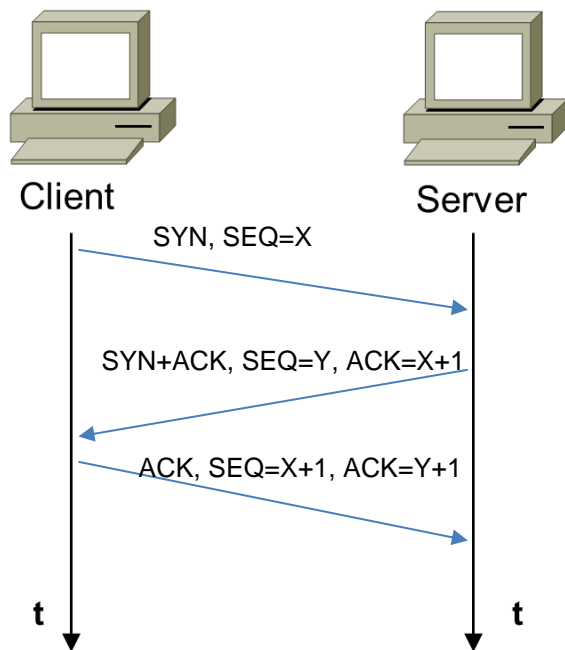


Dozent: Prof. Dr. Michael Massoth Datum: 04.06.2025	Übung-10
Vorname: <u>David</u> Nachname: <u>Schirrmeister</u>	UDP und TCP Punkte (von 21):

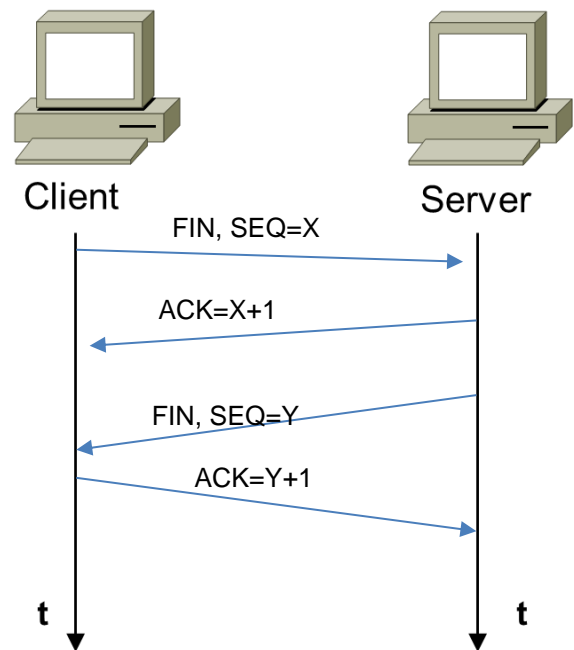
Aufgabe 1: (6 P = 30 * 0,2 P) **TCP Verbindungsaufbau und -abbau**

Skizzieren Sie den TCP verbindungs**aufbau** und **-abbau**. Denken Sie dabei auch an die Flags [SYN, ACK, FIN] und Sequenz- und Acknowledgments-Nummern [SeqNr, AckNr]:

TCP Verbindungsaufbau



TCP Verbindungsabbau



Aufgabe 2: (2 P) **TCP Fehlerbehandlung**

Durch welchen Algorithmus erfolgt die Fehlerbehandlung bei TCP?

TCP verwendet Fehlererkennung und -behandlung durch:

- Checksummen im Header zur Fehlererkennung
- Positive ACKs und Wiederholungen (Retransmission)
- Timeouts für verlorene Pakete
- Fast Retransmit (bei 3 Duplikat-ACKs)

Aufgabe 3: (5 P = 10 * 0,5 P) **Protokolleigenschaften von TCP und UDP**

Protokolleigenschaften	TCP	UDP	
Ist verbindungsorientiert	Ja	Nein	[Ja/Nein]
Ist verbindungslos	Nein	Ja	[Ja/Nein]
Garantiert eine zuverlässige Zustellung	Ja	Nein	[Ja/Nein]
Garantiert die richtige Reihenfolge	Ja	Nein	[Ja/Nein]
Verfügt über eine Flusssteuerung	Ja	Nein	[Ja/Nein]
Arbeitet mit Portnummern	Ja	Ja	[Ja/Nein]
Verfügt über eine Stauvermeidung	Ja	Nein	[Ja/Nein]
Verfügt über eine Checksumme	Ja	Ja	[Ja/Nein]
Verfügt über Quittungen	Ja	Nein	[Ja/Nein]
Zeigt die Art der Payload-Daten an	Nein (außer durch Portnummer)	Nein (außer durch Portnummer)	[Ja/Nein]

Aufgabe 4: (4 P = 2+2 P) **TCP Flow Control versus Congestion Control**

Erklären Sie den Unterschied zwischen Flusskontrolle (engl. Flow Control) und Überlast- bzw. Staukontrolle (engl. Congestion Control):

Flow Control:

- Verhindert, dass der Empfänger mit Daten überflutet wird.
- Gesteuert durch Window Size im TCP-Header.
- Sender wartet, bis Empfänger wieder „Platz“ signalisiert.

Congestion Control:

- Verhindert Überlastung des Netzwerks (nicht des Empfängers).
- Dynamische Anpassung des cwnd (Congestion Window).
- Bekannte Algorithmen: Slow Start, Congestion Avoidance, Fast Retransmit, Fast Recovery

Aufgabe 5: (2 P) **TCP Silly Window Syndrom**

Erklären Sie das Silly Window Syndrom von TCP [erstmalig beobachtet im Jahr 1984].

Silly Window Syndrom:

- Tritt auf, wenn Empfänger kleine freie Buffer meldet (z. B. 1 Byte), und der Sender sofort kleine Segmente sendet → Ineffizienz.

Lösung:

- Sender: Warten, bis genug Platz für ein ganzes Segment ist (Clark's Algorithmus).
- Empfänger: Vermeidet Senden von ACKs mit zu kleinen Fenstergrößen.

Aufgabe 6: (2 P) **TCP: Round-Trip-Times und Retransmission Timer**

TCP misst für jede Verbindung die Round-Trip-Time (RTT) zwischen dem versendeten Segment und dem zugehörigen Acknowledgement (Ack).

