

## Merkmale von Netzwerken

- verbinden mehrere Rechner (und andere Hardware-Komponenten – *Hosts*)
- Funktion: Kommunikation, Datenaustausch, Zentralisierung von Diensten auf Servern
  - Datenserver
  - NIS- oder LDAP-Server
  - Mailserver
  - Printserver, ...
- Ziele: Vereinfachung der Administration, Senkung von Kosten
- *Client-Server-Prinzip*

- verschiedene Medien
  - Koaxialkabel
  - Twisted Pair
  - Glasfaserkabel
  - Funk (WLAN, Internet-Provider)
  
- verschiedene Topologien (physikalisch wie logisch)
  - Bus (z.B. Ethernet mit bis zu 10 Mbps; *paketorientiert*)
  - Stern (z.B. Fast Ethernet mit 100 Mbps; *paketorientiert*)
  - Token-Ring (z.B. IBM, FDDI; *paketorientiert*)
  - Hochgeschwindigkeitsnetz ATM (*verbindungsorientiert*)
  
- Zusammenschluss mehrerer Teilnetze zu größeren Netzen (mit *Routern* als Vermittlungsstellen)

# Internet

- Netz der Netze
- logische Verbindung aller Hosts, die Daten vom Netz nehmen und/oder auf das Netz geben können
- Sammlung von Diensten
- dezentralisierte Architektur militärisch motiviert (frühe 70-er Jahre)
- anfangs meist akademisch genutzt, seit Anfang der 90-er Jahre kommerziell
- Jeder Host kann jeden anderen Host direkt ansprechen, dessen *Internet-Adresse* (*IP-Adresse*) er kennt.

## Adressierung im Internet

- IP-Adresse: weltweit eindeutige 4-Byte-Zahl
- Bytes als durch Punkte getrennt Dezimalzahlen, z.B.: 141.89.54.4
- vordere Bytes bezeichnen Netze, hintere bezeichnen Rechner
- Daneben gibt es symbolische Namen für die Hosts:  
    `delenn.babylon.cs.uni-potsdam.de.`  
    *Rechnername.Domäne. ... .Toplevel-Domäne.*
- hierarchische Ordnung der Domänen (*domains*)
  - Der letzte `.` bezeichnet die Wurzel der baumartigen Hierarchie (*Root*; kann meist weggelassen werden).
  - *Toplevel-Domain*: Länder- oder Funktionsbezeichner (z.B. `de`, `au`, `com`, `edu`, `org`, `net`, `mil`, `gov`)

## DNS (Domain Name Service)

- Zuordnung von symbolischen Internet-Adressen zu IP-Adressen durch sog. *Nameserver*

`nslookup IP-Adresse`      oder      `nslookup Internet-Adresse`

- Konfigurationsdatei: `/etc/resolv.conf`
  - Domänname des Rechners
  - Nameserver und Backup-Nameserver
  - `search` Liste automatischer Domännamen-Expandierungen
- Kennt ein Nameserver die Adresse nicht, so wird beim Nameserver der nächst übergeordneten Domäne angefragt, zuletzt beim *Root-Server*.

## Realisierung des Client-Server-Modells

- Client:*
- Prozess, der (von einem anderen Prozess) eine Dienstleistung wünscht / Verbindung aufbaut
- Server:*
- Prozess, der Dienstleistungen erbringt und wartet, bis er eine Anfrage von einem Client erhält / eine Verbindung annimmt
  - wird meist schon beim Systemstart als Hintergrundprozess erzeugt (*Dämon* oder *daemon*)
- Port:*
- Adresse eines Prozesses auf einem Host
  - für die wichtigsten Prozesse auf allen Hosts einheitlich
  - `/etc/services`
- `inetd:`
- „Internet Super Server“: überwacht alle Ports
  - bei Verbindung an einem Port wird zugehöriger Daemon gestartet
  - `/etc/inetd.conf`

## Protokolle

- regeln die Kommunikation der beteiligten Hard- und Softwarekomponenten
- legen Regeln über die Art und Weise dieser Kommunikation fest (Format und Reihenfolge der auszutauschenden Nachrichten sowie die Handlungen beim Senden und/oder Empfangen der Nachrichten)
- sind in Schichten organisiert, z.B.:

Anwendungsschicht	regelt Kommunikation der Prozesse einer Anwendung
Transportschicht	organisiert Datenaustausch dieser Prozesse
Vermittlungsschicht	Weiterleitung von Datenpaketen von Host zu Host
Sicherungsschicht	Weiterleitung von Router zum nächsten Router auf dem Weg von der Quelle zum Ziel
Bitübertragungsschicht	technische Verbindung und Kodierung etc.

- *Beispiele* für Protokollfamilien: XNS, Novell-Netware, Internet-Protokolle

## Internet-Protokolle (TCP/IP)

- herstellerunabhängig; seit 1982 in UNIX eingebunden, seither auch in Betriebssystemen von IBM, DEC, Microsoft
- auf verschiedenen physikalischen Netzen implementiert (Ethernet, Token-Ring, Token-Bus, ...)
- Verwendung für alle wesentlichen Netzdienste, insbesondere

### Netzwerkanwendung

Remote-Login (telnet, ssh)

File-Transfer (ftp, sftp)

Web-Anwendungen

Electronic Mail

andere

### Beispielprotokoll

TELNET

FTP

HTTP

SMTP

NFS, X-WINDOW, ...



## Wichtige Protokolle der TCP/IP-Protokollfamilie

- IP**
- Internet-Protocol (Vermittlungsschicht)
  - versendet die Daten als Datenpakete (*Datagramme*)
  - verbindungsloses Protokoll
  - legt für jedes Datagramm einen Weg durch das Netz fest (*Routing*)
- ICMP**
- Internet-Control-Message-Protocol, Teil jeder IP-Implementierung
  - sichert Erreichbarkeit des Empfängers und übt Flusskontrolle aus
  - `ping [-s] Hostname`: versendet Test-Datagramme als Echo-Auftrag

- TCP**
- Transmission-Control-Protocol (Transportschicht)
  - verbindet die kommunizierenden Prozesse auf den Hosts (Punkt-zu-Punkt-Verbindung für die gesamte Kommunikation)
  - zerlegt die Daten in Datagramme (Pakete) und adressiert sie mit
    - IP-Adressen von Sender- und Empfänger-Host,
    - Sender- und Empfänger-Portnummern
  - übergibt die adressierten Datagramme an IP
  - garantiert die vollständige und korrekte Zustellung (quittiert Empfang jedes Pakets, sendet ggf. Pakete erneut)
- UDP**
- User-Datagram-Protocol (Transportschicht)
  - minimale, verbindungslose Variante von TCP (schneller)
  - garantiert nicht die Zustellung und Reihenfolge der Pakete
  - Einsatz bei NFS (Mounten), Multimedia im WWW, ...

## Remote Login

- `telnet adresse`
- `ssh [-l user] adresse` (secure shell)
- startet Sitzung auf einem entfernten Rechner, beginnt mit Login
- Achtung! Bei `telnet` werden alle Daten unverschlüsselt übertragen.  
~> `ssh` bevorzugen!
- `ssh` trägt nach Rückfrage den Schlüssel in `~/.ssh/known_hosts` ein
- Beendigung durch `exit`
- `ssh`-Verbindungen werden erst beendet, wenn keine Ein- oder Ausgabedaten von Prozessen mehr verloren gehen können.  
~> „Hängen“ von `exit` bei aktiven Jobs

## Dateien übertragen

- [s]ftp [*user*]@*adresse* (file transfer protocol)
- Passwortabfrage, Ausgabe eines Prompt: sftp>
- ftp unverschlüsselt, sftp verschlüsselt
- Herunterladen mit `get Datei`
- Heraufladen mit `put Datei`
- Befehlsübersicht mit `help` oder `?`
- Ende mit `quit`
- Übertragung einzelner Dateien mit scp (secure copy)
  - `scp lokale_Datei [user@]adresse:absoluter_Pfad`
  - `scp f08.pdf rlehre@delenn.babylon:/home/rlehre/f08.pdf`oder umgekehrt

# XWindows

- Offenes, plattformunabhängiges Client-Server-System zur
  - Ansteuerung graphischer Bildschirme (im Allgemeinen)
  - Verwaltung fensterbasierter graphischer Benutzeroberflächen
- *Display* ist ein Rechner mit (mehreren) Bildschirmen
- *XServer* ist ein Prozess, der ein Display kontrolliert und steuert
- *XClients* sind Prozesse, die eine graphische Ausgabe an einem Display erzeugen wollen (auf demselben oder einem anderen System)
- `~/.Xdefaults` ist eine Datei, die graphische Ausgaben von *XClients* konfiguriert

## Umlenkung der Ausgaben von XClients

- Die Umgebungsvariable DISPLAY speichert den Bildschirm, auf den der XServer die graphischen Ausgaben der XClients lenkt
  - 0.0 ist der Hauptbildschirm des lokalen Rechners
  - andere Nummern bezeichnen andere Bildschirme (z.B. Terminals)
  - *Internet-Adresse: Bildschirmnummer*  
z.B. `delenn.babylon.cs.uni-potsdam.de:5.0`  
für Bildschirme des Rechners mit *Internet-Adresse*
- `xhost [+|-] [Internet-Adresse]`  
erlaubt|verbietet dem Host mit *Internet-Adresse*, X-Ausgaben auf den Bildschirm des Benutzers zu lenken

## Weitere wichtige Dienste

### 1. Archivierungen mit tar

- `tar -cf Archivdatei Dateiliste`  
     $\rightsquigarrow$  fasst Dateien der *Dateiliste* zur *Archivdatei* zusammen
- `tar -tf Archivdatei`  $\rightsquigarrow$  Inhaltsangabe
- `tar -xf Archivdatei`  $\rightsquigarrow$  Dearchivieren

### 2. Komprimierungsdienste

<code>compress Datei</code>	$\rightsquigarrow$	<code>Datei.Z</code>	(uncompress)
<code>gzip Datei</code>	$\rightsquigarrow$	<code>Datei.gz</code>	(gunzip)
<code>zip Zip-Datei Dateiliste</code>	$\rightsquigarrow$	<code>Zip-Datei.zip</code>	(unzip)

### 3. Formatumwandlung von Binärdatei in ASCII-Datei

`uuencode Datei Datei > Datei.uu`  $\rightsquigarrow$  `Datei.uu` (`uudecode`)

### 4. Formatumwandlung DOS- in UNIX-Datei und umgekehrt

`dos2unix DOS_Datei [UNIX_Datei]` oder `fromdos DOS_Datei`  
`unix2dos UNIX_Datei [DOS_Datei]` oder `todos UNIX_Datei`