Lehrstuhl für Betriebssysteme
Fakultät für Informatik
Technische Universität München

Grundlagen Rechnernetze und Verteilte Systeme

IN0010, SoSe 2018

Übungsblatt 9

18. Juni – 22. Juni 2018

Hinweis: Mit * gekennzeichnete Teilaufgaben sind ohne Lösung vorhergehender Teilaufgaben lösbar.

In dieser Woche werden in den Übungen auch die Teilaufgaben 1c) und 4e) der Midterm besprochen, weswegen dieses Blatt nur eine Aufgabe umfasst.

Aufgabe 1 Schiebefensterprotokolle

Wir betrachten ein Sliding-Window-Verfahren, dessen Sende- und Empfangsfenster $w_s = w_r = 2$ beträgt. Der Sequenznummernraum sei $\mathcal{S} = \{0, 1\}$. Die Fehlerbehandlung erfolge analog zu Go-Back-N. Abbildung 1 zeigt eine Datenübertragung, wobei die Blitze für durch Störungen verlorengegangene Segmente stehen. Die beiden ersten ACKs erreichen also nicht den Sender.

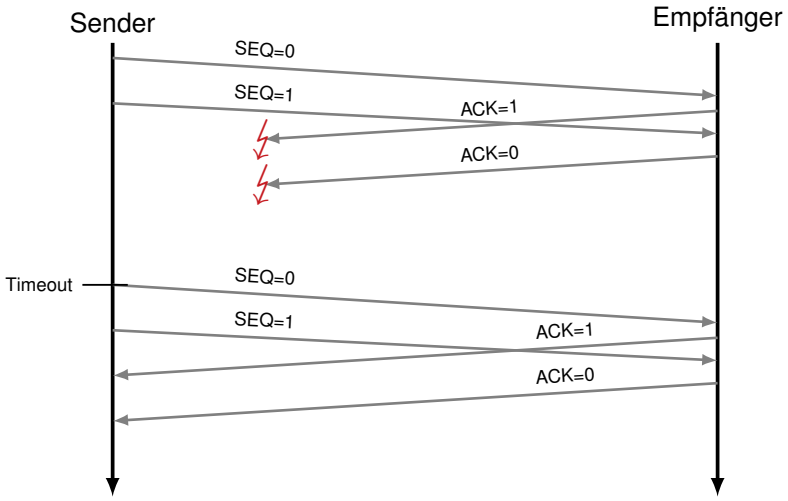

Abbildung 1: Modifiziertes Alternating-Bit-Protocol

a)* Welches Problem tritt in dem Beispiel bei der Übertragung auf?

Der Empfänger leitet nach Erhalt der ersten beiden Segmente diese an die nächsthöhere Schicht weiter. Er weiß nach dem Senden der ersten beiden ACKs nicht, dass diese den Sender nicht erreicht haben. Der Sender wird nach einem Timeout diese ersten beiden Segmente wiederholen. Diese tragen dieselben Sequenznummern wie die ersten beiden Segmente. Leider erwartet der Empfänger aber zwei **neue** Segmente mit eben diesen Sequenznummern. Der Empfänger ist also nicht in der Lage zu unterscheiden, ob es sich um eine Wiederholung oder um zwei neue Segmente handelt. Er wird daher auch diese beiden Segmente an die nächsthöhere Schicht weiterleiten, welche nun Daten **doppelt** erhalten hat.

b) Passen Sie $\mathcal{S}$ an, so dass das Verfahren korrekt funktionieren kann.

Da das Wiederholungsverfahren Go-Back-N ist, akzeptiert der Empfänger jeweils nur das nächste erwartete Segment (unabhängig davon, dass sein Empfangsfenster größer ist). In diesem Fall reicht bereits der Sequenznummernraum $\mathcal{S} = \{0, 1, 2\}$ aus, da auf diese Weise stets ein „Schutzabstand“ von einer Sequenznummer besteht.

Im Folgenden betrachten wir die beiden Verfahren Go-Back-N und Selective Repeat. Die Sequenznummern $s \in \mathcal{S}$ haben eine Länge von 4 bit. Beantworten Sie die folgenden Fragen **sowohl für Go-Back-N als auch**

Selective Repeat.

c)* Wie viele unbestätigte Segmente darf der Sender jeweils senden, um eine gesicherte Verbindung zu realisieren? Begründen Sie Ihre Antwort anhand von Beispielen. (Hinweis: Denken Sie an in möglichst ungünstigen Momenten verlorene Bestätigungen)

Bezeichne w_S das Sendefenster. Dann gilt allgemein:

- **Go-Back-N:**
  Der Empfänger akzeptiert bei Go-Back-N grundsätzlich immer nur das nächste erwartete Segment. Segmente, die *out-of-order* ankommen, werden ignoriert. Der ungünstigste Fall für Go-Back-N ist der, dass alle Segmente erfolgreich übertragen werden, dann aber alle Bestätigungen auf dem Weg zum Sender verloren gehen. Dieser Fall ist in Abbilung 1 dargestellt. Abhilfe kann geschaffen werden, indem stets ein Segment weniger gesendet wird als insgesamt Sequenznummern zur Verfügung stehen. Aus diesem Grund muss für das Sendefenster bei Go-Back-N stets $w_s \leq |\mathcal{S}| - 1$ gelten.
- **Selective Repeat:**
  Der ungünstigste Fall für Selective Repeat besteht darin, dass w_s Segmente erfolgreich übertragen werden, anschließend aber alle Bestätigungen verloren gehen. Sei x die Sequenznummer des ersten Segments. In diesem Fall wird der Empfänger – da er ja alle Segmente erfolgreich erhalten hat – sein Empfangsfenster um w_s weiterschieben. Der Sender denkt aber, dass alle Segmente verlorengegangen sind. Er wird aus diesem Grund die Segmente mit den Sequenznummern $x, x+1, \ldots, x+w_s-1$ wiederholen. Die Bedingung ist nun, dass keine der Sequenznummern der **wiederholten** Segmente in das aktuelle Empfangsfenster des Empfängers fallen darf. Andernfalls würde der Empfänger eine Wiederholung als neues Segment akzeptieren.

  Für $w_s = 4$ und $|\mathcal{S}| = 8$ sieht man nun an einem Beispiel schnell, dass alles funktioniert. Wählt man hingegen $w_s = 5$, tritt ein Konflikt auf.
  Für $w_s = 3$ und $|\mathcal{S}| = 7$ ist ebenfalls alles in Ordnung. Wählt man hier jedoch $w_s = 4$, so tritt wieder ein Konflikt auf.

  Allgemein gilt also für das Sendefenster

$$w_s \leq \left\lfloor \frac{|\mathcal{S}|}{2} \right\rfloor .$$

  Hinweis:
  Nimmt man an, dass es keine kummulativen Bestätigungen gibt, so gibt es einen weiteren Fall, der zum selben Ergebnis führt: Es geht die Bestätigung des ersten Segments verloren, die übrigen erreichen jedoch den Sender. Beispiel: $|\mathcal{S}| = 7$, $w_s = 4$. Der Sender sendet Segmente mit den Sequenznummer $0, 1, 2, 3$. Der Empfänger erwartet also als nächstes die Sequenznummern $4, 5, 6, 0$. Der Sender wiederholt das Segment mit Sequenznummer 0, was vom Empfänger aber fälschlicherweise als neues Segment interpretiert wird.

d)* Begründen Sie, welche oberen und unteren Grenzen für das Empfangsfenster des Empfängers bei den beiden Verfahren jeweils sinnvoll sind.

Bei Go-Back-N reicht prinzipiell ein Empfangsfenster der Größe $w_r = 1$, da stets nur das nächste erwartete Segment akzeptiert wird.

Bei Selective Repeat hingegen muss das Empfangsfenster mind. so groß wie das Sendefenster sein und darf natürlich nicht größer als etwa die Hälfte des Sequenznummernraums sein, also

$$w_s \leq w_r \leq \left\lfloor \frac{|\mathcal{S}|}{2} \right\rfloor .$$

Andernfalls verwirft der Empfänger u. U. Segmente, die nicht in der richtigen Reihenfolge eintreffen und infolge der zu geringen Größe des Empfangsfensters nicht in selbiges hineinfallen.

e)* Für eine praktische Implementierung benötigt der Empfänger einen Empfangspuffer. Wie groß sollte dieser bei den beiden Verfahren jeweils gewählt werden?

Unabhängig vom Verfahren sollte der Empfangspuffer stets die Größe des maximal erlaubten Sendefensters sein.

Bei Selective Repeat leuchtet das ein, da hier Segmente tatsächlich auf der Transportschicht zwischengespeichert werden müssen, bis fehlende Segmente eingetroffen sind.

Bei Go-Back-N hingegen könnte man argumentieren, dass Segmente ohnehin in der richtigen Reihenfolge eintreffen müssen und diese daher auch sofort an höhere Schichten weitergeleitet werden können. Dies trifft nur bedingt zu, denn die Verarbeitungsgeschwindigkeit des Empfängers könnte nicht ausreichen, um eintreffende Segmente schnell genug weiterzuleiten.

Unabhängig von der (etwas philosophischen) Frage, wie groß Puffer bei konkreten Implementierung zu wählen sind, sollten Sie sich den Unterschied zwischen dem Empfangsfenster und einem etwaigen Empfangspuffer klarmachen. Beispielsweise könnte die Größe des Empfangsfensters stets dem noch freien Speicher im Empfangspuffer entsprechen während das Sendefenster durch das Empfangsfenster nach oben beschränkt wird.