Rechnernetze -Übungsblatt 4 09.06.2024
Roman Schander
1598510

AUFGABE 1

Wireshark IPv4-Paket

```
Frame 16: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface en0, id 0
Ethernet II, Src: AVM_e7:86:7c (c0:25:06:e7:86:7c), Dst: Apple_32:33:73 (3c:06:30:32:33:73)
Internet Protocol Version 4, Src: 136.199.189.116, Dst: 192.168.178.45
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
    Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 40
    Identification: 0x3c28 (15400)
    010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 115
    Protocol: TCP (6)
    Header Checksum: 0x1296 [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 136.199.189.116
    Destination Address: 192.168.178.45
Transmission Control Protocol, Src Port: 443, Dst Port: 65311, Seq: 1, Ack: 33, Len: 0
```

In dem Wireshark IPv4 Paket sind folgende Elemente des Headers zu sehen:

- **Version**

Die Version trägt den binären Wert 0100 für Version 4

- **Header Length**

Die Länge des Headers (hier 20 Byte, 0101) wird in Paaren von 4-Bytes berechnet (4*5=20)

- **Differentiated Services Field**

Ist mit dem Type of Service Feld zu vergleichen und gibt die Art des Service an

- **Total Length**

Die Gesamtlänge des IP-Pakets (Header+Daten), hier 40 Bytes

- **Identification, Flags, Offset**

Die ID des Pakets und Kennzeichnung für Fragmentierung. Dieses Paket wird nicht Fragmentiert. Das Fragmentation-Offset ist auf 0 gesetzt, da hier nicht fragmentiert wird.

- **Time to Live (TTL)**

Die Anzahl der maximalen Weiterleitungen (Hops) welche das Paket erreichen kann. Hier auf 115 limitiert.

- **Protocol**

Angabe des Protokolls welches übergeordnet verwendet wird. Hier TCP

- **Header Checksum**

Wird verwendet, um nach Fehler im Header zu finden. Der wert 0x1296 gibt an, dass diese Fehlerprüfung deaktiviert ist.

- **Source, Destination Address**

Die Sender und Empfänger Adresse für das IPv4-Paket

Rechnernetze -Übungsblatt 4 09.06.2024
Roman Schander
1598510

Wireshark UDP-Paket

```
Frame 1: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface en0, id 0
Ethernet II, Src: ASRockIncorp_8a:74:57 (d0:50:99:8a:74:57), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 192.168.178.125, Dst: 192.168.178.255
User Datagram Protocol, Src Port: 57621, Dst Port: 57621
    Source Port: 57621
    Destination Port: 57621
    Length: 52
    Checksum: 0x7f4f [unverified]
    [Checksum Status: Unverified]
   UDP payload (44 bytes)
Data (44 bytes)
```

- **Source, Destination Port**

Portnummern der Anwendungen oder Prozesse, wovon die Nachricht ausgeht bzw. eingeht. Hier jeweils 57621

- **Length**

Länge des versendeten Pakets (Header und Daten) in Byte. Hier 52

- **Checksum**

Optionale Prüfsumme welche erneut der Fehlererkennung dient.

Wireshark TCP-Paket

```
Frame 7: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface en0, id 0
Ethernet II, Src: Apple_32:33:73 (3c:06:30:32:33:73), Dst: AVM_e7:86:7c (c0:25:06:e7:86:7c)
Internet Protocol Version 4, Src: 192.168.178.45, Dst: 136.199.189.116
Transmission Control Protocol, Src Port: 65310, Dst Port: 443, Seq: 32, Ack: 1, Len: 0
    Source Port: 65310
    Destination Port: 443
    [Conversation completeness: Incomplete (28)]
    [TCP Segment Len: 0]
    Sequence Number: 32    (relative sequence number)
    Sequence Number (raw): 1636840499
    [Next Sequence Number: 33    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 1781330748
    0101 .... = Header Length: 20 bytes (5)
    Flags: 0x011 (FIN, ACK)
    Window: 4096
    [Calculated window size: 4096]
    [Window size scaling factor: -1 (unknown)]
    Checksum: 0xf2ba [unverified]
    [Checksum Status: Unverified]
    Urgent Pointer: 0
```

- **Source, Destination Port**

Portnummern der Anwendungen oder Prozesse, wovon die Nachricht ausgeht bzw. eingeht. Hier 65310 und 443.

- **Sequence Number**

Sequenznummer welche zur Sortierung der Pakete dient. Dies gewährleistet dass die Pakete beim Empfänger korrekt zusammengefügt werden können, unabhängig davon in welcher Reihenfolge diese verschickt wurden bzw. empfangen wurden.

- **Acknowledgement Number**

Sequenznummer, welche der Absender des TCP-Segments als nächstes erwartet.

- **Header Length**

Länge des Headers. Hier 20 Bytes

- **Flags**

Verschiedene Flags welche mögliche Zustände des TCP-Pakets signalisieren:
CWR, ECE, URG, ACK, PSH, RST, SYN, FIN

- **Checksum**

Prüfsumme, wie in UDP beschrieben

- **Urgent Pointer**

Zeigt auf Position im Datensegment welche nach den wichtigen Daten auftreten.

AUFGABE 2

Die Werte der in CIDR Notation gegebenen IP-Adresse (103.161.122.83/18) setzt sich aus zwei Komponenten zusammen:

1. Die Eigentliche IP-Adresse wird durch den ersten Teil 103.161.122.83 beschrieben.
2. Der Präfix 18 beschreibt die Anzahl er Bits, welche für den Adressraum der Netzwerkadresse reserviert sind. In diesem Fall sind es 18 Netzwerkadressen und 32-18 = 14 Hostadressen.

Die **Subnetzmaske** kann aus der Präfixlänge direkt bestimmt werden und ist in diesem Fall:
11111111.11111111.11000000.00000000 —> 255.255.192.0

Die **Broadcastadresse** kann bestimmt werden, indem alle Host-Bits auf 1 gesetzt werden. Hier:
103.161.122.83 —> 01100111.10100001.01111010.01010011—>
01100111.10100001.01111111.11111111 —> 103.161.127.255

Um die **Netzwerkadresse** zu ermitteln muss lediglich die Subnetzmaske und die IP-Adresse mit einem logischen AND verknüpft werden:

11111111.11111111.11000000.00000000 AND 01100111.10100001.01111010.01010011
= 01100111.10100001.01000000.00000000
—> 103.161.64.0

Um zu prüfen ob die Adresse 103.161.193.83/18 im selben Netzwerk liegt muss auch hier die Subnetzmaske und die entsprechenden IP mit einem logischen AND verknüpft werden um anschließend die Netzwerkadressen abzugleichen.

103.161.193.83 —>
01100111.10100001.11000001.01010011 AND 11111111.11111111.11000000.00000000
= 01100111.10100001.11000000.00000000
—> 103.161.192.0

Die beiden Adressen liegen also in verschiedenen Netzwerken

Rechnernetze -Übungsblatt 4 09.06.2024
Roman Schander
1598510

AUFGABE 3

Das Aufstellen einer Verbindung zwischen zwei unterschiedlichen Chat-Implementationen stellt in erster Linie kein Problem dar.
Die Problemantik entsteht bei der Kommunikationsweise der Clients. In meiner Implementation folgen alle ausgehenden Nachrichten einer strengen Formatierung welche zusätzlich zu den Daten (Textnachricht) weitere Informationen versendet, welche anschließend vom Empfänger zu interpretieren sind.

Nachrichten welche von anderen Chat-Programmen versendet werden würden also nicht richtig interpretiert werden und somit nicht angezeigt werden.

Ein Lösungsvorschlag wäre sich auf ein einheitliches Verfahren zu einigen, welches regelt wie Nachrichten und Zusatzinfos versendet bzw. empfangen werden.