Rechnerkommunikation

Felix Leitl

3. August 2023

Inhaltsverzeichnis

Anwendungsschicht

Netzwerkanwendung:

- Anwendungsprozesse auf verschiedenen Hosts
- kann direkt unter Verwendung der Dienste der Transportschicht implementiert werden
- standardisieret Anwendung benutzen ein Anwendungsprotokoll, das das Format der Nachrichten und das Verhalten beim Empfang festlegt

Paradigmen

Client-Server

Server stellt Dienst zur Verfügung, der vom Client angefragt wird

Wechselnde Rollen

Hosts übernehmen mal die eine, mal die andere Rolle

Verteilte Anwendung

Besteht aus mehreren unabhängigen Anwendungen, die zusammen wie eine einzelne Anwendung erscheinen (z.B. WebShop mit Web-Server, Applikations-Server und Datenbank), Koordination ist zwar verteilt, findet aber für das Gesamtsystem statt

Peer-to-Peer

Vernetzung von Gleichen:

- dezentrale Architektur (z.B. Bitcoin)
- Hybridarchitektur: Initialisierung findet über zentrale Architektur statt, Anwendung dezentral zwischen Hosts

Anforderungen

- Verlust
- Bitrate
- Verzögerungszeit

Hypertext Transfere Protocol (HTTP)

Ablauf

1. Benutzer gibt URI (Uniform Resource Identifier) in Web-Browser ein
2. URI enthält Host-Namen eines Web-Servers und den Pfad zu einem Objekt
3. Web-Browser stellt Anfrage an Web-Server für dieses Objekt
4. Web-Server liefert Objekt an Web-Browser zurück
5. Web-Browser stellt Objekt für Nutzer lesbar da

Format der Anfragen

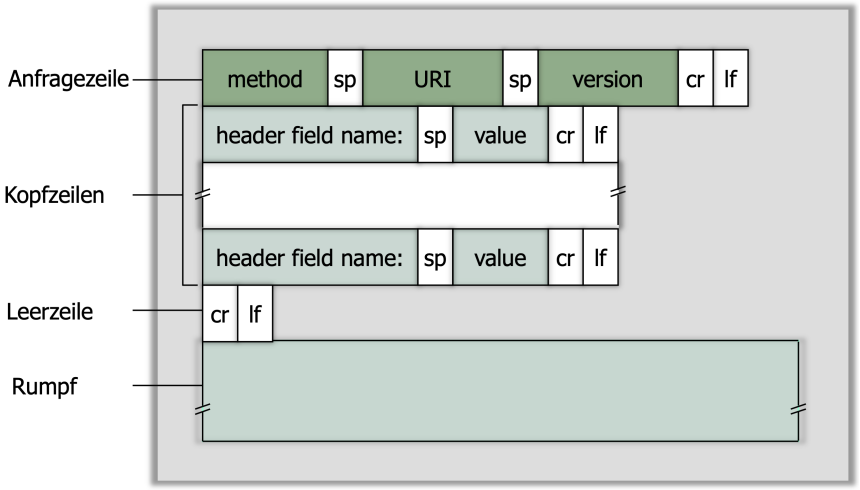

Abbildung 1: Anfrage

Anfragenachricht

- Methoden:
  - GET: Abruf eines Dokuments, besteht aus Methode, URI, Version
  - HEAD: Abruf von Metainformationen eines Dokuments
  - POST: Übergabe von Informationen an Server
  - PUT
  - DELET
- Kopfzeilen:
  - Typ/Wert-Paare, Typen: Host, User-agent, ...
- Rumpf:

– leer bei GET, kann bei POST Inhalt haben

GET:	/somedir/page.html HTTP/1.1
HOST:	www.someschool.edu
User-agent:	Mozilla/4.0
Connection:	close
Accept-language:	de-de

Abbildung 2: Anfragenachricht

Format der Antworten

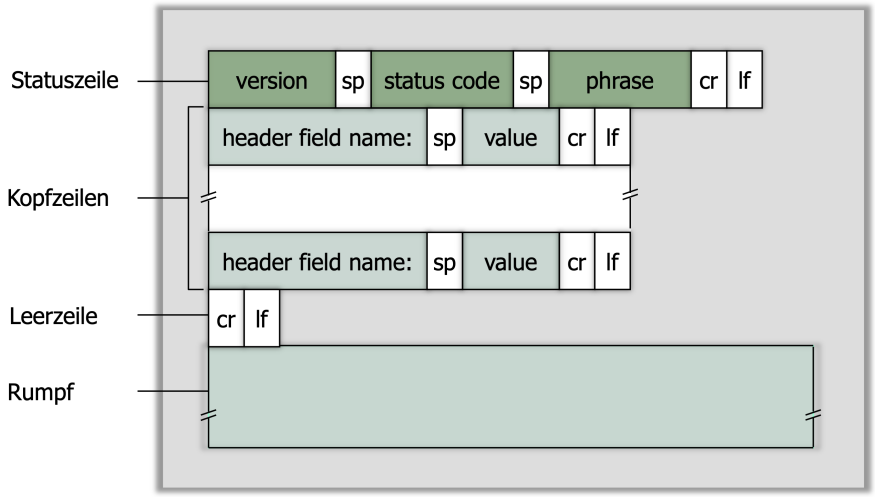

Abbildung 3: HTTP Antworten

Antwortnachricht

HTTP/1.1	200 OK
Connection:	close
Date:	Thu, 06 Aug 1998 12:00:15 GMT
Server:	Apache/1.3.0 (Unix)
Last-Modified:	Mon, 22 Jun 1998 ...
Content-Length:	6821
Content-Type:	text/html
data	data

Abbildung 4: Antwortnachricht

HTTP-Ablauf

Nicht-persistentes HTTP:
Für jedes Objekt wird eine einzelne TCP-Verbindung aufgebaut. Entweder Basisseite und eingebettete Objekte sequentiell oder parallele Verbindung für eingebettete Objekte

Persistentes HTTP:
Server lässt Verbindung bestehen, alle Objekte werden über eine TCP Verbindung gesendet. Ohne Pipelining wird jedes Objekt einzeln Angefragt, mit alle auf einmal

Antwortzeit

Basisseite: Aufbau der TCP-Verbindung (1x RTT) + Anfrage hin und Antwort zurück (1x RTT) $\Rightarrow$ 2RTT + Zeit zum Senden + weitere Wartezeiten durch TCP

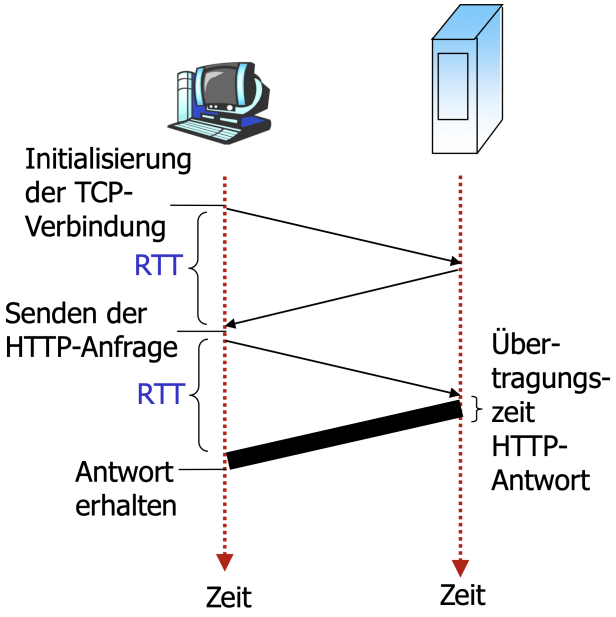

Abbildung 5: Antwortzeit

Dynamische Inhalte

Common Gate Interface (CGI) verarbeitet als externer Prozess die Information und liefert neue HTML-Seite an Server

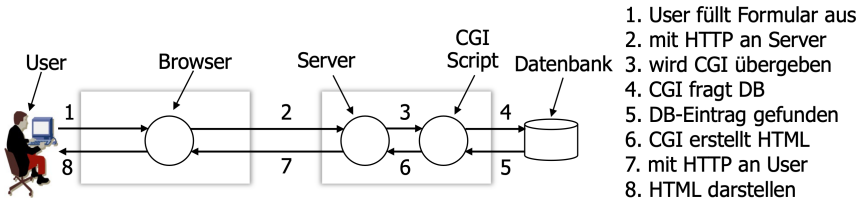

Abbildung 6: CGI

Scripting: Durch Interpretation von eingebetteten Skripten können dynamische Inhalte erzeugt werden.

Serverseitig: im HTML ist Code eingebettet, der vom Server interpretiert wird und dabei HTML erzeugt, z.B. PHP

Clientseitig: im HTML ist Code eingebettet, der vom Client interpretiert wird, z.B. JavaScript

Caching

Cache (Proxy Server) ist Server für Web-Browser und Client für Web-Server, der als Zwischenspeicher zur Verringerung der Wartezeit des Nutzers und des Netzverkehrs dient

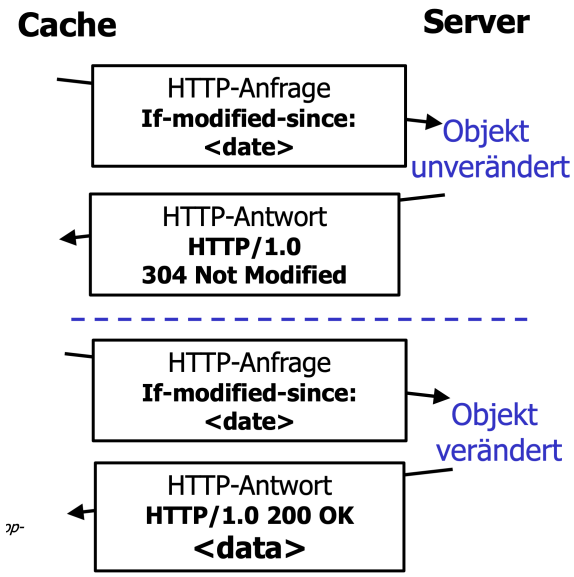

Abbildung 7: Cache Server

HTTP/2

Wesentliche Elemente:

- gleiche Methoden
- binäres statt textbasiertes Format
- Multiplexing verschiedener Ströme über eine TCP-Verbindung, Vermeidung von Head-of-Line (HOL) Blockierung durch große Objekte durch Aufteilung in kleinere Frames und Interleaving
- Header-Kompression
- Server-Push

File Transfere Protocol (FTP)

- Übertragung zwischen zwei Hosts
- eine TCP-Verbindung (Port 21) zur Steuerung
- lesbare Kommandos: USER username, PASS password, LIST, PETR filename, STOR filename, ...
- jeweils eine TCP-Verbindung (Port 20) zur Übertragung einer Datei
- 'out-of-band-controll'

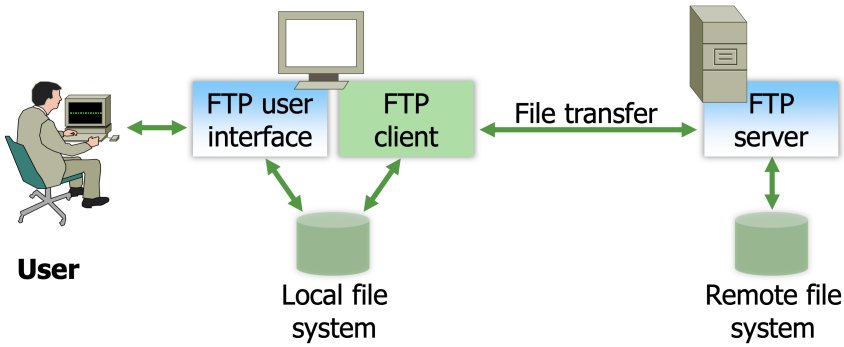

Abbildung 8: FTP

Simple Mail Transfer Protocol (SMTP)

- Nachrichten im ASCII-Format, Kopf, Rumpf
- andere Daten werden in ASCII umgewandelt angehängt
- Versenden mit SMPT über TCP (lesbar)
- Abholen mit POP3, IMAP, HTTP (lesbar)

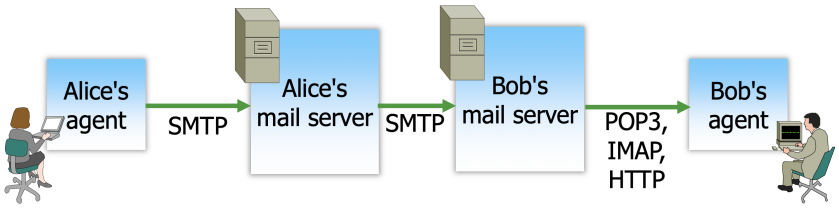

Abbildung 9: SMTP

- nutzt TCP (Port 25)
- direkte Übertragung: vom Sendenden zu empfangendem Server
- drei Phasen der Übertragung:
  - Handshake
  - Nachrichtenübertragung
  - Abschlussphase
- Interaktion mittels Befehlen und Antworten
  - Befehle: ASCII-text
  - Antworten: Statuscode und Text
- Nachrichten müssen 7-bit ASCII-text sein

Vertraulichkeit und Datenintegrität

1. Erzeugung eines Hashwerts der E-Mail
2. Signierung mit privatem Schlüssel K_A^- von Alice
3. Verschlüsselung der Mail und der Signatur mit K_S
4. Asymmetrische Verschlüsselung von K_S mit dem öffentlichen Schlüssel K_B^+ von Bob

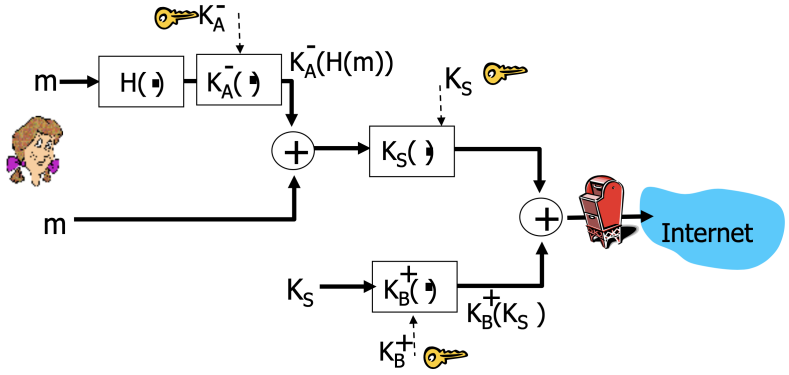

Abbildung 10: Vertraulichkeit und Datenintegrität

Netzwerkmanagement

Überwachung und Verwaltung eines Netzes = komplexes HW/SW Gebilde
FCAPS-Modell nach ISO:

- Fault Management: Monitoring, Dokumentation, Reaktionsmaßnahmen
- Configuration Management: Übersicht über Geräte und deren HW/SW-Konfigurationen
- Accounting Management: Verwendung von Netzwerkressourcen, Festlegung, Kontrolle, Dokumentation des Zugangs von Benutzern und Geräten
- Performance Management: Monitoring von Auslastung, Durchsatz, Antwortzeiten, Dokumentation, Reaktionsmaßnahmen
- Security Management: Monitoring und Kontrolle des Zugangs, Schlüsselverwaltung, Firewalls, Intrusion Detection

Simple Network Management Protocol (SNMP)

Organisationsmodell:

- Managing Entity: Prozess auf zentraler Management Station, Client
- Managed Device
- Managed Object: HW oder SW im Managed Device
- Managed Agent: Prozess auf Managed Device, kann lokal ausgeführt werden
- Anfrage/Antwort-Protokoll zwischen Manager und Agent über UDP
- Manager basierend auf Regeln, was zu tun ist

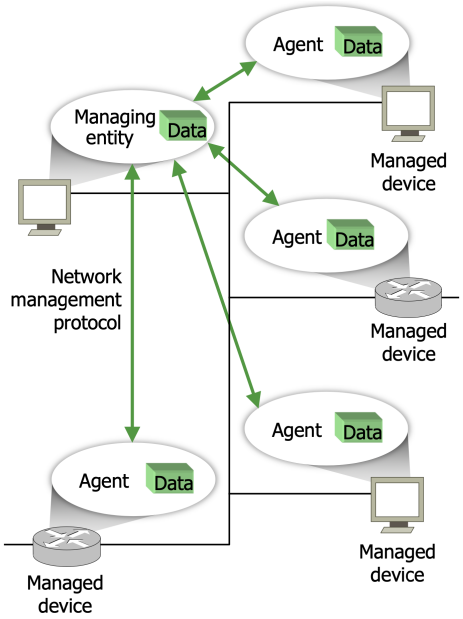

Abbildung 11: SNMP

SNMP Messages:

- **Get**: Manager an Agent, um Daten von Agent zu erhalten
- **Get-Next**: Manager an Agent, für nächsten Datensatz, Zugriff auf sequentielle Datensätze
- **Get-Bulck**: Manager an Agent, für mehrere Datensätze auf einmal
- **Set**: Manager an Agent, initialisiert oder ändert den Wert eines Datensatzes
- **Response**: Agent an Manager, Antwort auf Get- und Set-Nachrichten
- **Trap**: Agent an Manager, unaufgeforderte Nachricht über Fehlersituation

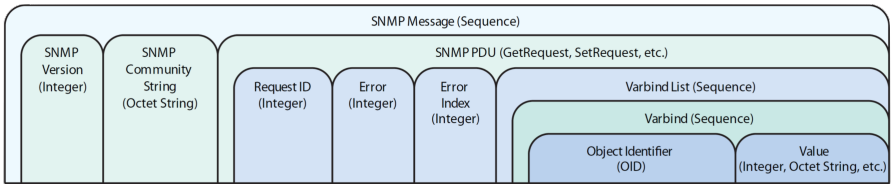

Abbildung 12: Format von SNMP Nachrichten

Management Information Base (MIB)

MIB-Module enthalten Datenstrukturen für die Managed Devices. Das genaue binäre Format der Übertragung wird mit Basic Encoding Rules (BER) festgestellt. Jedes MIB-Modul wird eindeutig durch die Objekt-Identifizierung (OID) der ASN.1 bezeichnet.

```
UDP-MIB DEFINITIONS ::= BEGIN
…
udp      OBJECT IDENTIFIER ::= { mib-2 7 }
…
udpInDatagrams OBJECT-TYPE
    SYNTAX      Counter32
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
            "The total number of UDP datagrams delivered to UDP users."
    ::= { udp 1 }
…
udpGroup OBJECT-GROUP
    OBJECTS   { udpInDatagrams, udpNoPorts,
                udpInErrors, udpOutDatagrams,
                udpLocalAddress, udpLocalPort }
    STATUS    current
    DESCRIPTION
            "The udp group of objects providing for management of UDP
            entities."
    ::= { udpMIBGroups 1 }

END
```

Abbildung 13: MIB-Modul

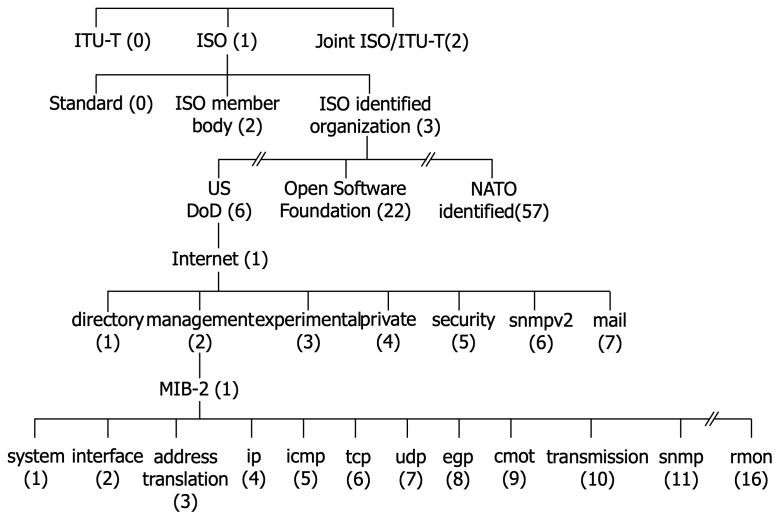

Abbildung 14: OID

Domain Name Service (DNS)

DNS bildet Domain-Name auf Werte ab und ist als verteilte Datenbank, bestehend aus vielen Name-Servern, repräsentiert

Domain-Struktur

DNS implementiert hierarchischen Namensraum für Internet-Objekte

Hierachie von Name-Servern

- **Root Name Server**: weltweit
- **Top-level Domain-Server** für com, org, net, ...
- ...
- **autoritativer Name-Server**: unterste Ebene für einzelne Organisationen

Ressource Records

- **TTL**: Time to Live, Dauer der Gültigkeit
- **Typ = A**:
  - Wert = IPv4-Adresse
  - AAAA-Einträge für IPv6-Adressen
- **Type = NS**:
  - Wert = Domainname eines Hosts, auf dem ein Server läuft, der Namen in der Domain auflösen kann
- **Type = CNAME**: (Canonical Name)
  - Wert = kanonischer Name eines Hosts, ermöglicht Aliasnamen
- **Type = MX**: (Mail Exchange)
  - Wert = Domain-Name des Hosts, auf dem der Mail-Server läuft

DNS-Protokoll

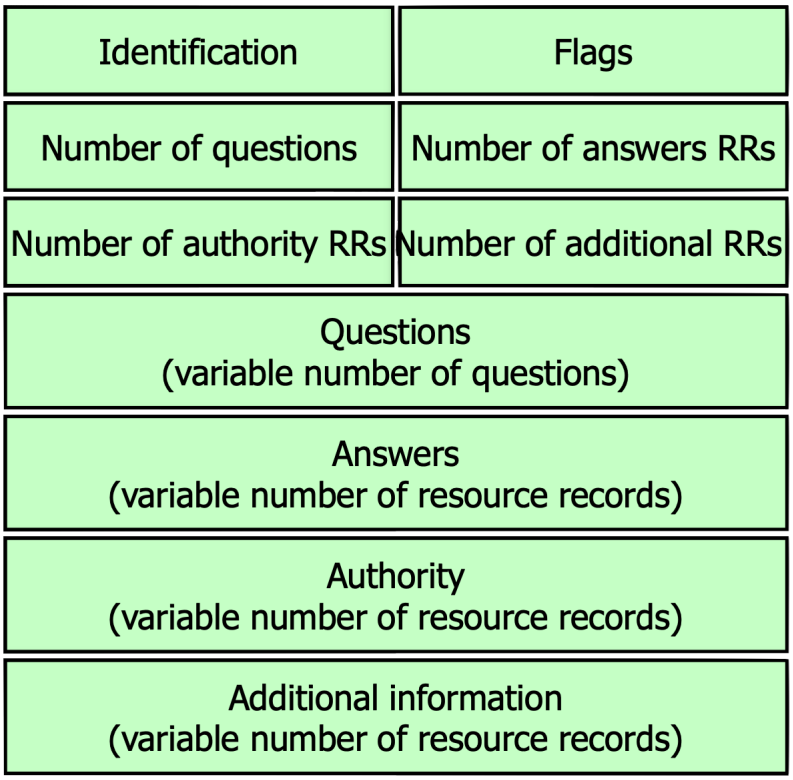

Abbildung 15: DNS Protokoll

Anfragen

iterativ:

- Antwort: anderer Server, der Name evtl. auflösen kann (oder keine Antwort)
- NS- und A-Datensatz
- Antwort wird sofort geliefert, es muss keine Information gespeichert werden, gut für hochfrequente Server

rekursiv:

- Antwort: Auflösung des Namens, die u.A. von anderen Servern geholt wird
- A-Datensatz
- bei Anfragen an einen anderen Server muss die Information gespeichert werden

Content Distribution Network

Ziel: massive gleichzeitige Nutzung einer Webseite mit mehreren Mbit/s pro User zu realisieren

Idee: sehr viele Spiegelserver geographisch verteilen und näher am Nutzer platzieren

Verteilung der Anfragen

- **Server-basierte HTTP Redirection**: Server liefert aufgrund der IP-Adresse des Clients einen geeigneten anderen Server, erfordert zusätzliche RTT, Server-Betreiber muss Verteilung des Inhalts kennen, nicht bewährt
- **Client-nahe HTTP Redirection**: z.B. durch Web-Proxy, nicht bewährt
- **URL Rewriting**: Server liefert Basisseite, die URLs der eingebetteten Objekte werden umgeschrieben, mit dem Domain-Namen eines geeigneten Servers
- **DNS-basierte Redirection**: DNS-Server bilden Domain-Namen des Services auf Domain-Namen und zuletzt auf die IP-Adresse eines geeigneten Servers ab

Socket-Programmierung

Socket-Schnittstelle

- verbreitetes API für Transportdienste
- Festlegung von TCP/UDP, IP-Adressen, Portnummern

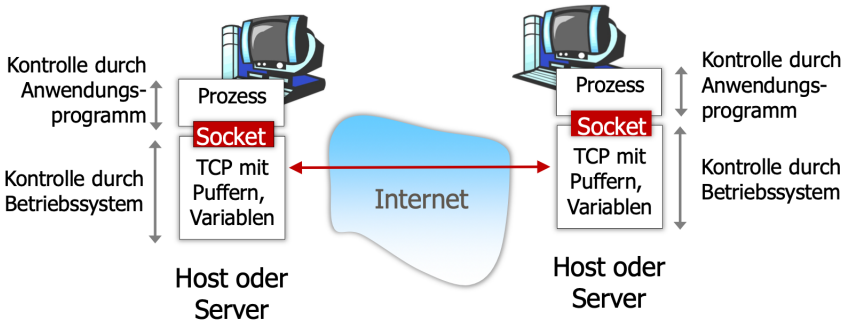

Abbildung 16: Socket

Peer-to-Peer

Grundidee: Inhalte nicht nur von zentralem Server, sondern auch von anderen Peers. Upload-Bitrate der Peers wird mitgenutzt

Unstrukturiert

Zentralisiertes Verzeichnis:

- Eintritt eines Peers: informiert zentralen Server über seine IP-Adresse und seine Inhalte
- Suche nach Inhalt: über zentralen Server
- Dateiübertragung: direkt zwischen Peers
- zentraler Server: juristischer und leistungstechnischer Schwachpunkt

Dezentralität durch Fluten von Anfragen

- Peers bilden Overlay-Netzwerk über TCP-Verbindungen
- anfragender Peer sendet Anfrage an alle seine Nachbarn im Overlay-Netzwerk
- diese vergleichen Anfragen mit den von ihnen angebotenen Inhalten
- wenn nicht gefunden werden die Anfragen an mehrere Nachbarn weitergeleitet
- das Fluten wird durch einen maximalen Hopcount begrenzt
- bei fund wird dem anfragendem Peer geantwortet (nicht dem ursprünglichen, womit seine Identität geheim bleibt)
- Verbindung wird mittels HTTP zwischen Quelle und Ursprung hergestellt

Hybrid

Peers bilden Gruppen, einer ist Group-Leader, P2P-Netzwerke in Gruppen und zwischen Leadern ⇒ besser skalierbar

Strukturiert

Verteilte Hash-Tabllen

- Peers bilden ringförmiges Overlay-Netz
- jedem Peer wird ein zufälliger Bezeichner $p (0 \leq p \leq 2^m - 1)$ aus ringförmigen Bezeichnerraum mit m Bits zugewiesen
- jedem Datenelement wird mittels Hash-Funktion ein Schlüssel k ebenfalls aus diesem Raum zugewiesen
- Nachfolger von k:
  - Datenelement mit Schlüssel k wird auf Nachfolger p von k gespeichert (Nachfolger darf Knoten k selbst sein)
  - der nächste Peer im Ring, formal: Peer p mit dem kleinsten $a \geq 0$, sodass $p = \texttt{succ}(k) = (k + a) \mod 2^m$ existiert
  - auslesen eines Datenelements mit Schlüssel k erfolgt auf Peer succ(k)

Bittorrent

- **Torrent**: Schwarm von Peers fu2r gleiche Datei
- **Chunks**: Teile der zu verteilenden Datei
- **Tracker**: Server, bei dem sich Peers registrieren
- .torret-Datei mit Metadaten über zu verbelibende Datei und Tracker
- neuer Peer tritt Schwarm bei:
  - A registriert sich bei Tracker und erhält IP-Adressen zufälliger anderer Peers des Schwarms
  - A baut TCP-Verbindung zu einigen dieser Peers auf, fragt Liste der Chunks in ihrem Besitz nach und sendet Anfragen für Chunks
  - Rarest First: A fragt die seltensten Chunks der Peers zuerst nach, dadurch gleichmäßige Verteilung
  - Incentive Mechanismus (Tit-for-Tat): A misst Antwortrate der Peers und antwortet an diese in entsprechendem Anteil der Upload-Rate
  - neue Nachbarn werden zufällig dazu genommen

Transportschicht

Netzwerkschicht

Sicherungsschicht

Physikalische Schicht