Address erkennt, so wird die Nachricht verarbeitet, andernfalls ignoriert (jedoch ist es legitim die Sender Hardware Address und Sender Protocol Address in den ARP-Cache zu übernehmen).

work3.local hat diese Nachricht empfangen und sendet eine Antwort. Dazu geht der Frame zurück an 01:23:9a:4b:8a:2b (Linklayer), Sender Hardware Address enthält 01:26:da:52:fb:6e (Ethernet-Adresse von work3.local), Sender Protocol Address enthält 192.168.1.43 (IP-Adresse von work3.local). Die Target Hardware Address ist hierbei 01:23:9a:4b:8a:2b und die Target Protocol Address ist 192.168.1.41. Der Wert für Operation wird auf 2 für REPLY gesetzt.

work3.local merkt sich praktischerweise die IP-Adresse und Hardware-Adresse von work1.local in seinem ARP-Cache, damit für weitere Anfragen keine Requests nötig werden. Wenn work1.local das Reply sieht, so fügt auch er diese Informationen seinem Cache hinzu.

Die Einträge im ARP-Cache haben üblicherweise einen Zeitstempel, damit sie nach einer gewissen Zeit verworfen werden können. Andernfalls würde eine Änderung der Zuordnung nur dadurch auffallen, daß ein Rechner mit der Ziel-IP-Adresse zwar am Netz ist, aber keine Frames erhält, weil jemand noch an die alte Hardware-Adresse sendet (diese Frames werden dann ignoriert, sofern nicht jemand anderes die alte Hardware-Adresse zugeteilt bekommt).

Dieses Beispiel ist nun das Senden innerhalb eines Netzwerkabschnitts. ARP existiert auch nur auf dieser Ebene, ARP-Frames werden nie über Routergrenzen hinweg verbreitet; außerhalb „ihres“ Netzes sind Hardwareadressen auch nutzlos.

Das Schicksal eines IP-Datagramms, das über das Internet reist, ist aber etwas verwickelter, denn es passiert in der Regel viele lokale Netze. Hierbei löst jeder Router ein ARP-Request aus, falls er die Hardwareadresse des nächsten Hops nicht kennt.

3.5. IP – das Internet Protocol

3.5.1. Überblick über das Internet Protocol

Das Internet Protocol (IP) ist in RFC 791 [17] beschrieben. Es dient der Internetschicht (vgl. Abschnitt 1.3) zum Transport der Daten von einem Host zum anderen. Lange Pakete können hierbei in kürzere Teile zerlegt (*fragmentiert*) werden (immer dann nötig, wenn ein Datagramm länger als die Framelänge der darunterligenden Übertragungsschicht ist, bei Ethernet z.B. 1500 Oktetts). Dieser Mechanismus wird in Abschnitt 3.5.2 ausführlich behandelt. Jedes IP-Paket (oder *Datagramm*) beginnt mit einem Header, gefolgt von Nutzdaten, die weitere Header wie z.B. TCP (4.1.1) oder UDP (4.2), enthalten können. Der Aufbau eines solchen Headers ist in Abbildung 3.2 dargestellt.

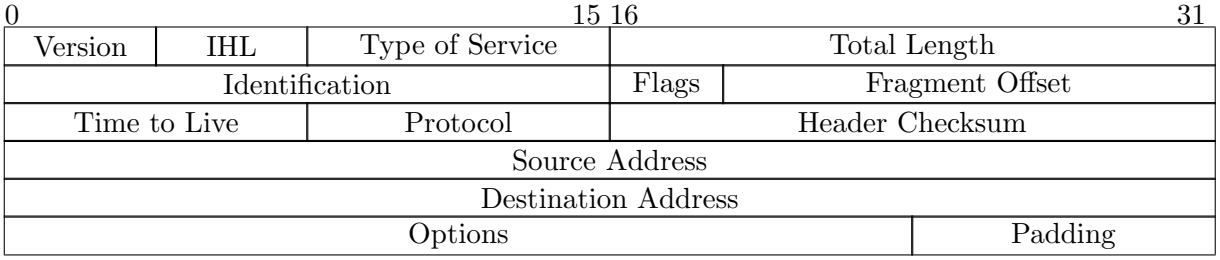


Abbildung 3.2.: Internet Protocol (Version 4) Header

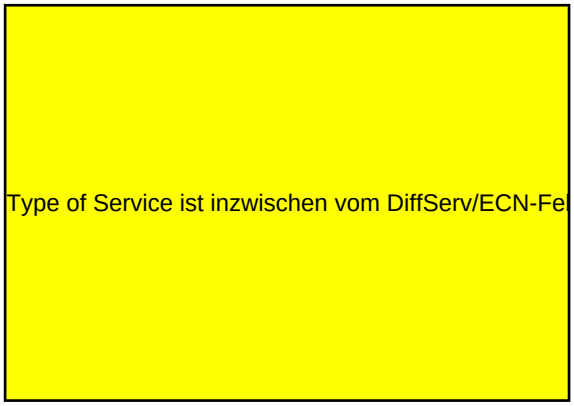
Version
Die Versionsnummer, bei IPv4 ist dies 4.

IHL
IHL steht für *Internet Header Length* und bezeichnet die Länge des IP-Headers in 32-Bit-Worten. Anhand diesen Eintrags kann erkannt werden, ab welchem Offset die Nutzlast beginnt. Die Mindestgröße eines IP-Headers ist 20 Oktetts, also ist der kleinste hier zulässige Wert 5. Auf Grund der Begrenzung auf 4 Bit kann der größte Header also $2^4 * 4 = 64$ Oktetts lang sein.

Type of Service
Dieses Headerfeld bestimmt in abstrakter Form die Qualität des angebotenen Dienstes. Router können anhand dieser Parameter zum Beispiel die Route wählen, die das Datagramm zum Ziel einschlagen soll.

Bits	Funktion	Bedeutung
0-2	Dringlichkeit	siehe Tabelle unten
3	Verzögerung	0 → normal, 1 → gering
4	Durchsatz	0 → normal, 1 → hoch
5	Zuverlässigkeit	0 → normal, 1 → hoch
6-7	reserviert	sollte auf 0 gesetzt werden

111	Network Control
110	Internetwork Control
101	CRITIC/ECP
100	Flash Override
011	Flash
010	Immediate
001	Priority
000	Routine



Diese Einträge sind nur als „Wünsche“ zu verstehen, ob und wie ein Router oder Host darauf reagiert ist völlig ihm selbst überlassen.

Total Length
Dieses Feld enthält die Gesamtlänge des Pakets, in Oktetts. Durch die Breite von 16 Bit ist die maximale Größe eines IP-Datagramms auf $2^{16} = 64KByte$ begrenzt.

Identification
Anhand diesen Felds können die Fragmente eines Pakets erfolgreich zusammengesetzt werden. Jedes Paket sollte eine einzigartige ID haben (was bei 16 Bit natürlich mit der Zeit zu Dopplungen führt).

Flags
Das Flag-Feld enthält drei mögliche Flags:

Bit	Name	Bedeutung
0	reserviert	muß auf 0 stehen
1	Don’t Fragment	verhindert das Fragmentieren des Pakets
2	More Fragments	es folgen weitere Fragmente

Ist das ”Don’t Fragment”-Bit gesetzt, so wird ein Datagramm entweder unfragmentiert weitergesendet, oder – wenn es zu „dick“ für die Leitung ist – weggeworfen. Hierbei wird eine ICMP-Nachricht mit dem Code für ”Destination unreachable” an den Absender des Datagramms gesendet (vgl. Abschnitt 4.3.1).

Das ”More Fragments”-Bit ist in jedem fragmentierten Paket auf 1 gesetzt, das noch Folgefragmente hat. Das letzte Fragment hat dieses Bit also auf 0 stehen.

Fragment Offset
Dieser Wert ist der Offset, an dem aus Sicht des Gesamtdatagramms das vorliegende Fragment

eingefügt werden soll. Der Wert ist mit 8 zu multiplizieren, um auf den Offset in Oktetts zu kommen.

Time to Live
Dieses Feld enthält einen Zähler, der bei jedem Hop um eins dekrementiert wird. Erreicht er 0, so wird das Datagramm verworfen. Der Hintergedanke ist das endlose Kreisen von Datagrammen zu verhindern (z.B. auf Grund einer falschen Routing-Information).

Protocol
Die über der Internetschicht liegende Transportschicht muß wissen, welches Protokoll verwendet wird. Diese Information wird im Protocol-Feld gespeichert. Werte für die einzelnen Protokolle sind (auszugsweise):

1	ICMP
2	IGMP
6	TCP
17	UDP

Die komplette Liste findet sich in [15].

Header Checksum
Dieses Feld enthält eine Checksumme über das komplette Datagramm. Die Checksumme wird folgendermaßen gebildet:

- Das Checksummen-Feld wird auf 0 gesetzt
- Die Daten werden als 16-Bit-Worte aufgefasst
- Von jedem 16-Bit-Wort wird das (Einer-)Komplement gebildet
- Diese Komplemente werden aufaddiert
- Das Komplement der Summe ist die Checksumme

Der Hintergedanke ist die Einfachheit beim Überprüfen: es werden wieder alle 16-Bit-Worte aufaddiert. Wenn die Summe nur aus 1-Bits besteht (also das Feld 0xffff als Wert hat), dann sind die Daten in Ordnung.

Eine sehr ausführliche Erklärung mit Beispielen findet sich in [18].

Source Address
Die Quelladresse des Datagramms.

Destination Address
Die Zieladresse des Datagramms.

Options
Das Internet Protocol bietet eine Reihe von Optionen teils variabler Länge.

Eine Option kann entweder ein Oktett lang sein, oder aber aus einem Oktett für die Kennung, einem für die Längenangabe, und dann den eigentlichen Daten bestehen. Ist die Gesamtlänge des IP-Headers kein Vielfaches von 32 Bit (wie es das *IHL*-Feld fordert), so müssen entsprechend viele Nullen eingefügt werden (*Padding*).

Die einzelnen Optionen werden in Abschnitt 3.5.3 vorgestellt.