

Rechnernetze und verteilte Systeme

Übungsblatt 7

Koenig.Noah@campus.lmu.de



Senderate (ins Netz)

1. **Slow Start** (langsamer Start, exponentielles Wachstum)
2. **Congestion Avoidance** (lineares Wachstum zur Überlastvermeidung)

Optimierung bei Verlust

1. **Fast Retransmit**

Bei 3 Quittungsduplikaten wird eine Kopie der Nachricht vor Erreichen des Timeouts versendet (*Schnelle Neuübertragung*)

1. **Fast Recovery** (nur TCP Reno)

Bei 3 Quittungsduplikaten wird Threshold auf $\text{CongWin} / 2$ gesetzt und CongWin auf Threshold + 3, damit nicht wieder bei CongWin = 1 angefangen wird (*Schnelle Wiederherstellung*)

Kapitel 3 Transport, Folien 131-140

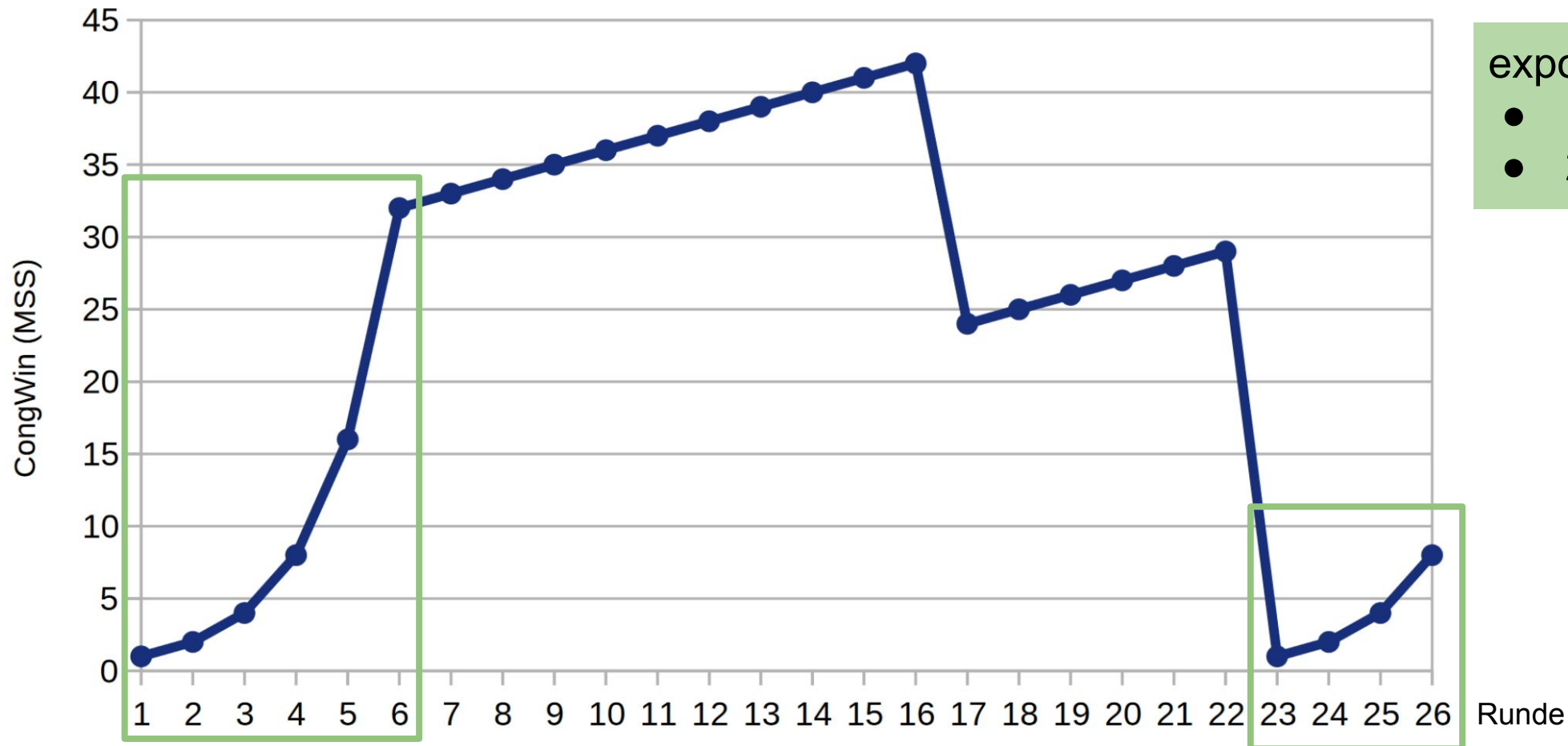
TCP Reno (H)

In Abbildung 1 ist das Verhalten von TCP-Reno nach RFC 2581 zu sehen. TCP-Reno verhält sich ähnlich wie TCP-Tahoe, jedoch mit Unterstützung für *Fast Recovery*:

Empfängt der Sender 3 ACK-Duplikate, geht er – anstelle von Slow Start – in den Fast Recovery Zustand über. Dabei wird zwar auch $\text{Threshold} = \frac{\text{CongWin}}{2}$ gesetzt, allerdings wird anschließend $\text{CongWin} = \text{Threshold} + 3\text{MSS}$ gesetzt. In der Fast Recovery Phase wird CongWin für jedes empfangene ACK-Duplikat um 1 MSS erhöht. Erreichen den Sender wieder *neue* Quittungen (also keine ACK-Duplikate), geht Reno in die lineare Phase (Congestion Avoidance) über.

In dieser Aufgabe nehmen wir an, dass während Fast Recovery keine weiteren Duplikate auftreten. Das heißt, dass das erneut übertragene Segment — die Ursache für die ACK-Duplikate — erfolgreich quittiert wird.

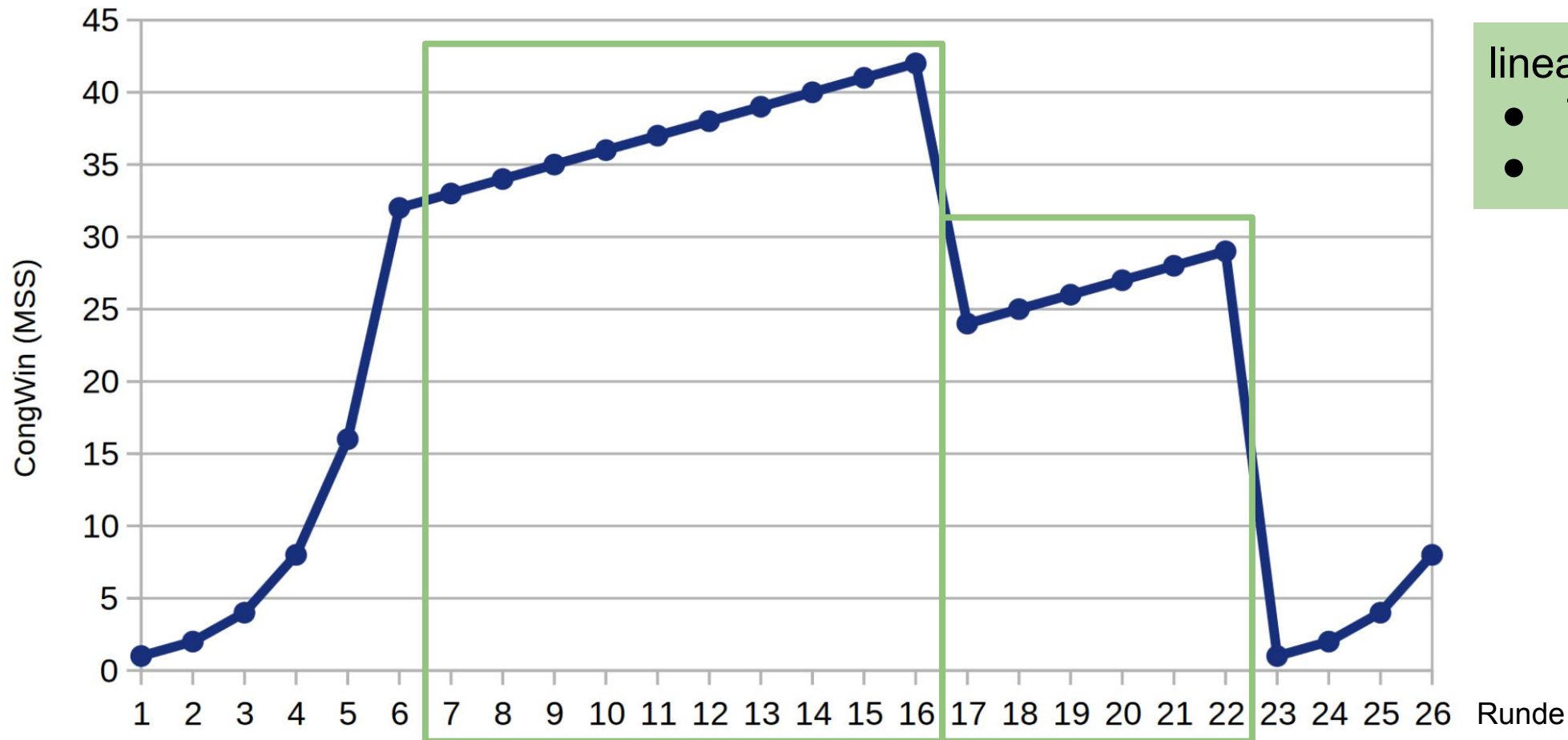
(a) Identifizieren Sie die Intervalle, in denen TCP Slow Start aktiv ist.



exponentielle Phasen:

- 1 bis 6
- 23 bis 26

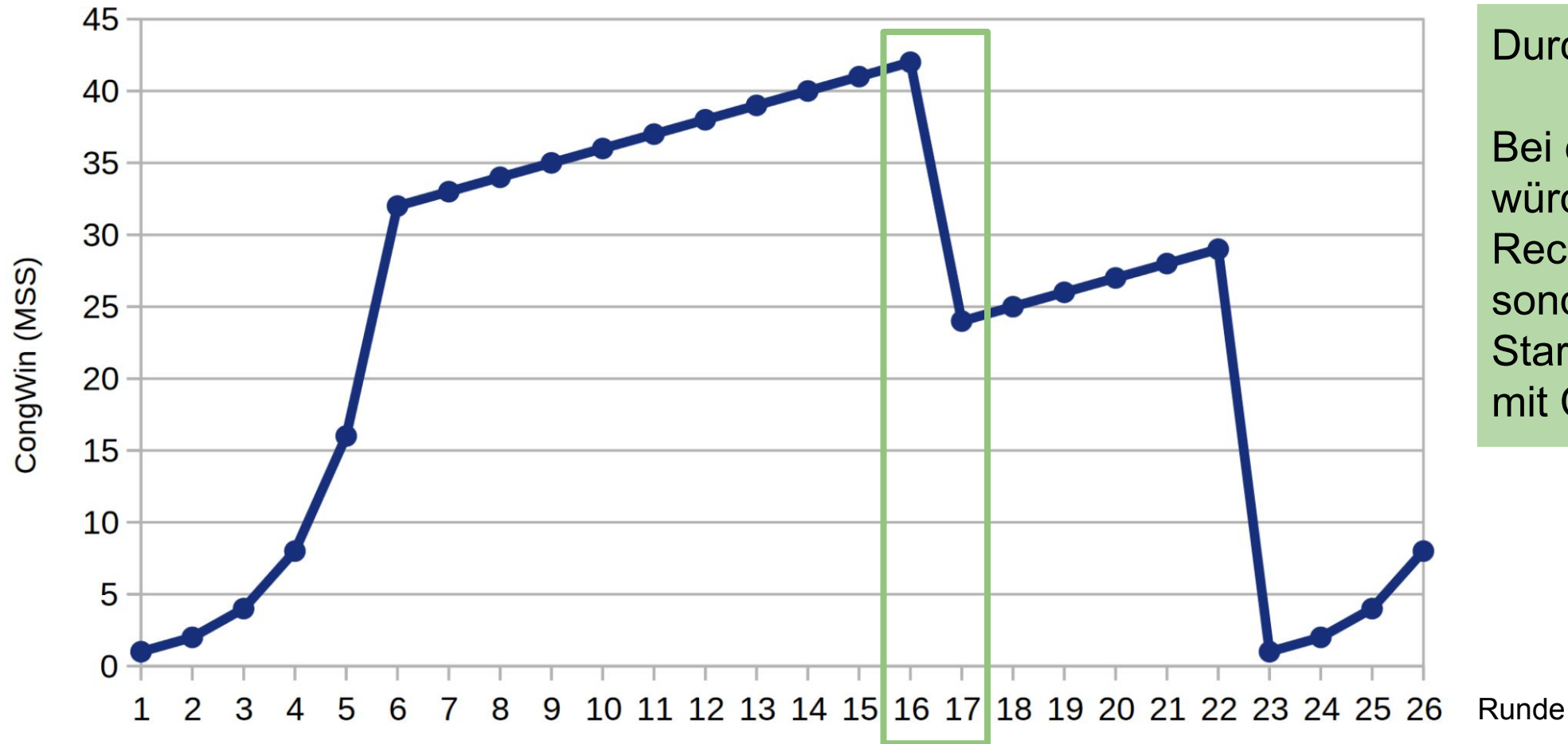
(b) Identifizieren Sie die Intervalle, in denen TCP Congestion Avoidance aktiv ist.



lineare Phasen:

- 7 bis 16
- 17 bis 22

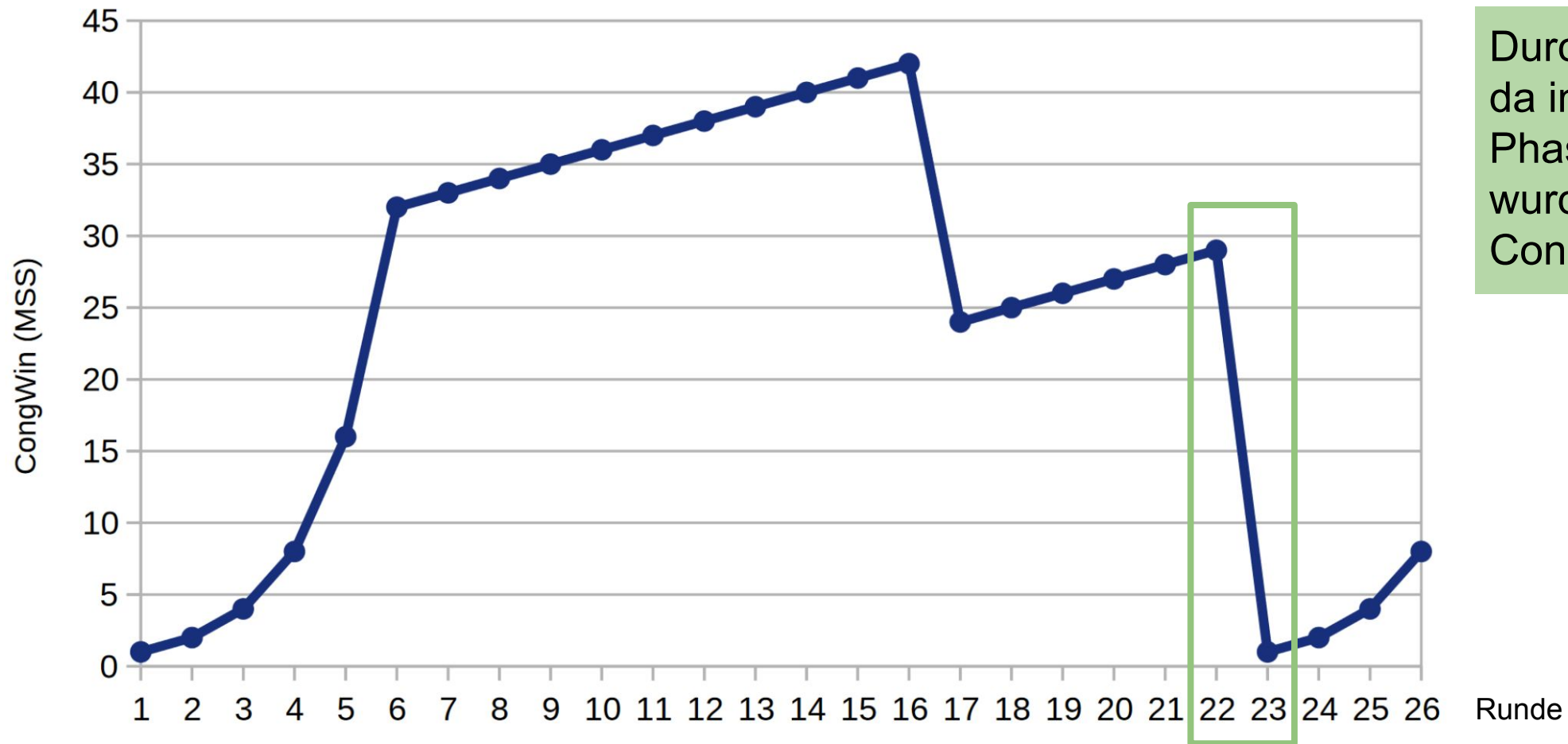
(c) Wurde der Paketverlust nach der 16. Runde durch duplizierte ACKs oder durch die Überschreitung des Timeouts erkannt? Warum?



Durch 3 ACK Duplikate

Bei einem Timeout würde man nicht Fast Recovery nutzen, sondern in die Slow Start Phase wechseln mit CongWin = 1

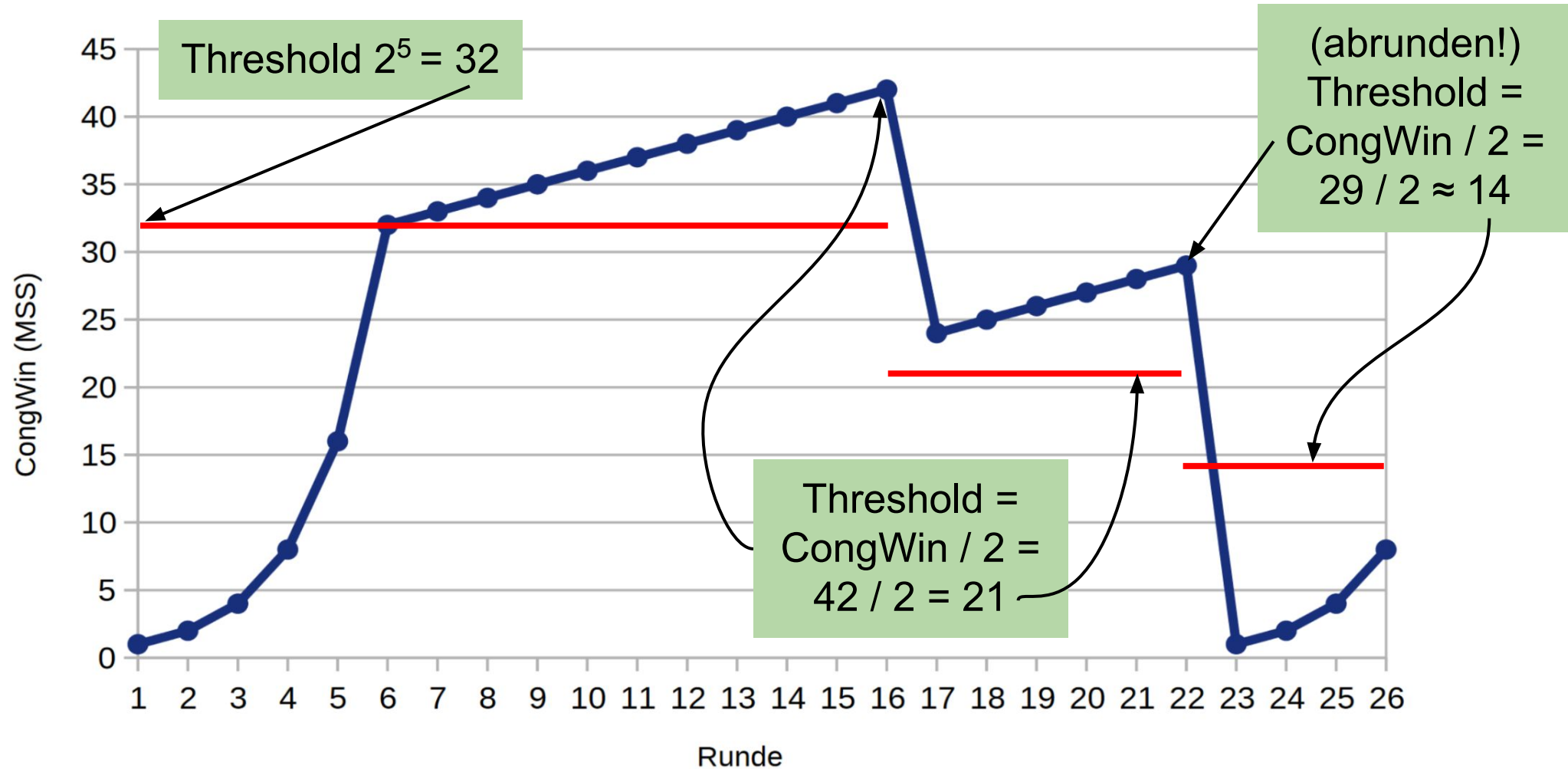
- (d) Wurde der Paketverlust nach der 22. Runde durch duplizierte ACKs oder durch die Überschreitung des Timeouts ausgelöst? Warum?



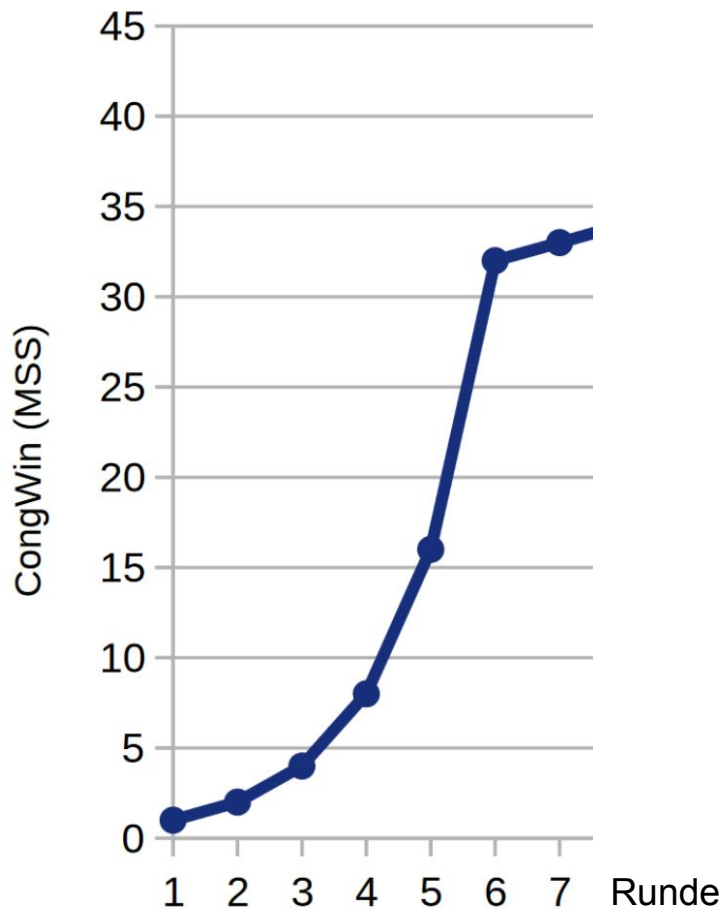
Durch einen Timeout, da in die Slow Start Phase gewechselt wurde mit $\text{CongWin} = 1$

Welchen Wert
hat **Threshold**
in Runde...

- (e) 1?
- (f) 18?
- (g) 24?



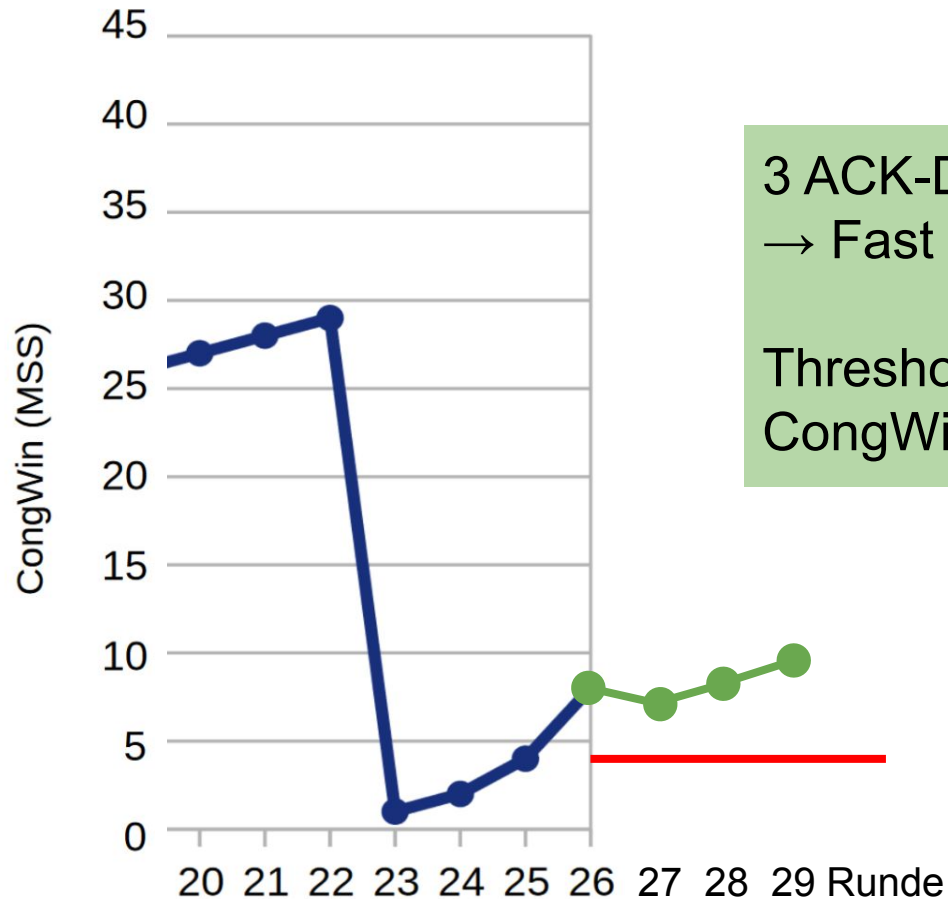
(h) In welcher Runde wird das 70. Segment gesendet?



Runde	CongWin	Segment(e)
1	1	1
2	2	2 bis 3
3	4	4 bis 7
4	8	8 bis 15
5	16	16 bis 31
6	32	32 bis 63
7	33	64 bis 96

→ Runde 7

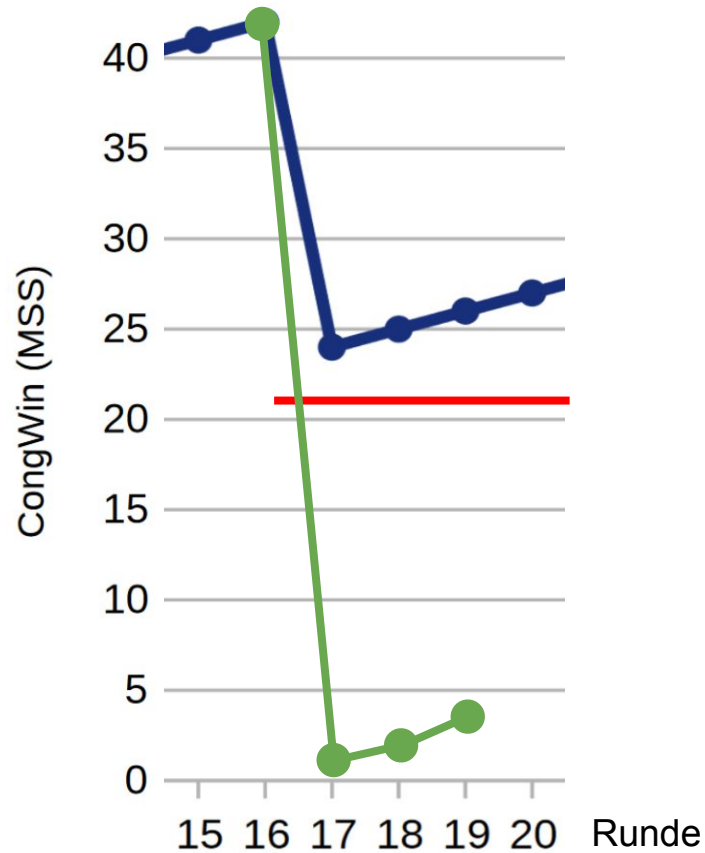
- (i) Angenommen in der 26. Runde wird ein Paketverlust durch ein (dreifaches) ACK-Duplikat festgestellt. Wie werden die Werte von CongWin und Threshold anschließend sein?



3 ACK-Duplikate
→ Fast Recovery

$\text{Threshold} = \text{CongWin} / 2 = 8 / 2 = 4$
 $\text{CongWin} = \text{Threshold} + 3 = 4 + 3 = 7$

- (j) Angenommen es würde TCP Tahoe statt Reno genutzt und in der 16. Runde wird durch ein (dreifaches) ACK-Duplikat ein Paketverlust festgestellt. Welche Werte haben **Threshold** und **CongWin** in der 19. Runde?



- $\text{Threshold} = \text{CongWin} / 2 = 21$
- Runde 17: $\text{CongWin} = 1$
- Runde 18: $\text{CongWin} = 2$
- Runde 19: $\text{CongWin} = 4$

Staukontrolle bei TCP (H)

Bei TCP kommt Slow-Start als Mechanismus zur Staukontrolle zum Einsatz. Es sollen über eine TCP-Verbindung mit Netzverzögerung von 100 ms (d.h. $RTD = 200$ ms) 7500 B Nutzdaten in 15 Segmenten gleicher Größe $S = 500$ B vom Server zum Client übertragen werden.

Hinweis: Nehmen Sie einen nicht erreichbaren Wert für **Threshold** an, und vernachlässigen Sie Verluste, Empfangspuffergröße, Verarbeitungsverzögerung, die Übertragungszeit für Protokollheader sowie den Verbindungsabbau. Gehen Sie also davon aus, dass die beiden TCP-Instanzen sofort nach dem Verbindungsaufbau die Übertragung beginnen und danach die Slow-Start-Phase nicht verlassen.

- (a) Sei die Übertragungsrate $R = 20$ kB/s. Erstellen Sie ein Sequenzdiagramm für die Übertragung und tragen Sie die Größe des jeweils aktuellen Überlastfensters (**CongWin**) in das Diagramm ein!

Sequenzdiagramm: [figma.com/file/R5UDzmzIXZk2sbedvxUs3F/RNVS-23-Staukontrolle-bei-TCP](https://www.figma.com/file/R5UDzmzIXZk2sbedvxUs3F/RNVS-23-Staukontrolle-bei-TCP)

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

- (b) Bestimmen Sie die Übertragungsdauer gemessen vom Absenden des SYN des Clients bis alle Nutzdaten empfangen wurden:
- mit Slow-Start.

- Nachrichtendauer: $500 \text{ Byte} / 20 \text{ kByte/s} = 0,5 \text{ kByte} / 20 \text{ kByte/s} = 0,025\text{s} = 25\text{ms}$
- Verbindungsaufbau: $3 * 100\text{ms} = 300\text{ms}$
- Quittungen des 4. Fensters werden hier nicht berücksichtigt
→ $3,5 \text{ RTTs} = 200\text{ms} * 3,5 = 700\text{ms}$
- Insgesamt $1 + 1 + 1 + 8 = 11$ Nachrichtendauern, da
 - bei den ersten 3 Fenstern nach dem ersten Segment geantwortet wird
 - beim 4. Fenster 8 Segmente abgewartet werden

→ $300\text{ms} + 700\text{ms} + 11 * 25\text{ms} = 1275\text{ms}$

ii. ohne Slow-Start mit fester Fenstergröße von 20.

- Verbindungsaufbau: $3 * 100\text{ms} = 300\text{ms}$
- Netzverzögerung des ersten Segments: 100ms
- Nachrichtendauer der 15 Segmente: $15 * 25\text{ms}$
→ $15 < 20$, d.h. auf Quittungen wird nicht gewartet, da Fenstergröße nicht voll

→ $300\text{ms} + 100\text{ms} + 15 * 25\text{ms} = 775\text{ms}$

(c) Wie lange würde die Übertragung jeweils mit und ohne Slow-Start für $R = 500 \text{ kB/s}$ dauern?

Lediglich die Nachrichtendauer wird von der neuen Übertragungsrate beeinflusst:

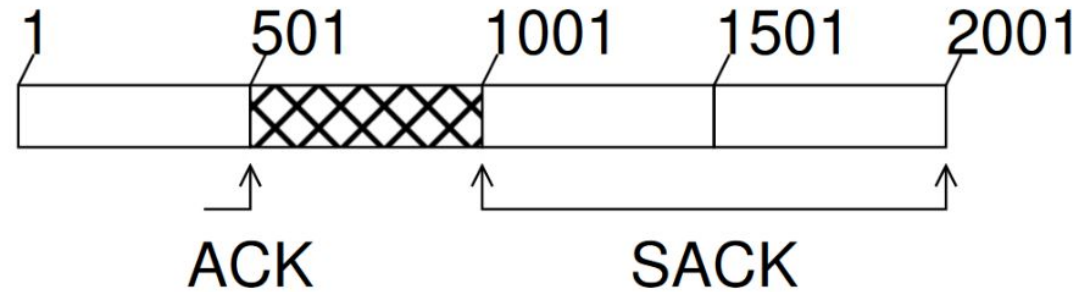
$$500 \text{ Byte} / 500 \text{ kByte/s} = 0,5 \text{ kByte} / 500 \text{ kByte/s} = 0,001 \text{ s} = 1 \text{ ms}$$

Lösungen aus (b) mit veränderter Nachrichtendauer:

- Mit Slow-Start: $300 \text{ ms} + 700 \text{ ms} + 11 * 1 \text{ ms} = 1011 \text{ ms}$
- Ohne Slow-Start: $300 \text{ ms} + 100 \text{ ms} + 15 * 1 \text{ ms} = 415 \text{ ms}$

Selektive Quittungen (H)

Der Verlust einzelner Segmente kann zur unnötigen Wiederholung großer Datenmengen führen, insbesondere bei Pfaden mit hoher Netzverzögerung. Durch die Einführung selektiver Quittungen (SACK, siehe RFC 2018) kann dieses Problem gemildert werden.



Statt wie bei den „normalen“ kumulativen TCP-Quittungen den bis dahin korrekt empfangenen *zusammenhängenden* Byte-Strom zu quittieren, kann ein Empfänger mit selektiven Quittungen zusätzliche Segmente (oder zusammenhängende Folgen von Segmenten, sogenannte Blöcke) im **Options**-Feld des TCP-Headers als empfangen notieren. Hierzu wird ein Bereich des Byte-Stroms mit Anfangs- und End-Byte notiert. Auf Grundlage der obigen Abbildung würde etwa eine Quittung mit AckNr=501 und SACK-Block=(1001,2001) gesendet; das verlorene Segment mit Bytes 501–1000 (schraffiert) wird so ausgespart. Es können mehrere solche Blöcke in den Optionen angegeben werden.

Gehen Sie von einem Sender mit aktueller SeqNr=5000 aus. Der Sender sendet 8 Segmente von jeweils 500 Byte Länge. Wie werden kumulative und selektive Quittungen benutzt, wenn:

(a) die ersten vier Segmente empfangen werden, die letzten vier aber verloren gehen?

Segment	SeqNr	ACK	SACK B1
1	5000	5500	-
2	5500	6000	-
3	6000	6500	-
4	6500	7000	-
5	7000	lost	-
6	7500	lost	-
7	8000	lost	-
8	8500	lost	-

(b) das zweite, vierte und sechste und achte Segment verloren gehen?

Segment	SeqNr	ACK	SACK B1	SACK B2	SACK B3
1	5000	5500	-	-	-
2	5500	lost	-	-	-
3	6000	5500	6000-6500	-	-
4	6500	lost	-	-	-
5	7000	5500	7000-7500	6000-6500	-
6	7500	lost	-	-	-
7	8000	5500	8000-8500	7000-7500	6000-6500
8	8500	lost	-	-	-

Approximierung von TCP Durchsatz und Verlustrate

Aus den Vorlesungsfolien geht hervor, dass der TCP Throughput während einer bestehenden Verbindung immer im Intervall zwischen $\frac{W}{2 \cdot RTD}$ bis $\frac{W}{RTD}$ pendelt. Während jeder Congestion-Avoidance-Phase geht 1 Paket verloren.

- (a) Zeigen Sie, dass die Verlustrate L (Anteil verloren gegangener Segmente) dem folgenden Term entspricht:

$$L = \frac{1}{\frac{3}{8}W^2 + \frac{3}{4}W}$$

Hilfreich: <https://www.thousandeyes.com/blog/a-very-simple-model-for-tcp-throughput>

- (b) Zeigen Sie, dass mit gegebener Verlustrate L während einer TCP-Verbindung die durchschnittliche Übertragungsrate¹ mit dem folgenden Term approximiert werden kann:

$$R \approx \frac{1.22 \cdot \text{MSS}}{\text{RTD} \cdot \sqrt{L}}$$