TCP-Verbindungsaufbau und TCP-Verbindungsabbau – Trainer

TCP-Verbindungsaufbau und TCP-Verbindungsabbau – Trainer

Dieser interaktive Trainer gehört zu den Aufgaben:

> Frage 2: TCP-Verbindungsaufbau mit Sequenznummern
> Frage 3: TCP-Verbindungsabbau mit Sequenznummern

Die Grundlage ist die Klausurübung Netzwerktechnik Schicht 4–7. Dort werden beim TCP-Aufbau und TCP-Abbau die Begriffe `SYN`, `FIN`, `seq`, `ACK`, `x`, `x+1`, `y` und `y+1` verwendet.

Interaktiver TCP-Trainer

https://trainer.ulrich-wiki.com/tcp-verbindung-trainer.html?v=2

TCP-Trainer im Vollbild öffnen

Was wird trainiert?

Bereich	Bedeutung
SYN	Startet den TCP-Verbindungsaufbau
SYN ACK	Server bestätigt SYN und sendet eigene Sequenznummer
ACK	Bestätigung einer Sequenznummer
FIN	Beendet eine Richtung der TCP-Verbindung
seq	Sequenznummer des sendenden Hosts
ACK-Nummer	Bestätigt die nächste erwartete Sequenznummer
x+1 / y+1	SYN und FIN verbrauchen jeweils eine Sequenznummer

Merksatz für den TCP-Verbindungsaufbau

> 1. Client → Server: SYN, seq = x
> 2. Server → Client: SYN + ACK, seq = y, ack = x+1
> 3. Client → Server: ACK, seq = x+1, ack = y+1

Beispiel:

```
x = 1000
y = 2000

x+1 = 1001
y+1 = 2001
```

Merksatz für den TCP-Verbindungsabbau

```
1. Client → Server: FIN, seq = x
2. Server → Client: ACK, ack = x+1
3. Server → Client: FIN, seq = y
4. Client → Server: ACK, ack = y+1
```

Beispiel:

```
x = 1000
y = 2000

x+1 = 1001
y+1 = 2001
```

Wichtige Regel

```
SYN verbraucht eine Sequenznummer.
FIN verbraucht eine Sequenznummer.
Deshalb wird jeweils mit +1 bestätigt.
```

Typische Fehler

Fehler	Warum falsch?
SYN ohne ACK im zweiten Schritt	Der Server muss das SYN des Clients bestätigen
ACK-Zahl nicht +1	SYN/FIN wurden nicht korrekt bestätigt
Pfeilrichtung falsch	Sender und Empfänger werden vertauscht

Fehler	Warum falsch?
FIN-Abbau nur mit 3 Schritten	Der normale TCP-Abbau wird meist mit 4 Schritten dargestellt
seq und ACK verwechselt	seq ist die eigene Nummer, ACK bestätigt die Gegenseite

---

Mini-Testfragen

1. Wofür steht SYN?

`SYN` steht für Synchronisation.

Es wird beim TCP-Verbindungsaufbau verwendet, um Sequenznummern zu synchronisieren.

2. Was passiert beim zweiten Schritt des TCP-Verbindungsaufbaus?

Der Server antwortet mit:

```
SYN + ACK
seq = y
ack = x+1
```

Damit bestätigt er das SYN des Clients und sendet seine eigene Sequenznummer.

3. Warum wird aus x die ACK-Nummer x+1?

Weil SYN eine Sequenznummer verbraucht.

Wenn der Client `seq = x` sendet, erwartet der Server danach `x+1`.

4. Wofür steht FIN?

`FIN` steht für Finish.

Es wird verwendet, um eine TCP-Verbindung beziehungsweise eine Kommunikationsrichtung zu beenden.

5. Warum wird FIN ebenfalls mit +1 bestätigt?

Weil FIN ebenfalls eine Sequenznummer verbraucht.

Darum wird ein FIN mit der nächsten erwarteten Nummer bestätigt:

```
ack = x+1
```

6. Wie viele Schritte hat der TCP-Verbindungsaufbau?

Der TCP-Verbindungsaufbau hat 3 Schritte.

```
SYN
SYN + ACK
ACK
```

7. Wie viele Schritte hat der TCP-Verbindungsabbau?

Der TCP-Verbindungsabbau wird typischerweise mit 4 Schritten dargestellt.

```
FIN
ACK
FIN
ACK
```

Nächste sinnvolle Trainer

Danach könnten folgen:

```
Firewall-Regel-Trainer
PNAT-Trainer
TCP/UDP/ICMP-Vergleichstrainer
```

Revision #1
Created 1 June 2026 12:04:41 by Admin
Updated 3 June 2026 06:18:12 by Admin