Kommunikation

December 12, 2016

Contents

OSI-Schicht		Einordnung	Standard	DoD-Schicht	Einordnung	Protokollbeispiel	Einheiten	Kopplungselemente
7	Anwendung (Application)	Anwendungs-orientiert	FTAM	Anwendung	Ende zu Ende (Multihop)	HTTP FTP HTTPS SMTP LDAP NCP	Daten	Layer 4-7 Switch, Content-Switch, Gateway
6	Darstellung (Presentation)		ASN.1					
5	Sitzung (Session)		ISO 8326					
4	Transport (Transport)	Transport-orientiert	ISO 8073	Transport		TCP UDP SCTP SPX	Segmente	
3	Vermittlung (Network)		CLNP	Internet	Punkt zu Punkt	ICMP IGMP IP IPX	Pakete	Router, Layer-3 Switch
2	Sicherung (Data Link)		HDLC	Netzzugang		Ethernet Token Ring FDDI ARCNET	Rahmen (Frames)	Switch, Bridge
1	Bitübertragung (Physical)		Token Bus				Bits	Hub, Repeater

Figure 1: OSI Referenzmodell

0.1 OSI-Referenzmodell

	DE	EN	Beschreibung
7	Anwendung	Application	
6	Darstellung	Presentation	
5	Sitzung	Session	
4	Transport	Transport	
3	Vermittlung	Network	
2	Sicherung	Data Link	
1	Bitübertragung	Physical	

Table 1: OSI-Referenzmodell

0.2 Internet-Referenzmodell

Name	OSI-Schicht
Anwendung	5,6,7
Transport	4
Internet	3
Netzzugang	1,2

Table 2: OSI-Referenzmodell

0.3 Dateneinheiten im Schichtenmodell

(N)-Interface Data Unit. Die (N)-IDU ist die Dateneinheit, die die (N+1)-Schicht an die (N)-Schicht über den Dienstzugangspunkt übergibt. Sie besteht aus der (N)-SDU und der (N)-ICI.

(N)-Service Data Unit. Die (N)-SDU entspricht den Daten der (N+1)-Schicht, die transparent weitergegeben werden. Sie entspricht der (N+1)-PDU.

(N)-Interface Control Information. Die (N)-ICI enthält Information über den auszuführenden Dienst, z.B. Parameter des Dienstes oder Länge der (N)-SDU. Sie wird in der (N)-Schicht abgetrennt und verworfen.

(N)-Protocol Control Information. Die (N)-PCI enthält Information, die zwischen (N)-Protokollinstanzen ausgetauscht wird, um den Protokollablauf zu steuern.

(N)-Protocol Data Unit. In der (N)-Schicht werden (N)-SDU und (N)-PCI zur (N)-PDU zusammengefügt.

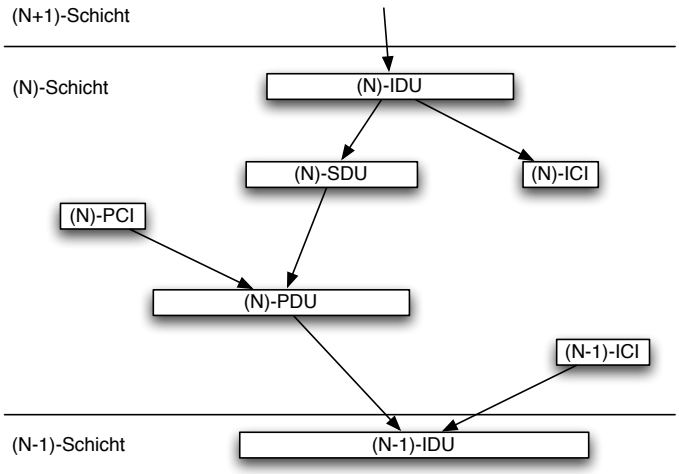

Figure 2: Dateneinheiten im Schichtenmodell

1 Bitübertragungsschicht (Physical Layer)

1.1 Modulation

Manchestercodierung. Eine Signalpegeländerung in der Intervallmitte codiert

$$1 \qquad High \rightarrow Low$$

$$0 \qquad Low \rightarrow High$$

Amplitudenmodulation. Bits werden durch unterschiedliche Amplitude einer Schwingung moduliert.

Frequenzmodulation. Bits werden durch unterschiedliche Frequenzen einer Schwingung moduliert.

Phasenmodulation. Bits werden durch eine Phasendrehung der Schwingung um 180 Grad unterschieden.

1.2 Betriebsarten

Simplex Kommunikation nur in eine Richtung, z.B. Telex, Feuermelder

Duplex Kommunikation gleichzeitig in beide Richtungen, z.B. Telefon

Halbduplex Kommunikationsrichtung abwechselnd, z.B. Walkie-Talkie

1.3 Kanalkapazität

Bandbreite.

$$B = f_{max} - f_{min} \quad [Hz]$$

Schrittgeschwindigkeit (Baudrate). Rate der Signalparameter-Zustandswechsel.

$$[baud = \frac{1}{s}]$$

Bei isochronen Digitalsignalen ist $s_{max} = \frac{1}{T}$ mit Schrittdauer T.

$$s_{max} = 2 \cdot B$$

Signal-Rausch-Abstand (Signal-Noise-Ratio) ist das Verhältnis von Signalstärke zur Stärke des Rauschens.

$$SNR = 10\ log_{10}\ \frac{P_{signal}}{P_{noise}} \qquad [dB]$$

Bei gegebener Kapazität und Bandbreite ist

$$SNR = 10\ log_{10}\ (2^{\frac{C}{B}} - 1)$$

Übertragungsgeschwindigkeit (Bitrate, Datenrate).

$$[\frac{bit}{s}]$$

Übertragungsgeschwindigkeit entspricht Schrittgeschwindigkeit wenn jeder Schritt ein Bit darstellt.

Kanalkapazität ist die maximale Datenrate. Ein idealer Kanal ohne Rauschen kann eine beliebige Datenrate durch die Anzahl der Signalstufen M realisieren.

$$C = s_{max} \cdot ld(M)$$

Mit Rauschen ist die maximale Datenrate nach Nyquist

$$C = B \cdot ld(1 + \frac{P_{signal}}{P_{noise}})$$

Signalstufen

$$C = s_{max}\ ld(M) \quad \Rightarrow \quad M = 2^{\frac{C}{s_{max}}}$$

1.4 Leitungskapazität, Laufzeiten

Sendezeit ist die Zeit, die benötigt wird, um die Daten auf das Medium zu legen. Es gilt

$$T_s = \frac{L}{d}$$

mit Datenvolumen L und Datenübertragungsrate d.

Ausbreitungszeit ist die Laufzeit der Signale über das Medium von Sender zu Empfänger. Es gilt

$$T_a = \frac{l}{v}$$

Übertragungsdauer

$$T_{ges} = T_s + T_a$$

Ausbreitungsgeschwindigkeit In üblichen Medien (Kabel, Glasfaser) ist

$$v \approx \frac{2}{3}\, c \approx 200\ 000 \frac{km}{s}$$

Bandbreiten-Verzögerungs-Produkt ist das Datenvolumen, das sich während der Übertragung auf dem Medium befindet.

$$L = d \cdot T_a$$

Auslastung des Mediums ist das Verhältnis von tatsächlicher zu möglicher Datenrate.

$$U = \frac{d}{d_{max}}$$

2 Sicherungsschicht (Data Link Layer)

Aufgaben der Sicherungsschicht.

- Fehlererkennung, z.B. durch CRC
- Fehlerbehebung, z.B. durch Quittungen und Sendewiederholungen
- Strukturierung des Datenstroms, z.B. durch Sequenznummern
- Medienzugangskontrolle, z.B. durch CSMA/CD

Unterteilung. Die Sicherungsschicht kann in zwei Unterbereiche eingeteilt werden:

Logical Link Control Sicherung der Punkt-zu-Punkt-Verbindung gegen Verfälschung, Verlust und Reihenfolgenvertauschung; Flusskontrolle, Strukturierung der Übertragung

Medium Access Control Zugangskontrolle für ein geteiltes Medium

2.1 Fehlererkennung

Paketfehlertypen. Ein zuverlässiger Dienst vermeidet folgende Fehlertypen:

- Verlust
- Duplizierung
- Vertauschung der Reihenfolge
- Phantom-Dateneinheiten

Paritätsbit. Mit einem Paritätsbit kann man eine ungerade Anzahl von Fehlern erkennen.

Frame Check Sequence bezeichnet eine Prüfsumme, die zu einem Datenblock (Frame) hinzugefügt wird, um Fehlererkennung oder Fehlerkorrektur zu ermöglichen.

Cyclic Redundancy Check Hierbei wird die Datensequenz als Polynom betrachtet und ein Generatorpolynom festgelegt, das Sender und Empfänger bekannt ist. Die Datensequenz wird vor dem Senden durch das Generatorpolynom divivdiert. Der Rest der Division wird an die Datensequenz (als *FCS*) angehängt und beides wird übermittelt. Der Empfägner führt wiederum die Polynomdivision durch G durch. Falls hierbei kein Rest bleibt, wurde die Sequenz entweder fehlerfrei übertragen, oder die Sequenz ist fehlerhaft, hat aber weiterhin G als Faktor, was sehr unwahrscheinlich ist.

Gegeben sei die Datensequenz M und das Generatorpolynom G.

1. Multipliziere M mit 2^d, wobei d der Grad von G ist. Das entspricht dem Anhängen von d Nullen an M.
$$M' = M \cdot 2^d$$
2. Dividiere M' modulo durch G.
$$C = M' \ mod \ G$$
3. Füge C an die Datensequenz an.
$$\tilde{M} = M' + C$$
4. Übertrage $\tilde{M}$
5. Dividiere die empfangene Sequenz modulo durch das Generatorpolynom.
$$C' = \tilde{M} \ mod \ G$$
Wenn $C' = 0$, dann war die Übertragung mit hoher Wahrscheinlichkeit fehlerfrei.

CRC: Implementierung als Schieberegister

2.2 Flusskontrolle

Ziel. Flusskontrolle dient der Anpassung der Übertragungsrate an die Verarbeitungskapazitäten des Empfängers.

Anzahl der Sequenznummern mit Sliding Window. Gegeben seien die Datenrate des Mediums d , die Größe einer Dateneinheit S, sowie die Lebenszeit einer Dateneinheit T_l. Eine Sendewiederholung findet maximal nach T_w statt und der Empfänger quittiert den Erhalt der Dateneinheit nach maximal T_q. Dann ist eine untere Schranke für die benötigte Bitanzahl n gegeben durch
$$2^n \ Dateneinheiten \geq (2T_l + T_w + T_q) \cdot D$$
wobei $D = \frac{d}{S}$ die Dateneinheiten-Rate auf dem Medium ist.

Implizite Flusskontrolle bezeichnet die Drosselung der Datenrate durch das Zurückhalten der Quittungen. Dieses Verfahren ist problematisch: Wartet der Sender zu lange auf eine Quittung, so wird nach dem Timeout die Dateneinheit erneut gesendet, was die Resourcen von Sender und Empfänger sowie die Kapazität des Mediums beansprucht.

Automatic Repeat Request steht für die automatische Wiederholung verlorengegangener oder beschädigter Dateneinheiten.

2.2.1 Stop-and-Wait

Stop-and-Wait ist ein einfaches ARQ-Verfahren. Dabei wartet der Sender auf die Quittung zu einer gesendeten Dateneinheit, bevor er die nächste verschickt. Falls vor dem Timeout keine Quittung eintrifft, wird die Dateneinheit wiederholt.

Übertragunszeit bei Stop-and-Wait. Unter der Annahme, dass Verarbeitungsdauer und Größe der Quittung vernachlässigbar sind, erreicht die Quittung den Sender nach $T_s + 2 \cdot T_a$. Damit ist die gesamte Übertragungszeit für n Dateneinheiten

$$T_{ges} = (T_s + 2 \cdot T_a) \cdot n$$

Auslastung bei Stop-and-Wait. Die Auslastung des Mediums beim Stop-and-Wait-Verfahren ist das Verhältnis von Sendezeit zur Summe aus Sende- und Wartezeit:

$$U = \frac{T_s}{T_s + 2T_a}$$

Leistungsfähigkeit im Fehlerfall. Gegeben sei die Fehlerwahrscheinlichkeit p.

$$U = \frac{1-p}{2a+1}$$

2.2.2 Go-Back-N

Prinzip. Der Sender kann eine gewisse Anzahl von Dateneinheiten (Fenstergröße W) senden, bevor er eine Quittung empfangen muss. Nach dem Senden der Dateneinheiten $D_i, \ldots, D_{i+W-1}$ wartet er auf das Eintreffen einer Quittung. Sobald die Quittung ACK_i eintrifft, schickt er D_{i+W}. Hat der Sender nach Ablauf des Timers keine Quittung für D_i erhalten, wiederholt er alle Dateneinheiten $D_i, \ldots, D_{i+W-1}$.

Beim Erhalt einer erwarteten Dateneinheit D_i schickt der Empfägner die Quittung ACK_i. Erhält er eine Dateneinheit D_j mit $j > i+1$, so wird sie verworfen und die Quittung ACK_i wiederholt, solange bis D_{i+1} eintrifft, die dann mit ACK_{i+1} quittiert wird.

Auslastung in Abhängigkeit von der Fenstergröße. Das Fenster kann so groß sein, dass der Sender kontinuierlich senden kann, oder er muss nach Ausnutzung des Fensters auf die Quittung warten. Im ersten Fall wird die Kapazität des Mediums voll ausgenutzt.

$$U(W,a) = \begin{cases} 1 & \text{wenn } W \geq 2a+1 \\ \frac{W}{2a+1} & \text{wenn } W < 2a+1 \end{cases}$$

$$a = \frac{T_a}{T_s}$$

2.3 Medienzugangskontrolle

2.3.1 Medienzugangsverfahren

Raummultiplex Einteilung des Raumes in Sektoren durch gerichtete Antennen, dedizierte Leitungen etc; Beispiel: Zellenstruktur in Mobilfunknetzen

Zeitmultiplex Kanal belegt gesamten Frequenzraum für eine gewisse Zeit; Beispiel: Ethernet

Frequenzmultiplex Einteilung der verfügbaren Bandbreite in Frequenzabschnitte; Beispiel: UKW-Radio

Codemultiplex Kanäle operieren gleichzeitig auf derselben Frequenz; durch Verknüpfung des Signals mit einem eindeutigen Code können die Kanäle vom Empfänger wieder getrennt werden

2.3.2 CSMA/CD (Carrier Sense Multiple Access with Collision Detection)

CSMA/CD ist ein Zeitmultiplex-Verfahren bei konkurrierendem Zugriff auf das Medium.

Ablauf.

1. *Listen before Talk*: Endsystem prüft, ob das Medium frei ist
2. Falls Medium frei beginnt das Endsystem zu senden
3. *Listen while Talk*: Zur Erkennung von Kollisionen hört der Sender während des Sendens das Medium ab
4. im Kollisionsfall: Sendeunterbrechung, Jamming-Signal
5. Wiederholung der Sendung, geregelt durch Backoff-Algorithmus

Mindestlänge der Dateneinheit. Eine Voraussetzung für CSMA/CD ist, dass das Senden der Dateneinheit nach der zweifachen Signallaufzeit noch nicht beendet sein darf. Daraus ergibt sich eine Mindestlänge für die Dateneinheit, die durch das Padding-Feld der Dateneinheiten sichergestellt wird.

Abkürzung	Länge [bit]	Beschreibung
PR	56	Präambel zur Synchronisation
SD	8	Start-of-Frame Delimiter zeigt Beginn an
DA	16/48	Destination Address, MAC-Adresse des Ziels
SA	16/48	Source Address, MAC-Adresse der Quelle
Length	16	Anzahl der Oktette im Datenfeld
Data	≤ 12000	Datenfeld, max. 1500 Oktette
PAD	optional	Padding um auf nötige Mindestlänge zu ergänzen
FCS	32	Frame Check Sequence, CRC-32

Table 3: Felder einer CSMA/CD-Dateneinheit

CSMA/CD-Dateneinheit

2.4 Lokale Netze

2.5 Ethernet

Topologie. Beim Ethernet sind die Endsysteme in Sterntopologie verbunden.

2.6 Token Ring

Topologie. Beim Token Ring sind die Endsysteme jeweils Punkt-zu-Punkt zu einem Ring verbunden.

Ablauf. Token Ring ermöglicht kontrollierten Zugriff auf das Medium mittels eines zirkulierenden Senderechts (Token). Empfängt ein Endsystem das Token, so hat es für eine bestimmte Zeit das Recht zu senden (Token Holding Time). Die gesendeten Daten kommen nach dem Umlauf im Ring wieder beim Sender an, der diese wieder vom Ring nimmt. Danach gibt er das Token an das nachfolgende System weiter.

Echtzeit. Token Ring ist für Echtzeitanwendungen geeignet, da eine maximale Wartezeit zwischen Sendewunsch und Anfang des Sendens garantiert werden kann:

$$Wartezeit_{max} = (n-1) \cdot Sendezeit(Datenmenge_{max}) + Tokenumlaufzeit$$

Quittierung. Ein Endsystem im Token Ring kann die Dateneinheit ändern, bevor es diese weitergibt. Der Empfänger kann also den Erhalt einer Dateneinheit quittieren, indem er z.B. ein bestimmtes Bit setzt. Der Sender erkennt diese Quittung während er die Dateneinheit vom Ring nimmt.

Aktiver Anschluss ans Medium. Beim Token Ring sind die Endsysteme aktiv ans Medium angeschlossen. Jedes System regeneriert die erhaltene Dateneinheit und schickt sie weiter, agiert somit als Verstärker. Das ermöglicht größere Leitungslänge. Die Regenerierung ist allerdings mit Verzögerung verbunden. Für den Anschluss neuer Systeme an das Medium muss das Medium unterbrochen werden.

2.7 High-Level Data Link Control

Phasen.

1. Verbindungsaufbau
2. Datentransfer
3. Verbindungsabbau

Flag. Die eindeutige Bitsequenz

`01111110`

wird verwendet, um Anfang und Ende einer Dateneinheit zu signalisieren. Ein Autreten des Flags in den Nutzdaten wird durch *Bitstopfen* verhindert. Hierbei wird nach 5 aufeinanderfolgenden Einsen eine Null eingefügt, die vom Sender wieder entfernt wird.

Codetransparenz bedeutet, dass es für eine Anwendung transparent bleibt, welche Mechanismen und Codes zur Übertragung verwendet werden. So wird z.B. durch Bitstopfen auf der Sicherungsschicht erreicht, dass die Anwendung die Existenz einer ausgezeichneten Bitfolge als Anfangs- und Endflag bei der Codierung ihrer Daten nicht berücksichtigen muss.

Piggyback Acknowledgement verwendet für das Senden einer Bestätigung keine separate Dateneinheit, sondern benutzt eine Dateneinheit mit, die selber Nutzdaten enthält.

3 Vermittlungsschicht (Network Layer)

3.1 Vermittlungsarten

Leitungsvermittlung durchgehender, nichtspeichernder Übertragungskanal zwischen den Endsystemen; Bitfolgen reihenfolgetreu; Vermittlung in den Zwischensystemen erfordert keine zusätzliche Kontrollinformation zur Adressierung; starre Leitungszuordnung; nur verbindungsorientierter Dienst; Beispiel: ISDN

Speichervermittlung Zwischensysteme verfügen über Speicher; flexible Ressourcenzuordnung und Fehlerbehandlung; verbindungsloser Dienst möglich; komplex; Verzögerung, Reihenfolgevertauschung möglich;

Nachrichtenvermittlung

Paketvermittlung Vermittlung aufgrund von Vermittlungsinformation in den Dateneinheiten

Datagramme

virtuelle Verbindung

3.2 Integrated Services Digital Network

Kanäle.

- 1 D-Kanal zur Signalisierung, 16 kbit/s
- 2 B-Kanäle zur Sprach-/Datenübertragung, 64 kbit/s

3.3 Internet Protocol

Dienst. IP stellt einen unzuverlässigen und verbindungslosen Dienst zur Verfügung.

Time To Live ist ein Feld im Header einer IP-Dateneinheit. Dieser Wert wird vom Sender gesetzt und gibt die maximale Anzahl an Hops an, die das Paket durchlaufen darf. In jedem Router, den das Paket durchläuft, wird der Wert um 1 dekrementiert. Pakete mit einem TTL-Wert von 0 werden gelöscht und es wird eine ICMP-Nachricht an den Absender geschickt. Durch TTL wird die Wegewahl unterstützt und es werden umherirrende Pakete vermieden.

3.4 Segmentierung und Reassemblierung

Motivation. Existieren auf einem Pfad durch das Netzwerk unterschiedliche Maximum Transfer Units, so müssen eventuell Dateneinheiten beim Übergang segmentiert und später reassembliert werden. Die Reassemblierung sollte nicht schon in zwischenliegenden Routern erfolgen, da sich Routen während der Übertragungszeit einer fragmentierten Einheit ändern können oder einzelne Fragmente verlorengehen können.

3.5 Zustandsübergangsdiagramm

Stimulus; *Reaktion*

Dienstprimitive beim Initiator.

TConReq Anforderung eines Verbindungsaufbaus

TConConf Verbindungsaufbau erfolgreich

TDisInd Beantworter lehnt Verbindungsaufbau ab

TPAboInd1 Abbruch durch Übertragungsabschnitt

Dienstprimitive beim Beantworter.

TConRsp Bestätigung des Verbindungsaufbauwunschs

TDisReq Ablehnung des Verbindungsaufbauwunschs

TConInd Mitteilung des Verbindungsaufbauwunschs

TPAboInd2 Abbruch durch Übertragungsabschnitt

Protokollinstanz.

Dienstspezifikation. Die Dienstspezifikation beschreibt das an der Dienstschnittstelle von außen betrachtete Verhalten des Dienstes.

Protokollspezifikation. Die Protokollspezifikation legt durch Regeln und Formate das Kommunikationsverhalten zweier Protokollinstanzen fest.

3.6 Routing

statisches Routing vs adaptives/dynamisches Routing. Bei statischen Routing-Algorithmen ist die Routing-Tabelle für längere Zeit fixiert. Adaptives oder dynamisches Routing basiert dagegen darauf, dass die Routing-Tabellen laufend angepasst werden.

3.6.1 Distanz-Vektor-Routing

Prinzip. Beim Distanz-Vektor-Routing kennt kein Knoten die komplette Route von der Quelle zur Senke. Der Knoten kennt nur die Distanz zu allen anderen Knoten.

Distanz-Vektor-Tabelle. Die Distanz-Vektor-Tabelle enthält eine Zeile für jedes mögliche Ziel und eine Spalte für jeden direkten Nachbarn (Next Hop). Knoten X möchte eine Nachricht über den direkten Nachbarn Z an Y verschicken.

$$X \to Z \to Y$$

Dann ist der Eintrag in der Distanz-Vektor-Tabelle von X gegeben durch die Summe aus Kantengewicht zwischen X und Z und dem minimalen Eintrag in der Y-Zeile der Tabelle von Z:

$$D^X(Y,Z) = c(X,Z) + \min_v\{D^Z(Y,v)\}$$

Algorithmus. Zunächst kennt jeder Knoten nur die Distanzen zu seinen direkten Nachbarn, d.h. die Einträge auf der Diagonalen der Tabelle. Diese schickt er an alle direkten Nachbarn, die ihre eigenen Distanzen entsprechend aktualisieren. Solange ein Knoten nach einer Aktualisierung neue kürzeste Wege kennt, schickt er sie an seine Nachbarn.

Algorithm 1: Distanz-Vektor-Algorithmus für Knoten s

3.6.2 Link State Routing

Prinzip. Die Routing-Tabelle jedes Routers enthält die kürzesten Wege zu allen anderen Routern. Zur Berechnung der kürzesten Wege kann z.B. Dijkstras Algorithmus eingesetzt werden. Dadurch, dass jeder Knoten Informationen über seine adjazenten Kanten broadcastet, kann der einzelne Knoten einen Graphen des gesamten Netzes konstruieren und die kürzesten Wege berechnen.

Algorithm 2: Dijkstras Algorithmus

Input: Graph $G = (V, E)$,Startknoten s, symmetrische, positive Kantengewichtsfunktion $c : E \to \mathbb{R}^+$

```
1  N ← {s}
2  for u ∈ V\{s} do
3      if {s, u} ∈ E then
4          d(u) ← c(s, u)
5          p(u) ← s
6      else
7          d(u) ← ∞
8      while N ≠ V do
9          N ∪ {u}, u ∉ N ∧ d(u) = min(d)
10         for {u, v} ∈ E do
11             if d(v) + c(u, v) < d(v) then
12                 d(v) ← d(v) + c(u, v)
13                 p(v) ← u
```

Kostenfunktion. Die Kostenfunktion der Kanten kann von verschiedenen Faktoren abhängig gemacht werden, z.B.

- physikalisches Medium
- Verkehrsaufkommen
- politische Bestrebungen der Netzbetreiber
- Dienstgütemerkmale, z.B. garantierte Bandbreiten, Latenzzeiten oder Zuverlässigkeitsgarantien

3.7 IP-Adressierung

Subnetzmaske Die Subnetzmaske zerlegt eine IP-Adresse in Netzwerk-Teil und Endsystem-Teil. Sie ist eine Bitsequenz der Form 1^m0^n. Den Netzwerkteil einer IP-Adresse erhält man durch und-Verknüpfung von IP-Adresse und Subnetzmaske in Binärdarstellung.

$$IPAdresse \wedge Subnetzmaske = Netzwerkteil$$

Adressbereich. Zur Darstellung eines Adressbereiches wird die Anzahl der 1er-Bits in der Subnetzmaske an die IP-Adresse angefügt. Eine IP-Adresse a liegt im Adressbereich b/m, wenn die ersten m Bits von a und b übereinstimmen.

3.8 Address Resolution Protocol

Aufgabe. Das Address Resolution Protocol ist für die Zuordnung

$$\texttt{IP} - \texttt{Adresse} \leftrightarrow \texttt{MAC} - \texttt{Adresse}$$

zuständig.

Ablauf.

1. suche im eigenen ARP-Cache; wenn Eintrag nicht gefunden:
2. sende ARP-Request mit gesuchter IP-Adresse, eigener IP-Adresse und eigener MAC-Adresse an alle (Broadcast)
3. Besitzer der gesuchten IP-Adresse sendet ARP-Reply mit seiner MAC-Adresse zurück
4. lege Wert im ARP-Cache ab

3.9 Domain Name System

DNS-Anfrage. Eine DNS-Anfrage zur Auflösung eines Domainnamen nach einer IP-Adresse stellt entweder
a) rekursive Anfragen:

- Der Client sucht im lokalen Cache.
- Der Client befragt den Cache des lokalen DNS-Servers.

- Der DNS-Server befragt den Cache (bzw. Datenbank im Falle autoritativer Server) anderer fest definierter DNS-Server.
- Der DNS-Server führt eine sukzessive Auflösung beginnend bei DNS-Rootservern durch

Dabei wird jede erhaltene Antwort im lokalen Cache für einige Tage gespeichert, um die Anfragen möglichst direkt beantworten zu können.

b) iterative Anfragen: Der Client befragt der Reihe nach weitere Server, wenn er keine Antwort erhält.

Effizienz. Aus Effizienzgründen sollte ein Endsystem meist den topologisch nahegelegensten DNS-Server zur Adressauflösung verwenden.

Transport. Der Transport von DNS-Anfragen wird über UDP abgewickelt. Der mit TCP verbundene Overhead kann vermieden werden, da die Sicherungsmechanismen von TCP nicht benötigt werden.

Hierarchie. Das DNS ist in einer Baumstruktur organisiert.

4 Transportschicht (Transport Layer)

4.1 User Datagram Protocol

4.2 Transmission Control Protocol

Charakteristika. Eine TCP-Verbindung stellt einen Duplex-Kanal zwischen Sender und Empfänger bereit. Sie stellt einen bytestromorientierten Dienst zur Verfügung. TCP nutzt verschiedene Mechanismen um eine zuverlässige Verbindung herzustellen:

- Quittungen zur Entdeckung verlorener Pakete
- Sequenznummern um Reihenfolge zu gewährleisten
- Prüfsumme um Integrität der Daten zu garantieren

Bezeichnung	Länge [bit]	Beschreibung
Quellport	16	
Zielport	16	
Sequenznummer	32	gemessen in Byte
Quittung	32	nächste vom Empfänger erwartete Sequenznummer
Offset		Anzahl der 32-Bit-Wörter im TCP-Header
reserviert		
URG	1	gesetzt falls Urgent-Pointer verwendet
ACK	1	unterscheidet bei SYN=1 TConReq von TConCnf, zeigt Quittung an
PSH	1	übergebene Daten sofort weiterleiten
RST	1	Zurücksetzen der Verbindung
SYN	1	zeigt TConReq oder TConCnf an
FIN	1	Sender möchte keine Daten mehr senden
Empfangsfenster	16	Fenstergröße für Sender in Byte zur Flusskontrolle
Prüfsumme	16	über TCP-Kopf und Daten
Urgent Pointer	16	Zeiger auf wichtige Daten
Optionen	$k \cdot 32$	
Daten		

Table 4: Felder einer TCP-Dateneinheit

TCP-Dateneinheit.

Verbindungsaufbau (3-Way Handshake). Der Verbindungsaufbau wird vom Client initiiert.

1. TConReq(SYN=1, seq=client-isn) →
2. ← TConCnf(SYN=1, ACK=1, seq=server-isn, ack=client-isn + 1)
3. ACK(SYN=0, ACK=1, seq=client-isn + 1, ack=server-isn + 1) →

Verbindungsabbau. Bei TCP wird wenn möglicher ein expliziter Verbindungsabbau durchgeführt, statt die Verbindung per Reset zurückzusetzen, da zum Zeitpunkt des Verbindungsabbaus noch Daten unterwegs sein können. Bei einem Reset wäre es nicht unbedingt erkennbar, welche Daten noch korrekt empfangen wurden. Der Verbindungsabbau kann sowohl vom Client als auch vom Server initiiert werden. Nach dem letzten ACK wird noch gewartet, bevor der lokaler Kontext gelöscht wird, da noch Pakete verlorengehen könnten.

1. FIN →
2. ← ACK

5 Sitzungsschicht (Session Layer)

6 Darstellungsschicht (Presentation Layer)

7 Anwendungsschicht (Application Layer)

7.1 Anwendungsprotokolle

HyperText Transfer Protocol Austausch von HTML-Dokumenten

File Transfer Protocol Austausch von Dateien zwischen Dateisystemen

Simple Mail Transfer Protocol Transport von elektronischer Post