IP: Internet Protocol

3.1 Einführung

IP ist das Arbeitstier unter den Protokollen der TCP/IP-Protokollsuite. Alle TCP-, UDP-, ICMP- und IGMP-Daten werden als IP-Datagramme übertragen (Abbildung 1.4). TCP/IP überrascht viele Neulinge – vor allem solche mit einem Hintergrund in X.25 oder SNA – damit, dass IP einen unzuverlässigen, verbindungslosen Datagramm-Delivery-Service bereitstellt.

Unzuverlässig bedeutet, dass es keine Garantie dafür gibt, dass ein IP-Datagramm das Ziel erreichen wird. IP stellt einen Best-Effort-Service bereit. Wenn irgendetwas schief läuft, wenn zum Beispiel einem Router die Pufferkapazität ausgeht, besitzt IP einen einfachen Fehlerbehandlungsalgorithmus: Es verwirft das Datagramm und versucht, eine ICMP-Nachricht an die Quelle zurück zu senden. Eine etwa erforderliche Zuverlässigkeit muss daher durch die höheren Schichten (z.B. TCP) bereitgestellt werden.

Der Begriff verbindungslos bedeutet, dass IP keine Statusinformationen über die aufeinander folgenden Datagramme speichert. Jedes Datagramm wird unabhängig von allen anderen Datagrammen betrachtet. Das bedeutet außerdem, dass IP-Datagramme in der falschen Reihenfolge eintreffen können. Wenn eine Quelle zwei aufeinander folgende Datagramme (erst A, dann B) an dasselbe Ziel sendet, wird jedes Datagramm unabhängig geroutet; mit anderen Worten können die Datagramm über unterschiedliche Routen geleitet werden, sodass B vor A eintrifft.

In diesem Kapitel sehen wir uns die Felder des IP-Headers kurz an und beschreiben das IP-Routing sowie Subnetting. Wir sehen uns außerdem zwei nützliche Befehle: `ifconfig` und `netstat` an. Auf eine detaillierte Besprechung einiger der Felder des IP-Headers kommen wir später zurück, weil wir dann genau untersuchen können, wie diese Felder genutzt werden. RFC 791 [Postel 1981a] liefert die offizielle Spezifikation für IP.

3.2 IP-Header

Abbildung 3.1 zeigt das Format eines IP-Datagramms. Die normale Größe des IP-Headers beträgt 20 Byte, ohne Berücksichtigung etwaiger Optionen.

Wir zeigen die Protokoll-Header in TCP/IP in Abbildung 3.1. Das signifikanteste Bit mit der Nummer 0 steht links und das am wenigsten signifikante Bit des 32-Bit-Werts mit der Nummer 31 steht rechts.

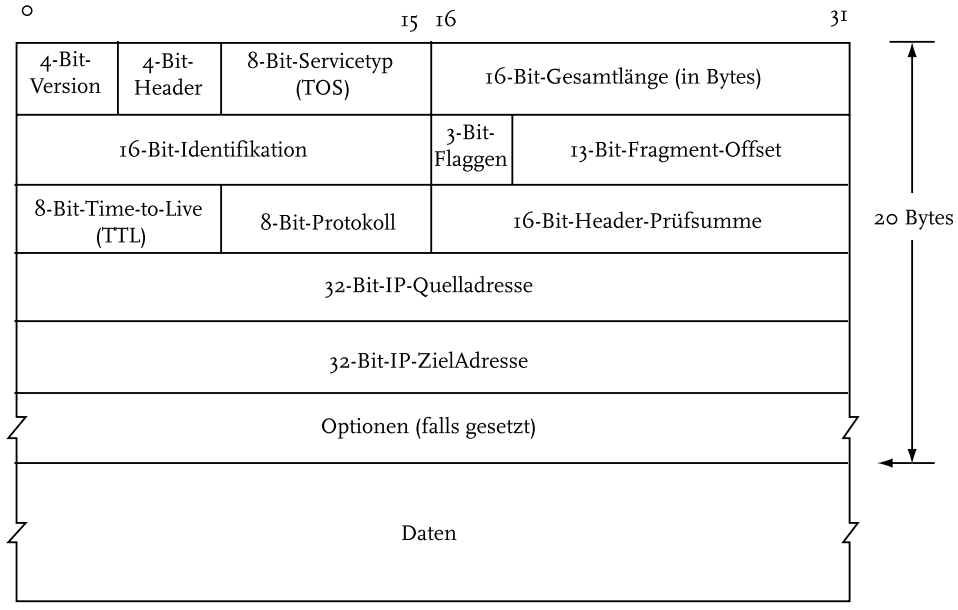

Abb. 3.1: IP-Datagramm mit den Feldern des IP-Headers

Die 4 Bytes im 32-Bit-Wert werden in der folgenden Reihenfolge übertragen: erst Bits 0-7, dann Bits 8-15, dann 16-23 und zum Schluss Bits 24-31. Man spricht in diesem Fall von einer Big-Endian-Byte-Reihenfolge; diese Byte-Reihenfolge ist für alle binären Integer der TCP/IP-Header während der Übertragung über das Netzwerk vorgeschrieben und wird deshalb als Netzwerk-Byte-Reihenfolge bezeichnet. Maschinen, die binäre Integer in anderen Formaten (wie das Little-Endian-Format) speichern, müssen die Header-Wert vor der Datenübertragung in die Netzwerk-Byte-Reihenfolge konvertieren.

Die aktuelle Protokollversion ist Version 4, sodass IP manchmal als IPv4 bezeichnet wird. In Abschnitt 3.10 werden einige Entwürfe für eine neue Version von IP vorgestellt.

Die *Header-Länge* ist die Anzahl der 32-Bit-Wörter im Header, einschließlich etwaiger Optionen. Da es sich um ein 4-Bit-Feld handelt, ist der Header auf 60 Bytes begrenzt. Wie wir in Kapitel 8 sehen werden, führt diese Beschränkung dazu, dass manche Optionen wie die Record-Route-Option heutzutage nutzlos sind. Der Standardwert für dieses Feld (wenn keine Optionen gesetzt wurden) ist 5.

Das Type-of-Service-(TOS-)Feld umfasst ein Präzedenzfeld mit 3 Bits, das heute ignoriert wird, 4 TOS-Bits sowie ein unbenutztes Bit, das 0 enthalten muss. Die 4 TOS-Bits stehen für: Verzögerung minimieren, Durchsatz maximieren, Zuverlässigkeit maximieren und Kosten minimieren.

Von diesen 4 Bits darf nur eines gesetzt sein. Wenn alle Bits nicht gesetzt sind, wird der normale Service angenommen. RFC 1340 [Reynolds und Postel 1992] spezifizieren wie diese Bits durch alle Standardanwendungen zu setzen sind. RFC 1349 [Almquist 1992] enthält einige Korrekturen für dieses RFC sowie eine detaillierte Beschreibung des TOS-Features.

Abbildung 3.2 zeigt die für das TOS-Feld empfohlenen Werte für verschiedene Applikationen. In der letzten Spalte wird der hexadezimale Wert angezeigt, da wir diesen weiter unten im Text in der `tcpdump`-Ausgabe sehen werden.

Anwendung	Verzögerung minimieren	Durchsatz maximieren	Zuverlässigkeit maximieren	Kosten minimieren	Hex-Wert
Telnet/Rlogin	1	0	0	0	0x10
FTP					
Control	1	0	0	0	0x10
Daten	0	1	0	0	0x08
beliebige Massendaten	0	1	0	0	0x08
TFTP	1	0	0	0	0x10
SMTP					
Befehlsphase	1	0	0	0	0x10
Datenphase	0	1	0	0	0x08
DNS					
UDP-Query	1	0	0	0	0x10
TCP-Query	0	0	0	0	0x00
Zonentransfer	0	1	0	0	0x08
ICMP					
Fehler	0	0	0	0	0x00
Abfrage	0	0	0	0	0x00
jedes IGP	0	0	1	0	0x04
SNMP	0	0	1	0	0x04
BOOTP	0	0	0	0	0x00
NNTP	0	0	0	1	0x02

Abb. 3.2: Empfohlene Werte für das Type-of-Service-Feld

Für Applikationen mit interaktivem Login wie Telnet und Rlogin ist eine minimale Verzögerung wichtig, da diese Anwendungen interaktiv von Usern benutzt werden, die kleine Datenmengen übertragen wollen. Für den Filetransfer mit FTP ist im Gegensatz dazu der maximale Durchsatz wichtig. Für Netzwerkmanagement- (SNMP) und Routingprotokolle ist die maximale Zuverlässigkeit wesentlich. Usenet News (NNTP) ist das einzige der gezeigten Protokolle, bei dem die Kosten reduziert werden sollen.

Das TOS-Feature wird von Mehrzahl der heutigen TCP/IP-Implementierungen nicht unterstützt, obwohl neuere Systeme seit 4.3BSD Reno es einsetzen. Darüber

hinaus können neue Routing-Protokolle wie beispielsweise OSPF und IS-IS Routing-Entscheidungen auf Basis dieses Feldes treffen[1].

Das Feld Gesamtlänge enthält die Gesamtlänge des IP-Datagramms in Bytes. Mit Hilfe dieses sowie des Header-Längenfeldes können wir erkennen, wo der Datenabschnitt des IP-Datagramms beginnt, und dessen Länge feststellen. Da es sich um 16-Bit-Feld handelt, beträgt die maximale Größe eines IP-Datagramms 65535 Bytes (aus Abbildung 2.5 können Sie sich in Erinnerung rufen, dass Hyperchannel eine MTU von 65535 hat. Mit anderen Worten gibt es in Wirklichkeit keine MTU – es wird einfach das größtmögliche IP-Datagramm benutzt). Dieses Feld ändert sich außerdem, wenn ein Datagramm fragmentiert wird, wie wir in Abschnitt 11.5 erfahren werden.

Obwohl es möglich ist, ein IP-Datagramm mit 65535 Bytes zu senden, wird dieses von den meisten Link-Layers fragmentiert. Darüber hinaus muss ein Host keine Datagramme entgegennehmen, die größer als 576 Bytes sind. TCP teilt die Userdaten auf, sodass diese Beschränkung in der Regel keine Auswirkung auf TCP hat. In weiteren Kapiteln begegnen uns zahlreiche Beispiele für UDP-Applikationen (RIP, TFTP, BOOTP, DNS und SNMP), die sich auf 512 Bytes an Userdaten beschränken, um die Grenze von 576 Bytes einhalten zu können. Aber realistisch betrachtet ermöglichen die meisten heutigen Implementierungen (vor allem solche, die das *Network File System* – NFS unterstützen), IP-Datagramme mit etwas über 8192 Bytes.

Das Feld Gesamtlänge ist im IP-Header erforderlich, weil manche Netzwerktypen (z.B. Ethernet) kleinere Rahmen auf eine Mindestgröße auspolstern. Obwohl die Mindestgröße eines Ethernet-Rahmens 46 Bytes beträgt (Abbildung 2.1), kann ein IP-Datagramm noch kleiner sein. Wenn das Feld Gesamtlänge nicht existieren würde, wüsste die IP-Schicht nicht, wie viel von einem 46-Byte-Ethernet-Frame tatsächlich durch ein IP-Datagramm belegt ist.

Das Feld Identifikation ermöglicht die eindeutige Kennzeichnung jedes von einem Host übertragenen Datagramms. Normalerweise wird jedes Mal, wenn ein Datagramm gesendet wird, das Feld um eins hochgezählt. In Abschnitt 11.5 kommen wir auf das Feld zurück, wenn wir die Fragmentierung und Reassemblierung

1 In Abschnitt 2.10 haben wir erwähnt, dass SLIP-Treiber normalerweise Type-of-Service-Funktionalität bieten, sodass der interaktive Traffic vor Massendaten bevorzugt behandelt werden kann. Da die meisten Implementierungen das TOS-Feld nicht nutzen, werden Warteschlangen ad hoc durch SLIP angelegt, wobei der Treiber das Protokollfeld (um festzustellen, ob es sich um eine TCP-Segment handelt) und dann die TCP-Quell- und Zielportnummern untersucht, um festzustellen, ob die Portnummer zu einem interaktiven Service gehört. Ein Kommentar in einem Treiber vermerkt, dass dieser »disgusting hack« (also diese hässliche Quick-and-Dirty-Lösung) notwendig ist, weil die meisten Implementierungen nicht zulassen, dass das TOS-Feld gesetzt wird.