

Ben Mohr 1605057

IPv4

Header

1

Kategorie (IPv4 Header)	Wert aus dem Paket
Version	4
Header Length	20 Bytes (5 × 4 Byte)
Type of Service (DSCP / ECN)	0x00 (DSCP: CS0, ECN: Not-ECT)
Total Length	129
Identification	0x2b5f (dezimal: 11103)
Flags	0x2 (Don't Fragment gesetzt)
Fragment Offset	0
Time to Live (TTL)	64
Protocol	6 (TCP)
Header Checksum	0x8977 (nicht validiert)
Source IP Address	192.168.2.1
Destination IP Address	192.168.2.79

Kategorie (TCP Header) Wert aus dem Paket

Source Port 53 (DNS)

Destination Port 29014

Sequence Number 2 (relativ), raw: 554895324

Acknowledgment Number 35 (relativ), raw: 3518966551

Header Length 20 Bytes (5 × 4 Byte)

Flags PSH, ACK

Window Size 229 (skalierte Größe: 29312)

Checksum 0xd01f (nicht überprüft)

Urgent Pointer 0

Optionen keine vorhanden (wegen Header-Länge 20 Byte)

Kategorie (UDP Header) Wert aus dem Paket

Source Port	12177
Destination Port	12177
Length	114
Checksum	0x5c31 (nicht überprüft)

2

Die IP-Adresse **103.161.122.83/18** besteht aus der Adresse selbst (103.161.122.83) und dem Präfix **/18**, das angibt, dass die ersten 18 Bits zur Netzwerkadresse gehören. Daraus ergibt sich die Subnetzmaske **255.255.192.0**. Um die **Netzwerkadresse** zu ermitteln, wird ein bitweises UND zwischen der IP-Adresse und der Subnetzmaske durchgeführt, was zu **103.161.64.0** führt. Die **Broadcastadresse** ergibt sich, indem alle Hostbits auf 1 gesetzt werden, was den Wert **103.161.127.255** ergibt. Die IP-Adresse **103.161.193.83/18** liegt hingegen im Netzwerk **103.161.192.0** und gehört somit **nicht** zum selben Subnetz wie 103.161.122.83/18.

3

Eine direkte Kommunikation zwischen einem UDP- und einem TCP-Chatprogramm ist nicht möglich, da TCP verbindungsorientiert arbeitet (mit Handshake, Zustellgarantie und Reihenfolge), während UDP verbindungslos ist und Daten ohne Bestätigung verschickt. Ein UDP-Client sendet einfach Pakete, die ein TCP-Server ignoriert, da er auf eine Verbindung wartet. Umgekehrt kann ein TCP-Client keine Verbindung zu einem UDP-Server aufbauen.

Lösungsmöglichkeiten:

- Beide Programme auf dasselbe Protokoll (TCP oder UDP) umstellen
- Optional: Protokoll-Übersetzer (Gateway), der UDP ↔ TCP umwandelt – jedoch aufwändig in Umsetzung und Verwaltung

4

Um die Zusammenarbeit verschiedener Implementierungen sicherzustellen, wurde in einem gemeinsamen Google-Dokument ein einheitliches Nachrichtenprotokoll definiert. Dieses legt fest, welche Befehle sowohl im TCP- als auch im UDP-Chatprogramm unterstützt werden und wie sie aufgebaut sind. Dadurch wird garantiert, dass alle Programme dieselben Formate verwenden und sich gegenseitig korrekt interpretieren können.

Beispielsweise erfolgt das Senden einer Nachricht über TCP mit `send <Name> <Nachricht>` und über UDP mit `send <Name> <Nachricht>`, das Broadcasting mit `broadcast` bzw. `send_all`, und die Clientliste wird mit `list` bzw. `peers` angefordert. Auch automatische Antworten auf definierte Fragen wie „Was ist deine IP-Adresse?“ oder „Welche Rechnernetze HA war das?“ wurden gemeinsam abgestimmt.

Diese Spezifikation ermöglicht die Interoperabilität zwischen den verschiedenen Chatprogrammen der Kommiliton:innen – unabhängig davon, wie deren konkrete Implementierung im Hintergrund aussieht.