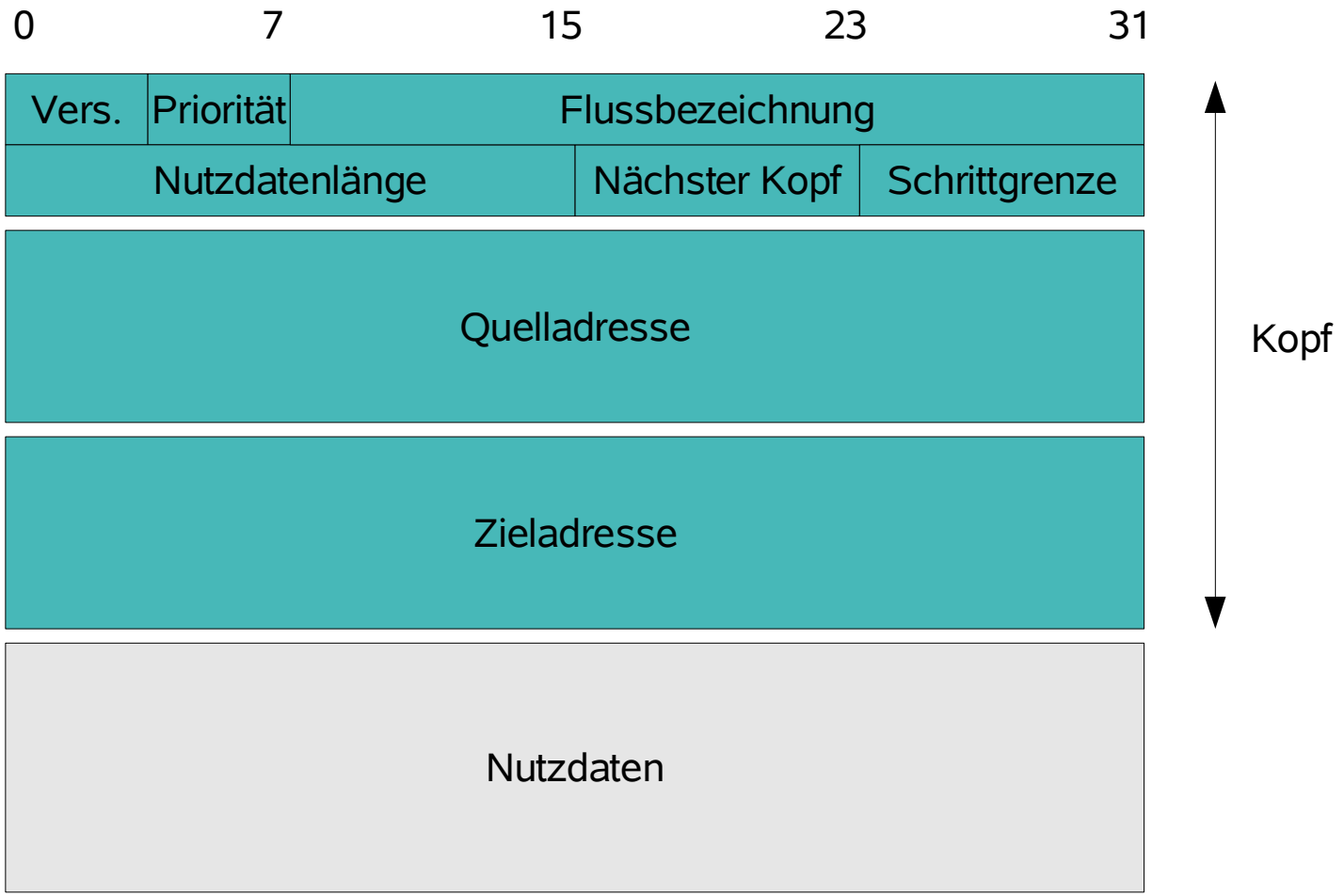
- explosionsartiges Wachstum + neue Anwendungsanforderungen machen IPv4 "zu schaffen"
- seit 1990 Arbeiten an einem neuen Protokoll IPv6
  - Version 6, weil "Stream Protocol" bereits als Version 5 bezeichnet
- Ausschreibung, Auswahl, Abstimmung, Politik etc. führten zu neuem Vorschlag im Jahr 1995 [RFC1752]

IPv6

- Ziele
  - größerer Adressbereich
  - Reduktion des Umfangs der Routing-Tabellen
  - Beschleunigung der Router-Funktion
  - mehr Sicherheit
  - Unterstützung neuartiger Datenströme
  - Unterstützung mobiler Hosts
  - Koexistenz mit IPv4 (für eine gewisse Zeit)
  - verbesserte Multicast*-Kommunikation

*Broadcast: alle Rechner eines Bereichs adressieren
Multicast: eine bestimmte Auswahl von Rechnern adressieren
Unicast: genau einen Rechner adressieren

Aufbau eines IPv6-Pakets



IPv6

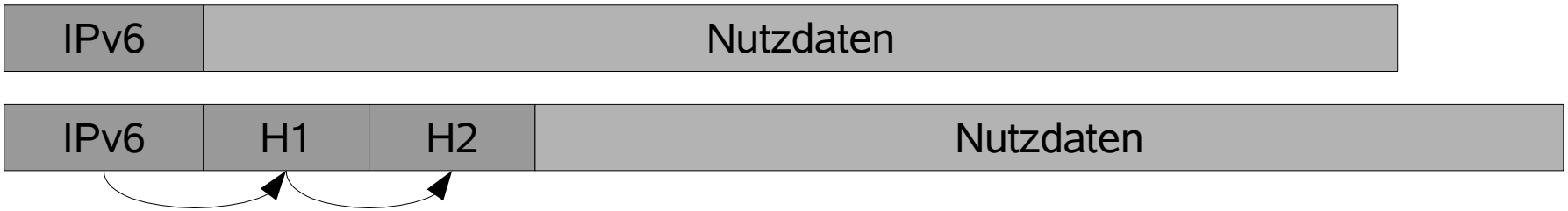
- Aufbau eines IPv6-Pakets
  - Feste Größe; einfache, schnelle Verarbeitung
  - Optionale Erweiterungs-Header
  - weniger Felder als bei IPv4
  - 128-Bit-Adressen (16 Bytes)
  - Keine Prüfsumme mehr
- Fragmentierung nur noch an der Quelle
  - Strategie: Sender stellt fest (anhand von Rückmeldungen), ob die Paketgröße für den gesamten Weg passend ist und verkleinert die Pakete, falls nötig
  - Anfängliche Fehlversuche in Kauf genommen für einfachere Verarbeitung (die Übertragungseigenschaften ändern sich in der Regel nicht während der Übertragung)

IPv6

- Aufbau des IPv6-Protokollkopfs
  - Version — Version des IP-Protokolls (hier: 6)
  - Priorität — Priorität zur Unterscheidung von Datenströmen
15-8: Echtzeitverkehr; 7-0: Datenverkehr
  - Flussbezeichnung — Markierung von (Pseudo-)Verbindungen
mit best. Eigenschaften, z.B. reservierter Bandbreite
  - Nutzdatenlänge — Gesamtlänge der Daten (in Bytes)
  - Nächster Kopf — ob und welche Erweiterungen folgen
  - Schrittgrenze — entspricht Verwendung des Time-to-Live-Zählers in IPv4
  - Quelladresse — Absender
  - Zieladresse — Empfänger

IPv6

- Neu: Modularer Protokollkopf
  - Fokussierung auf die meistbenutzten Felder
  - bei Bedarf weitere Köpfe über eine verkettete Liste erreichbar



- Beispiele für Zusatzköpfe
  - Hop-by-Hop: Optionen, die mit jedem Abschnitt neu berechnet werden
  - Routing: zusätzliche Routing-Informationen, z.B. Routenangabe
  - Jumbogram: Größere Paketlänge (neues Längenfeld mit 4 Byte → 2^{32} Byte)

IPv6-Adressen

- Adressen werden den Schnittstellen eines IPv6-Netzknotens (Host oder Router) zugeordnet
  - IPv6-Adressen sind 128 Bit (16 Byte) lang
  - Keine statische Aufteilung in <Netz, Rechner>
- Neue Schreibweise
  - 8 Gruppen von je vier Hex-Ziffern (d.h. je 2 Byte), z.B. 8000:0000:0000:0000:0123:4567:89AB:CDEF
- Konventionen für die Notation:
  - führende Nullen können weggelassen werden, z.B. 8000:0:0:0:123:4567:89AB:CDEF
  - Felder mit 0 können **einmal** zu einem Block zusammengefasst werden, z.B. 8000::123:4567:89AB:CDEF
  - Adresspräfixe in CIDR-Schreibweise: 12AB:0:0:CD30::/60

Adressbereiche

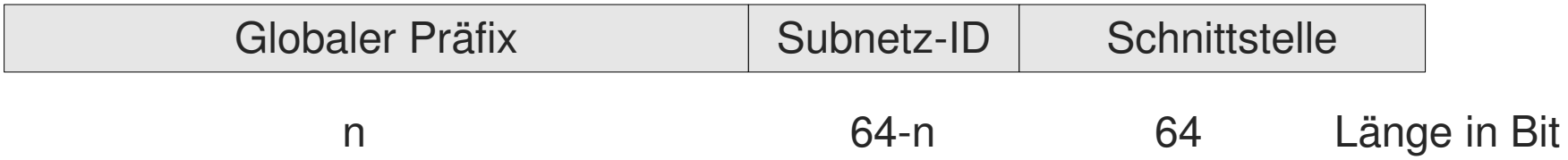
0000 0000 ... 0	Unspezifiziert (::/128)
0000 0000 ... 1	Loopback (::1/128)
1111 1110 10 ...	verbindungslokale Adressen (FE80::/10)
1111 1110 11 ...	standortlokale Adressen (FEC0::/10)
1111 1111 ...	Multicast (FF00::/8)
Alles andere	Adressen für Dienstprovider („Globale Unicast-Adressen“)

RFC 3513

- Knotenlokale Adressen (*interface-local*) nur auf diesem Rechner
- Verbindungslokale Adressen (*link-local*): nur innerhalb desselben Segments, werden nicht aus lokalem Netz oder ins Netz geroutet
- Standortlokale Adressen (*site-local*): brauchen keinen globalen Präfix, werden nicht vom oder zum Standort geroutet
- Organisationsweite Adressen (*organizational-local*): Multicast-Bereich für alle Knoten derselben Organisation, z.B. für das Providernetz

Globale Unicast-Adressen (RFC 3587)

- Stellen die weltweit erreichbaren Adressen dar (früher: aggregierbare Adressen)
  - Rolle der früheren Klasse-A/B/C-Adressen bei IPv4
  - Struktur
    - Globaler Präfix
    - Subnetz-ID: wird vom lokalen Administrator verwaltet
    - Beispiel: n=48 (mit ersten drei Bits 001), Subnetz 16 Bit
  - Schnittstellenangabe kann aus so genannter MAC-Adresse gewonnen werden (48 bit, welche die Schnittstelle kennzeichnen, z.B. 00:0D:60:8C:AC:F2)



Globale Unicast-Adressen

- Frühere Einteilung nach RFC 2374
  - Präfix (3 bit) 001 (Adressen beginnen mit 2 oder 3)
  - TLA-ID (13 bit) Top-Level Aggregation Identifier: großer, nationaler oder internationaler Provider
  - 8 bit reserviert
  - NLA-ID (24 bit) Next-Level Aggr. ID: Teilnetze eines Providers
  - SLA-ID (8 bit) Site-Level Aggr. ID: Kundenspezifisch (jetzt „Subnetz-ID“)
- Bedenken, dass dies den Einsatz von IPv6 komplizierter macht
  - Vergabepolitik sollte nicht so vorgeschrieben werden
  - Daher Neuauflage durch RFC 3587
  - Beschränkung auf Präfix gleichsam aufgehoben
  - IANA vergibt jedoch bis auf weiteres nur Adressen mit diesem Präfix

Betrieb in IPv4-Netzen

- Wie stellt man die bisherigen Adressen auf die neuen um?
- Repräsentation als ::<IPv4> oder ::FFFF:<IPv4>
  - „IPv4-kompatible Adresse“: Nullen vor IPv4-Adresse
    - Knoten versteht IPv4 und IPv6; diese Adresse ist zusätzlich zugeordnet, um über IPv4-Netze zu kommunizieren
    - Beispiel:141.51.110.124 wird zu 0000:0000:0000:0000:0000:0000:8d33:6e7c
    - Leichter lesbar in der Form ::141.51.110.124
  - „Auf IPv4 abgebildete Adresse“: Nullen, dann FFFF vor IPv4-Adresse
    - Knoten versteht kein IPv6; diese Adresse wird in IPv6-Netzen genommen, um IPv6-untaugliche Knoten zu integrieren
    - Beispiel: 141.51.110.124 wird zu 0000:0000:0000:0000:0000:FFFF:8d33:6e7c
    - Leichter lesbar in der Form ::FFFF:141.51.110.124

IPv6: Migration von v4

- Keine Umstellung „über Nacht“ möglich
- Inseln von IPv6-Hosts werden über IPv4-Netze durch „IP Tunneling“ verbunden
  - Tunnel-Endpunkte erhalten IPv4-kompatible Adressen
  - IPv6-Pakete werden von IPv4/IPv6-Routern in IPv4-Pakete ein- und ausgepackt
  - Tunnel wirkt wie ein Netzsegment mit einem Hop
- Hosts können parallel beide Protokoll-Stacks fahren
- Umstellung wird einige Jahre dauern, kann aber im lokalen Bereich bereits separat vollzogen und getestet werden