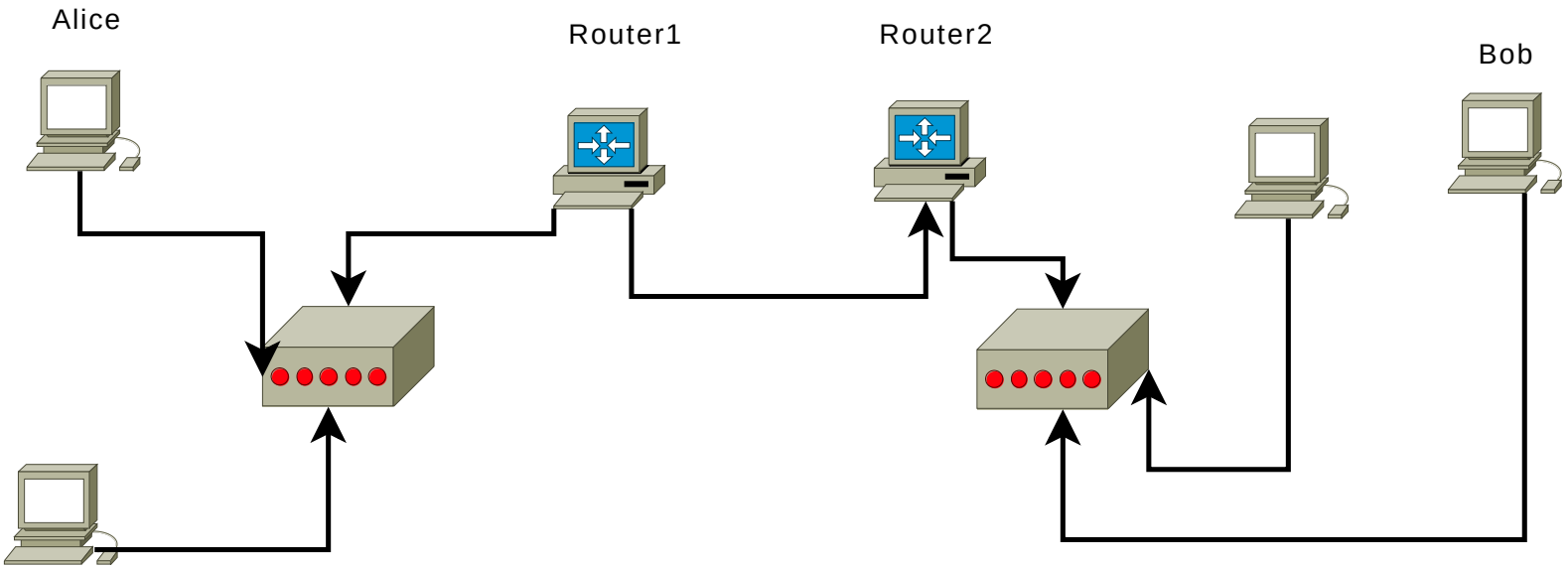
Motivation



Wir betrachten die Kommunikationsbeziehung von Alice zu Bob

Die Netzwerkschickt kennt die Netzwerkadresse des Zielrechners und kann damit die Netzwerkadresse des nächsten, d.h. mit einer Adresse der Sicherungsschicht erreichbaren, Knotens (Router1) bestimmen.

Aufgabe der Sicherungsschicht ist es, die Netzwerkadresse des nächsten Knotens in eine brauchbare Adresse umzuwandeln.

- Statische Konfiguration
  (z.B. Tabelle IP <-> Ethernet Adresse)
- Dynamisch mittels Abfrage im lokalen Netz

Rahmenstruktur, vgl. RFC826

Destination	Source	Type	Hard Type	Prot Type	Hard Size	Prot Size	Op	Sender MAC	Sender Network	Target MAC	Target Network
6	6	2	2	2	1	1	2				

- **Destination:** Ethernet Zieladresse
- **Source:** Ethernet Quelladresse
- **Type:** ARP, d.h. 0806_{16}, vgl. RFC1700

Felder im Rahmen

- **Hardware Type:** Art der nachgefragten Adresse, z.B. 1 für Ethernet
- **Protocol Type:** Art der Netzwerkadresse, z.B. 0800_{16} für IP, vgl. RFC1700
- **Hardware Size, Protocol Size:** Länge der Adressen, 6 für Ethernet, 4 für IPv4
- **Operation:** Typ des Requests
  1. ARP Request, d.h. Netzwerk $\rightarrow$ Hardware
  2. ARP Reply, Antwort
  3. RARP Request, d.h. Hardware $\rightarrow$ Netzwerk
  4. RARP Reply, Antwort dazu.
- **Sender MAC:** Identisch mit **Source**
- **Sender Network:** Netzwerkadresse der Quelle
- **Target MAC:** Hardwareadresse des Ziels
- **Target Network:** Netzwerkadresse der Ziels

Funktionsweise

Wann immer zu einer Netzwerkadresse die passende Adresse der Sicherungsschicht bestimmt werder muß, wird ein ARP Request mit entsprechenden Daten erzeugt und an alle Hosts des lokalen Netzes dieses Adapters geschickt.

Empfängt der Host mit passender Netzwerkadresse den ARP Request, antwortet er mit einem ARP Reply, in dem er seine Hardwareadresse als **Sender MAC** einsetzt. Das Ergebnis bleibt eine bestimmte (üblicherweise konfigurierbare) Zeit gespeichert (ARP Cache)

Wird der Request nicht beantwortet, wird das Verfahren nach wenigen Wiederholungen abgebrochen.

Empfängt ein Host einen ARP Request von einem Host, dessen IP im ARP Cache ist, wird die Quelladresse automatisch übernommen.

Beispiel ARP Request

```
Ethernet II
  Destination: Broadcast (ff:ff:ff:ff:ff:ff)
  Source: Agere_66:79:ca (00:02:2d:66:79:ca)
  Type: ARP (0x0806)
Address Resolution Protocol (request)
  Hardware type: Ethernet (0x0001)
  Protocol type: IP (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (0x0001)
  Sender MAC address: Agere_66:79:ca
  Sender IP address: 134.61.33.148
  Target MAC address: 00:00:00_00:00:00
  Target IP address: 134.61.32.1
```

Beispiel ARP Reply

```
Ethernet II
  Destination: Agere_66:79:ca
(00:02:2d:66:79:ca)
  Source: Ibm_3e:81:76 (00:14:5e:3e:81:76)
  Type: ARP (0x0806)
Address Resolution Protocol (reply)
  Hardware type: Ethernet (0x0001)
  Protocol type: IP (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: reply (0x0002)
  Sender MAC address: Ibm_3e:81:76
  Sender IP address: 134.61.32.1
  Target MAC address: Agere_66:79:ca
  Target IP address: 134.61.33.148
```

Gratuitous ARP

Von **Gratuitous ARP** spricht man, wenn ein Host mit einem ARP Request im LAN nach seiner eigenen Hardwareadresse fragt. Dies kann zwei Gründe haben:

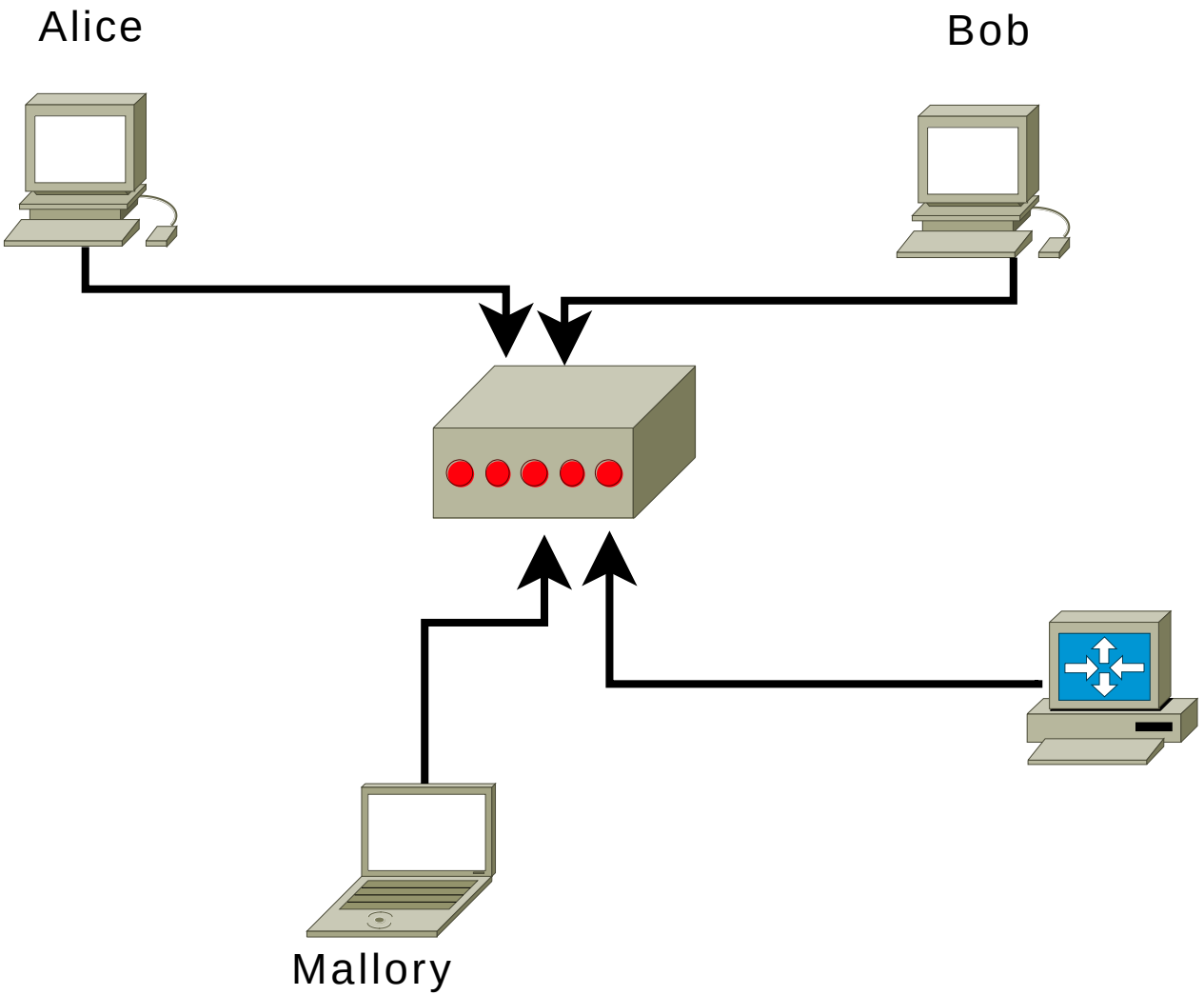
1. Wird der Request beantwortet, hat ein weiterer Host im LAN dieselbe Netzwerkadresse, was auf eine Fehlkonfiguration hindeutet und den Netzbetrieb stören kann.
2. Alle Hosts, die schon einen Eintrag zu dieser Netzwerkadresse im ARP Cache haben, werden über die (gegebenenfalls andere) Hardwareadresse informiert.

Grundlage

Falls ein Ethernet Switch die Quell-MAC eines Rechners kennt, leitet er Pakete an diesen Rechner nur noch auf dem Port weiter, hinter dem der Rechner erreichbar ist.

Möchte ein Angreifer die Kommunikation zweier anderer Rechner belauschen, kann er dazu den Switch oder die Endpunkte angreifen.

Beispielszenario



Trivialansatz

Manche Ethernet Switche haben eine (natürlich endliche) Tabelle mit bekannten Quelladressen. Gelingt es Mallory, durch Ethernet Pakete mit falschen Quelladressen, die MAC von Bob aus der Tabelle zu spülen, werden Pakete von Alice an Bob auf allen Ports sichtbar.

Der Ansatz funktioniert in der Regel nicht, da die Antwortpakete von Bob das Problem auf dem Switch sofort wieder beheben.

Beispielprogramm: macof

ARP Spoofing

Mallory sendet ARP Reply Pakete an Alice und Bob, in denen für die Netzwerkadresse vom jeweils anderen Host die Ethernetadresse von Mallory angegeben wird.

Alice und Bob werden mit diesen Paketen den ARP Cache auffrischen und daher Pakete an die Netzwerkadresse des Kommunikationspartners an den Ethernet Adapter von Mallory senden.

Mallory späht die Pakete aus und leitet sie an den echten Empfänger weiter.

Beispielprogramme: arpspoof, ettercap
Verteidigung: arpwatch
vgl. §202c StGB

Funktionen der Vermittlungsschicht

Aufgabe der Vermittlungsschicht ist es, Daten von einem Knoten zu einem (oder mehreren) Knoten des Netzes zu übertragen. Dabei können sowohl Switche als auch Router (Knoten, die die Weiterleitungsentscheidung anhand der Netzwerkadresse treffen) im Pfad liegen.

Typische Funktionen der Vermittlungsschicht sind

- **Weiterleitung** eines Paketes an den nächsten Knoten
- **Routing** eines Paketes von der Quelle zum Ziel
- **Verbindungsaufbau**, falls das Netzwerk diese Funktion erfordert.

mögliche Dienste

Die Vermittlungsschicht kann der Transportschicht eine Vielzahl von Diensten anbieten, darunter:

- Garantierte Auslieferung: Die Pakete erreichen ihr Ziel oder der Sender erhält eine Fehlermeldung.
- Garantierte Auslieferung in garantierter Zeit: Auch die Zeit bis zur Ankunft am Ziel ist garantiert.
- Garantierter Reihenfolge: Die Pakete kommen in der Reihenfolge an, in der sie gesendet wurden.
- Garantiertes Verzögerungsintervall: Die Auslieferzeit variiert nur in einem gegebenen Intervall.
- Sicherheitsdienste: Transparente gegenseitige Authentifizierung, Nachrichtenintegrität und Vertraulichkeit.

Leitungsvermittlung / (Virtual) Circuit Switching

- **Verbindungsaufbau:** Es wird zuerst eine Ende-zu-Ende Verbindung hergestellt. Diese kann Dienstgütevereinbarungen enthalten, die von jedem beteiligten Knoten einzuhalten sind.
- **Datenübertragung:** Zwischen je zwei Knoten besteht eine virtuelle Verbindung mit einer bestimmten Kennung. In Paketen wird von Routern jeweils die Kennung der Verbindung zum nächsten Knoten eingesetzt, z.B.:

Interface	Kennung	Interface	Kennung
E	15	B	12
E	7	F	13

- **Verbindungsabbau:** Die zu einer Verbindung gehörenden Einträge der Routingtabellen entlang des Pfades werden gelöscht.

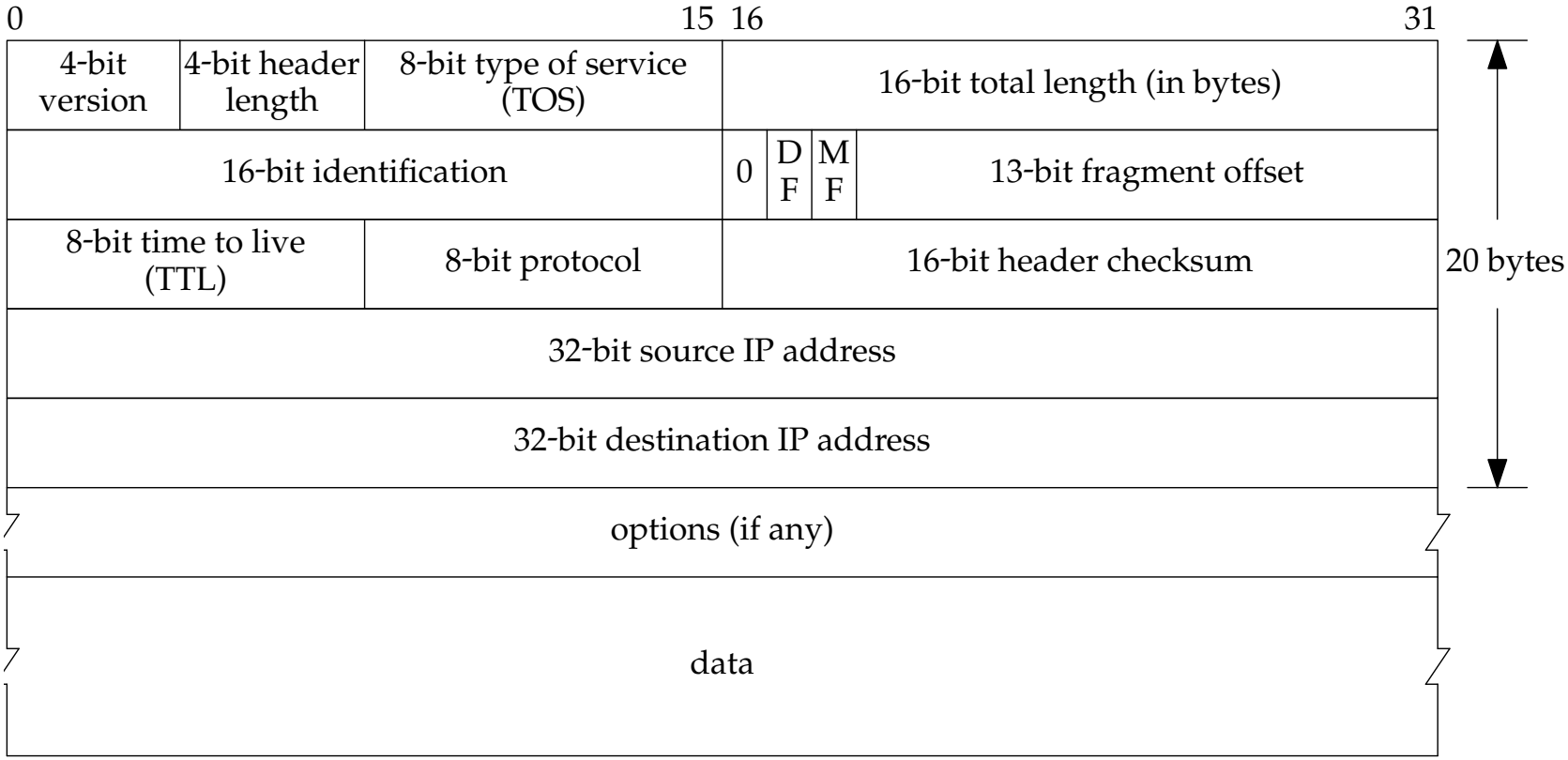
Paketvermittlung / Paket Switching

- **Verbindungsaufbau:** Gibt es nicht.
- **Datenübertragung:** Jedes Paket enthält den vollständigen Adresssatz und wird anhand der Zieladresse unabhängig von anderen Paketen weitergeleitet. Der Router stellt anhand einer Tabelle der Zieladressen fest, über welches Ausgangsinterface ein Paket weitergeleitet wird.
- **Verbindungsabbau:** Gibt es nicht.

Geschichte und Vergleich

- **Leitungsvermittlung** stammt aus der Telefonwelt
  - Endgeräte können extrem dumm sein.
  - Dienstgüteanforderungen sind leicht einzuhalten.
  - Kontrolle durch Diensteanbieter ist einfach.
  - Beispiel: POTS, ATM
- **Paketvermittlung** wurde zur robusten Verbindung von Rechnersystemen geschaffen
  - Endgeräte müssen ggfs. aus Paketen den Datenstrom rekonstruieren.
  - Dienstgüte ist kaum zu garantieren.
  - Leicht erweiterbar, da keine Dienstgüteanforderungen und wenig Kontrolle
  - Beispiel: Internet

IP Rahmen


0
15
16
31
4-bit version
4-bit header length
8-bit type of service (TOS)
16-bit total length (in bytes)
16-bit identification
0
D F
M F
13-bit fragment offset
8-bit time to live (TTL)
8-bit protocol
16-bit header checksum
20 bytes
32-bit source IP address
32-bit destination IP address
options (if any)
data

- **Version:** 4, falls IPv4, 6 für IPv6
- **Header Length:** Länge des IP Headers inklusive Optionen gezählt in 4Byte Worten, d.h. mindestens 5 (für 20Bytes). Maximale Länge des IP Headers ist demnach 60Bytes.
- **Type of Service:** Hinweis an Router, wie der Paketpfad zu optimieren ist, früher 3Bit Priorität, 4Bit Optimierungsrichtlinie (RFC1349)

  1000 Minimiere Verzögerungen
  0100 Maximiere Durchsatz
  0010 Maximiere Zuverlässigkeit
  0001 Minimiere Kosten
  1111 Maximiere Sicherheit (RFC1455)

  Abgelöst durch RFC2474 für DS, RFC3168 für ECN:
  - 2Bit Explicit Congestion Notification (00: Knoten kann kein ECN, 01/10: Knoten unterstützt ECN, 11: Link/Knoten in Überlast)
  - 6Bit Differentiated Services

- **Total Length:** Anzahl Bytes im gesamten Paket (Header + Daten), max. 64kByte
- **Identification:** Identifikationsnummer eines (noch unfragmentierten) Paketes, soll vom höheren Layer festgelegt werden.
- **Don't Fragment Flag:** Flag, das es Knoten im Pfad verbietet, das Paket zu fragmentieren
- **More Fragments Flag:** Zeigt an, daß das Paket fragmentiert ist und mindestens ein weiteres Fragment nach nach diesem kommt.
- **Fragment Offset:** Offset eines Fragments im Gesamtpaket, in 8Byte Einheiten
- **TTL:** Time To Life, jeder Router im Datenpfad dekrementiert dieses Feld, ist TTL 0 erreicht, wird eine Fehlermeldung erzeugt.

- **Protocol:** Code für den verwendeten höheren Layer, vgl. RFC1700, RFC3232 und `http://www.iana.org/assignments/protocol-numbers`
- **Header Checksum:** Prüfsumme über (nur) den Header des Paketes
- **Source Address:** Netzwerkadresse (IP) des Senders
- **Destination Address:** IP des Zieles
- **Options:** Folge von Protokolloptionen oder deren Resultate. Die Länge des Feldes ergibt sich aus der Header Length. Muß auf 4Byte Grenze aufgefüllt werden.
- **Data:** Nachricht der darüberliegenden Schicht

Fragmentierung

- Die Rahmen der Sicherungsschicht können nur Pakete einer bestimmten maximalen Größe (MTU, Maximum Transfer Unit) übertragen.
- Muß ein Knoten auf dem Weg zwischen Quelle und Ziel ein Paket übertragen, das diese Größe überschreitet, kann er
  1. das Paket auf mehrere Rahmen aufteilen (fragmentieren).
  2. eine Fehlermeldung generieren.
- Ist das **Don't Fragment Flag** gesetzt, wird eine Fehlermeldung erzeugt, andernfalls wird so fragmentiert, daß es dem nächsten Link genügt.
- Bei Bedarf kann ein Rahmen mehrfach fragmentiert werden.

Beispiel Fragmentierung

Eine Nachricht von 3000 Bytes soll in einem IP Paket ohne Optionen mittels Ethernet (MTU 1500Bytes) übertragen werden.

Wir benötigen 3 Fagmente:

Nutzdaten	Länge	Fragment Offset	DF	MF
1480 Bytes	1500	0	0	1
1480 Bytes	1500	185	0	1
40 Bytes	60	370	0	0

Bemerkung: Fragmentierung ist die Ursache für eine Reihe von Problemen in der Netzwerkinfrastruktur, z.B durch

- Senden in umgekehrter Reihenfolge.
- Senden nur eines späten Fragmentes.

Prüfsummenbildung

Die Länge eines IP Headers ist immer durch 4 teilbar.

Zur Bildung der Prüfsumme im IP Protokoll wird der Header (ohne Prüfsumme oder mit Wert 0) in 2 Byte Blöcken geschrieben, dann für jede Spalte gerade Parität gebildet.

Das Ergebnis wird in das Feld **Checksum** übernommen. Es folgt ein synthetisches Beispiel mit einem verkürzten (6 Byte) Header:

`Headerdaten:`	`10101001`	`01101010`
	`00100001`	`00101010`
Checksum `Füller:`	`00000000`	`00000000`
Checksum:	`10001000`	`01000000`

Bemerkung: Jeder Knoten, der TTL oder Optionen ändert, muß die Prüfsumme neu berechnen.

IP Optionen

Für eine Liste der IP Optionen siehe
`http://www.iana.org/assignments/ip-parameter`.

Kodierung ist für Optionen 0 und 1 ein Byte, sonst
Code, Länge, Wert.

Einige ausgewählte Optionen aus RFC791:

0x00 End of Options
0x01 No Option (Füller)
0x07 Record Route
0x83 Loose Source Route
0x89 Strict Source Route

Kodierung Loose Source Routing:

Code	Len	Ptr	n Addresses
131	$n * 4 + 3$	$4, 8, 12, \ldots$	$\ldots$

Beispiel eines IP Rahmens

```
#nc -g 127.0.0.1 10.1.195.159 12345
```

Bytes(hex)	Bedeutung
48	IPv4, 32 Bytes Header
00	Default DS, kein ECN
00 48	72 Bytes im Paket
b2 ed	Identification
40 00	Don't Fragment, Fragment Offset 0
40	Time To Live 64, Standardwert
06	Protocol: TCP
be 18	Checksum
7f 00 00 01	Source IP: 127.0.0.1
7f 00 00 01	Destination IP: 127.0.0.1

Beispiel eines IP Rahmens, Fortsetzung

IP Optionen

Bytes(hex)	Bedeutung
83	131, Loose Source Routing
0b	11 Bytes in dieser Option
04	Zeiger auf erste Adresse
0a 01 c3 9f	Erste Adresse: 10.1.195.159
0a 01 c3 9f	Zweite Adresse: 10.1.195.159
01	NOP

Netzklassen, RFC791

Der IP Adressbereich wurde eingeteilt in Subnetze, die - abhängig von ihrer Adresse - unterschiedliche Größen haben:

Klasse	Präfix	Type	Endadresse
A	0	127 Subnetze, je 24 Bit	127.255.255.255
B	10	16383 Subnetze, je 16 Bit	191.255.255.255
C	110	2097151 Subnetze, je 8 Bit	223.255.255.255
D	1110	Multicast Adressen	239.255.255.255
E	1111	reserviert	255.255.255.255

Beispiel: IP Adresse 192.168.10.1 bezeichnet eine Adresse in einem (privaten, vgl. RFC1918) Class-C Netzwerk.

Classless Inter-Domain Routing, CIDR, RFC1518

Die Einteilung in Netzklassen konnte nicht beibehalten werden, da mit dem schnellen Wachstum des Internets die Tabellen in Routern schnell nicht mehr administrierbar waren (für Details siehe RFC1519).

Die Routingentscheidung wird nicht mehr allein anhand der Zieladresse gefällt, sondern zusätzlich mit einer Netzmaske, die die Größe des zugehörigen Netzes angibt.

Die Netzmaske besteht aus 32Bit, vorne 1 für den Präfix, 0 für die Knoten im Netz. Für ein Class-C Netz ergibt sich z.B. die Netzmaske 255.255.255.0.

Schreibweise für 192.168.10.1 sind bei CIDR: 192.168.10.1/24 oder 192.168.10.1/255.255.255.0.

Beispiel CIDR

Für eine Adresse wird anhand der Routingtabelle geprüft, welcher Eintrag paßt, d.h. (Bits der Adresse) AND (Netzmaske) ergibt das Netz. Passen mehrere Einträge, wird derjenige mit der längsten Maske gewählt.

#	Destination	Gateway	Genmask	Iface
1	10.1.195.0	0.0.0.0	255.255.255.0	eth0
2	192.168.42.0	10.1.195.1	255.255.255.0	eth0
3	10.135.228.0	0.0.0.0	255.255.254.0	eth1
4	10.1.0.0	10.1.195.1	255.255.0.0	eth0
5	0.0.0.0	10.135.229.252	0.0.0.0	eth1

10.1.180.33 paßt auf Zeilen 4 und 5, gerouted wird über eth0, nächster Knoten ist 10.1.195.1.

Spezielle Adressbereiche

Adressen	Zweck	Referenz
0.0.0.0/8	Zero Addresses	RFC1700
10.0.0.0/8	Private Adressen	RFC1918
127.0.0.0/8	Loopback Address	RFC1700
169.254.0.0/16	Zeroconf, Link Local	RFC3927
172.16.0.0/12	Private Adressen	RFC1918
192.0.2.0/24	Beispielnetze/Test Domain	RFC3330
192.168.0.0/16	Private Adressen	RFC1918
198.18.0.0/15	Test Netze	RFC2544
224.0.0.0/4	IP Multicast	RFC3171
240.0.0.0/4	Reserviert	RFC1700

Network Address Translation (NAT)

Viel Router haben die Möglichkeit, Adressen im IP Header umzuschreiben.

Ziele sind:

- Rechner in privaten Netzen können Rechner im Internet erreichen
- IP Pakete können nur über diese Router ins Internet
- Eingehende Verbindungen aus dem Internet auf Rechner im privaten Netz funktinieren nur, wenn der NAT-Router entsprechend konfiguriert ist.

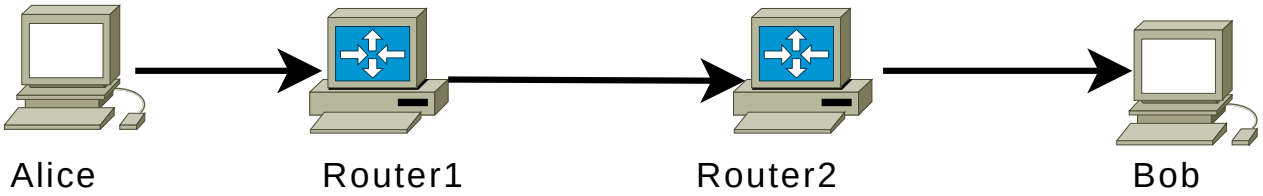
Man unterscheidet:

- **Source NAT:** Die Quelladresse wird verändert
- **Destination NAT:** Die Zieladresse wird verändert
- **Masquerading:** Mehrere Quelladressen werden hinter einer IP (oft der des Routers) versteckt.

Motivation

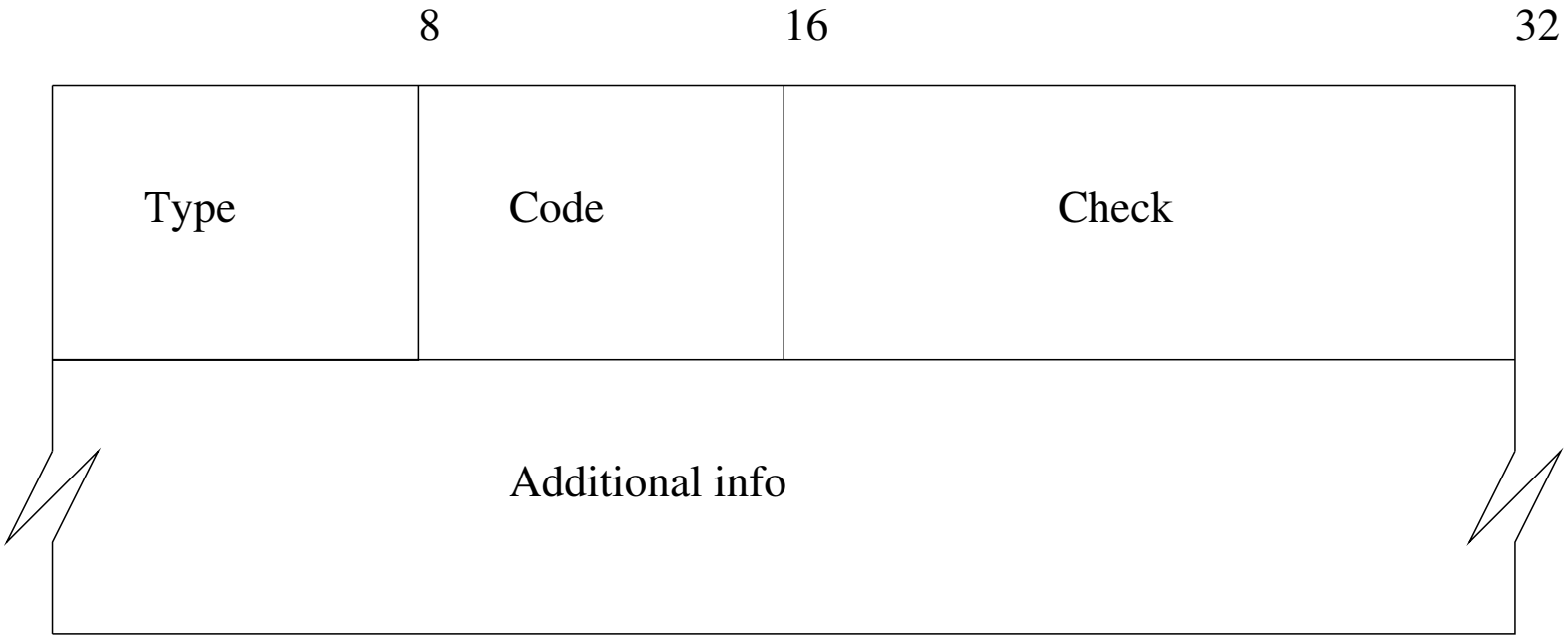
Wie kann ein Host über Probleme bei der Auslieferung von Daten informiert werden?

Beispiel:
Host Alice sendet ein Paket an Host Bob. Bob befindet sich in einem anderen Netzwerk, d.h. ist nicht über Adressen der Sicherungsschicht erreichbar.



Wie wird Alice informiert, wenn Router2 Bob nicht erreichen kann?

ICMP Rahmen



Type Typ der Nachricht, davon hängen die weiteren Felder ab.

Code Untergruppen für den jeweiligen Type

Check Prüfsumme, wie im IP Header berechnet

Rest Weitere Daten abhängig von Type und Code