

Vernetzte Systeme

Übung 8

Ausgabe: **19. Mai 2005**Abgabe: **27. Mai 2005**

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.

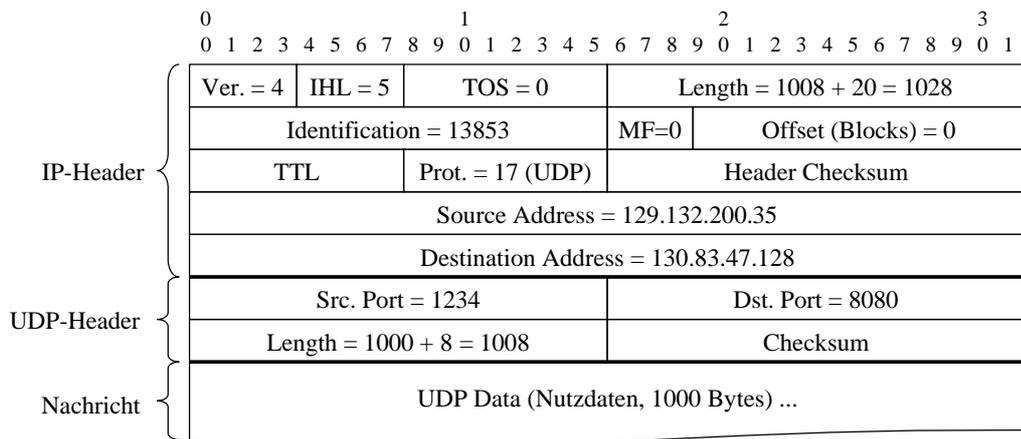
Verschiedene Felder im IP-Header¹ sorgen dafür, dass sich die Fragmente wieder zum Ausgangspaket zusammensetzen lassen. Beim Fragmentieren wird der ursprüngliche Wert des *Identification*-Feldes kopiert, so dass es für zusammengehörige Fragmente den gleichen Wert aufweist. Ein gesetztes *More Fragments (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-Datagramm² 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.

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

²Der Header eines UDP-Datagramms ist 8 Bytes lang und spezifiziert Quellport, Zielpport, 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?

Zusatzblatt Vernetzte Systeme. Übung 8.

2b)

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
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
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
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
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 ...																							