Lösung 04: Fehlererkennung, CSMA/CA, Ethernet

Aufgabe 1: Fehlererkennung, Internet Checksum

Die binäre Repräsentation des ASCII-Strings ergibt 3 16-Bit Wörter: Zunächst werden die ersten beiden Wörter *mit Übert*rag addiert (nicht XOR):

01000011 01001000 (1. Wort)
01001011 01010011 (2. Wort)
10001110 10011011 (Zwischenergebnis)

Nun addiert man zum Ergebnis das 3. Wort ebenfalls wieder mit Übertrag

10001110 10011011 (Zwischenergebnis)
01010101 01001101 (3. Wort)
11100011 11101000 (Ergebnis)

Die übertragene Checksumme ist dann das 1er-Komlement, also **00011100 00010111.**

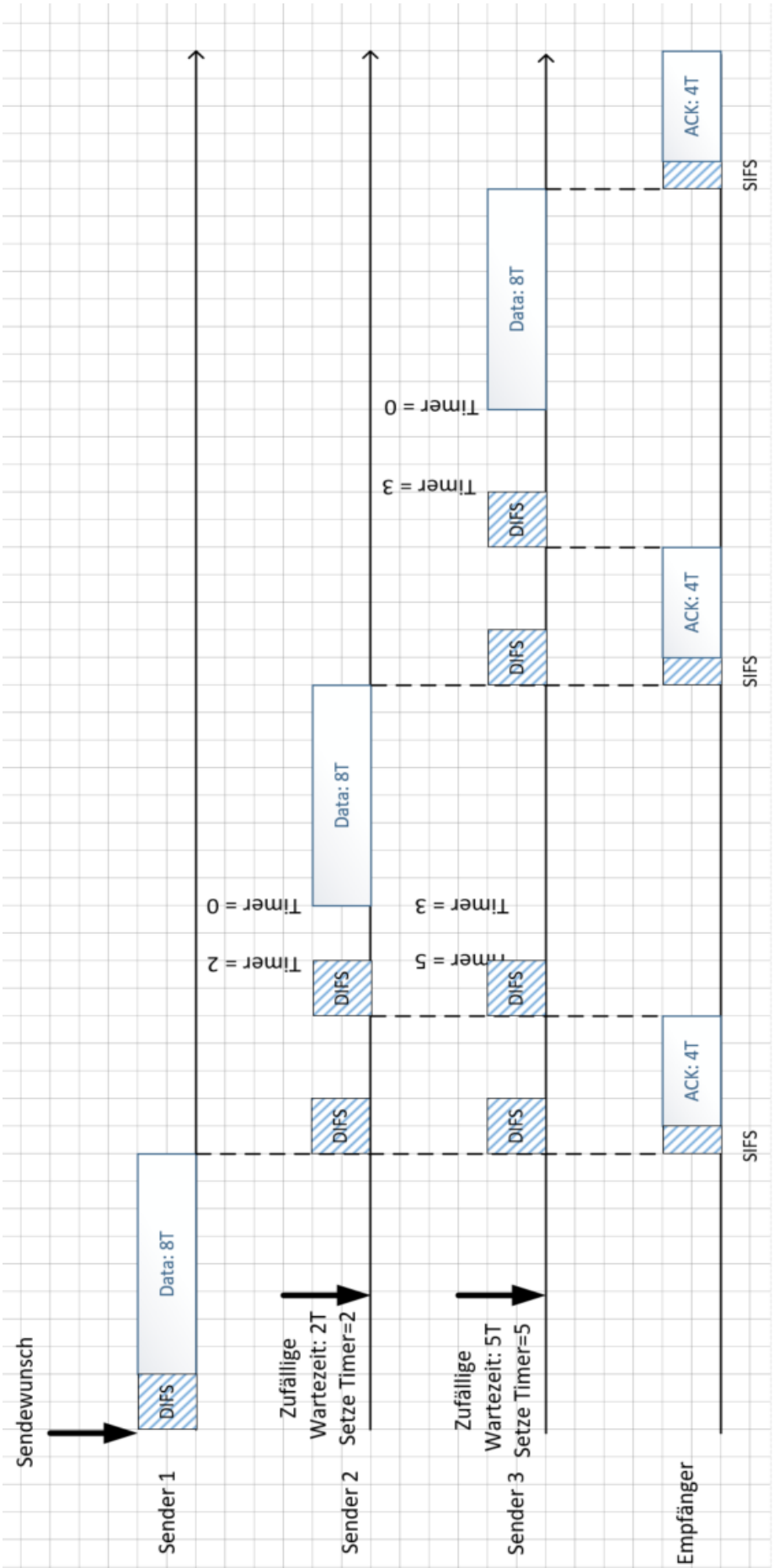
Hinweis: Man könnte die 3 Wörter natürlich auch alle auf „einmal addieren". In diesem Falle muss man lediglich darauf achten, dass man mit den Überträgen korrekt umgeht.

Aufgabe 2: CSMA/CA

a) Siehe Skizze. Achtung: Der Timer wird nur nach Ablauf der DIFS Periode heruntergezählt und wenn der Kanal frei ist.

b) Nein! Bei Hidden Station können sich nicht alle Stationen gegenseitig hören. Mit CSMA/CA kann durch fehlende ACKs eine Problem bei der Datenübertragung erkannt werden, eine Kollision aber nicht grundsätzlich verhindert werden. Eine Kollision beim Empfänger kann jederzeit auftreten.

c) Der Repeater halbiert die mögliche Datenrate. Man kann sich das wie folgt vorstellen. Annahme: Der Router ist an einem Ende des Hauses, der Repeater in der Mitte und die WLAN Station am anderen Ende des Hauses. Nur der Repeater erreicht sowohl den Router als auch die WLAN Station. Weder der Router noch die WLAN Station können sich gegenseitig direkt hören.
Sender der Router eine Nachricht an die Station, muss diese zuerst an den Repeater gesendet werden. Während dieses Versands kann der Repeater nicht sofort die Nachricht verstärken und an die WLAN Station weiterleiten. In der Umgebung des Repeaters käme es zur Auslöschung mit der Originalnachricht. Der Repeater muss deshalb warten, bis die Nachricht empfangen wurde und kann erst dann mit der Weiterleitung beginnen.
Hinweis: Das gilt nur für einen einfachen Repeater. Sollte die Kommunikation Router-Repeater eine andere Frequenz verwenden als die Kommunikation Repeater-WLAN Station gilt das nicht.


Sendewunsch
Sender 1
DIFS
Data: 8T
Zufällige Wartezeit: 2T
Setze Timer=2
Sender 2
DIFS
Timer = 2
Timer = 0
Data: 8T
Zufällige Wartezeit: 5T
Setze Timer=5
Sender 3
DIFS
Timer = 5
Timer = 3
Timer = 0
Data: 8T
Empfänger
SIFS
ACK: 4T
SIFS
ACK: 4T
SIFS
ACK: 4T

Aufgabe 3: Ethernet 802.3 in Wireshark

a) Alle folgenden Antworten beziehen sich auf den mitgelieferten Muster-Trace: `trace.pcapng`
Im konkreten Fall findet man die HTTP GET-Anfrage in Paket #86, die HTTP 200 OK Antwort in Paket #88. Hinweis: Am einfachsten nach HTTP filtern.

b) Sowohl die Präambel als auch die CRC Prüfsumme sind in Wireshark nicht zu sehen. Der Grund: Die Hardware / Netzwerkkarte entfernt diese Information bereits bevor sie ans Betriebssystem und an die Aufzeichnungssoftware weitergegeben wird.

c) Die eigene MAC Adresse liest man z.B. Source MAC Adresse der GET-Anfrage ab, oder aus der Destination MAC Adresse der 200 OK Antwort. Im konkreten Fall lautet sie: 00:50:b6:7c:53:53.

d) Die andere 2. Destination MAC Adresse (E0:28:6D:4B:D5:F9) gehört nicht dem Web Server, der diese Webseite bereitstellt. Sondern Sie gehört zum nächsten Router. Im konkreten Fall ist es die Fritzbox des Dozenten.

e) Broadcast MAC Adresse lautet: FF:FF:FF:FF:FF:FF.
Der Wireshark Filter lautet: `eth.dst == ff:ff:ff:ff:ff:ff`

f) Das Type-Feld hat hier den Wert 0x0800. Es steht für IPv4. Mit dieser Information weiß der Empfänger sofort, dass im Ethernet Frame ein IPv4 Datagramm enthalten ist. Für IPv6 würde hier der Wert 0x86DD stehen.

g) Normalerweise akzeptiert ein Netzwerkadapter nur alle Pakete, die entweder an die Broadcast MAC Adresse oder an seine eigene MAC-Adresse gerichtet sind. Ist der *Promiscuous* Mode jedoch aktiviert, werden alle (!) Pakete akzeptiert, unabhängig von der MAC Zieladresse.

h) Im Beispiel Trace wurden am meisten Bytes von 00:50:b6:7c:53:53 zur MAC Adresse E0:28:6D:4B:D5:F9 gesendet. Das entspricht der Fritzbox des Dozenten, siehe d).

Ethernet · 11 | IPv4 · 11 | IPv6 · 7 | TCP · 9 | UDP · 19

Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s B → A
00:11:32:7a:ac:e3	01:00:5e:00:00:16	1	62	1	62	0	0	3.138592	0.0000	—	
00:50:b6:7c:53:53	e0:28:6d:4b:d5:f9	75	11 k	40	6855	35	4945	0.000000	9.5615	5735	
00:50:b6:7c:53:53	ff:ff:ff:ff:ff:ff	9	1023	9	1023	0	0	4.272647	6.4294	1272	
00:50:b6:7c:53:53	01:00:5e:00:00:fb	8	592	8	592	0	0	4.273072	5.6660	835	
00:50:b6:7c:53:53	33:33:00:00:00:fb	8	752	8	752	0	0	4.273723	5.6657	1061	
00:50:b6:7c:53:53	33:33:00:01:00:03	6	528	6	528	0	0	4.275146	6.0834	694	
00:50:b6:7c:53:53	01:00:5e:00:00:fc	6	408	6	408	0	0	4.275297	6.0833	536	
00:50:b6:7c:53:53	01:00:5e:00:00:16	2	108	2	108	0	0	4.411146	0.4999	1728	
01:00:5e:00:00:16	b0:e8:92:0c:2d:9a	1	78	0	0	1	78	2.980002	0.0000	—	
01:00:5e:00:00:16	e8:df:70:df:19:52	1	86	0	0	1	86	5.291439	0.0000	—	
01:00:5e:7f:ff:fa	e0:28:6d:4b:d5:f9	2	330	0	0	2	330	0.028458	5.0003	0	

Aufgabe 4: `ethtool`

Der Screenshot zeigt eine mögliche Ausgabe[1]. Man erkennt, dass der Netzwerkadapter sowohl Halbduplex- (Hub) als auch Vollduplex (Switch) Betriebsmodi unterstützt. Ferner sind verschiedene Übertragungsraten möglich. Das alles wird automatisch ausgehandelt: Autonegotiation:

https://de.wikipedia.org/wiki/Autonegotiation

[1] Hier wurde nicht die Betriebssystem-VM verwendet. Das Ergebnis sollte jedoch ähnlich aussehen.

```
android@android:~$ ethtool enp0s3
Settings for enp0s3:
        Supported ports: [ TP ]
        Supported link modes:   10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Supported pause frame use: No
        Supports auto-negotiation: Yes
        Supported FEC modes: Not reported
        Advertised link modes:  10baseT/Half 10baseT/Full
                                100baseT/Half 100baseT/Full
                                1000baseT/Full
        Advertised pause frame use: No
        Advertised auto-negotiation: Yes
        Advertised FEC modes: Not reported
        Speed: 1000Mb/s
        Duplex: Full
        Port: Twisted Pair
        PHYAD: 0
        Transceiver: internal
        Auto-negotiation: on
        MDI-X: off (auto)
Cannot get wake-on-lan settings: Operation not permitted
        Current message level: 0x00000007 (7)
                               drv probe link
        Link detected: yes
```