

Tutorübung zur Vorlesung Grundlagen Rechnernetze und Verteilte Systeme Übungsblatt 10 (29. Juni – 2. Juli 2015)

Hinweis: Die mit * gekennzeichneten Teilaufgaben sind ohne Kenntnis der Ergebnisse vorhergehender Teilaufgaben lösbar.

Aufgabe 1 Hexdump

Gegeben sei der Hexdump aus Abbildung 1, welcher einen 86B langen Rahmen (Ethernet ohne FCS) darstellt. Die linke Spalte gibt den Offset (hexadezimal) in Vielfachen von Bytes an. Die beiden nachfolgenden Spalten repräsentieren die Daten (hexadezimal) in Blöcken zu je 8 Byte in Network-Byte-Order.

```
0x0000: 08 60 6e 45 dc e6 00 1c 14 01 4e 18 86 dd 60 00
0x0010: 00 00 00 20 06 40 2a 01 04 f8 0d 16 19 43 00 00
0x0020: 00 00 00 00 00 02 2a 02 02 e0 03 fe 10 01 77 77
0x0030: 77 2e 00 02 00 85 ce 44 00 50 9b 94 59 c9 2f e7
0x0040: 5d 10 50 10 65 00 85 88 00 00 47 45 54 20 2f 68
0x0050: 65 78 0d 0a 0d 0a
```

Abbildung 1: Hexdump eines Ethernet-Rahmens (inkl. L2-Header) in Network-Byte-Order.

Im Folgenden werden wir diese Nachricht schrittweise untersuchen. **Nutzen Sie zur Lösung die auf dem Beiblatt abgebildeten Protokoll-Header und Zusatzinformationen.**

a)* Was ist der Unterschied zwischen „Host-Byte-Order“ und „Network-Byte-Order“?

Host-Byte-Order ist die native Byte-Order eines Hosts, also Little- oder Big-Endian (bestimmt durch die CPU-Architektur).
Network-Byte-Order ist stets Big-Endian.

b)* Begründen Sie, weswegen überhaupt zwischen Host-Byte-Order und Network-Byte-Order zu unterscheiden ist.

Da unklar ist, welche Byte-Order ein Host verwendet, muss die Network-Byte-Order festgelegt sein.

— oder —

Schicht 1 spezifiziert im Netzwerkbereich typischerweise die Übertragung von 8-bit-Wörtern (“Oktetten”).
Wie größere Wörter auf Oktette aufgeteilt werden muss daher zusätzlich spezifiziert werden.

Für die nachfolgenden Teilaufgaben ist es sicher hilfreich, wenn Sie sich Anfang und Ende der jeweiligen Header in Abbildung 1 markieren. Verwenden Sie zur Lösung den bereitgestellten Cheatsheet.

c)* Geben Sie für das erste und letzte Byte des Ethernet-Headers den Offset in Bytes vom Beginn des Rahmens an.

0x0000 – 0x000D (14 B langer Ethernet-Header)

d) Welches Protokoll wird auf Schicht 3 verwendet?

Type-Feld (Ethertype) im Ethernet-Header: 0x86dd (Big-Endian) = IPv6

Falsch: Version-Feld im IP-Header, da zunächst unklar, ob überhaupt IP verwendet wird.

e) Geben Sie Funktion und Wert der L3-Header-Felder an, welche auf dem Transportweg von Routern verändert werden müssen.

Hop-Limit, Funktion: endlose Weiterleitung in Loops zu verhindern, hier: 0x40 = 64 Hops

f) Welche Länge hat die L3-SDU?

Payload-Length-Feld im IPv6-Header: 0x0020 = 32 B

g) Markieren Sie die Absender- und Empfänger-Adresse im L3-Header. (Zeichnen Sie es direkt in Abbildung 1 ein und machen Sie kenntlich, welche der Adressen zum Absender und welche zum Empfänger gehört.)

h) Woran ist zu erkennen, dass TCP als L4-Protokoll verwendet wird?

Next-Header-Feld im IPv6-Header: $0x06 = \text{TCP}$

i)* Geben Sie den Quellport der Nachricht in Dezimaldarstellung an.

$0x0ce44$ (BigEndian) = 52804

j)* Geben Sie den Zielport der Nachricht in Dezimaldarstellung an.

$0x0050$ (BigEndian) = 80

k) Für welches Protokoll auf der Anwendungsschicht ist die Nachricht offenbar bestimmt?

TCP 80 (well-known Port) = HTTP

l)* Geben Sie zwei Gründe an, weswegen Sie auf Basis der Ihnen bekannten Informationen nicht bestimmen können, wie viele Byte bis zum jetzigen Zeitpunkt über diese TCP-Verbindung bereits ausgetauscht wurden.

- Die initialen Sequenznummern, die beim Verbindungsaufbau ausgetauscht wurden, sind nicht bekannt. Es fehlt also der Bezugspunkt, auf den sich Sequenz- und Bestätigungsnummern beziehen.
- Es wäre möglich, dass die Sequenznummern bereits einen Wrap-Around hatten, also mehr als 4 GiB Daten ausgetauscht wurden.

m)* Wie groß ist die TCP-Payload für die Anwendungsschicht?

TCP-Header hat keine Optionen (Offset-Feld ist 5), d.h. 32 B - 20 B des TCP-Headers ergibt 12 B Payload.

n)* Können nach diesem Segment innerhalb der laufenden TCP-Verbindung weiterhin Daten in dieselbe Richtung übertragen werden?

Ja, da das FIN-Flag im TCP-Header nicht gesetzt ist.

o)* Können nach diesem Segment innerhalb der laufenden TCP-Verbindung noch Daten in die Gegenrichtung übertragen werden?

Unbekannt, da die Gegenseite die TCP-Verbindung mittels eines vorherigen FINs bereits geschlossen haben könnte.

Aufgabe 2 Kompression: Huffman-Kodierung

Gegeben sei das Alphabet $\mathcal{A} = \{a,b,c,d\}$ und die Nachricht

$$m = \text{aabccdacababbbcbcbddbbbaababdbcbabdbcbadba} \in \mathcal{A}^{40}.$$

a)* Bestimmen Sie die Auftrittswahrscheinlichkeiten $p_{i \in \mathcal{A}}$ der einzelnen Zeichen in m .

Aus den Zeichenhäufigkeiten ergibt sich:

$$p_a = \frac{11}{40}, p_b = \frac{17}{40}, p_c = \frac{6}{40}, p_d = \frac{6}{40}$$

b) Bestimmen Sie die den Informationsgehalt $I(p_{i \in \mathcal{A}})$ der einzelnen Zeichen.

Für den Informationsgehalt erhalten wir:

$$\begin{aligned} I(p_a) &= -\log_2(p_a) \approx 1.86 \text{ bit} \\ I(p_b) &= -\log_2(p_b) \approx 1.23 \text{ bit} \\ I(p_c) &= -\log_2(p_c) \approx 2.74 \text{ bit} \\ I(p_d) &= -\log_2(p_d) \approx 2.74 \text{ bit} \end{aligned}$$

c) Die Nachricht m stamme aus einer Nachrichtenquelle X . Bestimmen Sie auf Basis der bisherigen Ergebnisse die Quellenentropie $H(X)$.

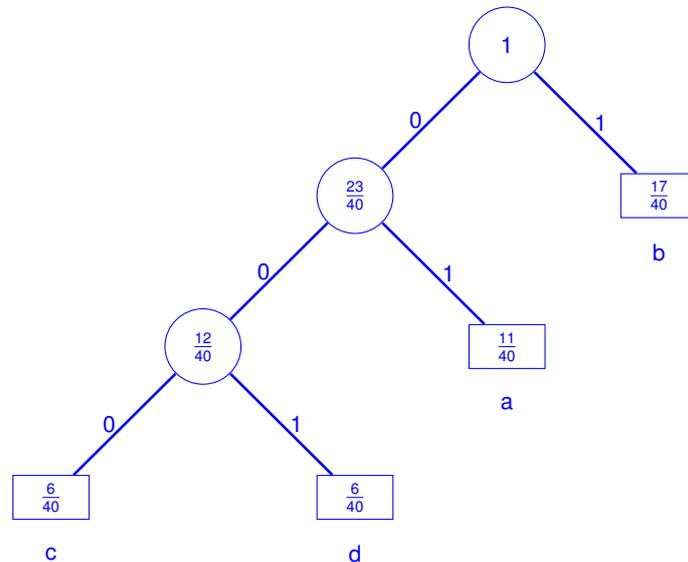
Die Quellenentropie ist nichts weiter als die mit den Auftrittswahrscheinlichkeiten gewichtete Summe des Informationsgehalts der Einzelzeichen:

$$H(X) = \sum_{i \in \mathcal{A}} p_i I(p_i) \approx 1.86 \text{ bit}$$

Dies bedeutet, dass sich die Zeichen der Quelle X mit durchschnittlich 1.86 bit pro Zeichen kodieren lassen.

d) Bestimmen Sie nun einen binären Huffman-Code C für diese Nachrichtenquelle.

Siehe Vorlesungsfolien. Beginnend bei den beiden Zeichen mit der geringsten Auftretswahrscheinlichkeit wird ein Baum beginnend bei den Blättern (den Zeichen) konstruiert. Dabei werden in jedem Schritt stets die beiden Knoten bzw. Blätter zusammengefasst, so dass die Summe deren Auftretswahrscheinlichkeiten über alle Knoten bzw. Blätter minimal ist:



Die Kanten werden mit 0 bzw. 1 beschriftet. Der Code lässt sich nun einfach ablesen, indem man von der Wurzel ausgehend die Kantenbeschriftungen abliest: $C = \{a \mapsto 01, b \mapsto 1, c \mapsto 000, d \mapsto 001\}$

Zeichen mit hoher Auftretswahrscheinlichkeiten erhalten kurze Codewörter. Außerdem lässt sich leicht überprüfen, dass C präfixfrei ist: Kein Codewort ist ein Präfix eines anderen Codeworts. Dies erleichtert die Dekodierung.

e) Bestimmen Sie die durchschnittliche Codewortlänge von C .

Die durchschnittliche Codewortlänge ergibt sich aus der mit den Auftretswahrscheinlichkeiten gewichteten Summe der Codewortlängen. Sei $l(c)$ die Länge eines Codeworts in C und $c(i)$ die Funktion, welche ein Zeichen $i \in \mathcal{A}$ auf ein Codewort aus C abbildet. Dann erhalten wir:

$$\bar{l}_C = \sum_{i \in \mathcal{A}} p_i \cdot l(c(i)) \approx 1.88$$

f) Vergleichen Sie die durchschnittliche Codewortlänge von C mit der Codewortlänge eines uniformen¹ Binärcodes.

Der kürzeste uniforme Code hat eine durchschnittliche Codewortlänge von $\bar{l}_U = 2$. Die Ersparnis beträgt also etwa 6%.

¹Ein Code heißt *uniform*, wenn alle Codewörter dieselbe Länge aufweisen.