Rechnernetze & Verteilte Systeme

Ludwig-Maximilians-Universität München

Sommersemester 2018

Prof. Dr. D. Kranzlmüller

MNM TEAM

Wiederholung: IPv4 und Subnetting

MNM TEAM

Notation von IPv4-Adressen

- Durch Punkte getrennte, byteweise Dezimalschreibweise: p.q.r.s
- wobei p,q,r,s Dezimalzahlen zwischen 0 und 255 sind.
- Beispiel: „208.77.255.0“

- Binärzahl-Darstellung oft hilfreich (Stichwort: Netzmasken)

Sonderadressen

p.q.r.s:

- alles 0: dieser Host
- alles 1: Broadcast innerhalb des lokalen Netzes
- 127.*.*.*: Loop-Back-Adressen (Schleifentest)

Später: 2 spezielle Adressen

- Netzadresse:
  niedrigste Adresse (Host-ID = alles 0)
- Broadcast-Adresse:
  höchste Adresse (Host-ID = alles 1)

Klassenbasierte Adressierung (Engl.: Classful Networking)

Grundidee:

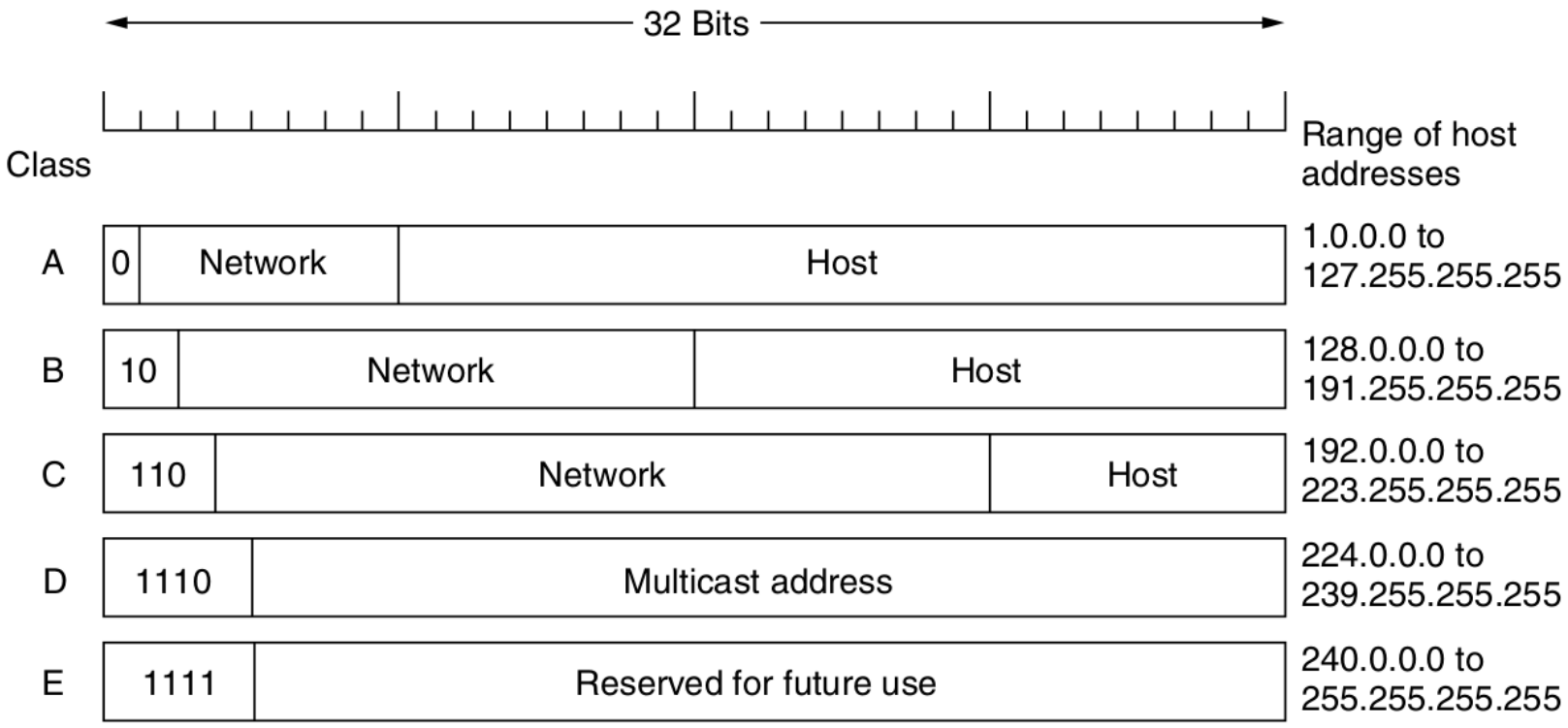
- Einteilung in Klassen anhand des Verwendungszwecks:
  - Netzgrößen für Unicast (A,B,C)
  - Multicast Network (D)
  - Future and Experimental Use (E)
- Grenze zwischen Netz-Teil und Host-Teil verläuft entlang der Byte-Grenzen ➔ Softwareimplementierung
- Netz-Teil wird in Präfix und Netz-ID unterteilt
  - Erlaubt schnelle Erkennung der Klasse anhand weniger Bits
  - Schnelle Routing-Entscheidungen anhand Klasse und Netz-ID

Klassenbasierte Adressierung (Engl.: Classful Networking)

https://de.wikipedia.org/wiki/Netzklasse

Klasse	Präfix	Länge Netz ID	Länge Host ID	Anzahl Netze	Hosts pro Netz
A	0	7 bit	24 bit (3 byte)	126	16 777 214
B	10	14 bit	16 bit (2 byte)	16 382	65 534
C	110	21 bit	8 bit (1 byte)	2 097 152	254
D	1110	Verwendung für Multicast-Anwendungen			
E	1111	Reserviert (für zukünftige Zwecke)			

Klassenbasierte Adressierung (Grafik)



Quelle: Tanenbaum, Computer Networks 5th Edition

Private IP-Adressen (1/2)

- I.d.R. vergeben ISPs (engl.: Internet Service Providers) an Privatkunden oder kleine Firmen nur eine einzige öffentliche IP Adresse (unabhängig davon wie viele Hosts diese in ihrem LAN anschließen).
- Innerhalb ihres LANs (engl.: Local Area Network) verwenden die Hosts sogenannte private IP-Adressen (nur innerhalb des LANs eindeutig)

Private IP-Adressen (2/2)

- Folgende Adressblöcke sind als privat reserviert:
  - 10.0.0.0/8 (Klasse A Adressblock)
  - 172.16.0.0/12 (16 Klasse B Adressblöcke)
  - 192.168.0.0/16 (256 Klasse C Adressblöcke)

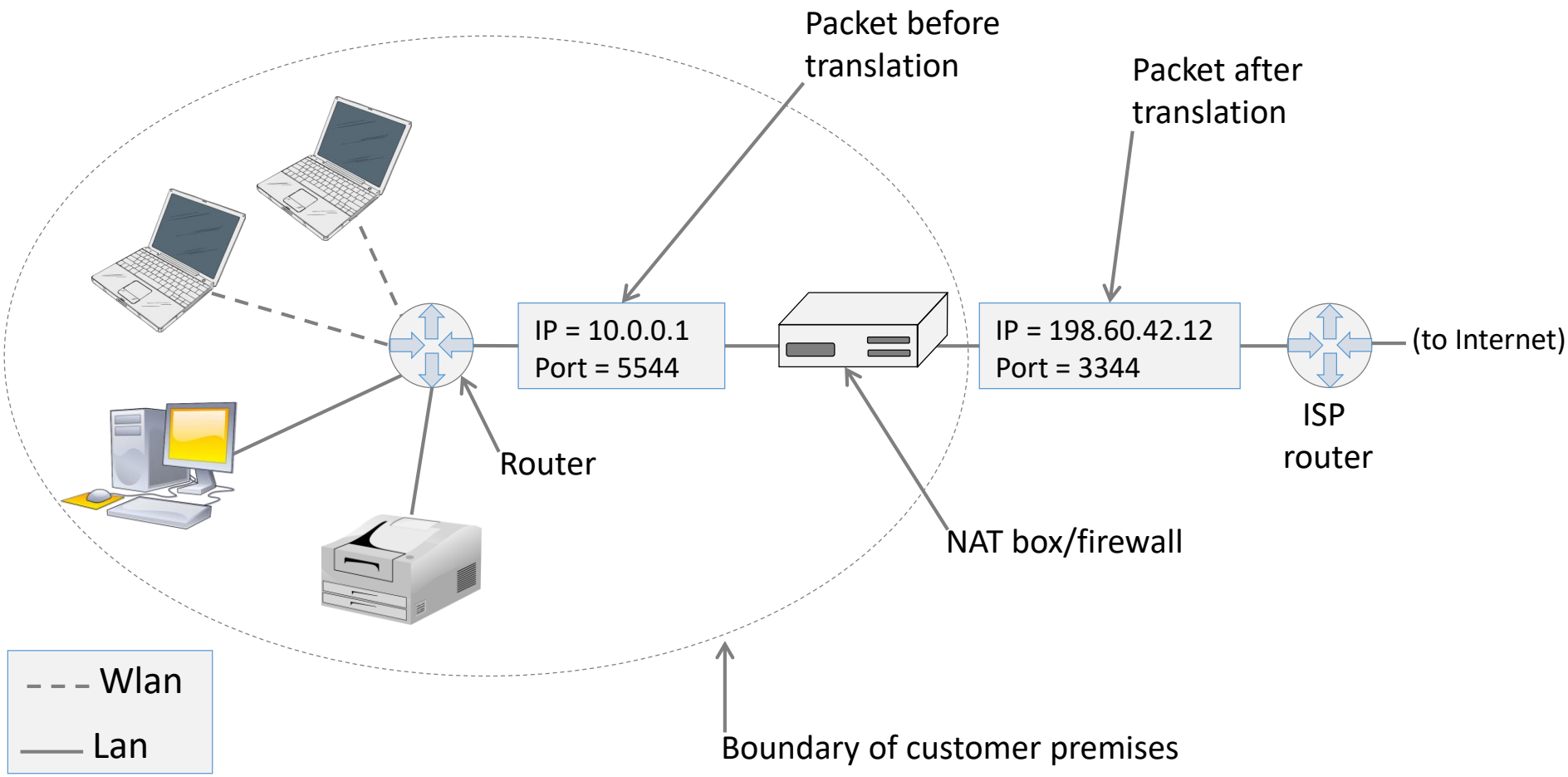
- Private Adressen werden nicht im Internet vermittelt
- Pakete mit privaten Adressen als Ziel oder Absender werden von Routern verworfen

NAT (Network Address Translation)

(Vergleiche auch „Kapitel 1.4: Ein Einführendes Beispiel“)

- Bevor Router ein Paket aus dem privaten LAN ins Internet weiterleitet, wird Absenderadresse in öffentliche IP-Adresse geändert
  ➔ „*Address Translation*“
- Router merkt sich die Änderung der Adresse
- Zusätzlich zur Adresse wird auch der verwendete Port gemerkt (siehe Sockets ➔ Kapitel 3)
- Auf „Rückweg“ erfolgt umgekehrte Übersetzung.

NAT/Mascquerading (als Grafik)



Klassenbasierte Adressierung Grundproblem

- Adressen werden in Blöcken vergeben (z.B.: alle Adressen mit einer gegebenen Netz-ID)
- Wird z.B. ein Klasse B Netz an eine Organisation vergeben welche deutlich weniger als 65000 Hosts besitzt, so bleiben viele Adressen ungenutzt.
- (Oft reicht bereits die Erwartung, dass 254 Hosts einmal nicht reichen könnten, für die Anforderung eines Klasse B Adressblocks.)
- Es gibt nur 16382 Klasse B Netze.
- Lösung: Netz IDs mit variabler Länge ➔ CIDR

CIDR (Engl.: Classless Inter-Domain Routing)

- Mit RFC 1517 bis 1520 ab 1993 in Verwendung.
- Grundidee: Die Grenze zwischen Netz-Teil und Host-Teil verläuft fließend.
  => Routing-Protokolle müssen die Länge der Netz-ID (Netzpräfix) zusätzlich zur Adresse übertragen.
- Verwendung von Subnetz-Masken
- Notation: <IP Adresse>/<Präfixlänge>

- Beispiel: 192.168.121.0/26
  (die ersten 26 Bit sind Netz-ID, der Rest Host-ID)

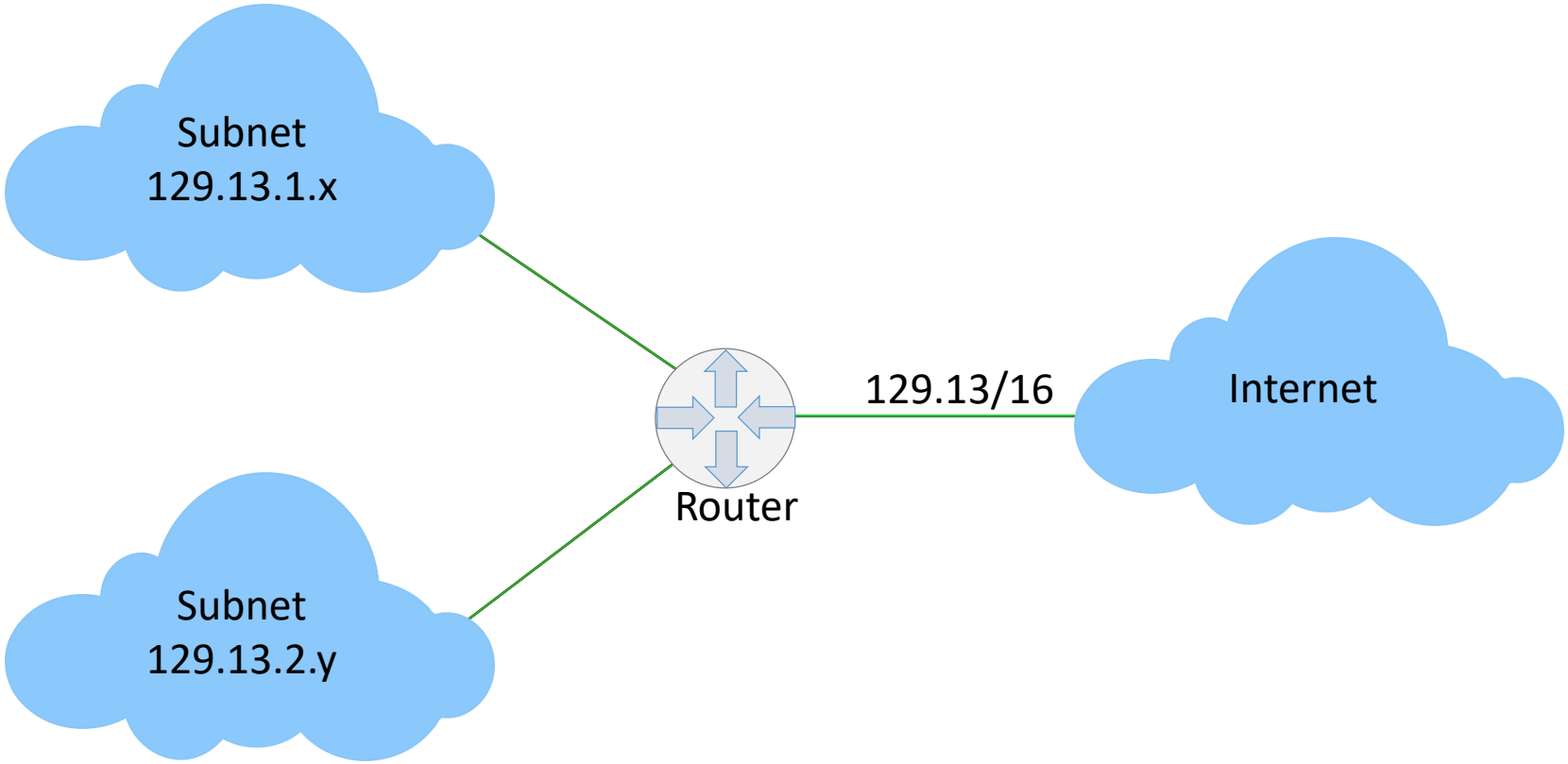
Subnetting

- Sowohl bei klassenbasierter Adressierung, als auch bei CIDR
- Grundidee: Host-Teil eines Adressblocks wird weiter unterteilt in Subnetz-ID-Teil und Host-ID-Teil.

Netz ID	Host ID	
Netz ID	Subnetz ID	Host ID

- Eine Organisation kann mit einem einzigem Adressblock mehrere eigene Netze bedienen

Subnetting (als Grafik)



Beispiel: Subnetting

Eine Organisation bekommt den Adressblock 131.42.0.0/16 zugewiesen und benötigt

- 1 Subnetz mit bis zu 32000 Hosts
- 15 Subnetze mit bis zu 2000 Hosts
- 8 Subnetze mit bis zu 250 Hosts

Machen Sie Vorschläge für eine Aufteilung in geeignete Subnetz-Adressen

Fragen zu Kapitel 2 (1/2)

- Welche Maßnahmen zur Begegnung von IPv4 Adressknappheit wurden in der Vorlesung behandelt?
- Wie viele Hosts kann ein IPv4 Adressblock mit einer Präfixlänge von 22 maximal fassen?
- In welches der folgenden Subnetze des Netzes 184.212.0.0/16 gehört die Adresse 184.212.2.36?
  - 184.212.128.0/17
  - 184.212.0.0/18
  - 184.212.96.0/19

Kapitel 6: Dienste der Anwendungsschicht

ISO/OSI-Schicht 7

(Engl. Application Layer)

Internet-Dienste (ISO/OSI-Schichten 5-7)

MNM TEAM

Inhalt von Kapitel 6

1. Das Anwendungssystem im Überblick
   - Anwendungsschicht im Prinzip
   - ISO/OSI vs. Internet
   - DNS Wiederholung
2. Electronic Mail (Email)
   - SMTP, MIME, POP, IMAP
3. World Wide Web (WWW)
   - HTML, URL, HTTP
4. File Transfer Protocol (FTP)

Einordnung von Kapitel 6

- Das Ziel von Rechnernetzen und verteilten Systemen: Endnutzer-Anwendungen sollen über *Rechnernetze* beliebige Nachrichten/Daten (virtuell/transparent) austauschen können.
- Das Mittel: Die Schichtenarchitektur ➔ Kapitel 1
- Insbesondere: Diensterbringung unter der Verwendung „tieferer Schichten“. ➔ Kapitel 3 bis 5
- Dieses Kapitel: wichtige Internet-Dienste (Diese setzen die bereits behandelten „tieferen Schichten“ einfach voraus.)

Kapitel 6.1
Das Anwendungssystem im Überblick

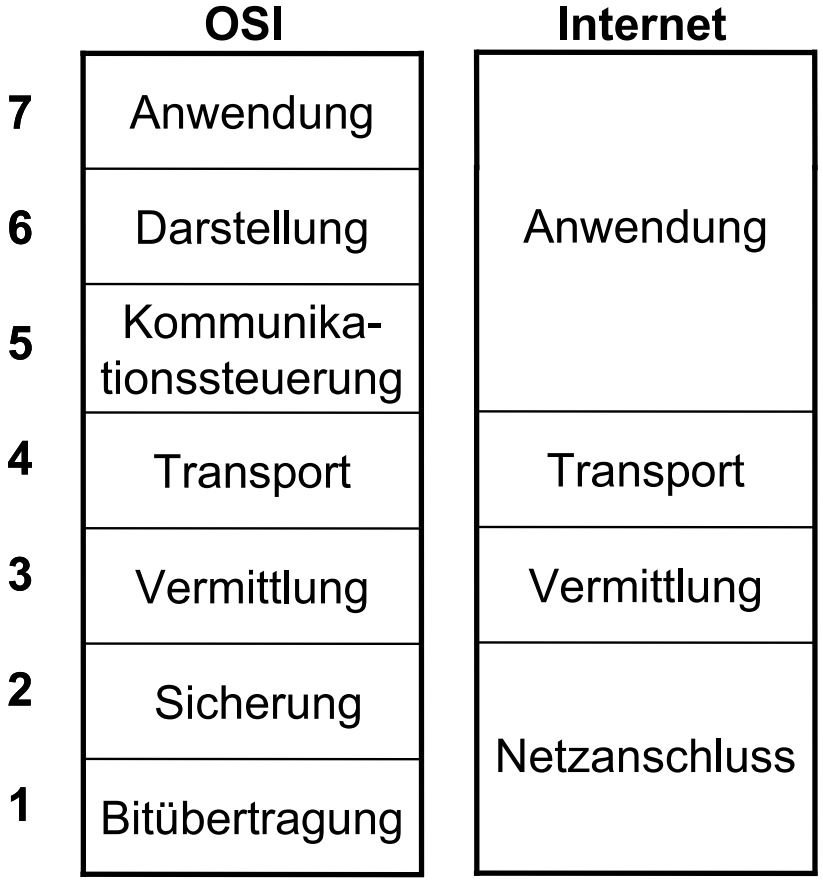
Anwendungsschicht im Prinzip, ISO/OSI vs. Internet, DNS
Wiederholung

Anwendungsschicht im Prinzip

- **Aufgabe**: Allgemein verwendbare Dienste werden als Protokolle spezifiziert und standardisiert.

- Anwendungen. . .
  - werden auf vernetzten Rechnern ausgeführt
  - tauschen Nachrichten aus, um einen Dienst zu erbringen
- Protokolle der Anwendungsschicht. . .
  - sind Teil der Anwendungen
  - nutzen darunterliegende Protokolle
  - spezifizieren die Art der ausgetauschten Nachrichten
  - spezifizieren die Aktionen, die auf Nachrichten folgen

ISO/OSI vs. Internet (Wiederholung)



- Die Anwendungsschicht im Internet-Modell umfasst die ISO/OSI Schichten 5 bis 7.
- Internet-Anwendungen (Gegenstand dieses Kapitels) müssen also auch Aufgaben der Schichten 5 und 6 übernehmen.

Dienstgütekriterien ausgewählter Anwendungen

Anwendung	**Verlust**	**Übertragungsrate**	**Verzögerung**
File Transfer	empfindlich	elastisch	tolerant
E-Mail	empfindlich	elastisch	tolerant
WWW	empfindlich	elastisch (wenige Kbps)	tolerant
Audio/Video (Echtzeit)	tolerant	Audio: wenige Kbps bis 1 Mbps Video: 10 Kbps bis ca. 15 Mbps	empfindlich, einige hundert Millisek.
Audio/Video (gespeichert)	tolerant	wie oben	empfindlich, wenige Sekunden
Interaktive Spiele	tolerant	wenige Kbps bis 10 KBps	empfindlich, einige hundert Millisek.
Finanzanwendungen	empfindlich	elastisch	ja und nein

Verzeichnisdienste

Wichtige funktionale Anforderung aller Dienste:
Abbildung von Namen auf Adressen

- DNS (Domain Name System)
  - dient der Abbildung von Endsystemen auf IP-Adresse
- X.500 Directory
  - Konzept der ITU-T für ein verteiltes Directory, einschließlich Verschlüsselung und Zertifizierung
- LDAP (Lightweight Directory Access Protocol)
  - ist ein Zugriffsprotokoll für Directories gemäß X.500
  - ist weniger aufwendig als OSI X.500-DAP
  - häufiger Einsatzbereich: Nutzerauthentifizierung

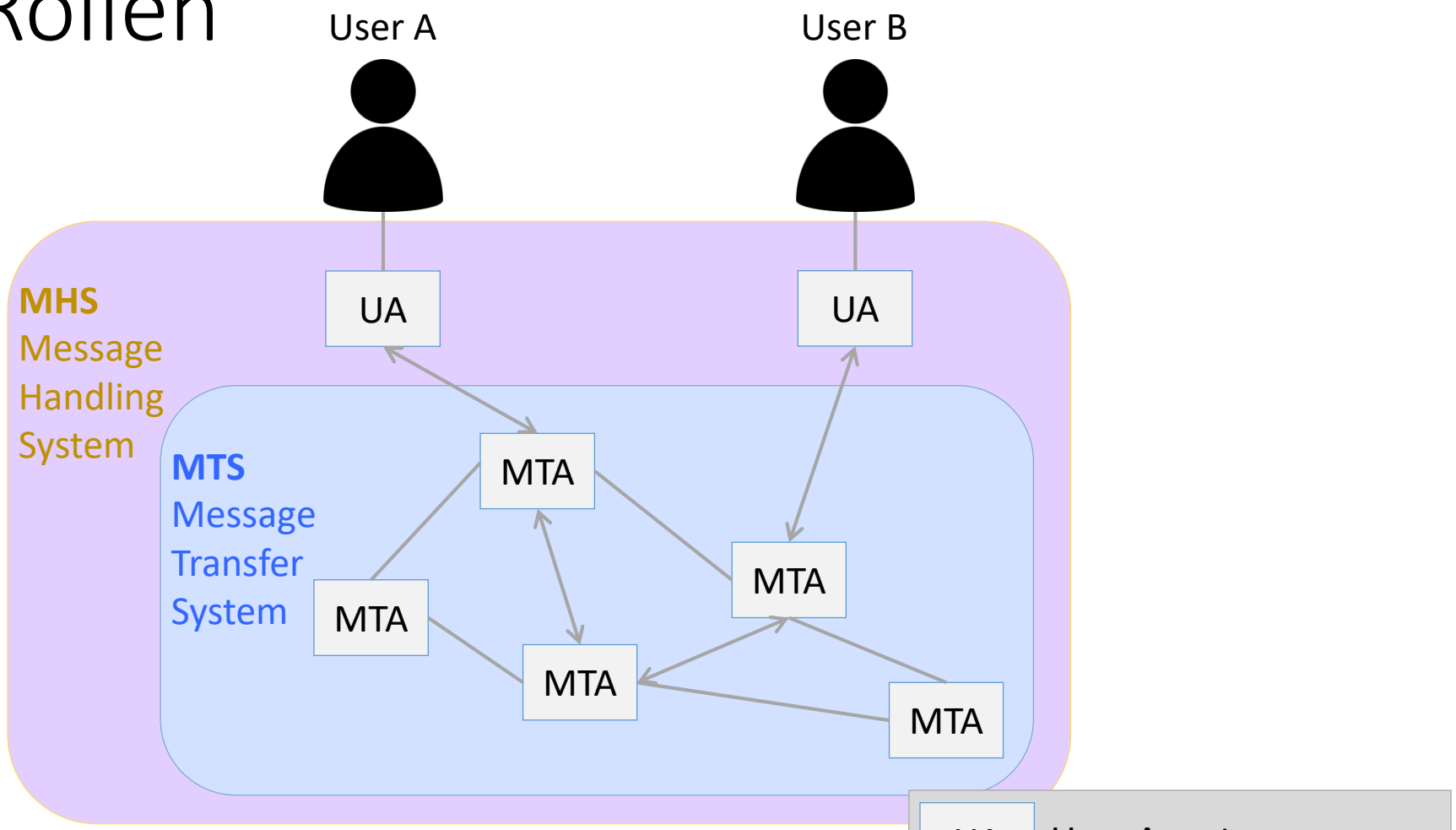
Kurze DNS Wiederholung

- DNS (engl. Domain Name System)
  ➔ Wurde im Kapitel 2 Ausführlich behandelt
- DNS bietet einen als Baum strukturierten Namensraum für Hosts im Internet.
- DNS ist als verteilte Datenbank implementiert. (Es besteht eine Hierarchie von Nameservern.)
- Der DNS Dienst bildet Domainnamen auf Resource Records (insbesondere IP-Adressen) ab.
- DNS ist ein Dienst der Anwendungsschicht im Internet.
- DNS ist kritischer Bestandteil der Infrastruktur des Internets.
- Im Vergleich zu typischen Anwendungsdiensten, wie Email, bekommen Endnutzer nur selten mit, dass sie DNS verwenden.

Kapitel 6.2
Electronic Mail (Email)

SMTP, MIME, POP, IMAP

Message Handling Systeme: Rollen



Protokolle

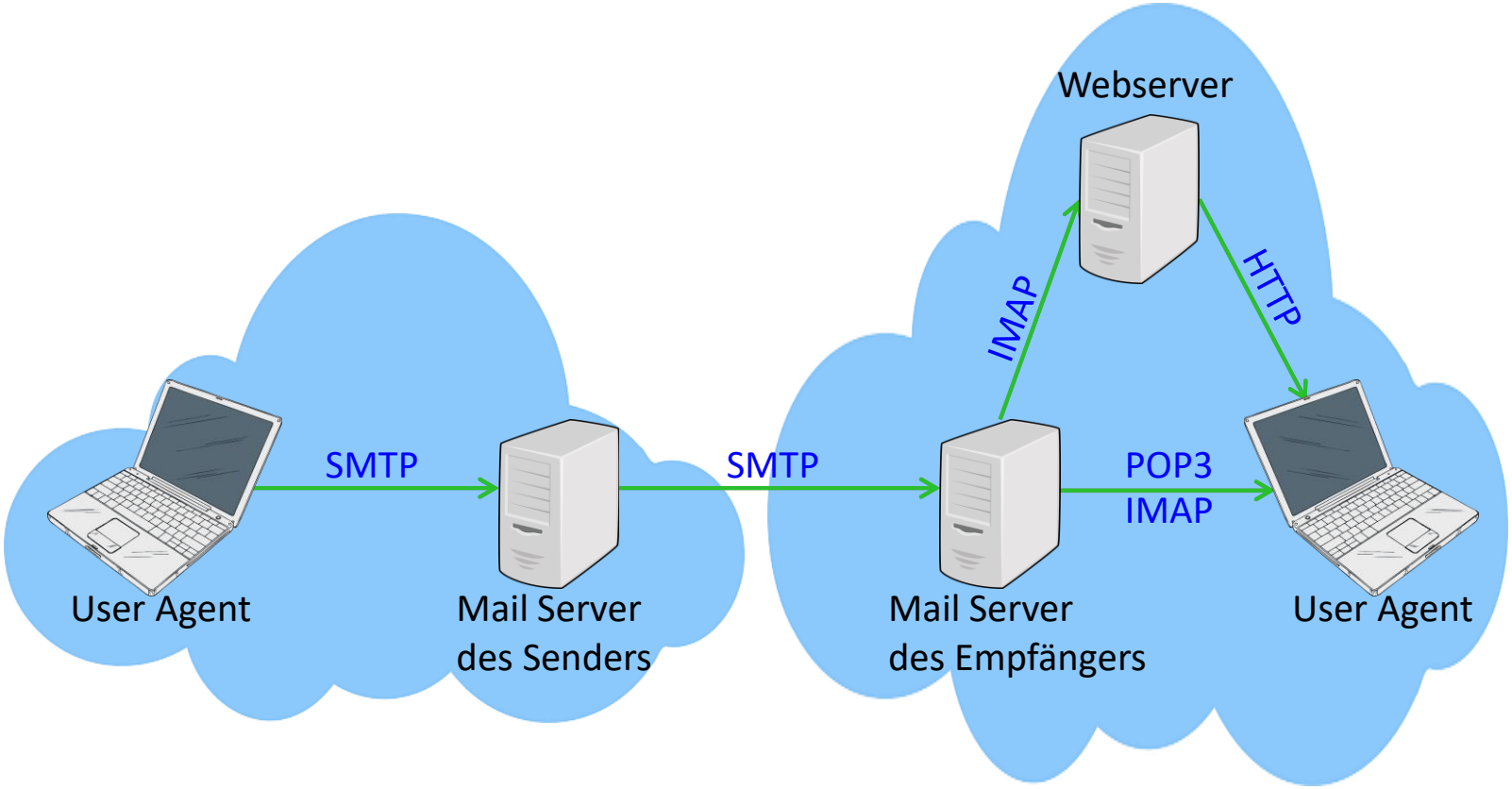
Senden von Emailnachrichten:

- SMTP (engl. Simple Mail Transfer Protocol)
  - in RFC 822 ausschließlich ASCII-Text
  - in RFC 2045/6 Erweiterung auf MIME (Binärdaten)

Abholen von Emailnachrichten:

- POP3 (engl. Post Office Protocol)
  - Authentifizierung/Autorisierung (User Agent, Server)
  - Übertragung der Nachrichten zum User Agent
- IMAP (engl. Internet Mail Access Protocol)
  - komplexer; Verwaltung mehrerer Ordner
  - Emailnachrichten können auf dem Server verwaltet werden
- HTTP (engl. Hypertext Transfer Protocol)
  - für browserbasierte Emaildienste

Protokolle (Animation)



Simple Mail Transfer Protocol (SMTP)

- Direkte, TCP-basierte (Port 25) Übertragung zwischen sendenden UA/MTA und empfangenden MTA
- spezifiziert in RFC 821
- Phasen: handshaking (greeting), transfer, closure
- Command/Response-Dialog (ASCII-basiert).
- Responses: Statuscode und Phrase.
- Relevanz der DNS-Informationen für Email
  - Problem: an welchen Host soll eine an nutzer@domain adressierte Email übertragen werden?
  - Resource Record des Typs **MX** nennt den für eine Domäne zuständigen Mailserver ➔ Emailadressen nicht notwendig an Namensraum für Rechner gebunden
  - Adressen wie vorname.nachname@unternehmen.com möglich

Beispiel: Einfache SMTP-Sitzung

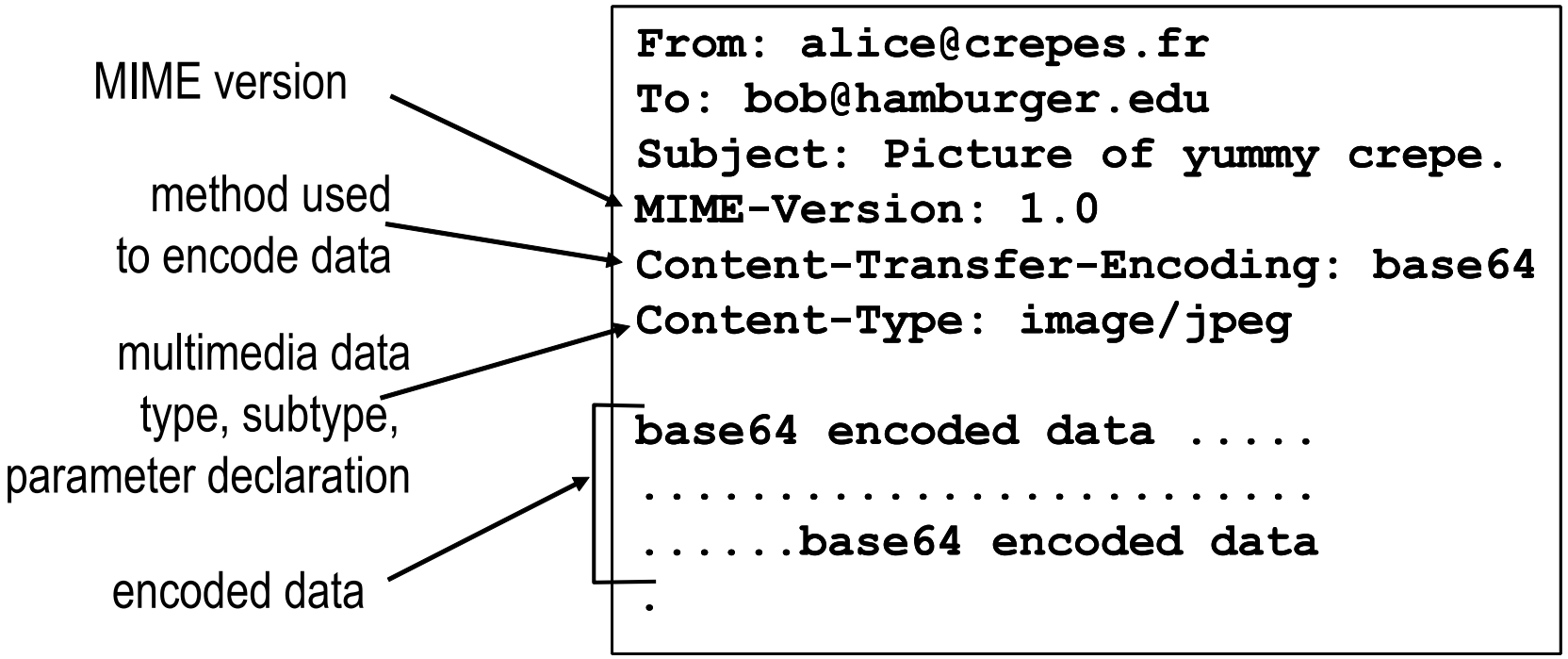
danciu@pcheger09: > **telnet mail 25** ←Server heißt „mail", Port ist 25
[. . .]
S: 220 mail.nm.ifi.lmu.de ESMTP Sendmail 8.12/Linux MNM 0.1; Mon, 28 Jan 2008 10:23:13 +0100 Unterstrichen: Statuscode
HELO nm.ifi.lmu.de Unterstrichen: Command
S: 250 mail.nm.ifi.lmu.de Hello pcheger09.in.nm.ifi.lmu.de, pleased to meet you
MAIL FROM:<danciu@nm.ifi.lmu.de>
S: 250 2.1.0 <danciu@nm.ifi.lmu.de>. . . Sender ok
RCPT TO:<rnp@nm.ifi.lmu.de>
S: 250 2.1.5 <rnp@nm.ifi.lmu.de>. . . Recipient ok
DATA
S: 354 Enter mail, end with "." on a line by itself
SMTP Probelauf. ←Text der Email
. ←Einsamer Punkt
S: 250 2.0.0 m0SJNDno027963 Message accepted for delivery
QUIT
S: 221 2.0.0 mail.nm.ifi.lmu.de closing connection
Connection closed by foreign host. ←Nachricht von telnet-Client

S:. . . :SMTP-Server **fett**: Benutzereingabe farbig: Kommentare

Nachrichtenformat

- Standard für Textnachrichten, spezifiziert in RFC 822
- Nachricht besteht aus Header, Body sowie Abschlusszeile
- Header-Zeilen (siehe auch Beispiel in Kapitel 3)
  - To, From, Subject
  - CC, BCC ([Blind] Carbon Copy)
  - Reply-To: Emailadresse, die für Antwort benutzt werden sollte
  - Message-Id: identifiziert eine Emailnachricht in späterer Kommunikation
  - In-Reply-To, References: Verweise auf Message-Ids von Emailnachrichten
  - Received: wird von jedem vermittelnden MTA dem Header hinzugefügt
- Body
  - Nutzdaten („Brief"); nur ASCII-Zeichen zulässig
  - Letzte Zeile markiert Nachrichtenende; sie enthält nur einen Punkt '.'
- Erweiterung zum Transport von Multimedia-Daten
  - MIME: multimedia mail extension, RFC 2045, 2056
  - zusätzliche Information im Header ➔ Angabe des MIME content type

Multi-purpose Internet Mail Extension (MIME)



Post Office Protocol (POP)

Autorisationsphase

- Client Befehle:
  - user: Benutzername
  - pass: Passwort
- Server Antworten: +OK, -ERR

Transaktionsphase

- list: list message numbers
- retr: retrieve message by number
- dele: delete message by number
- quit

S: +OK POP3 server ready
C: user alice
S: +OK
C: pass hungry
S: +OK user successfully logged on

C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: quit
S: +OK POP3 server signing off

Internet Mail Access Protocol (IMAP)

- Problem: Mit POP3 können Nachrichten nur abgeholt, aber nicht auf dem Server verwaltet werden.
  ➔ Schlechte Unterstützung für nomadische Nutzer
- IMAP: Verwaltung von Emailnachrichten auf dem Server
  - Hierarchie von Ordnern (folders), die Nachrichten enthalten (wie Dateisystem)
  - Abrufen, Verschieben (zwischen Ordnern), Löschen von Nachrichten durch UA
  - Selektive Übertragung von Teilen von Emailnachrichten (z.B. nur Header)
- Last / Performanz
  - Verwaltungsoperationen auf Server ausgeführt ➔ belastet Servermaschine
  - Selektive Übertragung: Abruf nur relevanter Nachrichtenteile, z.B. über Verbindung mit niedriger Übertragungsrate

Kapitel 6.3
World Wide Web (WWW)

HTML, URL, HTTP

Grundlagen (1/2)

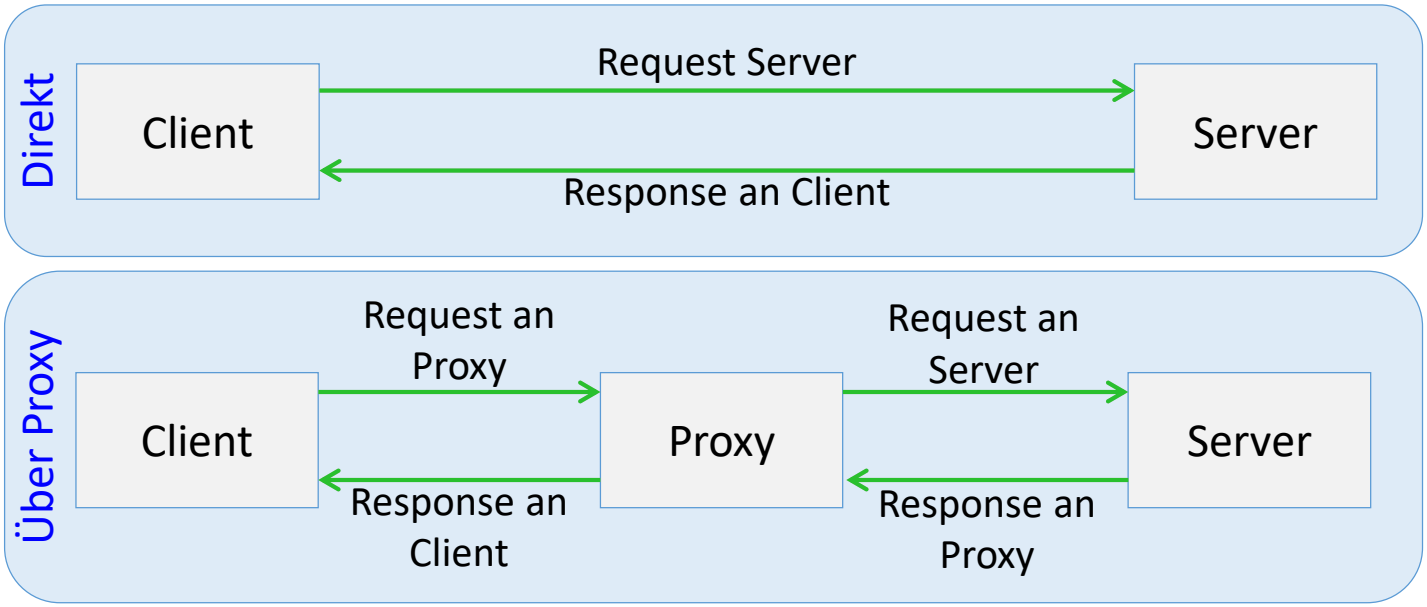
- Das World Wide Web (kurz Web) ist ein System durch *Hyperlinks* verknüpfter Web-Ressourcen (Webseiten/Hypertext-Dokumente), welches über das Internet zugänglich ist (mit Hilfe von Protokollen wie HTTP).
- HTML (engl. Hypertext Markup Language) zur Formatierung von Dokumenten
- URL (engl. Uniform Ressource Locator) zur Referenzierung von Objekten (Adressierung)
- HTTP (engl. Hypertext Transfer Protocol) zur Übertragung über das Internet

Grundlagen (2/2)

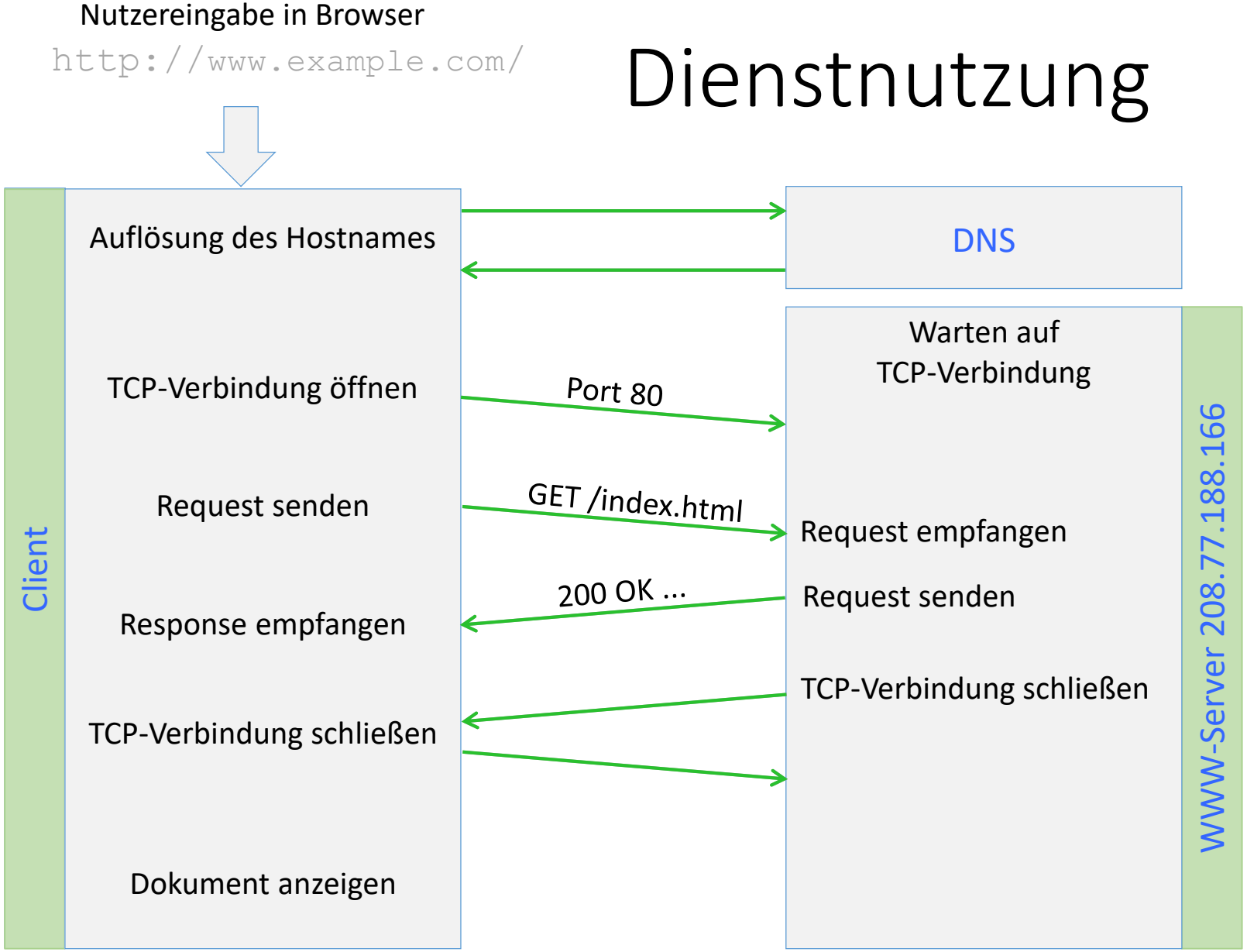
- Hypertext Markup Language (HTML): Beschreibungssprache für Web-Dokumente
  - standardisiert als W3C Recommendation, z.B. HTML 5
  - Dokumente bestehen aus Header (Titel, Formatierung, Metadaten) und Body (Inhalt)
  - Links durch Anker (anchor) angegeben, die eine URL enthalten.
  - Hilfen: Cascading Style Sheets (CSS), Scalable Vector Graphics (SVG)
- Referenzierung von Objekten (Adressierung)
  - **URI** (U. R. Identifier) RFC 1630: Objektbegriff für URL und URN
  - **URN** (U. R. Name): global eindeutiger langlebiger logischer Name für Objekt (ohne Lagerort)
  - **URL** (Uniform Resource Locator) RFC 1738: Lagerort eines Web-Dokuments durch Serverangabe und Pfadbezeichnung
  - URL Syntax:
    <protokoll>**://**[<user>[**:**<passwd>]**@**]<host>[**:**<port>]**/**[<path>]
  - z.B. http://www.example.com/documents/index.html
  - Optionale (und eher seltene) Felder: user, passwd, port

Hypertext Transfer Protocol (HTTP)

- Protokoll zum Transport von Anfragen/Objekten zwischen Browser, Server und Proxy (Zwischensystem)
- TCP-basiert, Port 80 (Proxy: typischerweise Port 8080)
- HTTP ist zustandslos, pull-orientiert, unterstützt bidirektionale Übertragung und Caches im Client bzw. Proxy
- HTTP/1.1 spezifiziert in RFC 2068, 2616



Dienstnutzung



Request-Methoden und Statuscodes in HTTP/1.1

Methoden/Dienstprimitive

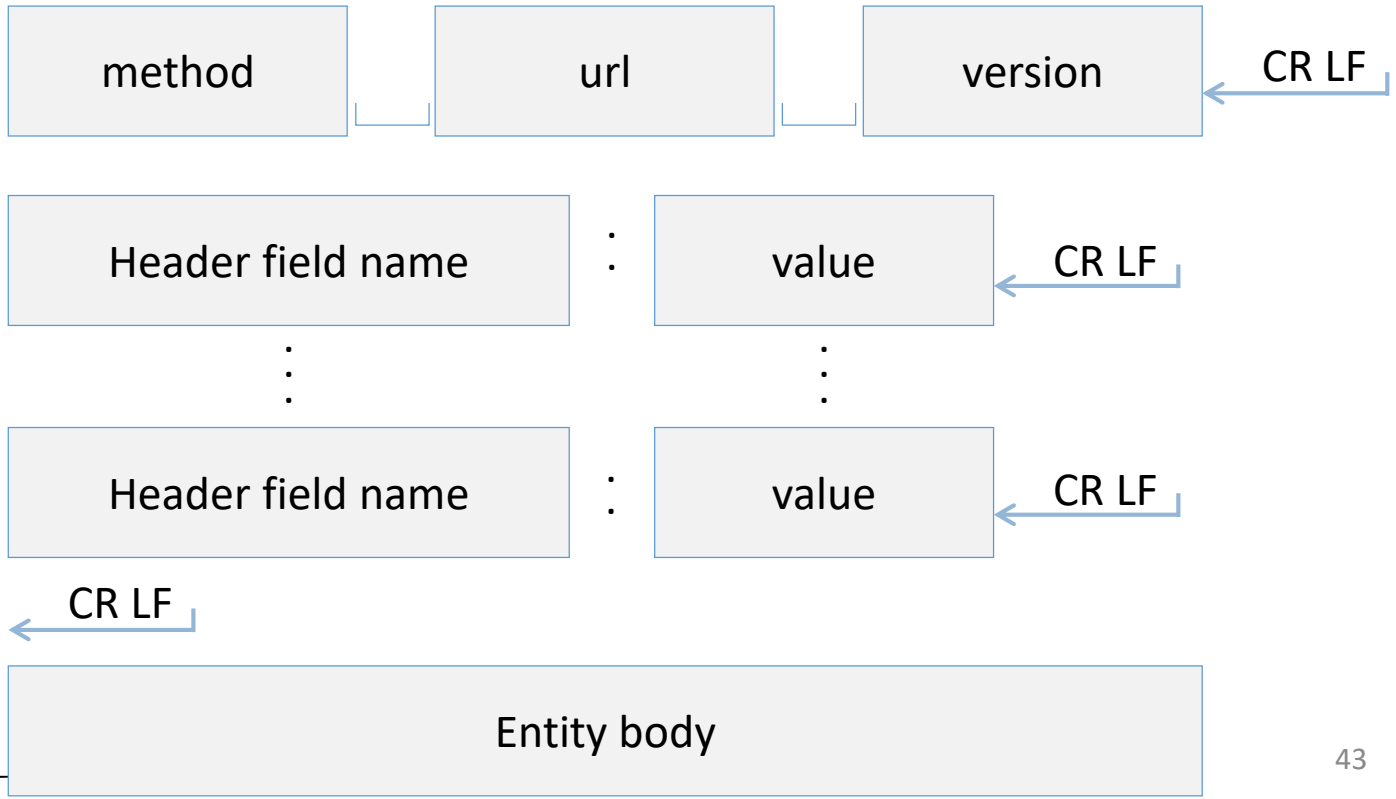
- GET: Abruf einer Datei vom Server.
- HEAD: Abruf von Metadaten über eine Datei.
- POST: Übertragen von Daten an den Server.
- PUT: Ablegen einer Datei auf Server
- DELETE: Löschen einer Datei auf dem Server.
- OPTIONS: Abfrage von Informationen über Kommunikationsoptionen.
- TRACE: für Testzwecke.

Statuscodes

- 1xx: Informational
  100 Continue
  101 Switching Protocols
- 2xx: Successful
  200 OK
  206 Partial Content
- 3xx: Redirection
  301 Moved Permanently
  302 Found
  307 Temporary Redirect
- 4xx: Client Error
  400 Bad Request
  401 Unauthorized
  403 Forbidden
  404 Not Found
- 5xx: Server Error
  500 Internal Server Error

Request Grammatik

- version: Protokollversion
- header field name: Name einer Variablen
- value: Wert einer benannten Variablen
- entity body: transportiert eine Nachricht als Teil des Dienstaufrufs



Request-Response Beispiel

Request (Client an Server)

GET /somedir/page.html HTTP/1.1 (Methode und Objekt)
Connection: close (Verbindung nach Response schließen)
User-agent: Mozilla/4.0 (Eigenschaften des Clients)
Accept: text/html, image/gif,image/jpeg
Accept-language:fr

Response (Server an Client)

HTTP/1.1 200 OK (Statuscode Numerisch und Klartext)
Connection: close
Date: Thu, 06 Aug 1998 12:00:15 GMT
Server: Apache/1.3.0 (Unix) (Servertyp)
Last-Modified: Mon, 22 Jun 1998 ...
Content-Length: 6821
Content-Type: text/html (Hinweis zum Typ des Objekts)
data data data data data ... (Objektdaten)

HTTP als zustandsloses Protokoll

- Vorteile: einfach und fehlerunempfindlich
- Nachteil: viele Dienste benötigen Zustandshaltung
  - Dienstsitzung besteht aus **mehreren Request/Response-Paaren** (Schritten) — z.B. Buchung einer Fahrkarte
  - Dienststatus muss **über alle Schritte** erhalten werden
- Abhilfe (zum Standard erhobene Notlösungen)
  - Statusvariablen in URL werden per GET-Methode übertragen
  - Cookies: clientseitige Speicherung einer Zeichenkette
  - SessionID: Vergabe eindeutiger Kennung für eine Dienstsitzung (oft in URL); Status serverseitig gespeichert.

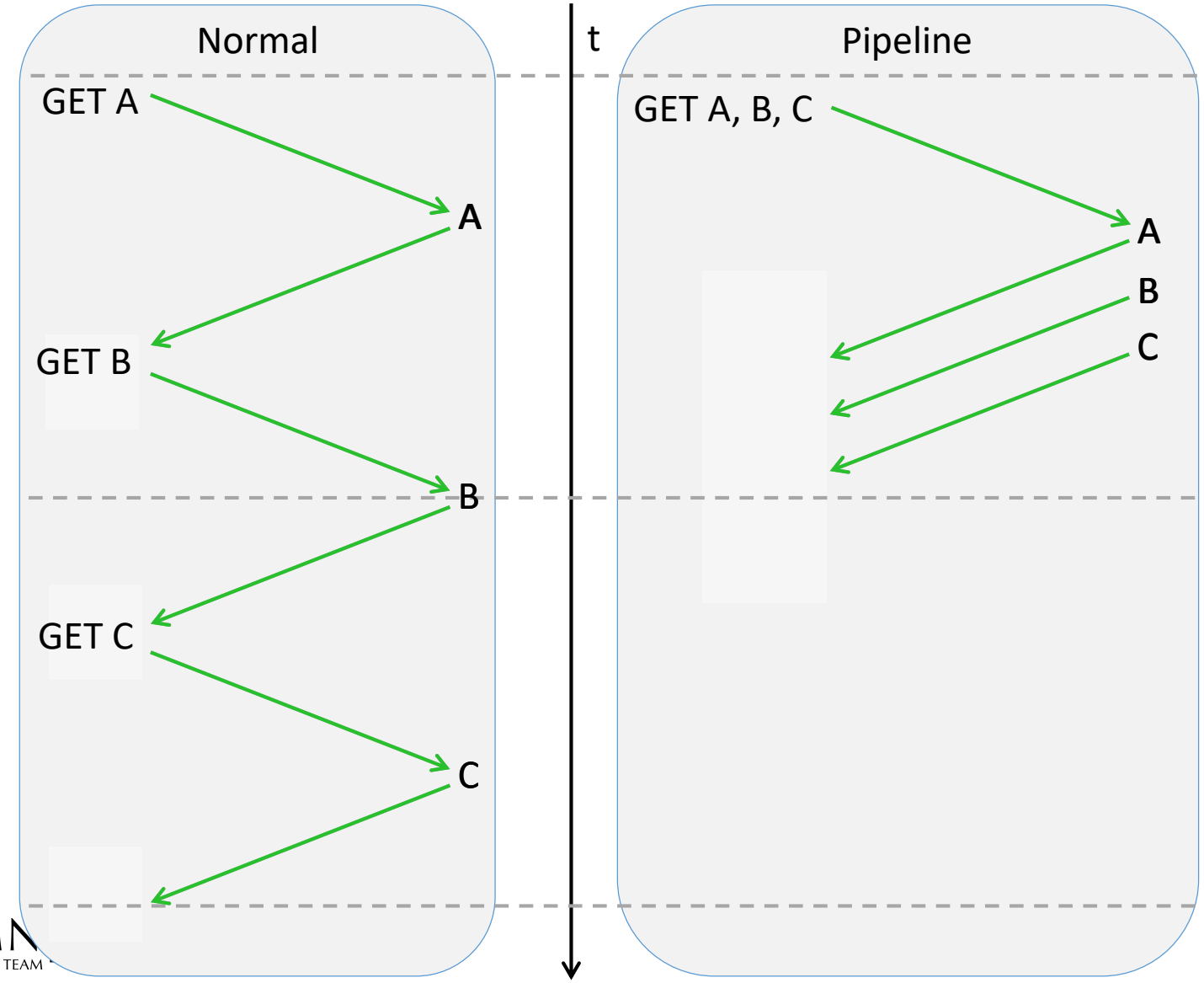
Sicherheit

- Authentifizierung
  - IP-Adresse (nicht praktikabel)
  - Kennung/Passwort (Basic Authentication) oder kryptographische Zertifikate
- HTTPS: Verschlüsselte Variante von HTTP (RFC 2660)
  - meist zusammen mit Serverauthentifizierung (Zertifikat)
  - Übertragung über Transport Layer Security (TLS), Secure Socket Layer (SSL)(Port 443)

Leistungssteigerung

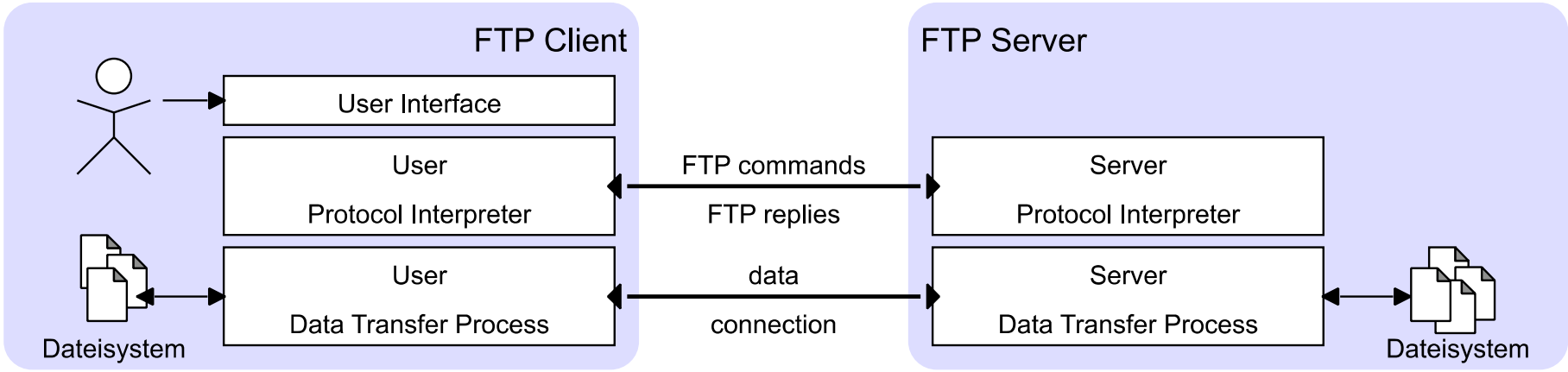
- Ziele: schnelle Anzeige von Webseiten; geringe Netzlast
- Einfachster Fall
  - Eine TCP-Verbindung zum Server pro Request/Response-Paar
  - Nach Request wird auf Response gewartet.
  - Client ruft Objekt jedes Mal vollständig vom Server ab.
  - Verbindung wird nach Interaktion geschlossen.
- Maßnahmen zur Leistungssteigerung
  - Mehrere gleichzeitige Verbindungen: Parallelisierung der Anfragen
  - Persistente Verbindung: Verbindung wird für mehrere Anfragen wieder benutzt ➔ Einsparung des Verbindungsaufbaus („Connection: keepalive“)
  - Pipelining: Client schickt mehrere Requests nacheinander; Server schickt entsprechende Responses.
  - Caching: Client verwaltet lokales Cache jüngst abgerufener Objekte.
  - Conditional GET: Objekt wird nur übertragen, falls neuer als Cache-Version.
  - Caching proxy: Anfragen werden durch Proxy geleitet. Proxy verwaltet Cache ➔ gemeinsames Nutzen des Cache

Pipelining



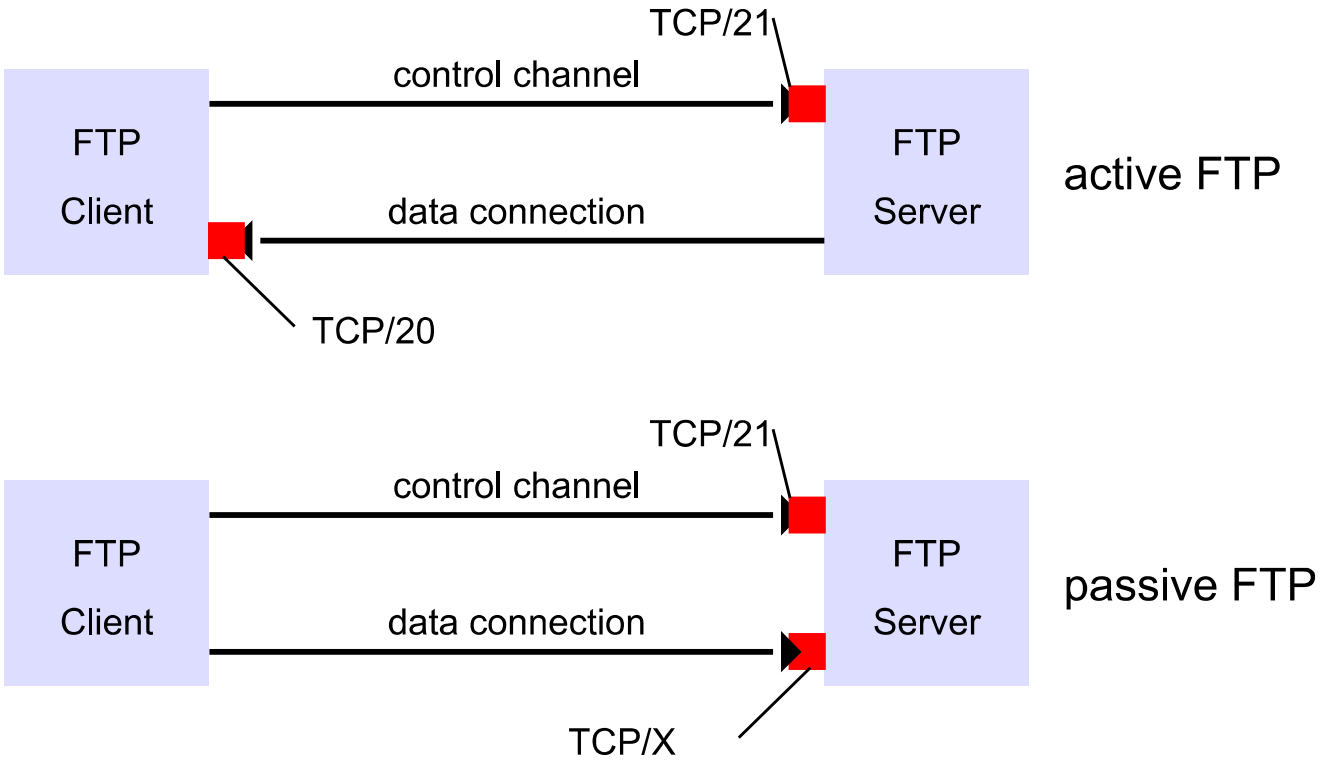
Kapitel 6.4
File Transfer Protocol (FTP)

Überblick



- Übertragung von Dateien zwischen zwei Hosts (RFC 959)
- control channel (TCP, Port 21): Befehle, Antworten
  - out-of-band Befehle (➔ nicht verwechselbar mit Nutzdaten)
- data connection (TCP): Übertragung von Dateien (bidirektional)
  - nur für Dateiübertragung geöffnet
  - aktives FTP: Server öffnet data connection
  - passives FTP: Client öffnet data connection

Aktiver und passiver Modus



In durch Paketfilter („Firewall“) geschützten Umgebungen ist passives FTP oft vorteilhaft

Auswahl von Befehlen und Statuscodes

- Authentifizierung
  - USER <Benutzername>
  - PASS <Passwort>
- Verbindung
  - PORT <Nr.>: clientseitig
  - PASV: passiver Modus
  - QUIT: abmelden
- Umgang mit Dateien
  - CWD: Change Working Directory
  - MKD: Verzeichnis anlegen
  - RMD: Verzeichnis löschen
  - LIST: Dateiliste abrufen
  - RETR: Datei abrufen
  - STOR: Datei ablegen

- 331 username OK, password required
- 125 data connection already open; transfer starting
- 425 can’t open data connection
- 452 error writing file

Fragen zu Kapitel 6

- Ohne welche zusätzliche Internetanwendung funktionieren Mail, Filetransfer, Web nicht?
- Kann man HTTP bzw. SMTP als verbindungslos oder verbindungsorientiert beschreiben?
- Welche Hauptbausteine machen ein Emailsystem aus? Welche Protokolle werden zwischen den Bausteinen benutzt?
- Welche Hauptkomponenten machen ein Websystem aus?
- Wie ist eine URL aufgebaut?
- Realisieren SMTP bzw. HTTP ein Push- oder Pull-Modell?