Karlsruher Institut für Technologie

Übungen zu Einführung in Rechnernetze
4. Übung

Matthias Flittner, Sebastian Friebe, Tim Gerhard, Markus Jung
[flittner, friebe, tim.gerhard, m.jung]@kit.edu

Institut für Telematik, Prof. Zitterbart

Einführung in Rechnernetze – 4. Übungsblatt

1. TCP-Analyse mit Wireshark
2. TCP-Mechanismen und Verbindungsverwaltung
3. TCP-Arbeitsweise und Flusskontrolle
4. TCP-Staukontrolle
5. Vermittlungsformen und -prinzipien
6. IP
7. IP-Adresskonfiguration und Subnetze

Aufgabe 1: TCP-Analyse mit Wireshark

b) Können Sie die Anfragen Ihres Chat-Clients in der Aufzeichnung ausfindig machen? Wodurch können Sie den Datenstrom Ihres Clients von anderen unterscheiden?

- Transportprotokoll ist TCP
- Port des Servers ist 8000
- IP des Servers ist 141.3.71.4
  (und der Hostname i72tmdjango.tm.uni-karlsruhe.de)

Aufgabe 1: TCP-Analyse mit Wireshark

c) Versuchen Sie alle Datenströme welche nicht zu ihrem Chat-Client gehören herauszufiltern.

```
tcp.port == 8000 and ip.addr == 141.3.71.4
```

Aufgabe 1: TCP-Analyse mit Wireshark

d) Analysieren Sie die ausgetauschten TCP-Segmente:
Wo findet der Verbindungsaufbau und -abbau statt?
Welche Informationen werden ausgetauscht?

- Verbindungsaufbau: Erste drei Segmente
  - SYN:
    - Initiale Sequenznummer
    - Empfangsfenster = 8192 Byte
    - MSS = 1360 Byte
    - TCP SACK erlaubt
    - TCP Window Scale = x256
      (Die beiden letzteren sind Optionen aus RFC1323 für High-Performance-Networking und Pfade mit hohem BDP)
  - SYN+ACK:
    - Initiale Sequenznummer
    - Empfangsfenster = 29200 Byte
    - MSS = 1460 Byte
    - TCP SACK erlaubt
    - TCP Window Scale = x128
  - ACK:
    - Empfangsfenster = 66560 Byte
- Verbindungsabbau: Letzte vier Segmente, FIN, ACK, FIN+ACK, ACK

Aufgabe 1: TCP-Analyse mit Wireshark

d) Analysieren Sie die ausgetauschten TCP-Segmente:

Wie erfolgt die Übertragung der HTTP-Requests?
Können Sie Datensegmente und Quittungen zuordnen?

- HTTP GET:
  - Ein Datensegment, wird direkt mit ACK quittiert
- HTTP/1.0 200 OK:
  - Zwei Segmente mit gesetztem Push-Flag. Zuerst Statuscode, anschließend der Hauptteil der Antwort.
    FIN wird direkt mit der zweiten Dateneinheit gesendet.

Aufgabe 1: TCP-Analyse mit Wireshark

d) Analysieren Sie die ausgetauschten TCP-Segmente:

Wo gibt es Unterschiede gegenüber den in der Vorlesung besprochenen Abläufen und Mechanismen?

- Zusätzliche Optionen:
  - Window-Scaling
  - TCP SACK
- Server schließt Verbindung
- FIN direkt im letzten Datensegment

Einführung in Rechnernetze – 4. Übungsblatt

1. TCP-Analyse mit Wireshark
2. TCP-Mechanismen und Verbindungsverwaltung
3. TCP-Arbeitsweise und Flusskontrolle
4. TCP-Staukontrolle
5. Vermittlungsformen und -prinzipien
6. IP
7. IP-Adresskonfiguration und Subnetze

Aufgabe 2 (a)

- Ein TCP-Segment wird mit `seq=17,ack=92` empfangen.
  Es sind keine weiteren Flags gesetzt.

- Verändert sich das Flusskontrollfenster?

Pingo

- Pingo-Link für diese Übung:

→ http://pingo.upb.de/548806

Aufgabe 2 (a) – Pingo

- Ein TCP-Segment wird mit `seq=17,ack=92` empfangen.
  Es sind keine weiteren Flags gesetzt.

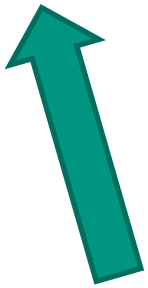
- Verändert sich das Flusskontrollfenster?

  - Ja
  - Nein

Aufgabe 2 (a)

- Ein TCP-Segment wird mit `seq=17,ack=92` empfangen.
  Es sind keine weiteren Flags gesetzt.

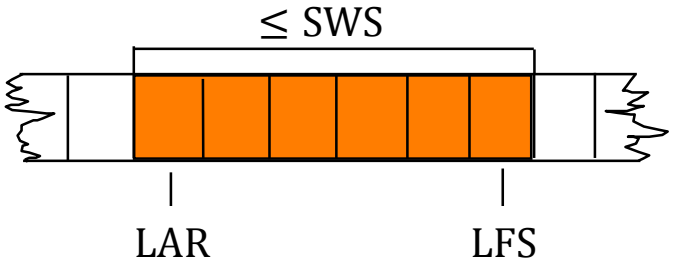
- Verändert sich das Flusskontrollfenster?



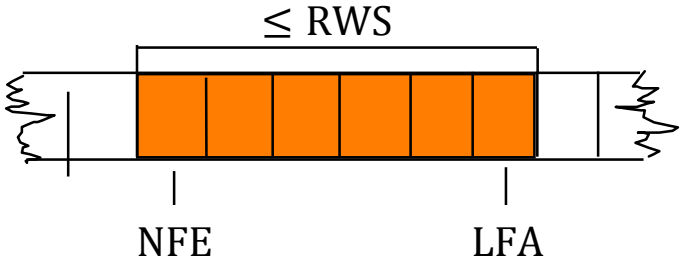
Welches?

Flusskontrolle mit Sliding Window

- Sender
  - SWS: Send Window Size (max. Anzahl ausstehender Pakete bzw. Bytes)
  - LAR: Last ACK Received (Sequenznummer des nächsten erwarteten Pakets bzw. Bytes)
  - LFS: Last Frame Sent (Sequenznummer des letzten gesendeten Pakets bzw. Bytes)
- Invariante
  - $\text{LFS} - \text{LAR} + 1 \leq \text{SWS}$



- Empfänger
  - RWS: Receiver Window S… (max. Anzahl nicht in Reihenf…e empfangener Pakete bzw. Bytes)
  - LFA: Last Frame Acceptable (Sequenznummer des letzten empfangbaren Pakets bzw. Bytes)
  - NFE: Next Frame Expected (Sequenznummer des nächsten in Reihenfolge erwarteten Pakets bzw. Bytes)
- Invariante
  - $\text{LFA} - \text{NFE} + 1 \leq \text{RWS}$

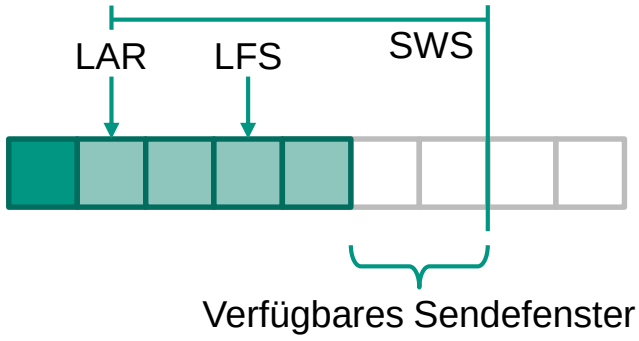


Aufgabe 2 (a)

- Ein TCP-Segment wird mit `seq=17,ack=92` empfangen.
  Es sind keine weiteren Flags gesetzt.

- Verändert sich das Flusskontrollfenster?

  - Senderseite
    - Position (LAR) und Größe (SWS) bleiben unverändert
      - Verschiebung nur durch eingehende Quittungen
      - Größenveränderung nur durch eingehende Segmente
    - „Freier Bereich“ im Sequenznummernraum *wurde* kleiner
      - Keine Veränderung bei Empfang des Segments auf Senderseite

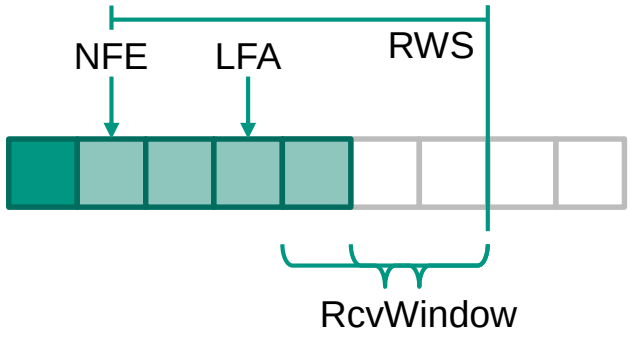


Aufgabe 2 (a)

- Ein TCP-Segment wird mit `seq=17,ack=92` empfangen.
  Es sind keine weiteren Flags gesetzt.

- Verändert sich das Flusskontrollfenster?

  - Empfängerseite
    - Position (NFE) und Größe (RWS/RcvBuffer) bleiben unverändert
      - Position verändert sich nur wenn Empfänger Daten aus dem Empfangspuffer entnimmt
    - Empfangsfenster (RcvWindow) wird kleiner wenn neue Daten im TCP-Segment
      - Änderung wird dem Sender mit Quittung mitgeteilt
      - … wurde vom Sender aber schon berücksichtigt (Kredit …)



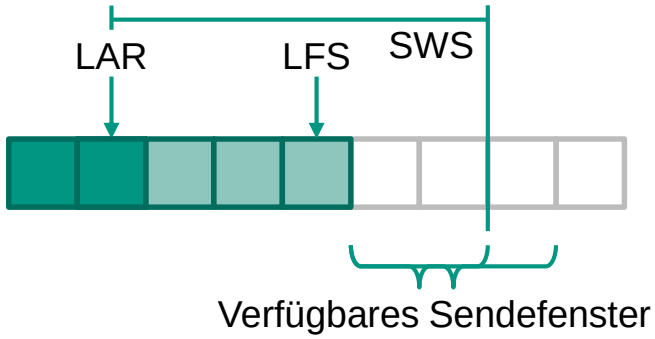
Aufgabe 2 (a)

- Ein TCP-Segment wird mit `seq=17,ack=92` empfangen.
  Es sind keine weiteren Flags gesetzt.

- Verändert sich das Flusskontrollfenster?

Was fehlt?

- Flusskontrollfenster des Empfängers in Senderichtung!
  - Quittung kann Position (LAR) verschieben
    - Verfügbares Sendefenster wächst
  - Könnte auch Fenstergröße verändern (SWS)



Aufgabe 2 (b)

- Wie groß muss Ihr Flusskontrollfenster sein, damit Sie ein Netz mit der Datenrate 1000 MBit/s voll auslasten können?
  - Die insgesamt zurückgelegte Strecke zwischen Sender und Empfänger betrage 800 km.
  - Die Ausbreitungsgeschwindigkeit sei 200.000 km/s.
  - Verzögerungen durch Zwischensysteme werden vernachlässigt.

Aufgabe 2 (b) – Pingo

- Wie groß muss Ihr Flusskontrollfenster sein, damit Sie das Netz voll auslasten können?

Datenrate: 1000 MBit/s

Strecke Sender-Empfänger: 800 km

Ausbreitungsgeschwindigkeit: 200.000 km/s.

- 100,54 KB
- 500 KB
- 1000 KB
- 1001,46 KB
- 1001,514 MB
- 1501,46 KB
- 2000 KB

Aufgabe 2 (b) – Pingo

- Wie groß muss Ihr Flusskontrollfenster sein, damit Sie das Netz voll auslasten können?

Datenrate: 1000 MBit/s

Strecke Sender-Empfänger: 800 km

Ausbreitungsgeschwindigkeit: 200.000 km/s.

- 100,54 KB
- 500 KB
- 1000 KB
- 1001,46 KB
- 1001,514 MB
- 1501,46 KB
- 2000 KB

Aufgabe 2 (b)

- Wie groß muss Ihr Flusskontrollfenster sein, damit Sie das Netz voll auslasten können?

- Datenrate r: 1000 MBit/s
- Strecke Sender-Empfänger l: 800 km
- Ausbreitungsgeschwindigkeit c: 200.000 km/s.

- Ansatz: Sender muss so lange Senden können bis …
  - … Segmente beim Empfänger ankommen
  - … und dessen Quittungen zurück beim Sender sind

- Ausbreitungsverzögerung:
  - $t_a = l/c = 800\ km/200.000\,\frac{km}{s} = 4 \cdot 10^{-3} s = 4ms$
- Bandbreiten-Verzögerungs-Produkt (BDP):
  - $BDP = t_a \cdot r = 4ms \cdot 1000\,\frac{MBit}{s} = 0{,}5MB$
- Hin- und Rückrichtung: 1 MB

Aufgabe 2 (b)

- Wie groß muss Ihr Flusskontrollfenster sein, damit Sie das Netz voll auslasten können?

- Datenrate r: 1000 MBit/s
- Strecke Sender-Empfänger l: 800 km
- Ausbreitungsgeschwindigkeit c: 200.000 km/s.

- Ansatz: Sender muss so lange Senden können bis …
  - … Segmente beim Empfänger ankommen …
  - … und dessen Quittungen zurück beim Sender sind

Problem:

Ansatz vernachlässigt Sendezeit für erstes Segment und erste Quittung

- Vollständige Modellierung:
  2 BDP + 1 MSS (1460 Byte) + 1 ACK (54 Byte)
  - Aber: Schicht 1+2 bleiben unberücksichtigt.
  - Echte Dateneinheiten sind „länger“ → 2 BDP ist eine gute Näherung

Aufgabe 2 (c)

- Warum sieht TCP ein Fluss- und ein Staukontrollfenster vor? Genügt nicht eines von beiden oder könnte man beide Fenster nicht zusammen fassen?

Aufgabe 2 (c)

- Warum sieht TCP ein Fluss- und ein Staukontrollfenster vor? Genügt nicht eines von beiden oder könnte man beide Fenster nicht zusammen fassen?

- Flusskontrolle
  - Empfänger kann von Sender überlastet werden
    - Daten können nicht empfangen werden, da Puffer nicht ausreichend

    → Datenverluste

Vorlesung

- Staukontrolle
  - Vermeidung von Überlastsituationen im Netz
  - Stau im Netz … Puffer im Router füllen sich
    - Pufferüberlauf → Verwerfen von Paketen
    - … Sendewiederholungen durch den TCP-Sender

    → Verstärkung der Stausituation!

Vorlesung

Aufgabe 2 (c)

- Warum sieht TCP ein Fluss- und ein Staukontrollfenster vor? Genügt nicht eines von beiden oder könnte man beide Fenster nicht zusammen fassen?
  - Fluss- und Staukontrolle erfüllen verschiedene Aufgaben
  - Ohne beide Funktionen können Empfänger oder Netz überlastet werden
  - Beide Fenster repräsentieren voneinander unabhängige Sachverhalte
    - Eine hohe mögliche Datenrate im Netz bedeutet nicht, dass der Empfänger diese auch verarbeiten kann!

Aufgabe 2 (d) – Analog-Pingo

- Befindet sich im TCP-Kopf ein Flag zum Anzeigen von Fragmentierung?

- Ja
- Nein

IPv4 fragmentiert und verfügt über entsprechende Flags, TCP nicht

Aufgabe 2 (e) – Analog-Pingo

- Warum etabliert TCP eine Verbindung? Könnte die Funktionalität von auch ohne Verbindungskontext realisiert werden?

1. Daten sind korrekt und vollständig:
   - Erfordert Kenntnis von Beginn und Ende des Datenstroms und der Zugehörigkeit aller Segmente zu diesem
2. Es werden keine Phantom-Pakete ausgeliefert:
   - Erkennen der Nichtzugehörigkeit, analog zu Punkt 1.

- Daten werden in der richtigen Reihenfolge ausgeliefert:
  - Erfordert Zähler für Sequenznummern
- Es werden keine Duplikate ausgeliefert:
  - Erfordert Überwachung von Sequenznummern

Aufgabe 2 (f)

- Welche Informationen müssen Endsysteme zum Zustand einer TCP-Verbindung speichern? Wozu dienen diese?
  - Verbindungszustand (Zustandsautomat!)
  - Letzte gesendete (unquittierte Sequenznummer)
  - Letzte quittierte Sequenznummer
  - Nächste erwartete Sequenznummer
  - Größe des Empfangsfensters (der Gegenseite)
  - Staukontrollparameter
  - …

Aufgabe 2 (g)

- Wieso ist bei TCP ein 3-Wege-Handshake erforderlich? Könnte der Verbindungskontext nicht einfach implizit mit Beginn der Datenübertragung durch den Sender etabliert werden?

  - Ohne 3-Wege-Handshake können Fehlersituationen auftreten, in denen die Verbindung nur halbseitig existiert. Erst die Rückbestätigung im dritten Schritt schließt solche Fehler aus.

  - Nachteile durch implizite Etablierung
    - Erkennung fehlender Segmente zu Beginn nur mit Mehraufwand möglich
    - Keine Aushandlung von Erweiterungen/Parametern zu Beginn (Flusskontrolle!)
    - Zustand der Verbindung bis zur ersten Quittung oder Timeout unklar
      - Segmente werden auf „gut Glück“ versendet

Einführung in Rechnernetze – 4. Übungsblatt

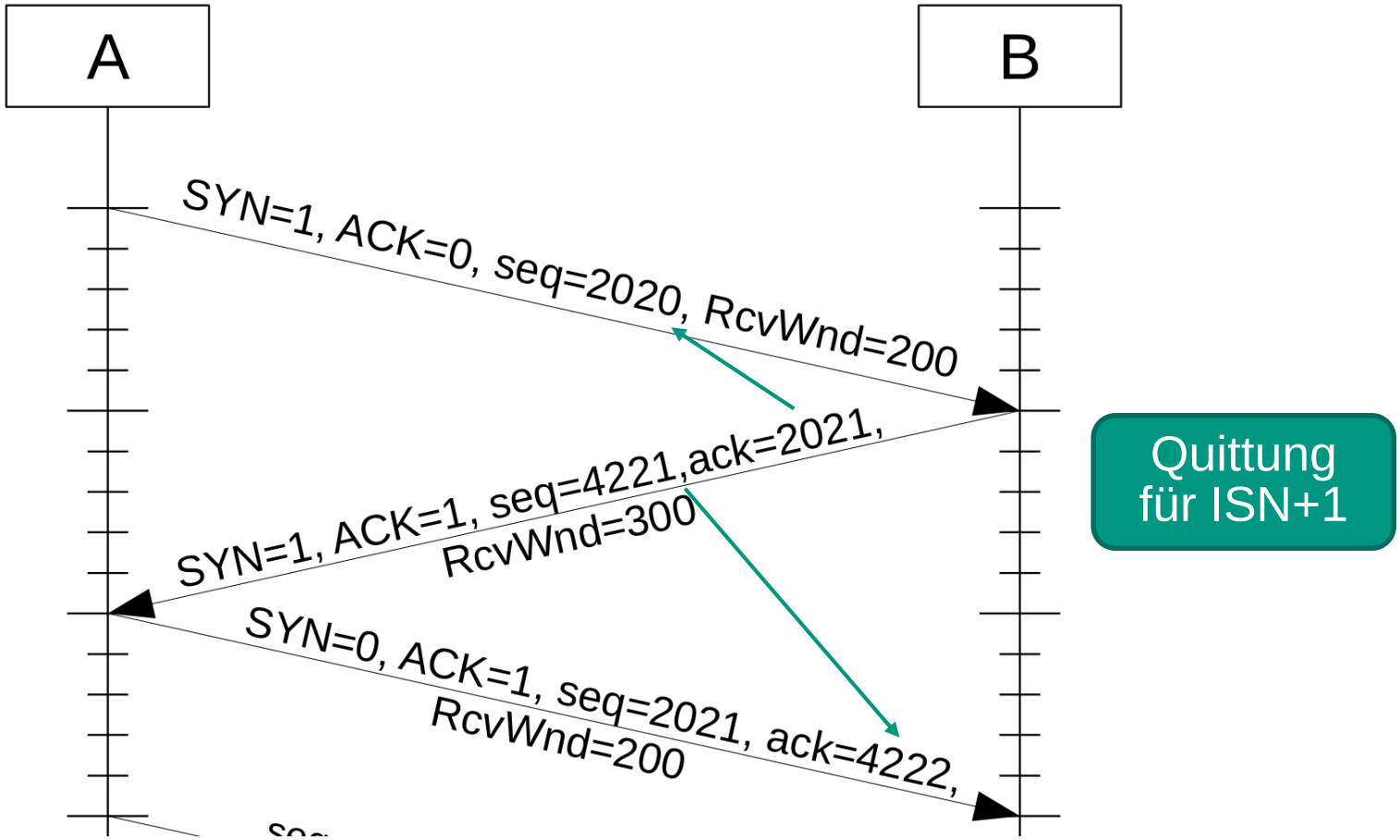
1. TCP-Analyse mit Wireshark
2. TCP-Mechanismen und Verbindungsverwaltung
3. TCP-Arbeitsweise und Flusskontrolle
4. TCP-Staukontrolle
5. Vermittlungsformen und -prinzipien
6. IP
7. IP-Adresskonfiguration und Subnetze

Aufgabe 3

- Host A will über eine TCP-Verbindung 700 Byte an Host B übertragen. Host B antwortet im Anschluss mit 350 Byte.
  - Host A arbeitet mit einem Empfangsfenster von 200 Byte, Host B mit 300 Byte.
  - Die MSS betrage 100 Byte.
  - Ein Segment kommt eine Zeiteinheit nach dem es abgesendet wurde beim Empfänger an.
  - Innerhalb einer Zeiteinheit können bis zu fünf Segmente versendet werden.
  - Der Timer für Quittungen laufe nach je drei Zeiteinheiten ab.
  - Berücksichtigen Sie, dass TCP Go-Back-N nutzt.
  - Es kommt keine Staukontrolle zum Einsatz.

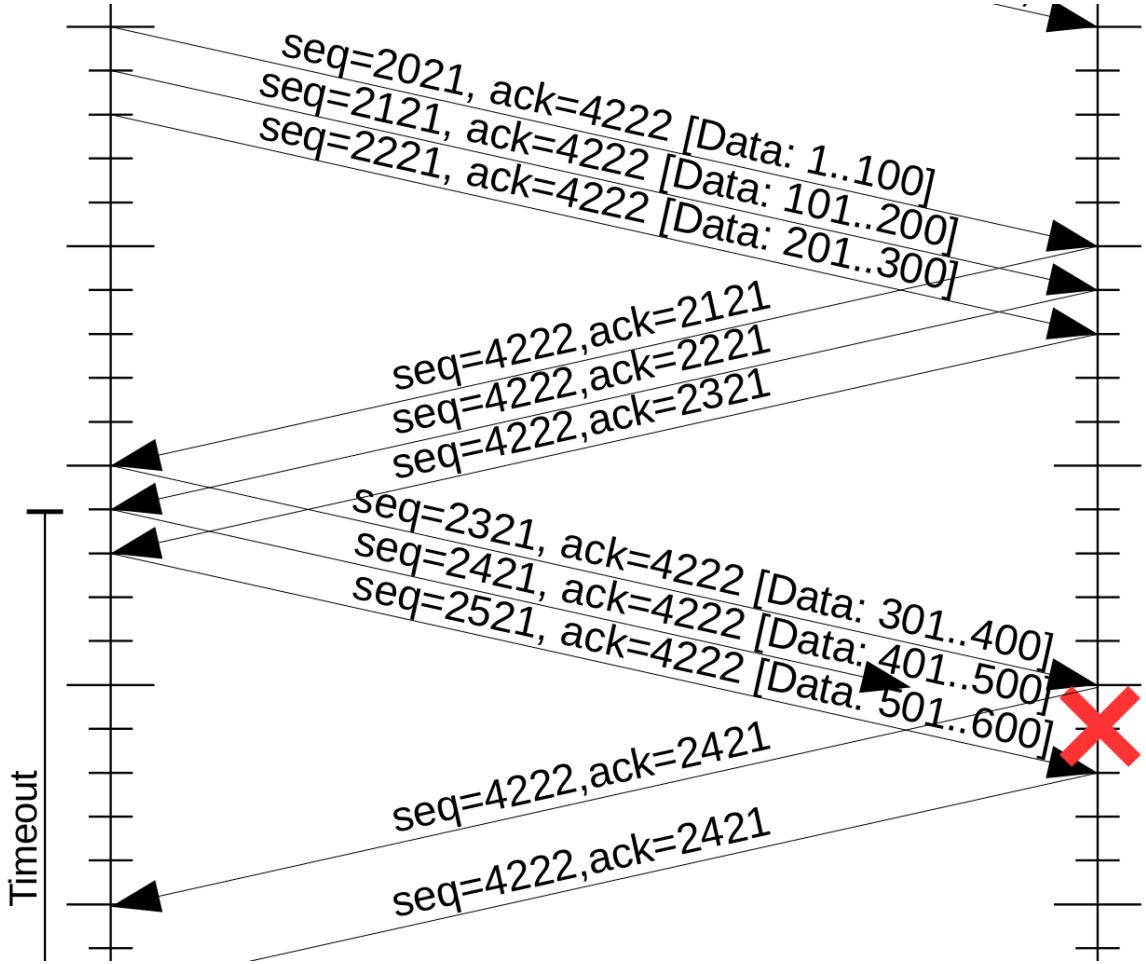
Aufgabe 3 (a)

- Zeichnen Sie den Verbindungsaufbau in das nachfolgende Weg-Zeit-Diagramm ein. Welche Informationen werden übertragen?



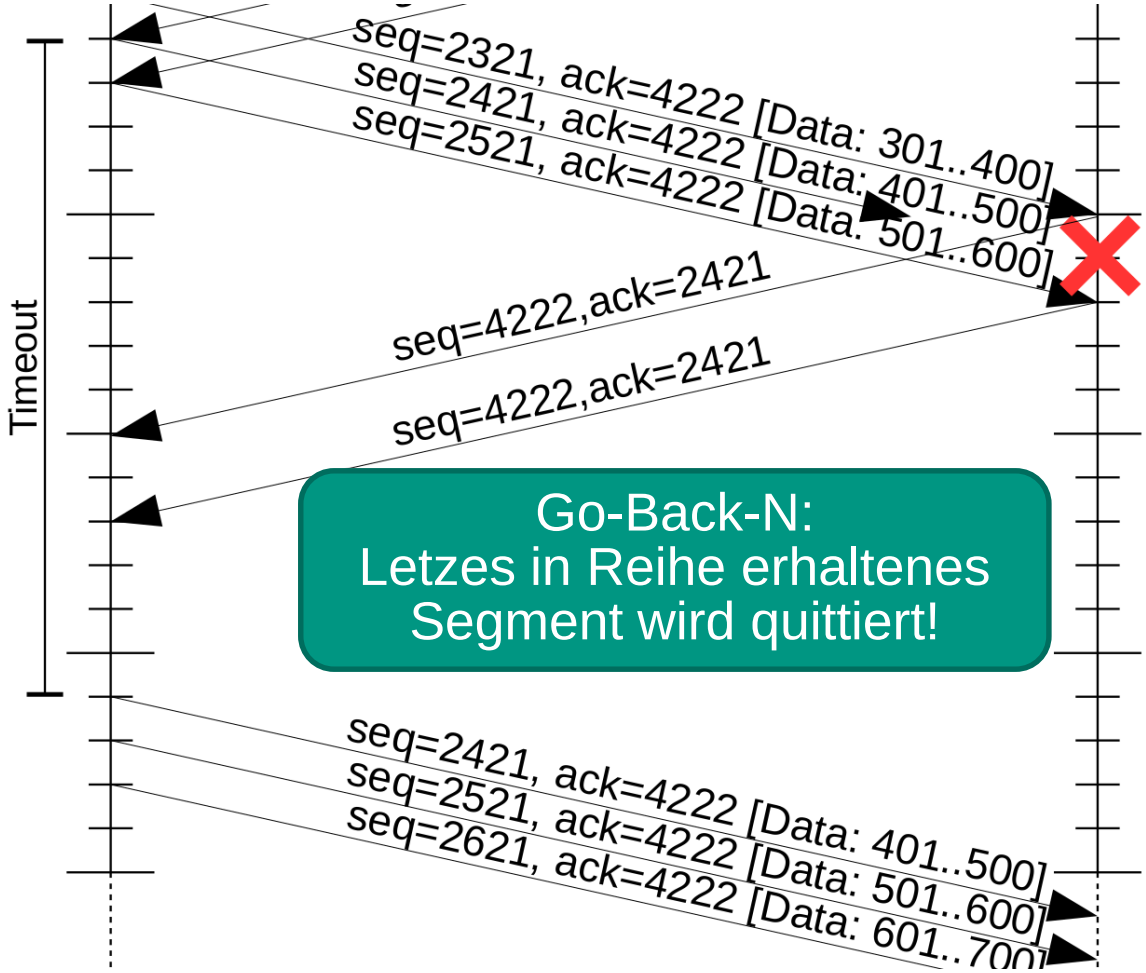
Aufgabe 3 (b)

- Bei der folgenden Übertragung gehe das fünfte Segment von Host A verloren. Vervollständigen Sie das Weg-Zeit-Diagramm.



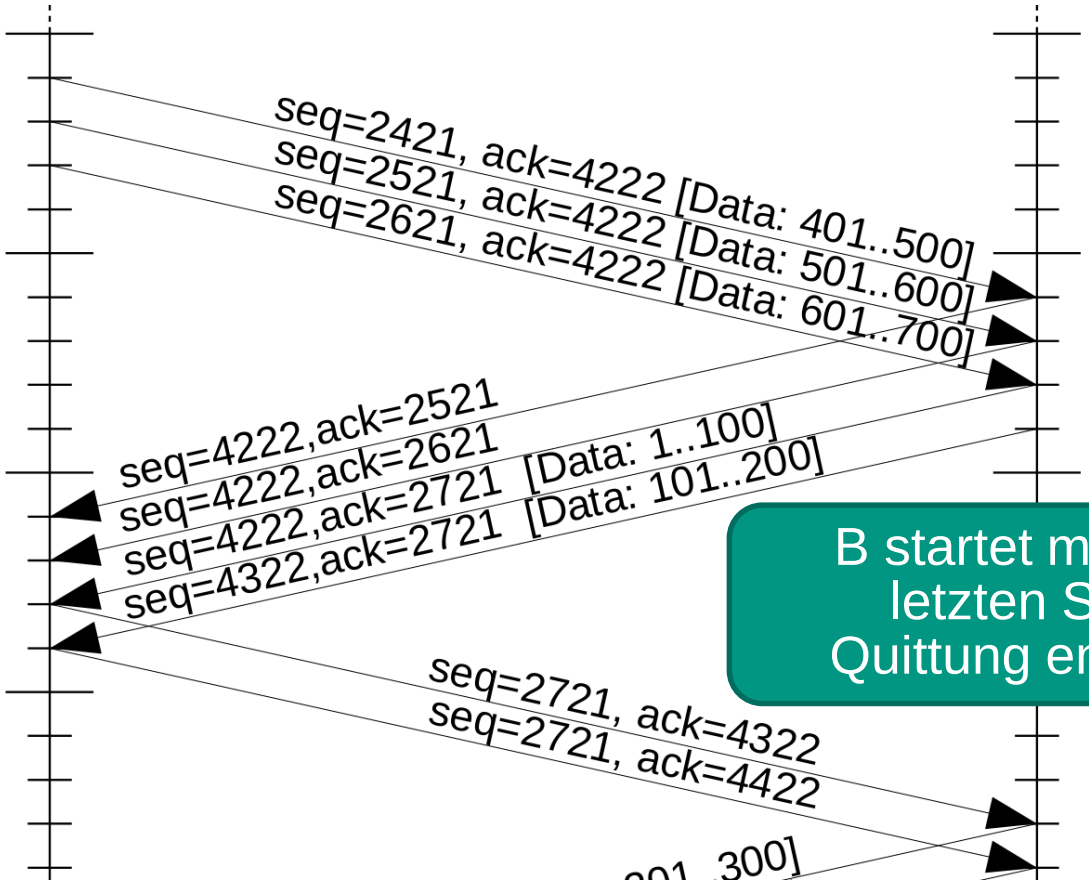
Aufgabe 3 (b)

- Bei der folgenden Übertragung gehe das fünfte Segment von Host A verloren. Vervollständigen Sie das Weg-Zeit-Diagramm.



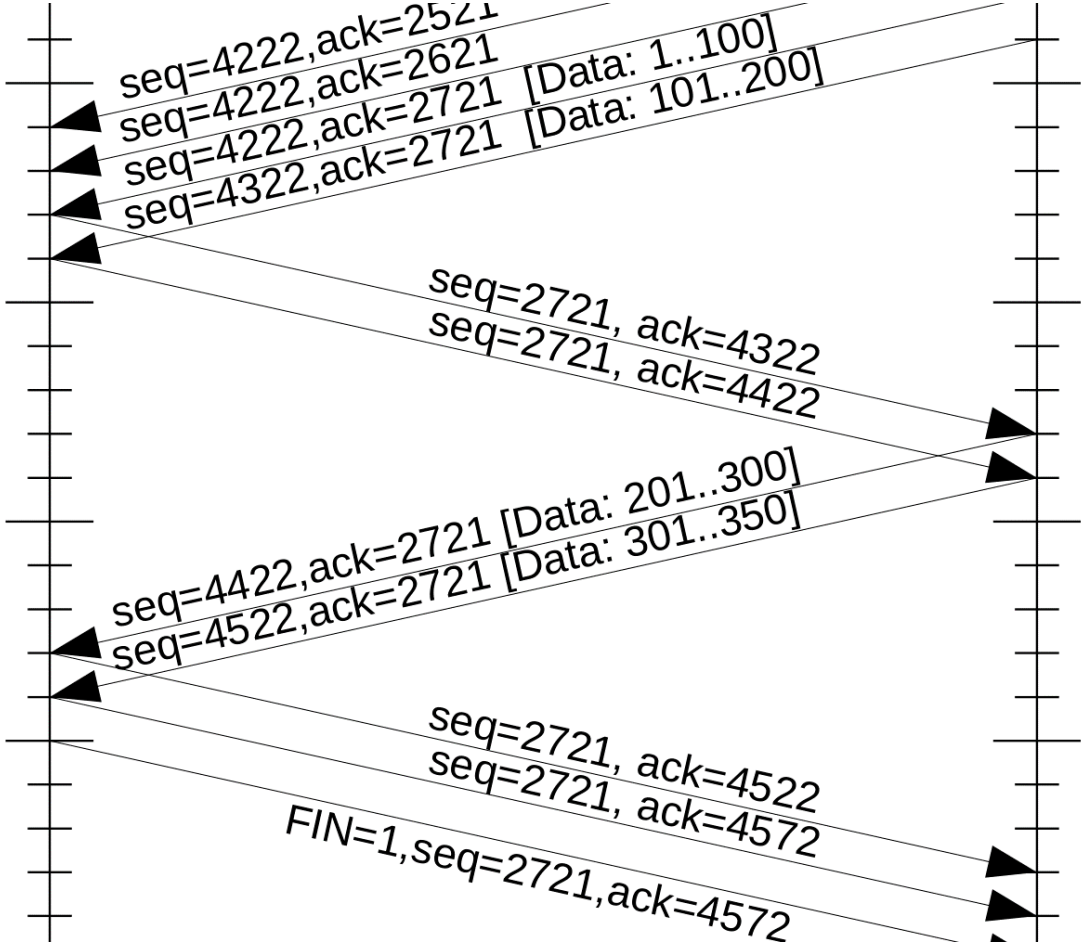
Aufgabe 3 (b)

- Bei der folgenden Übertragung gehe das fünfte Segment von Host A verloren. Vervollständigen Sie das Weg-Zeit-Diagramm.



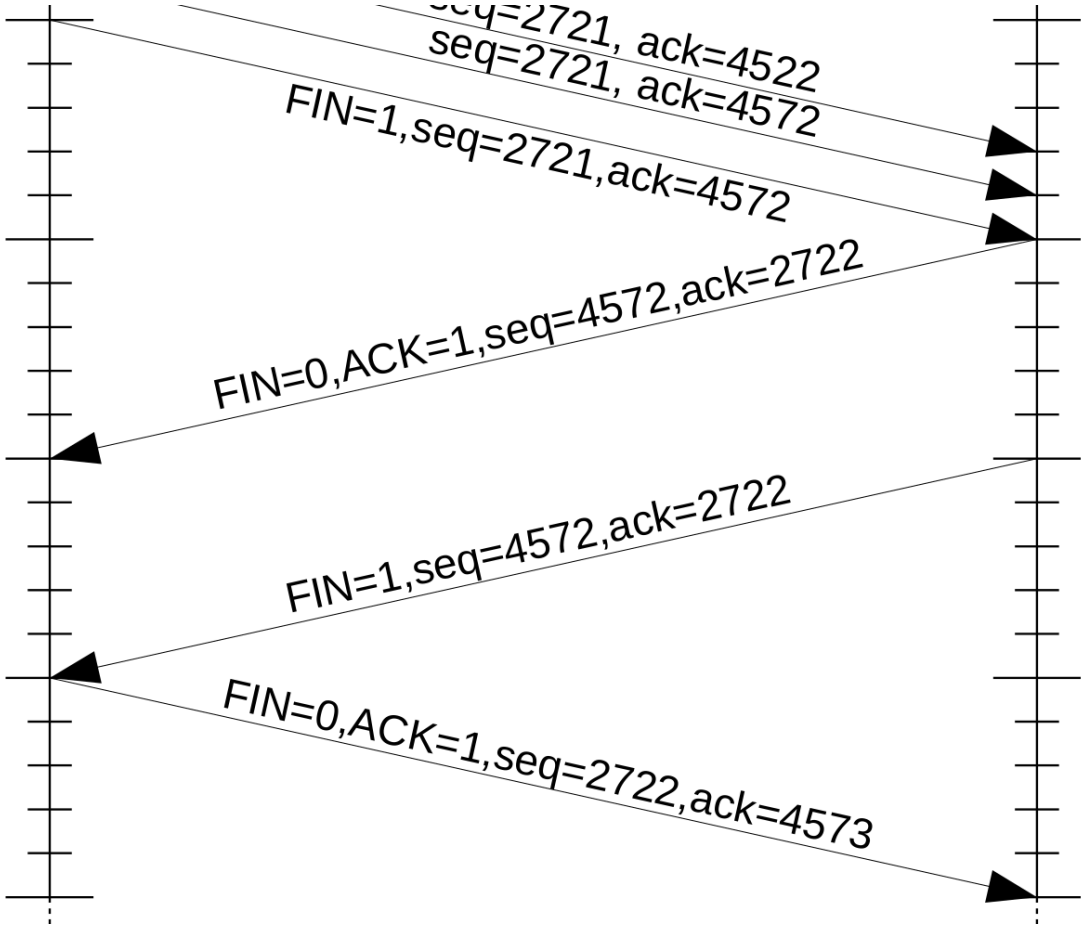
Aufgabe 3 (b)

- Bei der folgenden Übertragung gehe das fünfte Segment von Host A verloren. Vervollständigen Sie das Weg-Zeit-Diagramm.



Aufgabe 3 (c)

- Host A initiiert einen Verbindungsabbau.
  Ergänzen Sie den Ablauf im Weg-Zeit-Diagramm



Einführung in Rechnernetze – 4. Übungsblatt

1. TCP-Analyse mit Wireshark
2. TCP-Mechanismen und Verbindungsverwaltung
3. TCP-Arbeitsweise und Flusskontrolle
4. TCP-Staukontrolle
5. Vermittlungsformen und -prinzipien
6. IP
7. IP-Adresskonfiguration und Subnetze

Aufgabe 4

- Host A will über eine TCP-Verbindung 3500 Byte an Host B übertragen. Hierbei kommt das Staukontrollverfahren aus der Vorlesung, aber keine Flusskontrolle, zum Einsatz.
  - Die MSS betrage 100 Byte.
  - Ein Segment kommt eine Zeiteinheit nach dem es abgesendet wurde beim Empfänger an.
  - Innerhalb einer Zeiteinheit können bis zu zehn Segmente versendet werden.
  - Werden acht oder mehr Segmente innerhalb einer Zeiteinheit versendet, kommt es zu einer Stausituation.
  - Nehmen Sie vereinfachend an, dass dann alle in dieser Zeiteinheit gesendeten Segmente den Empfänger nicht erreichen.
  - Der Timer für Quittungen laufe nach je drei Zeiteinheiten ab.

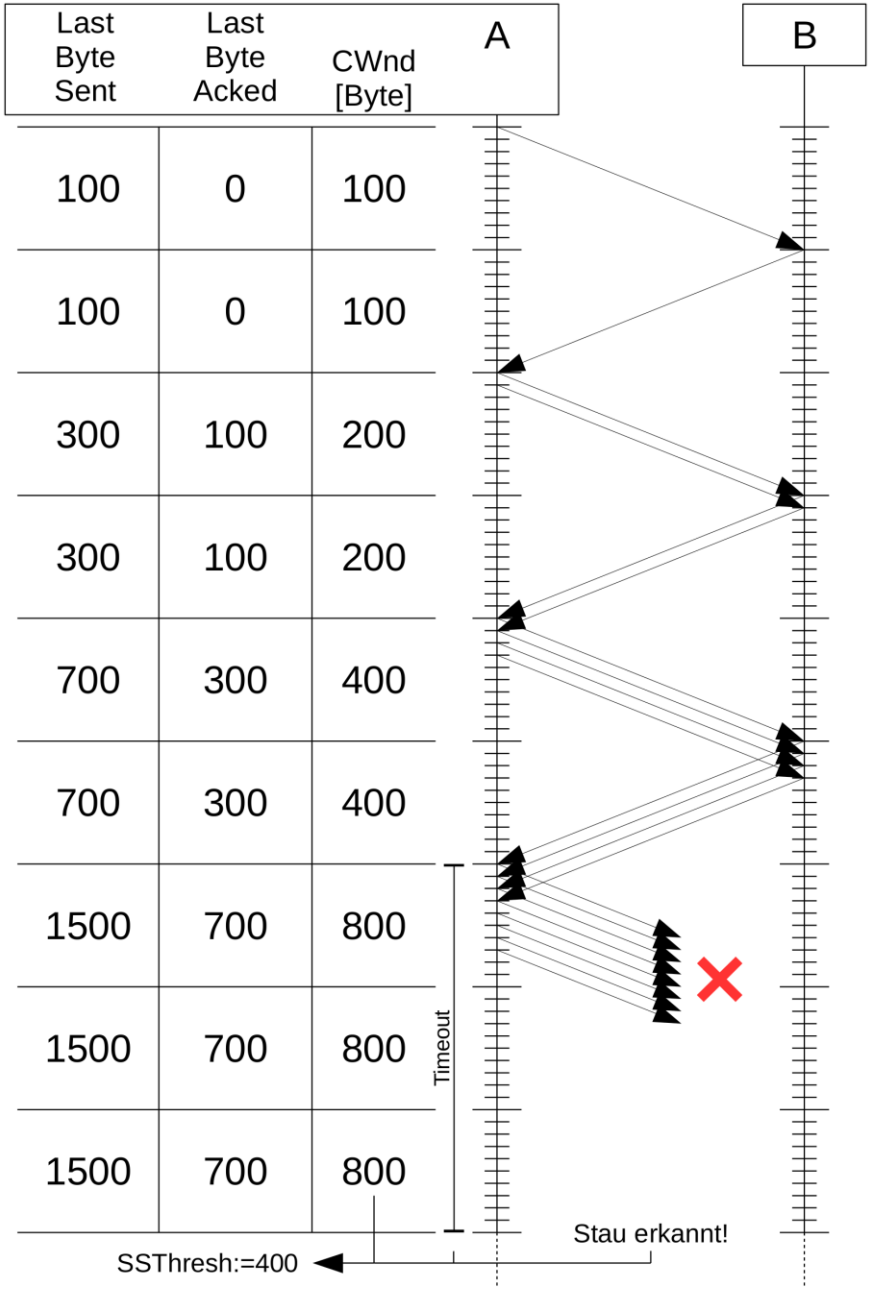
Aufgabe 4

- Host A will über eine TCP-Verbindung 3500 Byte an Host B übertragen. Hierbei kommt das Staukontrollverfahren aus der Vorlesung, aber keine Flusskontrolle, zum Einsatz.

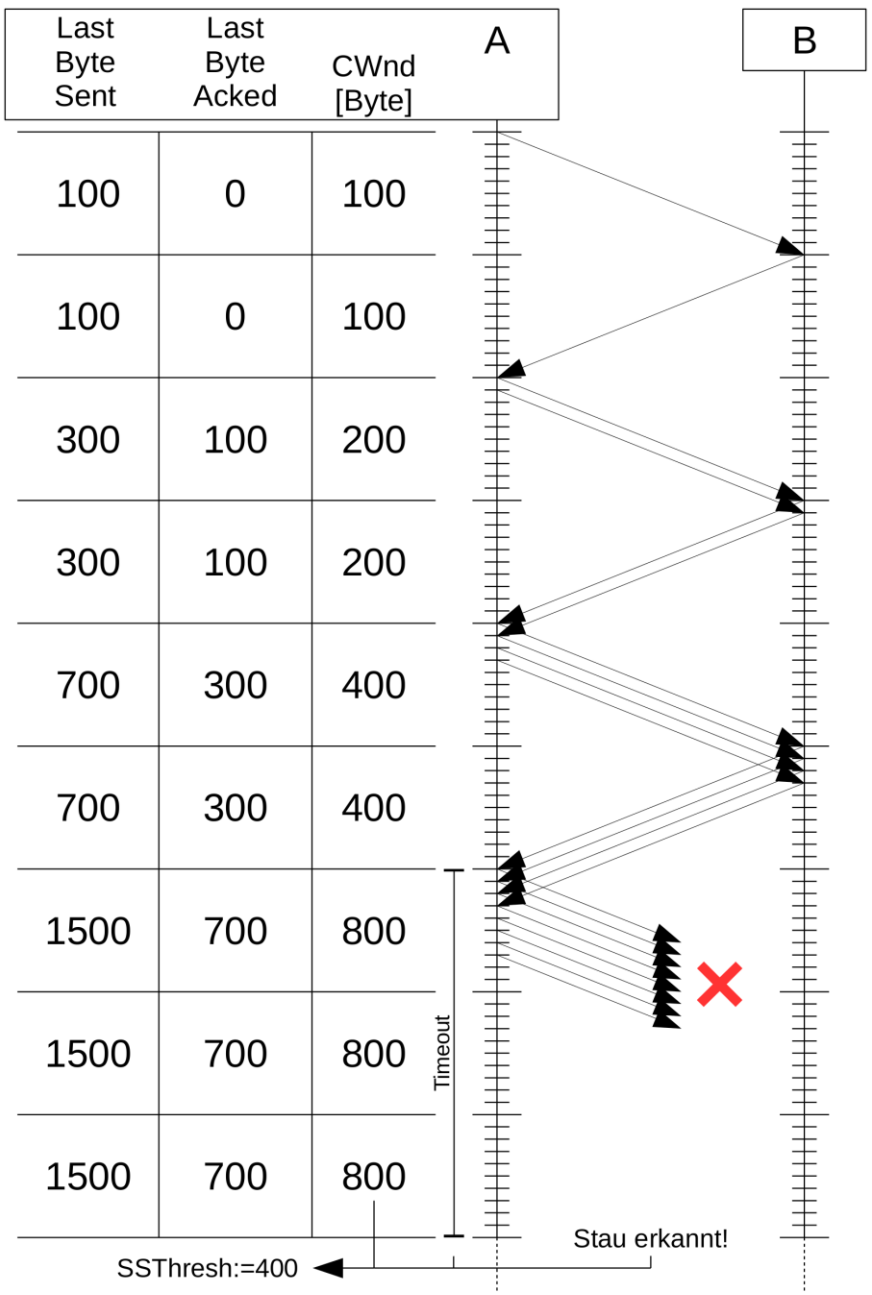
- Der Verlauf der Übertragung soll im nachfolgenden Weg-Zeit-Diagramm dargestellt werden.
  a) Markieren Sie, an welcher Stelle eine Stausituation erkannt wird.
  b) Zeichnen Sie ein, wo sich die Staukontrolle in der Slow-Start und Congestion-Avoidance-Phase befindet.
  c) Tragen Sie in jedem Zeitschritt die jeweils zuletzt gültigen Werte ein für: Größe des Staukontrollfensters (CWnd), jeweils größte gesendete Sequenznummer (LastByteSent) und größte quittierte Sequenznummer (LastByteAcked).

Aufgabe 4

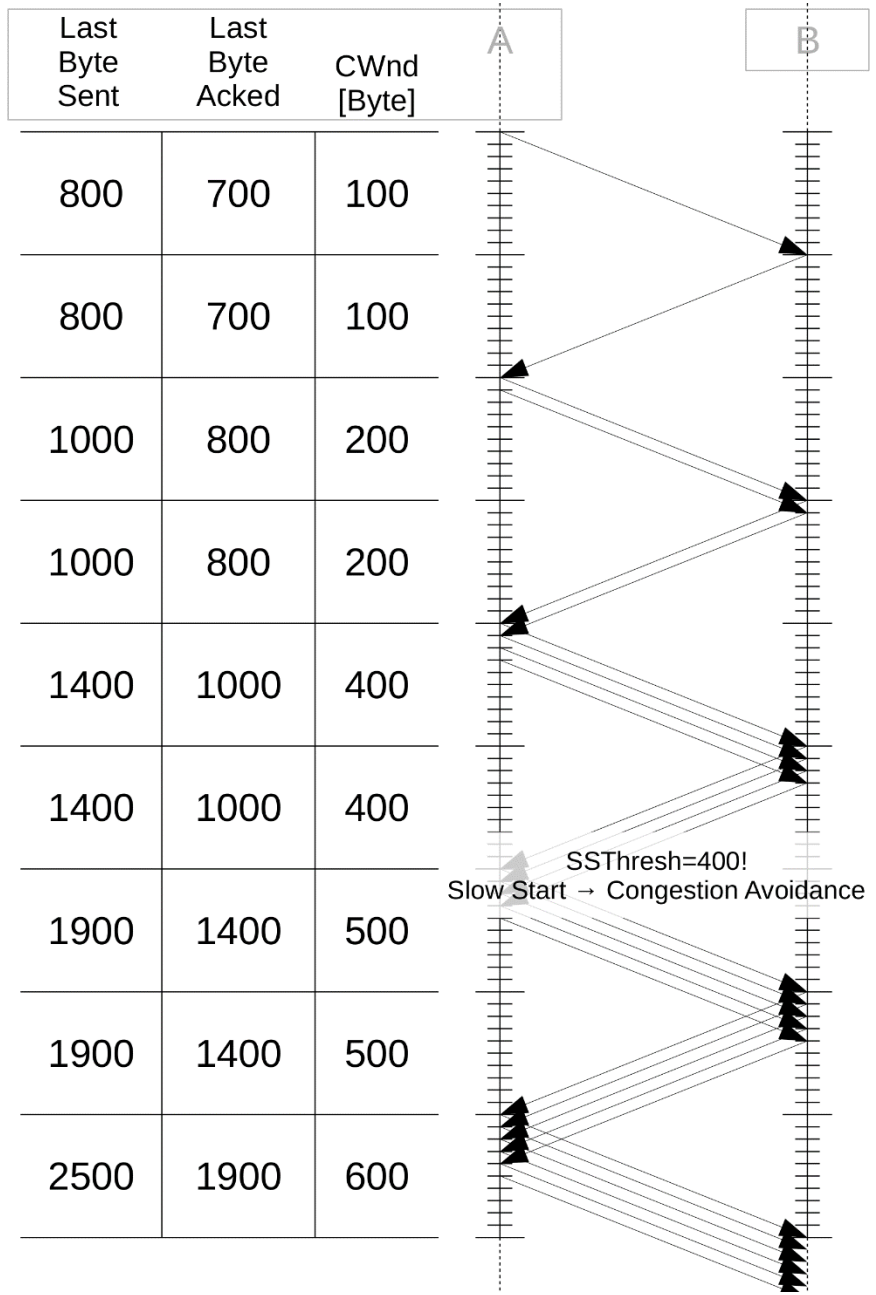
Last Byte Sent	Last Byte Acked	CWnd [Byte]
100	0	100
100	0	100
300	100	200
300	100	200
700	300	400
700	300	400
1500	700	800
1500	700	800
1500	700	800



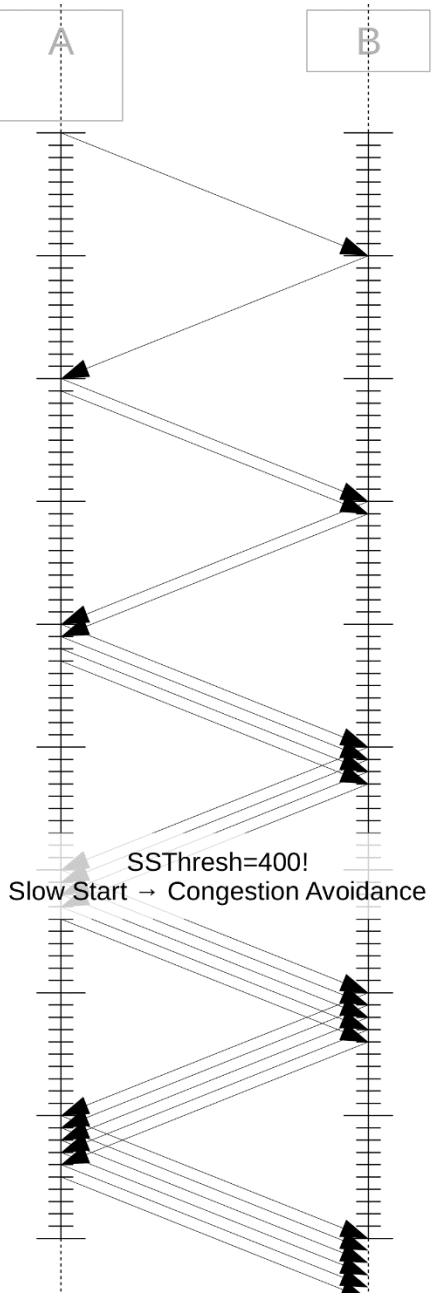
Last Byte Sent	Last Byte Acked	CWnd [Byte]
100	0	100
100	0	100
300	100	200
300	100	200
700	300	400
700	300	400
1500	700	800
1500	700	800
1500	700	800



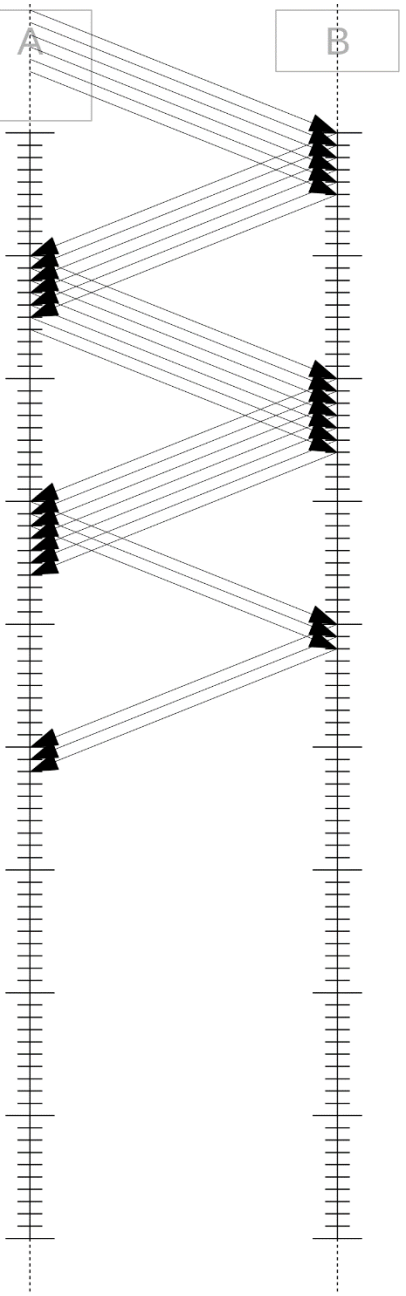
Last Byte Sent	Last Byte Acked	CWnd [Byte]
800	700	100
800	700	100
1000	800	200
1000	800	200
1400	1000	400
1400	1000	400
1900	1400	500
1900	1400	500
2500	1900	600



Last Byte Sent	Last Byte Acked	CWnd [Byte]
800	700	100
800	700	100
1000	800	200
1000	800	200
1400	1000	400
1400	1000	400
1900	1400	500
1900	1400	500
2500	1900	600



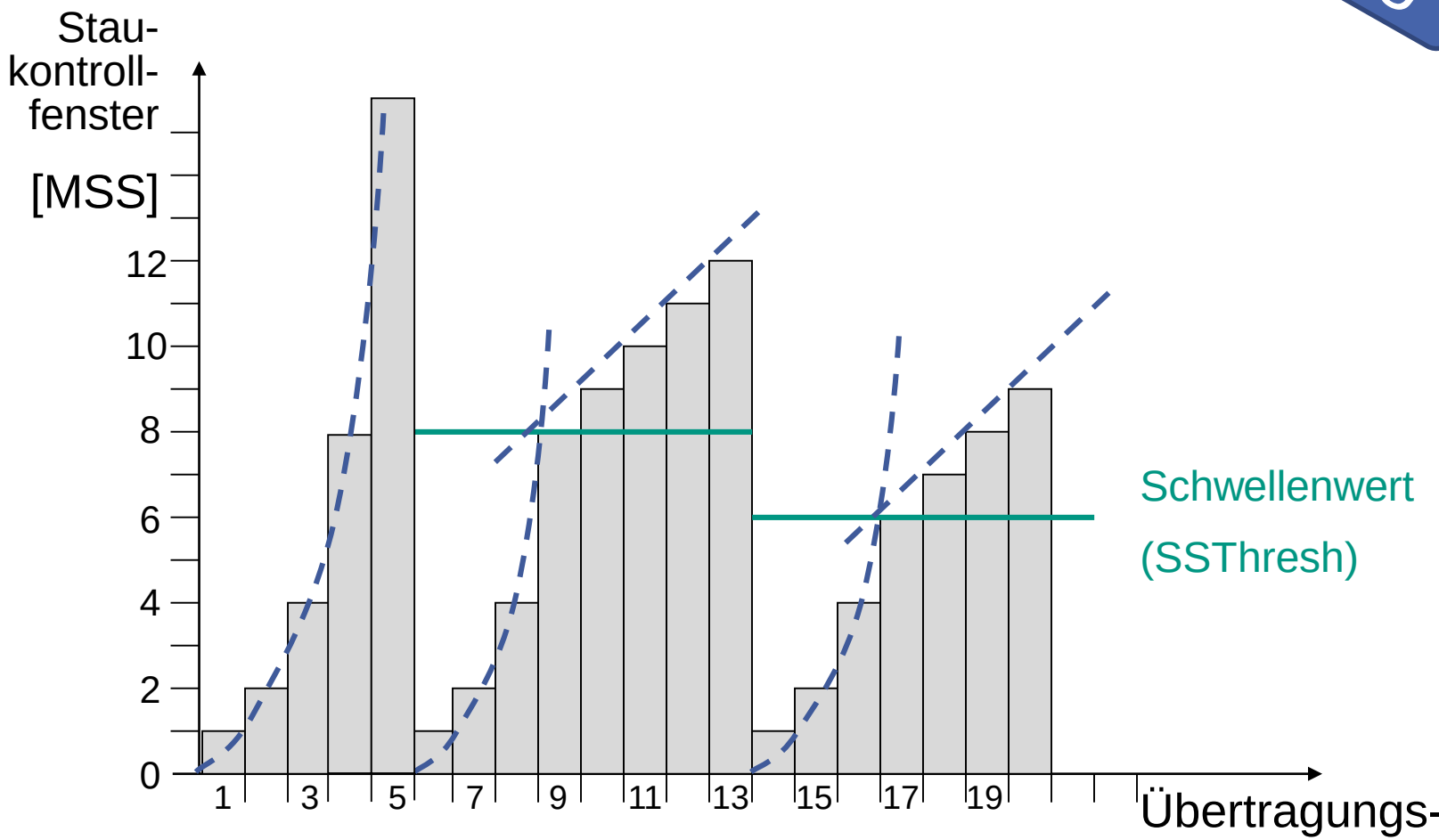
Last Byte Sent	Last Byte Acked	CWnd [Byte]
2500	1900	600
3200	2500	700
3200	2500	700
3500	3200	800
3500	3200	800
3500	3500	838



Entwicklung des Staukontrollfensters

Vorlesung

- Beispiel



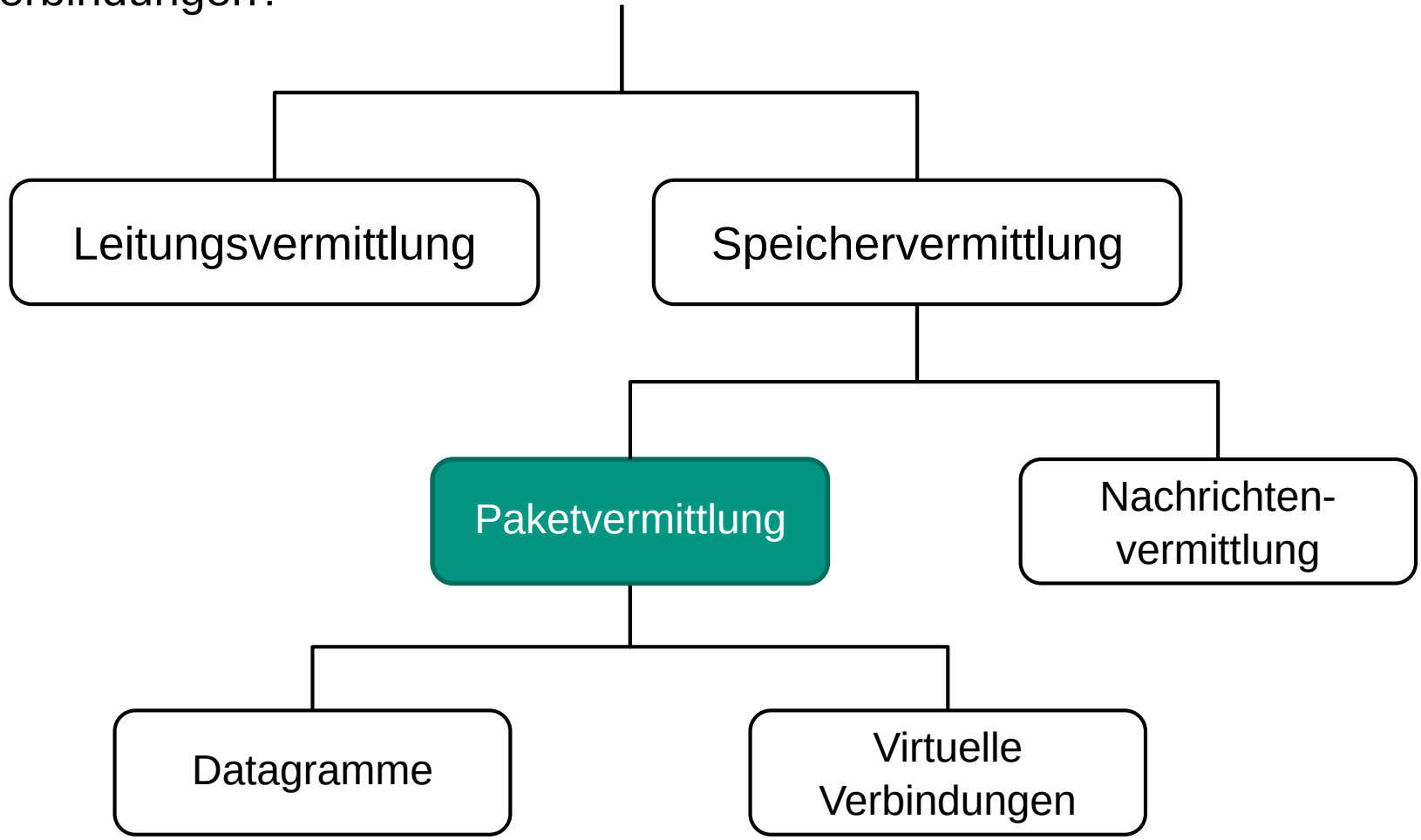
- Zu Beginn ist der Schwellenwert auf „Unendlich“ gesetzt

Einführung in Rechnernetze – 4. Übungsblatt

1. TCP-Analyse mit Wireshark
2. TCP-Mechanismen und Verbindungsverwaltung
3. TCP-Arbeitsweise und Flusskontrolle
4. TCP-Staukontrolle
5. Vermittlungsformen und -prinzipien
6. IP
7. IP-Adresskonfiguration und Subnetze

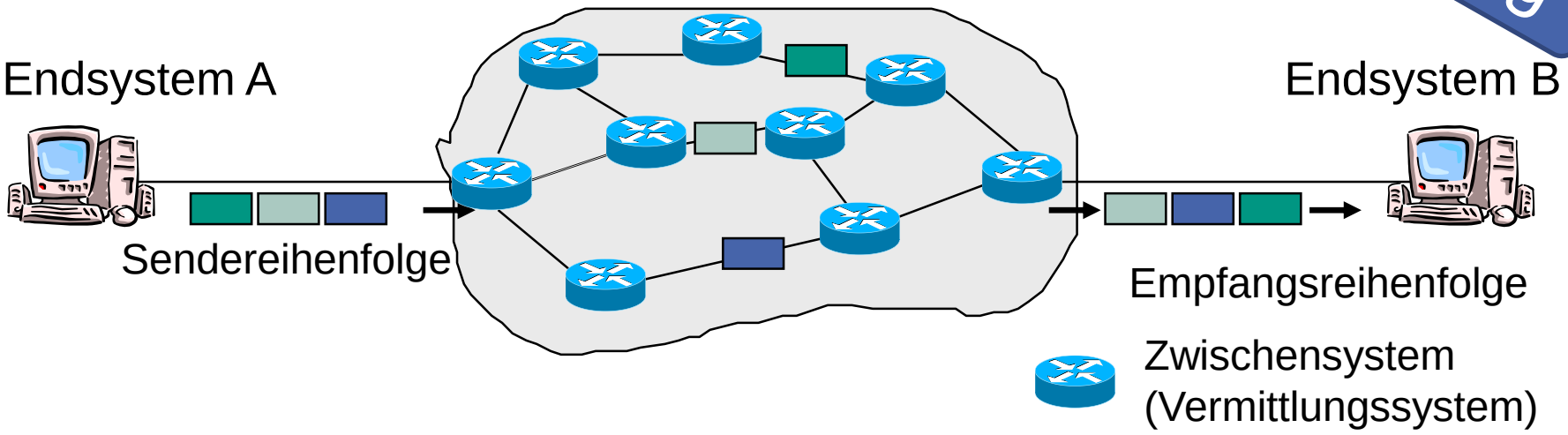
Aufgabe 5 (a)

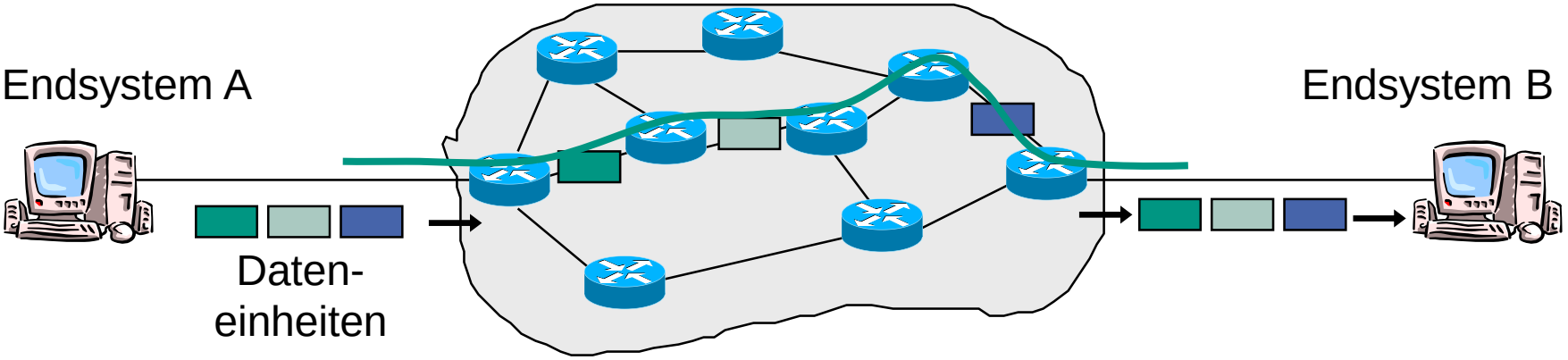
- Worin unterscheidet sich die Datagrammvermittlung von virtuellen Verbindungen?



Datagrammvermittlung und virtuelle Verbindunge

Vorlesung





Aufgabe 5 (a)

- Worin unterscheidet sich die Datagrammvermittlung von virtuellen Verbindungen?

	Virtuelle Verbindung	Datagramme
Verbindungsaufbau und -abbau	Notwendig	Nicht nötig
Zieladresse	Nur während des Verbindungsaufbaus nötig	In jeder Dateneinheit
Reihenfolge	Reihenfolgetreu	Nicht Reihenfolgetreu
Zustandshaltung in Zwischensystemen	Notwendig	Nicht nötig
(Sonstige)	Quality-of-Service einfacher realisierbar; mehr Kontrolle durch Netzbetreiber	
(Beispiele)	MPLS	Internet

Aufgabe 5 (b) – Analog-Pingo

- Sind bei virtuellen Verbindungen die Kennungen im ganzen Netz global?

  - Ja
  - Nein

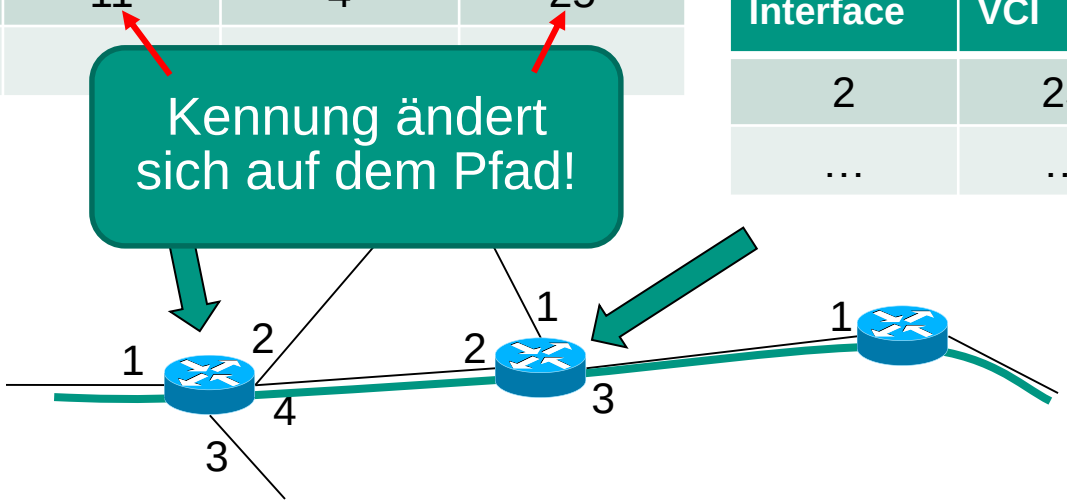
Virtuelle Verbindungen

Vorlesung

- Kennungen sind i.d.R. nur für einen Übertragungsabschnitt eindeutig
- 3 Phasen
  - Verbindungsaufbau: Virtuelle Verbindung durch Festlegung von Kennungen auf den Zwischensystemen etabliert (Zieladresse nötig)
  - Datenübertragung: Daten werden anhand der Kennungen vermittelt
  - Verbindungsabbau: Virtuelle Verbindung wird abgebaut, d.h. Vermittlungsinformationen in den Zwischensystemen werden gelöscht

Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
1	11	4	25
…			

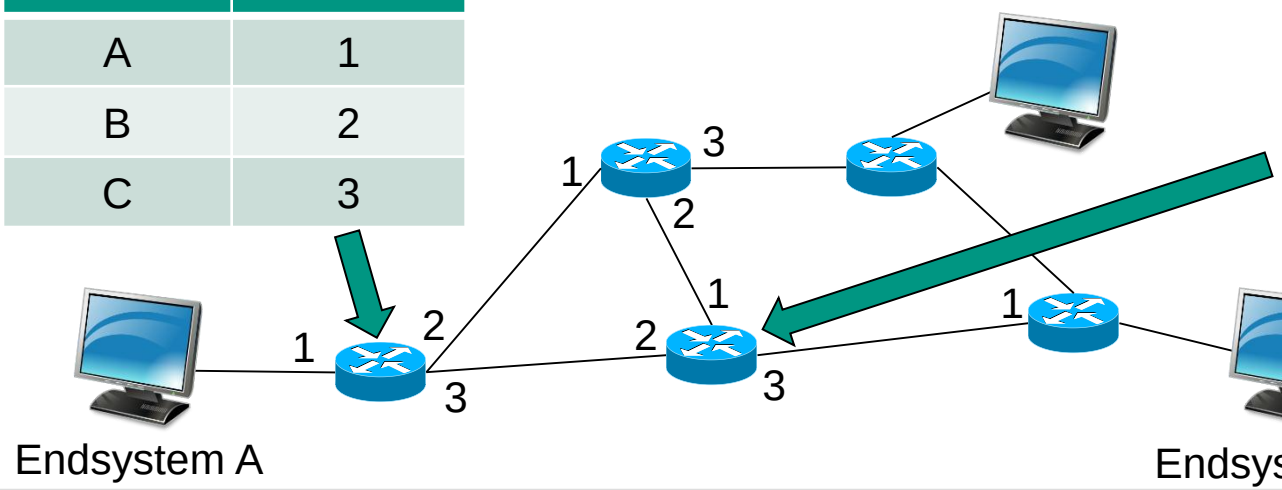
Incoming Interface	Incoming VCI	Outgoing Interface	Outgoing VCI
2	25	3	12
…	…	…	…



Aufgabe 5 (c)

- Wie funktioniert die Datagrammvermittlung?
  - Die Datagramme werden als unabhängige Einheiten betrachtet
  - Jedes Datagramm enthält seine Zieladresse
  - Jedes Zwischensystem im Netz weiß, über welches seiner Interfaces es welche Endsysteme/Subnetze erreichen kann
  - Anhand dessen lokale Weiterleitungsentscheidung für die Datagramme
    - Kann zu Verlusten oder Reihenfolgevertauschungen führen

Address	Outgoing Interface
A	1
B	2
C	3



A	Outgoing face
A	
B	1
C	3

Vorlesung

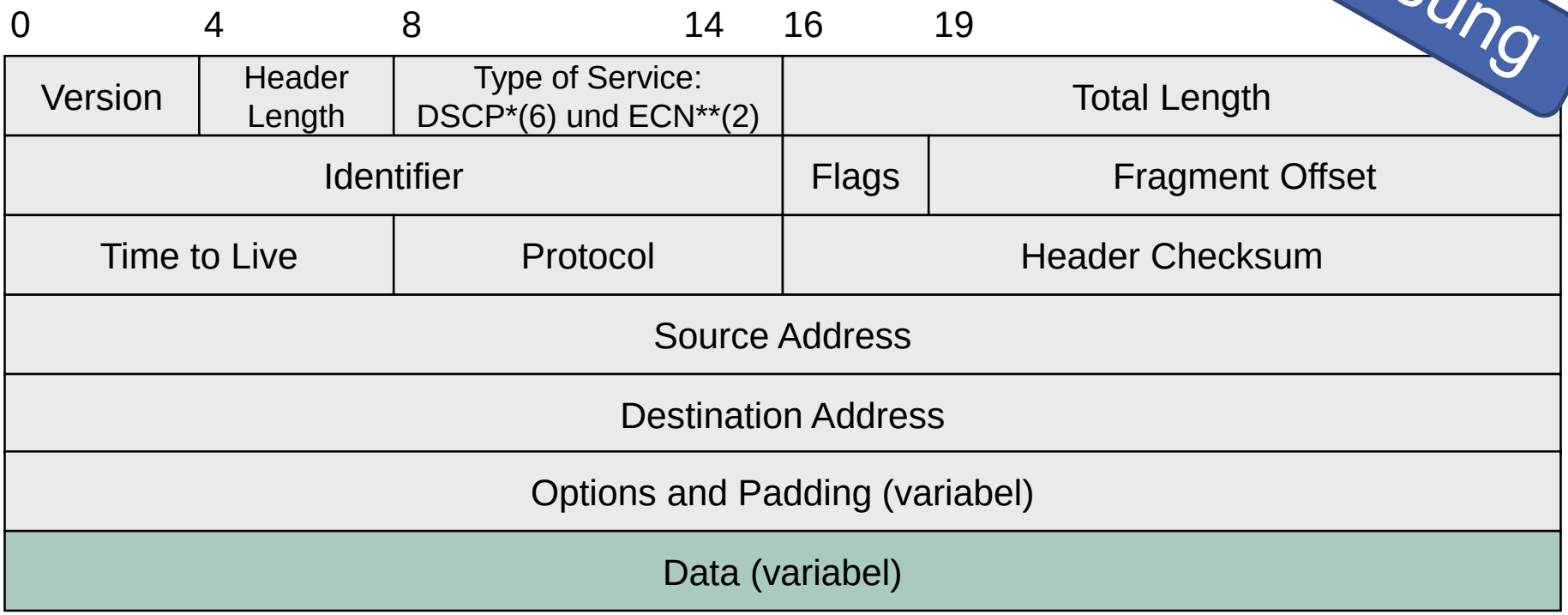
Einführung in Rechnernetze – 4. Übungsblatt

1. TCP-Analyse mit Wireshark
2. TCP-Mechanismen und Verbindungsverwaltung
3. TCP-Arbeitsweise und Flusskontrolle
4. TCP-Staukontrolle
5. Vermittlungsformen und -prinzipien
6. IP
7. IP-Adresskonfiguration und Subnetze

Aufgabe 6 (a) – Pingo

- Welche Informationen stehen im Kopf eines IP-Datagramms?
  - Time-to-Live
  - IP-Adresse des Zielsystems
  - IP-Adresse des nächsten Routers
  - IP-Adresse der Router auf dem Pfad
  - IPv4-Flag
  - Transportschichtprotokoll
  - SYN-Flag
  - Datagramm-Prüfsumme

Format eines IPv4-Datagramms



- Overhead: 20 Byte TCP-Kopf + 20 Byte IP-Kopf = 40 Byte Overhead

* Differentiated Services Code Point

NGI

- Kodiert Weiterleitungsklasse für die Erfüllung von Dienstgüte-Anforderungen (Quality of Service)

[RFC2474]

** Explicit Congestion Notification

TM

- Explizite Signalisierung von Stausituationen

[RFC3168]

Aufgabe 6 (a) – Beispiele für Optionen im IP-Header

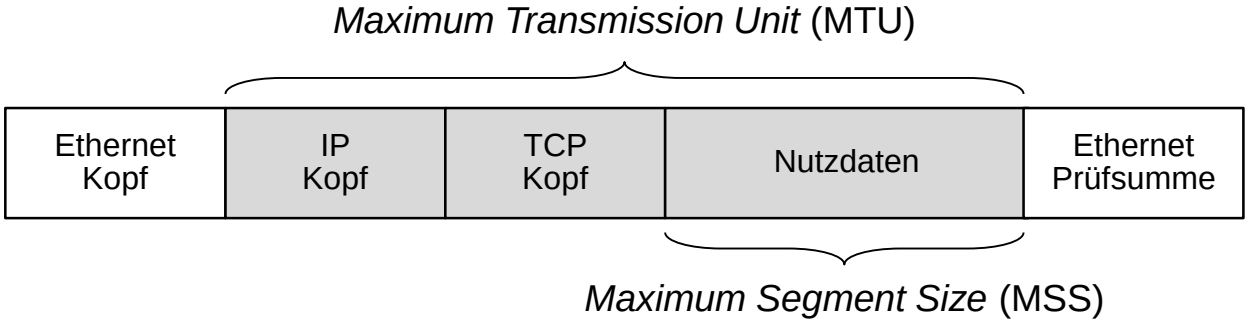
- Source-Routing
  - Absender gibt Route für Datagramm vor
- MTU-Probing (dazu gleich mehr)
- Zeitstempel (etwa für Latenz-Messungen)
- Sicherheit

```
This option provides a way for hosts to send
security, compartmentation, handling restrictions,
and TCC (closed user group) parameters. The format
for this option is as follows:

    Security (S field):
    Specifies one of 16 levels of security (eight
    of which are reserved for future use).

            00000000 00000000 - Unclassified
            11110001 00110101 - Confidential
            01111000 10011010 - EFTO
            10111100 01001101 - MMMM
            01011110 00100110 - PROG
            10101111 00010011 - Restricted
            11010111 10001000 - Secret
            01101011 11000101 - Top Secret
```

Aufgabe 6 (b)

- Die im Internet eingesetzten Protokolle TCP und IP bieten beide die Möglichkeit zur Segmentierung. Welche Möglichkeit wäre hierbei zu bevorzugen? Erläutern Sie entsprechende Vor- und Nachteile.

- Probleme bei IP-Segmentierung
  - Geht einzelnes Segment verloren, müssen alle Teil-Dateneinheiten wiederholt werden
  - Effizientes Reassemblieren in IP-Schicht schwierig
    - Wieviele Fragmente folgen noch?
    - Wie groß ist Dateneinheit insgesamt?
    - Problematisch mit NAT-Gateways
- TCP ist für Segmentierung zu bevorzugen
  - Bietet Sicherungsmechanismen (ARQ-Verfahren)
  - Arbeitet Byte-Strom orientiert, segmentiert nach *Maximum Segment Size* (MSS)
    - MSS richtet sich nach *Maximum Transmission Unit* (MTU) der Sicherungsschicht



Aufgabe 6 – Reality Check

- Verschiedene Schicht-2-Protokolle arbeiten mit unterschiedlichen MTU-Größen
  - Auszug aus RFC 1191

MTU Größe [Bytes]	Protokoll	Referenz
65535	Maximale MTU	RFC 791
8166	IEEE 802.4 (Token Bus)	RFC 1042
4464	IEEE 802.5 (4Mb max) (Token Ring)	RFC 1042
4352	FDDI	RFC 1188
1500	Ethernet II	RFC 894
1492	IEEE 802.3 (Ethernet)	RFC 1042
1280	Minimale MTU (IPv6)	RFC 2460
68	Minimale MTU (IPv4)	RFC 791

- Würde die Anwendung aus Aufgabe 2 auf Ethernet II aufsetzen, wäre die MTU auf 1500 Byte begrenzt
  - TCP würde bereits eine Segmentierung durchführen

Nachtrag zur MTU

- Auf einem Datenpfad können allerdings viele verschiedene Schicht-2-Technologien verwendet werden
  - Woher weiß Endsystem, welche Größe die minimale MTU auf dem Pfad hat?

- Immer kleinstmögliche Größe für Dateneinheit wählen
  - 576 Bytes laut IPv4-Standard ohne explizite Erlaubnis gestattet
  - 1280 Bytes bei IPv6
- Erraten der minimalen MTU entlang Pfad mittels Heuristik
- Tatsächliche minimale MTU entlang Pfad ermitteln
  - Liste aller MTUs auf Pfad?
  - Kleinste MTU kontinuierlich anpassen?
    - Verlangt Unterstützung zwischenliegender Router

- Diskussion in Paper „*Fragmentation Considered Harmful*“ und RFC 1063

→ „*Path MTU Discovery*“ (RFC 1191)

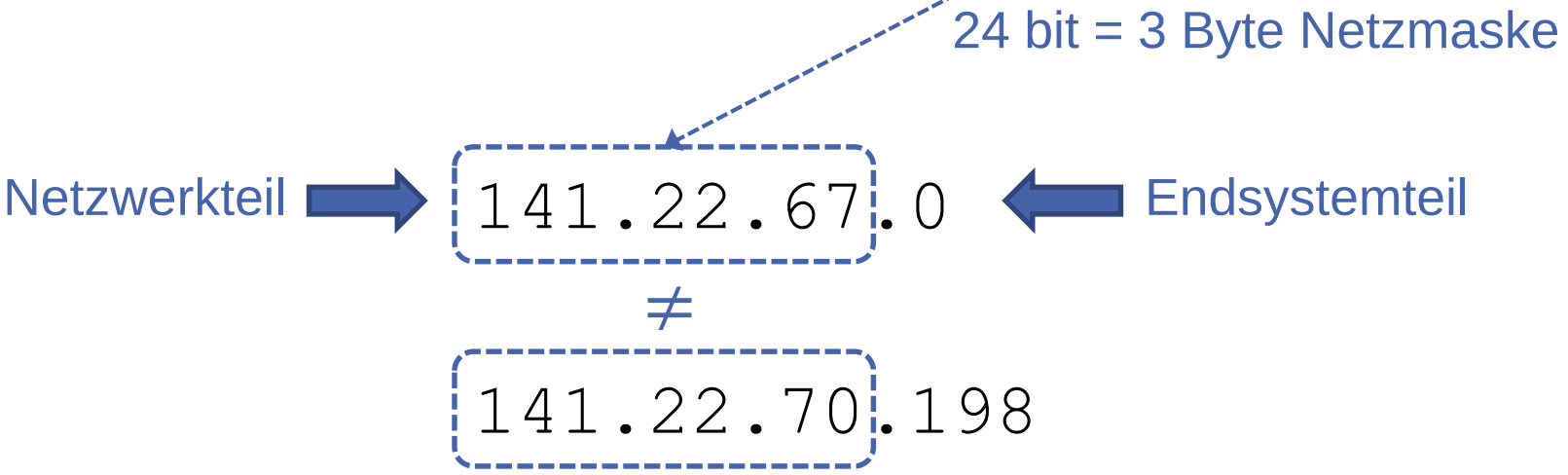
  - Sender schickt Dateneinheit mit maximal möglicher MTU
    - Gegeben durch Netzzugangspunkt
    - „*Don‘t Fragment*“-Bit im IP-Kopf gesetzt
  - Zwischenliegender Router mit geringerer MTU antwortet mit ICMP-Nachricht „*Destination Unreachable. Fragmentation needed and DF bit set*“
    - Größe der gewünschten MTU in dieser Dateneinheit angegeben

Einführung in Rechnernetze – 4. Übungsblatt

1. TCP-Analyse mit Wireshark
2. TCP-Mechanismen und Verbindungsverwaltung
3. TCP-Arbeitsweise und Flusskontrolle
4. TCP-Staukontrolle
5. Vermittlungsformen und -prinzipien
6. IP
7. IP-Adresskonfiguration und Subnetze

Aufgabe 7 (a)

- Gegeben sei der IP-Adressbereich 141.22.67.0/24
  - Liegt die Adresse 141.22.70.198 im zugehörigen Netzwerk oder nicht?



- 141.22.70.198 liegt nicht im zugehörigen Netzwerk

Aufgabe 7 (b)

- Gegeben sei der IP-Adressbereich 141.22.67.0/20
  - Liegt die Adresse 141.22.70.198 im zugehörigen Netzwerk oder nicht?

20 bit Netzmaske

141.22.67.0 =

10001101.00010110.0100 0011.00000000

=

10001101.00010110.0100 0110.11000110

= 141.22.70.198

- 141.22.70.198 liegt im zugehörigen Netzwerk

Aufgabe 7 (c)

- Gegeben sei der IP-Adressbereich 141.22.67.0/22
  - Welche IP-Adressen bilden den Anfang und das Ende dieses Adressbereichs, bzw. welche IP-Adressen liegen im zugehörigen Netzwerk?

22 bit Netzmaske

141.22.67.0 =

10001101.00010110.010000 11.00000000

Netzwerkteil mit angehängten Nullen stellt „Anfang“ dar

10001101.00010110.010000 00.00000000

= 141.22.64.0

Netzwerkteil mit angehängten Einsen stellt „Ende“ dar

10001101.00010110.010000 11.11111111

= 141.22.67.255

Aufgabe 7 (d)

- IP-Adressbereich 141.22.67.0/26 sei an 3 Standorte zu verteilen
  - Einmal 10, einmal 13 und einmal 21 Systeme
  - Mit welchen Adressen und Subnetzmasken konfigurieren Sie die einzelnen Standorte?

- Standort 1 mit 10 Systemen → mindestens 4 Bit für Endsystemteil
- Standort 2 mit 13 Systemen → mindestens 4 Bit für Endsystemteil
- Standort 3 mit 21 Systemen → mindestens 5 Bit für Endsystemteil

- Beispielhafte Aufteilung
- Standort 1:
  - 141.22.67.0/28 mit Adressen 141.22.67.0 bis 141.22.67.15
- Standort 2:
  - 141.22.67.16/28 mit Adressen 141.22.67.16 bis 141.22.67.31
- Standort 3:
  - 141.22.67.32/27 mit Adressen 141.22.67.32 bis 141.22.67.63

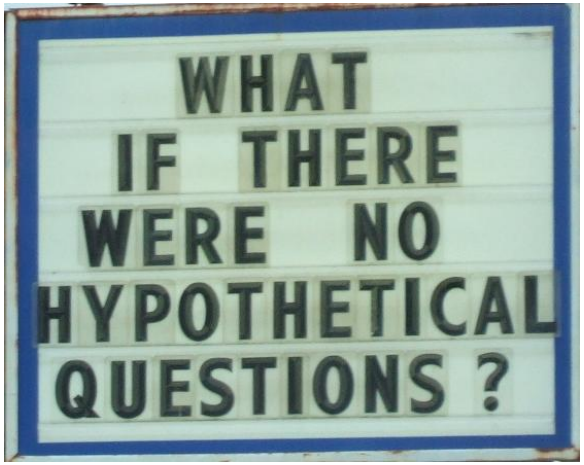
The day the routers died …

- RIPE NCC (Réseaux IP Européens Network Coordination Centre)
  - Eine von fünf „*Regional Internet Registries*“
  - Unabhängig, nicht profitorientiert
  - Mitglieder hauptsächlich Internet Service Provider, Telcos und große Unternehmen
  - Zuteilung von IPv4-, IPv6-Adressen und AS-Nummern in Europa, Zentralasien und dem Nahen Osten

- RIPE 55 in Amsterdam (Okt. 2007)
  - Address Policy Working Group
  - Anti-Spam Working Group
  - Database Working Group
  - DNS Working Group
  - ENUM Working Group
  - European Internet Exchange
  - IPv6 Working Group
  - RIPE NCC Services Working Group
  - Routing Working Group
  - Test Traffic Working Group
  - Secret Working Group

Fragen?

- Noch Fragen?
  - Jeglicher Art
    - Gute?
    - Hypothetische?
    - … sonstige?



- Vielen Dank für die Aufmerksamkeit
- Nächste Rechnernetze-Übung am

28.06.2016