



6. Übung zur Vorlesung Rechnernetze (Musterlösung)

Abgabe: 18.01.04

Aufgabe 1: Ethernet-Rahmen (3 Punkte, 2 + 1)

- a) In dieser Aufgabe sollen Sie die unten stehenden Ethernet-Hex-Dumps interpretieren. Identifizieren Sie die einzelnen Teile des Ethernet-Kopfs, des LLC-Headers und der SNAP-Erweiterung (wenn vorhanden)! Geben Sie auch an, welche Protokolle jeweils auf der nächst höheren Schicht verwendet werden!
- b) Geben Sie außerdem an, welche Rechner die Pakete jeweils in Empfang nehmen! Bei den dargestellten Paketen fehlt jeweils das CRC-Feld am Ende des Ethernetrahmens. Können Sie das in allen Fällen anhand des Hex-Dumps erkennen? Wenn ja, wie? (Hinweis die dargestellten Pakete haben unterschiedliche Ethernet-Rahmenvarianten, vgl. Skript Kapitel 7.4)

Lösungsvorschlag:

a)

Paket 1:

```
AA AA AA AA AA AA AA AB 0A 12 47 FF 00 BC 40 61  
FF 62 B5 F0 00 84 AA AA 03 08 00 07 80 9B 00 41  
00 00 00 00 00 46 FF B5 02 FE 02 21 18 00 46 B5  
FE 00 0E 53 65 6B 72 65 74 61 72 69 61 74 20 56  
53 0B 4C 61 73 65 72 57 72 69 74 65 72 11 56 65  
72 74 65 69 6C 74 65 20 53 79 73 74 65 6D 65 AB  
CD FE 63 54 73 AC D9 1F 12 90 73 71 72 71 64 AB  
AD 6E 63 54 75 AC D8 2B 13 91 74 71 A2 71 B4 AB  
56 6E 63 54 73 6C D9 1F 12 96 73 61 62 76 64 6B  
00 00 00 00 01 85 E8 00 01 0B
```

- a) Präambel AA AA AA AA AA AA AA AB
- b) Dest Addr 0A 12 47 FF 00 BC
- c) Source Addr 40 61 FF 62 B5 F0
- d) Type/Length 00 84 = 132 < 1536 => Ethernet 802.3
- e) DSAP AA
- f) SSAP AA => SNAP Erweiterung
- g) Ctrl 03
- h) SNAP 08 00 07 80 9B => Apple Talk

Paket 2:

```
AA AA AA AA AA AA AA AB 0A 12 47 FF 00 BC 40 61  
FF 62 B5 F0 08 06 00 01 08 00 06 04 00 02 00 E0  
7D BB F2 24 86 3C 4D 1F 00 E0 7D BB F2 15 86 3C  
4D D2 20 20 20 20 20 20 20 20 20 20 20 20 20  
20 20 20 20
```

- a) Präambel AA AA AA AA AA AA AA AB
- b) Dest Addr 0A 12 47 FF 00 BC
- c) Source Addr 40 61 FF 62 B5 F0
- d) Type/Length 08 06 = 2048 > 1536 => Ethernet II, ARP

Paket 3:

```
AA AA AA AA AA AA AA AB FF FF FF FF FF FF 40 61  
FF 62 B5 F0 00 64 E0 E0 03 FF FF 00 60 00 04 00  
00 00 00 FF FF FF FF FF FF 04 52 00 00 00 00 00  
02 44 2A FF 75 40 08 00 02 06 40 50 4C 58 32 00
```

```

00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 8A 7A 0F 81 00
00 00 00 00 01 85 E8 00 01 0B

```

- a) Präambel AA AA AA AA AA AA AA AB
b) Dest Addr FF FF FF FF FF FF
c) Source Addr 40 61 FF 62 B5 F0
d) Type/Length 00 64 = 100 < 1536 => Ethernet 802.3
e) DSAP E0 => IPX
f) SSAP E0 => IPX
g) Ctrl 03

b) Pakete 1 und 2 werden jeweils von dem Rechner mit der MAC-Adresse 0A 12 47 FF 00 BC in Empfang genommen, das dritte Paket ist ein Broadcast-Paket und wird daher von allen Rechnern im lokalen Netz in Empfang genommen. Bei allen Paketen fehlt das CRC Feld. Bei Paket 1 und 3 ist dies anhand des Hex-Dumps erkennbar durch Angabe des Längensfeldes. Das Längensfeld gibt die Anzahl der Bytes der LLC-PDU an (also ohne das 4 Byte lange CRC-Feld, welches nicht mehr zu LLC-PDU gehört). Durch Abzählen der Bytes stellen wir fest, dass bei Paket 1 und 2 auf das Längensfeld nur noch genau so viele Bytes folgen, wie im Längensfeld angegeben. Die 4 Bytes des CRC-Feldes fehlen also.

Aufgabe 2: Slotted ALOHA (1 Punkt)

1000 Computer in einem lokalen Netz streiten sich um den Zugang zu einem Slotted ALOHA Kanal. Eine Station setzt im Durchschnitt zufällig verteilt 50 Anfragen/Std. ab. Ein Slot ist 15 ms lang, und jede Anfrage passt genau in einen Slot. Wie groß ist der Durchsatz auf dem Kanal?

Lösungsvorschlag:

$L =$ angebotene Last

$R =$ Anzahl der Pakete/Zeiteinheit = $50 * 1000 / 3600s = 13,89$ Anfragen/s

$T =$ Slotlänge = $15ms = 0,015$ s/Slot

$L = R * T = 13,89$ Anfragen/s * $0,015$ s/Slot = $0,2083$ Anfragen/Slot

W_i : Wartezeit zwischen Paket i und $i+1$

V : Zeit zwischen Sendewunsch und nächstem Slotanfang

Annahme wie im Skript: Mittlere Wartezeit W_i unterliegt einer Exponentialverteilung.

Dichtefunktion der Wartezeit W : $p(W) = R * e^{-RW}$

(vgl. Skript 8.5.2)

$P(\text{Erfolg}) = p(W_{i-1} > T-V) * p(W_i > V)$

(vgl. Skript 8.5.3)

$$\begin{aligned}
 P(\text{Erfolg}) &= \int_{T-V}^{\infty} R * e^{-RW} dW * \int_V^{\infty} R * e^{-RW} dW = \left[-e^{-RW} \right]_{T-V}^{\infty} * \left[-e^{-RW} \right]_V^{\infty} \\
 &= (-e^{-\infty} + e^{-R(T-V)}) * (-e^{-\infty} + e^{-RV}) = e^{-R(T-V)} * e^{-RV} = e^{-RT} = e^{-L}
 \end{aligned}$$

(Kein Punktabzug, falls diese Herleitung fehlt! Diese Rechnung ist auch im Skript aber mit etwas weniger Zwischenschritten.)

Durchsatz $D = L * P(\text{Erfolg}) = 0,2083$ Anfragen/Slot * $e^{0,2083} = 0,1691$. (Bemerkung: Dasselbe ergibt sich für $2,81$ Anfragen/Slot = 187 Anfragen/s = 673 pro Stunde und Station).

Aufgabe 3: Link Protokoll nach Fletcher (6 Punkte)

Wir wollen unser Programm von den letzten Übungsblättern erneut etwas verändern. Dieses mal wollen wir nur einfache Strings versenden. Wir wollen nun jedoch statt einem einfachen Stop-and-Wait das Link-Protokoll nach Fletcher implementieren (siehe Kapitel 7.3 im Skript, Anmerkung: Das Protokoll von Fletcher ist eigentlich auf Layer 2 anzusiedeln, lässt sich jedoch ebenso gut auf einem anderen Layer implementieren, wenn man von einer direkten Punkt-zu-Punkt-Verbindung ausgeht). Als Ausgangsbasis für Ihre Implementierung können Sie folgendes Archiv herunterladen: http://www-vs.informatik.uni-ulm.de/teach/ws04/rn1/OSI_Aufgabe2b.zip. Sie werden jedoch vermutlich die Klasse Layer7 und evtl. auch die Schnittstelle zu den Anwendungen in Server.java und Client.java etwas umbauen müssen, und dürfen auch neue Hilfsklassen hinzufügen. Vermutlich werden Sie separate Threads zum Versenden und Empfangen der Pakete implementieren müssen. Die Implementierung der Klasse Layer4 soll jedoch unverändert bleiben. Des weiteren finden Sie den Originaltext von Fletcher unter <http://www-vs.informatik.uni-ulm.de/teach/ws04/rn1/p26-fletcher.pdf>. Ihre Pakete sollen am Ende auch ein CRC-Checksumme enthalten, die auf Empfängerseite überprüft wird. Der Code zum Erzeugen der CRC-Checksumme war Aufgabe des letzten

Übungsblattes und ist in der Layer7 Klasse bereits enthalten. Die Länge der Sequenznummern ist beim Protokoll von Fletcher nicht festgelegt, verwenden Sie 3 Bit lange Sequenznummern!

Geben Sie bitte den kompletten Quelltext ausgedruckt oder per Email ab! Schicken Sie den gesamten Quelltext gezippt an: schorr@informatik.uni-ulm.de mit dem Betreff „RN1 – Übung 6“.

Lösungsvorschlag:

Wie in der Übung erläutert, war die Aufgabe leider doch nicht korrekt lösbar, ohne die Layer4 Klasse zu verändern. Daher fließen die 6 Punkte für diese Aufgabe nicht in die Zahl der theoretisch erreichbaren Punkte ein. Wer die Aufgabe bearbeitet hat, wird jedoch dennoch Punkte dafür erhalten. Die Musterlösung enthält die korrekte Lösung mit veränderter Layer4 Klasse (siehe Quelltext unter: http://www-vs.informatik.uni-ulm.de/teach/ws04/rn1/OSI_Aufgabe2b_Musterloesung.zip)