

3. VO-Einheit :: 30.10.'17

** ">> x"; View page x, x being the page in script

HTTP Today

— Rep

HTTPS - Zertifizierungsproblem: Selbsterstelltes vs Geprüftes (Bei Firmen)

HTTP/2 - Erweitert um Session Layer & Streams; Server Pushes;

120' MPEG Meeting

Definierungsg'schichten:

- Point Cloud Compression: Große Objekte im 3D Raum definieren um darin zu navigieren - Darstellung via Video u.ä..
- Omnidirectional Media Format - OMAF: 360° Videos und Fotos >> Facebook Gruppe
- MPEG-G: Kompression und Transport von Genomic (Biomedizinischen) Daten.
- AVC → HEVC (Nachfolger): MPEG & VCEG call for proposals (für nächste Standards)
- MMT: Media Transport, Übertragungsformat in TV/...
- VB8, VPC: Internet Video Codex' definieren, ohne Patentbezug

— Alle mit x-hundert Patenten

File Transfer Protocol (FTP)

— Video-Streaming implementierbar

— TCP

→ 2 Kanäle: Steuerinformation & Datenfluss

— Klient bekommt Zugang zu entferntem Verzeichnis (Control Connection)

— Server startet TCP-Connection zum Klient; beinhaltet Zustände über Authentifizierung des Klienten, gebrowstes Verzeichnis, ...

— z.B. MPEG; Allerdings zurückgedrängt von Dropbox & Co

Electronic Mail

— Mailreader

- User Agent: Outlook, Thunderbird,...

— Mail Server

- Mailbox der gespeicherten Nachrichten

— Protokoll: Simple Mail Transfer Protocol: SMTP — Austausch zwischen Mailservern, welche die Nachrichten weiterleiten

- Permanente Verbindung bis .close() für Transfer, nutzt Port 25

— 3 Phasen des Verbindungsaufbau:

- Handshake
- Transfer of Data (Status Code & Phrase)
- Close

— Definiert in 7-bit ASCII text

Vgl.: SMTP Beispiel // Gekürzt mit Live

— Apple-Terminal:

\$ telnet mail mailsrv.aau.at 25 → Verbindung zum mailserver der AAU (INNERHALB DER UNI)

220 - Statuscode; Server; ESMT PostfixP Protokoll;

\$ Hello strolchi.at

250 mailsrv.aau.at

MAIL FROM: dude@dude.com

250 2.1.0 OK

RCPT TO: my-email

250 2.1.5 OK

DATA

354 End data with <CR><LF>.<CR><LF>

- Write a damn email

bla

.

250 2.0.0. OK: Queued as ...

QUIT

... Oder siehe 25 und 26 für Header

Multipurpose Internet Mail Extenseion - MIME

— Alternative Mailkodierung :: Header, HTML-embedded,...

Verschlüsselungsmodi: cbcs, cenc >> siehe 28

Mail Access Protocols

Datenempfang:

— POP: Post Office Protocol (POP3, veraltet; Port 110)

- Server wartet auf eingehende Verbindung
- Authentifizierung durch APOP
 - o Stat: Verfügbare byte
 - o List: Definiere Nachricht
 - o Retr: Get from POP-Server
 - o DELE x: Markiere X zu löschen
 - o Quit: Terminiert verbindung

— IMAP: Internet Mail Access Protocol (IMAP4, mehr functs)

1. IMAP wie POP mit mehr Funktionalitäten.

— HTTP

>> 30 & 31

Domain Name System: DNS

— Mapping von IPs

- Vergeben von Aliassen → Zusätzliche Hostnamen
 - Umziehen zwischen Hosts → Alias zeigt auf neue IP
- Load Distribution
 - IPs → Hostnamen
 - Hosting via Server: Netflix-Datenübertragung via KELAG-Server
- Bei Ausfall von DNS: IP-Adresse eingeben.
- Bei Ausfall eines DNS: Egal → No Bottlenecks
- >> Siehe 32: DNS Name Space

- DNS Server Hierarchie
- Jeder Server hat mindestens 2 Hosts.
- Local Name Server → ISP (THCP)
- DNS-Anfragen gehen primär zu diesen
- Top-Level-Domain (TLD)
 - Root name server
 - o Highest Level; 13 Root server weltweit

- DNS Protokoll:
 - Host kontaktiert lokalen DNS server
 - o Lokaler DNS kontaktiert root server
 - Root kontaktiert Toplevel domain
 - o ... kontaktiert lokalen DNS server
 - Zieladresse

— Modellentwicklung: Rekursiv, Iterativ

Rekursiv: Abfrage bis ein Server die Adresse kennt & zurückgabe über selben Weg, e.g. obriges
 Iterativ: Abfrage auf oberen Server, Antwort zurück mit möglichem DNS >> 37

- DNS: Caching & updating:
 - Caching abgerufener Adressen gemäß ttl, etwa 2 Tage.
 - Resource Records (RR): name, value, type, ttl
 - Typ = A: name & value → host & ip :: foo.com, 123.213.123.22
 - Typ = NS: Iterative Query :: Name is Domain, value is hostname
 - Type = CNAME
 - Type = MX
 - Mehr typen in iana.org; >> 39
- Syntax: Name - TTL - IN - Type - Value

— DNS Protocol

DNS-Protocol: Request & Response-Protocol bei 32-Bit messages (ähnlich Peer-to-Peer)

- eigene ID
- flags: Typen
- Def über Nachricht

– Question, answer, authority, information, ...

Sockets

Transport Service Access Point

– TCP oder UDP → Explizit verwendet

Sockets - TCP

Serverprozess läuft und wartet auf Anschluss

– Klient kontaktiert Server mittels

1. Lokalen TCP socket
2. Port und URL Spezifikation
3. Klient etabliert Verbindung zum Server

– Server

1. .. kreiert neuen Socket zur C/S-Kommunikation ↔ ermöglicht multiple Klienten
2. Source Port wird zugewiesen von OS

>> 43++ || 2b.pdf: Sockets - 4++

– Serversocket, Socket

Sockets - UDP

Sender gibt explizit IP adresse und Port an.

Server muss von gesendert IP Adresse und Port extrahieren.

– Datagramsocket