



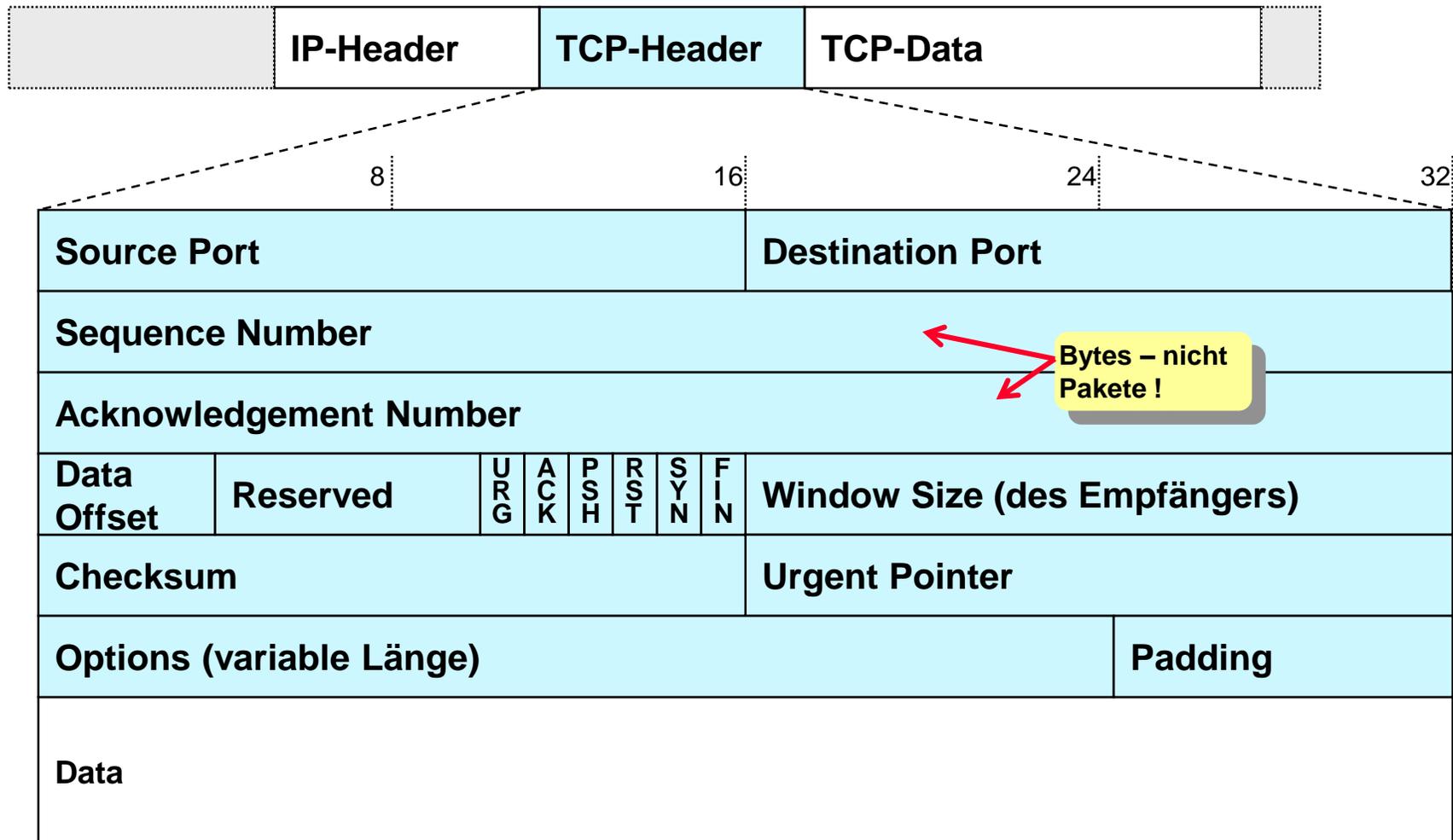
Vorab: Überblick TCP

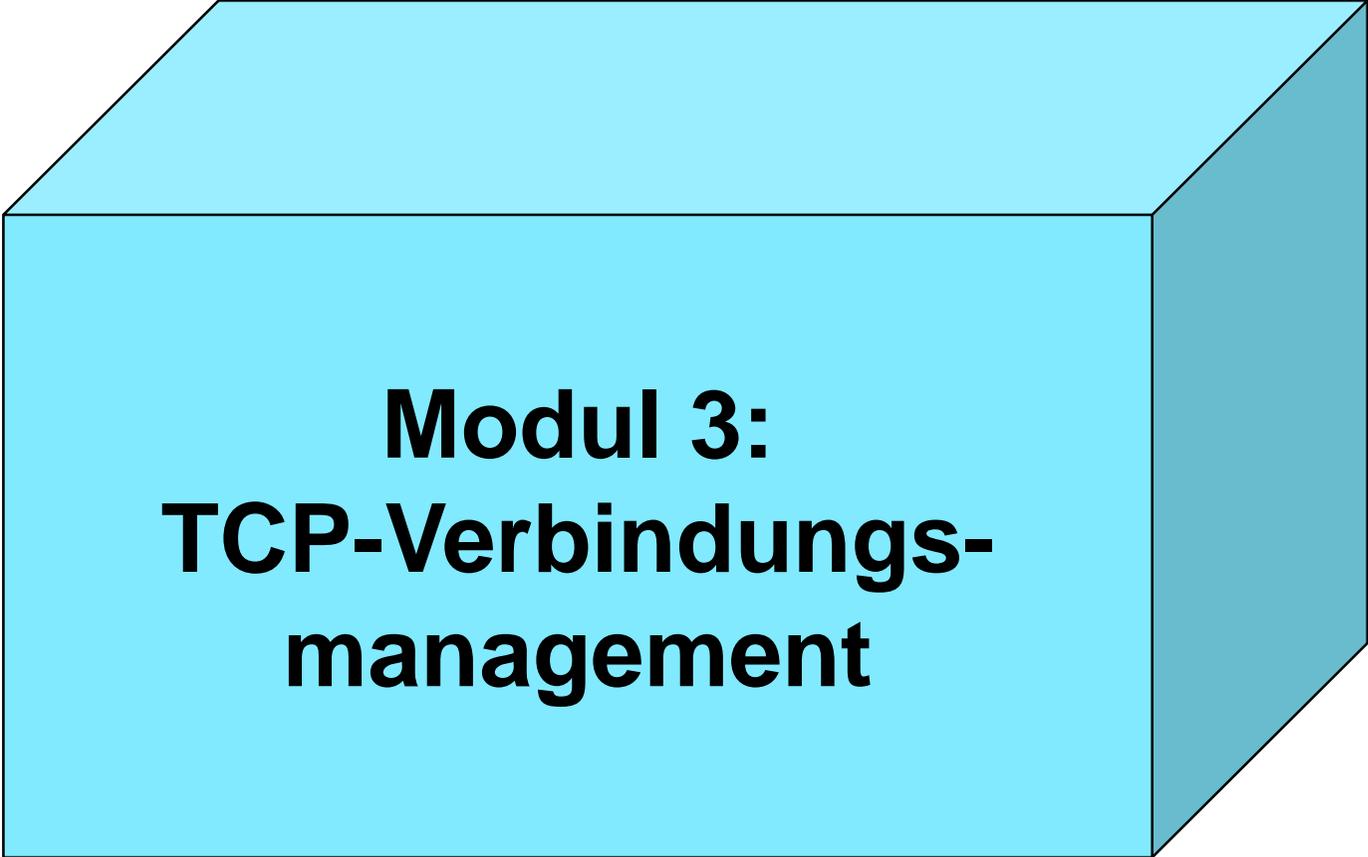
Grundeigenschaften

- **Punkt-zu-Punkt-Verbindung**
- **Streaming-Schnittstelle**
 - Byteorientiert
 - keine Fragment-/Segmentgrenzen
- **Zuverlässige Datenübertragung**
- **Verbindungsorientierte Übertragung**
 - zuverlässiger Verbindungsaufbau, Initialisierung des Empfänger/Sender-Kontexts
 - zuverlässiger Verbindungsabbau
- **Vollduplex-Verbindung**
 - bidirektionaler Datenfluss in derselben Verbindung
 - Optimale maximale Segmentgröße (512-1500 Byte)
- **Dynamische Sende- und Empfangspuffer**
- **Flusskontrolle**
- **Überlastkontrolle**



Vorab als Wiederholung: TCP-Segmenteaufbau

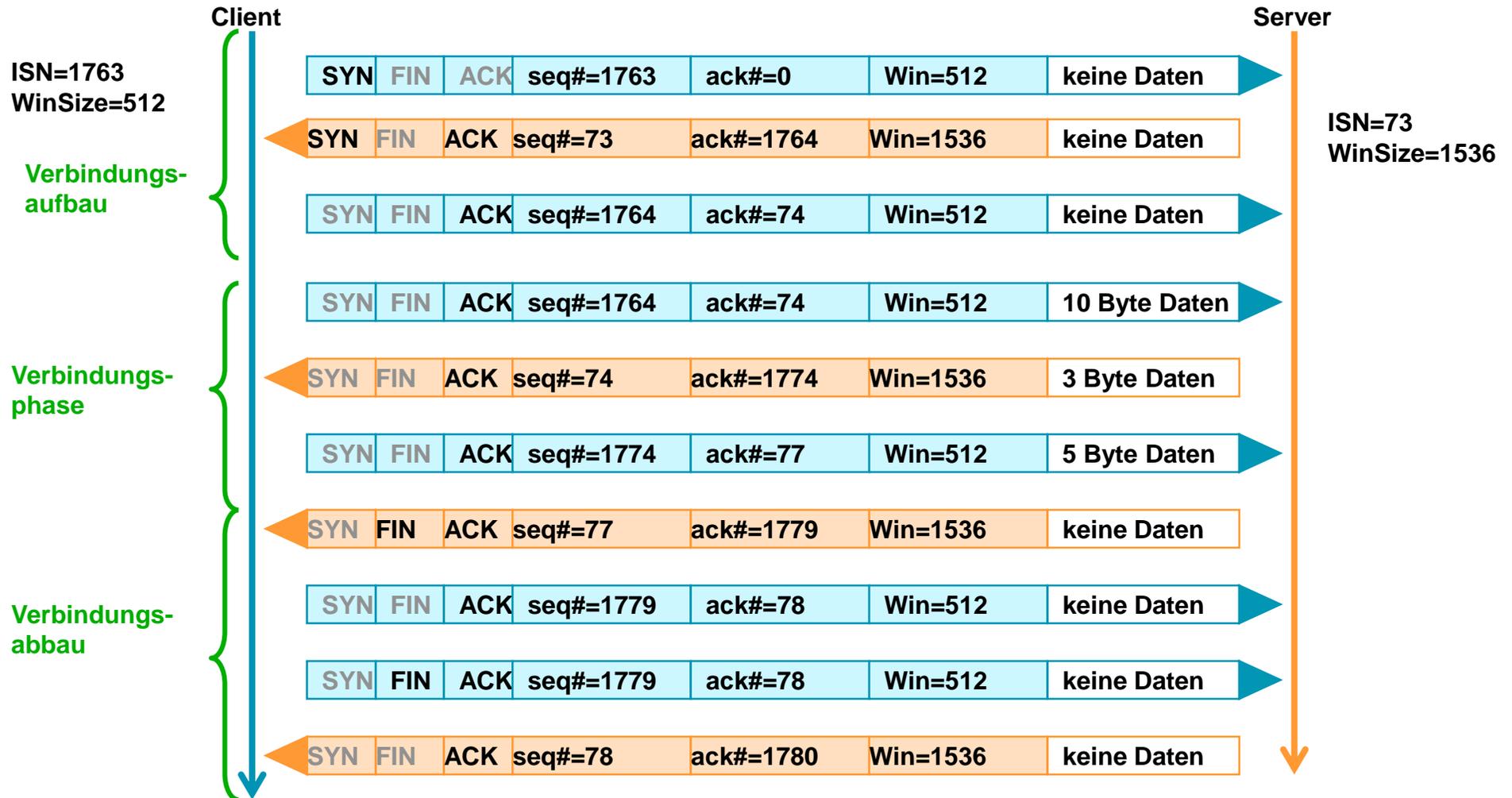




Modul 3: TCP-Verbindungs- management



TCP-Verbindungsablauf (Normalfall, stark vereinfacht)





Zuverlässigkeit bei TCP

- TCP will nicht nur sehr zuverlässig, sondern **absolut zuverlässig** sein.
- TCP will in **allen Protokoll-Phasen** einer Verbindung absolut zuverlässig sein:
 - Verbindungsaufbau
 - Verbindungsphase
 - Verbindungsabbau
- Fragen:
 - Was bedeutet Zuverlässigkeit in den verschiedenen Protokollphasen?
 - Was genau muss in den jeweiligen Phasen sichergestellt sein?
 - Warum ist es denn so schwierig, Zuverlässigkeit zu erreichen?



Kann es nicht Situationen geben, wo sich durch ein Protokoll – aus grundsätzlichen Überlegungen heraus - gar keine zuverlässige Datenübertragung erreichen lässt?

Das Zwei-Armeen-Problem

Situation: Die beiden blauen Armee sind zusammen stärker als die rote Armee, wenn sie gleichzeitig angreifen.

Die beiden blauen Armeen kommunizieren über einen Boten, der von der roten Armee abgefangen werden kann.

Frage: Mit welchem Protokoll können sich die blauen Armeen auf einen Angriffszeitpunkt einigen?

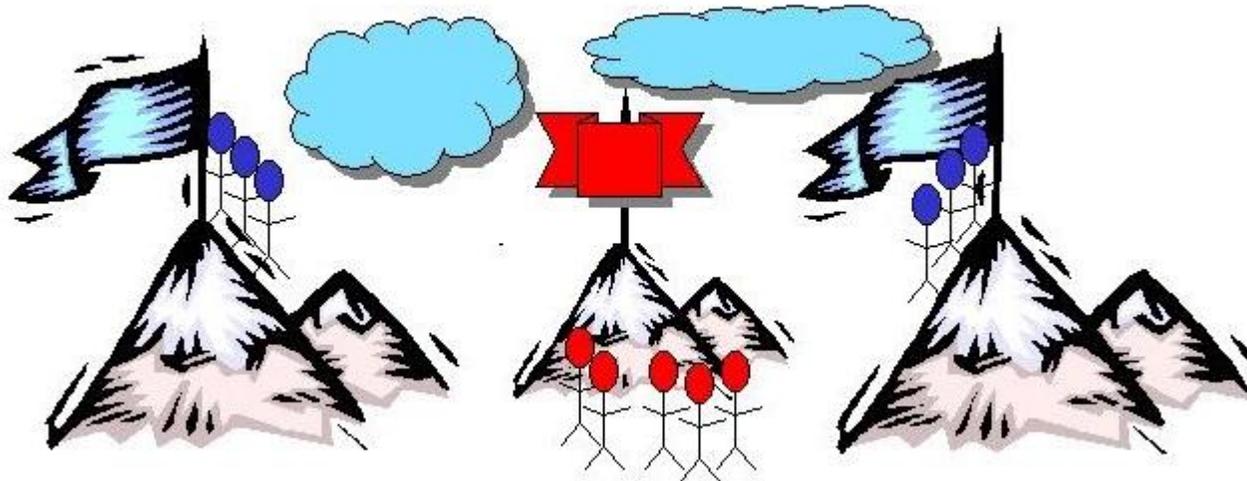


Bild: © Prof. Dr. Knut Barghorn, Fachhochschule Oldenburg, Ostfriesland, Wilhelmshaven. Standort Wilhelmshaven. E-mail: knut.barghorn@fh-ooow.de

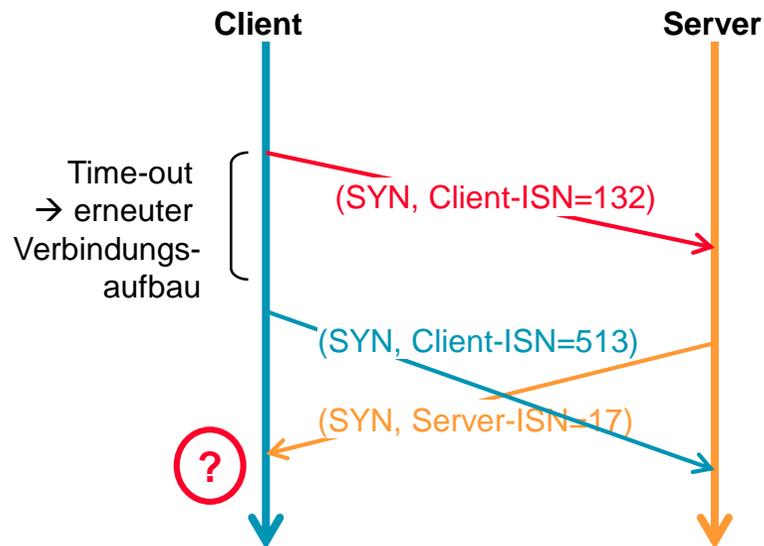


Three-Way-Handshake

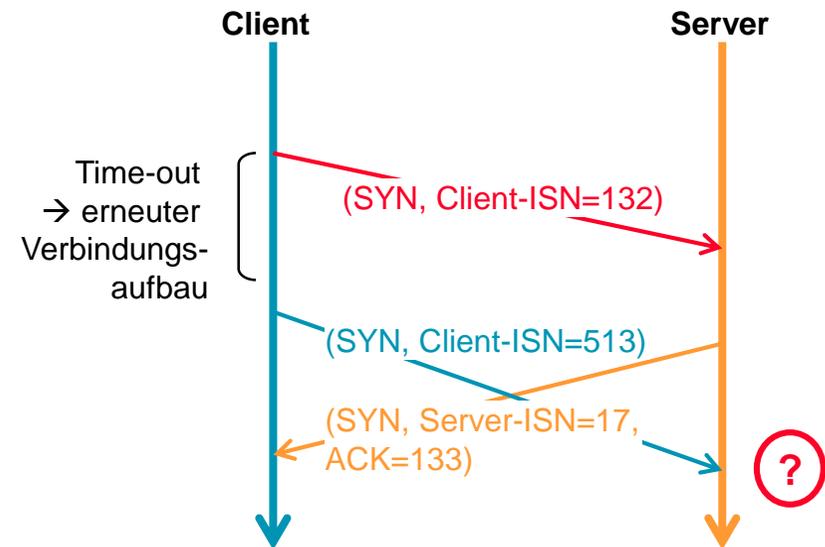
- **Ziel:** sicherer Verbindungsaufbau
d.h. Aufbau eines gemeinsamen Verbindungskontextes als Vorbedingung einer aktiven Verbindung
- **Methode:** Three-Way-Handshake:
 - Schritt 1 (*aktives Öffnen* der Verbindung):
Client sendet SYN mit Client-ISN und Client-Window-Size
 - Schritt 2 (*passives Öffnen* der Verbindung):
Server sendet SYN-ACK, Server-ISN, Server-Window-Size (und alloziert Puffer)
 - Schritt 3: Client sendet ACK (und alloziert Puffer)
- **Sicherheitsaspekt:**
DDOS (Distributed denial of service) Angriffe auf TCP-Server möglich, da in Schritt 2 Ressourcen alloziert werden.
- **Frage:** Geht der sichere Verbindungsaufbau nicht auch einfacher?

Ist ein Two-Way Handshake ausreichend?

**Two-Way-Handshake
ohne Bestätigung der Sequence Number**



**Two-Way-Handshake
mit Bestätigung der Sequence Number**

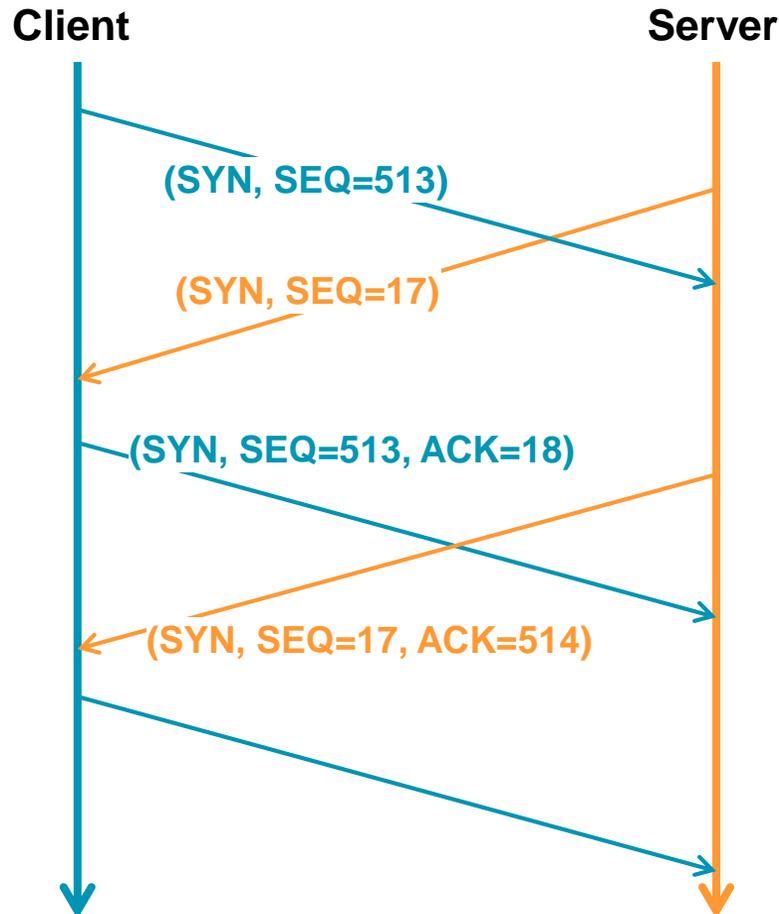


Fragen:

- **Wie wird der zweite Client-SYN behandelt?**
- **Welche ISNs sind für die aktive Verbindung auf Client und Serverseite gültig?**



Gleichzeitiger Verbindungsaufbau



Gleichzeitiger aktiver Verbindungsaufbau stellt kein Problem dar. Ergebnis ist aber eine einzige Verbindung

**Aber Problem der "verzögerten Duplikate" bleibt.
Lösung: Sicherstellen, dass keine Sequenznummer innerhalb der maximalen Paketlebensdauer zweimal vergeben wird (→ Anforderungen an die ISN).**

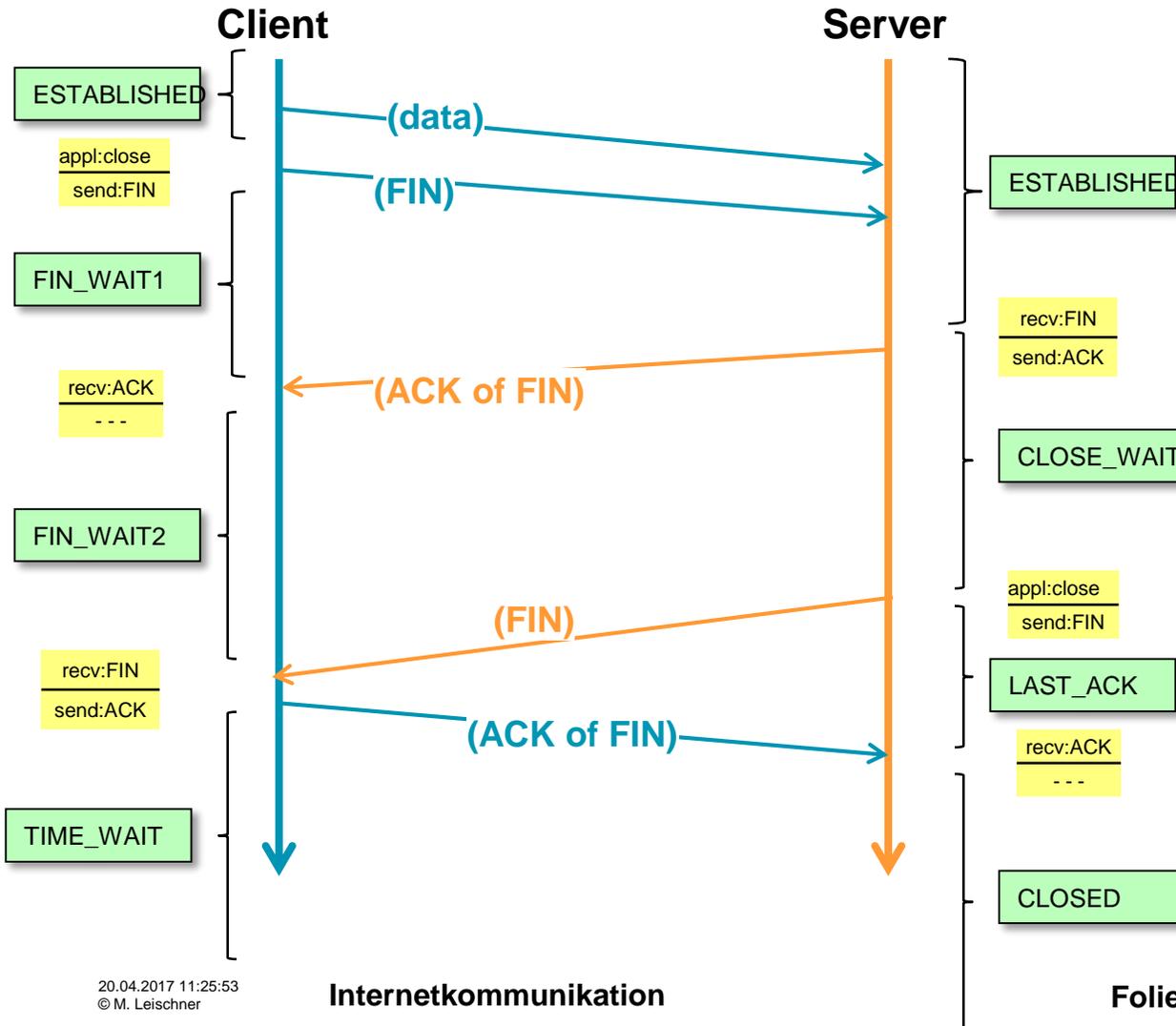


Abbau einer Verbindung – Grundtypen und Problemstellung

- **Grundtypen**
 - **Reset:**
Verbindung wird durch eine Seite "asynchron" beendet . (Problem des Datenverlusts)
 - **Einseitiges Schließen:**
Verbindung signalisiert der Gegenseite, dass sie keine weiteren Daten sendet.
(Problempunkt: Gegenseite kann noch weitere Daten senden)
 - **Asymmetrischer Abbau:**
Einer der beiden Rechner schließt seine Verbindung einseitig und baut gleichzeitig Gesamtverbindung ab.
 - **Symmetrischer Abbau**
jeder der beiden Seiten baut "ihre" unidirektionale Verbindung ab, sobald sich beide Seiten auf Verbindungsabbau geeinigt haben. (→ TCP-Ansatz)
- **Problem des letzten ACKs:**
Wie baue ich eine Verbindung *sicher* ab?
Reicht ein Three-Way-Handshake?
(→ Kernfrage: Ist letzte PDU für das Protokoll wichtig?)
- **Einsatz von Timern**



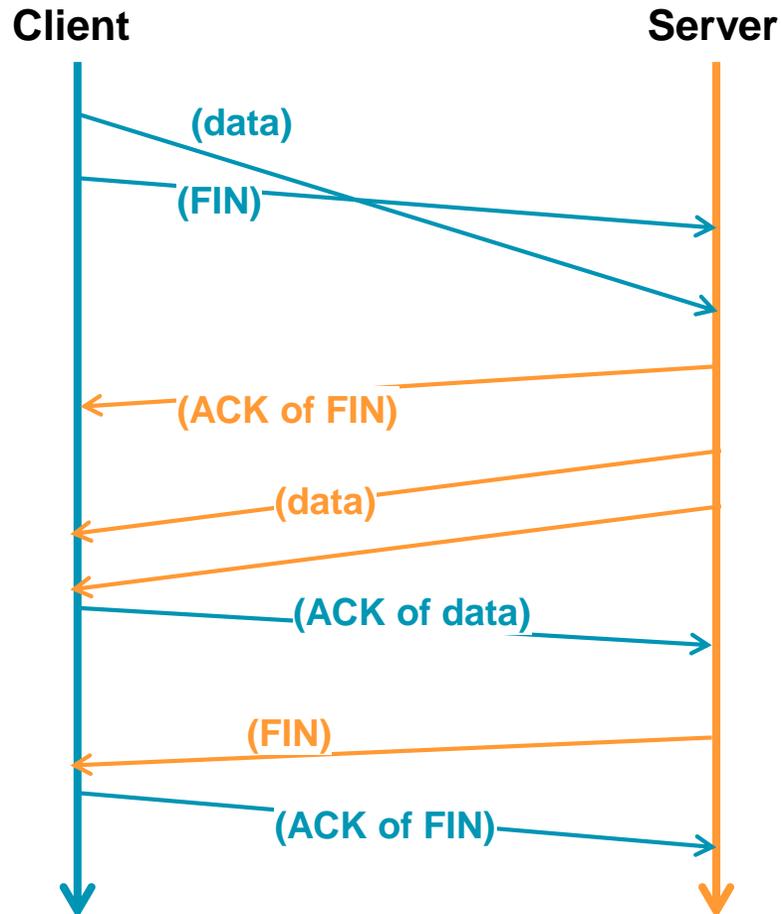
Normalfall TCP-Verbindungsabbau



- Beendigung des **CLOSE_WAIT** Zustandes nur durch die Applikation möglich
- Im **TIME_WAIT** wird **2MSL** gewartet
 - Nochmaliges Senden des ACKs wird ermöglicht.
 - Verbindung (=Client-IP+Socket, Server-IP+Socket) kann erst nach **2MSL** wiederverwendet werden.
- Frage: Was passiert bei Wiederstart des Servers in einem Zeitraum $< 2MSL$?



TCP einseitiger Verbindungsabbau



- mögliche Anwendungen des einseitigem Verbindungsabbaus:
 - Transaktionen
 - RPC
- Aufgabe:
Zeichnen Sie die Zustandsübergänge in das Diagramm ein.