

# Tutorübung zu GRNVS

## Übung 10

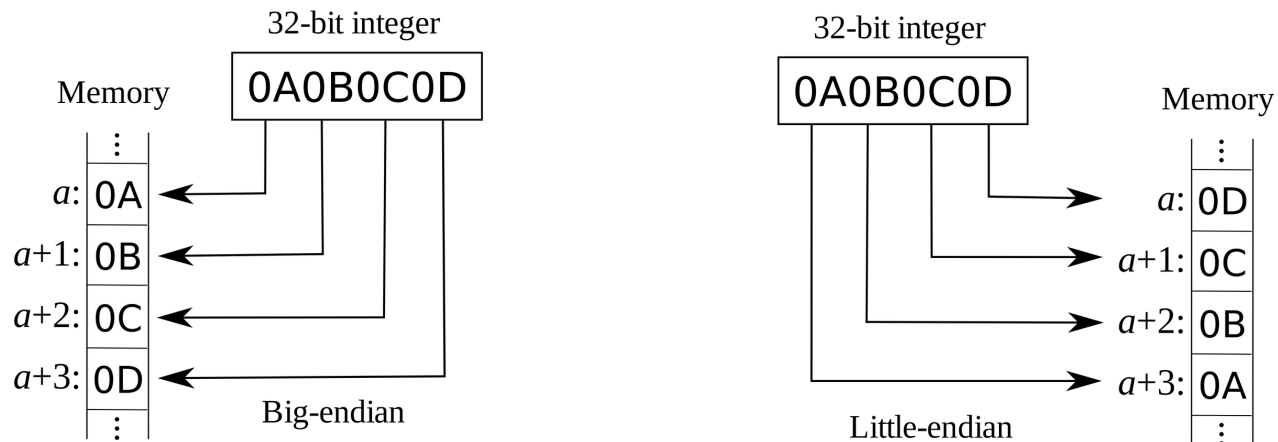
Jonas Andre

# Hexdump KLAUSUR!!!

```
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
```

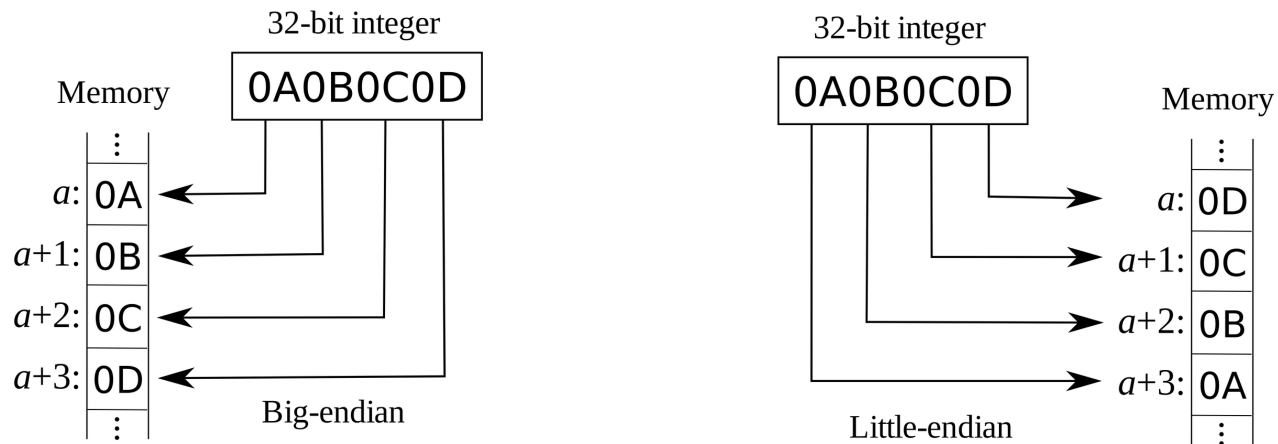
# Hexdump KLAUSUR!!!

```
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
```



# Hexdump KLAUSUR!!!

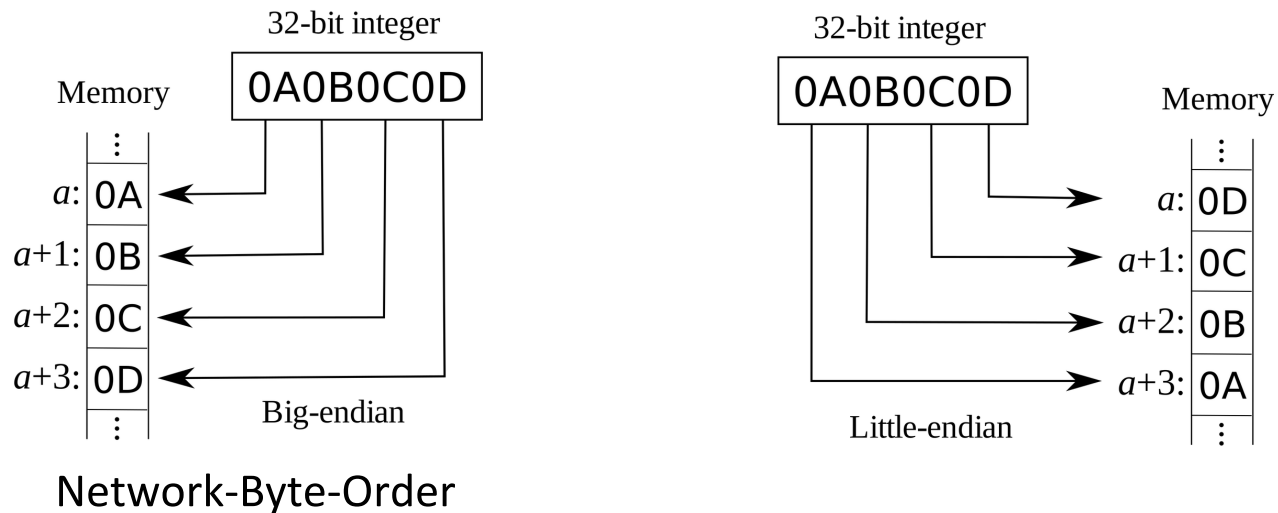
```
0x0000: 08 60 6e 45 dc e6 00 1c    14 01 4e 18 86 dd 60 00
0x0010: 00 00 00 20 06 40 2a 01    04 f8 0d 16 19 43 00 00
0x0020: 00 00 00 00 00 02 2a 02    02 e0 03 fe 10 01 77 77
0x0030: 77 2e 00 02 00 85 ce 44    00 50 9b 94 59 c9 2f e7
0x0040: 5d 10 50 10 65 00 85 88    00 00 47 45 54 20 2f 68
0x0050: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



Network-Byte-Order

# Hexdump KLAUSUR!!!

```
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
```



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

# Hexdump KLAUSUR!!!

```
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
```

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

# Hexdump KLAUSUR!!!

```
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
```

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

0x0000 – 0x000D

# Hexdump KLAUSUR!!!

```
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
```

d) Welches Protokoll wird auf Schicht 3 verwendet?



# Hexdump KLAUSUR!!!

```
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
```

d) Welches Protokoll wird auf Schicht 3 verwendet?

0x86dd → IPv6

# Hexdump KLAUSUR!!!

```
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
```

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

# Hexdump KLAUSUR!!!

```
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
```

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

Hop-Limit, Funktion: Verhinderung von Endlosschleifen

# Hexdump KLAUSUR!!!

```
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
```

f) Welche Länge hat die L3-SDU?

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?

# Hexdump KLAUSUR!!!

```
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
```

f) Welche Länge hat die L3-SDU?  $0x0020 = 32B$

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?

# Hexdump KLAUSUR!!!

```
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
```

f) Welche Länge hat die L3-SDU?  $0x0020 = 32B$

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?

# Hexdump KLAUSUR!!!

```
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
```

f) Welche Länge hat die L3-SDU?  $0x0020 = 32B$

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?

# Hexdump KLAUSUR!!!

```
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
```

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

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

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

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.

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

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

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



# Hexdump KLAUSUR!!!

```
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
```

- i)\* Geben Sie den Quellport der Nachricht in Dezimaldarstellung an.  $0xce44 = 52804$
- j)\* Geben Sie den Zielport der Nachricht in Dezimaldarstellung an.
- k) Für welches Protokoll auf der Anwendungsschicht ist die Nachricht offenbar bestimmt?
- 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.
- m)\* Wie groß ist die TCP-Payload für die Anwendungsschicht?
- n)\* Können nach diesem Segment innerhalb der laufenden TCP-Verbindung weiterhin Daten in dieselbe Richtung übertragen werden?
- o)\* Können nach diesem Segment innerhalb der laufenden TCP-Verbindung noch Daten in die Gegenrichtung übertragen werden?

# Hexdump KLAUSUR!!!

```
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
```

- i)\* Geben Sie den Quellport der Nachricht in Dezimaldarstellung an.  $0xce44 = 52804$
- j)\* Geben Sie den Zielport der Nachricht in Dezimaldarstellung an.  $0x0050 = 80$
- k) Für welches Protokoll auf der Anwendungsschicht ist die Nachricht offenbar bestimmt?
- 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.
- m)\* Wie groß ist die TCP-Payload für die Anwendungsschicht?
- n)\* Können nach diesem Segment innerhalb der laufenden TCP-Verbindung weiterhin Daten in dieselbe Richtung übertragen werden?
- o)\* Können nach diesem Segment innerhalb der laufenden TCP-Verbindung noch Daten in die Gegenrichtung übertragen werden?

# Hexdump KLAUSUR!!!

```
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
```

- i)\* Geben Sie den Quellport der Nachricht in Dezimaldarstellung an.  $0xce44 = 52804$
- j)\* Geben Sie den Zielport der Nachricht in Dezimaldarstellung an.  $0x0050 = 80$
- k) Für welches Protokoll auf der Anwendungsschicht ist die Nachricht offenbar bestimmt? TCP 80 → 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.
- m)\* Wie groß ist die TCP-Payload für die Anwendungsschicht?
- n)\* Können nach diesem Segment innerhalb der laufenden TCP-Verbindung weiterhin Daten in dieselbe Richtung übertragen werden?
- o)\* Können nach diesem Segment innerhalb der laufenden TCP-Verbindung noch Daten in die Gegenrichtung übertragen werden?

# Hexdump KLAUSUR!!!

```
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
```

- i)\* Geben Sie den Quellport der Nachricht in Dezimaldarstellung an.  $0xce44 = 52804$
- j)\* Geben Sie den Zielport der Nachricht in Dezimaldarstellung an.  $0x0050 = 80$
- k) Für welches Protokoll auf der Anwendungsschicht ist die Nachricht offenbar bestimmt? TCP 80 → 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.
- m)\* Wie groß ist die TCP-Payload für die Anwendungsschicht? 12B
- n)\* Können nach diesem Segment innerhalb der laufenden TCP-Verbindung weiterhin Daten in dieselbe Richtung übertragen werden?
- o)\* Können nach diesem Segment innerhalb der laufenden TCP-Verbindung noch Daten in die Gegenrichtung übertragen werden?

# Hexdump KLAUSUR!!!

```
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
```

- i)\* Geben Sie den Quellport der Nachricht in Dezimaldarstellung an.  $0xce44 = 52804$
- j)\* Geben Sie den Zielport der Nachricht in Dezimaldarstellung an.  $0x0050 = 80$
- k) Für welches Protokoll auf der Anwendungsschicht ist die Nachricht offenbar bestimmt? TCP 80 → 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.
- m)\* Wie groß ist die TCP-Payload für die Anwendungsschicht? 12B
- n)\* Können nach diesem Segment innerhalb der laufenden TCP-Verbindung weiterhin Daten in dieselbe Richtung übertragen werden? Ja, kein FIN-Flag
- o)\* Können nach diesem Segment innerhalb der laufenden TCP-Verbindung noch Daten in die Gegenrichtung übertragen werden?

# Hexdump KLAUSUR!!!

```
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
```

- i)\* Geben Sie den Quellport der Nachricht in Dezimaldarstellung an.  $0xce44 = 52804$
- j)\* Geben Sie den Zielport der Nachricht in Dezimaldarstellung an.  $0x0050 = 80$
- k) Für welches Protokoll auf der Anwendungsschicht ist die Nachricht offenbar bestimmt? TCP 80 → 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.
- m)\* Wie groß ist die TCP-Payload für die Anwendungsschicht? 12B
- n)\* Können nach diesem Segment innerhalb der laufenden TCP-Verbindung weiterhin Daten in dieselbe Richtung übertragen werden? Ja, kein FIN-Flag
- o)\* Können nach diesem Segment innerhalb der laufenden TCP-Verbindung noch Daten in die Gegenrichtung übertragen werden? Unbekannt, evtl. bereits FIN von Gegenseite

# Huffman-Kodierung

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

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

- a)\* Bestimmen Sie die Auftrittswahrscheinlichkeiten  $p_{i \in \mathcal{A}}$  der einzelnen Zeichen in  $m$ .
- b) Bestimmen Sie die den Informationsgehalt  $I(p_{i \in \mathcal{A}})$  der einzelnen Zeichen.
- c) Die Nachricht  $m$  stamme aus einer Nachrichtenquelle  $X$ . Bestimmen Sie auf Basis der bisherigen Ergebnisse die Quellentropie  $H(X)$ .

# Huffman-Kodierung

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

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

- d) Bestimmen Sie nun einen binären Huffman-Code  $C$  für diese Nachrichtenquelle.
- e) Bestimmen Sie die durchschnittliche Codewortlänge von  $C$ .
- f) Vergleichen Sie die durchschnittliche Codewortlänge von  $C$  mit der Codewortlänge eines uniformen<sup>1</sup> Binärcodes.

<sup>1</sup>Ein Code heißt *uniform*, wenn alle Codewörter dieselbe Länge aufweisen.