TCP/IP-Grundlagen

Veranstaltung im Rahmen der Übung zu Telematik-Anwendungen
im WS 2000/2001 am Lehrstuhl für Wirtschaftsinformatik
insbes. Informationsmanagement der Universität zu Köln

Erstellt von:
Dipl. Wirt. Inform. Henning Baars
Tel.: 470-5323
E-Mail: baars@wi-im.uni-koeln.de
Sprechstunde: dienstags, 14.00 - 15.00 Uhr

TCP-IP-Grundlagen

1. Datenübertragung mit TCP/IP
- 1.1 Was ist TCP/IP?
- 1.2 Die TCP-IP-Protokollarchitektur

2. IP und ICMP - Protokolle der Internet-Schicht
- 2.1 Aufgaben des IP-Protokolls
- 2.2 Adressierung mit IP
- 2.3 Aufbau eines IP-Datagramms
- 2.4 Routing im IP-Protokoll
- 2.5 Multiplexen nach Protokollen
- 2.6 Das Internet Control Message Protocol: ICMP
- 2.7 Grenzen von IPv4 – IPv6 als Lösungsansatz

3. TCP und UDP - Protokolle der Transport-Schicht
- 3.1 Ports und Sockets
- 3.2 UDP
- 3.3 TCP

4. Namensauflösung mit TCP/IP
- 4.1 Namensgebung im Internet
- 4.2 Der Domain-Name-Service (DNS)

5. Woher kommen Standards, Nummern und Namen?
- 5.1 Standardisierung von Internet-Protokollen
- 5.2 Namen und Nummern
- 5.3 Sonstige Organisationen

TCP-IP-Grundlagen
1. Datenübertragung mit TCP/IP

1.1 Was ist TCP/IP?
1.2 Die TCP-IP-Protokollarchitektur

1.1 Was ist TCP/IP? (1)

TCP/IP steht für eine **Familie von Protokollen** zur paktetvermittelten Übertragung von Daten zwischen Rechnern oder Anwendungsdiensten.

Dabei werden neben den eigentlichen **Protokollen zur Datenübertragung** (IP, ICMP, TCP, UDP, DNS,...) bzw. zur Konfiguration und Überprüfung der Netzknoten (z.B. SNMP) auch **Anwendungsnahe Protokolle** bereitgestellt, z.B.

- für **E-Mail** (SMTP, POP3, IMAP)
- für **Terminaldienste** (TELNET)
- für **Filetransfer** (FTP)
- für die **Übertragung von Webseiten** (HTTP)
- etc.

Daneben gibt es eine Reihe weiterer **Internet-Standards**, die mit TCP/IP in Zusammenhang stehen:

- z.B. Formate für die **Strukturierung und Darstellung von Webseiten** (HTML, CSS) oder
- z.B. Metaformate für die **Definition von strukturierten Inhalten** (XML)
- etc.

Die Spezifizierungen der Protokolle und Standards werden über sogenannte **Requests for Comments** (RFCs) veröffentlicht.

1.1 Was ist TCP/IP? (2)

TCP/IP definiert Protokolle, die **oberhalb der Schichten 1 (physical layer) + 2 (data link layer)** des ISO/OSI-Schichtenmodells liegen

Nicht zuständig ist TCP/IP damit u.a. für folgende Fragen:

- Welches Medium wird verwendet (z.B. Glasfaser oder verdrilltes Kupfer-Kabel)?
- Wie werden die Signale erzeugt (welche Spannung, welche Frequenz etc.)?
- Wie laufen die Signale physisch durch das Medium?
- Welche physischen Kopplungskomponenten kommen zum Einsatz: Hubs, Switches, Repeater oder Bridges?
- Wie werden Signale dargestellt und unterschieden (Codierung)?
- Basisbandverfahren (Signale direkt auf Leitung) oder Breitbandverfahren (Signale auf Trägerfrequenz aufmoduliert und parallel über mehrere „Kanäle" übertragen)?
- Überprüfung, dass die Daten zwischen den physikalischen Knoten korrekt übertragen werden
- Welches Übertragungsverfahren gewählt wird (z.B. Token-Passing,...)
- etc.

TCP/IP setzt hier auf vorhandenen Protokollen und Standards auf, z.B. auf Ethernet, Token-Ring, ISDN etc.

Nicht mehr zuständig ist TCP/IP für die Anwendungen, die die Datenübertragung nutzen, z.B. für Fragen der

- Anwendungs-Architektur
- Definition von Oberflächen oder Bedienabläufen
- u.w.s.

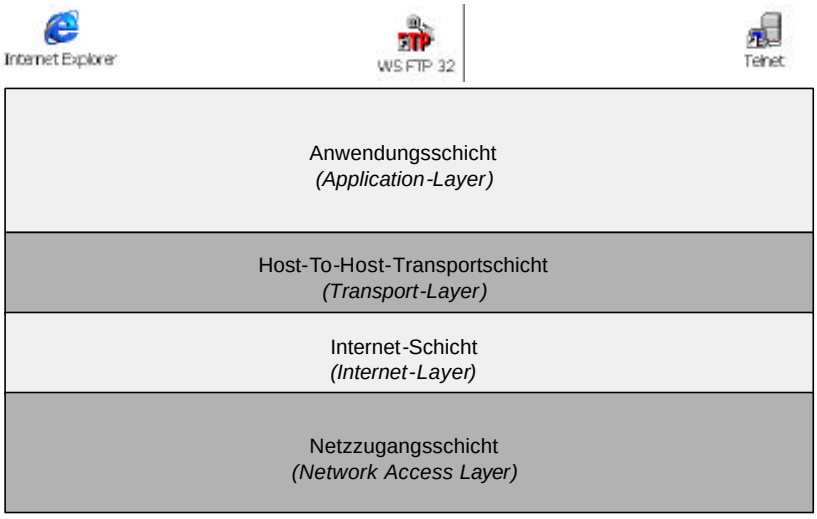
1.2 Die TCP-IP-Protokollarchitektur (1)

Keine vergleichbar klar festgelegte Schichteneinteilung wie bei OSI/ISO!

- **Bitübertragungsschicht**
  definiert Routinen für den Zugriff auf physikalische Netze
  ✍ setzt auf vorhandene Protokolle der ISO/OSI-Sicherungsschicht (data-link-layer) auf
  z.B. Protokolle zum Senden von IP-Datagrammen über Ethernet-Netze
- **Internetschicht**
  definiert den Aufbau von Datagrammen und routet Daten
  ✍ entspricht grob der ISO/OSI-Vermittlungsschicht (Network-Layer)
  wichtigstes Protokoll: IP
- **Transportschicht**
  stellt Ende-zu-Ende-Datendienste zur Verfügung
  ✍ übernimmt (grob) Funktionen, für die die ISO/OSI-Schichten Transportschicht (transport layer) und Kommunikationssteuerungsschicht (session layer) vorgesehen sind
  wichtigste Protokolle: TCP und UDP
- **Anwendungsschicht**
  enthält Anwendungsprozesse, die auf das Netzwerk zugreifen
  z.B. DNS, HTTP, SNMP, SMTP, TELNET, POP3 etc.

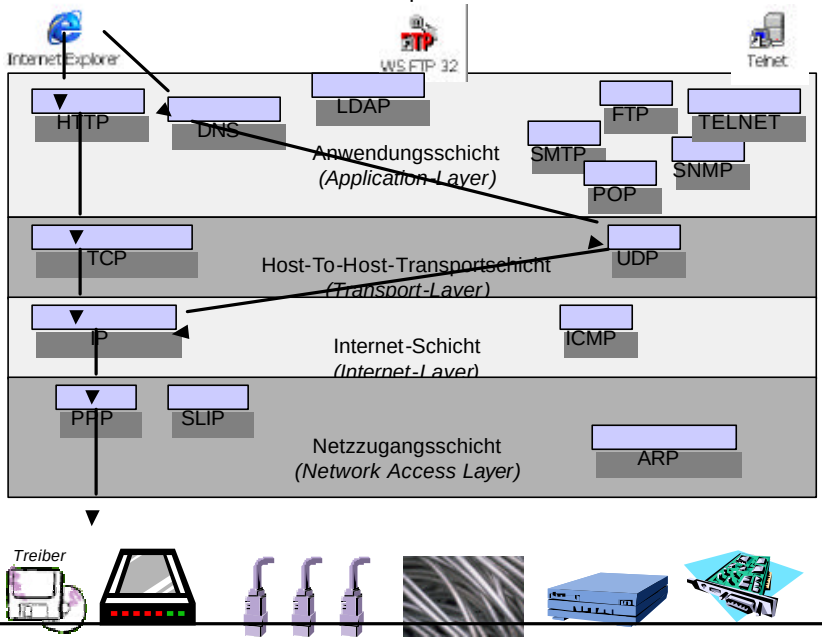
s.a. Hunt, C.: TCP/IP Netzwerkadministration; Bonn 1995, S. 9ff

1.2 Die TCP-IP-Protokollarchitektur (2)



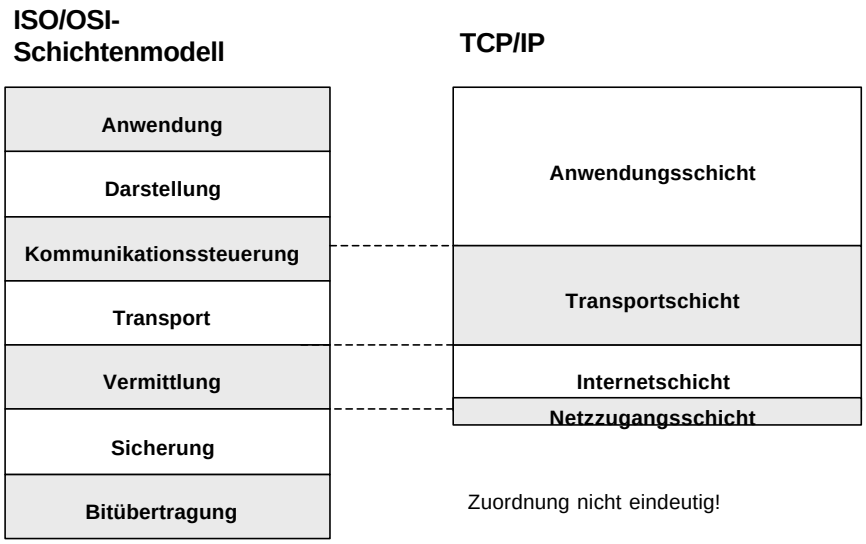
1.2 Die TCP-IP-Protokollarchitektur (3)

Beispiel



1.2 Die TCP-IP-Protokollarchitektur (4)

TCP/IP und das ISO/OSI-Schichtenmodell



1.2 Die TCP-IP-Protokollarchitektur (5)

einige wichtige Protokolle

Netzzugangsschicht

- SLIP: Serial Line Internet Protocol: IP über serielle Leitungen (v.a. Telefonverbindungen)
- PPP: Point To Point Procotol: wie SLIP, aber umfangreicher (u.a. auch andere Protokolle)
- ARP: Adress Resolution Protocol: Setzt Ethernet-Adressen in IP-Adressen um

Internet-Schicht

- IP: Internet Protocol: Basisprotokoll für verbindungslose Übertragung von IP-Paketen (Datagrammen)
- ICMP: Internet Control Message Protocol

Transportschicht

- TCP: Transmission Control Protocol für verbindungsorientierte Verbindungen zwischen Anwendungsprozessen
- UDP: Für verbindungslose Übertragung zwischen Rechnern

1.2 Die TCP-IP-Protokollarchitektur (6)
einige wichtige Protokolle (Forts.)

Anwendungsschicht

- DNS: Domain Name Service, Auflösung von Namen in IP-Adressen
- SNMP: Simple Network Management Protocol, zur Überwachung und Konfiguration von Netzwerken
- SMTP: Simple Mail Transfer Protocol, zur Übertragung von E-Mails
- POP: Post Office Protocol, zum Abruf von E-Mails
- IMAP: Internet Mail Procotol, Abruf von Mails mit erweiterter Funktionalität
- LDAP: Lightweight Directory Address Protocol; zum Zugriff auf Verzeichnisdienste
- HTTP: Hyper Text Transfer Protocol, Übertragung von Webseiten
- TELNET: Terminal Emulation
- FTP: File Transfer Protocol, Dateiübertragung

TCP-IP-Grundlagen
2. IP und ICMP - Protokolle der Internet-Schicht

2.1 Aufgaben des Internet-Protokolls
2.2 Adressierung mit IP
2.3 Aufbau eines IP-Datagramms
2.4 Routing im IP-Protokoll
2.5 Multiplexen nach Protokollen
2.6 Das Internet Control Message Protocol: ICMP
2.7 Grenzen von IPv4 – IPv6 als Lösungsansatz

2.1 Aufgaben des Internet-Protokolls (1)

Das **Internet-Protokoll = IP** ist **das** Basisprotokoll bei TCP/IP!

Inhalte des Internet-Protokolls = IP:

- Definition des **Datagramms** (Datenpaket des IP-Protokolls) und seiner Inhalte
- **Adressierung** im Internet
- **Routing** im Internet
- **Multiplexen und Demultiplexen** von Daten für bestimmte Protokolle
- Datenaustausch mit Netzzugangsschicht und Transportschicht

Derzeitige Version: IPv4 (vgl. RFC 791)

Eigenschaften des IP-Protokolls

- Sogenanntes **Verbindungsloses Protokoll**, d.h. kein Austausch von Kontrollinformationen („Handshake"), um Verbindung herzustellen.
- Sogenanntes **Unzuverlässiges Protokoll**, d.h. keine Fehlerkorrektur

2.1 Aufgaben des Internet-Protokolls (2)
Schnittstellen Netzzugangsschicht

Das IP-Protokoll kommuniziert mit dem ISO/OSI-Sicherungs-Schicht über spezielle Protokolle der Netzzugangsschicht, z.B.

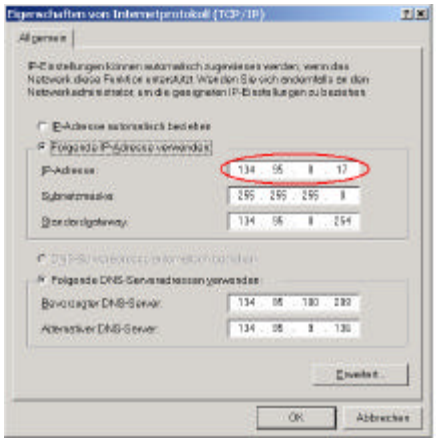
- RFC 826, **Address Resolution Protocol (ARP)**
  ✍ bildet IP-Adressen auf Ethernet-Adressen ab
- RFC 894 ✍ kapselt IP-Datagramme für den Transport über Ethernet-Netze
- RFC 1618 PPP over ISDN + RFC 1661: PPP ✍ Zugriff auf ISDN

- „Zugangspunkt" zur Netzzugangsschicht werden bei TCP/IP (Network-) **Interfaces** genannt

- Ein Knoten in einem TCP/IP-Netzwerk kann auf mehrere Interfaces zugreifen, z.B.
  - 1 Interface für den Zugriff auf eine ISDN-Karte mit PPP
  - 1 Interface für den Zugriff auf eine Ethernet-Karte
  - 1 Interface für den Zugriff auf ein Modem mit PPP oder SLIP
  - 1 Interface als Schnittstelle zu einem Protokoll auf Sicherungs- oder Vermittlungs-Schicht aus einer anderen Protokollfamilie

2.2 Adressierung mit IP (1)

- IP-Adresse besteht aus
  4 Byte = 4 x 8 = 32 Bits
- Zusammen mit der Subnetzmaske identifiziert IP-Adresse das **Netz**, des **Subnetz** und den **Knoten / Host**
- Die Adresse setzt sich zusammen aus
  1. Adresse des (Sub-) Netzes und
  2. Adresse des Netzknotens/Host
- Über die **Subnetzzmaske** kann abgelesen werden, welcher Teil der Adresse Netzteil, welcher Adressteil ist (Adressteil = IP-Adresse AND NOT Subnetzmaske;d.h. alle gesetzten Bits der Subnetzmaske gehören zur Netzadresse)
- Im Beispiel:
  134.95.8.17 AND NOT 255.255.255.0
  = 17
  Der Rechner hat also Adresse 17



2.2 Adressierung mit IP (2)

- Netzwerke können in unterschiedlich viele Hosts beinhalten:
  Je nach **Netztyp** ist der Netzwerkteil der Adresse unterschiedlich groß
- Es werden **3 grundsätzliche Netztypen** unterschieden:
  - **Klasse A-Netze** mit 8 Netzwerk-Bits (✍ 24 Adressbits),
    identifiziert durch eine **0** als erstes Bit
    d.h. das erste Byte der Adresse **< 128**
  - **Klasse B-Netze** mit 16 Netzwerk-Bits (✍ 16 Adressbits),
    identifiziert durch **1 0** als erste Bits
    d.h. das erste Byte der Adresse liegt **zwischen 128 und 191**
  - **Klasse C-Netze** mit 24 Netzwerk-Bits (✍ 8 Adressbits),
    identifiziert durch **1 1 0** als erste Bits
    d.h. das erste Byte der Adresse liegt **zwischen 192 und 232**
- Adressen mit erstem Byte **> 232**: reservierte Adressen, z.B. Multicast-Adressen
- Über die **Subnetzmaske** können die verfügbaren Adressen weiter in Subnetze unterteilt werden

Beispiel:

134.95.8.17 gehört zu einem **Klasse B-**Netzwerk ✍ 2 Byte für Adressen
Subnetzmaske 255.255.**255.0** ✍ 1 Byte Identifikation des **Subnetzes**
✍ Netz **134.95,** Subnetz **8,** Adresse **17**

2.2 Adressierung mit IP (3)

- **Adressteil 0: Netzwerk selbst**, z.B. 134.95.0.0 (Netzwerk 134.95)**,** bzw. 134.95.8.0 (Subnetz 8 des Netzwerkes 134.95)
- Adressteil **alle Bits auf 1**: **Broadcast an das Netz**
- 0.0.0.0 eigenes Netzwerk
- 127.0.0.1 eigener Rechner: ***„localhost"***
- Von der IANA reserviert für private Netzwerke:
  - **10.0.0.0** **privates Klasse-A-Netz**
  - **172.16.0.0** **privates Klasse-B-Netz**
  - **192.168.0.0** **privates Klasse-C-Netz**

  Adressen aus privaten Netzwerken haben keine Gültigkeit im Internet und werden im Internet i.d.R. nicht weitergeleitet.
  ✍ vgl. Rekhter, Y., Moskowitz, B., Karrenberg, D., de Groot, G.J., Lear, E.: RFC 1918: Address Allocation for Private Internets; Februar 1996

2.2 Adressierung mit IP (4)

Masquerading / Network Adress Translation:

Soll von einem privaten Netzwerk auf das öffentliche Internet zugegriffen werden, so müssen die Adressen „übersetzt" werden, d.h. es müssen privaten Adressen öffentliche Adressen zugeordnet werden
✍ **Network Adress Translation (NAT)**

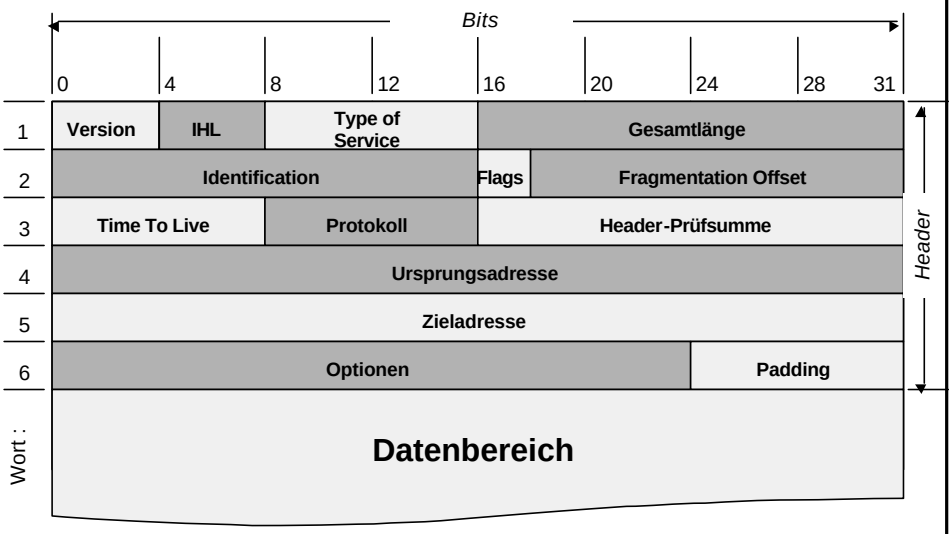
- wird jeder privaten Adresse eine öffentliche zugeordnet, spricht man von **n:m-NAT**
- Stehen nur 1 öffentliche Adressen mehreren privaten Adressen zur Verfügung, spricht man von einem **n:1-NAT**✍ Unterscheidung der Zielrechner im privaten Netz über **Ports**, s.u.
  (man spricht auch von **IP-masquerading**)

Dynamische Zuweisung von IP-Adressen

- **DHCP-Protokoll** („dynamic host configuration protocol"): Server können Informationen über IP-Nummern, Gateways und Nameserver dynamisch an Clients weitergeben
- Client fordert IP-Nummer i.d.R. beim Booten an
- IP Nummer wird Pool von IP-Adressen („**range**") entnommen
- Gültigkeit der Adressen für eine definierbare Dauer („**lease time**")

vgl. Kuri: Gruppenreise ins Internet: Gemeinsamer Internet-Zugang durch das LAN; In: c't 17/98, S. 118ff

2.3 Aufbau eines IP-Datagramms (1)



Quelle: C. Hunt: TCP/IP – Netwerkadministration, Bonn 1995, S. 14

2.3 Aufbau eines IP-Datagramms (2)

Datagramm besteht aus **Header** mit Steuerungsinformationen und **Daten**.

Der **Header** umfasst bis zu sechs 32-Bit-Wörter; die Informationen dort beinhalten u.a.

- Die **Länge** von Header (IHL) und des Paketes
- Das **Protokoll der Transportschicht**
- Die **Time To Live** - Zeit in Sekunden, bis zu der das Datagramm abgeliefert worden sein muss
- Eine **Prüfsumme**
- Die **Ursprungsadresse** (Wort 4)
- Die **Zieladresse** (Wort 5)
- **Identification**, **Fragmentation,** und **Fragmentation Offset** + **Flags** („More-Fragment-Bit“) kann ein IP-Datagramm zusammengebaut werden, das für die Verpackung in kleinere Pakete der Netzzugangsschicht zerlegt wurde

2.4 Routing im IP-Protokoll (1)

```
C:\>tracert www.nasa.gov

Routenverfolgung zu foundation.hq.nasa.gov [198.116.142.34]  über maximal 30
Abschnitte:

  1   <10 ms   <10 ms   <10 ms  pohlig-gw.rrz.Uni-Koeln.DE [134.95.8.254]
  2    40 ms   110 ms   110 ms  chemie-gw.rrz.Uni-Koeln.DE [134.95.4.1]
  3   <10 ms   <10 ms   <10 ms  B-WiN-gw.rrz.Uni-Koeln.DE [134.95.99.254]
  4   <10 ms   <10 ms   <10 ms  uni-koeln1.win-ip.dfn.de [188.1.6.5]
  5   <10 ms   <10 ms    10 ms  zr-koeln1.win-ip.dfn.de [188.1.160.13]
  6   <10 ms   <10 ms    10 ms  cr-koeln1.g-win.dfn.de [188.1.12.86]
  7    10 ms    10 ms    10 ms  cr-hannover1.g-win.dfn.de [188.1.18.9]
  8    80 ms    90 ms    90 ms  ir-nyc2.g-win.dfn.de [188.1.18.62]
  9    80 ms    90 ms    90 ms  dfn-IR-NYC2.ny4.ny.dante.net [212.1.200.45]
 10    80 ms    90 ms    91 ms  500.POS3-0.GW5.NYC9.ALTER.NET [157.130.254.241]
 11    80 ms    91 ms    90 ms  520.at-5-0-0.XR1.NYC9.ALTER.NET [152.63.24.18]
 12    80 ms    90 ms    91 ms  0.so-3-0-0.TR1.NYC9.ALTER.NET [152.63.22.98]
 13    90 ms   100 ms    90 ms  125.at-5-0-0.TR1.DCA6.ALTER.NET [152.63.2.125]
 14    90 ms   100 ms    90 ms  287.at-5-0-0.XR1.TCO1.ALTER.NET [152.63.34.21]
 15    90 ms    90 ms   100 ms  193.ATM8-0-0.BR2.TCO1.ALTER.NET [146.188.160.81]
 16  2664 ms  2633 ms  2604 ms  mae-east.nsn.nasa.gov [192.41.177.125]
 17  2604 ms  2644 ms  2684 ms  128.161.3.14
 18  2654 ms  2684 ms  2774 ms  128.161.1.62
 19  3004 ms  3014 ms  3055 ms  border.hcn.hq.nasa.gov [198.116.63.2]
border.hcn.hq.nasa.gov [198.116.63.2]  meldet: Zielnetz nicht erreichbar.

Ablaufverfolgung beendet.
```


2.4 Routing im IP-Protokoll (2)

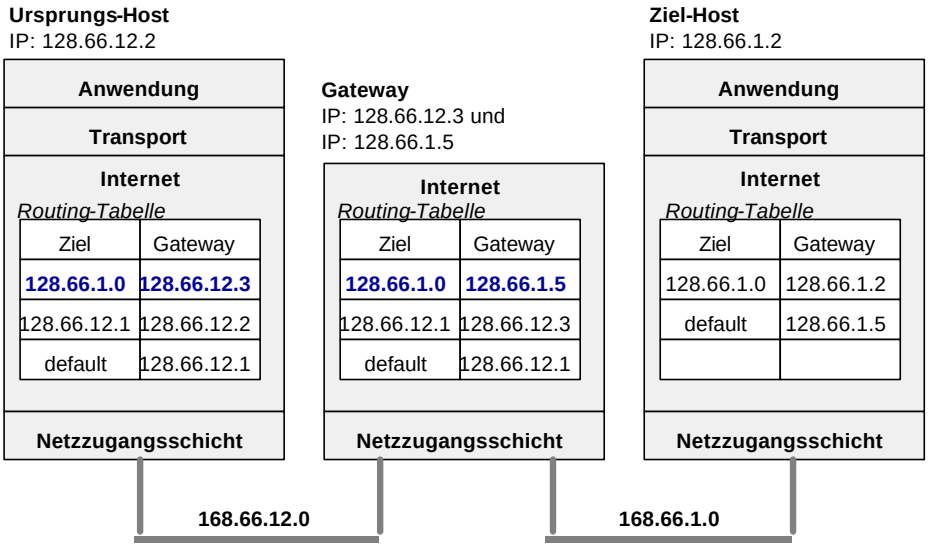
Es werden bei IP zum Routing folgende Arten von Geräten unterschieden:

- **Hosts** (bei TCP/IP: Host = Rechner)
- **Gateways** (leiten Datagramme zwischen verschiedenen Netzwerken weiter, im ISO/OSI-Sinn eigentlich Router)
- **Multi-Named-Hosts** (Hosts, die an zwei Netzwerke angeschlossen sind)

- Jeder Knoten, der das Datagramm weiterleitet, wird **Hop** genannt
- Das IP-Protokoll routet die Datagramme immer nur **mit Hilfe der Routingtabelle** weiter
- Ein Eintrag in der Routingtabelle ist **kein Pfad** zum Zielnetzwerken, sondern nur **zum nächsten Gateway (next hop)**
- Es ist **gleichgültig**, ob die Tabelle von einem Administrator oder einem Programm mit Hilfe eines entsprechenden Algorithmus erstellt wurde

2.4 Routing im IP-Protokoll (3)

Beispiel



Der UDP-Header (vgl. Hunt / TCP-IP 1995 / S. 41)

2.4 Routing im IP-Protokoll (4)

```
Aktive Routen:
Netzwerkziel     Netzwerkmaske    Gateway          Schnittstelle    Anzahl
127.0.0.0        255.0.0.0        127.0.0.1        127.0.0.1        1
134.95.8.0       255.255.255.0    134.95.8.17      134.95.8.17      1
134.95.8.17      255.255.255.255  127.0.0.1        127.0.0.1        1
134.95.255.255   255.255.255.255  134.95.8.17      134.95.8.17      1
224.0.0.0        224.0.0.0        134.95.8.17      134.95.8.17      1
255.255.255.255  255.255.255.255  134.95.8.17      134.95.8.17      1

Standardgateway: 134.95.8.254
```

Ausschnitt aus einer Routing-Tabelle (Win 2000)

- Routinginformationen werden innerhalb eines Netzwerkes häufig mit dem **Routing Information Protocol (RIP)** aufgebaut und ausgetauscht
  - ✍ **Internes Routing-Protokoll**
- Netzwerke und Netzwerkgruppen (**Autonomous Systems** bzw. **Routing Domains**) tauschen darüber hinaus **Erreichbarkeitsinformationen** über spezielle Protokolle (Border Gateway Protocol BGP oder Exterior Gateway Protocol EGP) aus ✍ **Externe Routing-Protokolle**
- Einsatz von externen Routingprotokollen z.B. an zentralen Knoten, an denen regionale Internet-Service-Provider zusammengeschaltet sind Z.B. **DE-CIX** in Frankfurt, **mae-east** und **mae-west** in den USA oder **Linx** in London
  (vgl.: Hintergrundinformationen zum DE_CIX unter www.eco.de)

2.5 Multiplexen nach Protokollen

- Die Datagramme sind nicht nur bestimmten Hosts, sondern auch **bestimmten Protokollen der Internet- und Transportschicht** zugeordnet, die gleichzeitig arbeiten können, z.B. TCP und UDP
- Die Datagramme verschiedener Protokolle werden gemeinsam über eine Verbindung übertragen ✍ **Multiplexing**
- Jedes Datagramm trägt eine **Protokoll-Nummer** (Wort 3)
- Über die Protokollnummern können die Datagramme wieder richtig auseinandersortiert weden ✍ **Demultiplexing**
- Sogenannte **well-known-protocols** werden in der Datei **etc/protocols** gespeichert

```
# Diese Datei enthält die Internetprotokolle gemäß
# RFC 1700 (Assigned Numbers).
# Bearbeiten Sie diese Datei mit einem ASCII-Editor.
#
# Format:
#
# <Protokollname>  <Nummer>    [Alias...]     [#<Kommentar>]

ip        0     IP        # Internet Protocol
icmp      1     ICMP      # Internet Control Message Protocol
ggp       3     GGP       # Gateway-Gateway Protocol
tcp       6     TCP       # Transmission Control Protocol
egp       8     EGP       # Exterior Gateway Protocol
udp       17    UDP       # User Datagram Protocol
hmp       20    HMP       # Host Monitoring Protocol
rdp       27    RDP       # "Reliable Datagram" Protocol
```

Ausschnitt aus etc/protocol (Win 2000)

2.6 Das Internet Control Message Protocol

ICMPv4: Internet Control Message Protocol (vgl. RFC 792 und RFC 795)
Benutzt Datagramm-Dienste von IP für Kontrollmeldungen

Funktionen von ICMP:

- **Flusskontrolle**
  Gateway oder Host kann eingehende Datagramme nicht bearbeiten
  ✍ *ICMP-Source-Quench-Meldung* (Überlauf) an Absender

- **Erkennen unerreichbarer Ziele**
  Gateway oder erkennt: Zielnetzwerk, -rechner oder -port nicht erreichbar
  ✍ schickt *Destination-Unreachable-Meldung* (Ziel nicht erreichbar) an Absender

- **Änderungen im Routing**
  Gateway teilt anderen Rechnern mit, andere (z.B. billigere) Route zu nutzen
  ✍ schickt *Route-Redirect-Meldung* an betroffene Gateways

- **Statusabfrage bei fremden Rechnern**
  ✍ Absender sendet *ICMP-Echo-Nachricht* an anderen Zielrechner, um dessen Ansprechbarkeit zu überprüfen. Zielrechner sendet Nachricht an Absender zurück
  von **PING** genutzt: Notation **ping [ip-nummer | hostname]**
  Befehl zur Überprüfung der Erreichbarkeit eines Rechners

2.7 Grenzen von IPv4 – IPv6 als Lösungsansatz (1)

Derzeitige Version v4

- **Größtes Problem:** Begrenzter Nummernvorrat durch 32-Bit-Adressen und Netzklassifizierung (v.a. Engpass bei Klasse B-Netzen)
- **Keine Hierarchie** der Netzwerk-Adressen möglich
- **Keine inhärenten Sicherheitsmechanismen**
- **Kein quality-of-service** für Multimedia-Anwendungen
- **Ungenügende Möglichkeiten für Multicasts**
- Weitgehend **starre Header-Struktur**

Abhilfe: **IPv6**

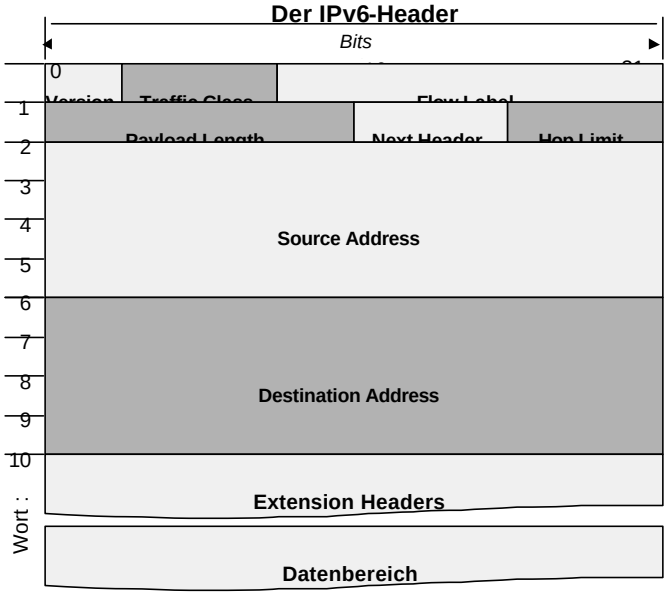
- 128-Bit-IP-Adressen, keine Netztypen
- neue Hilfsprotokolle: ICMPv6 (s. RFC 2463), DHCPv6 (noch „work in progress“) ARP wird zu NDP erweitert (Neighbor Discovery Protocol) (s. RFC 2461)
- Erweiterungen für das Domain Name System (s. RFC1886)

- **Neuer IP-Header**, neu insbes.:
  - **„Flow-Label“** für schnelle, virtuelle Ende-zu-Ende-Verbindung
  - **„Traffic Classes“** für Priorisierung und Klassifizierung von Daten
  - „**Extension Headers**“: optional weitere Steuerdaten

Quelle: Deering, S.; Hinden, R.: RFC 2460: Internet Protocol Version 6 (IPv6), December 1998

2.7 Grenzen von IPv4 – IPv6 als Lösungsansatz (2)

Der IPv6-Header



Quelle: Deering, S.; Hinden, R.: RFC 2460: Internet Protocol Version 6 (IPv6), December 1998

2.7 Grenzen von IPv4 – IPv6 als Lösungsansatz (3)

„**Extension Headers**“:
optional weitere Steuerdaten in Erweiterungs-Headern

Vorhandensein eines Extenstion Header: über „next-header“-Feld markiert

jeder Extension Header selbst hat „next-header“-Feld:
mehrere Extension-Header hintereinander möglich

Definierte Extension Header:

- **Hop-by-Hop-Options**-Header – Instruktionen für die Knoten auf dem Pfad
- **Destination Options-Header** – Instruktionen für den Ziel
- **Routing-Header** – definiert Gateways, die beim Routing angelaufen werden müssen
- **Fragment-Header** – stellt das richtige Wiederherstellen „zerlegter“ Pakte sicher ✍ Funktion der Fragmentation-Felder im IPv4-Header
- **Encapsulating Security Payload** – zur Verschlüsselung (s. RFC 2402)
- **Autentification Header** – Authentifizieren von Daten (s. RFC 2406)

Quelle: Deering, S.; Hinden, R.: RFC 2460: Internet Protocol Version 6 (IPv6), December 1998

2.7 Grenzen von IPv4 – IPv6 als Lösungsansatz (4)
IPv6-Adressen

- IPv6 nutzt **128-Bit-Adressen**, mit denen Interfaces (Schnittstellen zur Netzzugangsschicht - keine Knoten!) identifiziert werden; z.B. Ethernet-Karte eines Rechners, nicht der Rechner selbst
- Keine „Klassen“ mehr wie in IPv4
- Darstellung hexadezimaler Nummernfolgen in der Form **x:x:x:x:x:x:x:x**
  Gruppen von Nullen können weggelassen werden und durch „**::**“ ersetzt werden

  Beispiele:
  - **FEDC:BA98:7654:3210:FEDC:BA98:7654:3210**
  - **1080::::8:800:200C:417A**
  - **FF01::101**
  - **::1** (loopback)
  - **::** (unspecified address)
- IPv4-Adressen können in IPv6-Adressen eingebettet werden (in den letzten 4 Bytes)
  Format (d = Dezimalwert): **x:x:x:x:d:d:d:d**

  Beispiele:
  - **::FFFF:129.144.52.38** oder
  - **::13.1.68.3**

Quelle: Hinden, R.; Deering, S.: RFC2373 - IP Version 6 Addressing Architecture

2.7 Grenzen von IPv4 – IPv6 als Lösungsansatz (5)

IPv6-Adressen

- 3 Typen von IPv6-Adressen:
  - **Unicast** – 1:1-Verbindungen
  - **Anycast** – aus einer Gruppe von möglichen Empfängern in einer Region kann ein beliebiger gewählt werden (z.B. zur Identifikation eines Routers)
  - **Multicast** – 1:n-Verbindungen [es wurde 1/256 des Adressraums reserviert]
    - Unterscheidung der Reichweite: **node-local, site-local**, **organizational-local-scope**
    - Reservierte Multicast-Adressen für bestimmte Hosts (z.B. alle DHCP-Server)

  Keine Broadcast-Adressen; Funktionalität durch Multicast abgedeckt

- **Aggregatable Global Unicast Addresses**
  Adressen, die eine 3-Stufige hierarchische Zuordnung einer Adresse erlauben
  Aufbau:
  - Prefix 001
  - **Top-Level-Aggregation-ID** (13 Bit), z.B. für Superprovider, Backbone o.ä.
  - **Next-Level-Aggregation-ID** (24 Bit + 8 Bit „reserved for future use“)
    z.B. für den Provider; kann von Organisation weiter unterteilt werden
  - **Site-Level-Aggregation-ID** (16 Bit)
    z.B. für eine Organisation oder einen Standort
  - **Interface-ID** (64 Bit)

Quellen: Hinden, R.; Deering, S.: RFC 2373 - IP Version 6 Addressing Architecture
Hinden, R.; Deering, S.: RFC 2375 - IPv6 Multicast Address Assignments; Juli 1998

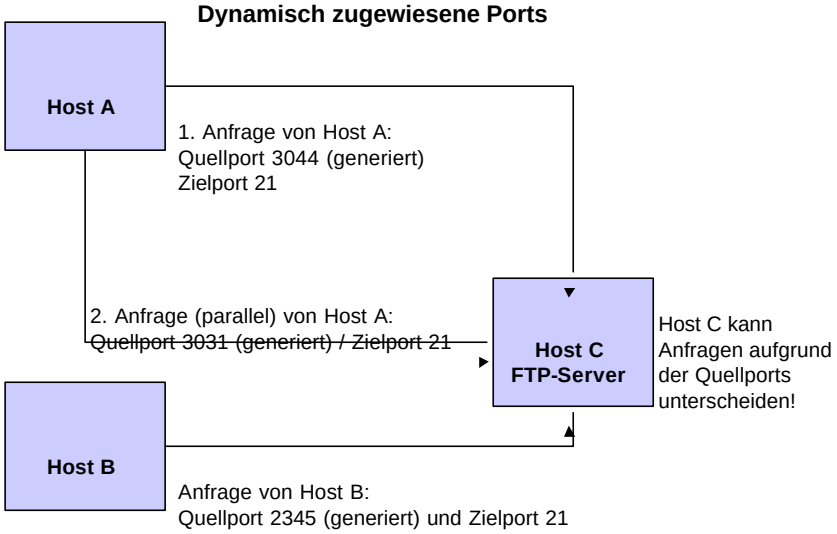
TCP-IP-Grundlagen
3. TCP und UDP - Protokolle der Transport-Schicht

3.1 Ports und Sockets
3.2 UDP
3.3 TCP

3.1 Ports und Sockets (1)

- Internet-Schicht erkennt Protokoll, an das Daten übergeben werden, über **Protokollnummer des Datagramms**
- Transportschicht erkennt das zuständige IP-Protokoll an der **protocols-Tabelle**
- Anwendungsprozesse werden in Transportschicht anhand einer **Portnummer** identifiziert (16 Bit)
  TCP- und UDP-Pakete haben Felder **für source** und **destination-port**
- Standard-Port-Nummern („well-known-ports“) in Datei **etc/services** abgelegt
- Portnummern nur innerhalb eines Protokolls eindeutig: UDP und TCP können gleiche Portnummern vergeben; nur Port- und Protokollnummern zusammen identifizieren den Anwendungsprozess
- Es gibt reservierte Ports (**well known-ports**), z.B.
  ftp 20
  smtp 25
  http 80
  und dynamisch zugewiesene Ports (**dynamically allocated ports**)
- Bei TCP werden dynamisch zugewiesene Ports beim sogenannten Handshake generiert und ausgetauscht
- Kombination aus IP-Nummer und Port wird **socket** genannt
- jeder socket wird über **IP-Nummer, Protokoll und Port** eindeutig identifiziert

3.1 Ports und Sockets (2)
Dynamisch zugewiesene Ports



Dynamisch zugewiesene Ports erlauben Mehrbenutzerdienste

3.1 Ports und Sockets (3)

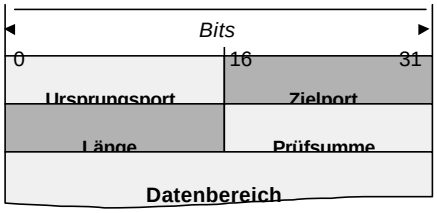
```
# Diese Datei enthält die Portnummern für bekannte Dienste gemäß IANA.
#
# Format:
# <Dienstname>  <Portnummer>/<Protokoll>  [Alias...]   [#<Kommentar>]

echo          7/tcp
echo          7/udp
daytime       13/tcp
daytime       13/udp
qotd          17/tcp      quote             #Quote of the day
qotd          17/udp      quote             #Quote of the day
ftp           21/tcp      #FTP. control
telnet        23/tcp
smtp          25/tcp      mail              #Simple Mail Transfer Protocol
domain        53/tcp      #Domain Name Server
domain        53/udp      #Domain Name Server
finger        79/tcp
http          80/tcp      www www-http      #World Wide Web
hostname      101/tcp     hostnames         #NIC Host Name Server
Pop3          110/tcp                       #Post Office Protocol - Version 3
nntp          119/tcp     usenet            #Network News Transfer Protocol
imap          143/tcp     imap4             #Internet Message Access Protocol
microsoft-ds  445/tcp
microsoft-ds  445/udp
ms-sql-s      1433/tcp    #Microsoft-SQL-Server
ms-sql-s      1433/udp    #Microsoft-SQL-Server
```

Ausschnitt aus etc/services (Win 2000)

3.2 UDP

UDP- User Datagramm Protocol (RFC 768)

- Datagrammdienst für Anwendungsdienste und -programme
- **unzuverlässig** – d.h. ohne Fehlererkennung und –korrektur
- **verbindungslos** – kein Handshake zwischen den Rechnern
- Header: Übergabe der Daten an den richtigen Anwendungsprozess über Destination-Port und Source-Port
- geringer Verwaltungsoverhead ✍ effiziente Kommunikation



Der UDP-Header
(Hunt / TCP-IP 1995 / S. 19)

Wann UDP?

- Für Anwendungen, die geringe Datenmengen versenden (evtl. erneute Übertragung effizienter)
- Für Anwendungen, die mit einem einfachen Frage-Antwort-Schema arbeiten oder eigene Fehlerkorrekturmaßnahmen besitzen
- Für Anwendungen, bei denen Fehlerkorrekturmaßnahmen nicht zwingend erforderlich sind

3.3 TCP (1)

TCP – Transmission Control Protocol (RFC 793)**:**
Ein zuverlässiges, verbindungsorientiertes, Byte-Strom-Protokoll

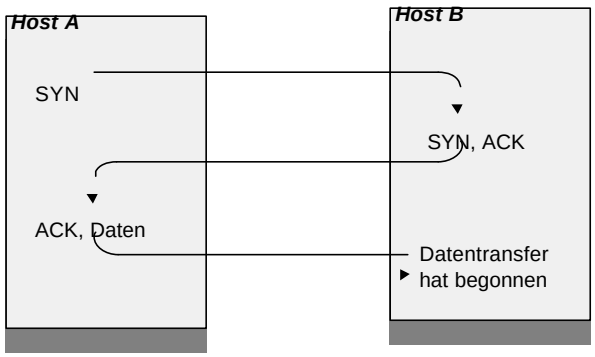
zuverlässiges Protokoll,
d.h. mit Fehlererkennung und -korrektur durch „Postive Acknowledgement with Re-Transmission“ (PAR, positive Bestätigung mit Neuübertragung):

- Dateneinheit bei TCP: **Segment**
- Jedes Segment: Prüfsumme ✍ Empfänger prüft Korrektheit und bestätigt positiven Empfang
- Rechner sendet Segmente
- noch einmal, wenn er keine Bestätigung über korrekten Empfang erhält

verbindungsorientiertes Protokoll,
d.h. es wird eine logische Rechner-zu-Rechner-Verbindung aufgebaut.

- Vor Nutzdatenversand werden Kontrollinformationen zwischen Sender und Empfänger ausgetauscht ✍ **handshake**
- Kontrollsegment durch Setzen eines Bits im Flags-Feld des Segment-Headers gekennzeichnet
- **3-Wege-Handshake:** es werden 3 Segmente ausgetauscht; danach haben Empfänger und Sender Gewissheit, dass der jeweilige Partner empfangsbereit ist
- Nach Abschluss der Übertragung: weiteren 3-Wege-Handshake um Verbindung zu schließen (Flag „FIN“)

3.3 TCP (2)



3-Wege-Handshake bei TCP (Hunt / TCP-IP 1995 / S. 21)

SYN: Synchronize sequence numbers

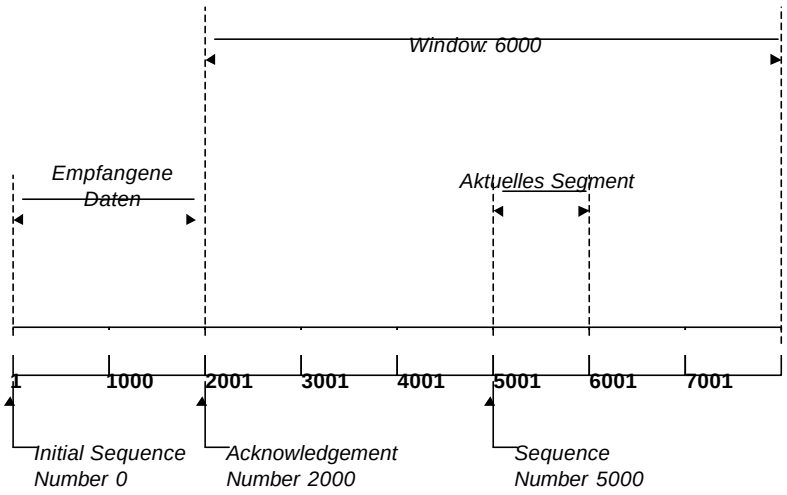
ACK: Acknowlegement

3.3 TCP (3)

Byte-Strom-Protokoll

- TCP sieht Daten als Datenstrom und nicht als Einzelpakete
- Bytes werden mit Hilfe der Felder **sequence-numbers** und **acknowledgement** in richtige Reihenfolge gebracht
- Während Handshake: Aushandeln des Startwertes: **initial sequence number** (ISN) im SYN-Segment (i.d.R. 0)
- Bytes werden durchnummeriert
- **Sequence-number** im Header eines TCP-Segments: Nummer des 1. Bytes im Datenbereich
- Positive Bestätigung erhaltener Daten und Flusskontrolle über **Acknowledgement-Segmente** (ACK)
  - Feld **Window** im TCP-Header des ACK-Segmentes: Wie viele Daten kann Adressat noch empfangen
  - Absender: Anzahl Bytes <= Window ✍ Empfänger steuert mit ACK-Segmenten Datenfluß

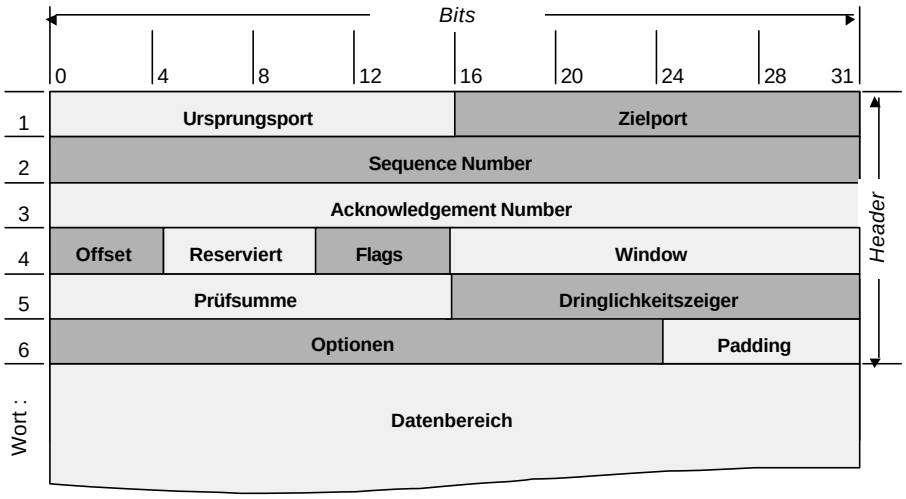
3.3 TCP (4)



Aufbau eines TCP-Segments, Quelle: C. Hunt: TCP/IP 1995 / S. 23

3.3 TCP (5)

Aufbau eines TCP-Segments



Aufbau eines TCP-Segments, Quelle: C. Hunt: TCP/IP 1995 / S. 20A

TCP-IP-Grundlagen
4. Namensauflösung mit TCP/IP

4.1 Namensgebung im Internet
4.2 Der Domain Name Service (DNS)

4.1 Namensgebung im Internet (1)

- **Name Service** ermöglicht das Verwenden von **Namen anstatt von IP-Adressen**
- Hierarchie auf Basis vorhandener Organisationsstrukturen:
  Namen der Form **Name.Second-Level-Domain.Top-Level-Domain**

- **Generic Top-Level-Domains (GTLD), bislang noch**:
  - (US-Amerikanische) Bildungseinrichtungen; primär Universitäten: .EDU
  - (US-Amerikanisches) Militär: .MIL
  - Kommerzielle Organisationen („commercial entities"): .COM
  - Internationale Organisationen oder Datenbanken: .INT
  - Netzwerk-Provider (Rechner des Network Information Services NIC, Administrationsrechner u.ä.): .NET
  - Diverse Organisationen: .ORG

- **Contry-Code TLDs:** (ccTLDs) ✍ nach ISO-Norm 3166, z.B. .DE, .CN (China), .RU (Russland). .NU (Niue) .TM (Turkmenistan), TV (Tuvalu) etc.
  Vollständige Liste der CTLDs: IANA: „Root-Zone Whois Information"
  http://www.iana.org/cctld/cctld-whois.htm
- Die Ebene der Second-Level-Domains ist länderspezifisch geregelt, z.B. Organisationstyp-Struktur (z.B. .ac, .co, .go, .re) oder Regionen-basierte Struktur (z.B. NY.US)

Vgl. Postel, J.: RFC 1591 - Domain Name System Structure and Delegation, 1994

4.1 Namensgebung im Internet (2)

Neue gTLDs (ab 2001)

- **.aero** – Registrar: Société Internationale de Télécommunications Aéronautiques (SITA)
  speziell für Fluglinien, Flughäfen; voraussichtlicher Preis: 50 US-Dollar
  Beispiele: www.berlin.de.air, www.lax.air
- **.biz** – Registrar: JVTeam
  für große und kleine Unternehmen
  Beispiel: www.heise.biz.
- **.info** - Registrar: Afilias (Konsortium aus 19 Registraren aus aller Welt, einschließlich mehrerer deutscher Unternehmen und CORE, dem in Genf ansässigen Council of Registrars);
  Beispiel: www.heise.info
- **.name** - Registrar: Global Name Registry, Tochter der britischen Nameplant.com
  für private Adressen, im wesentlichen nicht-kommerzielle Aktivitäten
  Beispiel: www.monika.ermert.name

Quelle: Emert, M.: Ein Königreich für einen (Internet-) Namen - Elf Direktoren wählen sieben neue Internet-Namensbereiche aus; In: c't 25/2000, S.66ff:

4.1 Namensgebung im Internet (3)

Neue GTLDs (ab 2001; Fortsetzung)

- **.pro** - Register.com, USA, Virtual Internet, UK, und Baltimore IT
  speziell für Berufsgruppen wie Anwälte, Ärzte
  Beispiel: www.PatrickGMayer.law.pro
- **.museum** - Museum Domain Management Association
  speziell für Museen in aller Welt
  Beispiel: www.guggenheim.us.museum.
- **.coop** - Cooperative League of the USA
  speziell für Genossenschaften

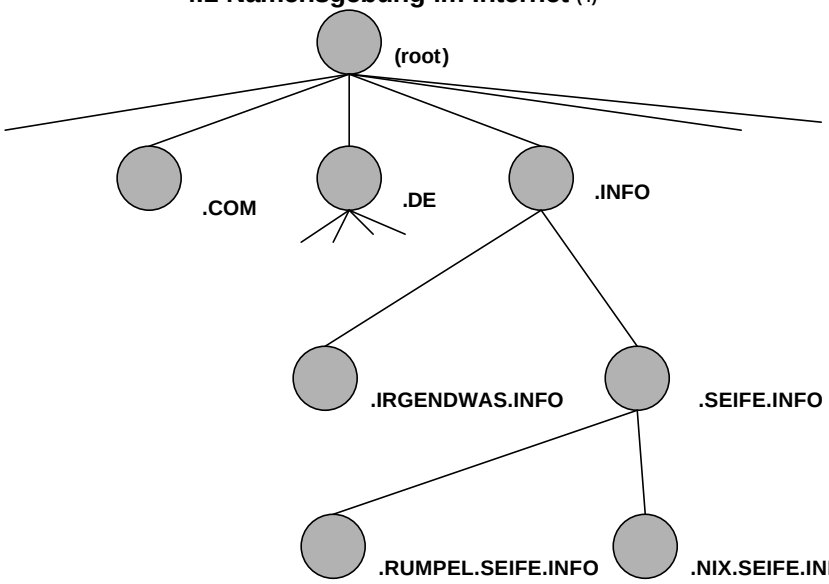
Quelle: Emert, M.: Ein Königreich für einen (Internet-) Namen - Elf Direktoren wählen sieben neue Internet-Namensbereiche aus; In: c't 25/2000, S.66ff

Organisation der TLDs

- „day-to-day-administration" der Domains durch **Internet Registries**
  - Zentrale IR ist **InternNIC**
  - **APNIC** - IR der Asiatisch-Pazifischen Region
  - **RIPE NCC** - IR für Europa
  - **DENIC** – IR für die DE-Domain
- Internet Registries verwalten sogenannte **whois-Datenbanken**, über die Ansprechpartner zu jeder Domain bestimmt werden können

vgl. Postel, J.: RFC 1591 - Domain Name System Structure and Delegation, 1994;
sowie www.denic.de, www.ripe.org, www.apnic.org, www.internic.net

4.1 Namensgebung im Internet (4)



4.2 Der Domain Name Service (DNS) (1)

- Namensauflösung durch den Rechner alternativ
  - lokal über Datei **hosts**
  - über den **Domain Name Service (DNS)**
- Verwaltung von Namen im Internet
  - früher: eine über das Internet verteilte Hosts-Datei mit allen Rechnernamen, die per FTP abgerufen wurde
  - Ende der 80er Jahre: **Domain Name Service**, erzwungen und ermöglicht durch
    - Größe der host-Datei
    - Neue Rechnerarchitekturen (Client"-Server") und
    - Neue Organisationsstrukturen im Internet
- DNS: Dienst auf Basis einer **verteilten, hierarchisch aufgebaute Datenbank**
- Daten in sogenannten **Master Files** abgelegt, die von **Name-Servern** verwaltet, ausgewertet und ausgetauscht werden
- **Service flexibel erweiterbar** und für **möglichst viele Anwendungen** offen

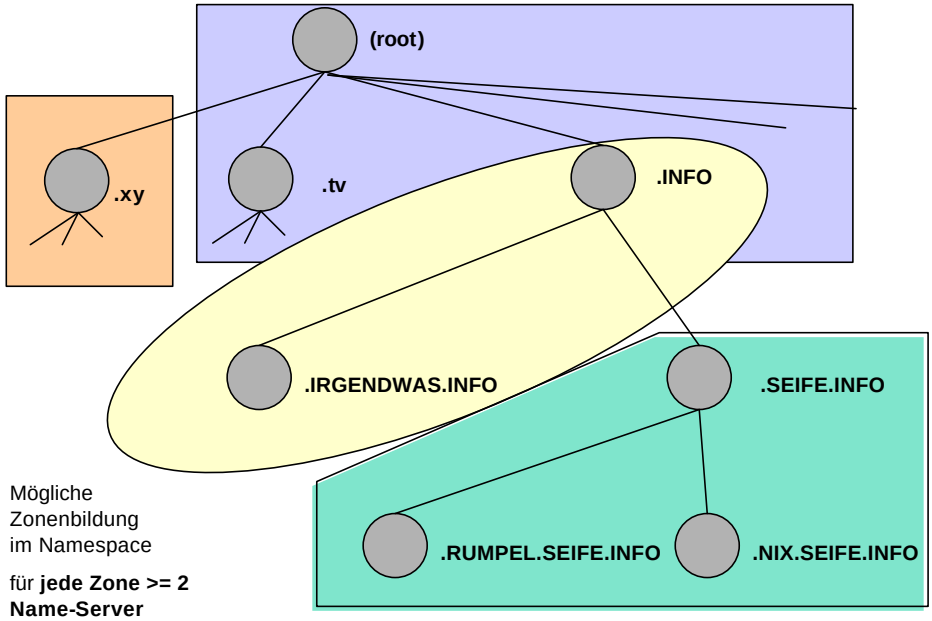
vgl. auch Mockapetris, P.: RFC 1034 - Domain Names: Concepts and Facilities ; November 1987

4.2 Der Domain Name Service (DNS) (2)

- DNS stellt bereit:
  - **Standardformat** für Master Files, in denen Informationen zu den Namen abgelegt werden
  - **Standardisierte Zugriffsverfahren** auf die verteilte Datenbank
  - Standardverfahren um die **lokale Datenbank mit Fremddaten zu aktualisieren**
- Bestandteile des DNS:
  - **Domain Name Space** (Baumstruktur für Namen) und **Resource Records** (Daten, die mit Namen verbunden werden); jeder Knoten hat Daten, nicht nur Blätter!
  - **NameServer**, die Informationen über einen zusammenhängenden Teilbaum - eine **Zone -** verwalten) Zone durch „Zerschneiden" des Domain-Name-Space-Baums in Teilbäume)
  - **Resolver**, Systemdienste die die Namensauflösung in standardisierter Form für beliebige Anwendungen bereitstellen (z.B. über eine Funktion **gethostbyname()**)
- Redundante Datenhaltung: Mindestens 2 Nameserver für eine Zone

vgl. auch Mockapetris, P.: RFC 1034 - Domain Names: Concepts and Facilities ; November 1987

4.1 Namensgebung im Internet (3)



Mögliche Zonenbildung im Namespace

für **jede Zone >= 2 Name-Server**

4.2 Der Domain Name Service (DNS) (4)

- an der Spitze steht der namenlose Root-Knoten
- Jeder andere Knoten im Namensraum hat einen Namen

- derzeit ASCII-basiert ohne Groß- und Kleinschreibung und ohne Sonderzeichen; Umstellung für internationale Zeichensätze wird derzeit verfolgt

- Jeder Knoten speichert in Resource-Records neben Nummern auch Informationen **über Services**
  dafür wurden verschiedene **Resource-Record-Typen** definiert, v.a.:
  - A host address
  - C Aliasname für einen Eintrag
  - MX „Mail Exchanger“ - nimmt Mails entgegen
  - HINFO CPU und OS des Hosts
  - NS authorative name server für die Domain
  - PTR Verweis auf anderen Ort im Domain-Name-Space

- außerdem hinterlegt: TTL (Zeit, die Name Gültigkeit hat und nach der ein Cache aufgefrischt werden muss)

vgl. auch Mockapetris, P.: RFC 1034 - Domain Names: Concepts and Facilities ; November 1987

4.2 Der Domain Name Service (DNS) (4)

- Wie läuft eine Abfrage eines Namens ab?
  1. Anfrage in einem Programm nach einer Internet-Adresse in Form einer Zeichenkette, z.B. wi-im.uni-koeln.de
  2. Programm gibt Zeichenkette an **System-Funktion** weiter, die für Namensauflösung zuständig ist (gethostbyname())
  3. Routine (Resolver) konstruiert ein **Anfragepaket für UDP-Port 53** für den vorgegebenen Namensserver (DNS-Server)
     dabei **Angabe der Art der abzufragenden Resource-Record-Typen** (z.B. nur Nummer, alle Resource-Records zu Namen, Resource-Records mit Mail-relevanten Informationen
  4. Nameserver gibt Antwort, falls er sie weiß, ansonsten „rekursive" Weitergabe der Anfrage an weitere Rechner

- Kommunikation **über UDP** (Effizienz!); in DNS: Maßnahmen zur Wiederholung von Anfragen bei Übertragungsfehlern
- Zuordnung von Anfragen über sogenannte **Query-IDs** (16-Bit-Zahlen, vom Absender gewählt)
- Vorsicht DNS-Spoofing: Missbrauch des DNS-Systems für Angriffe
  (z.B. falsche Informationen für DNS-Caches, Erraten von Query-Ids, Ausnutzen, dass einzelne Programme nicht die IP-Namens-Zuordnung überprüfen)

vgl.: Mraz, V., Weidner, K.: Falsch verbunden: Gefahr durch DNS-Spoofing; In: c't 10/97, S. 286ff und Mockapetris, P.: RFC 1034 - Domain Names: Concepts and Facilities ; November 1987