

# Vernetzte Systeme

## Übung 11

Ausgabe: **30. Juni 2006**

Abgabe: **keine Abgabe, Musterlösung online ab 7.7.2006**

Bitte schreiben Sie immer Ihre(n) Namen auf die Lösungsblätter.

### 1 Fragmentierung von IPv4-Paketen

Die Gesamtlänge eines IP-Paketes, inklusive Header und Daten, kann bis zu 65535 Bytes betragen. Dieser Wert ist für die meisten Netze jedoch zu hoch. Ethernet erlaubt z.B. eine maximale Paketlänge (Daten) von 1500 Bytes. Diese maximale Paketlänge für ein Netz wird *Maximum Transmission Unit (MTU)* genannt. Um dennoch grosse IP-Pakete über Netze mit kleiner MTU transportieren zu können, werden diese in mehrere kleine Teilstücke *fragmentiert* und beim Empfänger wieder zusammengesetzt.

Verschiedene Felder im IP-Header<sup>1</sup> sorgen dafür, dass sich die Fragmente wieder zum Ausgangspaket zusammensetzen lassen. Beim Fragmentieren wird der ursprüngliche Wert des *Identification*-Feldes kopiert, so dass es für zusammengehörige Fragmente den gleichen Wert aufweist. Ein gesetztes *More Fragments (MF)*-Bit zeigt an, dass es sich bei einem IP-Paket um ein Fragment handelt, dem noch weitere Fragmente folgen. Das *Offset*-Feld gibt die Position an, die das Fragment im Gesamtpaket einnimmt. Dabei bezeichnet der Wert des Feldes die Anzahl der 8-Byte-Blöcke, die vor dem Fragment liegen. Ein Fragment mit MF=1 enthält also immer komplette 8-Byte-Blöcke als Nutzdaten.

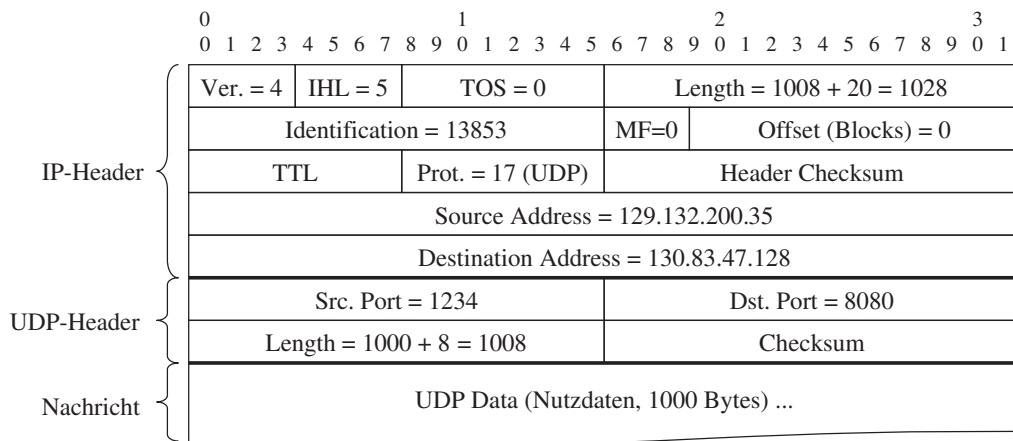
- Warum gibt das *Offset*-Feld im IP-Header den Offset in 8-Byte-Einheiten an?
- Eine Nachricht mit einer Länge von 1000 Bytes soll per UDP/IPv4 verschickt werden. Dazu wird sie zunächst in ein UDP-Datagramm<sup>2</sup> eingebettet. Die Adresse des Senders sei 129.132.200.35:1234, die des Empfängers 130.83.47.128:8080. Nehmen Sie an, dass Sender und Empfänger über zwei benachbarte Netze verbunden sind: Pakete laufen vom Sender über Netz 1 zu einem Router und von dort über Netz 2 zum Empfänger. Das erste Netz hat eine MTU von 1024 Bytes; das zweite hat eine MTU von 512 Bytes. Beachten Sie, dass der IPv4-Header selbst 20 Bytes lang ist.

Skizzieren Sie die Pakete, die auf der Vermittlungsschicht (Network Layer) beim Empfänger ankommen. Tragen Sie dazu die fehlenden Angaben in die Offset-, Length- und MF-Felder in die Schablonen auf dem Zusatzblatt ein. Untenstehende Abbildung zeigt das unfragmentierte Ausgangspaket.

---

<sup>1</sup>Die genaue Spezifikation der Felder in den IP- bzw. UDP-Headern ist in den RFCs 791 (IP) und 768 (UDP) zu finden.

<sup>2</sup>Der Header eines UDP-Datagramms ist 8 Bytes lang und spezifiziert Quellport, Zielport, Länge des UDP-Datagramms inkl. Header und eine Prüfsumme.



- c) Das Ausgangspaket aus Teilaufgabe b) wird schon beim Sender fragmentiert. Warum darf die Netzwerkschicht auf Senderseite das UDP-Datagramm nicht bereits auf zwei oder mehr IP-Pakete aufteilen, die sie dann unfragmentiert verschickt?

## 2 Cyclic Redundancy Check

- a) Berechnen Sie für das Generator-Polynom  $G(x) = x^4 + x + 1$  und die Daten  $D = 110101111001$  die Checksumme nach dem in der Vorlesung vorgestellten CRC-Verfahren (Folien 5/11ff). Welcher Bitstring  $T$  wird übertragen?
- b) Die Leitung, über welche die Daten gesendet werden, ist fehlerbehaftet, so dass nicht  $T$ , sondern das veränderte  $T' = 1100010010010111$  empfangen wird. Kann dieser Fehler erkannt werden? Geben Sie Ihre Rechnung an!
- c) Welche Struktur müssen Fehler aufweisen, damit sie **nicht** erkannt werden? Geben Sie für das Beispiel ein verändertes  $T'' \neq T'$  an, das nicht als Fehler erkannt werden würde. Halten Sie fest, wie Sie  $T$  modifizieren, um  $T''$  zu erhalten. Tip: Vergleichen Sie  $T$  und  $T'$ . Wie sind die Änderungen zustande gekommen?

## 3 Slotted Aloha

In dieser Aufgabe beschäftigen wir uns mit dem in der Vorlesung besprochenen Slotted Aloha-Verfahren (Folien 5/26ff). Dort wurde gezeigt, dass in einem Netzwerk mit  $n$  Stationen der Durchsatz optimiert werden kann, wenn jede Station mit einer Wahrscheinlichkeit von  $1/n$  sendet.

Sie dürfen zum Lösen dieser Aufgabe ein Mathematikprogramm wie Maple verwenden. Legen Sie dann einen Ausdruck der einzelnen Rechenschritte (Ein- und Ausgaben) bei.

- a) Hier betrachten wir eine kleines lokales Netzwerk, in dem sich neben Clients auch ein File-Server befindet. Da dieser Server von allen Clients verwendet wird, soll die Wahrscheinlichkeit, mit der er sendet, auf einen konstanten Wert  $p_s$  gesetzt werden. Berechnen Sie analog zum Beispiel auf Folie 5/27 die optimale Sende-Wahrscheinlichkeit  $p$  für die übrigen  $n$  Clients.
- b) Nehmen Sie an, dass  $n = 10$  und  $p_s = 1/3$  ist. Mit welcher Wahrscheinlichkeit  $p$  sollten die Clients also senden?
- c) Wie hoch ist bei gleichen Annahmen die Wahrscheinlichkeit, dass irgendeine Station erfolgreich sendet?
- d) Vergleichen und kommentieren Sie die Werte aus b) und c) mit den Ergebnissen für das *normale* Slotted Aloha aus der Vorlesung.
- e) Abschliessend soll Slotted Aloha mit einem Reservations-Verfahren verglichen werden. Sie haben in der Vorlesung gehört, dass die Effizienz von Slotted Aloha bei vielen Sendern maximal 37% beträgt. Wie hoch liegt diese im optimalen Fall bei der Verwendung eines Reservations-Verfahrens? Welche Annahmen treffen Sie dabei? Nennen Sie weitere Vor- und Nachteile der beiden Verfahren. Unter welchen Voraussetzungen würden Sie sich für welches Verfahren entscheiden?

## 4 Verschachtelung von Paketen

Wie aus der Vorlesung bereits bekannt ist, wird ein Datenpaket auf jedem Layer um einen Header ergänzt. Folgende Daten haben wir auf unserem ETH-Rechner empfangen, wobei die Checksumme des Ethernet-Frames nicht angegeben ist:

```
AA AA AA AA AA AA AB 00 90 27 A2 C3 A7 00 D0
BC EC AA 64 08 00 45 00 00 4B 87 06 40 00 FE 06
62 C3 81 84 0D 42 81 84 82 98 00 6E 0A DA 78 20
5C 99 93 F6 04 FB 50 18 22 38 CE CA 00 00 2B 4F
4B 20 50 6F 70 20 73 65 72 76 65 72 20 61 74 20
76 73 20 73 69 67 6E 69 6E 67 20 6F 66 66 2E 0D
0A
```

- a) Welche Pakete sind hier ineinander verschachtelt, bzw. um welche Protokolle handelt es sich hier? Extrahieren Sie alle Informationen und geben Sie sie in übersichtlicher Form wieder!
- b) Wie lang sind die eigentlichen Daten?
- c) Um welche Anwendung handelt es sich wahrscheinlich? Warum?
- d) Zeigen Sie, dass die IP-Checksumme richtig ist! Beachten Sie den Übertrag!

# Zusatzblatt Vernetzte Systeme. Übung 11.

1b)

0				1				2				3																			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Ver. = 4				IHL = 5				TOS = 0				Length =																			
Identification = 13853								MF=				Offset (Blocks) =																			
TTL				Prot. = 17 (UDP)				Header Checksum																							
Source Address = 129.132.200.35																															
Destination Address = 130.83.47.128																															
IP Data ...																															

0				1				2				3																			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Ver. = 4				IHL = 5				TOS = 0				Length =																			
Identification = 13853								MF=				Offset (Blocks) =																			
TTL				Prot. = 17 (UDP)				Header Checksum																							
Source Address = 129.132.200.35																															
Destination Address = 130.83.47.128																															
IP Data ...																															

0				1				2				3																			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Ver. = 4				IHL = 5				TOS = 0				Length =																			
Identification = 13853								MF=				Offset (Blocks) =																			
TTL				Prot. = 17 (UDP)				Header Checksum																							
Source Address = 129.132.200.35																															
Destination Address = 130.83.47.128																															
IP Data ...																															

0				1				2				3																			
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
Ver. = 4				IHL = 5				TOS = 0				Length =																			
Identification = 13853								MF=				Offset (Blocks) =																			
TTL				Prot. = 17 (UDP)				Header Checksum																							
Source Address = 129.132.200.35																															
Destination Address = 130.83.47.128																															
IP Data ...																															