

Rechnernetze und verteilte Systeme

Übungsblatt 6

Koenig.Noah@campus.lmu.de



Fenstergröße beim Sliding-Window-Verfahren (H)

Eine Sendestation verschickt PDUs an Satelliten, mit einer maximalen Übertragungsrate von 64 kBit/s (1 kBit sind 10^3 Bits). Die Größe der PDUs beträgt 512 Byte und jede PDU, den der Satellit empfängt, wird einzeln mit einer 8 Byte langen Antwort über einen separaten Rückkanal bestätigt (Quittung), ebenfalls mit einer maximalen Übertragungsrate von 64kBit/s. In diesem Szenario soll das *Sliding-Window-Verfahren* (Fenstertechnik) eingesetzt werden, um Flusssteuerung (*Flow Control*) zu ermöglichen.

Es sollen folgende Annahmen gelten:

- Die Signalverzögerung zwischen Sender und Satellit beträgt 270 ms.
- Es handelt sich um einen idealen Satellitenkanal, d.h. keine Nachricht geht verloren.

(a) Berechnen Sie jeweils die maximale effektive Übertragungsrate in kBit/s, wenn die Fenstergröße folgende Werte annimmt:

- i. 1 PDU
- ii. 7 PDUs
- iii. 15 PDUs

Geben Sie zusätzlich die Nutzungseffizienz des Kanals in Prozent an!

Visualisierung: <http://www.cip.ifi.lmu.de/~diefenthaler/window.html>

i. Fenstergröße 1 PDU

- maximale Übertragungsrate von 64 kBit/s = 64 000 Bit/s
- PDU (Größe 512 Byte = 4 096 Bit)
 - gesendet nach $4\,096 \text{ Bit} / 64\,000 \text{ Bit/s} = 64\text{ms}$
 - erhalten nach $64\text{ms} + 270\text{ms} = 334\text{ms}$
- Quittung (Größe 8 Byte = 64 Bit)
 - gesendet nach $64 \text{ Bit} / 64\,000 \text{ Bit/s} = 1\text{ms}$
 - erhalten nach $1\text{ms} + 334\text{ms} + 270\text{ms} = 605\text{ms}$
- **Übertragungsrate** ist $4\,096 \text{ Bit} / 0,605\text{s} \approx \mathbf{6770 \text{ Bit/s}}$
- **Nutzungseffizienz** ist $6\,770 \text{ Bit/s} / 64\,000 \text{ Bit/s} \approx \mathbf{10,58\%}$

ii. Fenstergröße 7 PDUs

- 7 PDUs
 - Größe 512 Byte * 7 = 28 672 Bit
 - gesendet nach 64ms * 7 = 448ms
- Nach 605ms kommt die erste Quittung an
→ Sender darf weiter senden
- **Übertragungsrate** ist $28\,672 \text{ Bit} / 0,605\text{s} \approx 47\,391 \text{ Bit/s}$
- **Nutzungseffizienz** ist $47\,391 \text{ Bit/s} / 64\,000 \text{ Bit/s} \approx 74,05\%$

iii. Fenstergröße 15 PDUs

- 15 PDUs sind gesendet nach $64\text{ms} * 15 = 960\text{ms}$
- Nach 605ms kommt die erste Quittung an
- Sender sendet zu diesem Zeitpunkt noch weiter, weil Fenster nicht voll
→ es wird dauerhaft gesendet
- **Übertragungsrate** ist **64 000 Bit/s**
- **Nutzungseffizienz** ist **100%**

(b) Berechnen Sie die minimale Fenstergröße, mit der die Nutzungseffizienz des Kommunikationskanals zum Satelliten 100% erreicht!

- 10 PDUs sind gesendet nach $64\text{ms} * 10 = 640\text{ms}$
- Nach 605ms kommt die erste Quittung an
→ es wird dauerhaft gesendet

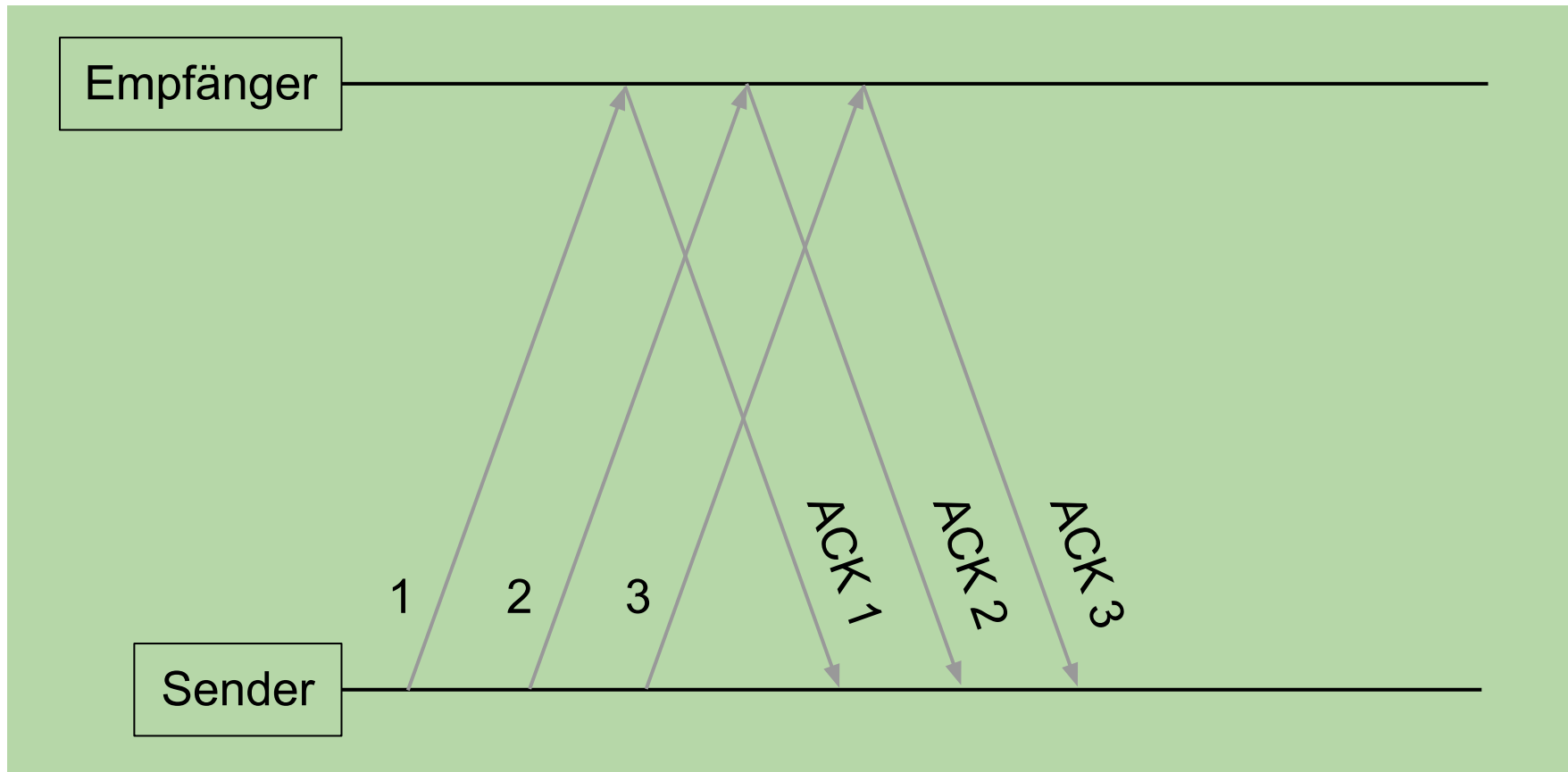
Sequenznummern 2 (mit Sendefenster) (H)

Im Übungsblatt der letzten Woche wurden *Sequenznummern* im Fall betrachtet, dass nach jeder Nachricht auf die Quittung gewartet werden muss. In dieser Aufgabe können mehrere Nachrichten auf einmal gesendet werden. Ansonsten gelten die selben Festlegungen, sofern nicht anders angegeben:

- Der Sender benutzt einen Zeitgeber/Timer, um die Zeit bis zum Erhalt der Quittung zu messen. Er wiederholt die Nachricht, wenn innerhalb eines Zeitintervalls (Timeout) von $800ms$ keine Quittung eingetroffen ist.
- Der Empfänger sendet nur positive Einzelquittungen.
- Der Empfänger kann Nachrichten nur verarbeiten, wenn sie in der richtigen Reihenfolge eintreffen.
- Alle Nachrichten (mit Nutzdaten, bzw. nur Quittung) sind gleich groß.
- Die Netzverzögerung beträgt $20ms$ und ist konstant für alle Nachrichten.
- Der Sender verwendet ein *Sendefenster* $w = 3$, das ihm erlaubt 3 Nachrichten auf einmal abzusen-
den, bevor die erste Quittung zurück ist.

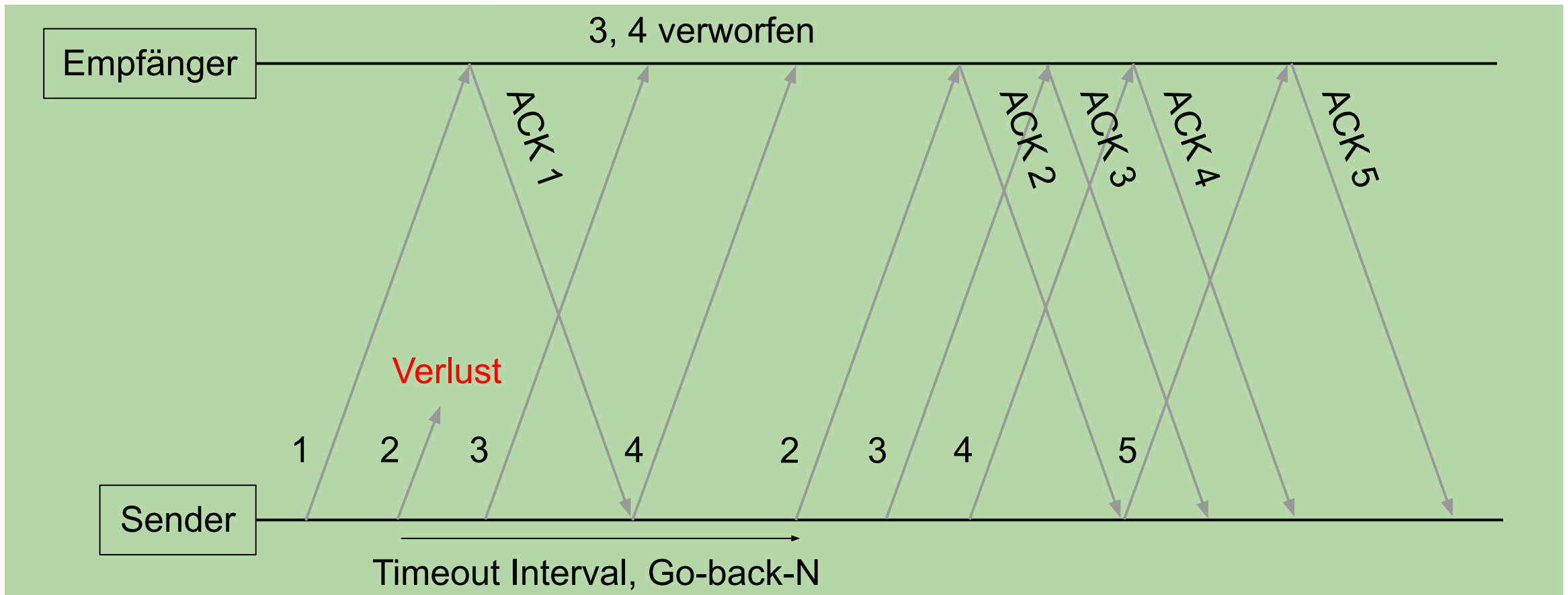
Sequenznummern 2 (mit Sendefenster)

- (a) Geben Sie ein Sequenzdiagramm für die fehlerfreie Übertragung dreier Nachrichten an. *Hinweis:* Bedenken Sie das festgelegte Sendefenster!



Sequenznummern 2 (mit Sendefenster)

- (b) Zeichnen Sie ein Sequenzdiagramm, in dem der Sender fünf Nachrichten sendet, aber die zweite Nachricht auf dem Weg zum Empfänger verloren geht. Führen Sie das Diagramm fort, bis alle Nachrichten erfolgreich übertragen wurden.



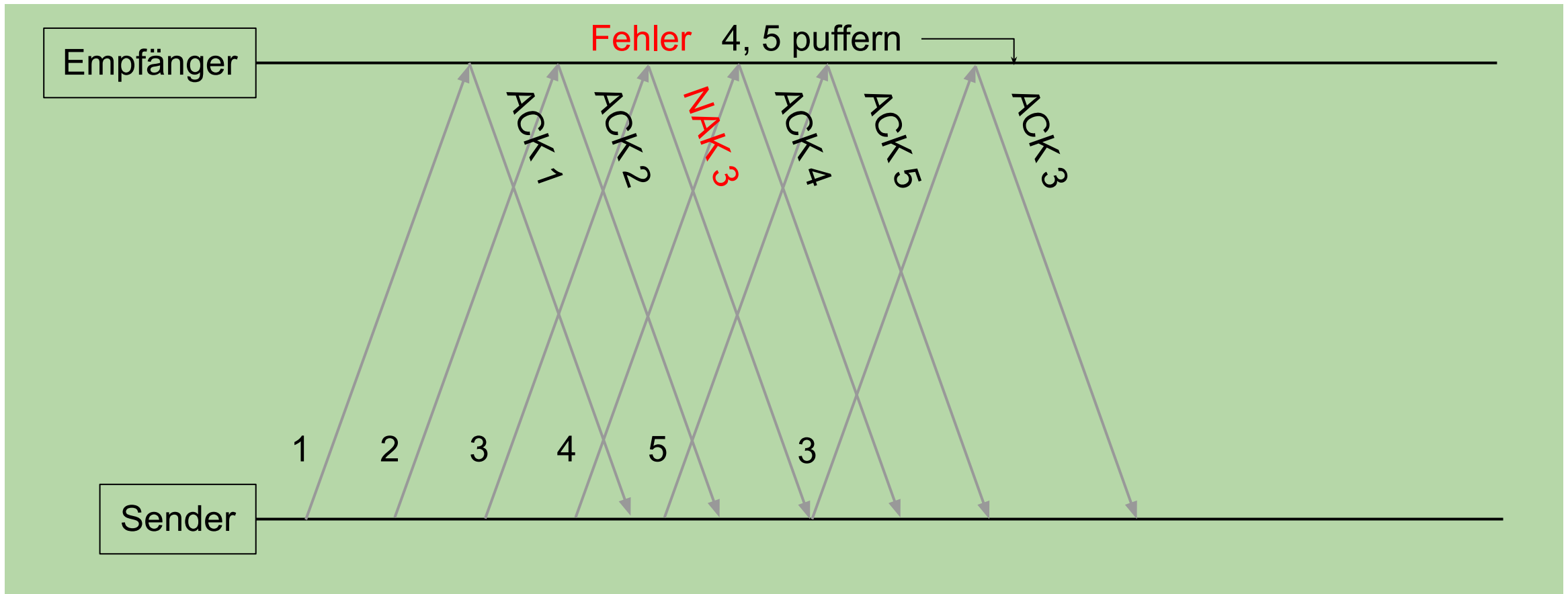
Sequenznummern 2 (mit Sendefenster)

Für die folgenden Teilaufgaben gelten neue Festlegungen:

- Der Sender verwendet ein größeres *Sendefenster* von $w = 5$.
- Der Empfänger sendet nun auch negative Quittungen, sollte eine Nachricht fehlerhaft ankommen.
- Der Empfänger kann korrekte Nachrichten, die in der falschen Reihenfolge ankommen, speichern und später verarbeiten.

Sequenznummern 2 (mit Sendefenster)

- (c) Zeichnen Sie ein Sequenzdiagramm, in dem der Sender fünf Nachrichten sendet, aber die dritte Nachricht fehlerhaft beim Empfänger ankommt.



(d) Welchen Vorteil haben negative Quittungen?

Fehler werden dem Sender aktiv mitgeteilt
→ Kein unnötiges Warten auf Timeout, Zeitersparnis

(e) Wie könnte man den Umgang mit positiven Quittungen optimieren, wenn der Empfänger mehrere Nachrichten quittieren soll?

Sammelquittung, die mehrere Quittungen zusammenfasst
→ weniger Quittungen

3-Way-Handshake und Sequenznummern bei TCP (H)

Protokollkonzepte wie *3-Way-Handshaking* und *Sequenznummern* sind Mechanismen für das Verbindungsmanagement und für die zuverlässige Kommunikation. Diese Mechanismen werden z.B. im *Transmission Control Protocol (TCP)* eingesetzt. Zur Bearbeitung dieser Aufgabe wird die Trace-Datei `trace3.pcap` bereitgestellt, die mitgeschnittenen TCP-Verkehr enthält.

Die Datei lässt sich z.B. mit dem freien Programm Wireshark¹ öffnen, das den mitgeschnittenen TCP-Verkehr grafisch aufbereitet anzeigen und filtern kann.

Wireshark stellt die PDUs verschiedener Schichten tabellarisch dar. In dieser Aufgabe soll es um TCP-PDUs (Segmente) gehen. Sie sind daran zu erkennen, dass in der *Protocol*-Spalte TCP steht. Die Informationen der TCP-PCI werden von Wireshark übersichtlich aufbereitet und je Segment dargestellt.

3-Way-Handshake und Sequenznummern bei TCP

(a) Identifizieren Sie die zum 3-Way-Handshaking Vorgang gehörenden Segmente in `trace3.pcap`.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.101	128.153.4.131	TCP	62	1226 → 22 [SYN] Seq= Win=64240 Len=0 MSS=1460 SACK_PERM
2	0.049886	128.153.4.131	192.168.0.101	TCP	60	22 → 1226 [SYN, ACK] Seq=0 Ack=1 Win=1460 Len=0 MSS=1460
3	0.049935	192.168.0.101	128.153.4.131	TCP	54	1226 → 22 [ACK] Seq= Ack=1 Win=64240 Len=0

SYN → SYN ACK → ACK

3-Way-Handshake in Segmenten 1 bis 3

(b) Identifizieren Sie die zum Verbindungsabbau gehörigen Segmente.

171	25.968266	192.168.0.101	128.153.4.131	TCP	54	1226 → 22	[FIN, ACK]	Seq=2974	Ack=5634	Win=63496	Len=0
172	26.024317	128.153.4.131	192.168.0.101	TCP	60	22 → 1226	[ACK]	Seq=5634	Ack=2975	Win=8760	Len=0
173	26.031474	128.153.4.131	192.168.0.101	TCP	60	22 → 1226	[FIN, ACK]	Seq=5634	Ack=2975	Win=8760	Len=0
174	26.031492	192.168.0.101	128.153.4.131	TCP	54	1226 → 22	[ACK]	Seq=2975	Ack=5635	Win=63496	Len=0

FIN → ACK → FIN → ACK

Verbindungsabbau in Segmenten 171 bis 174

3-Way-Handshake und Sequenznummern bei TCP

- (c) Berechnen Sie aus den Paketen des 3-Way-Handshake das so genannte *Round Trip Delay (RTD)*. Das ist die Zeit, die vom Versenden eines Segments bis zum Erhalt einer Antwort vergeht.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.0.101	128.153.4.131	TCP	62	1226 → 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM
2	0.049886	128.153.4.131	192.168.0.101	TCP	60	22 → 1226 [SYN, ACK] Seq=0 Ack=1 Win=1460 Len=0 MSS=1460
3	0.049935	192.168.0.101	128.153.4.131	TCP	54	1226 → 22 [ACK] Seq=1 Ack=1 Win=64240 Len=0

SYN → SYN ACK benötigt 0.049886 Sekunden

(d) Welche absoluten und relativen TCP-Sequenznummern besitzen diese Segmente?

1	0.000000	192.168.0.101	128.153.4.131	TCP
Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)				
Ethernet II, Src: Intel_53:87:d9 (00:07:e9:53:87:d9), Dst: LinksysG_8d:be:1d				
Internet Protocol Version 4, Src: 192.168.0.101, Dst: 128.153.4.131				
Transmission Control Protocol, Src Port: 1226, Dst Port: 22, Seq: 0, Len: 0				
Source Port: 1226				
Destination Port: 22				
[Stream index: 0]				
[Conversation completeness: Complete, WITH_DATA (31)]				
[TCP Segment Len: 0]				
Sequence Number: 0 (relative sequence number)				
Sequence Number (raw): 3332460122				

z.B. für Segment 1:

- absolut: 3332460122
- relativ: 0

- (e) In dem Mitschnitt werden (in der Standardkonfiguration von Wireshark) auch PDUs vom Protokoll der Anwendungsschicht (*SSHv2*) angezeigt. Nutzt dieses Protokoll auch TCP? Begründen Sie Ihre Vermutung kurz.

SSH (Anwendungsschichtprotokoll) nutzt aufgebaute TCP-Verbindung

TCP Sequenznummern (H)

Diese Aufgabe wird bereits zu Beginn der Vorlesung am 31.05.2024 besprochen.

Zwei Hosts A und B kommunizieren über eine TCP Verbindung. Host B hat bereits 126 Bytes von Host A vollständig empfangen und Host A sendet zwei weitere Segmente der Größen 80 sowie 40 Bytes. Die Sequenznummer des ersten Segments ist 127, der Quellport ist 302 und der Zielport ist 80. Host B sendet ein Acknowledgement immer, sobald es ein Segment von Host A empfangen hat.

(a) Wie lauten Sequenznummer, Quell- sowie Zielport des *zweiten* Segments von Host A an B?

- Erstes Segment ist erfolgreich angekommen
→ Sequenznummer ist $127 + 80 = 207$
- Host A sendet an Host B
→ Quellport 302, Zielport 80

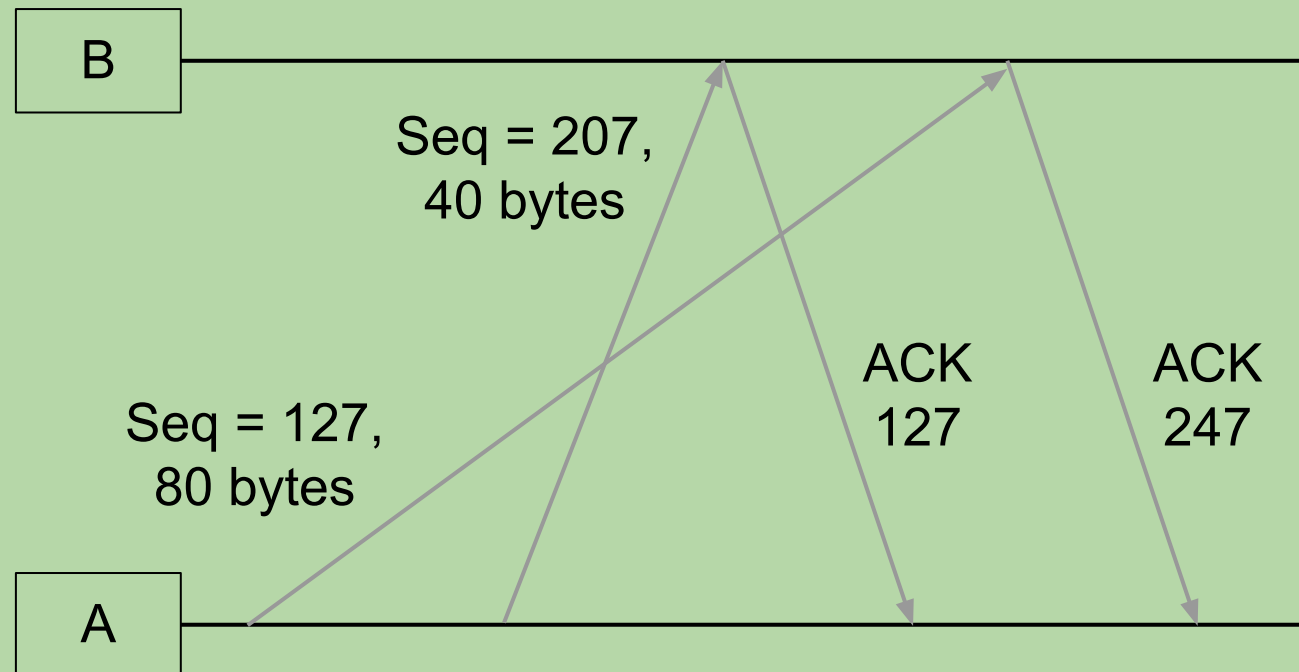
(b) Falls das erste Segment *vor* dem zweiten Segment bei B eintritt, wie lauten im ACK (Quittung) die ACK-Nr., Quell- und Zielpport?

- Erstes Segment ist erfolgreich angekommen, B erwartet Sequenznummer 207 → ACK-Nr. ist 207
- Host B sendet an Host A
→ Quellport 80, Zielpport 302

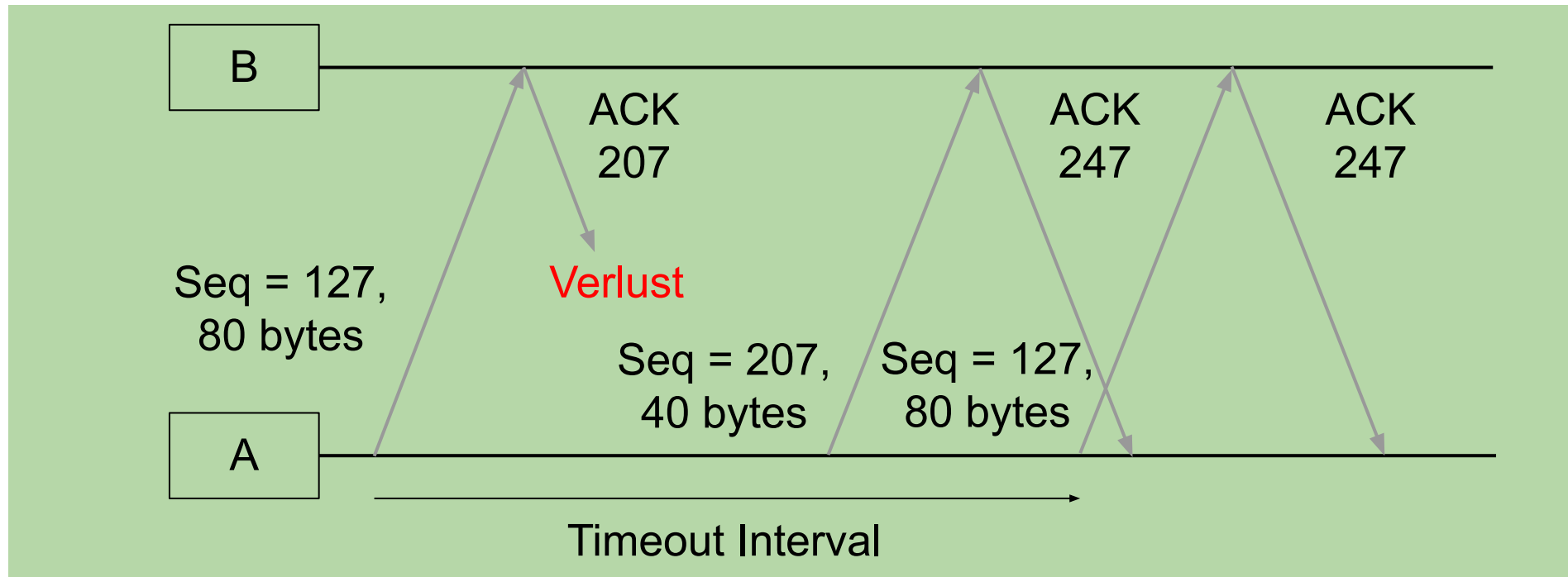
(c) Falls das erste Segment *nach* dem zweiten Segment bei B eintritt, wie lauten im ACK (Quittung) die ACK-Nr., Quell- und Zielpport?

- B wartet immer noch auf erstes Segment mit Seq = 127, 80 Bytes
→ ACK-Nr. ist 127
- Host B sendet an Host A
→ Quellport 80, Zielpport 302

zu c)



- (d) Angenommen, beide Segmente kommen in der richtigen Reihenfolge von A zu B. Das erste ACK von B geht verloren und das zweite ACK erreicht A nach dem ersten Timeout-Intervall. Zeichnen Sie ein Sequenzdiagramm und beschriften Sie jedes versendete Segment vollständig mit Sequenznummer, Anzahl der Nutzdaten-Bytes. Beschriften Sie des Weiteren alle Quittungen (ACKs) mit der korrekten ACK Nummer.



TCP-Verbindung

Ein Protokoll der Anwendungsschicht (z.B. HTTP) führt einen Anfrage-Antwort-Dialog aus, der über eine TCP-Verbindung zwischen einem Client- und einem Serverprozess transportiert werden soll. Die Netzverzögerung zwischen Client und Server betrage 150 ms, unabhängig von der Nachrichtenlänge. Ferner betragen die Größe der Anfrage (Request) 50 Byte und die Größe der Antwort (Response) 1000 Byte.

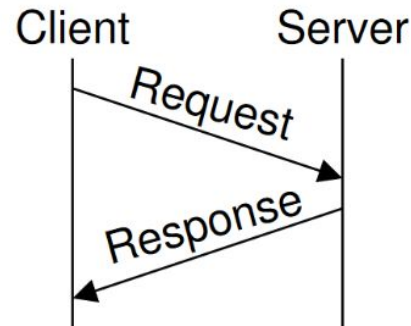


Abbildung 1: Ein Sequenzdiagramm der Anfrage und der Antwort des Servers.

- (a) Zeichnen Sie ein Sequenzdiagramm des gesamten TCP-Austausches zwischen Client und Server! Beschriften Sie dabei die Pfeile mit den dabei relevanten Teilen der TCP-Segmentstruktur (relevante Flags, Sequenznummer, ACK-Nummer). Initiale Sequenznummern seien 6000 für den Client und 9000 für den Server.

Client

Server

Flags

SeqNr

AckNr

Data

SYN

6000

SYN ACK

9000

6001

ACK

6001

9001

6001

Request 50 Byte

ACK

9001

6051

Response 1000 Byte

ACK FIN

6051

10001

ACK FIN

10001

6052

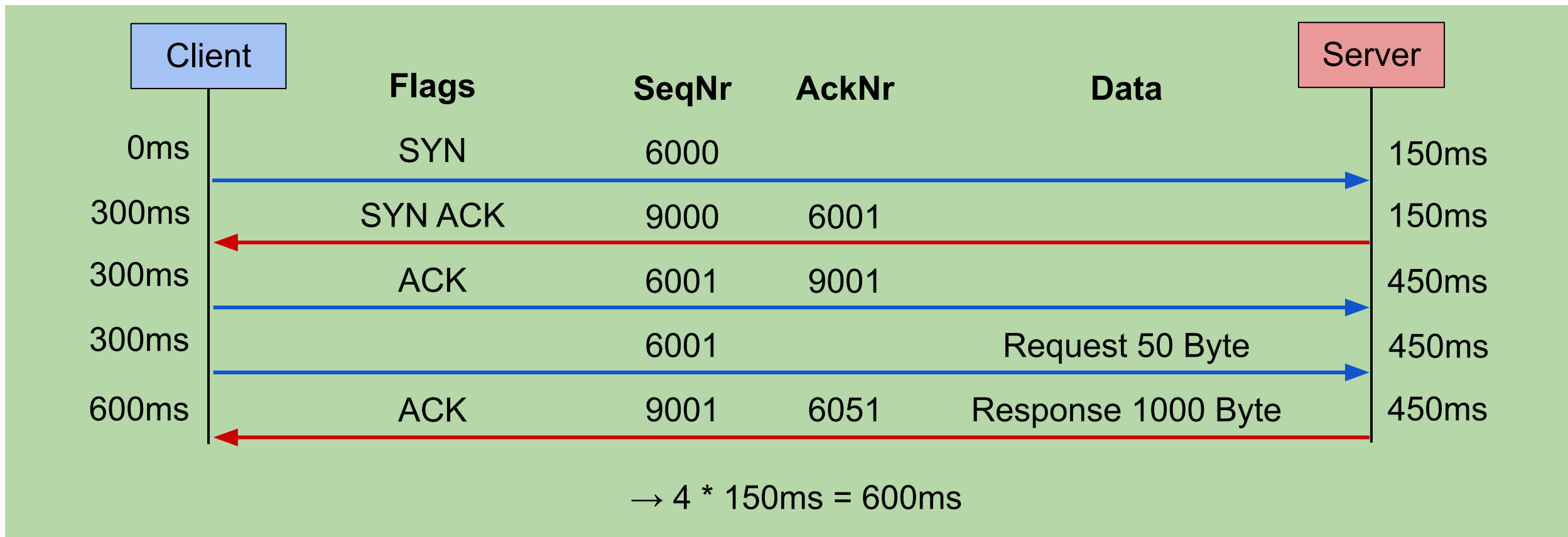
ACK

6052

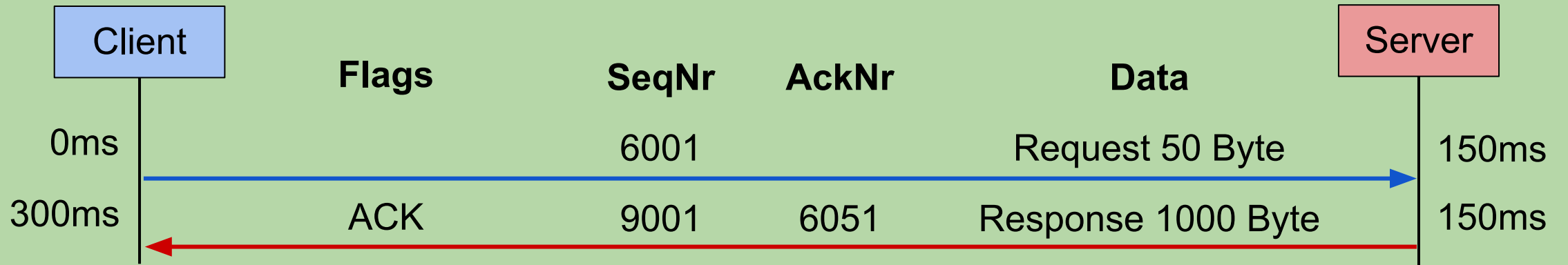
10002

(b) Zeitverhältnisse

i. Wie lange dauert es, bis die Antwort (Response) beim Client angekommen ist?

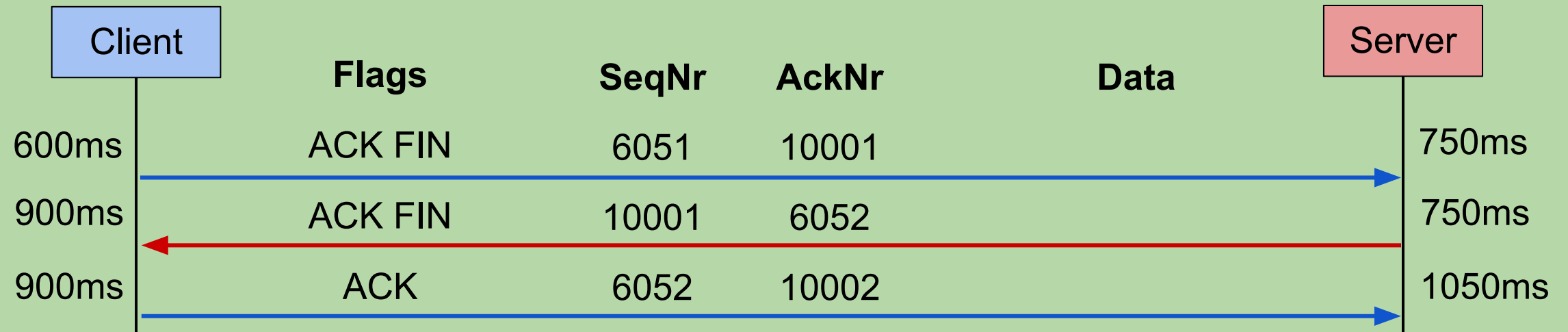


- ii. Um welchen Faktor schneller wäre der Austausch von Anfrage/Antwort mittels eines verbindungslosen Protokolls?



Ohne Verbindungsaufbau sind es 300ms
→ doppelt so schnell, Faktor 2

iii. Wie viel Zeit vergeht vom Versand des ersten bis zum Empfang des letzten Segments?



Aus dem Ergebnis aus b) i. kommt noch der Verbindungsabbau hinzu:
 $600\text{ms} + 3 * 150\text{ms} = 1050\text{ms}$