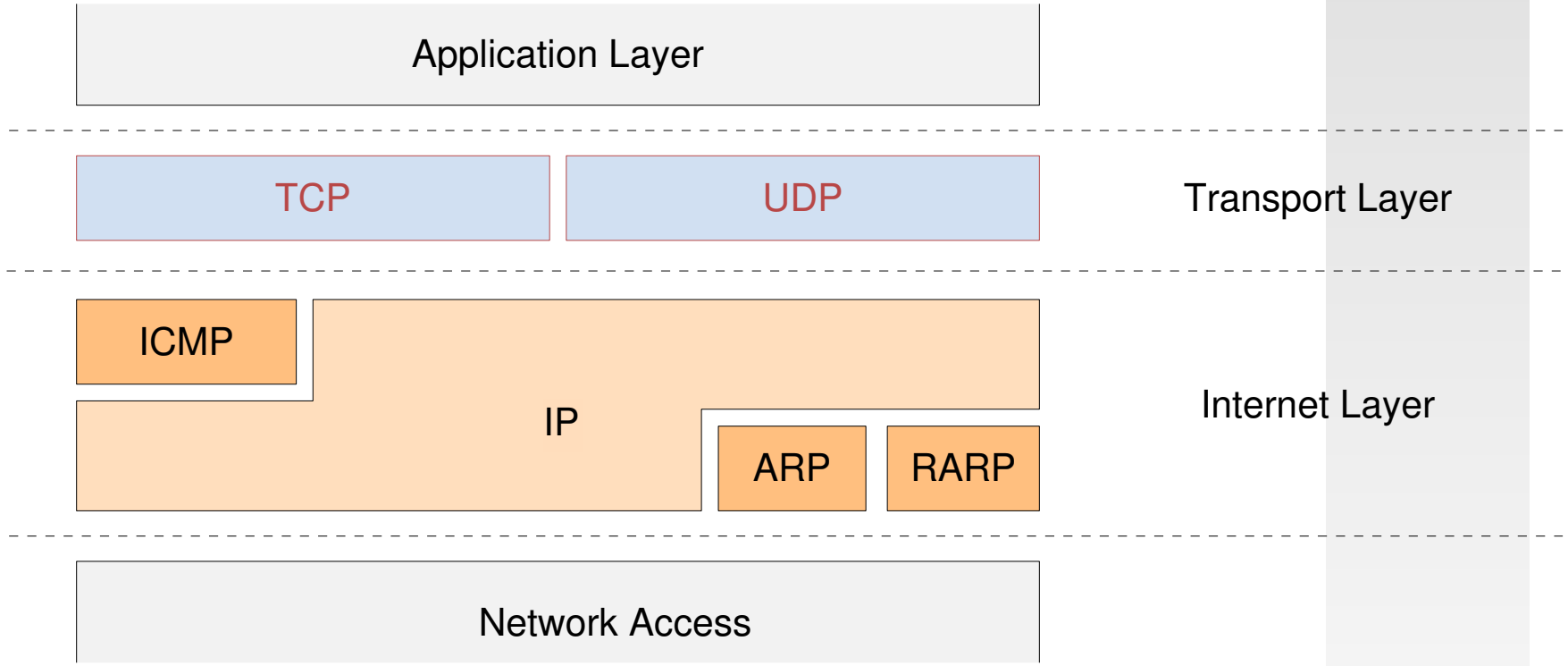
Transportprotokolle

- Einführung
- Adressierung und Formate
- Verbindungen
- Fluss- und Fehlerkontrolle
- Staukontrolle

Transportprotokolle: Einordnung

FACHHOCHSCHULE ERFURT UNIVERSITY OF APPLIED SCIENCES
Angewandte Informatik



IP - Internet Protocol
TCP - Transport Control Protocol
UDP - User Datagram Protocol

ICMP - Internet Control Message Protocol
ARP - Address Resolution Protocol
RARP - Reverse Address Resolution Protocol

Transportprotokolle: Adressierung (1)

- Adressierung
  - IP Address
  - Protocol (UDP oder TCP)
  - Port Number

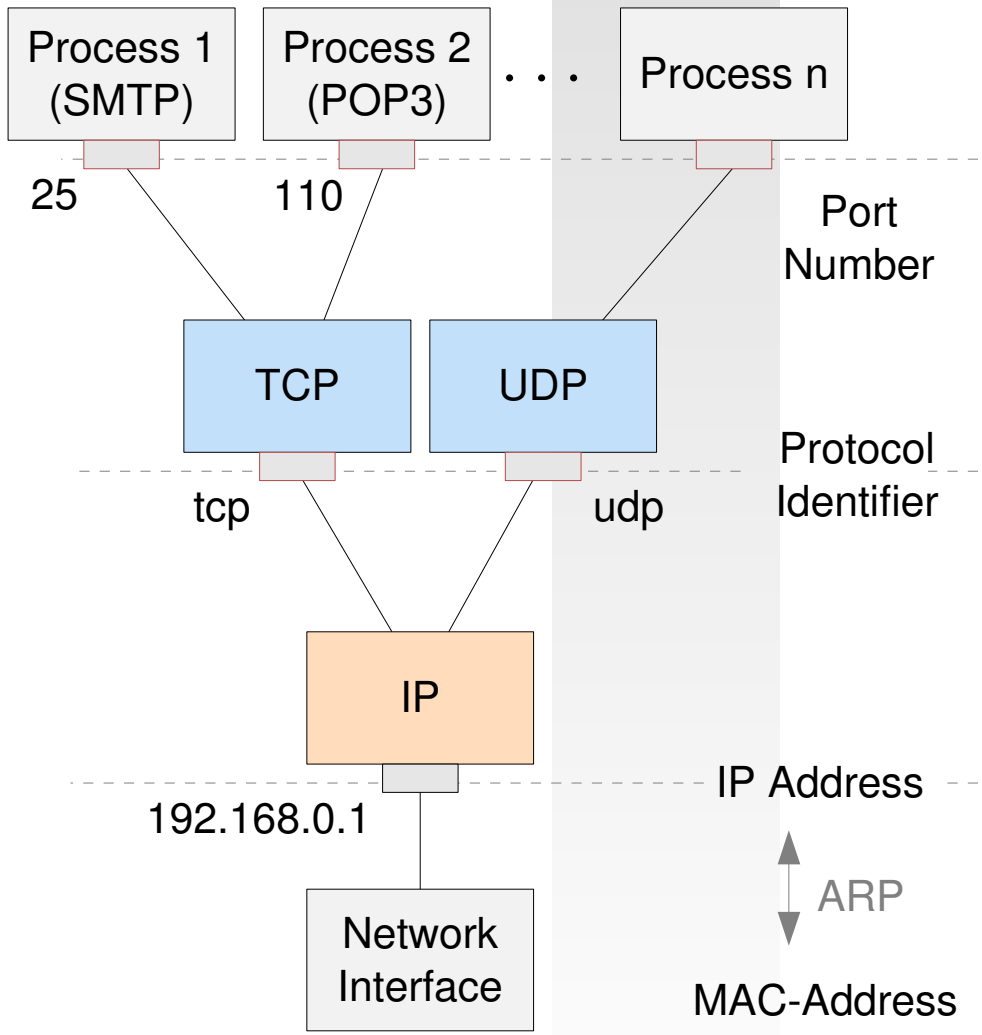
  } Socket

- Ports
  - 0 ... 1023: Well Known Ports

Port	Protocol	Application
20	TCP	FTP Data (File Transfer)
21	TCP	FTP Control
22	TCP	SSH (Secure Shell)
23	TCP	TELNET (Terminal Emulation)
25	TCP	SMTP (Simple Mail Transfer)
53	UDP	DOMAIN (Name Service)
69	UDP	TFTP (Trivial FTP)
80	TCP	HTTP (Hypertext Transfer)
111	TCP	SUN Remote Procedure Call

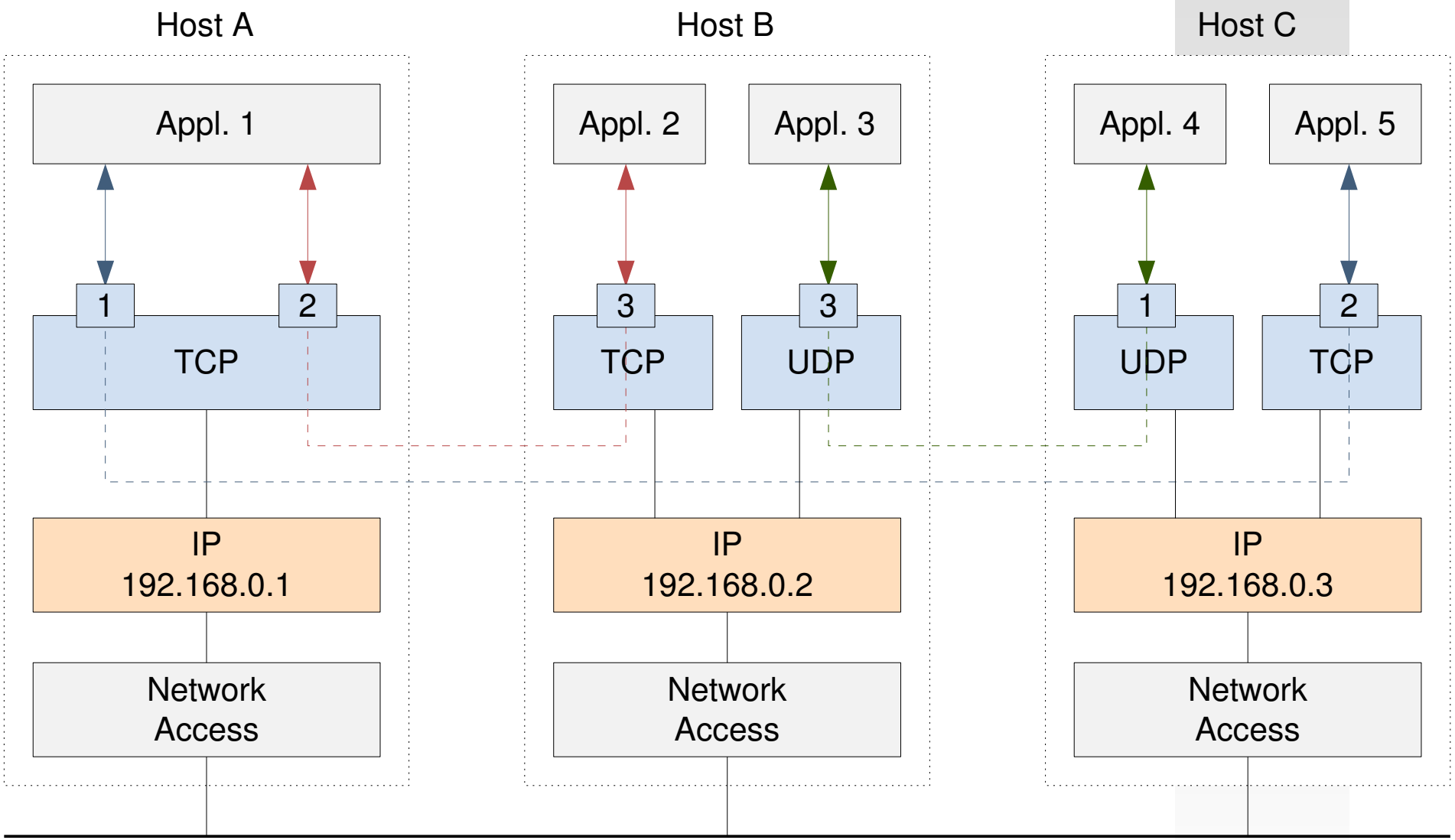
  - ≥ 1024: Anwendungen

- Beispiel: 192.168.0.1:tcp/110



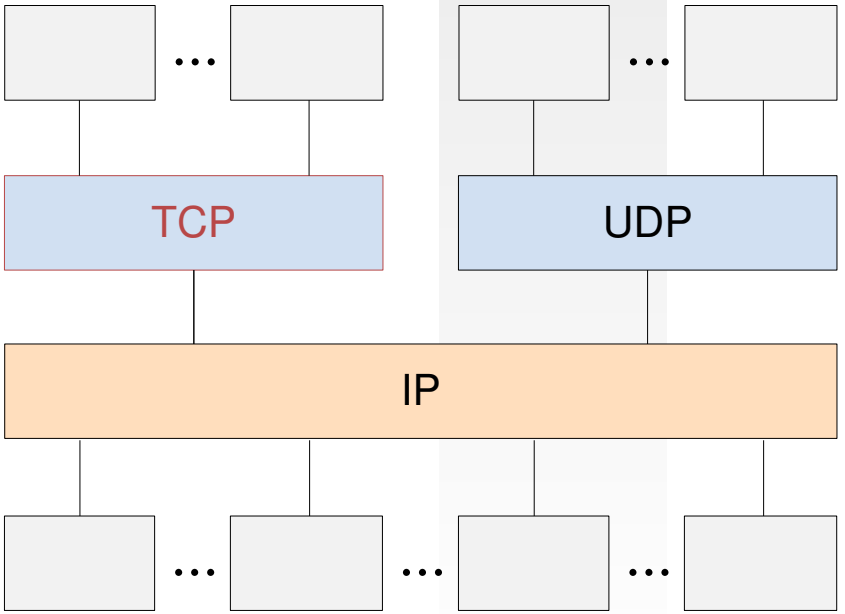
TCP/IP-Interaktionspunkte

Transportprotokolle: Adressierung (2)



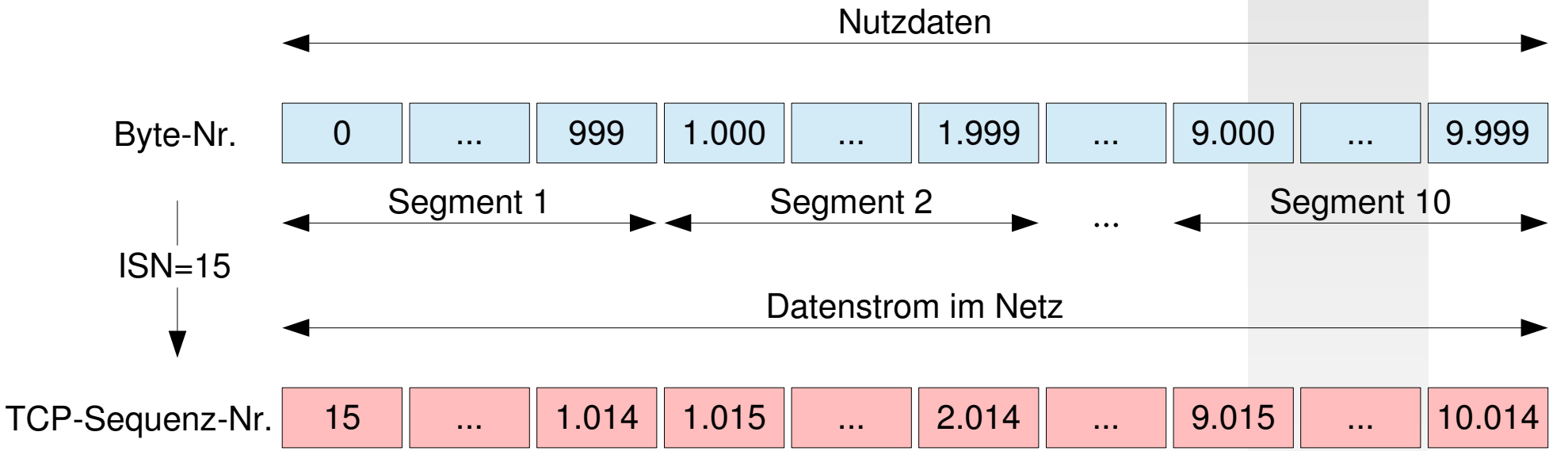
Transmission Control Protocol (TCP)

- Eigenschaften
  - verbindungsorientiert (virtuelle Verbindungen)
  - zuverlässige Übertragung (Go-Back-N, Selective-Repeat-Request)
  - realisiert transparenten Vollduplexdatenstrom (Byte-Strom)
  - besitzt Verbindungsüberwachung (Flow Control, Congestion Control)
  - dynamische Ports
  - unterstützt Piggybacking
- Standard: RFC 793 (1981)
- Anwendungen
  - Simple Mail Transfer Protocol (SMTP)
  - Hypertext Transfer Protocol (HTTP)
  - File Transfer Protocol (FTP)
  - Teletype Network TELNET
  - ...



TCP: Bytestrom / Sequenznummern

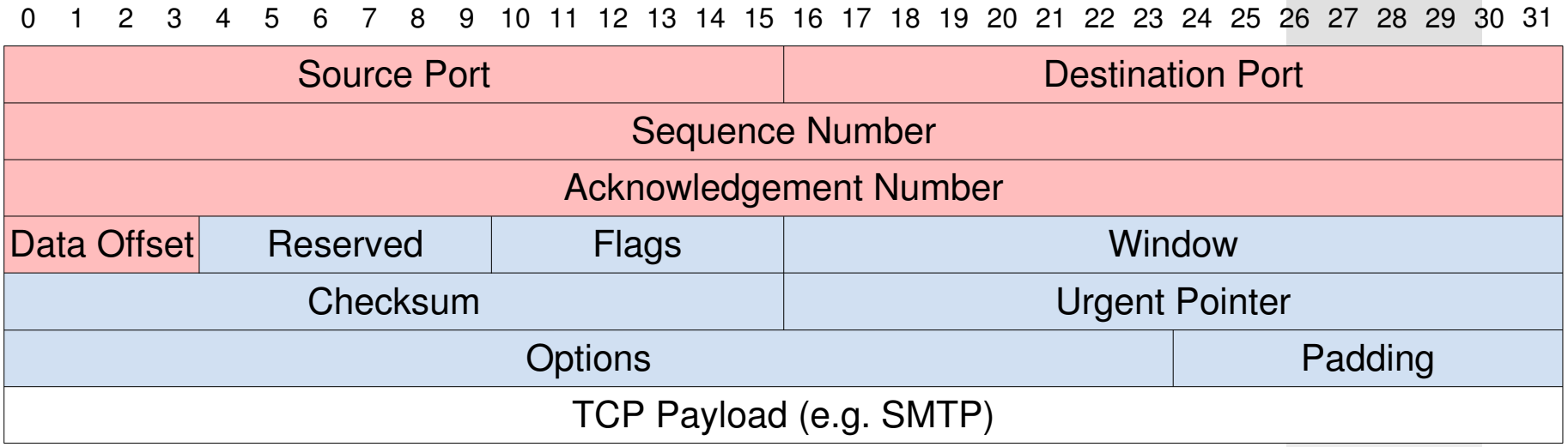
- Nutzdatenstrom
  - unstrukturiert
  - geordneter Strom von Bytes



- Datenstrom (Vollduplex)
  - geordneter Strom von Bytes
  - nummeriert entsprechend der initialen Sequenznummer = ISN (Offset)

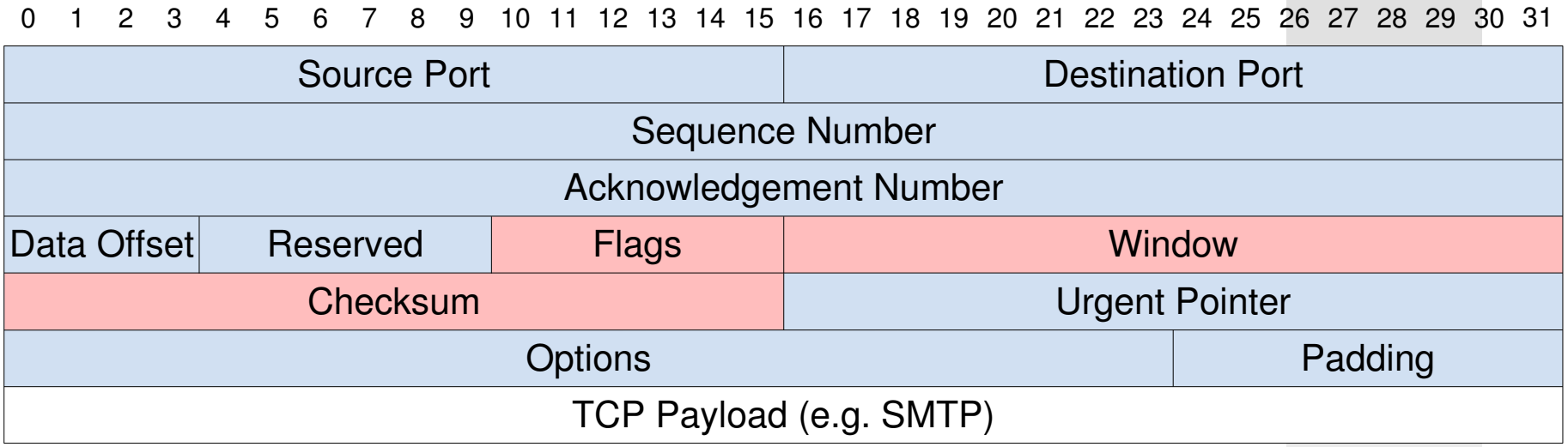
Bytestrom-Uebertragung

TCP: Paketformat (1)



- Source/Destination Port
  - Ports Anwendungsprozesse
  - Quelle/Ziel der Daten
- Sequence Number
  - Sequenznummer erstes Daten-Byte
  - initiale Sequenznummer bei SYN-Flag
  - 2^{32}-1 Sequenznummern (4 GByte)
- Acknowledgement Number
  - gültig wenn ACK-Flag gesetzt
  - Sequenznummer des nächsten erwarteten Daten-Byte
  - Quittung aller vorherigen Daten-Bytes
- Data Offset
  - Länge Header in 32-Bit-Worten

TCP: Paketformat (2)



- Flags

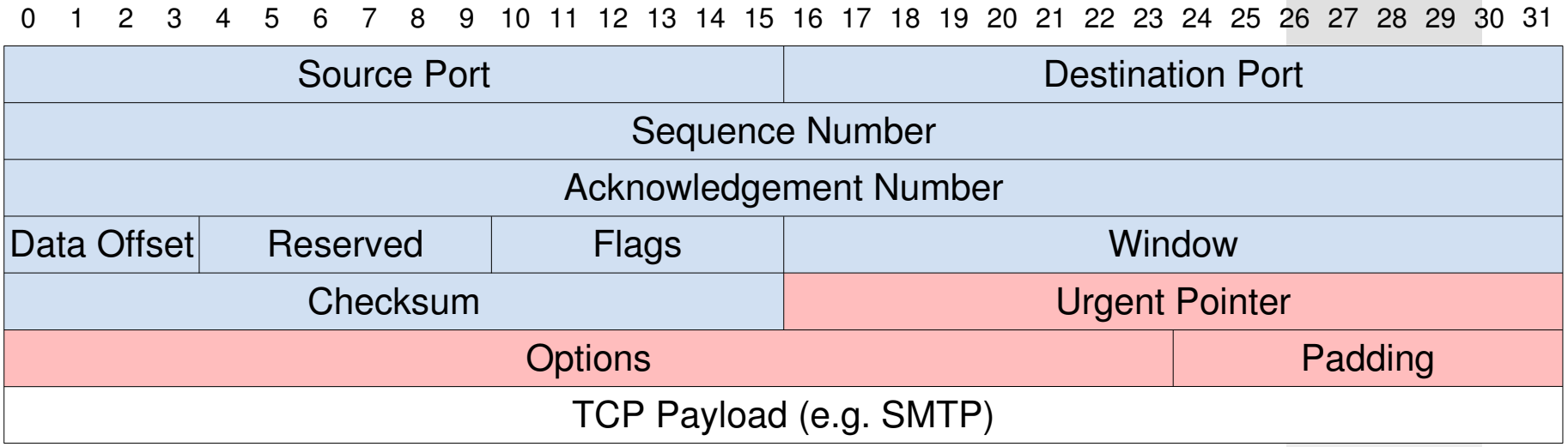
URG	ACK	PSH	RST	SYN	FIN

  - URG: Urgent Pointer gesetzt
  - ACK: Acknowledgemt Number gültig
  - PSH: Push Function
  - RST: Zurücksetzen der Verbindung
  - SYN: Starten einer Verbindung
  - FIN: Daten komplett übertragen

- Window
  - dient der Flusskontrolle
  - Anzahl Bytes, die Absender noch empfangen kann

- Checksum
  - Prüfsumme Segment+Pseudo-Header
  - bei Berechnung 0

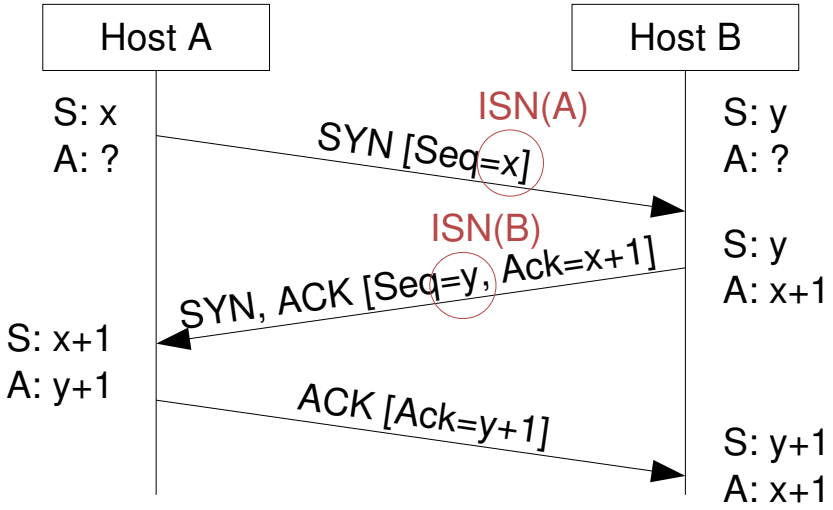
TCP: Paketformat (3)



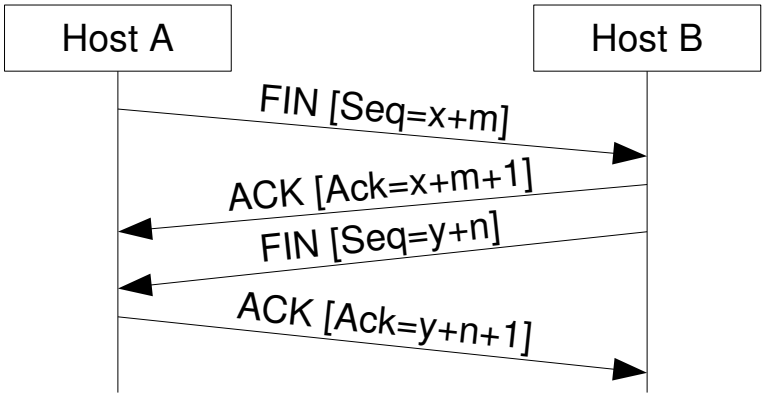
- Urgent Pointer
  - Offset zur Sequence Nummer
  - gültig bei gesetztem URG-Flag
  - zeigt auf erstes normales Daten-Byte
  - dringende Daten folgen auf Header
- Options
  - spezielle Informationen
  - z.B. Maximum Segment Size
- Padding
  - Auffüllen Header auf Wortgrenzen
  - bei Verwendung von Optionen

TCP: Verbindungsaufbau/-abbau

- Verbindunsaufbau: 3-Wege-Handshake



- Vebindungsabbau



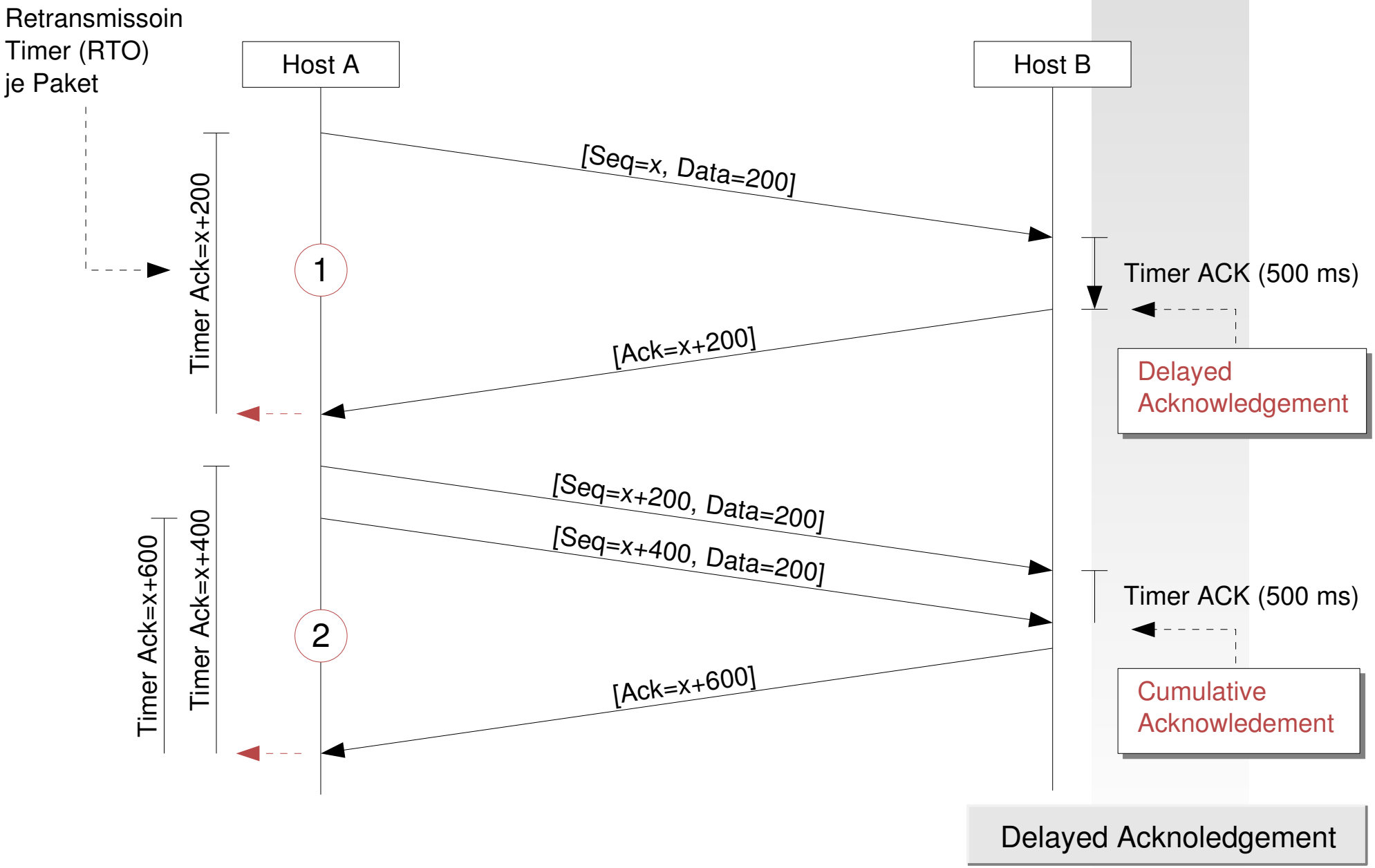
Verbindungsaufbau

Verbindungsabbau

x – Initial Sequence Number A (ISN)
y – Initial Sequence Number B (ISN)

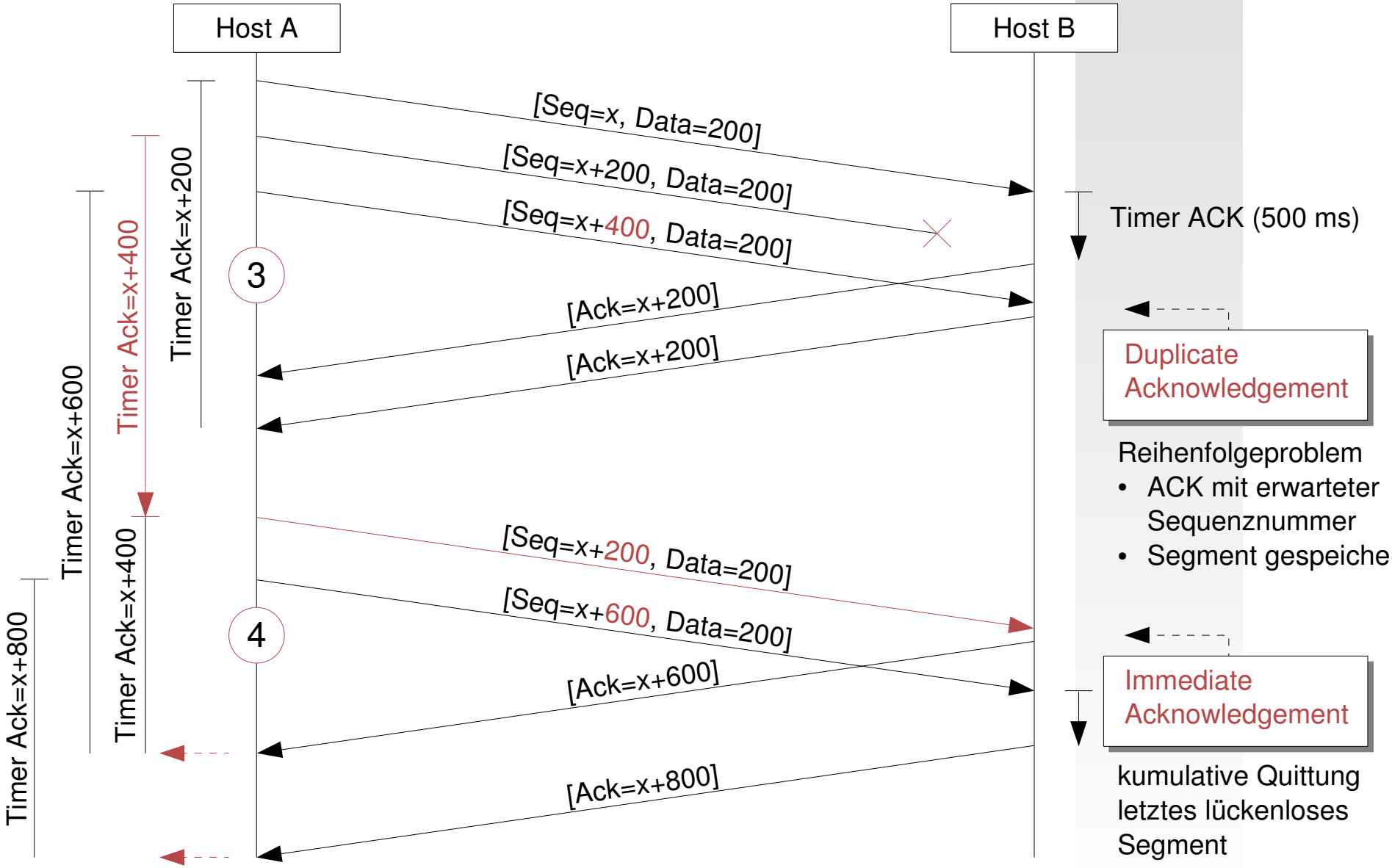
TCP: Fehlersicherung - Positive Acknowledge

FACHHOCHSCHULE ERFURT UNIVERSITY OF APPLIED SCIENCES
Angewandte Informatik



TCP: Fehlersicherung - Verlust Segment

FACHHOCHSCHULE ERFURT UNIVERSITY OF APPLIED SCIENCES
Angewandte Informatik



TCP: Fehlersicherung – Regeln Acknowledgement

Nr.	Ereignis	Reaktion TCP-Empfänger
1	Ankunft eines Segmentes in Reihenfolge. Alle bereits empfangenen Segmente quittiert	*Delayed Acknowledgement*: Bis zu 500 ms auf weitere reihenfolgetreues Segment warten. Wird kein weiteres Segment empfangen, dann Senden einer Quittung.
2	Ankunft eines Segmentes in Reihenfolge. Zuvor bereits ein reihenfolgetreues Segment empfangen.	*Cumulative Acknowledgement*: Sofortiges Senden einer kumulativen Quittung. Beide Segmente werden quittiert.
3	Ankunft eines Segmentes mit nicht erwarteter höherer Sequenznummer, Erkennen einer Lücke	*Duplicate Acknowledgement*: Sofortiges Senden einer duplizierten Quittung mit der Sequenznummer des nächsten erwarteten Bytes.
4	Ankunft eines Segmentes, das eine Lücke in empfangenen Segmenten teilweise oder komplett auffüllt	*Immediate Acknowledgement*: Sofortiges Senden einer Quittung, falls Segment an der unteren Begrenzung beginnt, sonst Zwischenspeichern des Segmentes. Falls Segment Lücke komplett füllt, Senden einer kumulativen Quittung mit Sequenznummer aller empfangenen Pakete.

TCP: Staukontrolle (Congestion Control)

- Realisierung mittels veränderlichem Congestion Control Window (CWnd)

- Slow Start
  - initial $CWnd = 1\ MSS$ (Maximum Segment Size)
  - solange $CWnd \leq SSThresh$ und ACKs vor Timeout
  - exponentielles Erhöhen $CWnd$ nach ACK aller Segmente: $CWnd = 2 \cdot CWnd$
- Congestion Avoidance
  - wenn $CWnd > SSThresh$ und ACKs vor Timeout
  - lineares Erhöhen $CWnd$ nach ACK aller Segmente: $CWnd = CWnd + 1\ MSS$
- Timeout
  - $SSThresh = CWnd/2$
  - $CWnd = 1\ MSS$

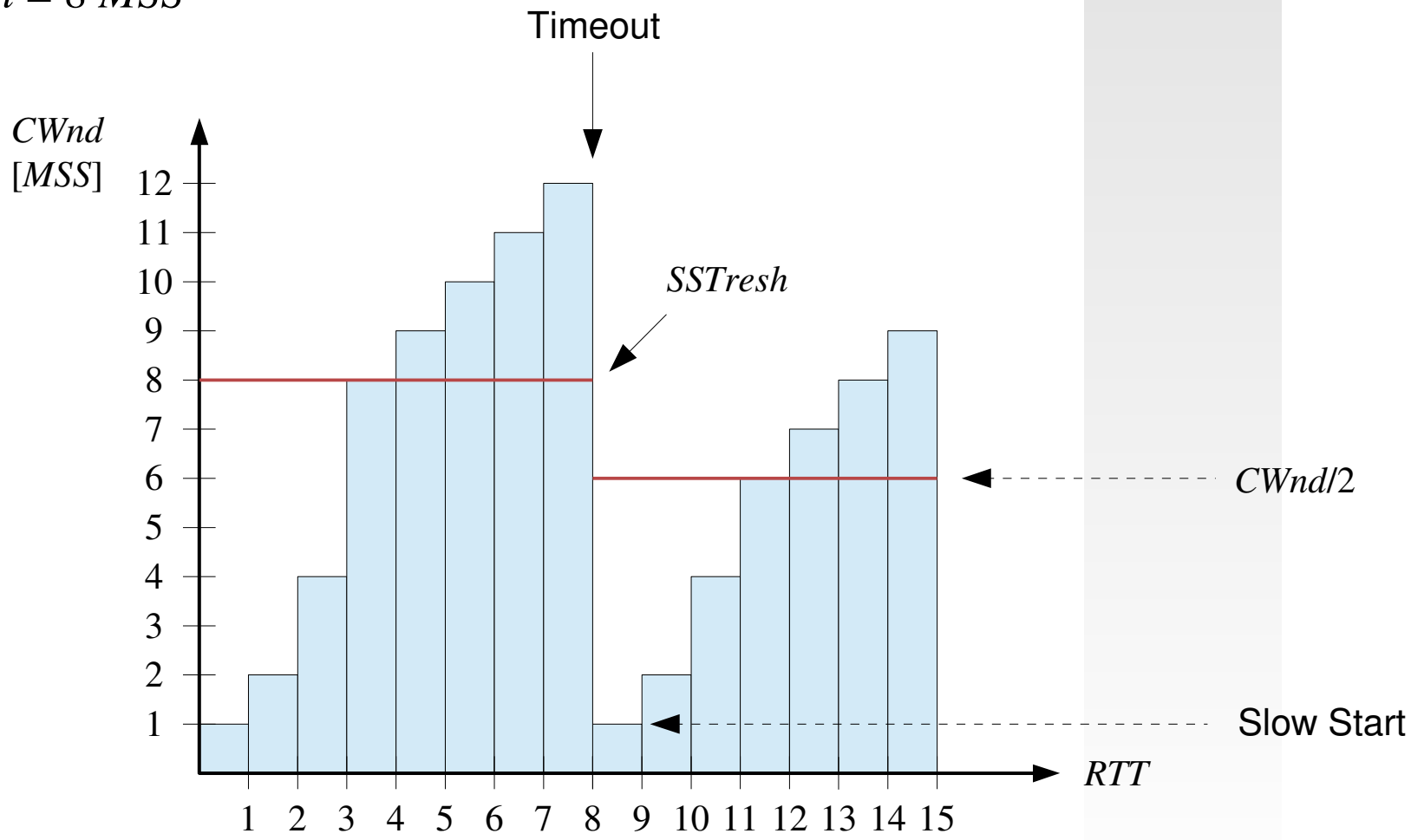
AIMD

$SSThresh$ – Slow Start Threshold
AIMD – Additive-Increase, Multiplicative-Decrease

TCP: Staukontrolle – Congestion Window

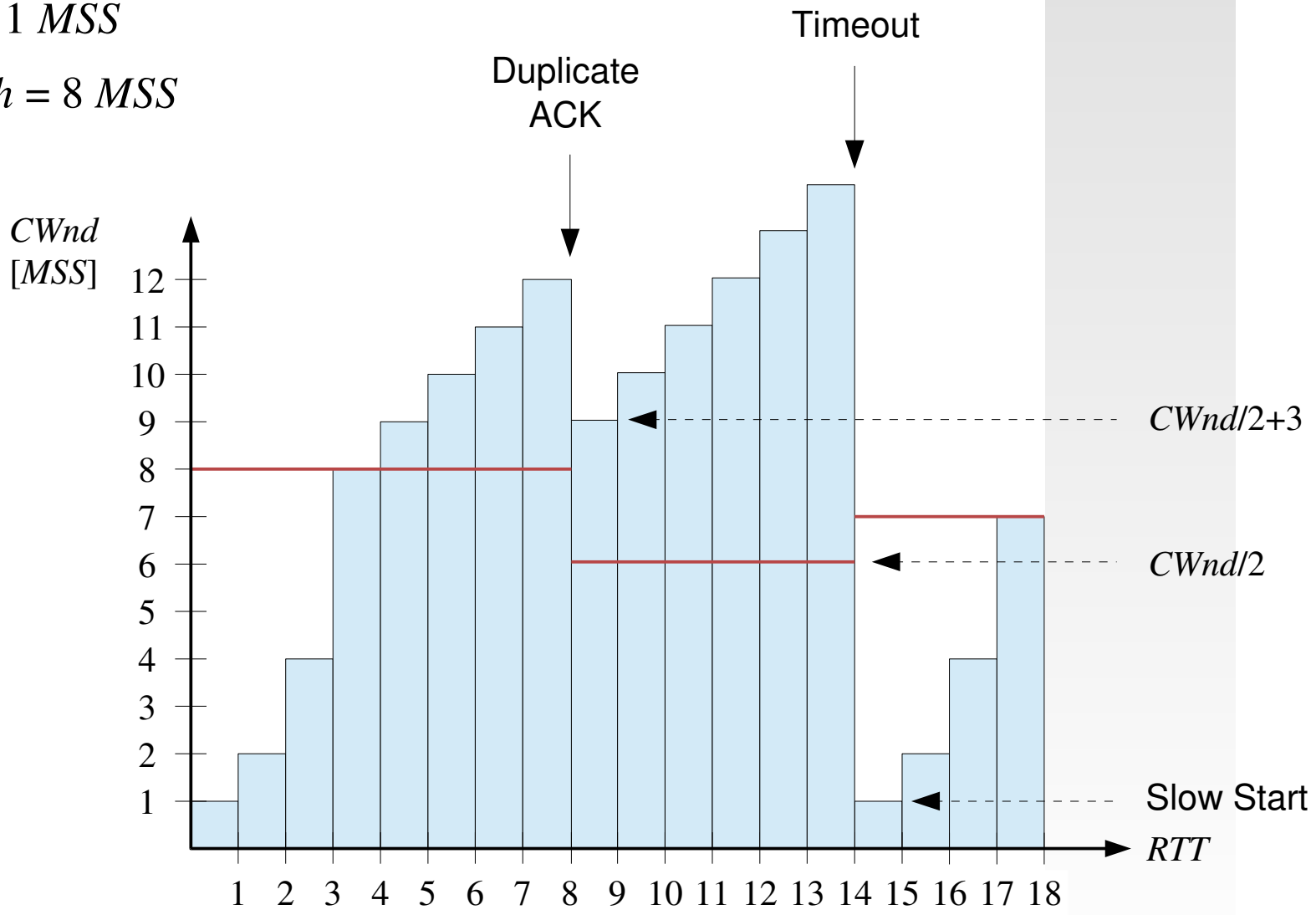
Prinzip des Slow Start

- Beispiel: initiale Variablen
  - $CWnd = 1\ MSS$
  - $SSThresh = 8\ MSS$



TCP: Staukontrolle – Fast Recovery

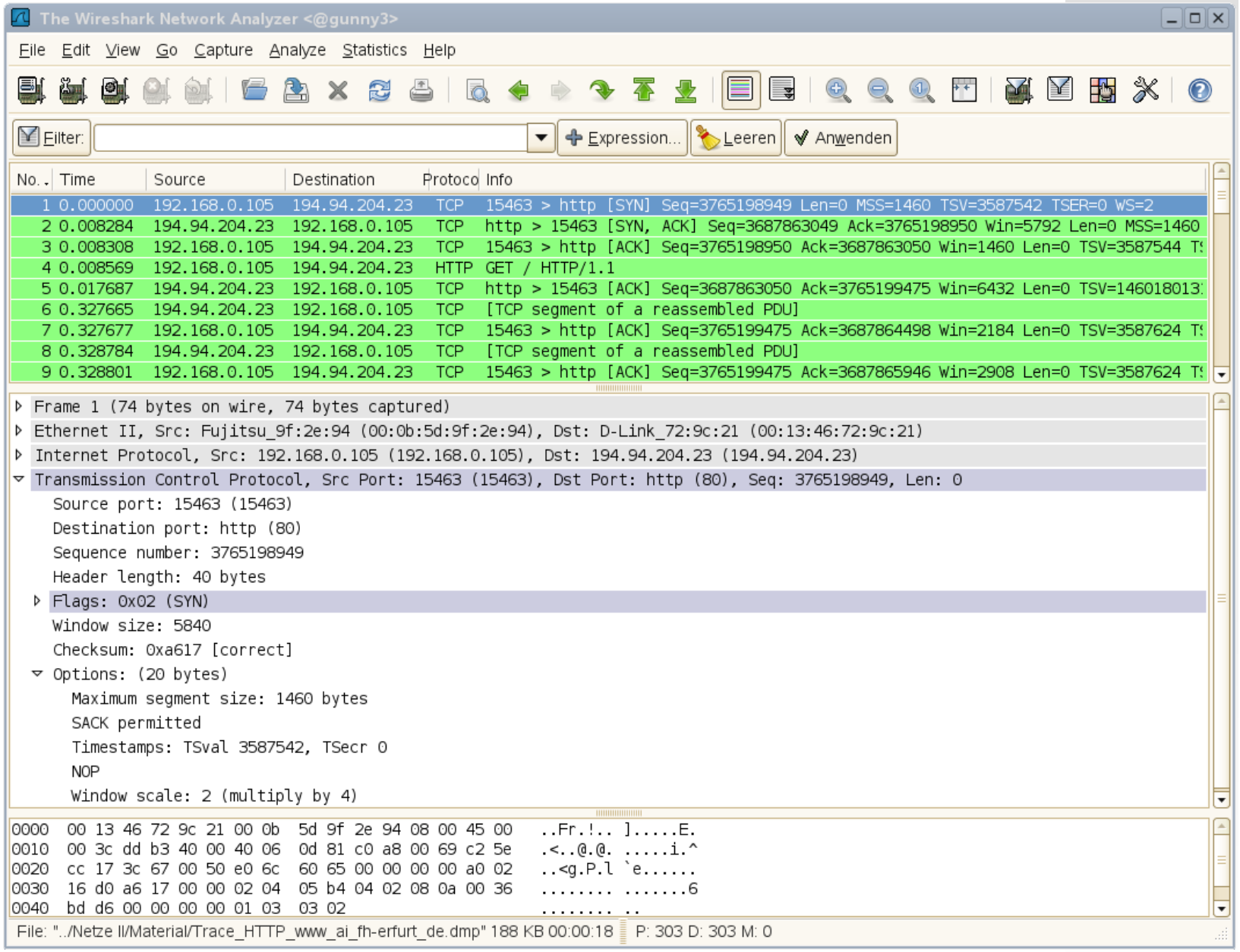
- Beispiel: initiale Variablen
  - $CWnd = 1\ MSS$
  - $SSThresh = 8\ MSS$



TCP: Varianten

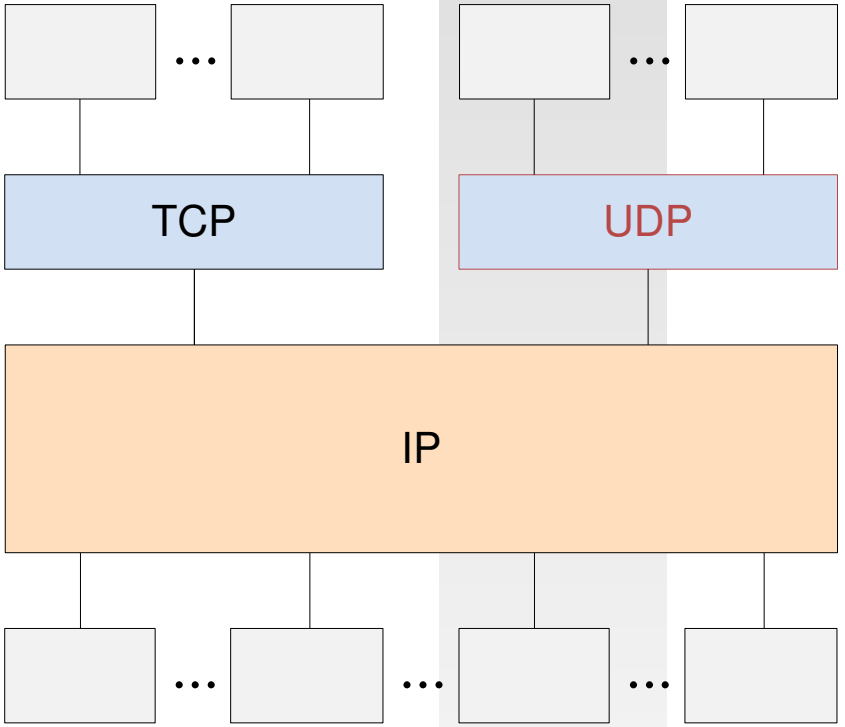
- Tahoe
  - Congestion Window und Threshold
  - Slow Start und Congestion Avoidance
- Reno
  - zusätzlich Fast Retransmit und Fast Recovery
  - in den meisten Betriebssystemen implementiert
- NewReno
  - verbessertes Fast Recovery, wenn mehrere Segmente nicht quittiert wurden
- Vegas
  - Stausituation bereits vor Segmentverlust erkennen
  - Vermutung Stausituation bei steigenden Umlaufzeiten
  - selten implementiert

TCP: Beispiel

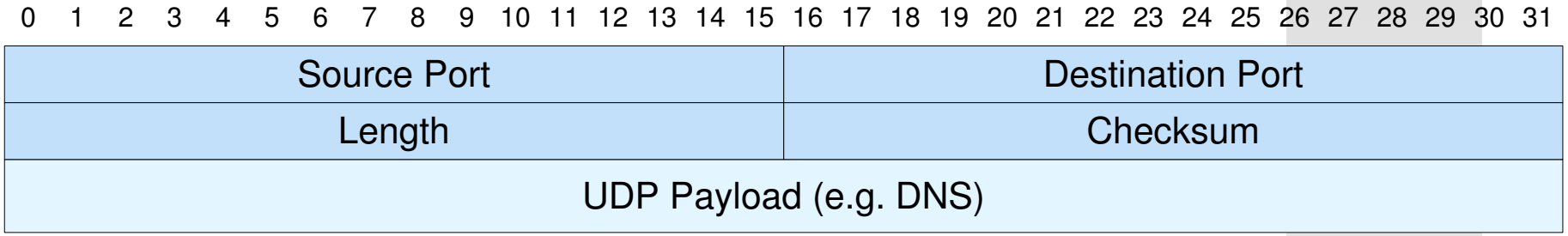


UDP: User Datagram Protocol

- Eigenschaften
  - unzuverlässige Übertragung
  - verbindungslos
- Standard: RFC 768 (1980)
- Anwendungen
  - Network File System (NFS)
  - Remote Procedure Call (RPC)
  - Domain Name Service (DNS)
  - Bootstrap Protocol (BOOTP)
  - Real-Time Transport Protocol (RTP)
  - Simple Network Management Protocol (SNMP)
  - ...



UDP: Paketformat



- Source/Destination Port
  - Ports Anwendungsprozesse
  - Quelle/Ziel der Daten
- Length
  - Länge Datagram in Byte
  - mindestens 8 Byte (Header)
- Checksum
  - optional (0 falls nicht genutzt)
  - Prüfsumme Segment+Pseudo-Header
  - bei Berechnung 0

Zusammenfassung

- Transportprotokolle
  - TCP: zuverlässig, verbindungsorient.
  - UDP: unzuverlässig, verbindungslos
- TCP-Header
  - Adressierung: Ports
  - Reihenfolge: Sequence Numbers
  - Flusskontrolle: Advertised Window
- Acknowledgements
  - Cumulative
  - Delayed
  - Immediate
  - Duplicate
- Fehlerkontrolle
  - Positive Acknowledge with Retransmit
  - Fast Retransmit
- Staukontrolle
  - Slow Start
  - Fast Recovery nach Segmentverlust