Aufgabe 1

Die Zuordnung wird einem beim IPv4-Header von Wireshark bereits abgenommen. Lediglich das Feld 'Type of Service' wird hier unter einem anderen Namen aufgeführt: 'Differentiated Services Field'. Der Rest ist bereits mit dem korrekten Begriff assoziiert.

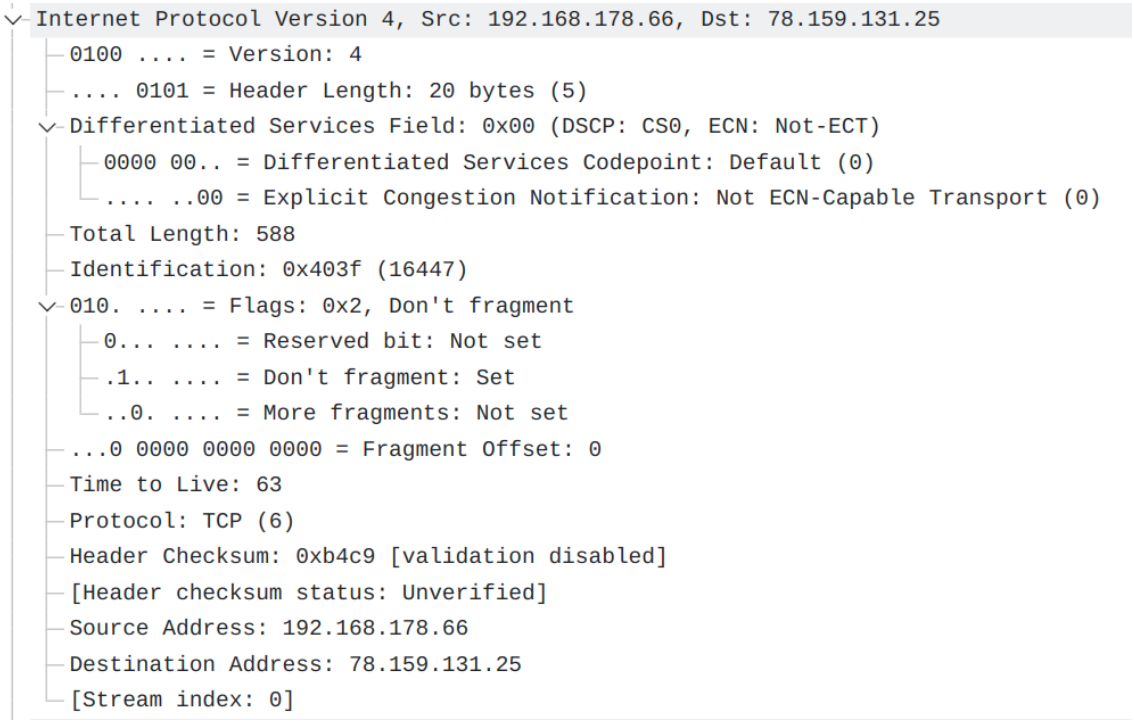

Abbildung 1: Beliebiges IPv4 Paket

IM UDP-Header sind auch hier schon richtig beschriftet nur der Absender, der Empfänger, die Länge und die Prüfsumme enthalten.

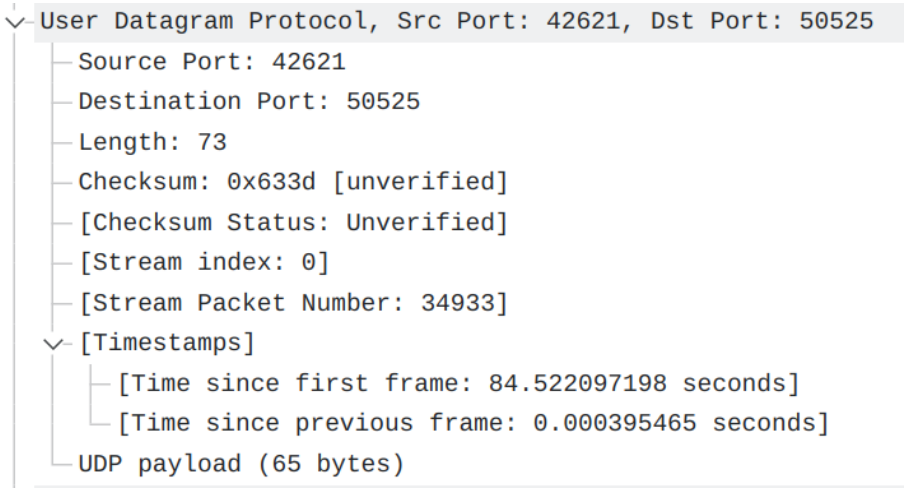

Abbildung 2: UDP Paket

Im Vergleich zum UDP-Header sind im TCP-Header zusätzlich 'Sequence Number', 'Ackknowlegment Number', 'Data Offset', 'Urgent Pointer', 'Reserved', 'Flags', 'Window' und 'Options' enthalten.

Abbildung 3: TCP Paket

Aufgabe 2

Die 103.161.122.83 ist die IPv4-Adresse und die 18 gibt die Subnetzmaske an.

/18 ist die Präfixlänge. Sie gibt an, dass die ersten 18 Bits zur Netzwerkadresse gehören. Der Rest ist für Hostadressen innerhalb dieses Netzwerks reserviert:

- Binär: 11111111.11111111.11000000.00000000
- Dezimal: 255.255.192.0

Die Netzwerkadresse ist die Bitweise Verundung der Subnetzmaske und der IPv4-Adresse.:

- 103.161.122.83 AND 255.255.192.0 = 103.161.64.0

Die Broadcast-Adresse ist die Bitweise Veroderung der Subnetzmaske und der IPv4-Adresse:

- 103.161.122.83 OR 255.255.192.0 = 103.161.127.255

Die Netzwerkadresse zu 103.161.193.83/18 können wir auf die gleiche Weise ermitteln:

- 103.161.193.83 AND 255.255.192.0 = 103.161.192.0

Da die beiden Netzadressen unterschiedlich sind folgt, dass die beiden Adressen nicht im selben Netzwerk liegen.

Aufgabe 3

Bei dem Versuch mit einem Kommilitonen einen Kommunikation der Programme herzustellen, ist an vielen Stellen aufgefallen, dass wir die Inhalte der Pakete unterschiedlich parsen, beziehungsweise anders interpretieren. Wir haben Anpassungen an wegen der jeweils anderen Implementierung machen müssen um eine funktionierende Kommunikation aufbauen zu können. Einen Einigung auf eine einheitliche Kodierung was dafür essentiell. Mit den entsprechenden Änderungen war es uns möglich über UDP zu kommunizieren. Diw TCP-Chatprogramme haben wir nicht zu einer beidseitigen Kommunikation bekommen.

Aufgabe 4

Ich habe als Basis für die neuen Funktionen meine Abgabe für das Blatt 3 genutzt und darauf aufgebaut.

TCP_Chat.py und UDP_Chat.py enthalten bereits die Funktionen:

- **send :** in Form von der bereits existierenden Funktion 'send to'
- **list/peers :** in Form von der bereits existierenden Funktion 'get contacts'
- **stop/exit :** in Form von der bereits existierenden Funktion 'stop'

Noch zu implementieren waren:

- **Was ist deine IP-Adresse?**→ Empfänger sendet IP-Adresse
- **Wie viel Uhr haben wir?** → Empfänger sendet Systemzeit
- **Welche Rechnernetze HA war das?** → Empfängers sendet vordefinierte Nachricht: "4. HA, Aufgabe 4"
- **broadcast/send all <Nachricht>** → "Nachricht an alle Clients"

Zusätzlich war noch **register** für UDP_Chat.py zu implementieren.

Die automatisierten Antworten habe ich nur für bereits bekannte Kantakte zugelassen.