Grundlagen Rechnernetze und Verteilte Systeme
TUM Sommersemester 2012
Dozent: Georg Carle

Janosch Maier

19. Juli 2012

Inhaltsverzeichnis

0 Grundlagen **5**

1 Physikalische Schicht **6**
1.1 Signale / Informationen 6
1.1.1 Signal 6
1.1.2 Information 6
1.1.3 Entropie 6
1.2 Klassifizierung von Signalen 6
1.2.1 Periodische Signale 6
1.2.2 Nicht-periodische Signale 6
1.2.3 Abtastung, Rekonstruktion, Quantisierung 6
1.3 Übertragungskanal 7
1.3.1 Rauschfreier, binärer Kanal 7
1.3.2 Rauschfreier, M-ärer Kanal 7
1.3.3 Rauschen 7
1.3.4 Rauschbehafteter, M-ärer Kanal 7
1.3.5 Zusammenfassung 7
1.4 Nachrichtenübertragung 7
1.4.1 Quellencodierung 7
1.4.2 Kanalkodierung 8
1.4.3 Impulsformung 8
1.4.4 Modulation 9
1.5 Übertragungsmedien 9

2 Sicherungsschicht **10**
2.1 Probleme & Motivation 10
2.2 Darstellung von Netzwerken als Graphen 10
2.2.1 All-pair-shortest-distance-Problem 10
2.2.2 Generierung von Baumstrukturen 10
2.3 Verbindungscharakterisierung, Mehrfachzugriff, Medienzugriffskontrolle 10
2.4 Verbindungscharakterisierung 10
2.5 Mehrfachzugriff (Multiplexing) 11
2.6 Medienzugriffskontrolle 11
2.7 Rahmenbildung, Adressierung & Fehlererkennung 12
2.8 Erkennung von Rahmengrenzen 12
2.8.1 Adressierung und Fehlererkennung 13
2.9 Verbindung auf Schicht 1 & 2 14
2.9.1 Hub 14
2.9.2 Brücke 14
2.9.3 Switch 14
2.10 Schleifen auf Schicht 2 14

3 Vermittlungsschicht **15**
3.1 Motivation 15
3.2 Vermittlungsarten 15
3.2.1 Leitungsvermittlung 15
3.2.2 Nachrichtenvermittlung 15

3.2.3 Paketvermittlung . . . 16
3.3 Adressierung im Internet . . . 16
3.3.1 IPv4 . . . 16
3.3.2 IP-Header . . . 16
3.3.3 Adressauflösung . . . 17
3.3.4 Internet Control Message Protocol – ICMP . . . 17
3.3.5 Adressklassen – Classfull Routing . . . 17
3.3.6 Subnetting – Classless Routing . . . 17
3.4 Wegwahl (Routing) . . . 18
3.4.1 Statisches Routing . . . 18
3.4.2 Dynamisches Routing . . . 18
3.4.3 Routing Information Protocol (RIP) . . . 18
3.4.4 Autonome Systeme . . . 19
3.5 IPv6 . . . 19

4 Transportschicht **20**
4.1 Motivation . . . 20
4.1.1 Aufgaben . . . 20
4.2 Multiplexing . . . 20
4.3 Verbindungslose Übertragung . . . 20
4.3.1 Funktion . . . 20
4.3.2 Probleme . . . 20
4.3.3 User Datagram Protocol (UDP) . . . 21
4.4 Verbindungsorientierte Übertragung . . . 21
4.4.1 Verbindungsphasen . . . 22
4.4.2 Sliding-Window Verfahren . . . 22
4.4.3 Tranmission Control Protocol (TCP) . . . 23
4.4.4 Fluss- & Staukontrolle . . . 23
4.5 Network Address Translation (NAT) . . . 24
4.5.1 NAT-Tabelle . . . 24
4.5.2 NAT-Varianten . . . 25
4.5.3 Anmerkungen . . . 25
4.5.4 NAT und ICMP . . . 25

5 Sitzungsschicht **26**
5.1 Dienste der Sitzungsschicht . . . 26
5.2 Funktionseinheiten . . . 26
5.2.1 Kombination von Funktionseinheiten . . . 26
5.3 Synchronisation . . . 27
5.4 Quality of Service . . . 27

6 Darstellungsschicht **28**
6.1 Aufgaben . . . 28
6.2 Datenkompression und Umkodierung . . . 28
6.2.1 Huffman-Code . . . 28
6.3 Einheitliche Syntax . . . 29
6.3.1 Abstrakte Syntaxnotation Nummer 1: ASN.1 . . . 29
6.3.2 Basic Encoding Rules (BER) . . . 29

7 Anwendungsschicht **31**
7.1 Domain Name System (DNS) 31
7.1.1 Zonendatei 31
7.1.2 Rekursive Namensauflösung 31
7.1.3 Iterative Namensauflösung 31

8 Verteilte Systeme **32**
8.1 Motivation 32
8.2 Homogene, skalierbare Paradigmen 32
8.2.1 Message Passing Interface (MPI) 32
8.2.2 MpaReduce 32
8.2.3 Pipes 32
8.3 Remote Procedure Call 33
8.4 Shared Memory 33
8.4.1 Non-Uniform Memory Access (NUMA) 33
8.4.2 Distributed Shared Memory über Paging 33
8.4.3 Distributed Software Transactional Memory 33
8.5 Einbettung in Programmiersprachen – Erlang 34

9 Kyptografie **35**
9.1 Ziele kryptografischer Verfahren 35
9.2 Klassifizierung von Verschlüsselungsverfahren 35
9.3 Symmetrische Verfahren: RC4 35
9.3.1 Cipher Stream 35
9.3.2 Zu beachten 35
9.3.3 Schlüsselaustausch: Diffie-Helmann 35
9.3.4 Sicherheit 36
9.4 Transport Layer Security (TLS) 36
9.4.1 Dienste von TLS 36
9.4.2 TLS Handshake Protocol 36

0 Grundlagen

7 Schichten:

- Anwendungsschicht
- Darstellungsschicht
- Sitzungsschicht
- Transportschicht – Segmente
- Vermittlungsschicht – Pakete
- Sicherungsschicht – Rahmen
- Physikalische Schicht

1 Physikalische Schicht

1.1 Signale / Informationen

1.1.1 Signal

- Physikalische Größe
- Transportiert Information $\Rightarrow$ Zuordnungsvorschrift

1.1.2 Information

- Änderung eines Signals
- Abhängig von Wahrscheinlichkeit $I(p) = -\log_2(p)$ bit

1.1.3 Entropie

Mittlerer Informationsgehalt $H(x) = p_0 I(p_0) + p_1 I(p_1) + \ldots$ bit/Zeichen

1.2 Klassifizierung von Signalen

1.2.1 Periodische Signale

Darstellbar durch Fourier Reihe (Überlagerung von Sinus-/Kosinusschwingungen)

$$s(t) = \frac{a_0}{2} + \sum_{k=1}^{\infty}(a_k \cos(k\omega t) + b_k \sin(k\omega t))$$

$$a_t = \frac{2}{T}\int_0^T s(t)\cos(k\omega t)dt, b_k = \frac{2}{T}\int_0^T s(t)\sin(k\omega t)dt$$

1.2.2 Nicht-periodische Signale

- Nicht entwickelbar
- Kontinuierliches Spektrum
- Frequenzbereich durch Fouriertransformation darstellbar

1.2.3 Abtastung, Rekonstruktion, Quantisierung

- Abtastung zu diskreten Zeiten
- Quantisierung auf Wertebereich

$\Longrightarrow$ Digitales Signal als Wort fester Länge festgelegt

Abtastwerte $\hat{s}[n]$ sind Stützstellen, dienen als Gewichte für passende Ansatzfunktion zur Rekonstruktion des Signals.

Abtastheorem von Shannon und Nyquist Signal mit $|f| \leq B$ ist vollständig beschrieben, wenn Abtastwerte nicht weiter als $\frac{1}{2B}$ entfernt sind. Vollständige Signalrekonstruktion bei $f_a > 2B$ (Sonst Aliasing).

Wortbreite von N bit $\rightarrow 2^N$ diskrete Signalstufen. Lineare Quantisierung, bei gleicher Wahrscheinlichkeit aller I_Q.

1.3 Übertragungskanal

Kanal beeinflusst Signal durch:

- Dämpfung
- Tiefpassfilterung (Frequenzen $f \geq B$ werden gesperrt)
- Verzögerung
- Additive White Gaussion Noise

Zusätzlich (aber nicht hier behandelt:)

- Interferenzen, Reflexion, Zeitvariante Einflüsse

1.3.1 Rauschfreier, binärer Kanal

$f_N = 2B$ heißt Nyquist Rate (Minimale Abtastrate, Obere Schranke and differenzierbaren Werten)

1.3.2 Rauschfreier, M-ärer Kanal

$M = 2^N$ unterscheidbare Symbole. Begrenzung der Kanalkapazität durch (Hartley's Law):

$$C_H = 2B \log(M) \text{bit}$$

1.3.3 Rauschen

Erschwert Unterscheidung von Signalstufen.

Stärke des Rauschens: Signal to Noise Ratio $SNR = \frac{\text{Sendeleistung}}{\text{Rauschleistung}} = \frac{P_S}{P_N}$. Angabe in Dezibel:

$$SNR \text{ dB} = 10 * \log_{10}(\frac{P_s}{P_n})$$

1.3.4 Rauschbehafteter, M-ärer Kanal

Maximale Datenrate (Shannon-Hartley-Theorem)

$$C_S = B \log_2(1 + \frac{P_S}{P_N}) \text{bit}$$

1.3.5 Zusammenfassung

Maximale Kanalkapazität:

$$C < \min\{C_H, C_S\} = \min\{2B \log_2(M), B \log_2(1 + SNR)\}$$

1.4 Nachrichtenübertragung

1.4.1 Quellencodierung

Entfernen von Redundanz / Verlustloses Komprimieren (vgl. Kapitel 6)

1.4.2 Kanalkodierung

Gezieltes hinzufügen von Redundanz zur

- Erkennung von Bitfehlern
- Korrigierung von Bitfehlern

Bei digitalen Übertragungen i.A. nicht ausreichend.

Blockcodes: Blöcke der Länge k, Kanalwörter der Länge $n > k$. Coderate: $R = \frac{k}{n}$

1.4.3 Impulsformung

Erzeugen von gewichteten Sendeimpulsen um analoges Signal zu erzeugen.

Leitungscode

- Abfolge von Grundimpulsen (bestehtend aus Symbolen = physische Veränderung des Signals)
- Anzahl von Signalstufen (binär, …)
- Anzahl codierter Bits pro Symbol
- Schrittgeschwindigkeit (Symbolrate) in Boud (bd)

Spektrum hängt von Anzahl der Signalwechsel ab.

Beispiele für Leitungscodes

- NRZ – Non-Return-To-Zero (Binär, 1 Symbol / Bit, Keine Taktrückgewinnung, Keine Gleichstromfreiheit, Hohe Bandbreite)
- RZ – Return-To-Zero (Binär, 2 Symbole / Bit, Taktrückgewinnung, Keine Gleichstromfreiheit, Bandbreite ist Doppelt so hoch, wie bei NRZ)
- Manchester Code (Steigende / Fallende Kanten, Binär, 2 Symbole / Bit, Taktrückgewinnung, Gleichstromfreiheit, Bandbreite ist Doppelt so hoch, wie NRZ)
- MLT3 – Multi Level Transfer 3 (Ternär, 1 Symbol / Bit, Keine Taktrückgewinnung, Keine Gleichstromfreiheit, Bandbreite ist $\frac{1}{4}$, wie NRZ)

Erkennung von Nutzdaten

- Coderegelverletzung
  - Idle $\rightarrow$ Basisbandimpulse
  - Präambel
  - SFD – Start Frame Delimiter
- Steuerzeichen
  - Blockcode zur Bereitstellung von Steuerzeichen
  - 4B5B-Code 8B10B-Code

1.4.4 Modulation

Modulation des tiefpass-gefilterten Basisbandsignals auf Trägersignal.

- 4ASK – Amplitude Shift Keying (4 Signalstufen ⇒ 2 Bit / Symbol, Amplitudenmodulation)
- QAM – Quadratur-Amplituden-Modulation (Mischung von Sinus- / Kosinussignalen, Doppelte Datenrate als Ursprungssignal, aber doppelte Bandbreite durch Modulation)

1.5 Übertragungsmedien

- Elektromagnetische Wellen (Optisch, $<$ ~10 GHz)
- Optische / Elektrische Leiter, Funk ⇒ Unterschiedliche Ausbreitungsgeschwindigkeiten

2 Sicherungsschicht

2.1 Probleme & Motivation

- Steuerung des Medienzugriffs
- Fehlerüberprüfung von Nachrichten
- Adressierung

2.2 Darstellung von Netzwerken als Graphen

Netztopologie darstellbar, als (un-)gerichtete Graphen

2.2.1 All-pair-shortest-distance-Problem

Lösbar durch Potenz der Distanzmatrix $D^n = D^{n+1} = D^*$. Beschränkt durch längsten einfachen Pfad im Netzwerk $n \leq N$, mit N ist Anzahl der Knoten.

2.2.2 Generierung von Baumstrukturen

- Shortest Path Tree $\rightarrow$ Bellman-Ford Algorithmus, Dijkstra-Algorithmus
- Minimum Spanning Tree

Vgl. Kapitel 5

2.3 Verbindungscharakterisierung, Mehrfachzugriff, Medienzugriffskontrolle

2.4 Verbindungscharakterisierung

- Übertragungsrate
- Übertragungsverzögerung
- Übertragungsrichtung
- Mehrfachzugriff (Multiplexing)

Übertragungsrate, Serialisierungszeit

$$t_s = \frac{L}{r} \tag{1}$$

- Übertragungsrate: r $[\frac{bit}{s}]$
- Serialisierungszeit (Serialzation Delay): t_s $[s]$
- Anzahl Datenbits: L $[bit]$

Ausbreitungsverzögerung

$$t_p = \frac{d}{\nu c} \tag{2}$$

- Distanz: d $[m]$
- Ausbreitungsverzögerung (Propagation Delay): t_p $[s]$
- Relative Ausbreitungsgeschwindigkeit: ν
- Lichtgeschwindigkeit: $c \approx 3 * 10^8 \frac{m}{s}$

Übertragungszeit (Delay)

$$t_d = t_s + t_p \tag{3}$$

Bandbreitenverzögerungsprodukt

$$C = t_p r = \frac{d}{\nu c} r \tag{4}$$

- Bandbreitenverzögerungsprodukt (Kapazität): $C[bit]$

Übertragungsrichtung

- Simplex
- Halbduplex
- Vollduplex

2.5 Mehrfachzugriff (Multiplexing)

Übertragung von Nachrichten mehrerer Teilnehmer über die selbe Leitung.

- Zeitmultiplex (Telefonnetz, Ethernet) – TDM
- Frequenzmultiplex (Funkübertragung, Radiosender) – FDM
- Raummultiplex (Kanalbündelung bei ISDN) – SDM
- Codemultiplex (UMTS) – CDM

2.6 Medienzugriffskontrolle

(durch Multiplexing) Problem bei synchronem TDMA

- Statische Aufteilung des Kanals
- i.A. Datenverker stoßartig

Daher, Asynchrones TDMA

- Zufälliger, Konkurierender oder dynamisch geregelter Medienzugriff

Random Access (ALOHA) / Carrier Sense Multiple Acces (SCMA)

- Zufällig senden $p_0 = \lambda e^{-2\lambda}$, λ ist Sendewahrscheinlichkeit
- Medium Abhören, Nur senden, wenn frei

CSMA/CD (Collision Detection)

- Erkenne Kollisionen, Widerhole Übertragung, wenn Kollision erkannt
- Übertragung gilt als erfolgreich, wenn keine Kollision erkannt

Mindestlänge der Nachrichten benötigt:

$$L_min = \frac{2dr}{\nu c} \tag{5}$$

Binary Exponential Backoff nach Kollision:

- Zufällige Wartezeiten
- Ereignisraum der Wartezeiten bei mehrfacher Wiederholung größer
- Maximal 16 Sendeversuche

CSMA/CA (Collision Avoidance) Basiert auf p-persistentem CSMA. Erweitert durch RTS/CTS

- Alle Knoten in Reichweite zu Basisstation
- Request to Send (RTS) an Basisstation
- Nur Übertragung, wenn Clear to Send (CTS) geantwortet

Token Passing

- Station in physikalischem Ring
- Token im Ring
- Station nimmt Token, darf senden
- Wenn Nachricht gelesen, als gelesen markiert, Sender nimmt vom Netz
- Problem, wenn Token verloren geht → Monitor Station

2.7 Rahmenbildung, Adressierung & Fehlererkennung

Nachrichten als Rahmen (Frame)

2.8 Erkennung von Rahmengrenzen

Längenangaben

- Am Anfang des Rahmens steht die Länge der Folgenden Nutzdaten
- Beginn der Nachricht muss eindeutig erkennbar sein

Steuerzeichen (IEEE 802.3u – FastEthernet)

- Bsp: 4B5B Code
- evtl. Escapen & Character Stuffing (Verdopplen des Escapezeichens) nötig

Begrenzungsfelder & Bit-Stopfen

- Markierung von Start und Ende einer Nachricht mit Bitfolge
- Bit Stuffing, um Endmarkierung in Nutzdaten zu verhindern

Coderegelverletzung (IEEE 802.3a/i – Ethernet)

- Von Übertragungsdaten unabhängige Signalwechsel
- Erzeugung von ungültigem Code durch Auslassen der Signalwechsel

2.8.1 Adressierung und Fehlererkennung

Adressierung

- Eindeutige Identification eines Knotens im Direktverbindungsnetz – MAC Adresse (Media Access Control)
- Mac Adresse besteht aus Organizatinal Unique Identifier und Device Identifier
- Broadcast Adresse zum Senden an Alle Knoten im Netz (ff:ff:ff:ff:ff:ff)

Fehlererkennung

- Checksumme, FCS (Frame Check Sequencer) zur Fehlererkennung – NICHT Korrektur
- CRC (Cyclic Redundancy Check) berechnet Checksumme fester Länge mit Hilfe eines Generatorpolynoms. (CRC-32 in Ethernet)
  - \+ 1 Bit Fehler
  - \+ isolierte 2 Bit Fehler
  - \+ Burst Fehler kürzer als $\deg(g(x))$
  - − Burst Fehler länger als $\deg(g(x))$
  - − Fehler aus mehreren Burst
  - − Fehler, die Vielfaches von $g(x)$ sind

$$g(x) = x^3 + x^2 + 1 \qquad m(x) = x^7 + x^5 + x^2 + 1 \tag{6}$$

$$g(x) = 1101 \qquad m(x) = 10100101 \tag{7}$$

$m'(x) = m(x) * x^{\deg(g(x))}$ (3 Nullen anhängen). Rest der Polynomdivision berechnen

$$m'(x)/g(x) = \ldots \qquad r(x) = 001 \tag{8}$$

$$s(x) = m'(x) \oplus r(x) \tag{9}$$

Empfänger prüft Nachricht:

$$r'(x) = s'(x)/g(x) \tag{10}$$

- $r'(x) \neq 0 \Rightarrow$ Fehler aufgetreten
- $r'(x) = 0 \Rightarrow$ Wahrscheinlich kein Fehler aufgetreten

2.9 Verbindung auf Schicht 1 & 2

2.9.1 Hub

- Links zu Bus zusammengefasst
- Rahmen erreicht alle Knoten
- $\Rightarrow$ Kollisionsdomäne / Segment

2.9.2 Brücke

- Verbindet zwei Kollisionsdomänen
- Lernphase: Senden an beide Ports
- Wenn Eintrag in Switching-Table: Nur Senden an einen Port
- Transition Bridges um verschiedene Zugriffsverfahren zu kopplen (WLAN Access Point – CSMA/CD CSMA/CA)
- Transparent (Hosts kennen Brücke nicht)
- Keine eigen Adresse benötigt
- Store-And-Forward (Überprüfung und Pufferung) vs. Cut-Through (Sofortige Serialisierung)

2.9.3 Switch

- „Brücke mit mehreren Ports“
- Nur ein Host pro Port, Vollduplex $\Rightarrow$ Kollisionsfreier Betrieb

2.10 Schleifen auf Schicht 2

- Schleifen auf Schicht 1: Kurzschluss
- Schleifen auf Schicht 2: Mehrere Kopien von Rahmen
- Spanning Tree Protocoll (STP): Deaktivierung redundanter Pfade $\rightarrow$ Schleifenfreiheit

3 Vermittlungsschicht

3.1 Motivation

- Kopplung unterschiedlicher Datenverbindungsnetze
- Aufteilung in Subnetze
- Logische, eindeutige adressierung von Geräten
- Wegwahl über mehrere Hops

3.2 Vermittlungsarten

3.2.1 Leitungsvermittlung

Reserviere Leitung zwischen Sender und Empfänger – Interneteinwahl (Letzte Meile)

$$T_{LV} = \frac{l}{r} + 4\frac{d}{\nu c} \tag{11}$$

\+ Schnelle Datenübertragung nach Verbindungsaufbau

\- Resourcenverschwendung, da exklusive Nutzung

\- Verbindungsaufbau komplex, aufwändig, lanwierig

Verbindungsaufbau

- Signalisierungsnachrichten zum Verbindungsaufbau
- Wegwahl

Datenaustausch

- Exklusive Kanalnutzung
- Weitgehender Verzicht auf Adressierung (Punkt-zu-Punkt Verbindung)

Verbindungsabbau

- Signalisierungsnachrichten zum Verbindungsabbau
- Freigabe der Ressourcen

3.2.2 Nachrichtenvermittlung

Wähle Weg für ganze Nachricht – Nur aus Sicht höherer Schichten verwendet

$$T_{LV} = (n+1)\frac{l+l_h}{r} + \frac{d}{\nu c} \tag{12}$$

- Voranstellen von Headerinformationen zur Adressierung

\+ Gemeinsame Nutzung von Teilstrecken (Zeitmultiplexing)

\+ Bessere Ausnutzung der Kanalkapazität

+ Kein Langwieriger Verbindungsaufbau

- Pufferung von Nachrichten bei Auslastung, Verlust bei begrenztem Puffer möglich (vgl. Kapitel 4)

- Mehrfache Serialisierung der ganzen Nachricht

3.2.3 Paketvermittlung

Teile Nachricht in Pakte; Verschicke diese unabhängig voneinander – In modernen Datennetzen

$$T_{PV} = \frac{1}{r}(\left\lceil \frac{l}{p_{max}} \right\rceil l_h + l) + \frac{d}{\nu c} + n\frac{p_{max} + l_h}{r} \tag{13}$$

+ Faire Nutzung von Engpässen

+ Nur Pufferung von Paketen notwendig

+ Bei Paketverlusst muss nur Paket wiederholhlt werden

- Verlust von Paketen durch begrenzten Puffer möglich

- Jedes Paket benötigt Header (Overhead)

- Empfänger muss Pakete zusammensetzen

3.3 Adressierung im Internet

3.3.1 IPv4

- 3 Byte Netzwerkidentifikator
- 1 Byte Computeridentifikator
- Weiterleitung auf Basis der IP-Adresse

3.3.2 IP-Header

- Version: 4 / 6
- IHL: IP Header Length – Länge des IP Headers als Vielfaches von 32Bit
- TOS: Types of Service – Klassifizierung / Priorisierung
- Total Length: Paketlänge, Header + Nutzdaten
- Identification: Zufälliger 16 Bit Wert
- Flags: Don't Fragment, More Fragments
- Fragment Offset: Position des Fragments im Paket
- TTL: Time To Live
- Protocol: Schicht 4 Protokol der Nutzdaten
- Header Checksum: Checksumme für den IP-Header

- Source Address: IP-Adresse des Senders
- Destination Address: IP-Adresse des Empfängers
- Options / Padding: z.B. Zeitstempel – evtl. Padding nötig

3.3.3 Adressauflösung

- ARP (Address Resolution Protocol)
  - ARP-Request: Who has 192.168.1.2? Tell 192.168.1.1 – Broadcast Message
  - ARP-Reply: 192.168.1.2 is at 04:0c:ce:e2:c8:2e – Unicast Message
- Paket gesendet an Default Gateway, wenn in anderem Adressbereich

3.3.4 Internet Control Message Protocol – ICMP

- Problembenachrichtigung bei Routingschleife, Kein Weg zum Zielnetz, MAC-Adresse nicht auflösbar
- Prüfen der Erreichbarkeit von Hosts – Ping
- Pakete Umleiten – Redirect

3.3.5 Adressklassen – Classfull Routing

- Aufteilung in Adressklassen
- Netzadresse ist 1. Adresse im Netzwerk
- Broadcastadresse ist letzte Adresse im Netzwerk

3.3.6 Subnetting – Classless Routing

192.168.0.0/24:

Netzadresse:	11000000 . 10101000 . 00000000 . 00000000	192.168.0.0
Broadcastadresse:	11000000 . 10101000 . 00000000 . 11111111	192.168.0.255
Subnetzmaske:	11111111 . 11111111 . 11111111 . 00000000	255.255.255.0

- 24 Bit Netzanteil
- 8 Bit Hostanteil
- $2^8 - 2 = 254$ nutzbare Adressen

Zusammenfassung von Netzen nur möglich, wenn größeres Subnetz entsprechende Netzadresse und Broadcastadresse enthält. Nur bei richtiger zweierpotenz möglich.

3.4 Wegwahl (Routing)

3.4.1 Statisches Routing

Routing-Table

- Netzadresse des Ziels
- Subnetzmaske
- Next-Hop (Gateway)
- Interface
- Kosten

Longest Prefix Matching um besten Weg auszuwählen

3.4.2 Dynamisches Routing

- Distanz-Vektor-Protokolle
  - Router kennt Next Hop + Kosten
  - Keine Information über Netzwerktopologie
  - Bellman-Ford Algorithmus
- Link-State-Protokolle
  - Genauer Zustand von Lniks
  - Genaue Informationen über Netzwerktopologie
  - Dijkstra Alogrithmus

3.4.3 Routing Information Protocol (RIP)

- Distanz-Vektor-Protokoll
- Metrik: Hop-Count (bis max 15)
- Updatenachricht mit Inhalt von Routingtable
- Router inkrementiert Metrik, vergleicht mit eigener Tabelle
- Bei 5maligem Ausbleiben der Updatenachricht wird Link aus Routingtable entfernt
- Count to Infinity bei Ausfall eines Links möglich
  - Split Horizon – Keine Route, die von A gelernt an A zurück (Nur Verbesserung, keine Lösung)
  - Poison Reverse – Route von A gelernt mit Metrik ∞ an A zurück (Nur Verbesserung, keine Lösung)
  - Path Vector – Vollständigen Pfad mitschicken (Lösung auf Kosten von Größe der Updatenachrichten)

3.4.4 Autonome Systeme

- Netzwerken unter einheitlicher administrativer Kontrolle
- i.A. Policy-Based Routing auf Ländereben, …

3.5 IPv6

- 128 Bit Adressraum – Hex-Schreibweise, 8 Gruppen zu 16 Bit
- Header fester Länge
- Extension Headers
- Keine Fragmentierung ⇒ MTU anpassung

Header:

- Version – 4 / 6
- Traffic Class – Priorisierung
- Flow-Label – Priorisierung, Datenstromzugehörigkeit
- Payload Lengt – in Vielfachen von 1 Byte
- Next Header – Extension Header / Datenheader
- Hop Limit – $\widehat{=}$ TTL
- Source Address
- Destination Address

4 Transportschicht

4.1 Motivation

4.1.1 Aufgaben

- Multiplexing (Segmentierung)
- Bereitstellung von Transportmechanismen
- Stau- & Flusskontrolle

4.2 Multiplexing

- Segmentierung des Datenstroms
- Segmentheader
  - Quellport
  - Zielport

⇒ Anwendungsidentifikation: (Protocol, SrcPort, DestPort, SrcIP, DstIp)

- Portnummern i.A. 16 Bit
- 5-Tupel zur Bereitstellung von Sockets
- Anwendung adressiert Socket mit File-Deskriptor

4.3 Verbindungslose Übertragung

4.3.1 Funktion

Transportpotokoll Header:

- Quell- & Zielport
- Längenangabe der Nutzdaten

⇒ Sendeanwendung adressiert via IP-Adresse, Protokoll & Zielport

4.3.2 Probleme

- ungesichert (Pakete können verloren gehen)
- verbindungslos (Reihenfolge der Segmente unabhängig)
- nachrichtenorientiert

4.3.3 User Datagram Protocol (UDP)

- ungesicherte / nachrichtenorientierte Verbindug
- Wenig Overhead

Header:

- Source Port
- Destination Port
- Lenght (in Byte)
- Checksum (optional)

Vorteile

- Wenig Overhead
- Keine Verzögerung durch Verbindungsaufbau
- Gut für Echtzeitanwendungen
- Keine Beeinflussung durch Fluss- & Staukontrollmechanismen

Nachteile

- Keine zugesicherte Dienstqualität
- Keine Ordnung von Segmenten
- Keine Möglichkeit zur Flusskontrolle
- Keine Staukontrollmechanismen

UDP-Chat

- select() auf verschiedene File Deskriptoren
- recvfrom() / sendto() statt read() / write()

4.4 Verbindungsorientierte Übertragung

Sequenznummer im Protokollheader

- Bestätigung
- Identifikation fehlender Segmente
- Erneutes Anfordern
- Zusammensetzen

⇒ Synchronisation (Sequenznummer, bestätigte Segmente) von Sender und Empfänger.

4.4.1 Verbindungsphasen

- Verbindungsaufbau (Handshake)
- Datenübertragung
- Verbindungsabbau (Teardown)

4.4.2 Sliding-Window Verfahren

Mehrere Segmente schicken, bevor Bestätigung angekommen

+ Effizientere Zeitnutzung
+ Flusskontrolle durch Empfänger (festgesetzte Fenstergröße)
+ Staukontrolle (Anpassung der Fenstergröße an Pfad)
- Mehr Zustände bei Sender/Empfänger
- Endliche Sequenznummern

Sende- w_s & Empfangsfenster w_r aus Sequenznummerraum S mit $|S| = N$

- Kommulative Bestätigung $ACK = m + 1$ bestätigt alle Segmente mit $SEQ \leq m$
- Forward Acknowledgement: Empfangenes Seqment $SEQ = n$ löst $ACK = n + 1$ aus.

Go-Back-N

- Akzeptiere nur nächste Sequenznummer
- Verwerfe alles andere
- Theoretisch $w_r = 1$ ausreichend
- $w_s \leq N - 1$, dass Segmente eindeutig
- Einfach, aber ineffizient

Selective Repeat

- Akzeptiere alle Sequenznummern in w_r
- Puffere Segmente, bis alle übertragen sind
- Degeneration zu Go-Back-N bei $w_r = 1$
- $w_s \leq \lfloor \frac{N}{2} \rfloor$, dass Segmente eindeutig

Allgemein

- Empfangspuffer nötig
- Fluss- & Staukontrolle durch dynamische w_s & w_r

4.4.3 Tranmission Control Protocol (TCP)

- gesicherte / Stromorientirete Übertragung
- Fluss- & Staukontrolle

Header:

- Quell- & Zielport (wie UDP)
- Sequenz- & Bestätigungsnummer (für einzelne Bytes)
- Offset – Headerlänge (in Vielfachen von 4 Byte)
- Reserved – nicht Verwendet (0)
- URG – Urgent Flag (selten verwendet) Daten bis „Urgent Pointer“ sofort an höhere Schichten weiterleiten
- ACK – Acknolegement (nächstes erwertetes Byte), auch Huckepack (piggy backing) möglich
- PSH – Push (Umgehen von Puffern in Interaktiven Anwendungen)
- RST – Reset (Abbruch)
- SYN – Synchronisation (Verbindungsaufbau)
- FIN – Finish (Verbindungsabbau)
- Receive Window – w_r in Byte
- Checksum – Über Header und Daten
- Urgent Pointer (selten verwendet)
- Options – z.B. Window Scaling, Maximum Segment Size (MSS)

MSS

- Maximale größe des TCP-Segments
- MTU sind Nutzdaten auf Schicht 2
- MSS sollte entsprechend kleiner sein, um Fragmentierung auf Schicht 2 zu vermeiden.
- Bsp: Ethernet MTU = 1500B, jeweils 20 B IP- / TCP-Header $\Rightarrow$ MSS = 1460 B

4.4.4 Fluss- & Staukontrolle

TCP-Flusskontrolle Überlastung beim Empfänger vermeiden $\rightarrow w_s$ entsprechend setzen.

- w_r in Feld Receive Window im TCP-Header schreiben
- w_s = Receive Window

TCP-Staukontrolle Überlastung im Netz vermeiden $\rightarrow w_s$ entsprechend setzen $\Rightarrow$ Staukontrollfenster w_c

- w_c++, wenn Daten verlustfrei übertragen
- w_c--, bei Verlusten
- $w_s = \min\{w_c, w_r\}$

TCP-Reno

- 3 Duplicate ACKs
  - Schwellenwert für Stauvermeidung $= \frac{w_c}{2}$
  - $w_c =$ Schwellenwert
  - Stauvermeidungsphase (Je ACK $w_c+ = 1 \rightarrow$ Verdoppelung pro RTT)
- Timeout
  - Schwellenwert $= \frac{w_2}{2}$
  - $w_c = 1$
  - neuer Slow-Start (Nur bei komplettem Frame: $w_c+ = 1 \rightarrow +1$ pro RTT)

$\Rightarrow$ Reduzierung der Datenrate bei Paketfehlern

- Interpretation von Paketfehlern als Überlastungssituation
- Niedere Schichten müssen Paketfehlerrate gering genug halten ($\leq 10^{-3}$)
- Evtl Bestätigungsverfahren auf Schicht 2

4.5 Network Address Translation (NAT)

- IP Adressen nicht unbedingt eindeutig (private IP Bereiche)
- NAT: Mapping von N privaten auf M öffentliche IP-Adressen
  - $N > M$: Port-Multiplexing ($M = 1$: privater DSL-Anschluss)
  - $N \leq M$: Statische / Dynamische Adresszuweisung
- Private Adressbereiche: $10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16, 169.254.0.0/16$ (für automatische Adressvergabe)

4.5.1 NAT-Tabelle

- Paket von zufälligem Port $[1024, 65535]$ über Router gesendet
- Router tauscht Absenderadresse gegen eigene öffentliche IP aus, tauscht Quellport aus, wenn Kollision
- Router erstellt Eintrag in NAT-Tabelle mit Local IP, Local Port, Global Port
- Bei Antwort von Server wird Adressübersetzung rückgängig gemacht

4.5.2 NAT-Varianten

- Full Cone Nat: Protokoll zusätzlich gespeichert. Keine Prüfung von Absender IP / Port
- Port Restricted NAT
- Address Restritced NAT
- Port and Address Restricted NAT
- Symmetric NAT

4.5.3 Anmerkungen

- Verhinderung von eingehenden Verbindungen ohne vorherigen Verbindungsaufbau von lokaler Seite
- Beschränkung der NAT Tabelle auf theoretisch 2^{16} Einträge (Full Cone NAT) pro Protokoll & globaler IP Adresse. Praktisch durch Routerkapazität
- Einträge werden implementierungsabhängig wieder gelöscht
- Einträge in NAT Tabelle von Hand ist Port Forwarding

4.5.4 NAT und ICMP

- Nutzung der ICMP-ID statt Portnummer
- Rückübersetzung bei TTL-Exceeded (originaler Header mitgeliefert) nötig

5 Sitzungsschicht

- Connection-Oriented Mode: Verbindung über Datentransfer hinweg (Session $\neq$ Verbindung auf Transportschicht)
- Connectionless Mode: Weiterreichen der Daten + Adressierung an Transportschicht

5.1 Dienste der Sitzungsschicht

- Session: Kommunikation zwischen zwei oder mehr TN, mit definiertem Anfang und Ende
- Connection-Oriented Mode
  - Auf- / Abbau von Sessions
  - Datentransfer
  - Teilnehmerkoordination
  - (Re-)Synchronisation
  - Fehlermeldungen / Aktivitätsmanagement
  - Erhaltung / Wiederaufnahme von Sessions
- Connectionless Mode
  - Datentransfer

5.2 Funktionseinheiten

- Kernel – Basisfunktionen
- Halb-Duplex – Abwechselnd Senden / Empfangen
- Duplex – Gleichzeitig Senden / Empfangen
- Negotiated Release – Session sauber Beenden
- Expedit Data – Beschleunigter Datentransfer
- Activity Management – Logische Strukturierung
- Major Synchronization – Strukturierung der Sessions in Dialogeinheiten
- Minor Synchronization – Strukturierung der Dialogeinheiten

5.2.1 Kombination von Funktionseinheiten

- Aktivitäten bestehen aus Dialogeinheiten (evtl. über Sessions hinweg)
- (Re-)Synchronisation von Sessions oder Aktivitäten an Synchronisationspunkten
- Kombinationen (Aushandlung vor Beginn der Session)
  - BCS – Basic Combined Subset: Kernel, Halb-Duplex / Duplex

  - BSS – Basic Synchronized Subset: Kernel, Halb-Duplex, Negotiated Release, Minor/Major Synchronize, Resynchronzie
  - BAS – Basic Activity Subset: Kernel, Halb-Duplex, Minor Synchronize, Exception, Activity Management

5.3 Synchronisation

- Major Sychronisationspunkt: Strukturierung in Serie von Dialogeinheiten. Explizite Bestätigung
- Minor Synchronisationspunkt: Strukturierung innerhalb von Dialogeinheiten. Bestätigung möglich. Vorherige Daten werden bei Resynchronisation nicht verworfen
- Tokens zur Steuerung der Kommunikation
  - Data Token – Senden bei Halb-Duplex
  - Release Token – Sitzungsbeendigung
  - Synchronize-Minor Token
  - Activity-Major Token

5.4 Quality of Service

- Service Parameter
  - Protection – Schutz gegen Monitoring, Manipulation
  - Priorität – Ressourcenaufteilung zwischen Session
  - Resilience – Fehlerwahrscheinlichkeit
- Performance Parameter
  - Establishment / Release Delay und Establishment / Release Failure Probability
  - Durchsatz und Transit Delay
  - Residual Error Rate und Transfer Failure Probability

$$RER = \frac{S_e + S_l + S_x}{S}$$

    * S_e: Falsch übertragene Daten
    * S_l: Verloren gegangene Daten
    * S_x: Dupliziert empfangene Daten
    * S: Gesendete Daten

6 Darstellungsschicht

- Einheitliche Interpretation der Daten für Teilnehmer
- Darstellung der Daten (Syntax) – KEINE Semantik
- Datenstrukturen zur Übertragung
- Aktionen der Datenstrukturen
- Datentransformationen

6.1 Aufgaben

- Datenkompression (Entfernung von Redundanz) und Quellencodierung
- Umkodierung – Übersetzung zwischen Datenformaten (Darstellungen)
- Strukturierte Darstellung
- Ver- / Entschlüsselung

6.2 Datenkompression und Umkodierung

- Kodierungsverfahren
  - Fixed-Length Code (Uniformer Code) – USCII, UNICODE
  - Variable-Length Code – Huffman Code
- Kompressionsverfahren
  - Verlustfreie Komprimierung (Lossless Data Compression) – ZIP
  - Verlustbehaftete Komprimierung (Lossy Data Compression) – JPEG

6.2.1 Huffman-Code

- Häufigere Buchstaben bekommen kürzere Code-Wörter → Baum
- Verlustfrei
- Statischer Huffan Code – Verteilung der Zeichen muss Erwartungen entsprechen
- Dynamischer Huffman Code – Übermittlung des Codebuchs notwendig
- Optimaler Code – Mimierung der mittleren Codewortlänge:

$$\sum_{i \in \mathcal{A}} p(I)|c(i)|$$

  - $p(i)$: Auftrittswahrscheinlichkeit von $i \in \mathcal{A}$
  - $c(i)$: Länge des Codeworts
- Präfixfrei – Gültiges Codewort ist niemals Präfix eines anderen Codeworts
- Entropy-Encoding Verfahren
- Varible-Length Code

6.3 Einheitliche Syntax

- Abstrakte Syntax definiert durch Anwendungsprozesse, kodiert als Lokale Syntax
- Transfersyntax zum Datenaustausch auf Darstellungsebene
- Kodierregeln zur Umwandlung von Abstrakter Syntax in Konkrete / Transfersyntax
- Presentation Context – Übergangsregeln für Kodierung

6.3.1 Abstrakte Syntaxnotation Nummer 1: ASN.1

- Backus-Naur-Notation – Beschreibung einer formalen Sprache
  - Syntaktische Variablen. Nachtterminale mit <>
  - Zuweisung: ::==
  - Terminalsymbole
  - Operatoren
  - Verknüpfung durch Verkettung, Klammerung (), Auswahl |, Wiederholung *, Optionale Ausdrücke []
- Semantiksprache

Elemente in ASN.1

- Gliederung in Module
- Ex- / Importieren von Modulen
- Types
- Values
- Macros

6.3.2 Basic Encoding Rules (BER)

- Konkrete Transfersyntax zur Darstellung der ASN.1 Datentypen – z.B. SNMP
- Typfeld (Universal, Application, Context-Specific, Private)
- P/C-Bit (Primitiver / Konstruierter Typ)
- Subtype (End of Content, Boolean, Integer, …)
- Längenfeld
  - Kurzform: 0-126 (127 Byte)
  - Langform: Most Significant Bit auf 1, Danach Anzahl der Oktette ($2,47 \cdot 10^{303}$ Byte)

  - Unbestimmte Form: Delimiter zeigt Ende an – Senden kann beginnen, bevor Länge fest steht

- Weitere Transfersyntax: Packed Encoding Rules (PER) – Nachfolger von BER

7 Anwendungsschicht

- Schnittstelle zwischen Anwendung und Netzwerkprotokollen
- Bsp: DNS, HTTP, SMTP, SSH

7.1 Domain Name System (DNS)

- Hierachisch aufgebauter Name
- Fully Qualified Domain Name (FQDN)
  - Hostname
  - Suffix
    - Hierachisch
    - Von Rechts nach Links, Bei root (.) beginnend
- Pro Zone: Primärer DNS-Server + Sekundäre Server (redundanz)
  - Liste aller Hosts dieser Zone
  - Liste mit autoritativen Servern untergeordneter Domänen
  - Autoritativ für diese Zone
- Gecachte Werte auf anderen Server sind nicht autoritativ

7.1.1 Zonendatei

- A – Hosteintrag IPv4
- AAAA – Hosteintrag IPv6
- NS – Nameserver
- CNAME – Common Names
- MX – Mail Exchanger

7.1.2 Rekursive Namensauflösung

- Request an DNS-Server
- Nicht bekannt, wie DNS-Server Request auflöst
- DNS-Server schickt komplette Antwort zurück

7.1.3 Iterative Namensauflösung

- Sofern nicht (Teile) gecached: Iterative Auflösung des FQDN, bei Wurzel beginnend
- Root Hints (Adressen der DNS-Rootserver) hat jeder DNS-Server gespeichert

8 Verteilte Systeme

8.1 Motivation

- Skalierbarkeit
- Ausfallsicherheit
- Heterogenität

⇒ Standardisierte Kommunikationsprotokolle

8.2 Homogene, skalierbare Paradigmen

8.2.1 Message Passing Interface (MPI)

- Abbildung einer Datenstruktur auf Netztopologie
- Standardisiertes Interface
- Nachrichtenorientierte Kommunikation (Point-to-Point, Broadcast, scatter, gather)
- Kommunikation blockiert Programm
- Kein Load-Balancing, Wenig Fehlertoleranz

8.2.2 MpaReduce

- Abbildung – Mapping $K_1 \times V_1 \to (K_2 \times V_2)^*$
- Gruppierung (automatisch) – Group $(K_2 \times V_2)^* \to (K_2 \times (V_2)^*)^*$
- Reduktion – Reduce $K_2 \times (V_2)^* \to (V_2)^*$
- Automatische Parallellisierung, Lastverteilung, Fehlertoleranz
- Überlappung von Map und Reduce Prozessen möglich
- Nur Möglich, wenn Abbildung auf Map/Reduce Prozesse möglich

8.2.3 Pipes

- Inter-Prozess-Kommunikation via File-Handle (stdin, stdout, stderr)

DUP

- Graph, über Pipes verbunden
- DUP-Dämon verwaltet Pipes über TCP-Verbindungen (Name @ Host [Umleitungen] $ Shell-Befehl;)
- Prozess-Graph statisch, keine Anpassung; Dafür einfache Integration von Konsolenprogrammen

8.3 Remote Procedure Call

- Funktionsaufruf über Prozessgrenzen hinweg
- Client-/Server-Modell
- Übergabe per Stack / Register schwierig, wegen inkompatiblen kodierungen
- Stub einer RPC Funktion auf Server- & Client-Seite – Konvertierung ((Un)-Marshalling) der Daten
- Interface Description Language (IDL) – Sprachunabhängige Definition von Funktionen / Datentypen
- Binding – rpcbind (Port 111) Dynamic Binding des Servers zur Laufzeit
- Java RMI – Remote Method Invocation (RPC für Java)
- Heterogene Systeme möglich
- Marshalling teuer
- Keine Fehlertoleranz / Lastverteilung

8.4 Shared Memory

- Arbeitsspeicher von mehreren Prozessen gleichzeitig verwendet → Verteilte Anwendung arbeitet wie Multi-Threaded-Anwendung

8.4.1 Non-Uniform Memory Access (NUMA)

- Direkter Zugriff auf Speicher anderer Prozessoren
- Single System Image – Ein großer Arbeitsspeicher
- Einfache Parallellisierung
- Latenz der Arbeitsspeicher

8.4.2 Distributed Shared Memory über Paging

- Seitenaustausch zwischen Hauptspeichern des Clusters
- False-Sharing, wegen Anforderung ganzer Seiten zweier Knoten

8.4.3 Distributed Software Transactional Memory

- Verteilter STM zur Zurückrollung von Transaktionen
- Innerhalb von Transaktionen nur Operationen ohne direkte Wirkung nach Außen
- Konfliktüberprüfung schwierig (Single-Server, Konsens-Protokoll)

8.5 Einbettung in Programmiersprachen – Erlang

- Actor Model – Event queue (Synchronisationspunkt) + event handler
- Verteilte Fehlerbehandlung über spezielle Nachrichten

9 Kyptografie

9.1 Ziele kryptografischer Verfahren

- Integrität – Daten nicht unterwegs verändert
- Authentizität – Daten stammen von Alice
- Vertraulichkeit – Eve soll nicht mitlesen
- Verbindlichkeit / Nichtabstreitbarkeit – Alice soll nicht abstreiten können, die Nachricht geschrieben zu haben

⇒ Verschlüsselung + Hashverfahern + Schlüsselaustauschprotokolle + Sitzungsprotokolle

9.2 Klassifizierung von Verschlüsselungsverfahren

- Symmetrische Verfahren – Shared Secret
- Asymetrische Verfahren – Private + Public Key

9.3 Symmetrische Verfahren: RC4

- Shared → Cipher Stream (pseudozufälliger Bitstrom)
- Cipher Stream XOR Nachricht → Cipher Text
- Cipher Text XOR Cipher Stream → Klartext Nachricht

9.3.1 Cipher Stream

- Substitution-Box mit 256 Byte
- Permutieren von zwei Werten, Summe ist modulo 256 ist Index zu einem neuen Wert. Dieser wird als Zufallszahl zurück gegeben

9.3.2 Zu beachten

- Erste Bytes des Cipher-Stream lassen Rückschlüsse auf Schlüssel zu
- Nie zwei Nachrichten mit selbem Schlüsselstrom verschlüsseln, ergäbe neuen Angriffsvektor
- Cipher-Stream wiederholt sich irgendwann
- RC4 heute nicht mehr sicher, da Korrelation zwischen Cipher-Stream und Schlüssel

9.3.3 Schlüsselaustausch: Diffie-Helmann

- Primzahl p, Primitive Kongruenzwurzel g
- Unabhängige Zufallszahlen a, b
- Austausch von $B = g^p \mod p$, $A = g^a \mod p$
- $K = A^b \mod p = B^a \mod p$

9.3.4 Sicherheit

- Anfälligkeit gegenüber Man-in-the-Middle Angriffen

9.4 Transport Layer Security (TLS)

- Setzt auf TCP auf
- Handshake Protocol – Aushandeln der Verschlüsselung und Kompression, Authentifizierung
- Record Protocol – Ver- / Entschlüsselung und Kompression
- Arbeitet auf Sitzungsschicht – Verwaltung von Sitzungsinformationen
  - Session ID
  - Peer Certificate
  - Cypher Suite
  - Compression Method
  - ClientHello.random und ServerHello.random
  - Premaster-Secret und daraus abgeleitete Schlüssel
- Arbeitet auf Darstellungsschicht – Ver- / Entschlüsselung, Kompression

9.4.1 Dienste von TLS

- Authentifizierung – Durch Zertifikate (Bestätigung durch dritte Partei, dass öffentlicher Schlüssel dem Kommunikationspartner gehört), Entweder Server gegenüber Client, oder Gegenseitig
- Vertraulichkeit – Symmetrisches Verschlüsselungsverfahren
- Integrität – Hashfunktion mit Shared Secret

9.4.2 TLS Handshake Protocol

- Client Hello – Mögliche Sitzungsinformationen + Seed an Server
- Server Hello – Gewählte Sitzungsinformationen + Seed an Client
- Server Certificate – Öffentlichen Schlüssel an Client
- Server Hello Done – Ende des Serverseitigen Handshakes
- Client Key Exchange – Verschlüsseltes Premaster Secret an Server
- Change Cipher Spec – Letzte unverschlüsselte Nachricht an Server
- Handshake Finished – Erste verschlüsselte Nachricht an Server
- Change Cipher Spec – Letzte unverschlüsselte Nachricht an Client
- Handshake Finished – Erste verschlüsselte Nachricht an Client

Anmerkungen

- Außer Simple TLS Handshake auch noch Client-Authenticated TLS Handshake möglich
- Mit Session ID: Resumed TLS Handshake möglich