Technische Universität Braunschweig

Institut für Betriebssysteme und Rechnerverbund
Connected and Mobile Systems

Computernetze 1

Übung 10 – Transport Layer

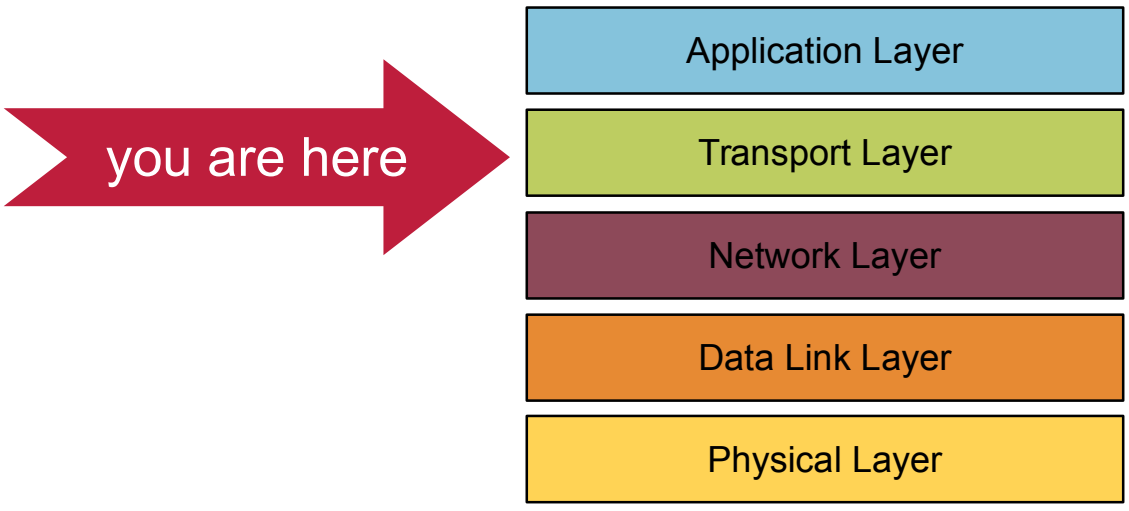
Fynn Schulze, 17. Juli 2025

Technische Universität Braunschweig, IBR

Überblick

- 1) Transportschicht
- 2) UDP und TCP
- 3) Segmentierung/Fragmentierung in TCP/IP
- 4) Sequenznummern und Datenübertragungsrate
- 5) 2-Armeen-Problem
- 6) Verbindungsmanagement
- Weitere Lehrveranstaltungen
- Klausur

Überblick


you are here
Application Layer
Transport Layer
Network Layer
Data Link Layer
Physical Layer

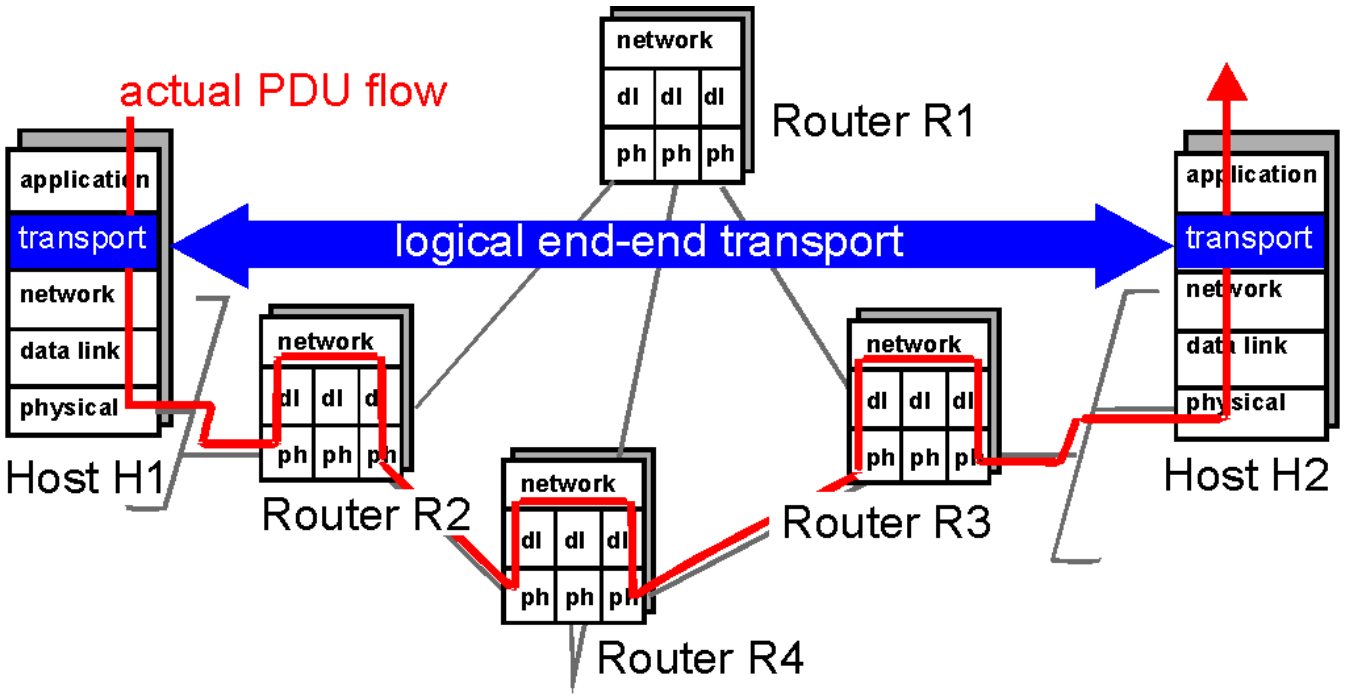
Aufgabe 1: Transportschicht

a) Die Sicherungsschicht (Data Link Layer) verbindet direkt benachbarte Geräte. Die Vermittlungsschicht (Network Layer) verbindet Geräte in einem größeren Netz. Wen oder was verbindet die Transportschicht?

1 a) Aufgabe der Transportschicht

Aufgabe der Transportschicht

Die Transportschicht koppelt die Anwendung (bzw. den Nutzer) mit dem Netzwerk (z.B. über eine Socket-Schnittstelle).



jpkc.ncwu.edu.cn

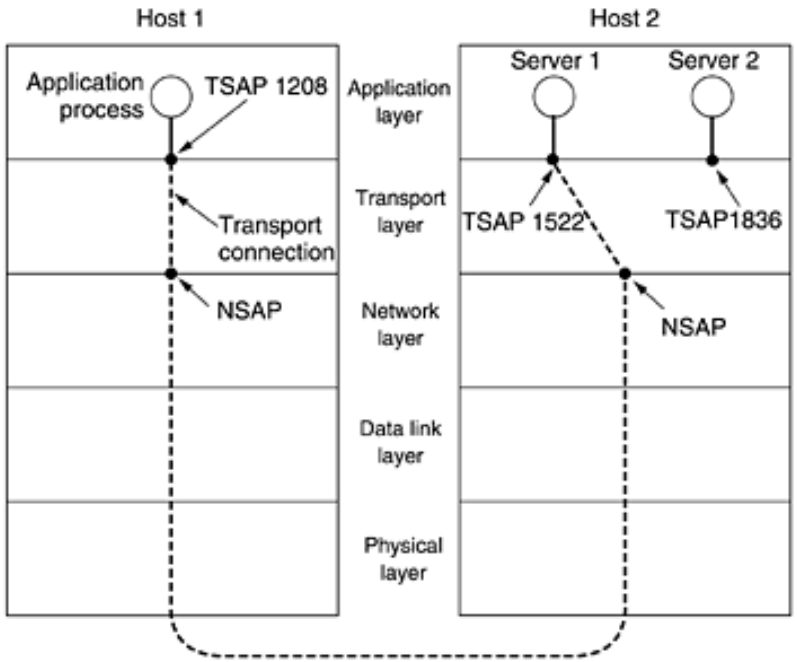
1 b) TSAPs

b) Wofür steht TSAP? Wie heißen die TSAPs im Internet?

Lösung

TSAP steht für **T**ransport **S**ervice **A**ccess **P**oint.

- Schnittstelle zur Interaktion mit der Transportschicht
- Im Internet entsprechen die *Ports* den TSAPs



Andrew S. Tanenbaum, Computer Networks (4th Edition)

1 c) TSAP herausfinden

c) Welche drei Möglichkeiten gibt es, um den TSAP eines Service-Providers zu ermitteln?

Lösung

1. TSAP implizit bekannt (Ports: 0 - 1023 „*Well-known Ports*“)
   ⇒ Beispiel: HTTP Port 80; HTTPS Port 443
2. Initiale Aushandlung des TSAP
   ⇒ FTP: Port 21 (Well-known Port), Datenübertragung dann auf einem ausgehandelten Port.
3. Name Server (Namensdienst): Portmapper + Remote Procedure Call
   ⇒ Port 111: Well-known, dort dann nähere Informationen (z.B. NIS: Port 1016)

Well-known Ports

Auszug aus der /etc/services (Linux/Unix)

```
# Dienst    Port/Protokoll [Aliase]
echo           7/tcp                  # Echo
echo           7/udp                  #
discard        9/tcp  sink null       # Discard
discard        9/udp  sink null       #
daytime       13/tcp                  # Daytime
daytime       13/udp                  #
chargen       19/tcp  ttytst source   # Character Generator
chargen       19/udp  ttytst source   #
ftp-data      20/tcp                  # File Transfer Protocol (Data)
ftp           21/tcp                  # File Transfer Protocol (Control)
telnet        23/tcp                  # Virtual Terminal Protocol
smtp          25/tcp                  # Simple Mail Transfer Protocol
nntp         119/tcp  readnews        # Network News Transfer Protocol
```

Aufgabe 2: UDP und TCP

a) Warum gibt es UDP? Würde es nicht ausreichen, Benutzerprozessen zu erlauben, direkt IP-Pakete zu verschicken?

b) Sowohl TCP als auch UDP verwenden Port-Nummern, um ihre Ziele zu identifizieren. Warum verwendet man Ports als abstraktes Konzept und nicht Prozess-IDs? Geben Sie zwei Gründe an.

c) Erklären Sie die TCP-Flags `SYN`, `ACK` & `FIN`.

d) Was sind die Unterschiede zwischen TCP und UDP?

e) Was sind die Gemeinsamkeiten von TCP und UDP?

f) Beschreiben Sie die Gemeinsamkeiten und Unterschiede der Flusskontrolle auf der Sicherungsschicht und der Transportschicht.

2 a) Warum gibt es UDP?

a) Warum gibt es UDP? Würde es nicht ausreichen, Benutzerprozessen zu erlauben, direkt IP-Pakete zu verschicken?

Lösung

- Nein
- IP Datagramme nutzen IP Adressen (die Hosts identifizieren)
- Ohne zusätzliche Adress-Informationen würde der Vermittlungsschicht-Prozess im Zielnetzwerk nicht wissen, wie er das IP-Paket an die Anwendungsschicht weitergeben soll
- Kommunikation zwischen Prozessen basiert auf Ports
- Ein UDP Paket enthält die Nummer des Zielports

2 b) Warum Ports statt PIDs?

b) Sowohl TCP als auch UDP verwenden Port-Nummern, um ihre Ziele zu identifizieren. Warum verwendet man Ports als abstraktes Konzept und nicht Prozess-IDs? Geben Sie zwei Gründe an.

Lösung

- Prozess-IDs sind nicht statisch
  - Prozess-IDs werden dynamisch festgelegt, wenn ein Prozess erzeugt wird
  - Funktioniert nicht mit „Well-known Ports“
- Prozesse mit mehreren TSAPs könnten nicht umgesetzt werden
  - Jeder Prozess hat nur eine ID

2 c) TCP-Flags

c) Erklären Sie die TCP-Flags `SYN`, `ACK` & `FIN`.

- SYN:
  - Wird zum Verbindungsaufbau genutzt (Sequenznummern synchronisieren)
  - SYN=1 & ACK=0: connection request
  - SYN=1 & ACK=1: connection confirm
- ACK: 0: Keine Ack-No. gesetzt, 1: Ack-No. ist gesetzt
- FIN: Verbindungsabbau

2 d) Unterschiede zwischen TCP und UDP

d) Was sind die Unterschiede zwischen TCP und UDP?

UDP

- Verbindungslos
- Unzuverlässig
- Ungeordnet
- keine Stau/Flusskontrolle
- Übertragung von einzelnen Nachrichten
- Schlank

TCP

- Verbindungsorientiert
- Zuverlässig
- Reihenfolge garantiert
- Stau/Flusskontrolle
- Übertragung von geordneten Byteströmen
- Größerer Overhead

2 e) Gemeinsamkeiten von TCP und UDP

e) Was sind die Gemeinsamkeiten von TCP und UDP?

Gemeinsamkeiten

- Transportprotokolle
- Setzen auf IP auf
- Benutzen Ports als TSAPs

2 f) Flusskontrolle

f) Beschreiben Sie die Gemeinsamkeiten und Unterschiede der Flusskontrolle auf der Sicherungsschicht und der Transportschicht.

Unterschiede

Layer 2

- Nur wenige Verbindungen
- Puffer auf beiden Seiten

Layer 4

- Hosts können viele Verbindungen haben
- Viele Puffer unpraktisch
- Sender muss puffern (bei unzuverlässigem Netzwerk), Empfänger kann puffern

2 f) Flusskontrolle

Gemeinsamkeiten

Eingesetzte Verfahren
- Sliding Window ohne Puffer, mit statischer Pufferverwaltung oder mit dynamischer Puffer-Verwaltung

Zweck
- Bremsen eines schnelleren Senders, um einen langsameren Empfänger nicht zu überfordern

Aufgabe 3: Segmentierung/Fragmentierung in TCP/IP

a) Wie sieht im schlechtesten Fall (1 Byte Nutzdaten) das Verhältnis Nutzdaten zu Gesamtgröße der versendeten Daten aus? Berücksichtigen Sie dabei auch die versendeten Acknowledgements (kein Piggybacking möglich).

b) Wie ist das im Falle von UDP?

c) Eine Anwendung schickt IP-Pakete, die jeweils ein TCP-Segment mit 400 Byte Nutzdaten enthalten. Im Netz existiert ein Bereich, in dem IP-Fragmentierung notwendig ist. Die maximale Paketlänge eines IP-Paketes beträgt in diesem Bereich 150 Byte. Wie viele Nutzdaten sind pro IP-Fragment in diesem Bereich noch enthalten?

3 a) Verhältnis Nutzdaten zur Gesamtgröße

a) Wie sieht im schlechtesten Fall (1 Byte Nutzdaten) das Verhältnis Nutzdaten zu Gesamtgröße der versendeten Daten aus? Berücksichtigen Sie dabei auch die versendeten Acknowledgements (kein Piggybacking möglich).

Lösung

Worst case: 1 Byte Nutzdaten (z.B. remote shell, telnet)

Daten 20 Byte IP-Header + 20 Byte TCP Header (+ evtl. Optionen), 1 Byte Nutzdaten

ACK 20 Byte IP, 20 Byte TCP, 0 Byte Nutzdaten

$\Rightarrow$ Verhältnis 1 : 80 (dabei ist MAC-Header noch vernachlässigt...)

3 b) UDP

b) Wie ist das im Falle von UDP?

Lösung

Worst case: 1 Byte Nutzdaten (z.B. Messdaten)

Daten 20 Byte IP-Header + 8Byte UDP Header, 1 Byte Nutzdaten

ACK 0 Byte (kein ACK)

$\Rightarrow$ Verhältnis 1 : 28

3 c) IP-Fragmentierung

c) Eine Anwendung schickt IP-Pakete, die jeweils ein TCP-Segment mit 400 Byte Nutzdaten enthalten. Im Netz existiert ein Bereich, in dem IP-Fragmentierung notwendig ist. Die maximale Paketlänge eines IP-Paketes beträgt in diesem Bereich 150 Byte. Wie viele Nutzdaten sind pro IP-Fragment in diesem Bereich noch enthalten?

3 c) IP-Fragmentierung

Hinweis

Die Größes des Payloads eines IP-Fragments ist immer ein Vielfaches von 8 Byte. Der TCP-Header zählt mit, sodass nur 108 Bytes (echte) Nutzdaten möglich sind.

- max. IP-Paketgröße: 150 *Byte*
- IP-Header: 20 *Byte*
- max. IP-Payload: 130 *Byte* $\Rightarrow 16 \cdot 8$ *Byte* $= 128$ *Byte*
- TCP-Header: 20 *Byte*
- echte Nutzdaten: 128 *Byte* $-$ 20 *Byte* $=$ 108 *Byte*

3 c) IP-Fragmentierung

Lösung

- Je 20 Byte IP-Header, 20 Byte TCP-Header, 400 Byte Nutzdaten
- Gesamtgröße: 440 Byte
- Fragmentierung auf 150 Byte

1. Paket	20 Byte IP-Header + 20 Byte TCP-Header + 108 Byte Nutzd.
2. + 3. Paket	20 Byte IP-Header + 128 Byte Nutzdaten
4. Paket	20 Byte IP-Header + 36 Byte Nutzdaten

3 c) IP-Fragmentierung

Fazit

⇒ Aus einem Paket werden vier!

(und nicht drei, was herauskommt, wenn man 440 Byte durch 150 Byte teilt...)

⇒ **Paketheader sind zu beachten!**

Aufgabe 4: Seq.-Nr. und Datenübertragungsrate

Zur Vermeidung von Duplikaten bei der Datenübertragung von Paketen werden gewöhnlich individuelle Sequenznummern pro PDU vergeben. Betrachten Sie ein Netz mit einer maximalen Paketgröße von 2048 Byte. Die maximale Netzverweildauer T betrage 90 Sekunden und die Länge der Sequenznummer sei 15 Bit. Wie hoch ist die maximal mögliche Datenübertragungsrate pro Verbindung?

Aufgabe 4: Seq.-Nr. und Datenübertragungsrate

Lösung

- maximalen Paketgröße: 2048 Byte
- maximale Netzverweildauer T: 90 Sekunden
- Länge der Sequenznummer: 15 Bit

$\Rightarrow$ maximal $2^{15} = 32768$ verschiedene Sequenznummern

$\Rightarrow$ halber Sequenznummernraum gleichzeitig nutzbar: 16384 Pakete

- jedes Paket ist maximal 90 Sekunden im Netz

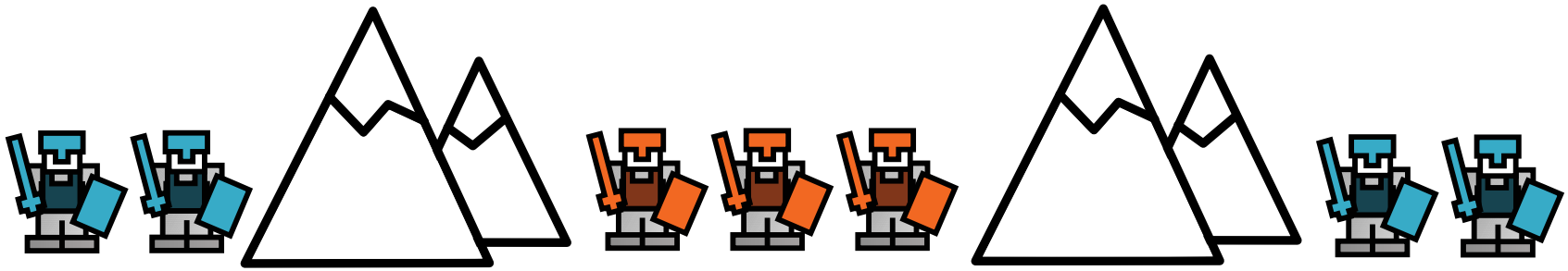
$\Rightarrow$ 16384 Pakete pro 90s $\rightarrow \frac{16384 \cdot 2048 Byte}{90s}$

$\Rightarrow$ 372827 Byte/s = **2,98 Mbit/s**

Aufgabe 5: 2-Armeen-Problem

a) Beschreiben Sie das 2-Armeen-Problem.

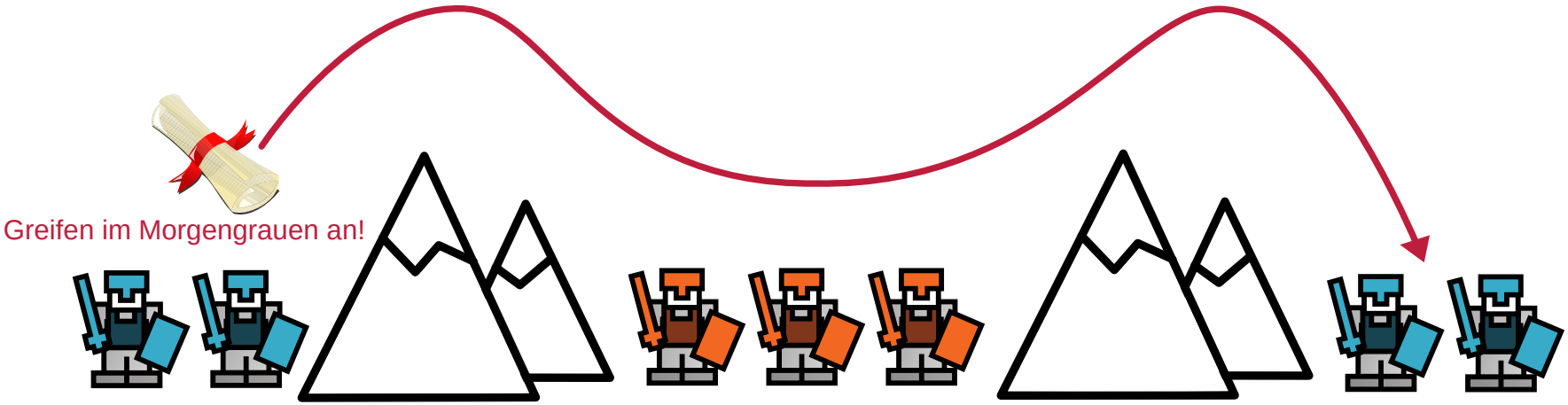
5 a) 2-Armeen-Problem – Beschreibung



Ausgangssituation

- 2 Armeen (Blau + Rot)
- Blaue Armee ist aufgeteilt (B1 und B2)
- blaue Truppen müssen gleichzeitig angreifen um zu gewinnen

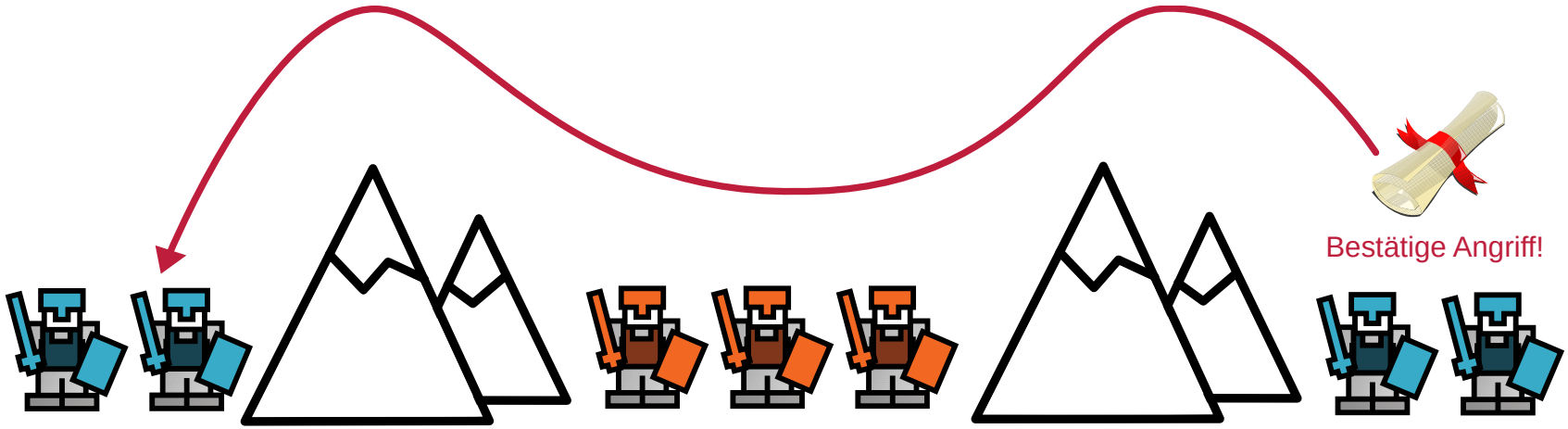
5 a) 2-Armeen-Problem – Beschreibung



Senden der Nachricht

- B1 schickt Nachricht: „Greifen im Morgengrauen an!“
- B1 benötigt eine Bestätigung, dass die Nachricht angekommen ist

5 a) 2-Armeen-Problem – Beschreibung



Problem der Bestätigung

- B2 bestätigt die Nachricht
- B2 kann nicht sicher sein, dass die Bestätigung ankommt
- B2 braucht Bestätigung für die Bestätigung (usw.)!

5 b) 2-Armeen-Problem – Lösung

b) Wie löst TCP das 2-Armeen-Problem?

Lösung

- Das 2-Armeen-Problem ist nicht lösbar!
- Abschwächung: Drei-Wege-Handshake

Aufgabe 6: Verbindungsmanagement

a) Aus welchem Grund ist ein Zwei-Wege-Handshake nicht ausreichend für einen Verbindungsaufbau? Zeigen Sie einen Fall, bei dem es zu Fehlern kommt.

b) Erklären Sie den Verbindungsaufbau bei TCP.

c) Welches Sicherheitsproblem kann bei einem Drei-Wege-Handshake auftreten?

d) Wie kann man eine Verbindung zuverlässig abbauen?

e) Welche Arten des Verbindungsabbaus gibt es?

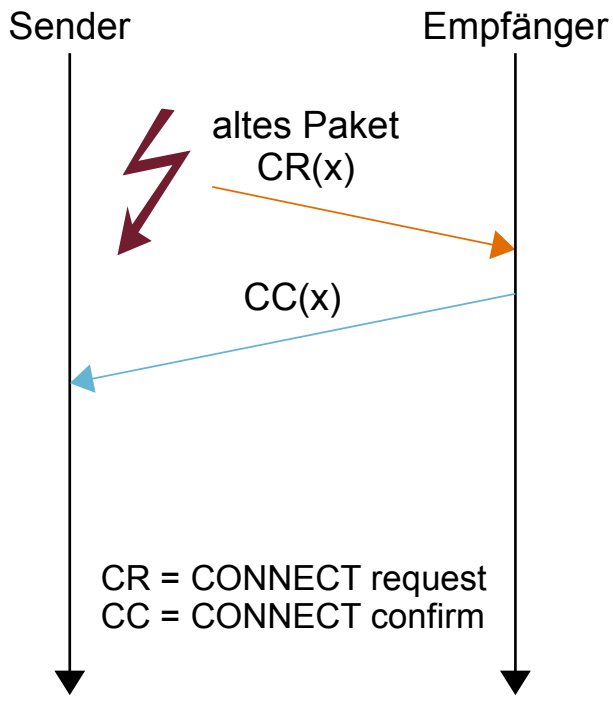
6 a) 2-Wege-Handshake

a) Aus welchem Grund ist ein Zwei-Wege-Handshake nicht ausreichend für einen Verbindungsaufbau? Zeigen Sie einen Fall, bei dem es zu Fehlern kommt.

Kernproblem

Beim Verbindungsaufbau können noch verzögerte Duplikate von Paketen einer früheren Verbindung existieren.

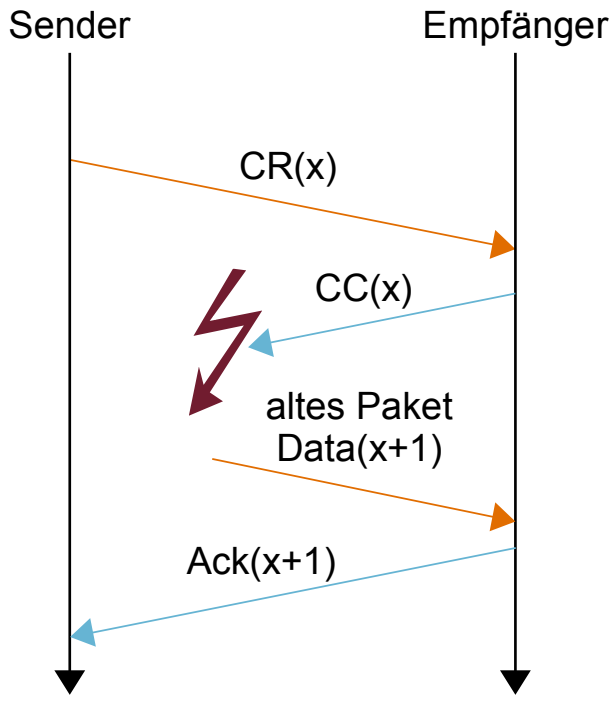
6 a) 2-Wege-Handshake



Beispiel 1: Falscher Verbindungsaufbau

Sender verwirft zwar CC(x), aber Empfänger erkennt Problem nicht. $\Rightarrow$ zusätzliches ACK Sender $\rightarrow$ Empfänger $\Rightarrow$ 3-Wege-Handshake

6 a) 2-Wege-Handshake



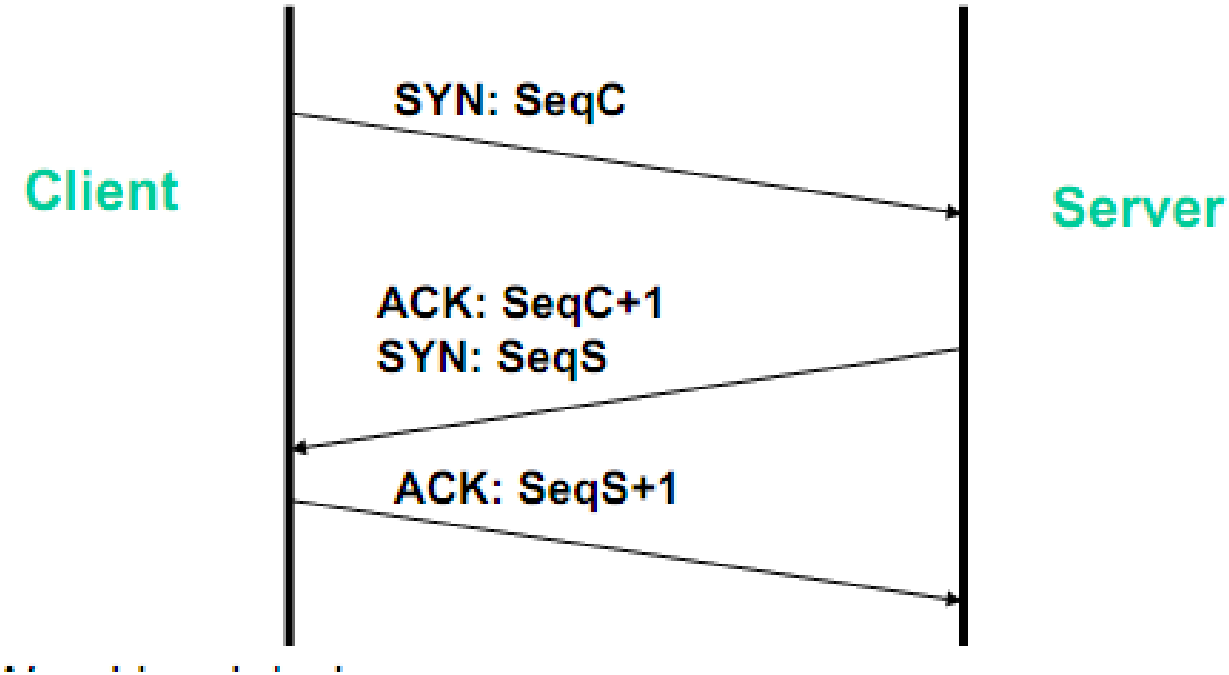
Beispiel 2: Deadlock (aber: künstliche Situation)

- Sender wartet auf CC(x)
- Empfänger wartet auf Data(x+2) oder Verbindungsabbau

6 b) TCP-Verbindungsaufbau

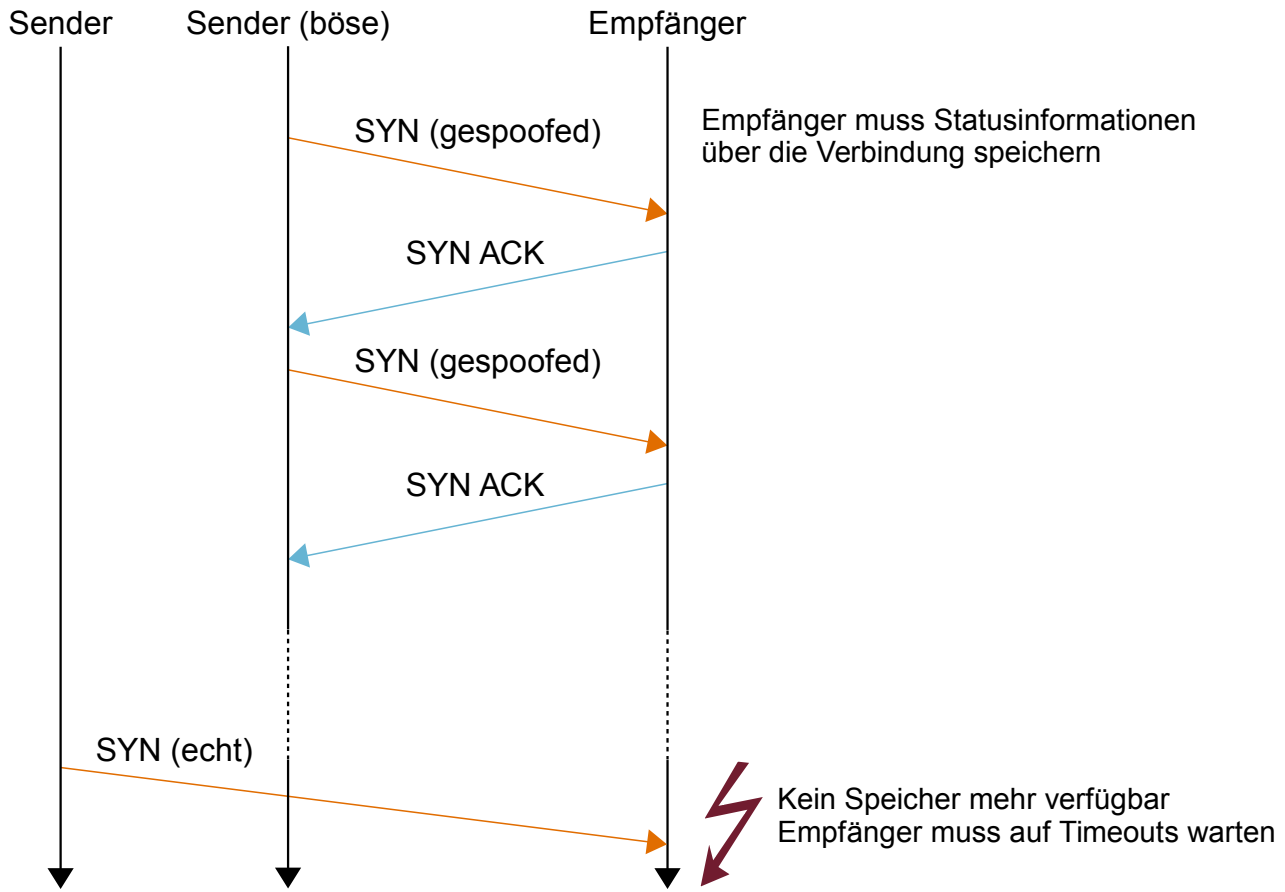
b) Erklären Sie den Verbindungsaufbau bei TCP.

- Drei-Wege-Handshake



6 c) Syn-Flooding

c) Welches Sicherheitsproblem kann bei einem Drei-Wege-Handshake auftreten?



6 c) Syn-Flooding

Syn-Flooding

- Im besten Fall: Denial-of-Service-Attacke, keine Verbindungen zum Empfänger mehr möglich
- Aber in der Realität: Entweder Crash des Empfängers wegen unsauberer Betriebssystemprogrammierung oder Administrator muss den Rechner rebooten

6 d) Verbindungsabbau

d) Wie kann man eine Verbindung zuverlässig abbauen?

Lösung

Gar nicht! $\Rightarrow$ siehe zwei-Armeen-Problem

Generelles Problem

Für beide Seiten gilt stets, dass unbekannt ist, ob die Gegenseite über jüngste eigene Erkenntnisse auch Kenntnis hat.

6 e) Arten des Verbindungsabbaus

e) Welche Arten des Verbindungsabbaus gibt es?

Lösung

asymmetrisch

- Nur eine von beiden Seiten beendet die gesamte Verbindung
- Initiator kann keine Daten mehr empfangen

symmetrisch

- Jede Sender-Empfänger-Beziehung wird einzeln geschlossen
- Beide Seiten müssen Disconnect-Request senden
- Initiator hört auf zu senden, kann aber weiter empfangen

Weitere Lehrveranstaltungen

`https://www.ibr.cs.tu-bs.de/cm/courses.html`

Sommersemester

- Mobilkommunikation
- Mensch-Computer-Interaktion
- Softwareentwicklungspraktikum
- Praktikum: WSN-Lab
- Praktikum Computernetze Administration

Wintersemester

- Computernetze 2
- Recent Topics in Computer Networking
- Praktikum: Computernetze
- Praktikum: V2X-Lab
- Teamprojekt: TPCM-Esys

Jedes Semester

- Advanced Networking
- Seminar (Bachelor & Master)

InformatiCup
Studierendenwettbewerb
der Gesellschaft für Informatik

InformatiCup

InformatiCup

Studierendenwettbewerb
der Gesellschaft für Informatik

GESELLSCHAFT FÜR INFORMATIK

Jobs und Abschlussarbeiten

Jobs und Abschlussarbeiten

Offene Jobs/Themen sind zu finden unter:
`https://www.ibr.cs.tu-bs.de/cm/theses.html`
Oder schreibt uns einfach eine Email

Aktuell offene Themen

- Various Topics on the use of eBPF in Wireless Networking
- Multiple Topics on Improving Wireless Network Resilience using AI
- Implementation and Evaluation of Multi-Connectivity on the Network Layer
- Detection and Classification of a Wireless Communication Disruption
- PTPSec: Implementing and analyzing PTP (IEEE 1588) in IoT networks

Fragestunde

Fragen für Fragestunde

Bitte vorher per Mail an mich: fschulze@ibr.cs.tu-bs.de
Deadline: Sonntag, 10. August 2025, 23:59 Uhr

Termin

Dienstag, 12. August 2025, 13:00 Uhr
IZ 160 und BBB-Raum der CN1-Übung

Klausur: 18. August 2025, 11:00Uhr

Eckdaten

- 90 Minuten, 90 Punkte ($\Rightarrow$ 1 Punkt pro Minute)
- Räume: Audimax, UP 3.007 (Bunker), ZI 24.2, ZI 24.3 (Grotrian)
  - Raumaufteilung wird wenige Tage vorher online bekannt gegeben
- Theorie (Vorlesung, teilweise Übung)
- Praktische Anwendung (Übung)
- Keine Hilfsmittel! *(siehe nächste Folie)*

Aufgabentypen

- Wissensfragen
- Algorithmen (z.B. Sliding Window, Djikstra, ...)
- Berechnungen (z.B. Bandbreite, Verzögerung, Nyquist, ...)

Klausur: 18. August 2025, 11:00Uhr

Es sind keine Hilfsmittel erlaubt!

Somit sind auch Taschenrechner, Mobiltelefone, Headsets, Smartwatches, Smartglasses oder sonstige (kommunikationsfähige) Hilfsmittel verboten. Ebenfalls sind Hilfsmittel auf Papier, wie z.B. Formelsammlungen oder Wörterbücher, sowie sonstige Medien verboten. In der Hand, auf dem Tisch oder anderweitig in Reichweite befindliche Hilfsmittel werden unmittelbar als Täuschungsversuch gewertet!

Klausur: 18. August 2025, 11:00Uhr

Erlaubt

- blaue oder schwarze dokumentenechte Stifte
- zusätzliches Schreibmaterial (z. B. Textmarker, Lineal, usw.)
- Getränke, Snacks, Taschentücher
- Medikamente
- Uhr (die nur die Uhrzeit anzeigen kann!)
- Bei Unsicherheiten bitte die Aufsicht fragen!

Weitere Regeln

- Keine roten Stifte, Bleistifte oder Korrekturflüssigkeiten!
- Kein eigenes Papier! Nur das bereitgestellte Papier verwenden!
- Antworten nur in Deutsch oder Englisch!

Sommerfest der Informatik:

heute, 18:00 Uhr

Fragestunde:

12. August 2025, 13:00 Uhr

Klausur:

18. August 2025, 11:00Uhr