

Grundlagen der Informationstechnik

Übung 05 - TCP

Technische Universität Carolo-Wilhelmina zu Braunschweig
Institut für Datentechnik und Kommunikationsnetze (IDA)
Abteilung Kommunikationsnetze

- a) Client A initialisiert eine SSH Session zum Server S, welcher zeitgleich mit Client B kommuniziert. Nennen Sie eine mögliche Kombination von Quell- und Zielportnummern für
- Segmente, die von A zu S gesendet werden, und Segmente, die von B zu S gesendet werden
 - Segmente, die von S zu A gesendet werden, und Segmente, die von S zu B gesendet werden.
- b) Welche Bedingungen müssen erfüllt sein, damit Clients A und B die selben Quellportnummern verwenden dürfen?

1)

- a)
- Segmente, die von A zu S gesendet werden, und Segmente, die von B zu S gesendet werden
 - Segmente, die von S zu A gesendet werden, und Segmente, die von S zu B gesendet werden.

Portnummer von SSH 22, http 80

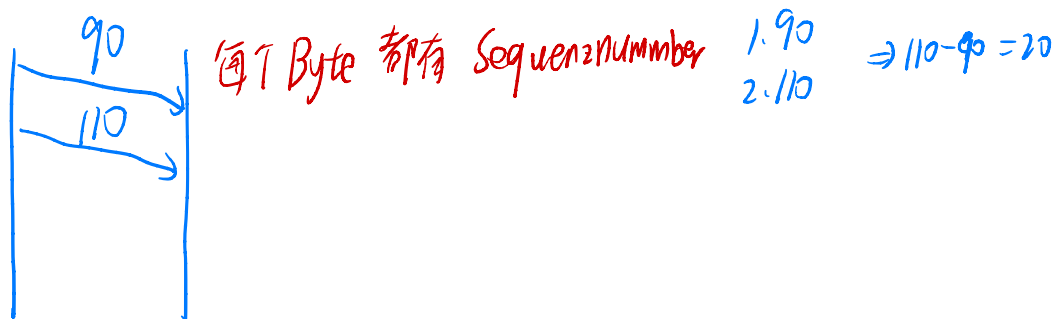
	Source Port Number	Destination Port Number
A → S	1467	22
B → S	1513	22
S → A	22	1467
S → B	22	1513

-
- b) Welche Bedingungen müssen erfüllt sein, damit Clients A und B die selben Quellportnummern verwenden dürfen?

Die Clients A und B dürfen dann die selben Quellportnummern verwenden, wenn sie sich auf verschiedenen Hosts befinden, da sie so durch die Kombination der Quell- und Ziel-IP Adressen unterschieden werden können. Ist dies nicht der Fall, ist die Unterscheidung der Clients nicht möglich.

Host A sendet zwei aufeinanderfolgende Segmente an Host B. Das erste Segment besitzt die Sequenznummer 90, das zweite die Sequenznummer 110.

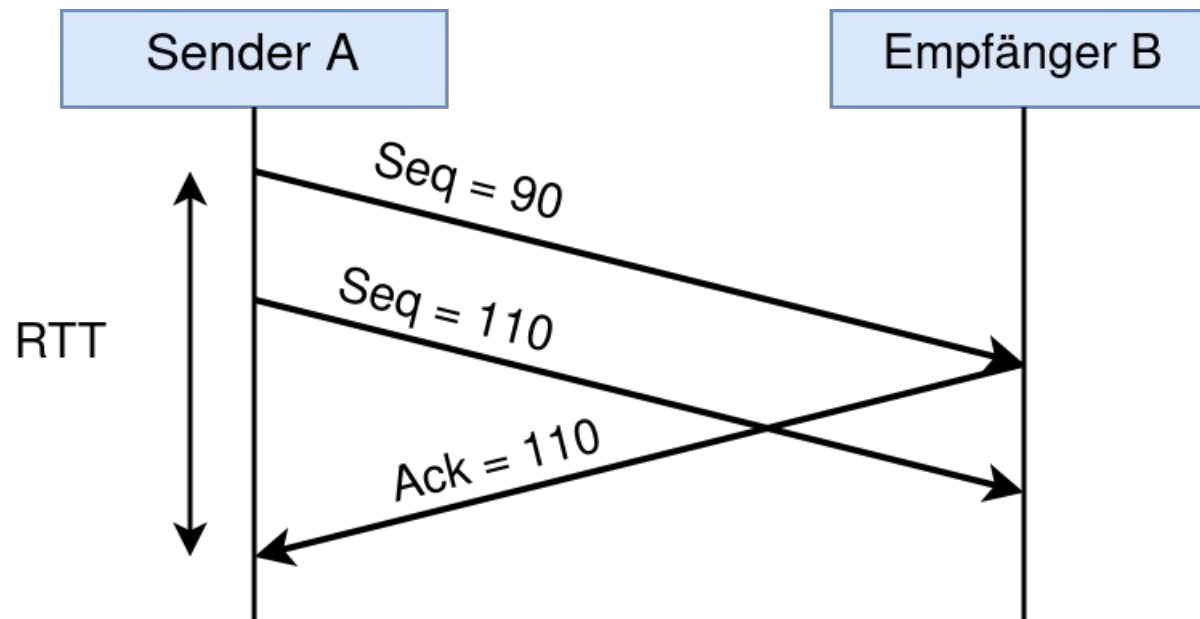
- a) Wie groß sind die Nutzdaten im ersten Segment?
- b) Das erste Segment geht zu Verlust, das zweite Segment wird fehlerfrei vom Host B empfangen. Alle vorangegangenen Segmente wurden bereits bestätigt. Welche Acknowledgement Nummer hat das ACK, das Host B an Host A sendet?



a) Wie groß sind die Nutzdaten im ersten Segment?

Die Sequenznummer nennt immer das erste Byte im Payload des Segmentes:

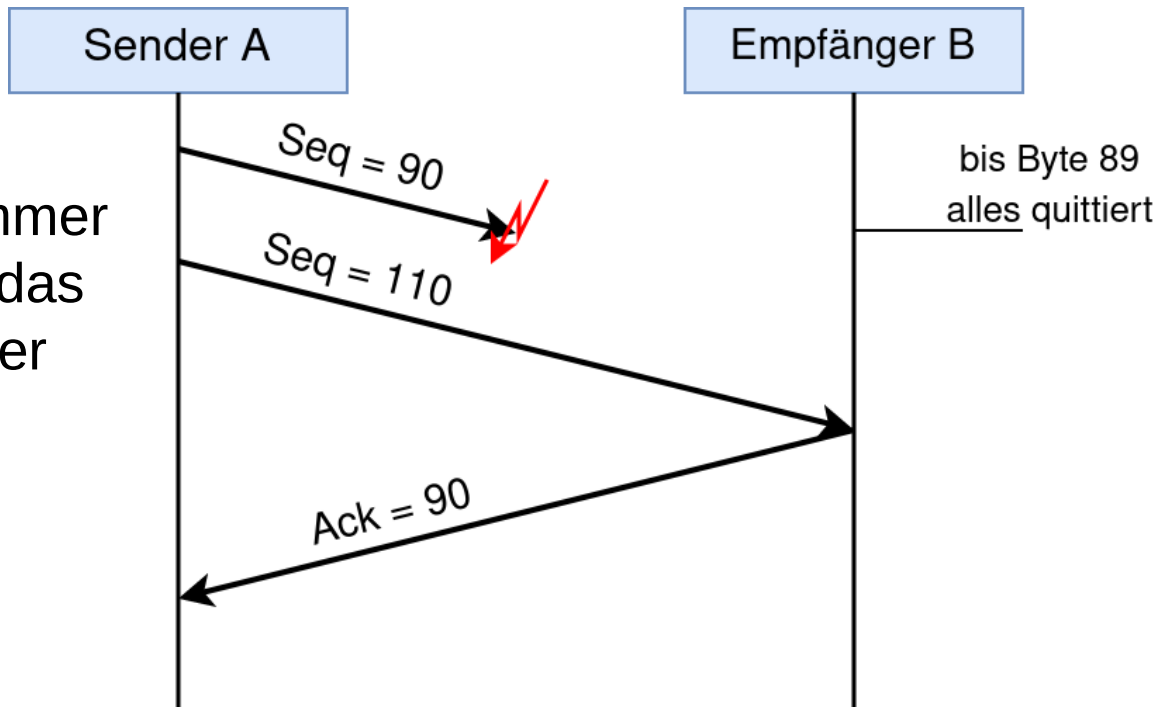
→ 20 Byte Nutzdaten im ersten Segment



- b) Welche Acknowledgement Nummer hat das ACK, das Host B an Host A sendet?

Die Acknowledgement Nummer ist die Nummer des Bytes, das als nächstes vom Empfänger erwartet wird

→ Acknowledgement Nummer = 90



- a) TCP überträgt über eine 40 Gbit/s optische Verbindung. Es wird angenommen, dass der Sender ununterbrochen senden kann. Wie lange dauert es, bis es zum Sequenznummern-überlauf (sequence number overflow) kommt?
- b) Der TCP Header wird um ein 32-bit Time Stamp erweitert, welcher jede ms erhöht wird. Wie lange dauert es, bis der Zeitstempel überläuft?
- c) Welche fundamentale Beziehung besteht zwischen dem maximalen Wert des Advertised Windows und der maximalen Sequenznummer (N_{\max}) eines TCP Frames, wenn das Receive Window (RW) im Empfänger gleich dem maximal erreichbaren Advertised Window ist?

- d) In einem vereinfachten Beispiel verwendet der Sender einen Retransmission Timer für jedes Segment, unabhängig von anderen Segmenten. Der Empfänger bestätigt alle empfangenen Segmente fehlerfrei und ohne Verzögerung (keine delayedAcks). Die Sequenznummern liegen in $\text{SeqNo}=\{0,1, 2, 3, 4, 5, 6, 7, 8\}$ ($N_{\max} = 9$). Das Sending Window hat den Wert $\text{SW} = 5$ und jedes Segment besitzt genau 1 Byte an Daten. Es soll nun angenommen werden, dass die ersten 5 Segmente fehlerfrei vom Empfänger empfangen werden. Ihre Sequenznummern befinden sich im Receive Window mit $\text{RW} = 5$. Die Segmente werden von der Applikation aus dem Fenster gelesen, sobald das letzte Segment empfangen wurde. Alle gesendeten ACKs gehen bei der Übertragung zu Verlust. Erläutern Sie mit Hilfe einer einfachen Skizze den Prozess der Übertragungswiederholungen der betroffenen Segmente nach einem Timeout und die Situation des Receive Windows im Empfänger.

1)

- a) TCP überträgt über eine 40 Gbit/s optische Verbindung. Es wird angenommen, dass der Sender ununterbrochen senden kann. Wie lange dauert es, bis es zum Sequenznummernüberlauf (sequence number overflow) kommt?

$$\frac{4.3 \text{ GByte}}{40 \text{ Gbit/s}} = \frac{4.3 \cdot 8}{40}$$

Sequence number overflow:

32 bit lange Sequenznummer

→ maximal 2^{32} Bytes unterscheidbar

$$2^{32} \cdot 8 \text{ Bit} / 40 \text{ Gbit/s} = 859 \text{ ms}$$

→ Es braucht 859 ms bis die Sequenznummern überlaufen.

1)

- b) Der TCP Header wird um ein 32-bit Time Stamp erweitert, welcher jede ms erhöht wird. Wie lange dauert es, bis der Zeitstempel überläuft?
- 32 bit stamp, 1ms ↑ -> -共可上升 $2^{32} \cdot 1ms$*

TCP – Options: 32 Bit time stamp in TCP-Header (RFC 1323)

- Fined grained timer

Um die Genauigkeit der gemessenen RTTs zu erhöhen erhalten alle gesendeten TCP Segmente eine Sendezeit. Der Receiver schickt den Zeitstempel anschließend in einem ACK zurück.

- Der Zeitstempel im TCP Header (Options Feld) erlaubt die Unterscheidung von Segmenten im Fall eines Sequenznummernüberlaufes.

→ Time stamp erhöht sich jede 1 ms

→ $2^{32} * 1ms = 4,29 * 10^6 s = 49,7 \text{ Tage}$ bis der Time Stamp überläuft

- c) Welche fundamentale Beziehung besteht zwischen dem maximalen Wert des Advertised Windows und der maximalen Sequenznummer (N_{\max}) eines TCP Frames, wenn das Receive Window (RW) im Empfänger gleich dem maximal erreichbaren Advertised Window ist?

N_{\max} 和 RW 之间的关系

Receive Window (RecBuffer):	RW
Sending Window (AdvWindow):	SW
Sequence Number (per Byte):	N_{\max}

Bedingung für die Anzahl an Sequenznummern um einen Sequenznummernüberlauf im Empfänger zu vermeiden:

$$N_{\max} > 2 \text{ RW} - 1$$

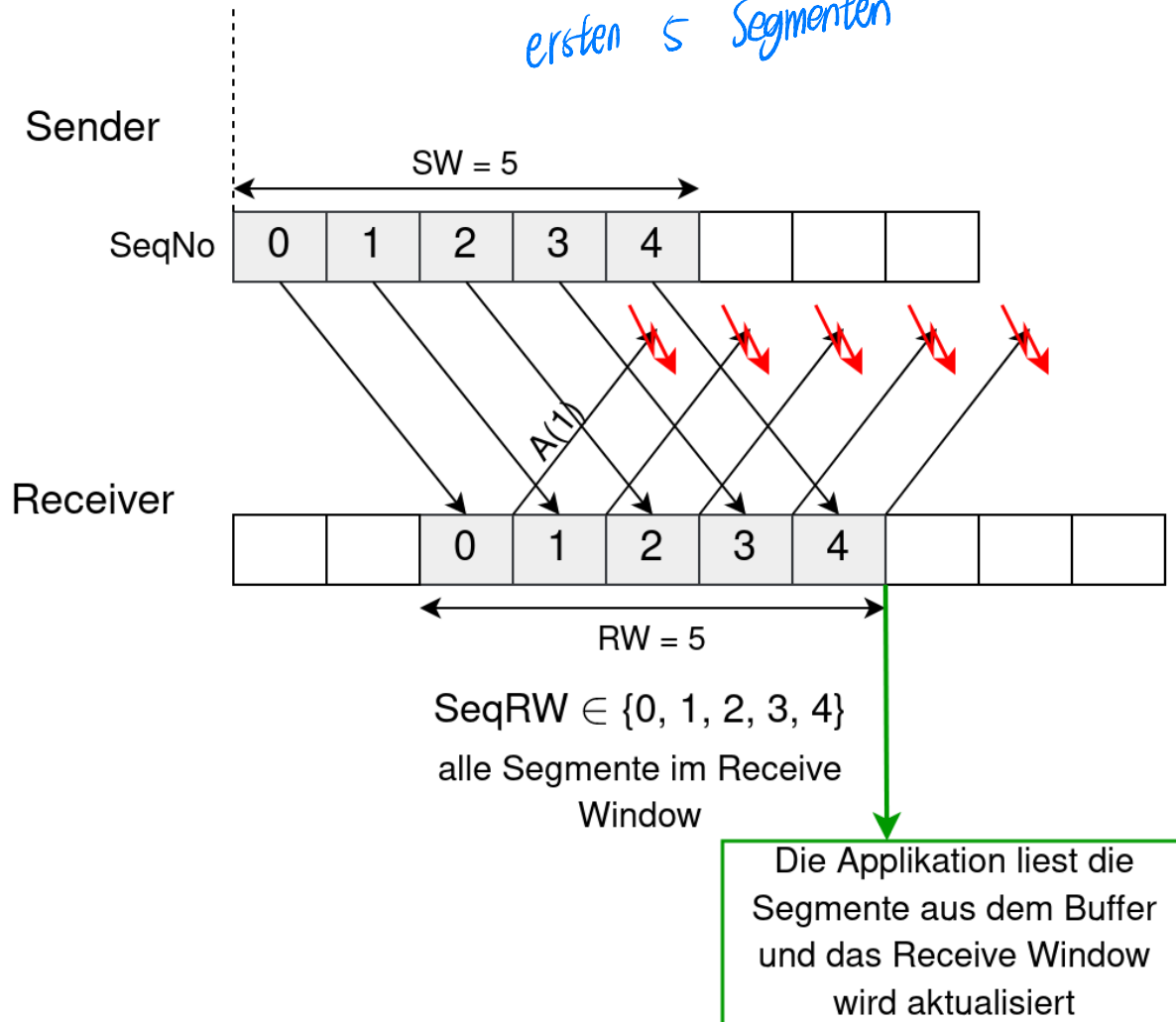
d) Erläutern Sie mit Hilfe einer einfachen Skizze den Prozess der Übertragungswiederholungen der betroffenen Segmente nach einem Timeout und die Situation des Receive Windows im Empfänger.

- Unabhängige Retransmission Timer für jedes Segment
- Alle fehlerfrei empfangenen Segmente werden sofort bestätigt (keine delayedACKs)
- SeqNo = {0, 1, 2, 3, 4, 5, 6, 7, 8} (Nmax = 9)
- SW = 5
- Jedes Segment enthält genau 1 Byte an Daten
- RW = 5
- Ersten 5 Segmente fehlerfrei empfangen und SeqNo im RW
- Segmente werden aus Receive Buffer gelesen, sobald letztes Segment empfangen
- Alle ACKs verloren

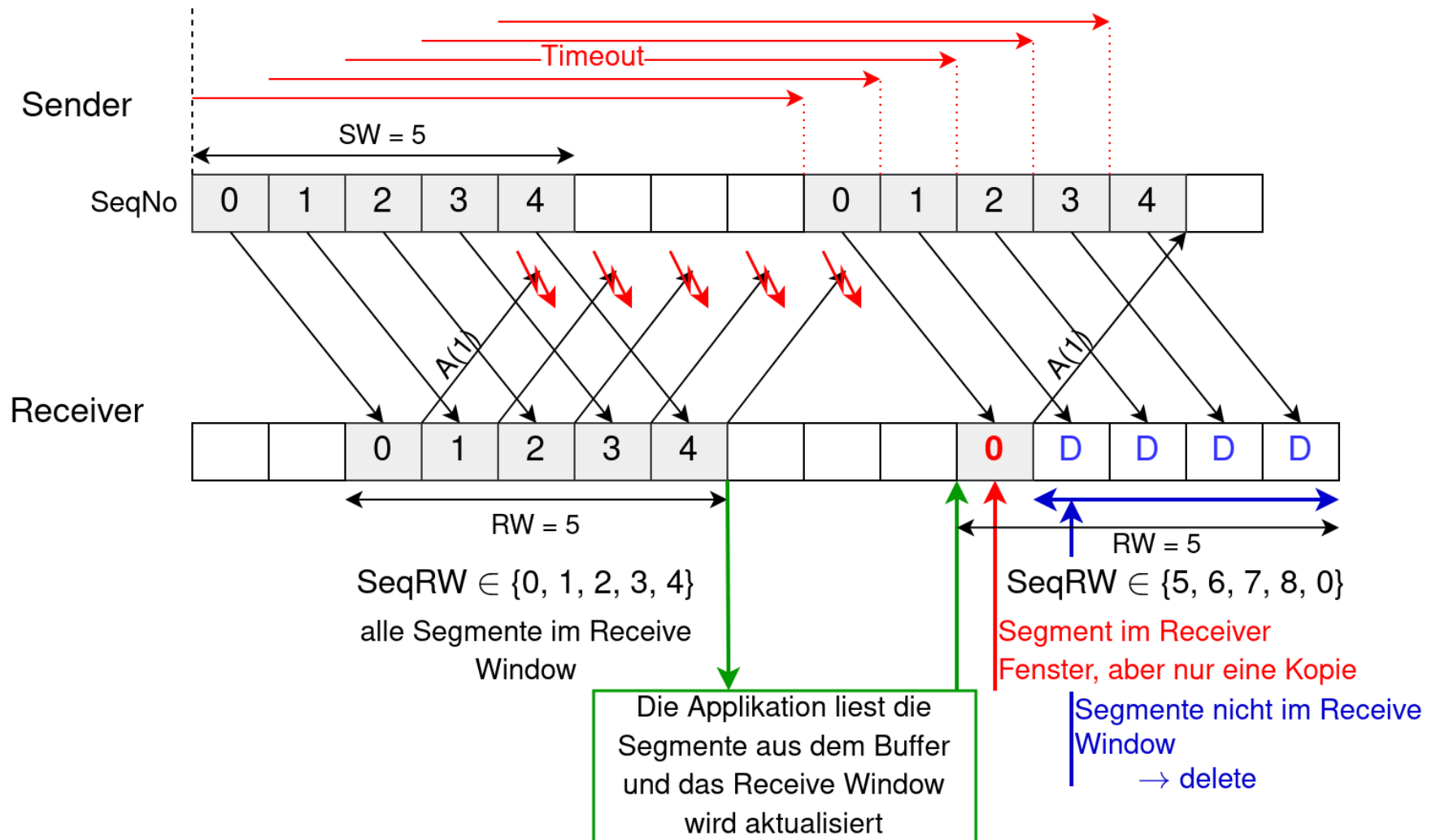
1)

$SW = RW = 5$; $N_{max} = 9$; $SeqNo = \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$

ersten 5 Segmenten



1)

 $SW = RW = 5; \quad N_{\max} = 9; \quad \text{SeqNo} = \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$


Eine TCP-Verbindung überträgt über einen stark gestörten Funkkanal und verwendet delayed ACKs (der Abstand aufeinanderfolgender Segmente ist kleiner als die Verzögerungsdauer des delayed ACKs). Ist mehr als ein Segment auf einmal zu bestätigen werden kumulative ACKs verwendet. Die Verlustwahrscheinlichkeit aufgrund von Bitfehlern beträgt 0.25; vereinfacht wird angenommen, dass jedes vierte Segment verfälscht wird. Der Retransmission Timer betrage 2 RTT. Für das Bandbreite-Verzögerungsprodukt gilt:

$RTT \gg \text{maximales Advertised Window} \times \text{Übertragungsdauer/Segment}$

Das Advertised Window soll hierbei einen festen Wert von 2MSS betragen.

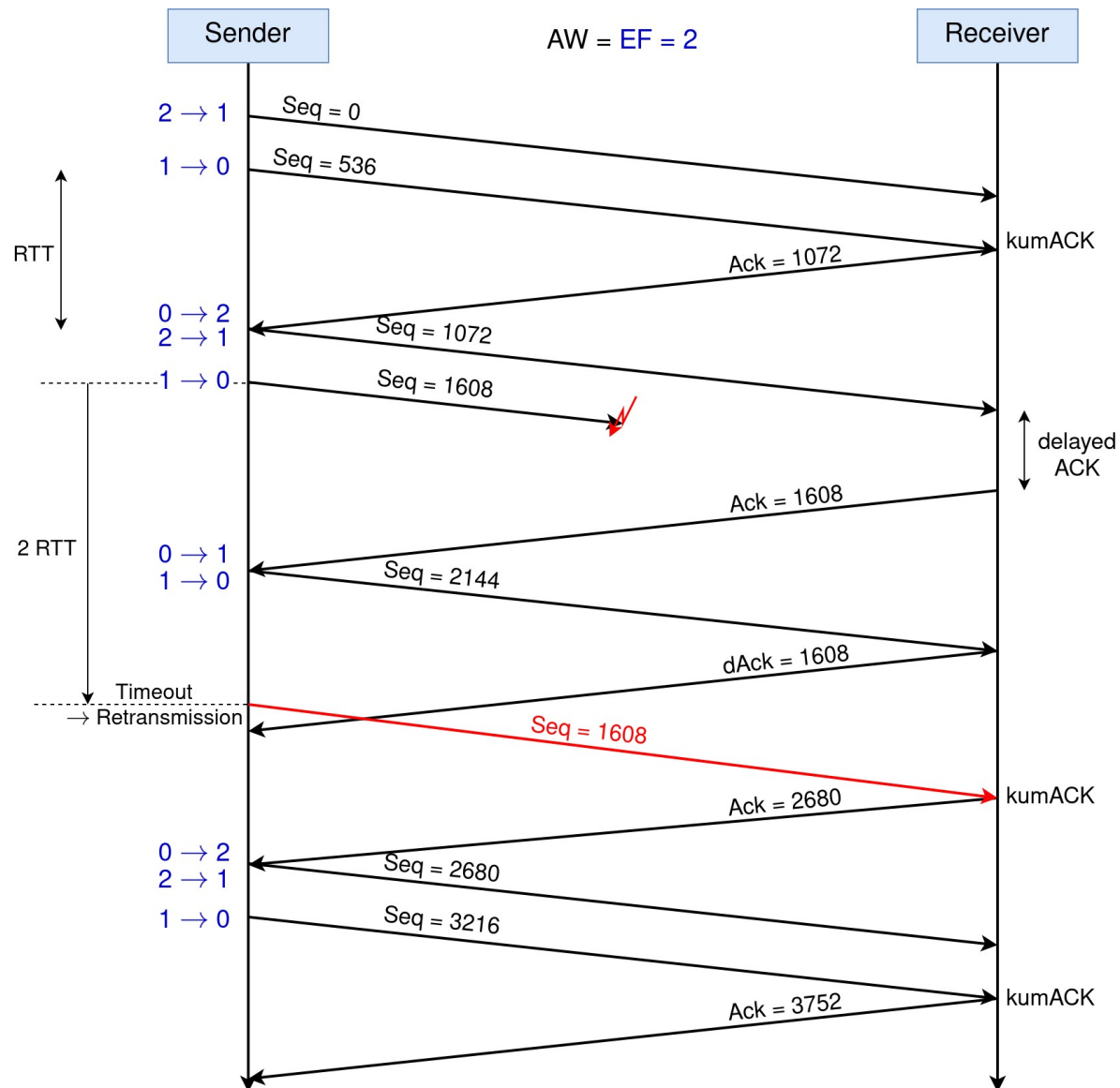
Ermitteln Sie den Verlauf des Effective Window für die ersten 7 Segmente (MSS = 536 Byte)

Ermitteln Sie den Verlauf des Effective Window für die ersten 7 Segmente (MSS = 536 Byte)

- 7 Pakete, MSS = 536 Byte
- delayed ACKs
- Paketverlustwahrscheinlichkeit $p = 0,25$
→ jedes vierte Paket geht verloren
- Retransmission Timer = 2 RTT
- AdvWin = 2 MSS

AdvWin = 2ms


$$CW=1 \rightarrow 2^0=1$$

Eine TCP- Verbindung überträgt über einen stark gestörten Funkkanal die ersten 8 Segmente ($MSS = 500$ Byte) vom Server (Sender) zum Client (Empfänger). Vereinfachend wird angenommen, dass Daten nur vom Sender zum Empfänger übertragen werden. Der Empfänger verwendet duplicate ACKs aber keine delayed ACKs. Ist mehr als ein Segment in richtiger Reihenfolge zu bestätigen, wird sofort ein kumulatives ACK gesendet. Die Round Trip Time betrage $RTT = 300$ ms und der Retransmission Timer wird zu $2 RTT$ gewählt, wobei angenommen wird, dass für jedes Segment ein eigener Retransmission Timer gesetzt wird. Es wird angenommen, dass lediglich das vierte gesendete Segment verfälscht wird.

Der Grenzwert für das Slow Start Verfahren liegt bei $Thres = 30$ MSS und das initiale Congestion Window ist $CW = 1$. Für das Advertisement Window (AW) wird ein konstanter Wert von $AW = 90$ MSS angenommen.

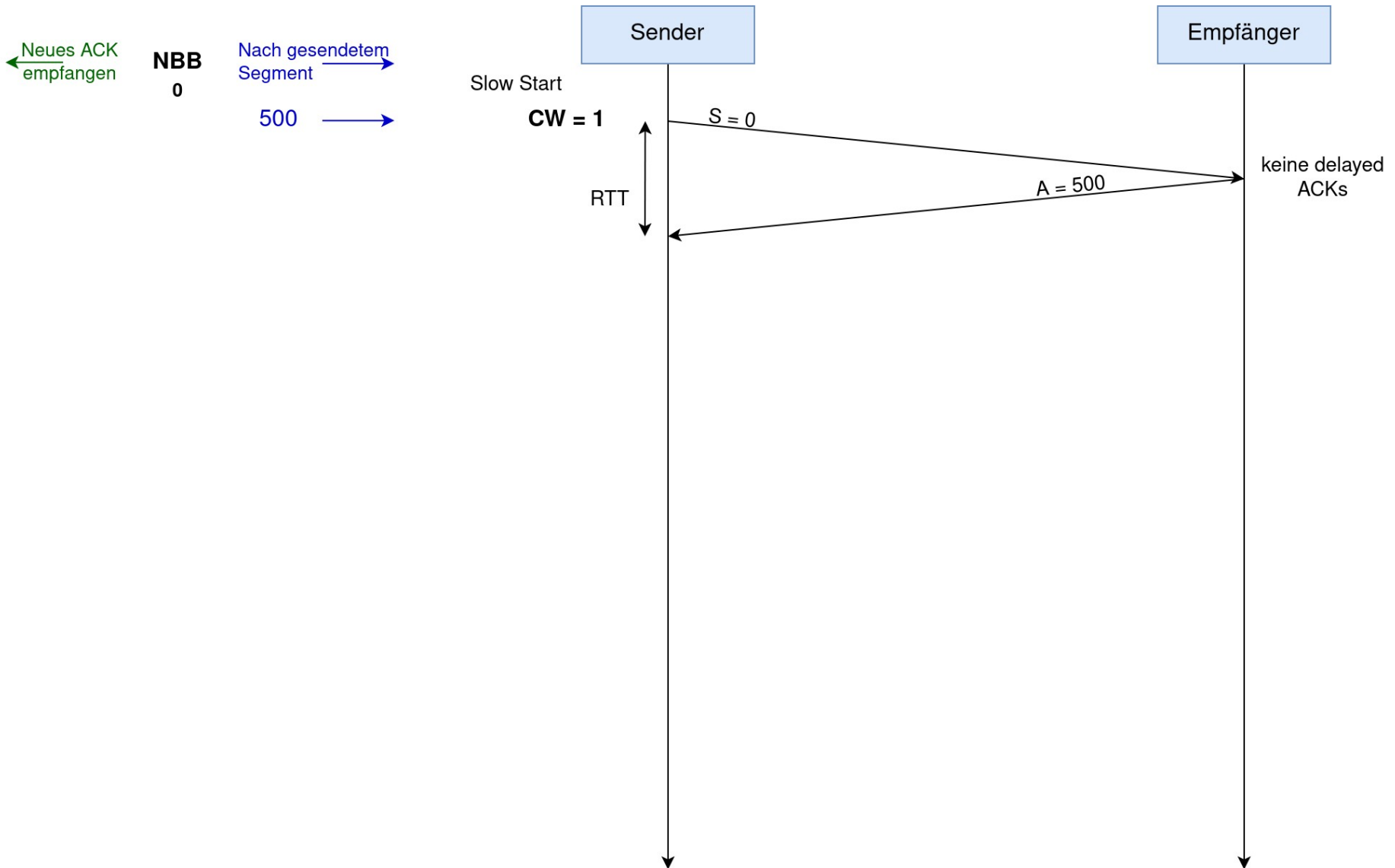
Ermitteln Sie die zeitliche Abfolge der Segmente und ACKs. Geben Sie dabei jeweils die Sequenznummer der gesendeten Datensegmente und die Bestätigungsnummer der ACKs an.

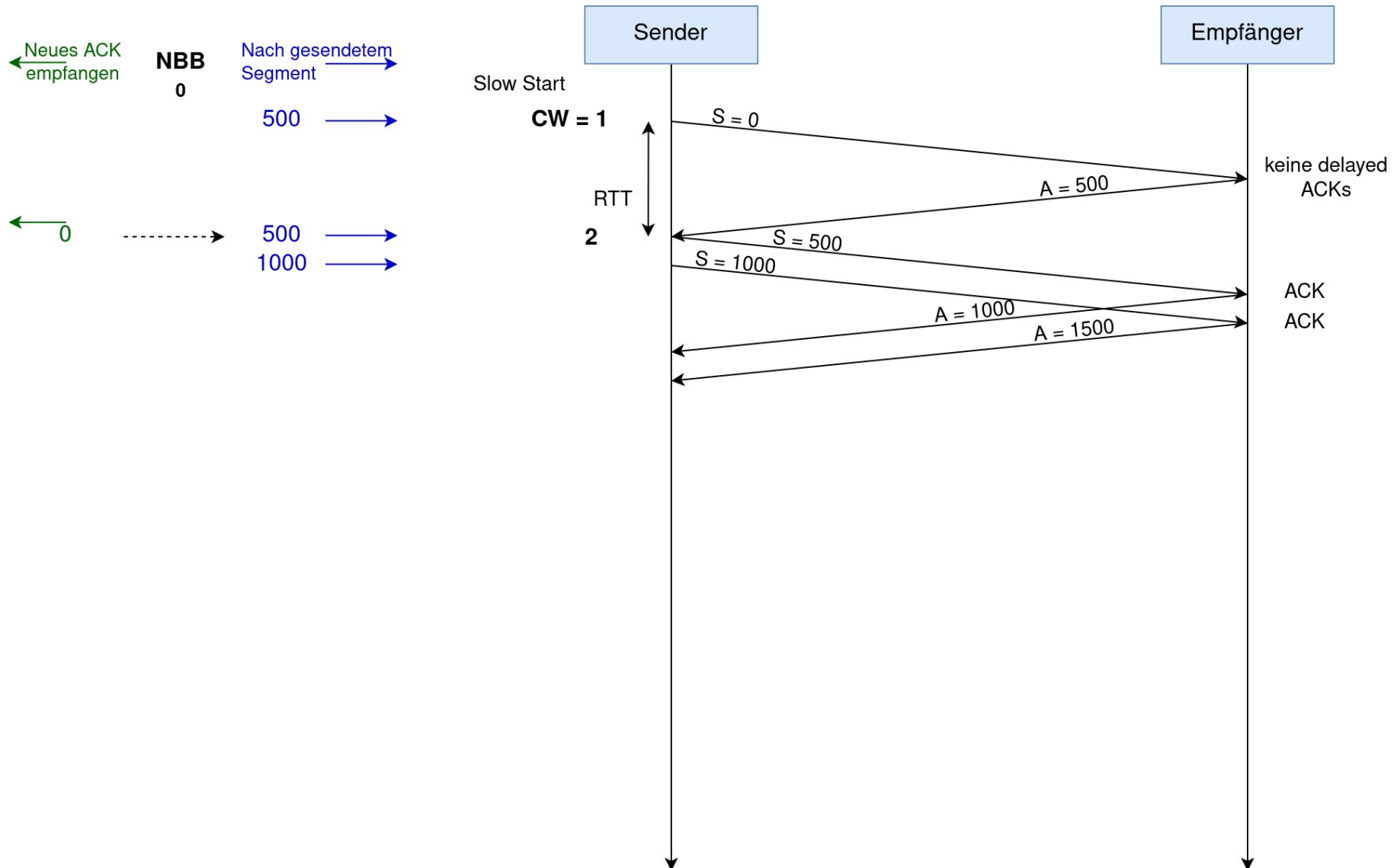
Ermitteln Sie den Verlauf des Congestion Windows am Sender.

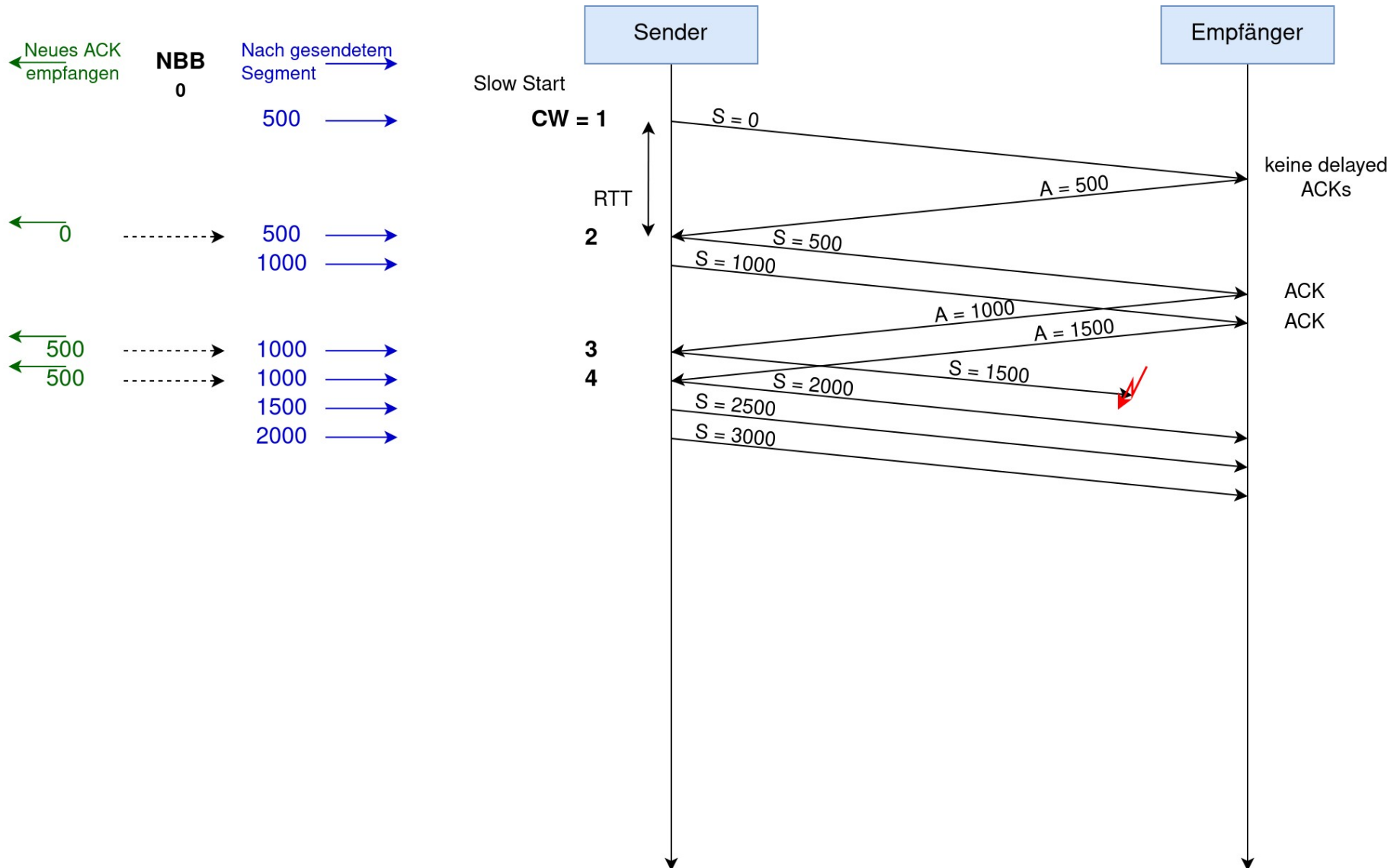
Kann die Slow Start Phase verlassen werden?

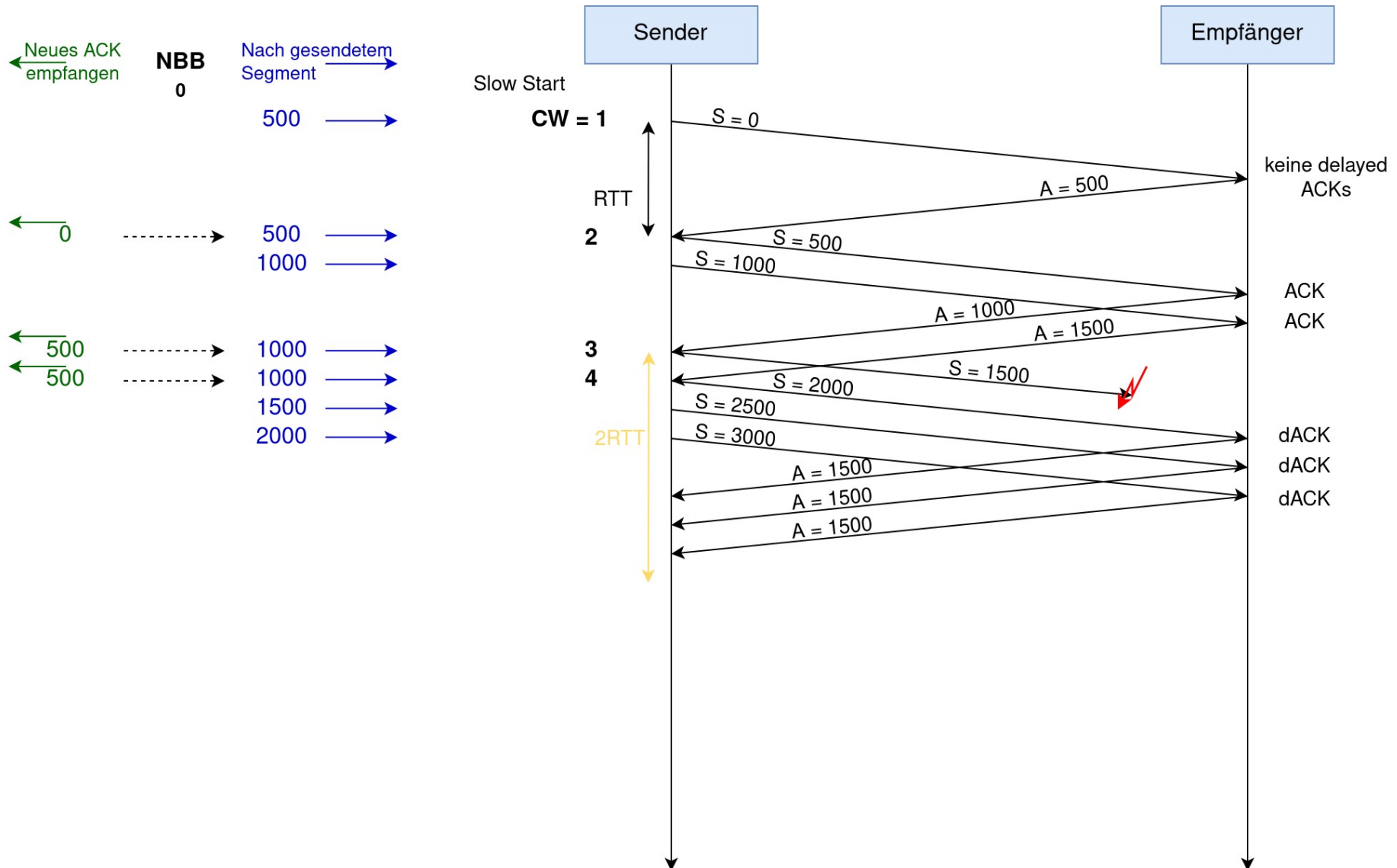
Bestimmen Sie am Sender dabei ebenfalls die Anzahl der gesendeten, aber noch nicht bestätigten Bytes (NBB) jeweils nach Empfang eines ACKs und nach den ggf. gesendeten Segmenten.

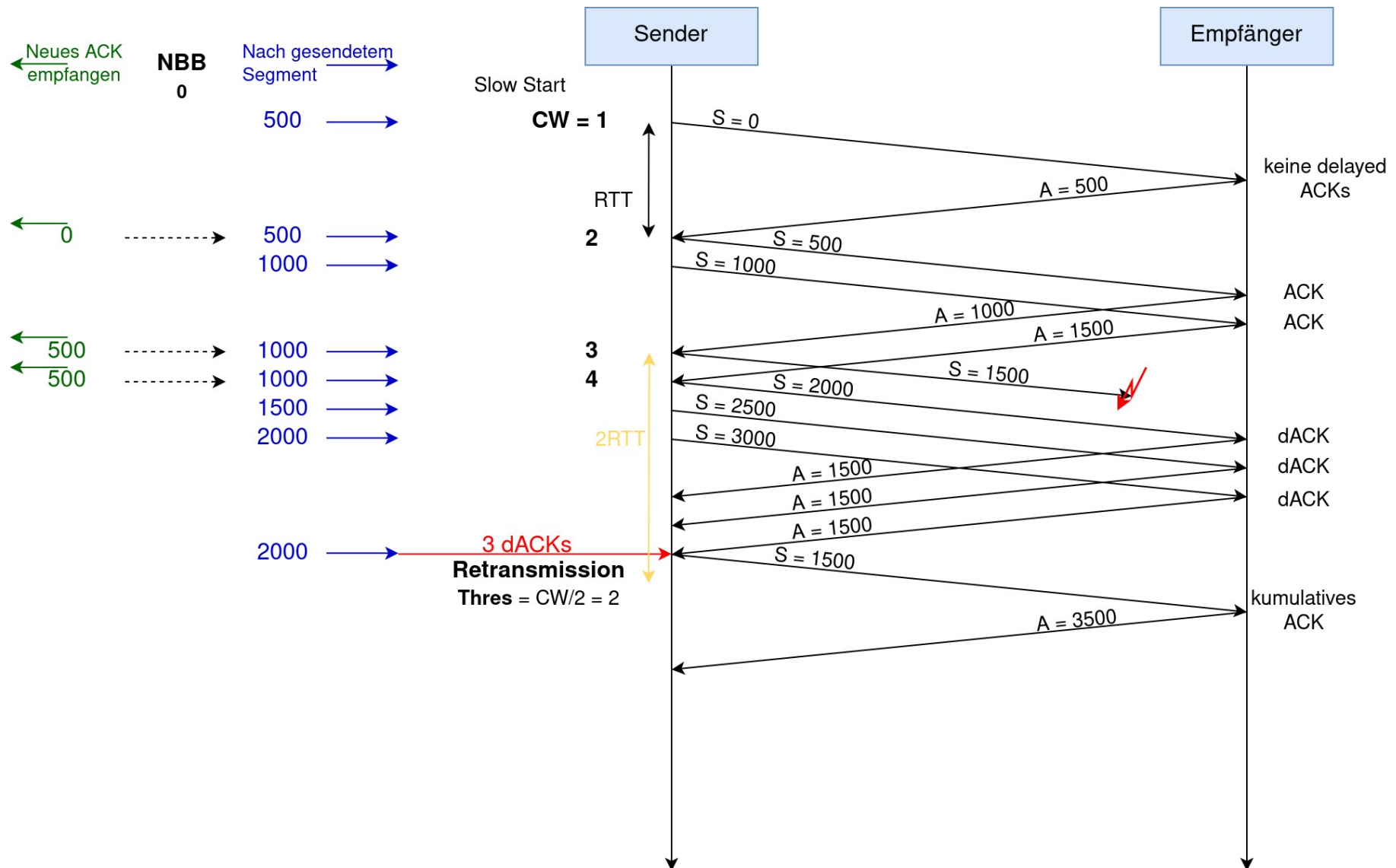
- 8 Segmente, MSS = 500 Byte
- Keine delayed ACKs
- Viertes Segment geht zu Verlust
- RTT = 300 ms
- Retransmission Timer = 2 RTT
- Slow Start
- Thres = 30 MSS
- CW = 1
- AW = 90 MSS

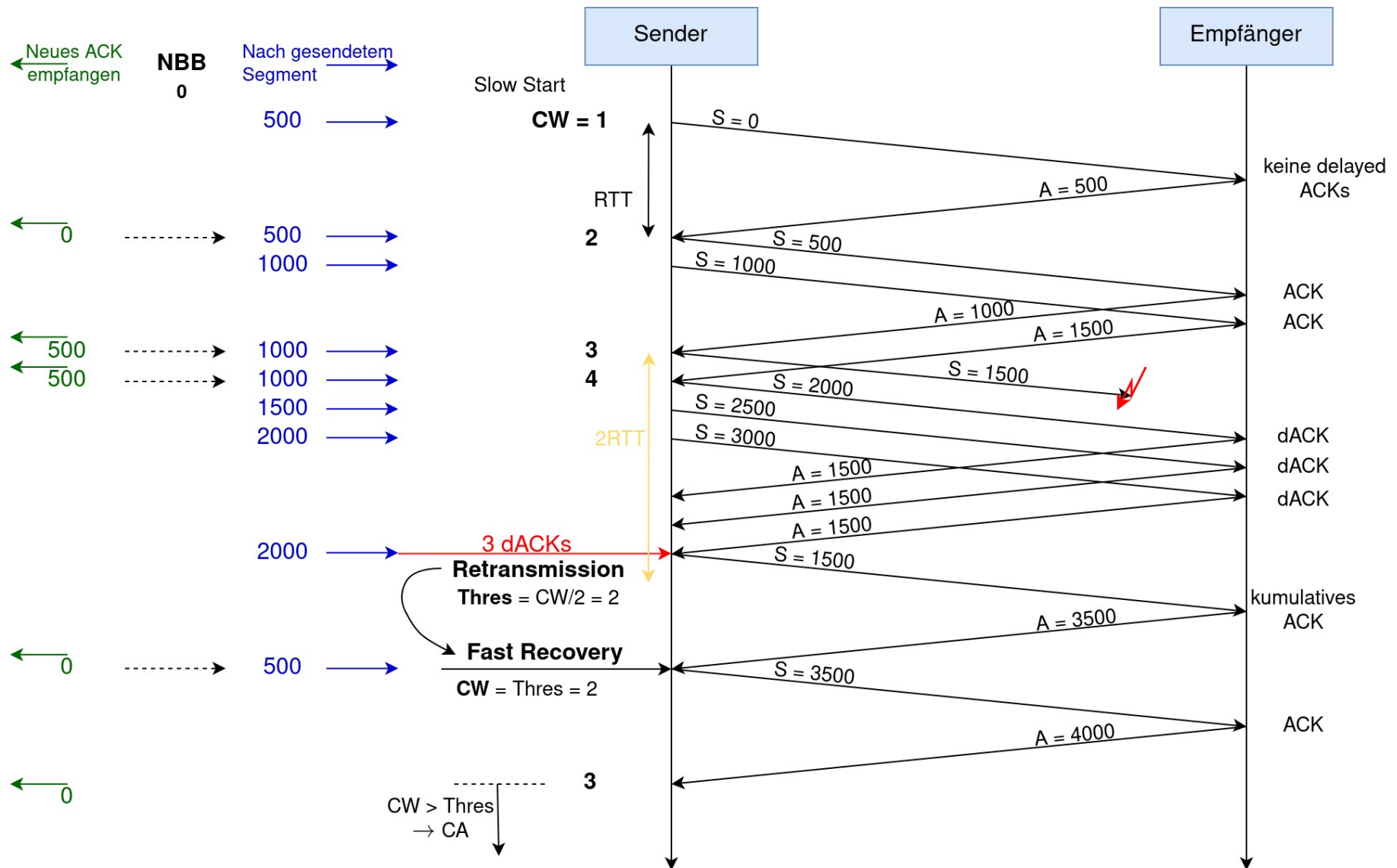












Während der Phase “Congestion Avoidance” erhöht TCP das Congestion Window linear nach jedem empfangenen ACK.

Es wird angenommen, dass das Sendefenster der Quelle nur durch das Congestion Window beschränkt wird und dass alle Segmente eines Fensters unmittelbar aufeinanderfolgend gesendet werden können. Die Übertragungsdauer eines Segmentes ist dabei klein im Vergleich zur Round Trip Time.

- a) Berechnen Sie den Verlauf des Congestion Window für die nächsten 4 gesendeten Segmente und einen anfänglichen Wert $\text{ConWin} = 4$, wobei jedes gesendete Segment genau ein $\text{MSS} = 536$ Byte an Daten beinhaltet. Nehmen Sie an, dass jedes Segment separat bestätigt wird. Zeigen Sie den Verlauf des Congestion Window in Abhängigkeit von der bis zum jeweiligen Zeitpunkt erfolgreich quitierten Datenmenge.
- b) Beantworten Sie die Frage nach a) für den Fall, das ein Segment nur $\text{MSS}/4$ Byte an Daten enthält.
- c) Nehmen Sie den Fall von delayed ACKs an und beantworten Sie die Frage gemäß a).

- a) Berechnen Sie den Verlauf des Congestion Window für die nächsten 4 gesendeten Segmente und einen anfänglichen Wert $\text{ConWin} = 4$, wobei jedes gesendete Segment genau ein $\text{MSS} = 536$ Byte an Daten beinhaltet. Nehmen Sie an, dass jedes Segment separat bestätigt wird. Zeigen Sie den Verlauf des Congestion Window in Abhängigkeit von der bis zum jeweiligen Zeitpunkt erfolgreich quitierten Datenmenge.

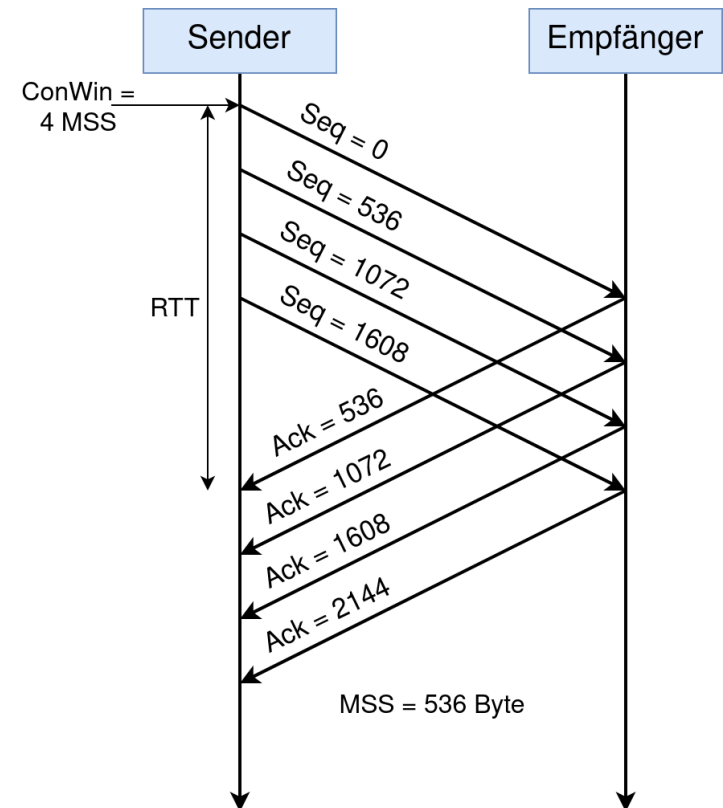
$\text{ConWin} = 4 \cdot \text{MSS}$ mit $\text{MSS} = 536$ Byte und jeweils MSS Bytes je Segment.

Anstieg des Congestion Windows in Byte:

$$\text{ConWin} = \text{ConWin} + \text{MSS} \cdot \frac{\text{MSS}}{\text{ConWin}}$$

$$\frac{\text{ConWin}}{\text{MSS}} = \frac{\text{ConWin}}{\text{MSS}} + \frac{1}{\text{ConWin}/\text{MSS}}$$

$$W = \frac{\text{ConWin}}{\text{MSS}} \rightarrow W = W + \frac{1}{W}$$



- a) Berechnen Sie den Verlauf des Congestion Window für die nächsten 4 gesendeten Segmente und einen anfänglichen Wert $\text{ConWin} = 4$, wobei jedes gesendete Segment genau ein $\text{MSS} = 536$ Byte an Daten beinhaltet. Nehmen Sie an, dass jedes Segment separat bestätigt wird. Zeigen Sie den Verlauf des Congestion Window in Abhängigkeit von der bis zum jeweiligen Zeitpunkt erfolgreich quitierten Datenmenge.

ConWindow		ACK	Gesamtmenge quittierter Daten		Anstieg ConWindow
in [MSS]	in [Byte]	#	a) in Byte	b) in Byte	in [MSS]
4	2144	1	536		$4 + 1/4$
4,25	2278	2	1072		$4,25 + 1/4,25$
4,4853	2404	3	1608		$4,4853 + 1/4,4853$
4,708	2524	4	2144		$4,708 + 1/4,708$
$4,92 \approx 5$					

- b) Beantworten Sie die Frage nach a) für den Fall, das ein Segment nur MSS/4 Byte an Daten enthält.

ConWindow		ACK	Gesamtmenge quittierter Daten		Anstieg ConWindow
in [MSS]	in [Byte]	#	a) in Byte	b) in Byte	in [MSS]
4	2144	1	536	134	$4 + 1/4$
4,25	2278	2	1072	268	$4,25 + 1/4,25$
4,4853	2404	3	1608	402	$4,4853 + 1/4,4853$
4,708	2524	4	2144	536	$4,708 + 1/4,708$
$4,92 \approx 5$					

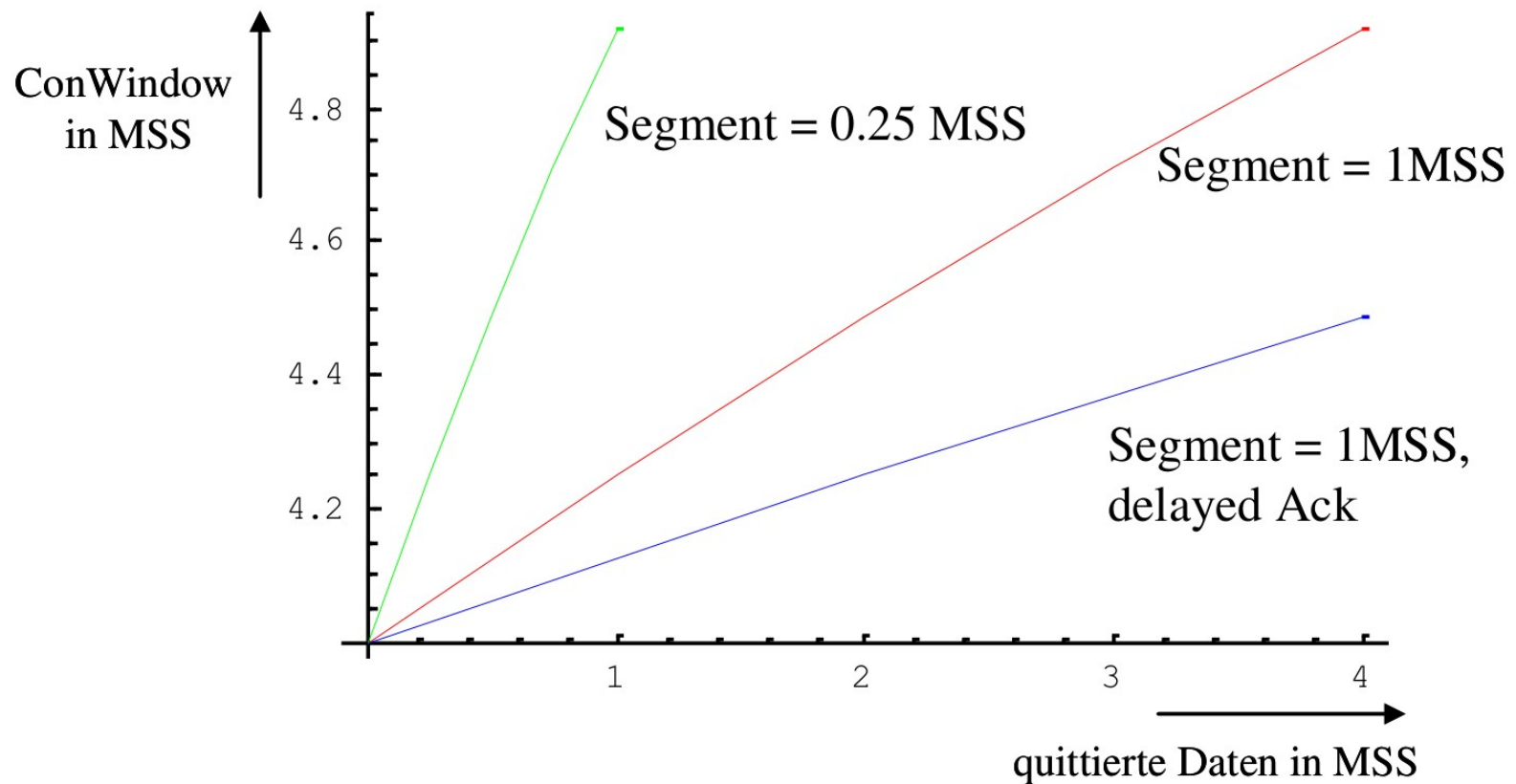
- c) Nehmen Sie den Fall von delayed ACKs an und beantworten Sie die Frage gemäß a).

Delayed ACKs

→ es werden jeweils 2 Segmente gemeinsam quittiert

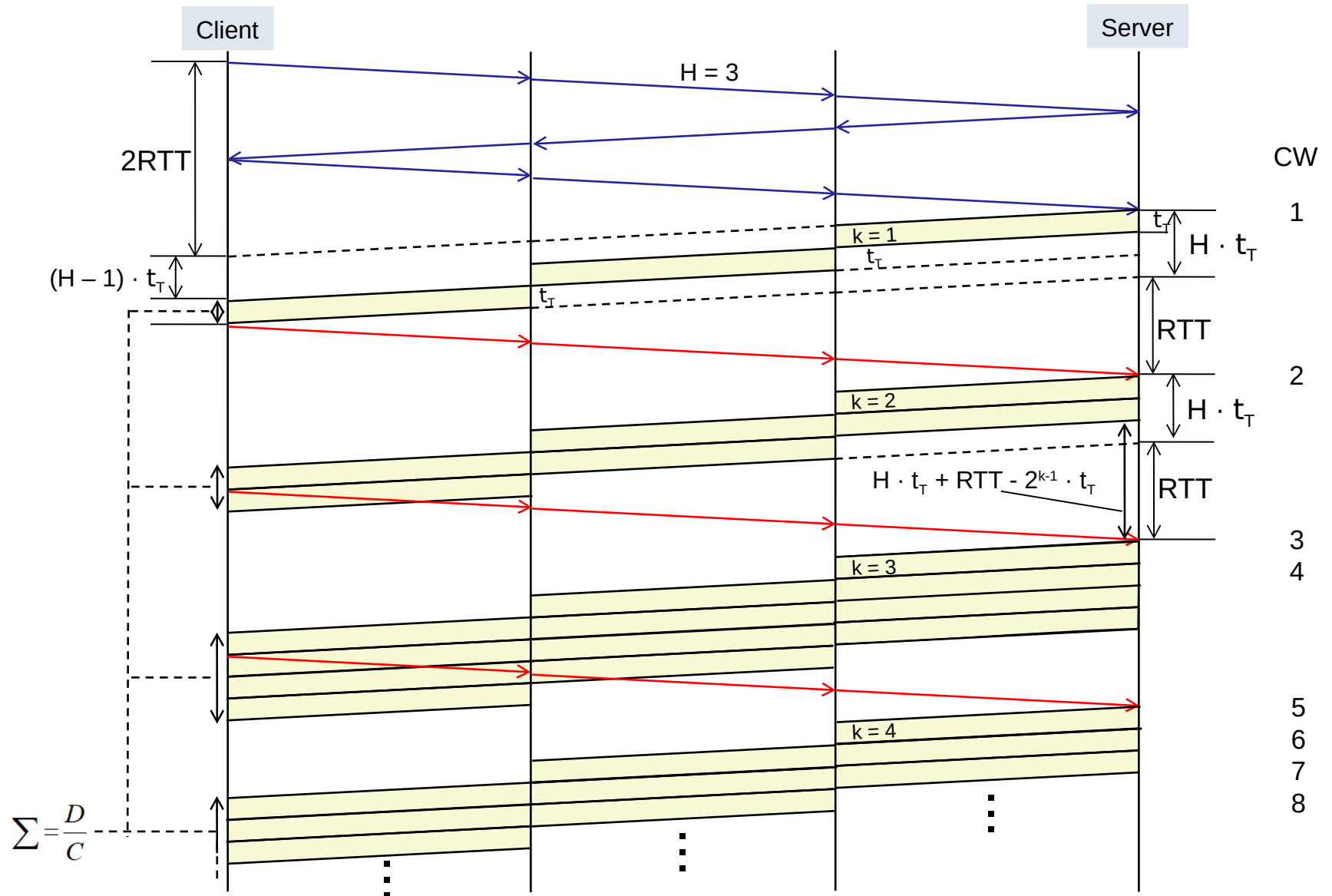
ConWindow		ACK	Gesamtmenge quittierter Daten	Anstieg ConWindow
in [MSS]	in [Byte]	#	in [Byte]	in [MSS]
4	2144	1	1072	$4 + 1/4$
4,25	2278	2	2144	$4,25 + 1/4,25$
4,4853	2404			

- c) Nehmen Sie den Fall von delayed ACKs an und beantworten Sie die Frage gemäß a).



Nehmen Sie an, dass es zwischen Client und Server insgesamt H Links (Übertragungsabschnitte) gibt. Die Warteschlangen in den Routern sind leer, so dass kein Queueing Delay auftritt. Jedoch müssen die Segmente gemäß dem Store-and-Forward Prinzip erst zwischengespeichert werden, bevor sie auf dem nächsten Abschnitt übertragen werden können.

Erweitern Sie die Analyse der in der Vorlesung durchgeführten Berechnung der TCP Latenz unter Berücksichtigung der Slow Start Phase.



Anzahl der Fenster Perioden K:

$$K = \min\{k: 2^0 + 2^1 + \dots + 2^{k-1} \geq S\}$$

$$= \min\{k: 2^k - 1 \geq S\} = \lceil \log_2(S + 1) \rceil$$

Anzahl der Idle Perioden Q:

$$Q = \max\{k: \text{RTT} + H \cdot t_T - t_T 2^{k-1} \geq 0\}$$

$$= \max\left\{k: 2^{k-1} \leq H + \frac{\text{RTT}}{t_T}\right\} = \max\left\{k: k \leq \log_2\left(H + \frac{\text{RTT}}{t_T}\right) + 1\right\}$$

$$= \left\lfloor \log_2\left(H + \frac{\text{RTT}}{t_T}\right) \right\rfloor + 1$$

Latenz, wenn der Server $F = \min(Q, K - 1)$ -mal idle ist:

$$\text{Latency} = 2\text{RTT} + \frac{D}{c} + (H - 1)t_T + \sum_{k=1}^F (\text{RTT} + H \cdot t_T - t_T 2^{k-1})$$

$$= 2\text{RTT} + \frac{D}{c} + (H - 1)t_T + \underbrace{F(\text{RTT} + H \cdot t_T) - (2^F - 1)t_T}_{\text{Sendelücken}}$$

Beispiel:

- Daten in Bit: $D = 56000 \text{ bit}$,
 $MSS = 4000 \text{ bit}$,
 Segmente: $S = D/MSS = 14$
- Durchsatz: $C = 10 \text{ Mbit/s}$
- Round Trip Time: $RTT = 1,2 \text{ ms}$
- Übertragungsdauer eines Segmentes:

$$t_T = \frac{MSS}{C} = 400\mu s, \rightarrow \frac{RTT}{t_T} = 3$$

- $K = \lceil \log_2(15) \rceil = 4$
- $Q = \lceil \log_2(3 + 3) \rceil + 1 = 3$
- $F = \min(Q, K - 1) = 3$
- Latenz = $2 \cdot 1,2\text{ms} + \frac{56000\text{bit}}{10\text{Mbit/s}} + 2 \cdot 0,4\text{ms}$
 $+ 3(1,2 + 3 \cdot 0,4)\text{ms} - 0,4(2^3 - 1)\text{ms} = 13,2\text{ms}$