Lehrstuhl für Netzarchitekturen und Netzdienste
Fakultät für Informatik
Technische Universität München

Grundlagen Rechnernetze und Verteilte Systeme (IN0010)

Übungsblatt 10

28. Juni – 2. Juli 2021

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 1.1 zeigt eine Datenübertragung, wobei die Blitze für durch Störungen verlorengegangene Segmente stehen. Die beiden ersten ACKs erreichen also nicht den Sender.

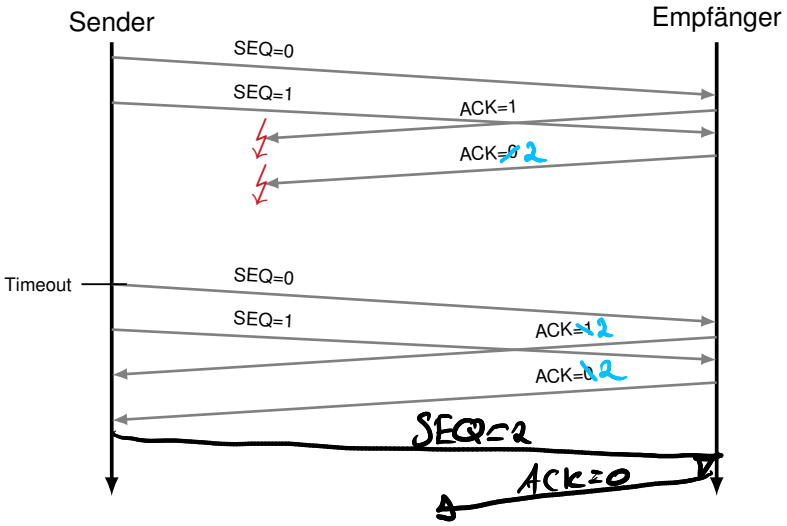


Abbildung 1.1: Modifiziertes Alternating-Bit-Protocol

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

Für den Empfänger sind beide Pakete erfolgreich angekommen, aber die ACKs gehen verloren.
Der Sender denkt, dass die Pakete nie erfolgreich angekommen sind, da er keine ACKs erhält.
=> Der Sender wiederholt die selben Pakete
Da diese aber die selben Sequ. Nummern tragen, denkt der Empfänger es sind zwei neue Pakete.

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

Mehr Sequentnummern z.B: $\mathcal{S} := \{0,1,2\}$
Schiebefenstergröße

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)

4 bit $\hat{=}$ 2^4 Sequenznummern $= 16 \hat{=} S := \{0, \ldots, 15\}$

Go-Back-N:

- Es wird immer nur das akuelle Segment akzeptiert.
- Ungünstigste Fall:
  - ↳ Alle Segmente empfangen
  - ↳ Acks gehen verloren

$\Rightarrow$ Sendefenster $ws \leq N - 1$

Selective Repeat:

Analog zu Go-Back-N

$ws \leq \lfloor \frac{|S|}{2} \rfloor$ N

Das neue Sendefenster darf nicht in den wiederholten Bereich fallen.

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

Go-Back-N:

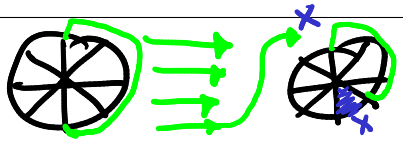
$wr = 1$

Da wir immer nur das nächste Paket akzeptieren.

Selective Repeat:

$ws \leq wr \leq \lfloor \frac{|S|}{2} \rfloor$

Da sonst Segmente verworfen werden. ($wr < ws$)



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

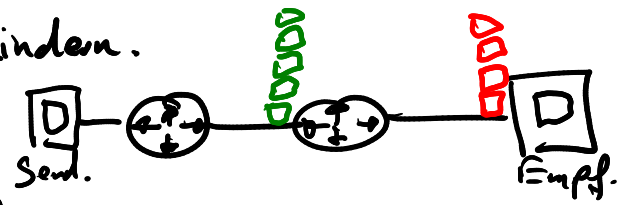
Mindestens so groß, wie das max. Sendefenster.

Aufgabe 2 Fluss- und Staukontrolle bei TCP

Das im Internet am weitesten verbreitete Transportprotokoll ist TCP. Dieses implementiert Mechanismen zur Fluss- und Staukontrolle.

a)* Diskutieren Sie die Unterschiede zwischen Fluss- und Staukontrolle. Welche Ziele werden mit dem jeweiligen Mechanismus verfolgt?

* Flusskontrolle:
  ⇒ Überlast beim Empfänger verhindern.
  ↳ Empfänger sendet updates
* Staukontrolle:
  ⇒ Überlast im Netz verhindern
  ↳ Viele verlorene Pakete



b) Ordnen Sie die folgenden Begriffe jeweils der TCP-Fluss- bzw. Staukontrolle zu:

- Slow-Start
- Empfangsfenster
- Congestion-Avoidance
- Multiplicative-Decrease

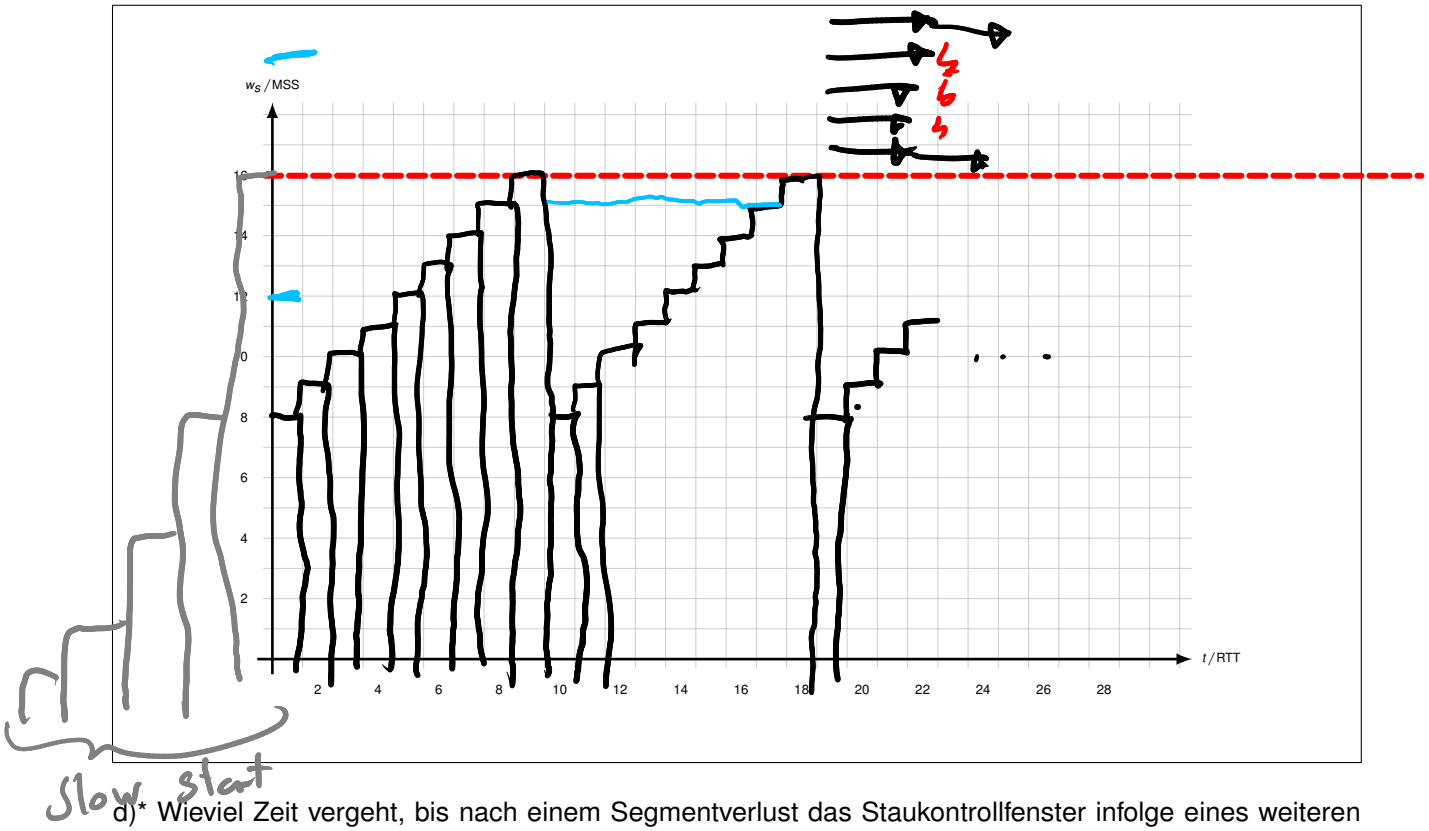
Flusskontrolle	Staukontrolle
- Empfangsfenster	- Slow Start ↳ Exp. Wachstum der #, der gesendeten Pakete - Congestion-Avoidance ↳ wc um $\frac{1}{wc}$ erhöhen bei erfolgreichem Senden - Mult. Decrease ↳ Halbieren des wc bei Paketverlust.

Zur Analyse der mit TCP erzielbaren Datenrate betrachten wir den Verlauf einer zusammenhängenden Datenübertragung, bei der die Slow-Start-Phase bereits abgeschlossen ist. TCP befinde sich also in der Congestion-Avoidance-Phase. Wir bezeichnen die einzelnen Fenster wie folgt:

- Sendefenster W_s, $|W_s| = w_s$
- Empfangsfenster W_r, $|W_r| = w_r$
- Staukontrollfenster W_c, $|W_c| = w_c$

Wir gehen davon aus, dass das Empfangsfenster beliebig groß ist, so dass das Sendefenster allein durch das Staukontrollfenster bestimmt wird, d. h. $W_s = W_c$. Es treten keinerlei Verluste auf, solange das Sendefenster kleiner als ein Maximalwert x ist, also $w_s < x$.
Wird ein vollständiges Sendefenster bestätigt, so vergrößert sich das aktuell genutzte Fenster um genau 1 MSS. Hat das Sendefenster den Wert x erreicht, so geht genau eines der versendeten TCP-Segmente verloren. Den Verlust erkennt der Sender durch mehrfachen Erhalt derselben ACK-Nummer. Daraufhin halbiert der Sender das Staukontrollfenster, bleibt aber nach wie vor in der Congestion-Avoidance-Phase, d. h. es findet kein erneuter Slow-Start statt. Diese Vorgehensweise entspricht einer vereinfachten Variante von TCP-Reno (vgl. Vorlesung).
Als konkrete Zahlenwerte nehmen wir an, dass die maximale TCP-Segmentgröße (MSS) 1460 B und die RTT 200 ms beträgt. Die Serialisierungszeit von Segmenten sei gegenüber der Ausbreitungsverzögerung vernachlässigbar klein. Segmentverlust trete ab einer Sendefenstergröße von $w_s \geq x = 16$ MSS auf.

c)* Erstellen Sie ein Schaubild, in dem die aktuelle Größe des Sendefenster w_s gemessen in MSS über der Zeitachse t gemessen in RTT aufgetragen ist. In Ihrem Diagramm soll zum Zeitpunkt $t_0 = 0$ s gerade die Sendefenstergröße halbiert worden sein, also $w_s = x/2$ gelten. Zeichnen Sie das Diagramm im Zeitintervall $t = \{0, \ldots, 27\}$.



d)* Wieviel Zeit vergeht, bis nach einem Segmentverlust das Staukontrollfenster infolge eines weiteren Segmentverlusts wieder reduziert wird?

Chetsheet

e)* Bestimmen Sie allgemein die durchschnittliche Verlustrate θ. Hinweis: Da das Verhalten von TCP in diesem idealisierten Modell periodisch ist, reicht es aus, lediglich eine Periode zu betrachten. Setzen Sie die Gesamtzahl übertragener Segmente in Relation zur Anzahl verlorener Segmente (Angabe als gekürzter Bruch ist ausreichend).

f) Bestimmen Sie mit Hilfe der Ergebnisse aus den Teilaufgaben (c) und (e) die in der betrachteten TCP-Übertragungsphase durchschnittlich erzielbare Übertragungsrate in kB/s.
Hinweis: Verwenden Sie den exakten Wert (Bruch) aus Teilaufgabe e).

g)* Bis zu welcher Übertragungsrate könnten Sie mit UDP maximal über den Kanal senden, ohne einen Stau zu erzeugen? Berücksichtigen Sie, dass der UDP-Header 12 B kleiner als der TCP-Header ohne Optionen ist.

Aufgabe 3 Network Address Translation

In dieser Aufgabe soll die Weiterleitung von IP-Paketen (IPv4) bei Verwendung eines NAT-fähigen Routers betrachtet werden. Für die Zuordnung zwischen öffentlichen und privaten IP-Adressen verfügt ein NAT-fähiger Router über eine Abbildungstabelle, die die Beziehung zwischen lokalem und globalem Port speichert. Viele NAT-fähige Geräte speichern zusätzlich noch weitere Informationen wie die entfernte IP-Adresse oder die eigene globale IP-Adresse (z. B. wenn der Router mehr als eine globale IP besitzt). Davon wollen wir hier absehen.
Abbildung 3.1 zeigt die Netztopologie. Router R1 habe NAT aktiviert, wobei auf IF1 eine private und auf IF2 eine öffentliche IP-Adresse verwendet werde. Router R2 nutze kein NAT. PC2 habe bereits mit Server 2 kommuniziert, wodurch der Eintrag in der NAT-Tabelle von R1 entstanden ist (siehe Abbildung 3.1). Wählen Sie dort, wo Sie die Freiheit haben, sinnvolle Werte für die IP-Adressen und Portnummern.

a)* Geben Sie PC1 und Interface 1 von R1 eine passende IP-Adresse. Das Subnetz ist 10.0.0.0/24.

PC1: 10.0.0.1

R1 (IF1): 10.0.0.254

b)* PC1 baue nun eine HTTP-Verbindung zu Server 2 auf. Geben Sie die Felder für die Quell-IP, Ziel-IP, Quell-Port, Ziel-Port und TTL des IP- bzw. TCP-Headers für die Pakete an den drei markierten Stellen in Abbildung 3.1 an. Geben Sie außerdem neu entstehende Einträge in der NAT-Tabelle von R1 an.

Bitte in Abbildung 3.1 eintragen. Die Box dient nur dazu, Ihnen im Lösungsvorschlag eine Erklärung an dieser Stelle zukommen zu lassen!

c) Server 2 antworte nun PC1. Geben Sie in Abbildung 3.2 analog zur vorherigen Teilaufgabe die Header-Felder an den drei benannten Stellen sowie neu entstehende Einträge in der NAT-Tabelle von R1 an.

Bitte in Abbildung 3.1 eintragen. Die Box dient nur dazu, Ihnen im Lösungsvorschlag eine Erklärung an dieser Stelle zukommen zu lassen!

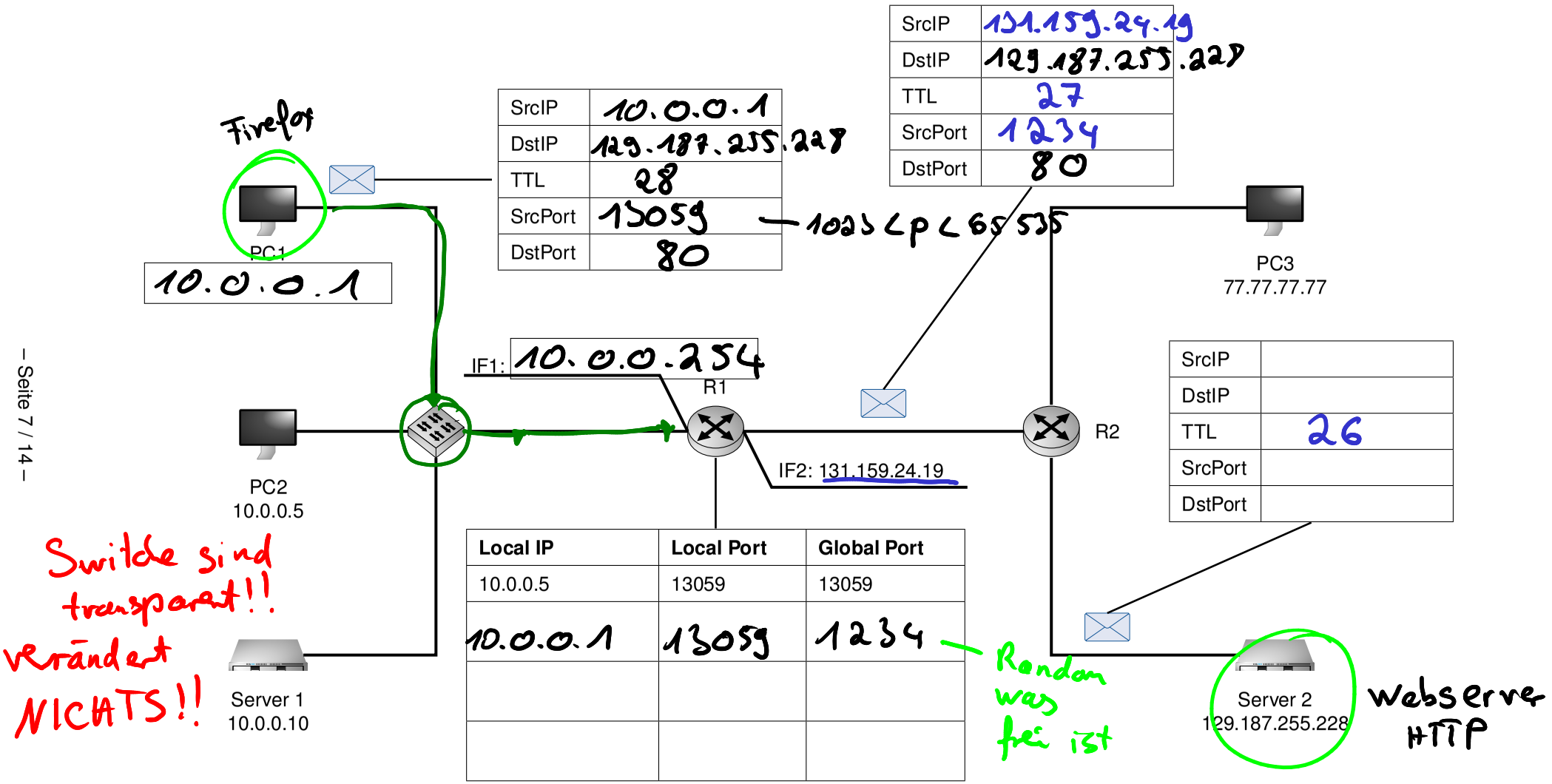


Abbildung 3.1: Lösungsblatt für Aufgabe 3a)/b)

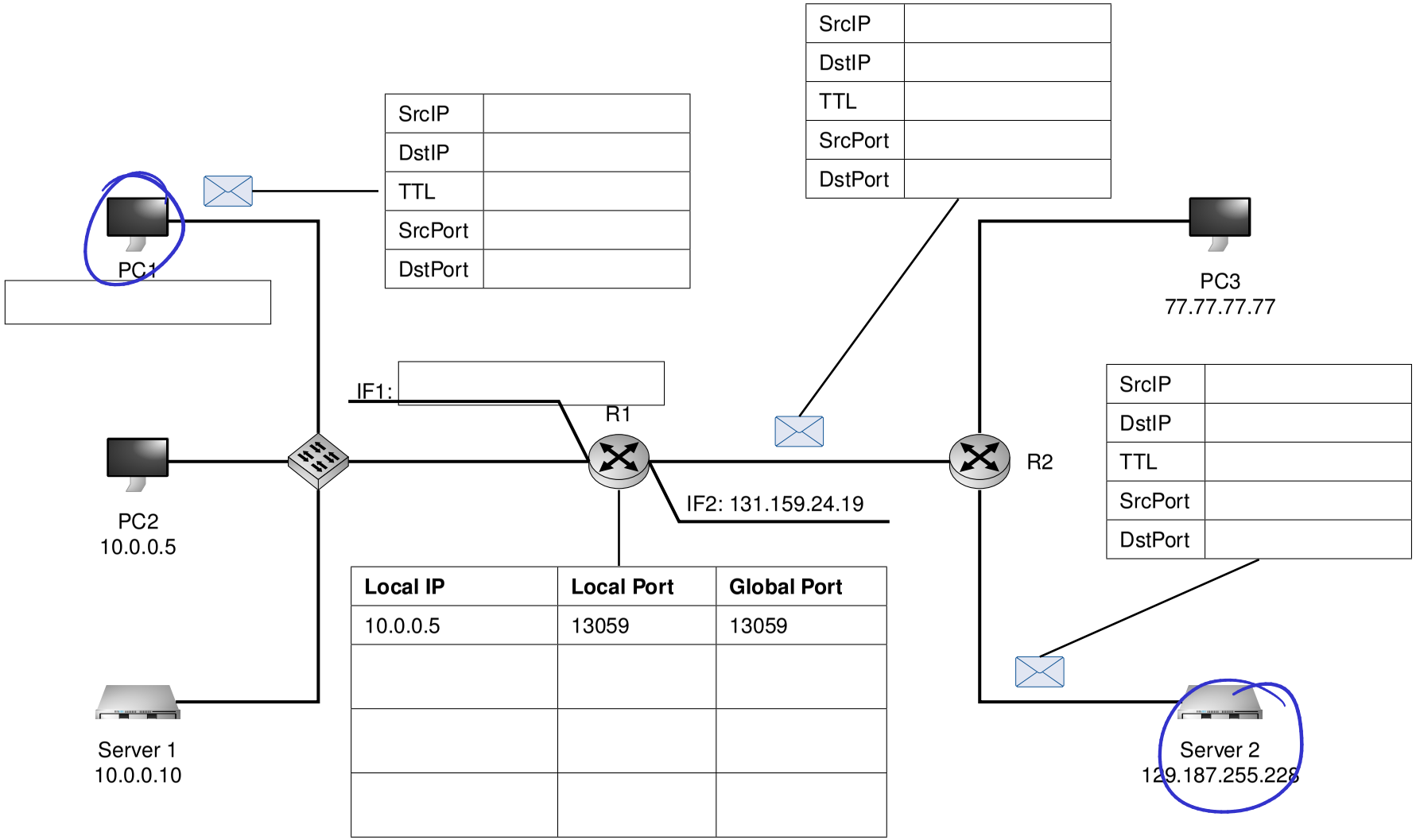


Abbildung 3.2: Lösungsblatt für Aufgabe 3c)

d)* Server 1 baut nun ebenfalls eine TCP-Verbindung zu Server 2 auf Port 80 auf. Dabei wählt er zufällig den Absender-Port 13059. Beschreiben Sie das am NAT auftretende Problem und wie dieses gelöst wird.

e)* R1 erhält von PC3 ein an `131.159.24.19:13059` adressiertes Paket. Wie wird R1 mit diesem Paket verfahren? Welche Probleme können sich daraus ergeben?

f) Ergibt sich für PC2 ein Problem, wenn dieser ein „zufälliges“ Paket mit TCP-Payload auf einem Port mit einer bestehenden Verbindung erhält?

g)* Welche weiteren Unterscheidungskriterien könnten von einem NAT-Router verwendet werden?

h)* Welches Problem tritt auf, wenn PC1 einen Echo Request an Server 2 sendet?

i) Beschreiben Sie eine mögliche Lösung für das in der vorherigen Teilaufgabe aufgetretene Problem.

j) Welches Problem ergibt sich, wenn ein NAT-Router ICMP TTL-Exceeded Nachrichten empfängt und an den Empfänger (Absender des auslösenden Pakets) weiterleiten möchte? Wie kann dieses Problem umgangen werden?

k)* Nun möchte PC3 eine HTTP-Verbindung zu Server 1 aufbauen. Kann dies unter den gegebenen Umständen funktionieren? (Begründung!)

l) Wie könnte das Problem unter Beibehaltung des NATs umgangen werden?

Aufgabe 4 TCP und Long Fat Networks (Hausaufgabe)

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?

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 $r = 24\,\text{Mbit}/\text{s}$.

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

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

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).

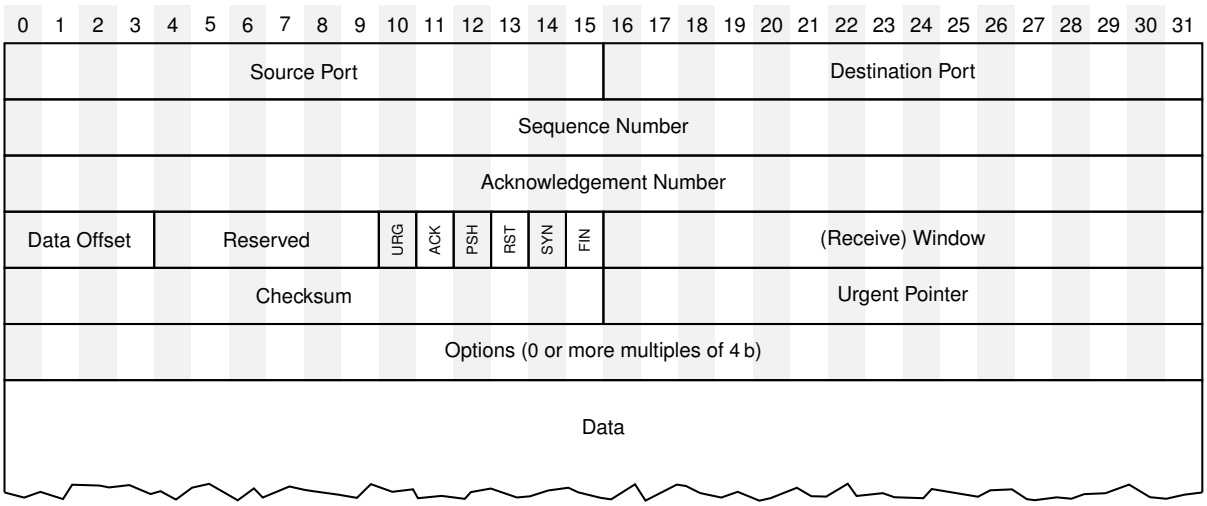
e) Bestimmen Sie den minimalen Wert für das `shift.cnt`-Feld der TCP-Window-Scaling-Option.

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 4.1 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.

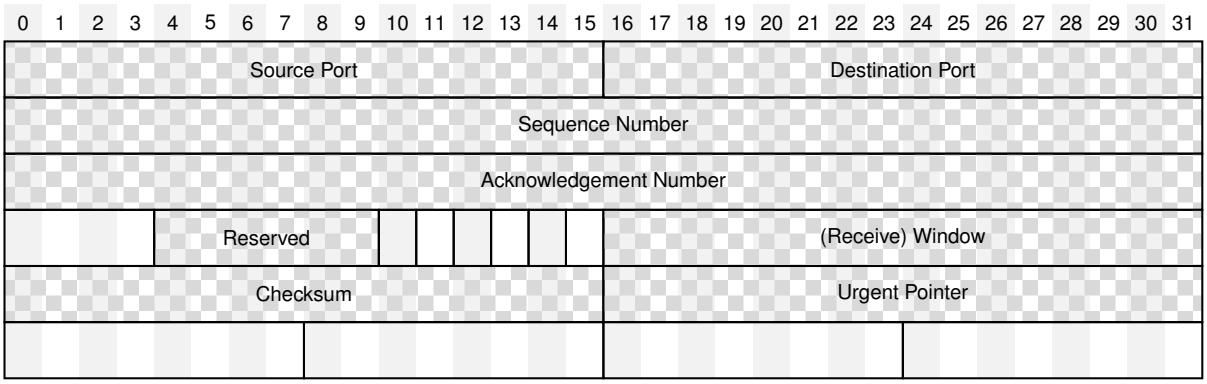
Angenommen die Größe des Staukontrollfensters betrage derzeit die Hälfte des in Teilaufgabe b) berechneten Werts. Die MSS betrage 1200 B und die TCP-Verbindung befinde sich derzeit in der Congestion-Avoidance-Phase.

g) Wie lange dauert es, bis das Fenster die Leitung komplett ausnutzen kann?
Hinweis: Das Staukontrollfenster wird durch TCP-Window-Scaling nicht beeinflusst.

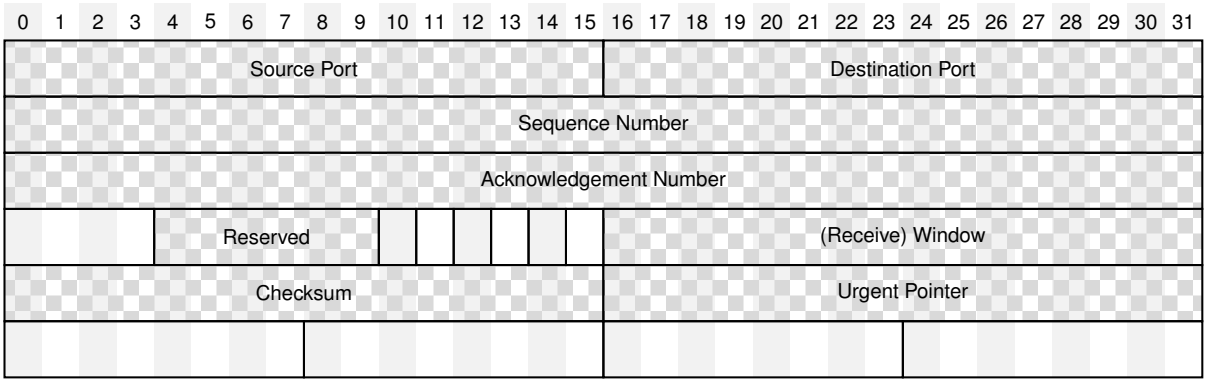
h) Ergibt sich aus dem Ergebnis von Teilaufgabe g) ein Problem?



(a) TCP-Header



(b) Vordruck



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

Abbildung 4.1: TCP-Header und Vordrucke zur Lösung von Aufgabe 4

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.)

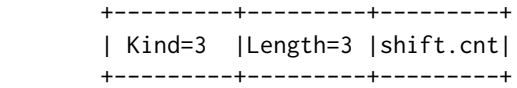
The maximum receive window, and therefore the scale factor, is determined by the maximum receive buffer space. In a typical modern implementation, this maximum buffer space is set by default but can be overridden by a user program before a TCP connection is opened. This determines the scale factor, and 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



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 op- tion 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.