

Übungsblatt 6 (Musterlösung)

1. UDP - Prüfsumme

a) Berechnen Sie die UDP-Prüfsumme, welche sich über die folgenden drei 16Bit-Wörter ergibt:

- 1011 1011 0001 0000
- 1001 1001 0101 1001
- 0101 1011 0101 1001

b) Wie kann der Empfänger feststellen, dass im UDP-Paket ein Fehler aufgetreten ist?

c) Wie werden fehlerhafte Pakete auf der Empfängerseite behandelt?

Lösung

a) Die Idee ist, dass man die Bitfolgen aufaddiert. Überträge werden hinten wieder aufaddiert:

1011 1011 0001 0000	1. Wort
1001 1001 0101 1001	2. Wort
0101 0100 0110 1001	Zwischensumme (mod 2)
1	Übertrag
0101 0100 0110 1010	Neue Zwischensumme
0101 1011 0101 1001	3. Wort
1010 1111 1100 0011	Gesamtsumme
0101 0000 0011 1100	Einerkomplement = Prüfsumme

Beachten Sie: Zur eigentlichen UDP Prüfsummenberechnung wird nicht nur der Inhalt des UDP Paketes (also die Daten) herangezogen, sondern es werden auch weitere Informationen (u.a. Sender und Empfänger IP Adressen) verwendet.

- b) Alle empfangenen 16Bit Wörter inklusive Prüfsumme werden modulo 2 aufaddiert. Hier muss ebenfalls der Übertrag beachtet werden. Wenn das Ergebnis nicht gleich 1111 1111 1111 1111 ist, so wurde das Paket fehlerhaft empfangen.
- c) Hier kommt es auf die konkrete Implementierung an. Im Normalfall werden korrupte Pakete umgehend verworfen. Es gibt aber auch die Möglichkeit das Paket an die Applikation weiterzuleiten, es dabei aber als „fehlerhaft“ zu markieren.

2. TCP Congestion Window

Es wird eine TCP-Reno-Verbindung betrachtet (mit Fast Recovery), die schon seit einer unbekannten Anzahl x von Transmission-Rounds (Übertragungsrunden) besteht. Der folgende Graph gibt die Größe des variablen Congestion Windows in den Transmission-Rounds x bis $x+17$ an. Zur Erinnerung: In TCP Reno werden Timeouts anders behandelt als mehrfache ACKs.

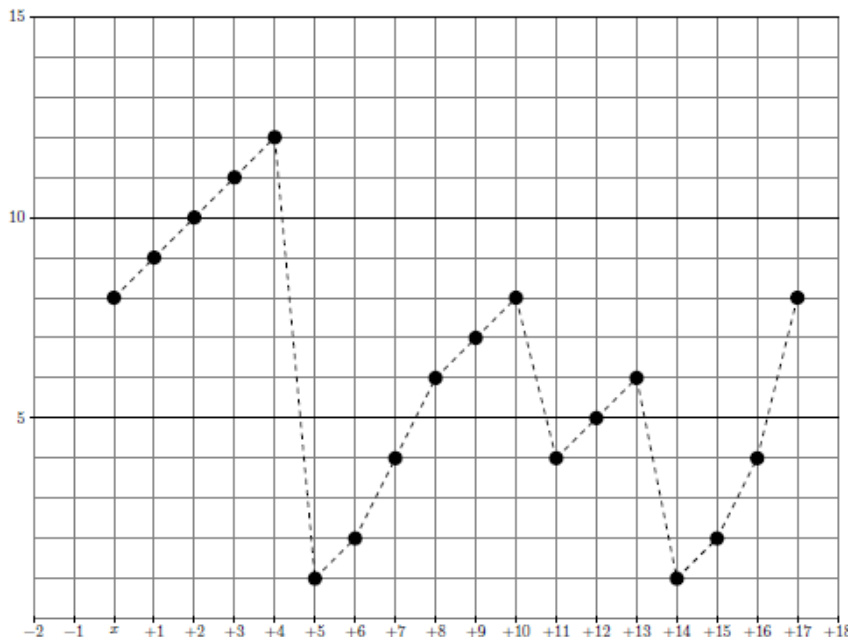


Abbildung 1: TCP-Verlauf

Hinweis: Der Graph enthält einen Fehler (irgendwann nach Übertragungsrunde $x+10$). Beachten Sie diesen Fehler erst ab Teilaufgabe (e).

Beantworten Sie folgende Fragen:

- Wurden die Segmentverluste in den Runden $x+4$, $x+10$ und $x+13$ durch Timeouts oder durch mehrfache ACKs erkannt?
- Welchen Wert hat der Threshold in Runde $x+6$?
- Identifizieren Sie alle Zeitintervalle zwischen den Runden x und $x+13$, in denen TCP Slow-Start aktiv ist.
- Wie hoch ist der durchschnittliche Datendurchsatz in den Runden x bis einschließlich $x+3$? Bitte geben Sie auch Ihren Rechenweg an.
- Der Graph enthält einen Fehler. Lokalisieren und korrigieren Sie diesen direkt in der Grafik bis Runde $x+17$. Gehen Sie dabei davon aus, dass nach der Runde des Fehlers kein Segmentverlust mehr stattfindet. Sollten Sie den oberen Rand der Grafik erreichen, beenden Sie bitte den Korrekturvorgang.
- Wie groß ist das Congestion Window in Runde $x-1$, wenn vor Runde x (seit Beginn der Übertragung) kein Segmentverlust aufgetreten ist, und TCP in Runde x in die Congestion-Avoidance-Phase wechselt?
- Wie groß ist der Threshold in Runde $x-2$?
- Warum ist in Runde $x+8$ die Größe des cwnd nur 6 und nicht 8?

Lösung

- a) x+4: Timeout, da das cwnd auf 1 gesetzt wird
 x+10: Mehrfacher ACK, da das cwnd halbiert wird
 x+13: Timeout, da das cwnd auf 1 gesetzt wird
- b) $12/2 = 6$. Dies kommt daher, da das Maximum vorher 12 war und nach einem Timeout der Threshold halbiert wird.
- c) x+5 bis x+8, da nur hier eine exponentielle Steigung stattfindet. Für den Bereich x bis x+1 lässt sich keine definitive Aussage treffen.
- d) Hier gilt es, die übertragenen Daten zu addieren und dann in ein Verhältnis zu setzen. Damit gilt: $d = 8+9+10+11 = 38$. Die Datenrate L bestimmt sich nun folgendermaßen:

$$L = \frac{\text{Menge}}{\text{Zeit}} = \frac{38 \cdot \text{MSS}}{4 \cdot \text{RTT}} = 9.5 \cdot \frac{\text{MSS}}{\text{RTT}}$$

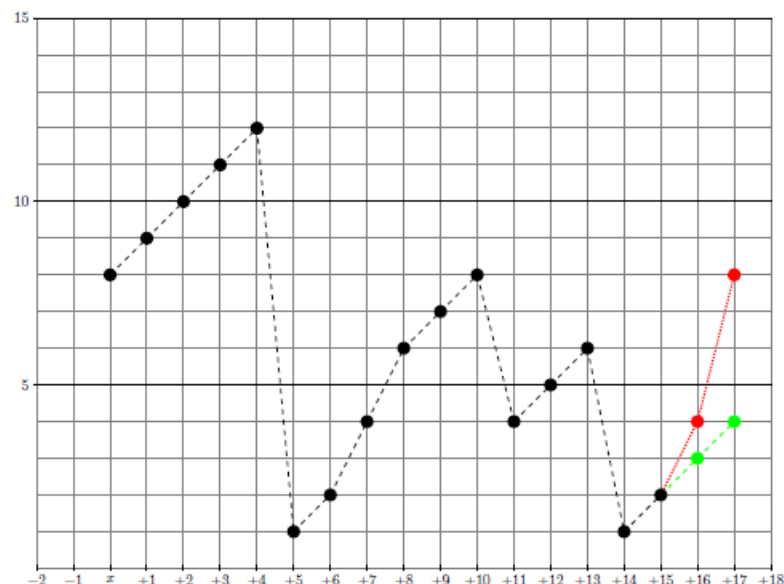


Abbildung 2: TCP-Verlauf Lösung

- e) Der Fehler liegt ab Runde x+15, da der Threshold auf 3 gesetzt wird. In anderen Worten wird ab 3 nur noch mit linearer Steigerung gearbeitet. Das ist bildlich dargestellt.
- f) Da sich in der Slow-Start-Phase die Senderate verdoppelt, folgt, dass cwnd 4 in Runde x-1 war.
- g) Der Threshold muss dementsprechend 8 gewesen sein, da ab Runde x TCP nach Aufgabenstellung in Congestion-Avoidance-Phase ist. Wenn wir diese Information nicht hätten, könnte der Threshold auch 9 gewesen sein.
- h) Da der Threshold auf 6 steht.

Merkregel: Der neue Threshold ergibt sich durch die aktuelle Größe des Congestion Windows, geteilt durch 2:

$$\text{threshold}_{i+1} \leftarrow \lfloor \frac{\text{cwnd}_i}{2} \rfloor$$

Wenn ein mehrfaches Acknowledgement auftritt, wird cwnd auf den neuen Threshold gesetzt:

$$\text{cwnd}_{i+1} \leftarrow \text{threshold}_{i+1}$$

Wenn der Paketverlust durch einen Timeout entsteht, wird jedoch im Gegensatz zum mehrfachen ACK cwnd auf 1 gesetzt. Solange $\text{cwnd} \leq \text{threshold}$ gilt, befindet sich TCP in der Slow-Start-Phase:

$$\text{cwnd}_{i+1} \leftarrow 2 \cdot \text{cwnd}_i$$

Das entspricht $2^{(i-1)}$ mit i Element der natürlichen Zahlen, die momentane Runde nach Erkennen des Paketverlusts ausdrückt.

Es wird allerdings bei Erreichen des Thresholdes abgeschnitten:

$$\text{cwnd}_{i+1} \leftarrow \min(2^{i-1}, \text{threshold})$$

Das sieht man im Graphen an den Runden x+8 und x+16.

3. Wireshark

→ Siehe Übung

Viel Erfolg !!!