

## Tutorübung zur Vorlesung Grundlagen Rechnernetze und Verteilte Systeme Übungsblatt 8 (15. Juni – 19. Juni 2015)

**Hinweis:** Die mit \* gekennzeichneten Teilaufgaben sind ohne Kenntnis der Ergebnisse vorhergehender Teilaufgaben lösbar.

### Aufgabe 1 TCP und Long Fat Networks

In dieser Aufgabe betrachten wir sog. *Long Fat Networks*. Darunter versteht man Verbindungen, welche zwar eine hohe Übertragungsrate aber insbesondere auch eine hohe Verzögerung aufweisen. Beispiele dafür sind u. a. Satellitenverbindungen in Folge der hohen Ausbreitungsverzögerungen. Wir wollen insbesondere die Auswirkungen auf die TCP-Staukontrolle untersuchen.

a)\* Bei TCP wird das Sendefenster in Abhängigkeit des Empfangsfensters sowie des Staukontrollfensters gewählt. Wie lautet der genaue Zusammenhang?

$$w_s = \min(w_r, w_c)$$

Zwei Nutzer seien nun über einen geostationären Satelliten an das Internet mit hoher Übertragungsrate angebunden. Die RTT zwischen beiden Nutzern betrage 800 ms, die Übertragungsrate sei  $r = 24 \text{ Mbit/s}$ .

b)\* Wie groß muss das Sendefenster (gemessen in Byte) gewählt werden, damit kontinuierlich gesendet werden kann?

Das erste ACK kann frühestens nach einer RTT eintreffen, sofern man die Serialisierungszeiten vernachlässigt. Es ergibt sich also für das Sendefenster

$$w_s \geq \text{RTT} \cdot r = 800 \cdot 10^{-3} \text{ s} \cdot 24 \cdot 10^6 \frac{\text{bit}}{\text{s}} = 2,4 \text{ MB.}$$

c)\* Warum ist die Situation in Teilaufgabe b) ein Problem für die TCP-Flusskontrolle?

Da das Sendefenster als Minimum aus Empfangs- und Staukontrollfenster gewählt wird und der Empfänger dem Sender sein Empfangsfenster über das Receive-Window-Feld mitteilt, welches auf 16 bit beschränkt ist, ist auch das Sendefenster auf einen Maximalwert von  $(2^{16} - 1) \text{ B} = 65535 \text{ B}$  beschränkt. Wir bräuchten laut Teilaufgabe b) allerdings ein Sendefenster der Größe  $2,4 \cdot 10^6 \text{ B}$ .

d)\* Lesen Sie Sektion 2 von RFC 1323 (<http://www.ietf.org/rfc/rfc1323.txt>, siehe Anhang). Beschreiben Sie die Lösung für das Problem aus Teilaufgabe c).

Wir benötigen die Option *TCP-Windows-Scaling*, welche dafür sorgt, dass das Receive-Window mit  $2^x$  skaliert wird. Das Feld „shift.cnt“ der TCP-Windows-Scaling-Option gibt den Exponenten  $x$  an.

e) Bestimmen Sie den korrekten Wert für das `shift.cnt`-Feld der TCP-Window-Scaling-Option. Berücksichtigen Sie dabei, dass es sich bei TCP um ein Selective-Repeat-Verfahren handelt.

Es muss gelten:

$$\begin{aligned} (2^{16} - 1) \cdot 2^x &\geq 2,4 \cdot 10^6 \\ x &\geq \text{ld} \left( \frac{2,4 \cdot 10^6}{2^{16} - 1} \right) \approx 5,19 \\ \Rightarrow x &= 6 \end{aligned}$$

### Erklärung:

Wir suchen den kleinsten Exponenten  $x$ , so dass das maximale Receive-Window größer als der in Teilaufgabe b) berechnete Wert von  $2,4 \cdot 10^6 \text{ B}$  ist. Das Receive-Window ist 16 bit breit, kann also maximal den Wert  $0\text{x}\text{ffff} = 2^{16} - 1$  annehmen. Dieser Wert muss also nun mit  $2^x$  skaliert werden.

Ein kurzer Blick auf die Größe des Sequenznummernraums von TCP ( $|\mathcal{S}| = 2^{32}$ , da SEQ- und ACK-Nummern 32bit lange Felder sind) zeigt, dass wir kein Problem wie in der Aufgabe "Schiebefensterprotokolle" bekommen.

f) Geben Sie den Header des ersten TCP-SYN-Pakets an, welches die Verbindung aufbaut. Verwenden Sie dazu die konkreten Zahlenwerte aus der Angabe. Ein TCP-Header ist zur Erinnerung nochmals in Abbildung 1 dargestellt. Dort finden sich auch zwei Vordrucke zur Lösung.

**Hinweis:** Es ist nicht notwendig, den Header binär auszufüllen. Machen Sie aber bitte deutlich, ob es sich um hexadezimale, dezimale oder binäre Darstellung der Zahlen handelt.

Siehe Vordruck. Die  $(6)_{10}$  des Data Offsets sagt aus, dass der Header eine Länge von  $6 * 32\text{bit}$  hat, die Nutzdaten also nach jener Anzahl von Bits beginnen.

Angenommen die Größe des Staukontrollfensters betrage derzeit die Hälfte des in Teilaufgabe b) berechneten Werts. Die MSS betrage 1500 B und die TCP-Verbindung befinde sich derzeit in der Congestion-Avoidance-Phase.

g) Wie lange dauert es, bis das Fenster die Leitung komplett ausnutzen kann?

**Hinweis:** Das Staukontrollfenster wird durch TCP-Window-Scaling nicht beeinflusst.

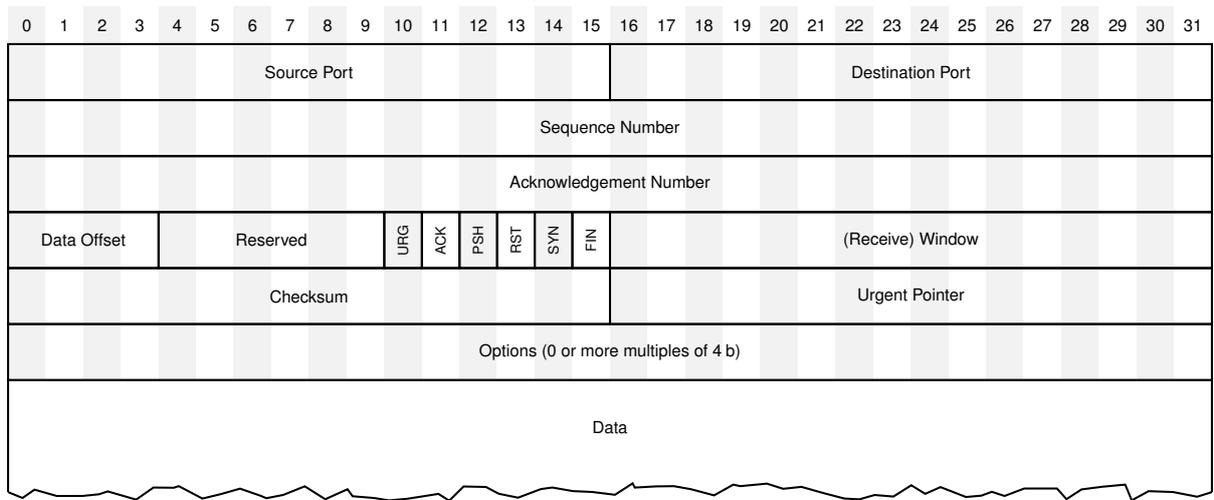
Das Fenster wird pro RTT um 1 MSS vergrößert. Folglich werden

$$\frac{1.2 \cdot 10^6 \text{ B}}{1500 \text{ B}} \cdot 0,8 \text{ s} = 640 \text{ s}$$

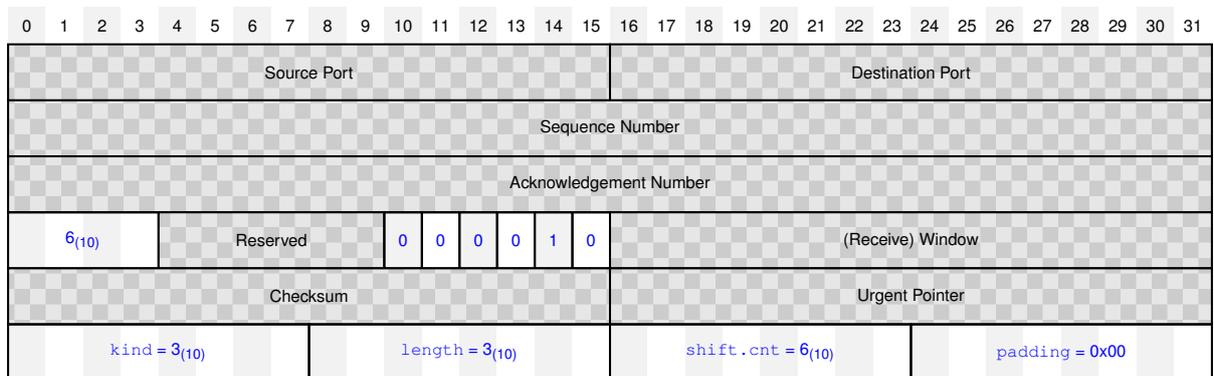
benötigt.

h) Ergibt sich aus dem Ergebnis von Teilaufgabe g) ein Problem?

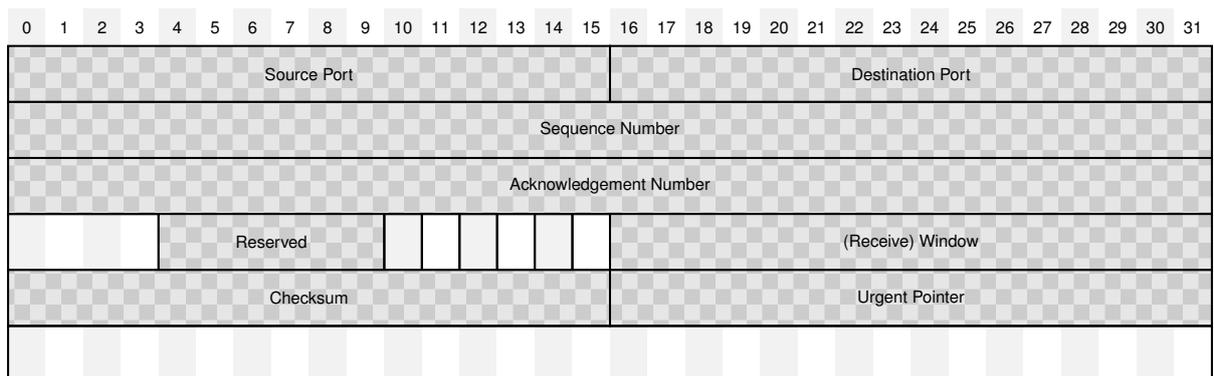
Ja. Es dauert mehr als 10 min bis TCP das Receive-Window wieder vollständig ausnutzt – viel zu lange.



(a) TCP-Header



(b) Vordruck



(c) Noch ein Vordruck, falls man sich verallt hat

Abbildung 1: TCP-Header und Vordrucke zur Lösung von Aufgabe 1

## 2. TCP WINDOW SCALE OPTION

### 2.1 Introduction

The window scale extension expands the definition of the TCP window to 32 bits and then uses a scale factor to carry this 32-bit value in the 16-bit Window field of the TCP header (SEG.WND in RFC-793). The scale factor is carried in a new TCP option, Window Scale. This option is sent only in a SYN segment (a segment with the SYN bit on), hence the window scale is fixed in each direction when a connection is opened. (Another design choice would be to specify the window scale in every TCP segment. It would be incorrect to send a window scale option only when the scale factor changed, since a TCP option in an acknowledgement segment will not be delivered reliably (unless the ACK happens to be piggy-backed on data in the other direction). Fixing the scale when the connection is opened has the advantage of lower overhead but the disadvantage that the scale factor cannot be changed during the connection.)

The maximum receive window, and therefore the scale factor, is determined by the maximum receive buffer space. In a typical modern implementation, this maximum buffer space is set by default but can be overridden by a user program before a TCP connection is opened. This determines the scale factor, and therefore no new user interface is needed for window scaling.

### 2.2 Window Scale Option

The three-byte Window Scale option may be sent in a SYN segment by a TCP. It has two purposes: (1) indicate that the TCP is prepared to do both send and receive window scaling, and (2) communicate a scale factor to be applied to its receive window. Thus, a TCP that is prepared to scale windows should send the option, even if its own scale factor is 1. The scale factor is limited to a power of two and encoded logarithmically, so it may be implemented by binary shift operations.

TCP Window Scale Option (WSopt):

Kind: 3 Length: 3 bytes

```
+-----+-----+-----+
| Kind=3 |Length=3 |shift.cnt|
+-----+-----+-----+
```

This option is an offer, not a promise; both sides must send Window Scale options in their SYN segments to enable window scaling in either direction. If window scaling is enabled, then the TCP that sent this option will right-shift its true receive-window values by 'shift.cnt' bits for transmission in SEG.WND. The value 'shift.cnt' may be zero (offering to scale, while applying a scale factor of 1 to the receive window).

This option may be sent in an initial <SYN> segment (i.e., a segment with the SYN bit on and the ACK bit off). It may also be sent in a <SYN,ACK> segment, but only if a Window Scale option was received in the initial <SYN> segment. A Window Scale option in a segment without a SYN bit should be ignored.

The Window field in a SYN (i.e., a <SYN> or <SYN,ACK>) segment itself is never scaled.