Verbindungsmanagement bei TCP:

- CLOSED: keine Verbindung aktiv oder erwartet
- TIME WAIT:
    - Rechner soll nicht überflutet werden → bei vielen Verbindungsanfragen
    - Speicherung der Verbindungsanfragen über einen Zeitraum → Ressourcen bleiben belegt → Rechner kann lahmgelegt werden
    - Pakete, die noch im Netzwerk liegen, sollten nicht zu Störungen führen (Nachlauf, um Pakete noch zuordnen und löschen zu können)

Flusssteuerung:

- Die Flusssteuerung bei TCP funktioniert nach dem Sliding Window Prinzip
- Es kann eine bestimmte Anzahl von Bytes ohne Bestätigung gesendet werden
- im „Ur-TCP" sind nur kumulative ACKs zugelassen (d.h. nach einem Paketverlust gilt das Prinzip „Gehe X zurück") → Ur-TCP, Entwicklung vor ca. 30 Jahren
- dieses Verhalten wird durch den „TCP Selective Acknowledgement Option" ergänzt (TCP-Option: SACK permitted)
- TCP-Fenster hat keine fest vorgegebene Größe, sondern kann dynamisch verändert werden
- Die SEQ- bzw. ACK-Nummern bezeichnen keine Pakete, sondern einzelne Bytes
- Bsp. Paket 1 → Empfänger bestätigt Eingang, Sender schickt währenddessen schon 2,3,4
  ➔ Immer so viele Pakete, wie durch den Speicher beim Empfänger möglich sind

Sliding Window Prinzip:

- Sender verschickt zunächst so viele Pakete, wie der Empfänger in den Speicher aufnehmen kann
- Empfänger bestätigt jedes neue Paket mit einer entsprechenden ACK
- Ist ein Paket fehlerhaft oder verloren gegangen, wird vom Empfänger die ACK des letzten vollständigen Paketes nochmals gesendet
- Der Sender schickt nun das fehlende Paket nochmal (bei Ur-TCP wurde ab der fehlerhaften Stelle alles nochmal gesendet)
- Wenn ein Paket im Speicher des Empfängers angekommen ist, teilt dieser dem Sender die neue freie Fenstergröße mit (bei jeder Änderung des Fensters)
- Nachteil:
  3 Pakete werden über das Netzwerk geschickt mit je 20+20 = 40 Bytes Header (TCP + IP)
  →ACK-Delay kann dies optimieren, indem zunächst einen Moment gewartet wird, ob das Paket im Speicher schon verarbeitet worden ist → es können dann weniger WinSize-Informationen geschickt werden
  ➔ Was passiert, wenn keine Bestätigung gesendet wird?

Dynamischer Retransmission-Timeout (RTO):

- Der Retransmission-Timeout(RTO) darf nicht fest definiert sein → verschiedene Situationen im Netzwerk, z.B. Vergleich LAN-Sendezeiten mit Transatlantikverbindung
- Internet → ca. 5sek (5000ms) bis zur erneuten Sendung (Vgl. LAN → ca. 5ms)
- Messung der Round Trip Time (RTT), z.B. bei der SYN Analyse
- Berechnung einer gemittelten RTT (smoothed):

$$T_{S,i} = \alpha \cdot T_{S,i-1} + (1-\alpha) \cdot T_{RTT}$$

Mit:

- o α = Gewichtungsfaktor → stellt eine Gewichtung zwischen vorheriger und jetziger RTT her → üblich ist 0,9
- o $T_{S,i-1}$ = zuvor berechnete gemittelte RTT
- o T_{RTT} = Umlaufzeit des aktuellen Pakets

- Berechnung der RTO:

$$T_R = \min T_0, \max\ T_U, \beta \cdot T_{S,i}$$

Stau-Vermeidung (Slow Start):

- Bsp. Router kann die Daten nicht so schnell bearbeiten oder alle Daten auf eine Leitung
  ➔ Router voll, Pakete, die ankommen, gehen verloren → Vermittlungsschicht hat keinen Mechanismus zur Stauvermeidung → Transportschicht muss dies optimieren
- Beim TCP Verbindungsaufbau wird eine bestimmte Advertised Window-Size gesetzt
- Zusätzlich zum Advertised Window wird ein sog. „Congestion Window“ (freiwillige Selbstbeschränkung) vom Sender gesetzt
- Der Sender initialisiert das Congestion Window mit einer (1) MSS
- Verdopplung nach Eintreffen eines ACKs;
- es wird maximal nur diejenige Datenmenge abgeschickt, die durch das Congestion Window aktuell festgesetzt ist
- Das Congestion Window wird solange verdoppelt, bis die Größe des Advertised Window erreicht ist

Congestional Avoidence:

- eine Überlast kann sich durch zwei Ereignisse zeigen:
  - o doppeltes ACK (fehlende Pakete)
  - o Ablauf des RTO beim Sender
- Mit Congestion Avoidence wird wie folgt darauf reagiert:
  - o Bei Eintreffen eines doppelten ACK wird das Congestion Window auf die halbe aktuelle Fenstergröße reduziert
  - o Zusätzlich wird eine sog. Slow Start Threshold (ssthres) auch auf diesen Wert gesetzt
  - o Das Congestion Window wird nun langsam (linear) vergrößert, bis es wieder die Größe des Advertised Window erreicht hat
  - o Bei erneutem doppeltem ACK während des „Hochlaufens“ erfolgt wieder eine Halbierung des Congestion Windows
  - o Bei Ablauf des RTO wird von einer größeren Überlast ausgegangen
  - o Reaktion: ssthres auf den halben Wert der aktuellen Fenstergröße setzen und das Congestion Window auf „Slow Start“ zurücksetzen
  - o Der Sender erhöht daraufhin das Congestion Window exponentiell bis zur ssthres und danach linear, bis wieder die Größe des Advertised Window erreicht ist

  ➔ Heutzutage wird häufiger das verbindungslose UDP verwendet → Stausituationen nehmen zu, weil UDP diese Stauverhinderung nicht beinhaltet

Kommunikationsschicht:

- Übernimmt Aufgaben zur Steuerung der Kommunikation zwischen Stationen (welche soll senden bzw. empfangen)
- Aufgaben der Kommunikationsschicht werden heutzutage in den Anwendungsverlauf angefügt
- Einsatz bei Client-/Server-Anwendungen
- Beispiel: Remote Procedure Call(RPC)
    - RPC wandelt Service-Anforderung eines Clients an den Server in ein standardisiertes Format um
    - Vorteil: ein Client-Prozess braucht den Server bzw. einen bestimmten Port nicht zu kennen und muss nicht selbsttätig eine TCP-Verbindung initiieren

Darstellungsschicht:

- Umwandlung der Datendarstellung auf betriebssystem-spezifische Merkmale
- Dateien werden in ein Netzwerkformat umgewandelt
- Der Zielrechner setzt diese in sein eigenes Dateisystem-Format um
- Beispiel: XDR (eXternal Data Representation) → Backup von UNIX auf Windows Rechner
- Eigentliche Darstellungsschicht hat heute keine Anwendung

Anwendungsschicht:

- Beispiele:
    - FTP (File Transfer Protocol)
    - SMTP (Simple Mail Transfer Protocol)
    - http (HyperText Transfer Protocol)
    - DNS (Domain Name Service)
    - NNTP (Network News Transport Protocol)
    - Telnet

Namensdienste im Internet:

- Im Internet (bzw. IP-Protokoll) werden von allen Protokollen zur Kommunikation nur die 32-Bit-Adressen verwendet
- Zum leichteren Merken von Adressen wird oft zusätzlich eine Zuordnung von Rechner-Namen zu den IP-Nummern vorgenommen
- Diese Namen sollten im ganzen Internet zur Verfügung stehen
- Früher: Eine zentrale Namensstelle, die einer IP einen Namenzugewiesen hat
- In kleinen Netzen kann dies per Hosts Datei geschehen (in verschiedenen Verzeichnissen je nach Betriebssystem zu finden)
- Problem: Die Datei muss ständig auf allen Rechnern aktuell sein

DNS (Domain Name Service):

- Verteilte Datenbank, in der die Zuordnungen von Rechnernamen und IP-Adressen eingetragen sind
- Wie erreicht man z.B. www.mt.haw-hamburg.de?
- Weiß man nichts über die Adresse, adressiert man das root-Verzeichnis
- Dieses gibt die Info über die Top Level Domain zurück, falls es diese kennt (z.B. de)

- Nun wird der Nameserver der Top Level Domain gefragt (z.B. Denic für .de), dieser gibt, falls bekannt, die Second Level Domain zurück (z.B. haw-hamburg)
- Der Domain-Namensraum:
    - Die Wurzel des Baumes nennt sich root-Domain (mit einem Punkt bezeichnet)
    - Die Abzweigungen heißen Knoten (engl. nodes)
    - Maximale Verzweigungstiefe: 127 Ebenen
    - Maximal 63 Zeichen pro Knotenname
    - Ein absoluter Domainname wird als „Fully Qualied Domain Name“ (FQDN) bezeichnet
    - Knotennamen müssen relativ zur Parent-Domain eindeutig sein
- Man bekommt immer nur Teilinformationen, bis man die nötige IP Adresse erhält
- Die administrative Verantwortung für eine Subdomain kann von einer beliebigen Domain übernommen werden
- Vorteile: Vermeidung von Namenskonflikten durch die hierarchische Struktur, es gibt nur eindeutige Domainnamen
- Eine heute sehr häufig eingesetzte Implementierung von DNS ist BIND (Berkley Internet Name Domain) → Betreiben eines DNS Dienstes ist nicht zwingend
- Resolver:
    - Funktion auf dem Client, die den Zugriff auf einen Nameservice ermöglicht
    - Die Abfrage des Nameservers wird auch als „Query“ bezeichnet
    - Der Resolver interpretiert die Antworten
    - Weitergabe an die Programme, die eine Namensauflösung angefordert haben
- Die Namensauflösung:
    - Nameserver können den Domain-Namensraum nach Daten absuchen (name resolution)
    - Die Auflösung beginnt bei einem „Root-Nameserver“
    - Es wird zwischen rekursiver und iterativer Auflösung unterschieden
    - Bei der rekursiven Auflösung verlangt der Resolver vom befragten Nameserver eine vollständige Namensauflösung
    - Bei der iterativen Auflösung antwortet der befragte Nameserver nur mit einer Teilantwort

    ➔ Benachbarte IP-Adressblöcke haben nichts mit der Lokalisierung zu tun!

- Bsp. Client möchte Serverdienst in Anspruch nehmen → soll auf lokales Netzwerk beschränkt werden (anhand von Freigabe) → Quell IP kann aufgegriffen und missbraucht werden → bei DNS kommen die Daten immer vom Admin (vertrauenswürdig) → wenn Anfrage beim Server eintrifft, wird der Name geprüft, nicht die IP Adresse

Inverse Abfrage:

- Die inverse Abfrage ist eine schwierige Aufgabe, da DNS nach Namen organisiert ist
- Als Lösung wurde ein gesonderter Ast im Namensbaum angelegt: in-addr.arpa
- Die Knotennamen von in-addr sind Zahlen von IP-Adressen (0-255)
- Die Subdomains sind wieder Zahlen 0-255
- Dies führt zu einer Abbildung aller IP-Adressen in einem 4-stugen Teilbaum von in-addr.arpa
- Jede Subdomain muss daher außer dem eigenen Namens-Ast einen zusätzlichen Adress-Ast bereitstellen

➔ Heutzutage sollte die Authentifizierung von Rechnern besser mit Hilfe der Kryptografie geschehen.

SMTP:

- Ablauf einer SMTP-Übertragung:
  - Aufbau einer Verbindung zu Port 25
  - Warten auf die Bereit-Meldung des Empfängers
  - Identifikation des Sendes
  - Übermittlung von Header-Informationen
  - Übermittlung der Nachricht
  - Abbau der TCP-Verbindung
- Problem älterer SMTP-Protokolle → max. 64KB → „Mailstürme“
- Ablauf einer SMTP-Verbindung anhand von „Rechner-Dialog“:
  - Empfänger (E): SMTP ready
  - Sender (S): HELO Sender-IP
  - E: Sender ok
  - S: MAIL FROM: absender@irgendwas.de (Empfänger bestätigt diese Schritte, dies wird der Vollständigkeit halber hier weggelassen)
  - S: MAIL TO (recipient to): zieladresse@empfaenger.de
  - S: DATA

    Subject: Inhalt der Mail (Betreff)
    Text [Daten]
    . (Punkt beendet den Datenteil)
  - S: QUIT (für SMTP und TCP)
- Kann auch wird per Telnet durchgeführt werden → Voraussetzung: Auf dem Zielrechner läuft ein Mailserver