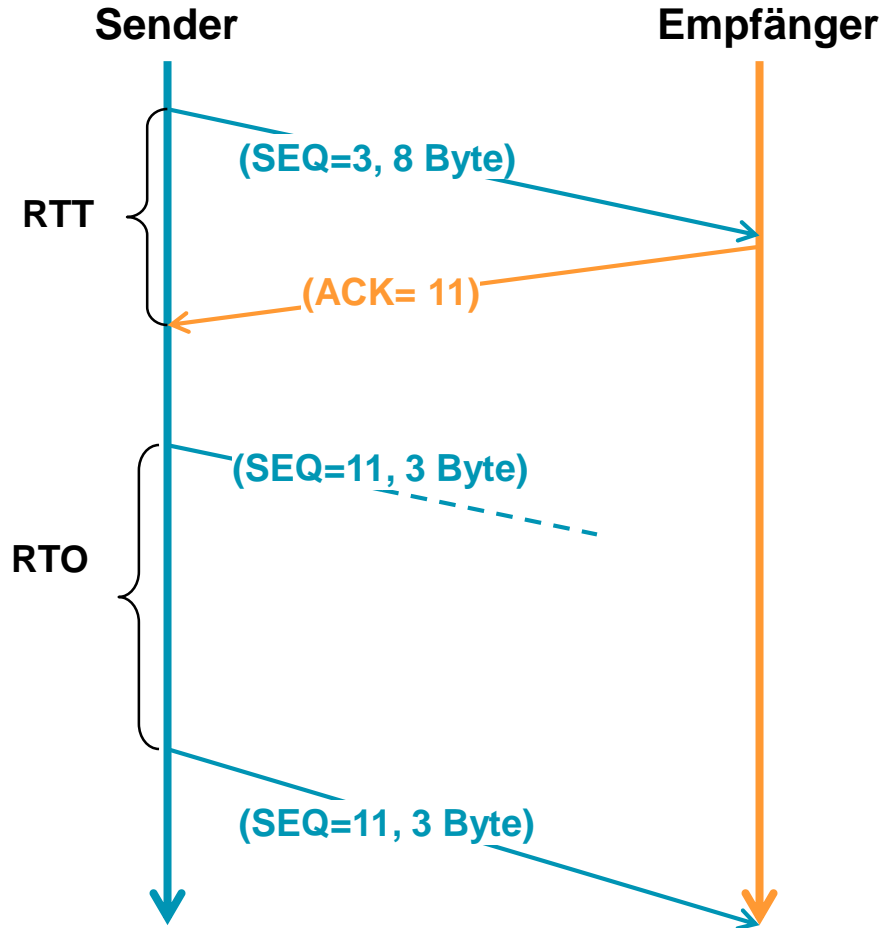


# Modul 4: TCP-Retransmission

## TCP Retransmission - Aufgabenstellung



- **Faustregel:**  
RTO etwas größer als erwartete RTT
- RTO zu groß:  
schlechter Durchsatz
- RTO zu klein:  
unnötige  
Übertragungswiederholung

**Also:** Für effizientes TCP gute  
Schätzung der RTT notwendig

**Fragen:**

- Anfangswert für RTO ?
- Wie soll zeitliche Änderung der RTT berücksichtigt werden ?
- Wie soll Timer nach Ablauf des RTOs gesetzt werden ?



## RTO-Timer nach RFC 793 (1981)

**Schritt 1: Kontinuierliche Mittelwertbildung der RTT  
(EWMA = Exponentially Weighted Moving Average):**

$$SRTT_{akt} = \alpha * SRTT_{old} + (1-\alpha) * RTT_{akt}$$

**SRTT** Smoothed Round Trip Time

$\alpha$

smoothing Faktor (typisch 0,8 – 0,9 z.B. 7/8)

**Schritt 2: Varianzfaktor (+ Limits)**

$$RTO = \min(o\text{-grenz}, \max(u\text{-grenz}, \beta * SRTT))$$

$\beta$

Varianzfaktor (typisch 2)

u-grenz

minimales RTO (z.B. 1 msec)

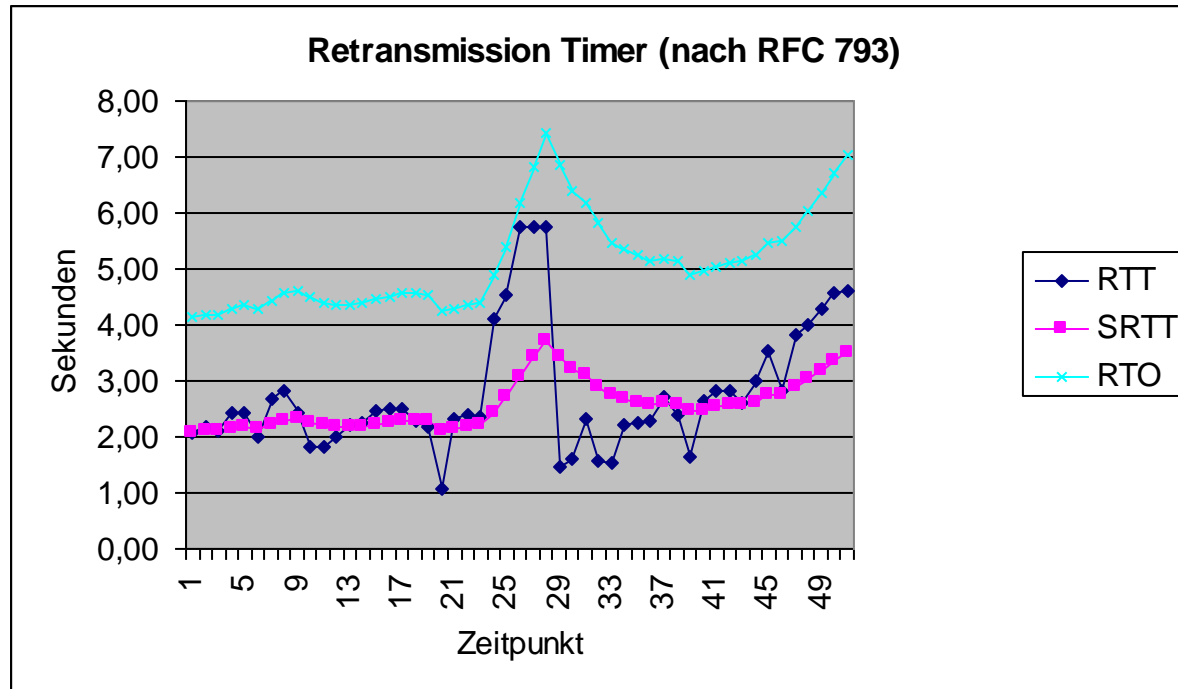
o-grenz

maximales RTO (z.B. 10 sec)

**Bemerkungen:**

- mathematisch einfache, fortlaufende Berechnung (Integer mögl.),
- bei geringen RTT-Schwankungen liegt RTO stets zu hoch,
- Problem der Initialisierung/Reinitialisieren des RTO-Timers

## Beispielberechnung

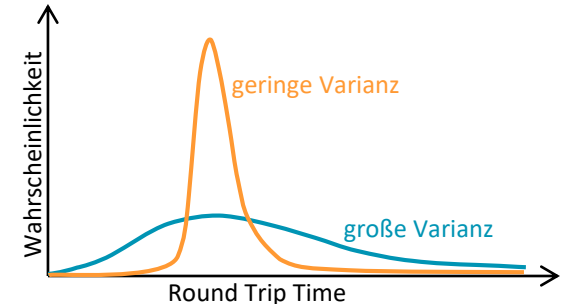


**Interpretation und kritische Diskussion der Kurve !!**

## RTO-Timer nach Jacobson (1988)

### Schlüsselbeobachtung:

- Bei großer Last (starke Schwankung der RTT) ist größere Sicherheitszugabe für RTO notwendig
- Bei hoher RTT und geringer Varianz der RTT ist eine geringe Sicherheitszugabe notwendig



### Lösungsidee:

- Mache den Wert des RTOs abhängig von der Standard-Abweichung

$$\sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \mu)^2}$$

### Problem:

- Berechnung der Standardabweichung ist recht komplex  
→ nehme „mittlere“ Abweichung



## RTO-Timer nach Jacobson/Karel (1988)

### 1. Kontinuierliche Mittelwertbildung der RTT (EWMA-Verfahren):

$$\text{SRTT}_{\text{akt}} = \alpha * \text{SRTT}_{\text{old}} + (1-\alpha) * \text{RTT}_{\text{akt}}$$

**SRTT** Smoothed Round Trip Time  
 **$\alpha$**  smoothing Faktor Mittelwertbildung  
(typisch 0,8 – 0,9 z.B. 7/8)

### 2. Kontinuierliche Schätzung der mittleren Abweichung nach EWMA-Verfahren:

$$\text{SDEV}_{\text{akt}} = \beta * \text{SDEV}_{\text{old}} + (1-\beta) * (\text{abs}(\text{RTT}_{\text{akt}} - \text{SRTT}_{\text{akt}}))$$

**SDEV** smoothed Deviation  
 **$\beta$**  smoothing Faktor Standardabweichung  
(typisch 0,8 – 0,9 z.B. 7/8)

### 3. Sicherheitsspielraum (Erfahrungswert):

$$\text{RTO}_{\text{akt}} = \text{SRTT}_{\text{akt}} + \gamma \text{SDEV}_{\text{akt}}$$

**SDEV** smoothed Deviation  
 **$\gamma$**  Spielraumfaktor (typisch 4)



## Initialisierung des RTOs nach RFC 1122 (October 1989)

The following values **SHOULD** be used to initialize the estimation parameters for a **new** connection:

- (a) **RTT = 0** seconds.
- (b) **RTO = 3** seconds. (The **smoothed variance** is to be initialized to the value that will **result in this RTO**).

The recommended upper and lower bounds on the RTO are known to be inadequate on large internets. The **lower bound** **SHOULD** be **measured in fractions of a second** (to accommodate high speed LANs) and the **upper bound should be  $2 \cdot \text{MSL}$** , i.e., 240 seconds.

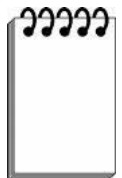


## Aufgabe zur Initialisierung des RTOs

**Gegeben seien**

- die smoothing Faktoren  $\alpha=7/8$  und  $\beta=7/8$  sowie
- der Spielraumfaktor  $\gamma=4$  .

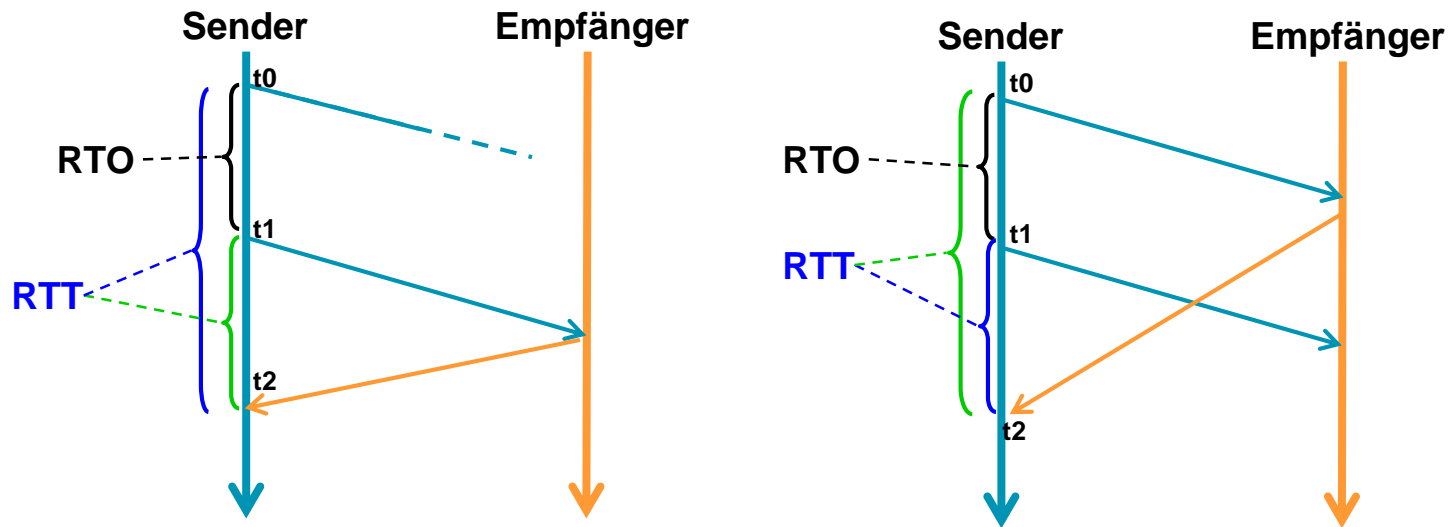
- (a) Setzen Sie die Startwerte für  $SRTT_0$  ,  $RTT_0$  und  $SDEV_0$  gemäß RFC 1122 fest.
- (b) Als erster RTT-Wert wird 2 Sekunden gemessen (also  $RTT_1=2$ ). Berechnen Sie nach dem Algorithmus von Jacobson (1988) den neuen RTO-Werte (also  $RTO_1$ ).





## Problem „acknowledgement ambiguity“ und Karns Algorithmus

- Problem bei Retransmission: Nach Neuübertragungen können ACKs nicht mehr eindeutig zugeordnet werden. Probleme, wenn adaptiver Jacobson-Algorithmus mit „falscher“ RTT fortgesetzt wird.



Frage:  $RTT_{\text{new}} = t2 - t0$  ?

$RTT_{\text{new}} = t2 - t1$  ?

oder Alternativen ?

## Problem „acknowledgement ambiguity“ und Karns Algorithmus

### Alternative 1: $RTT_{new} = t2 - t0$

- Bei wiederholten Paketverlusten wächst RTO unverhältnismäßig stark  
→ Ineffizienz

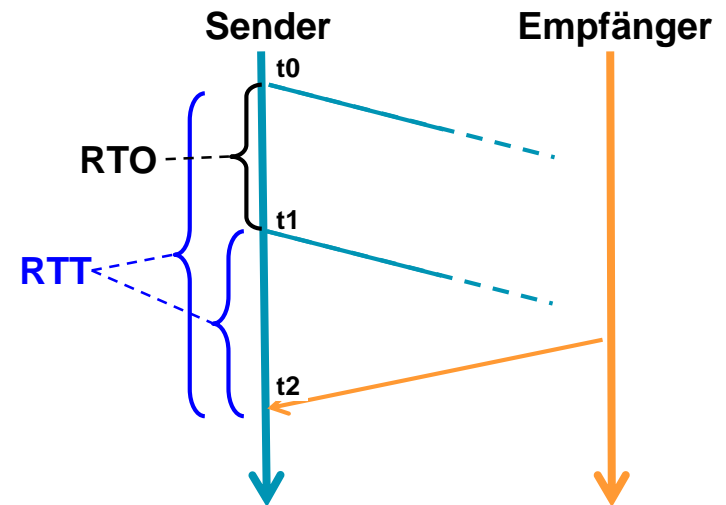


### Alternative 2: $RTT_{new} = t2 - t1$

- RTO wird zu klein → Pakete werden ohne Grund wiederholt übertragen

### Alternative 3: Ignoriere ACK von wiederholt übertragenem Segment

- kein adaptives Verhalten, wenn RTT bei Internet-Überlast steigt





## Problem „acknowledgement ambiguity“ und Karns Algorithmus

**Alternative 4: Ignoriere ACK von erstem wiederholt übertragenem Segment, setze RTO auf  $2 \cdot \text{MSL}$  und warte bis erstes Segment übertragen worden ist.**

- **Schlechter Durchsatz, da sehr hohe Wartezeiten bei Paketverlusten.**

**Lösung „Karns Algorithmus“ mit folgendem Backoff-Verfahren:**

- 1. Das ACK eines wiederholt übertragenen Pakets wird für die Berechnung von SRTT ignoriert ("schlechtes ACK, führt zu zweideutigem RTT"). D.h. das adaptive Verfahren zur Bestimmung des RTO wird ausgesetzt.**
- 2. Das RTO wird verdoppelt.**
- 3. Erst bei korrekter unzweideutiger Bestätigung wird das korrekte RTT für die Berechnung der SRTT wieder verwendet.**
- 4. → Karns Algorithmus gemäß:  
Karn, P. and C. Partridge, "Improving Round-Trip Time Estimates in Reliable Transport Protocols", SIGCOMM 87 verwendet.**



## Die RTO-Berechnung wurde mehrfach optimiert

**RFC 793 (09/81): Transmission Control Protocol.**  
*RTO gemäß Folie 3*

**Artikel:** Jacobson, V., "Congestion Avoidance and Control", *Computer, Communication Review*, vol. 18, no. 4, pp. 314-329, Aug. 1988.  
*Theoretische Grundlage für RTO gemäß Folie 6*

**RFC 1122 (10/89): Requirements for Internet Hosts – Communication Layers.**  
*RTO gemäß Folie 6*

**RFC 6298 (06/11): Computing TCP's Retransmission Timer.**  
*Abschließender Standard für RTO*