

## Aufgabe 1:

```
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.178.43, Dst: 65.108.120.159
 0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  ▾ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    0000 00.. = Differentiated Services Codepoint: Default (0)
    .... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
  Total Length: 297
  Identification: 0x79a8 (31144)
  ▾ 010. .... = Flags: 0x2, Don't fragment
    0... .... = Reserved bit: Not set
    .1.. .... = Don't fragment: Set
    ..0. .... = More fragments: Not set
    ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 128
  Protocol: TCP (6)
  Header Checksum: 0x5347 [validation disabled]
  [Header checksum status: Unverified]
  Source Address: 192.168.178.43
  Destination Address: 65.108.120.159
```

Dies ist ein beliebiges IPv4-Paket, welches den 20 byte header enthält.

In diesem Fall ist die Version = 4

Header length = 20 bytes

Type of service = 0x00 nicht spezifiziert

Total length = 297

Identification = 31144

F = Don't fragment

Fragment offset = 0

TTL = 128

Header Checksum = 0x5347

Source Address = 192.168.178.43

Destination Address = 65.108.120.159

```
▾ User Datagram Protocol, Src Port: 57621, Dst Port: 57621
  Source Port: 57621
  Destination Port: 57621
  Length: 52
  Checksum: 0xf265 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 6]
  ▸ [Timestamps]
  UDP payload (44 bytes)
```

Der UDP Header besteht nur aus einem Source Port = 57621, einem Destination Port = 57621, der Length = 52 und der Checksum = 0xf265.

```
Transmission Control Protocol, Src Port: 49413, Dst Port: 443, Seq: 1, Ack: 1, Len: 1
  Source Port: 49413
  Destination Port: 443
  [Stream index: 18]
  [Conversation completeness: Incomplete (12)]
  [TCP Segment Len: 1]
  Sequence Number: 1 (relative sequence number)
  Sequence Number (raw): 3753815369
  [Next Sequence Number: 2 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 2536014127
  0101 .... = Header Length: 20 bytes (5)
  Flags: 0x010 (ACK)
  Window: 1025
  [Calculated window size: 1025]
  [Window size scaling factor: -1 (unknown)]
  Checksum: 0x3a74 [unverified]
  [Checksum Status: Unverified]
  Urgent Pointer: 0
  [Timestamps]
  [SEQ/ACK analysis]
  TCP payload (1 byte)
  [Reassembled PDU in frame: 26221]
  TCP segment data (1 byte)
```

Der TCP-header beinhaltet:

Source Port: 49413

Destination Port: 443

Sequence Number 1: Position der Daten im Datenstrom

Acknowledgement Number: 1, Bestätigungsnummer

Header length = 10 bytes

Flags:

```
Flags: 0x010 (ACK)
  000. .... = Reserved: Not set
  ...0 .... = Accurate ECN: Not set
  .... 0... = Congestion Window Reduced: Not set
  .... .0.. = ECN-Echo: Not set
  .... ..0. = Urgent: Not set
  .... ...1 = Acknowledgment: Set
  .... .... 0... = Push: Not set
  .... .... .0.. = Reset: Not set
  .... .... ..0. = Syn: Not set
  .... .... ...0 = Fin: Not set
  [TCP Flags: .....A....]
Window: 1025
```

Window Size = 1025 Gibt an wie viele Bytes ab der Acknowledgement Number empfangen werden können.

Checksum: 0x3a74

Urgent Pointer: 0 Zeigt an, dass die im Urgent Pointer Feld angegebene Anzahl von Bytes vorrangig behandelt werden soll.

## Aufgabe 2:

CIDR: Die 103.161.122.83/18 nutzt Classless Inter domain Routing, wobei die 18 angibt, wie viele Bites Netzwerkanteil sind und wie viele Hostanteil. In diesem Fall wäre 18 bits Netzwerkanteil und  $32-18 = 14$  Bits Hostanteil.

Die Subnetzmaske gibt dort 1 an, wo der Netzwerkanteil ist und 0, wo der Hostanteil ist, also in diesem Fall: 11111111.11111111.11000000.00000000 oder 255.255.192.0 in Dezimal.

Die Netzwerkadresse ermitteln wir dann mittels einer logischen UND Verknüpfung mit der Subnetzmaske:

IP:Adresse: 103.161.122.83 == 01100111.10100001.01111010.01010011

UND

Subnetzmaske: 255.255.192.0 == 11111111.11111111.11000000.00000000

Netzwerkadresse: 103.161.64.0 == 01100111.10100001.01000000.00000000

Die Broadcast Adresse bekommen wir, indem wir die Bits des Host Anteils der Netzwerkadresse auf 1 setzen.

Broadcast-Adresse: 103.161.127.255 == 01100111.10100001.01111111.11111111

Liegen 103.161.122.83/18 und 103.161.193.83/18 im selben Netz?

Dazu Berechnung der Netzwerkadresse mit der Subnetzmaske.

103.161.122.83/18 -> 103.161.64.0

103.161.193.83/18 == 01100111.10100001.11000001.01010011

Daraus die Netzwerkadresse mit der Subnetzmaske

01100111.10100001.11000000.00000000 == 103.161.192.0

Ein Vergleich der Netzwerkadressen

1. 103.161.64.0
2. 103.161.192.0

Ergibt, dass es unterschiedliche Netze sind, also liegen sie nicht im selben Netz.

Aufgabe 3: Beim Kommunizieren mit verschiedenen Implementierungen von Kommunikation können folgende Probleme auftreten. Die Schnittstelle zur Kommunikation ist anders definiert, d.h. eine Kommunikation muss über andere Commands erfolgen, weswegen der Code überall angepasst werden muss. Deswegen ist ein Standard zum Senden/Empfangen am besten. Die genaue Logik und Datenverarbeitung sind relativ egal, solange die nach außen sichtbare Schnittstelle einheitlich ist. Bei lokal laufenden Programmen kommt es nicht zu Netzwerkproblemen aufgrund der Firewall o.ä., aber sobald die Kommunikation über mehr als einen Rechner geschehen soll, bzw. über das Internet kann es zu Problemen kommen.