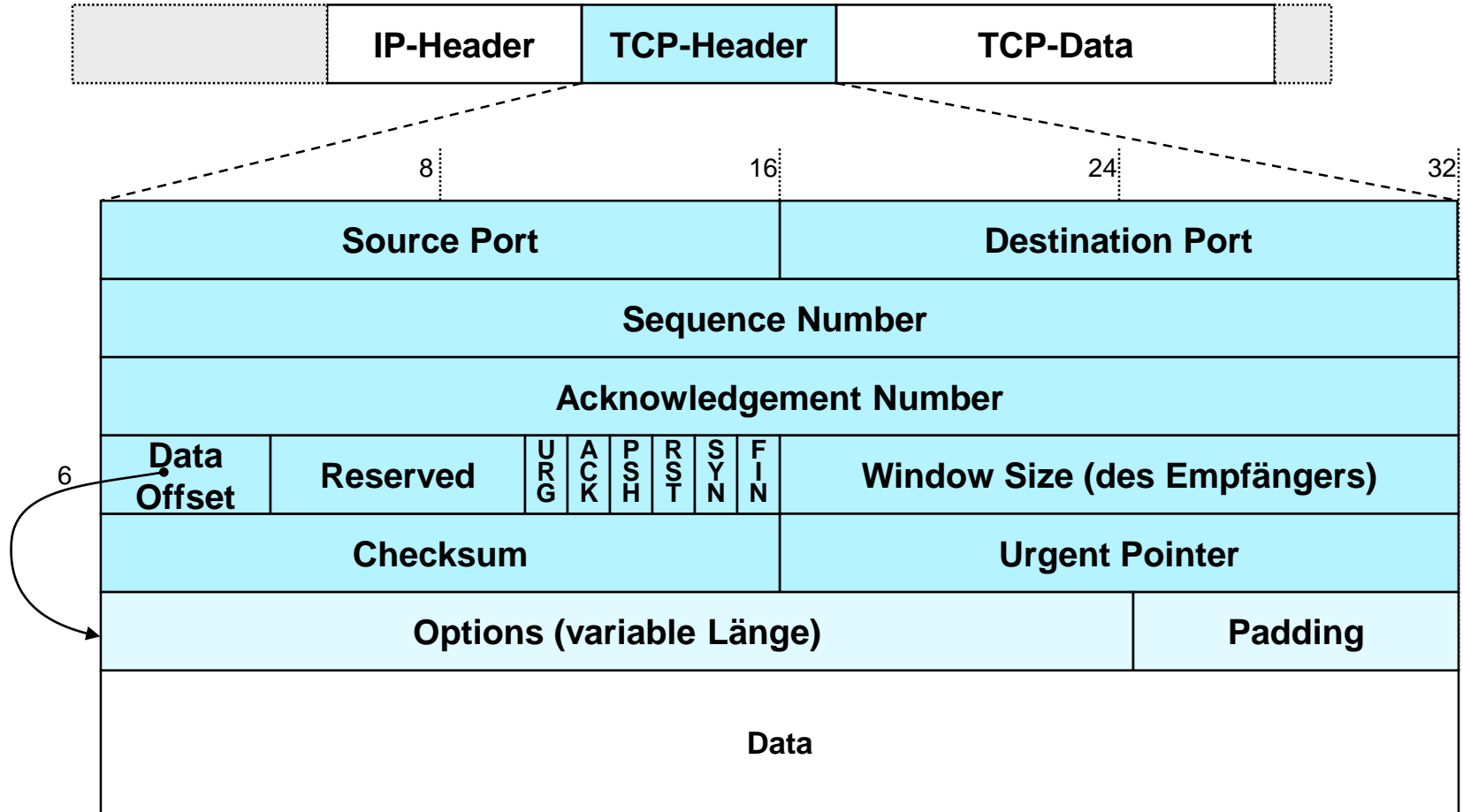


# Modul 11: TCP-Optionen



## TCP-Optionen





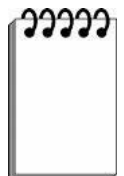
## TCP-Optionen: Überblick

- **Optionen, die nicht unterstützt werden, werden – ohne Fehlermeldung – einfach ignoriert**
- **Optionen mit falschen Parametern sollen zu keinen Problemen führen ("crashing").  
Maximal Rücksetzen der TCP-Verbindung ist erlaubt.**
- **TCP-Optionen können in jedem TCP-Segment gesendet werden.**
- **Der TCP-Optionenteil kann mehrere TCP-Optionen enthalten.  
Die Optionen sind (bis auf zwei Ausnahmen) TLV kodiert.**



## TCP-Optionen: Überblick

Tag	L.	Bedeutung	rfcs
0	-	Ende der Optionenliste	rfc793
1	-	No-Operation	rfc793
2	4	Maximum Segment Size	rfc793
3	3	Window Scale (Unterstützung großer Empfängerfenster + Schutz vor PAWS)	rfc7323, 09/14
4	2	SACK Permitted	rfc2018, 10/96
5	N	SACK	rfc2018, 10/96
8	10	Timestamps (alter RFC: 1323)	rfc7323, 09/14
16		Skeeter ( <i>mysteriöse Optionen: options are parts of an encryption system</i> )	Stev_Knowles
17		Bubba ( <i>mysteriöse: "Stev Knowles did most likely not want to be reminded"</i> )	Stev_Knowles
27	8	Quick-Start Response (Höhere Senderaten, wenn ungenutzte Bandbreite)	rfc4782, exp., 01/07
28	4	User Timeout Option (legt fest, wie lange übertragene Daten unbestätigt bleiben können, bevor Verbindung gewaltsam geschlossen wird.)	rfc5482, 03/09
29		TCP Authentication Option (TCP-AO)	rfc5925, 03/09
30	N	Multipath TCP (MPTCP)	rfc6824, 01/13
34	var.	TCP Fast Open (TFO)	rfc7413, exp., 12/14
253	N	RFC3692-style Experiment 1 (für lokale oder experimentelle Anwendung)	
254	N	RFC3692-style Experiment 2 (für lokale oder experimentelle Anwendung)	





## TCP-Option Selective Acknowledgment (SACK)

### ● Problem:

- Mehrfacher Paketverlust schlägt massiv auf die TCP-Performance durch. Warum?

### ● Lösungskonzept:

- Der Sender wird über alle Segmente informiert, die erfolgreich übertragen wurden.
- Der Sender überträgt gezielt alle Segmente die tatsächlich fehlen

### ● Umsetzung:

- Die SACK-Permitted Option erlaubt der Gegenseite die Benutzung von SACK
- Die SACK-Permitted Option darf nur im SYN gesendet werden.
- Mit der SACK Option informiert der Empfänger den Sender über nicht-zusammenhängende Datenblöcke
- Bei geringer Loss-Rate ist SACK recht effizient (da nur über nicht-zusammenhängende, "isolierte" Blöcke informiert wird.



## TCP-Option Timestamps (TSopt)

- **Probleme:**
  - Messung der RTT-Zeit (→ ACK Ambiguity)
  - Große Empfängerfenster (→ Erkennung von Duplikaten)
- **Lösungskonzept:**
  - Der Sender und Empfänger benutzen in jedem gesendeten Segment die Zeitstempel-Option
- **Umsetzung:**
  - Die Option enthält zwei Timestamp-Felder  $TS_{val}$  und  $TS_{seq}$  der Länge 4 Byte
  - Das Feld  $TS_{val}$  enthält den aktuellen Wert der timestamp-Uhr.
  - Das Feld  $TS_{seq}$  ist nur gültig, wenn das ACK Bit gesetzt ist. In diesem Fall enthält es den  $TS_{val}$  Wert der Gegenseite
  - Sobald die Timestamp-Option einmal erfolgreich genutzt wurde, muss sie fortlaufend weiterverwendet werden.

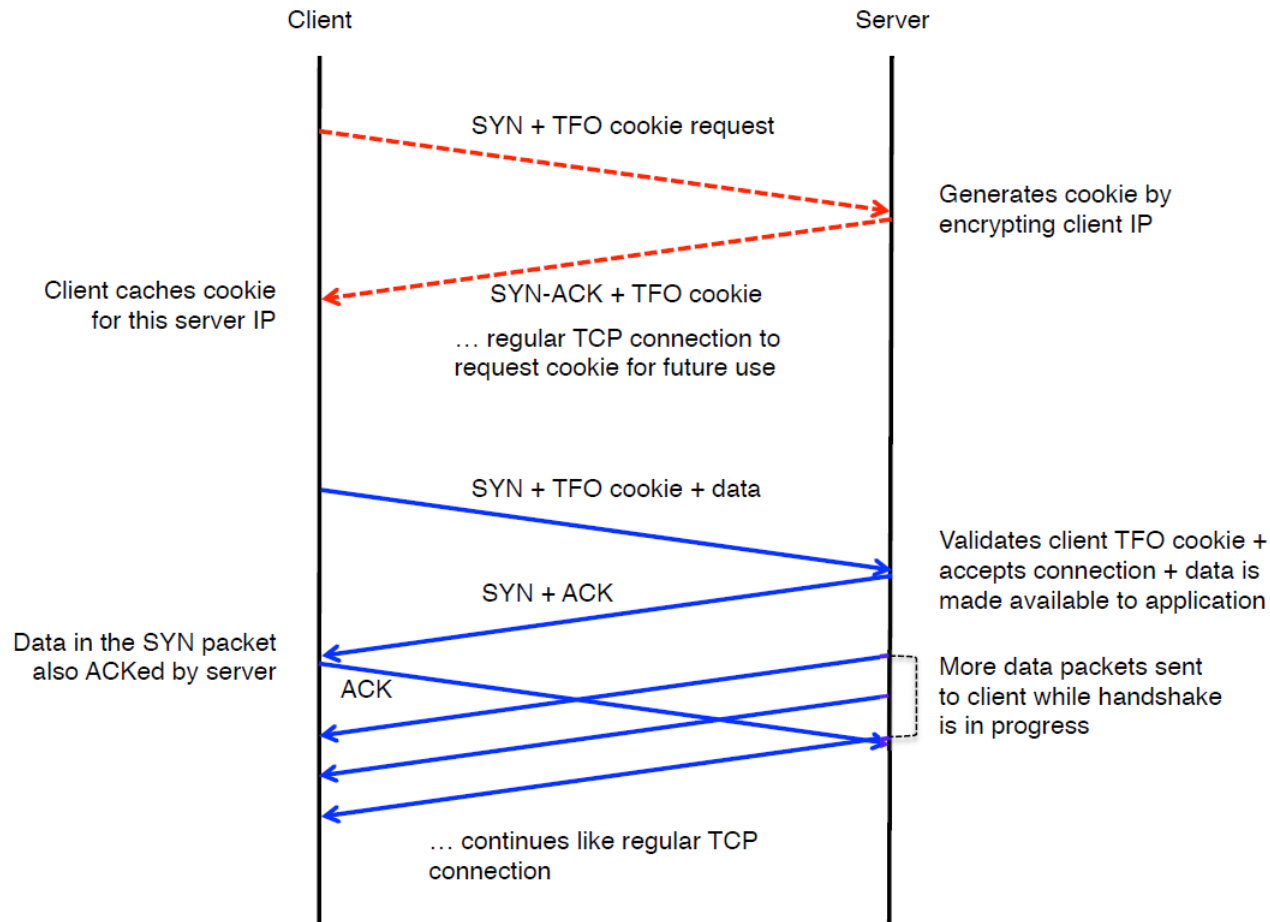


## TCP-Option Fast Open (TFO)

- **Beobachtung in Browserstudie (\*):**
  - HTTP-persistent connections bleiben 5-10 Minuten "idle"
  - 3WHS ist zu 25% für die HTTP-Latenz verantwortlich
- **Ziele:**
  - schnelles Öffnen einer nachfolgenden TCP-Verbindung
  - Austausch von Daten bereits während des 3WHS
- **Problem:**
  - DoS-Angriffe werden erleichtert, wenn Server "früh" aktiv wird.
- **Lösungskonzept:**
  - Einsatz von Krypto-Cookies ("Fast Open Cookie")
  - Das Fast Open Cookie wird vom Server als Funktion der IP-Adresse + Secret erzeugt. Es wird nach einer gewissen Zeit ungültig
  - Der Client cached das Cookie
  - Problem beim Load-Balancing. Server in der Farm müssen gleiches Cookie benutzen.
- **Einsatz:**
  - Verlangt Unterstützung von Client+Server (ab Linux 3.7, 12/12)
  - Signifikante Leistungssteigerung

(\*) Radhakrishnan, S., Cheng, Y., Chu, J., Jain, A., and B. Raghavan, "TCP Fast Open", in Proceedings of the 7th ACM CoNEXT Conference, December 2011.

## TCP-Option Fast Open (TFO) – Protokollablauf im Überblick



from: Sivasankar Radhakrishnan: Network Performance Improvements For Web Services – An End-to-End View, UNIVERSITY OF CALIFORNIA, SAN DIEGO, dissertation, 2014