

Internet Konzepte

I. Motivation

„Wenn du Dinge kompliziert findest, bedeutet das, du hast sie nicht verstanden!
Einfachheit ist die Grundlage von allem!“

Was ist das Internet?

- Ein weltweites Computer Netzwerk mit grenzenloser Informationsfülle; eine „Kommunikations-Infrastruktur“
- Ein „Netzwerk von Computer-Netzwerken“
- Ein virtuelles Netzwerk
- Eine Anhäufung von elektronischen Diensten die eine gemeinsame Infrastruktur nutzen

Entwicklung / Geschichtliches

PHASE 1: Experimentierphase: Arpanet 1969 - 4 Unis in USA vernetzt

1971 – eMail Dienst

1973 – Interneting Project (DARPA)

Ethernet wird erfunden

1974/75 – Kahn/Cerf

Beginn der TCP/IP Entwicklung

1980 – Aufteilung des ARPANET

Milnet – ARPANET

01.01.1983 – TCP/IP wird Internet Grundlage

PHASE 2: Skalierung: wichtigster Motor der Entwicklung
UNIX mit TCP/IP Fähigkeiten

1986 – NFS NET – DNS wird eingeführt
vorher alle Pcs im /etc/hosts

PHASE 3: Universelle Globale
Anwendung

1991 – Tim Berners Lee – CERN – www

April 1993 – Software freigabe CERN Server

Dez. 1993 – HTW – 1. Web-Server

PHASE 4: Allgegenwärtiges
Internet

- schneller
- neue Dienste (Chat, IP Telefone, ...)
- Online Banking
- ...

ISO/OSI in 10 Minuten

- Schicht 1: Bitübertragungsschicht, physikalische Schicht
Stecker, Kabel, Bauformen, RS232, ...
- Schicht 2: Sicherungsschicht, Data Link Layer
Zugriffsverfahren (Ethernet – CSMA/CD)
Datenpakete – Frames
PPP/SLIP
- Schicht 3: Vermittlungsschicht
Network Layer – Entstehung virtuelles Netzwerk – Internet
IP – Internet Protokoll, ICMP/ARP, DHCP
Es gibt hier KEIN gemeinsames Übertragungsmedium, Routing!
(Host-to-Host)
- Schicht 4: Transport-Schicht / Transport Layer
Programm zu Programm / Point to Point
- Datenstrom (~Dateiverarbeitung) – TCP
 - Nachrichten – UDP
- Entkoppelt die höheren Schichten von Kommunikationsdetails
- Schicht 5: Sitzungs-Schicht / Session Layer
Kommunikations-Steuerungs-Schicht
telnet, rlogin, ssh, web login
- Schicht 6: Darstellungs-Schicht / Presentation-Layer
Big-Endian, Little-Endian
ASCII <-> EBCDIC
SSL – Secure Socket Layer (https, pops)
- Schicht 7: Anwendungsschicht / Application Layer
Die Anwendungsprogramme: Web-Browser
Server Software
- Realisieren von Anwendungsprogrammen-Protokollen
www – http
eMail – smtp / pop
ftp – ftp
...

RFCs <http://www.rfc.editor.org/>

Alle Standards des Internets werden/wurden als RFC geboren. RFC beschreiben Protokolle.

RFC1700 – Assigned Number

RFC3232 – DB basierte <http://www.iana.org/numbers.html>

RFC3300 – Internet Official Protocol Standards

Zustände:

- Standard – bereits ein Standardprotokoll
- Draft standard – Standard in Entwicklung
- Prepared – Vorgeschlagener Standard
- Experimental – in Experimentier Phase
- Informational – Nur zur Information
- Historic – Ein alter Standard

Anforderungsstufen:

- Required - benötigt
- Recommended - empfohlen
- Electiv
- Limited use
- Not recommended – nicht empfohlen

Host requirements: RFC 1122/1123

Protokoll	Status	A-Stufe
Ipv4	standard	required
ICMPv4	standard	required
Telnet	standard	recommended
ARP	standard	elective

II. Grundlagen der WAN/LAN Technologie

Das Paket-Konzept

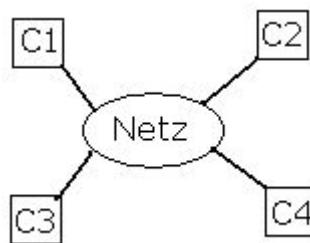
C-Netzwerke sind heutzutage meist Paketvermittelnde (packet switching) Netzwerke.

Von Paketen spricht man, wenn kleine Dateneinheiten verschickt werden.

Warum Pakete?

- Weniger Fehlermöglichkeiten
- Kein Dauersender
- Schnellere Fehlerkorrektur
- Schnellere Koordination von Sender/Empfänger

Beispiel 1: Übertragung großer Dateien



C1 sendet 500 Mbytes an C4 bei 5Mbit/s Netzdurchsatz $\rightarrow 800s = 13 \text{ Min}$

Wenn C2 kurz nach C1 1 kByte an C4 senden will, dann muss C2 ~13Min warten und kann dann 1,6 ms senden.

Lösung: Paketsystem, alle Rechner dürfen nur 1kByte-Pakete versenden.

C1 sendet 1. Paket
C2 sendet sein Paket
C1 senden den Rest

Zeit:

C1 $\rightarrow 13\text{Min} + 1,6\text{ms}$

C2 $\rightarrow 1,6\text{ms} + 1,6\text{ms}$

Das Paket-Konzept sorgt für PROMPTEN Zugang zum Netz.

Aber:

- Die Natur ist nicht perfekt, es gibt Übertragungsfehler
- Das durchschnittliche Fehleraufkommen im Beispielnetz ist 1 : 1 Million übertragenen Bits
- Sollen 10 Millionen Bit auf einmal übertragen werden, so entstehen im Durchschnitt 10 Fehler. **Wie Oft muss man übertragen?**
Übertragungszeit nicht abschätzbar!

Paket-Größe	Durchschnittliche Fehler
1 Mio Bit	1
100 000 Bit	0,1
1000 Bit	0,001

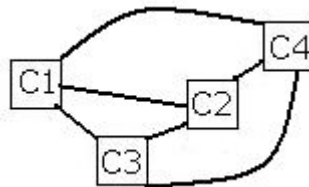
Reale Fehlerraten:

Kupfer	→ 1 : 10^6 – 10^7 Bit
Glas	→ 1 : 10^{12} – 10^{14} Bit

Netzwerk-Topologien

Geschichte:

→ Punkt zu Punkt Netzwerk



Vorteil:

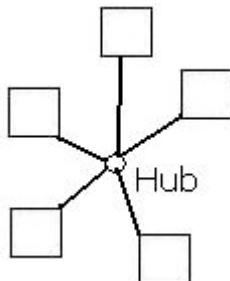
- Exklusive Verbindungen zum Kommunikationspartner
- Falls eine Verbindung ausfällt dann nur diese
- Parallele Kommunikation

Nachteil:

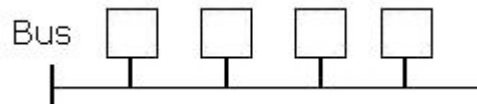
- Vernetzungsaufwand

LAN

Stern:



Bus:



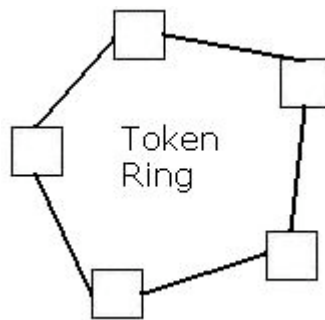
Vorteil: Bei ausfall eines
→ kein Problem

Preis

Nachteil: Preis

Bei Ausfall → Problem

Ring:



Nachteil → bei Ausfall → Problem

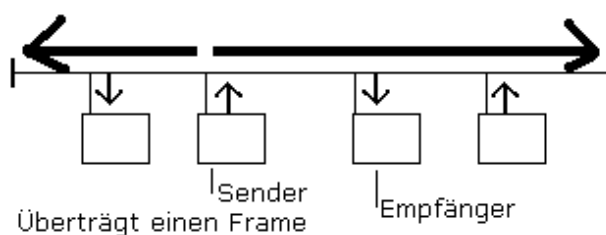
Busnetz: Ethernet

entwickelt, Anfang 1970er, von der Xerox Cooperation, weiterentwickelt von Digital Equipment, Intel, Xerox → DIX-Ethernet, Ethernet 8023

Ethernet → Äther → Übertragungsmedium

Namen:

10	Base	5
10	Base	2
100	Base	T
1000	Base	T
Mbit/s	Basisband	twisted Pair Kabel Bzw. bei der Zahl 5 = 500m Maximallänge



Das Signal belegt das Netz komplett, alle angeschlossenen Netzwerk-Geräte empfangen das Signal

Wie wird die Datenübertragung koordiniert?

- Es gibt **keine** Zentrale Koordination
- CSMA – Carrier Sense with multiple access (Trägerabtastung)
- Das elektrische Signal ist der Träger der Information, ist auf dem Netz kein Signal, dann kann ein Computer seine Datenübertragung beginnen.
- Ist das Netz belegt, d.h. ein Signal auf dem Netzwerk, so darf ein Computer nicht mit einer neuen Sendung beginnen.

Problem: 2 Computer beginnen gleichzeitig zu senden

- Gleichzeitiges Senden ist möglich, da die Ausbreitungsgeschwindigkeit des elektrischen Signals $\sim 2/3 c$ ist.
- Gleichzeitiges Senden führt zu Überlagerung der Sendesignale → Kollisionen genannt
- CD – Collision detect
- Das Ethernet Medium Zugriffsschema: CSMA/CD

Collision detect

Jede Netzwerk-Karte hat einen Sende- und Empfangskanal.

Die sendende Karte vergleicht das Sendesignal mit dem Empfangssignal, sind beide gleich → alleiniger Sender → weitersenden erlaubt

Sind die Signale unterschiedlich, so wird das Datensenden abgebrochen, es wird ein Störsignal gesendet, danach eine zufällige Zeitspanne gewartet. Ist das Netzwerk „leer“, d.h. kein Signal vorhanden, so wird wieder gesendet.
Zum zuverlässigen Erkennen einer Kollision, muss das kleinste Datenpaket das komplette Netz belegen

10 Mbit/s	→ 64 Bytes = 512 Bit	→ 51,2 μ s
100 Mbit/s	→ 64 Bytes	→ 5,12 μ s
1 Gbit/s	→ 64 Bytes = 4096 Bit	→ 4,01 μ s

Das LAN-Adressierungs Schema

LAN ↔ Netzwerk-Karte

- Gesendete Daten erreichen alle am Netzwerk angeschlossenen Geräte, der Empfänger muss irgendwie identifiziert oder bestimmt werden → Adresse Hardwareadresse, MAC Adresse, Ethernet Adresse, Physikalische Adresse
- Eine Netzwerk-Karte hat **eine** Adresse.
- Eine Netzwerk-Karte empfängt Datenpakete, Frames genannt.
- Anhand ihrer Adresse kann die Netzwerk-Karte, alle Frames erkennen, die an sie adressiert wurden.
- Nur die Frames wird sie weiterverarbeiten.

Netzwerk-Karte ↔ CPU

- Die Netzwerk-Karte sendet empfangene Daten an die CPU

Merke:

Eine Netzwerk-Karte hört permanent den Netzwerk-Datenverkehr ab. Sie filtert alle für sie bestimmten Datenpakete [Frames] aus dem Datenstrom des Netzwerkes und gibt die empfangenen Daten an die CPU weiter.

Wie werden Hardware-Adressen zugewiesen?

Statisch: Hardware-Hersteller vergeben die Adressen
Jeder Hardware-Hersteller erzeugt weltweit eindeutige Adressen
[<http://www.ethermanage.com>]

Vorteil:

- Einfache Installation
- Hersteller-Mix ist möglich
- Bei Neustart, kennt das Gerät seine Adresse.
- Statische IP/DHCP

Konfigurierbar: gedacht als vereinfachte Adressumsetzung. (MAC \leftrightarrow IP)
Der Geräte-Eigner definiert die Hardware-Adresse.

Vorteil:

- Die Adressumsetzung berechenbar

Nachteil:

- MAC-Spoofing

Dynamisch: Beim starten wird dem Gerät eine Hardware-Adresse übergeben

Vorteil:

- Kleine, kurze Adressen verwendbar

Nachteil:

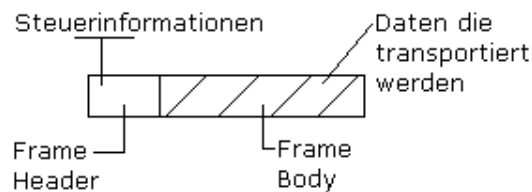
- Rechner nicht mehr identifizierbar

Broadcast Adressen

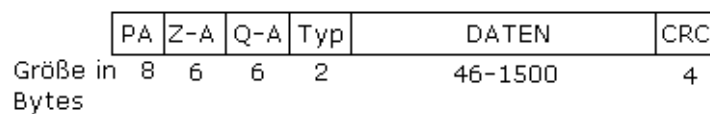
Broadcasting = Rundsenden \rightarrow ff:ff:ff:ff:ff:ff

Netzwerk-Karten verarbeiten alle Datenpakete, die als Empfänger-Adresse ihre MAC-Adresse oder die Broadcast-Adresse enthalten.

Wie ist ein Ethernet-Datenpaket (Frame) aufgebaut?



Jede LAN Technologie definiert ihr eigenes Rahmenformat, Rahmengröße 64-1518 Bytes



Typ - Rahmentyp - beschreibt die Art der transportierten Daten z.B.:

IP Pakete: 0x0800

ARP Pakete 0x0806

Q-A - Quell Sender Adresse

Z-A - Ziel/Empfänger Adresse

Präambel: 64Bit -> 0101...11 wird von dem Empfangs-Netzwerk zur Synchronisation genutzt

Ethernet II → Rahmentyp (> 1500) – 2 Byte – Typ der Daten (10 / 100 MBit)

Ethernet IEEE 802.3 Größe des Datenbereichs ≤ 1500 (10 / 100 MBit)

MTU → Maximum Transport Unit

Mtu-Ethernet II → 1500 Bytes

Ethernet 802.3 → 1492 Bytes

Path-MTU → was ist der kleinste Hardware Rahmen unterwegs

Links:

<http://www.ethermanage.com>

<http://wikipedia.org/Ethernet>

Kleine Hardwarekunde

10 Base 5 / Yellow / Thick Ethernet

10 Base 2 / cheaper / Thin Ethernet (BNC)

10 Base T

100 Base T

Andere Link-Layer (Sicherungsschicht) Protokolle

Zweck: verschicken von Daten der Schicht 3
Bei TCP/IP Daten der Art: IP, ARP, RARP | IPX (Novell)

Beispiel: LAN → Ethernet, Token Ring, FDDI

WAN → SLIP, PPP; PPPoE

SLIP – Serial Line IP RFC 1055

Einfachste Form der Verpackung von IP Paketen, die über eine serielle Leitung/Seriellen Anschluss verschickt werden sollen.

Wird für die Anbindung von Home-Pcs an das Internet via RS232 Schnittstelle benutzt.

SLIP Paket:



Schwächen

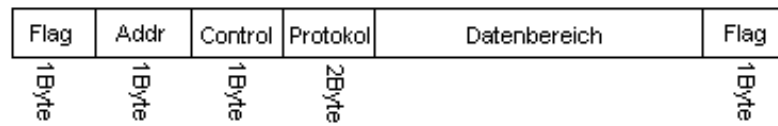
- Nur für IP (es gibt kein Typ-Feld)
- Es gibt keine Prüfsumme (aber IP, TCP, UDP haben eigene Prüfsummen)
- Jede Seite muss vor Beginn der Kommunikation die Gegenseite kennen

PPP – Point to Point Protokoll RFC 1661

Das Standard Protokoll zum anbinden von Home-Pcs via Modem oder ISDN ans Internet. Es umfasst 3 Komponenten:

1. Synchron (1Bit)/Asynchron (8Bit) Verbindungen nutzbar
2. Link Control Protokoll (LCP) RFC 1548
Verbindung-Aufbau, -Konfiguration, -Abbau
0. Eine Network Control Protokoll Familie welche die Verschiedenen Vermittlungsschicht Protokolle (IP, DecNet, AppleTalk) bedient.

PPP – Rahmen



FLAG: immer 0x7E
 ADDR: immer 0xFF
 CONTROL: immer 0x03
 PROTOKOL: 0x0021 → IP Daten im Datenbereich
 0xC021 → Link Control Daten im Datenbereich
 0x8021 → Network Control Daten im Datenbereich

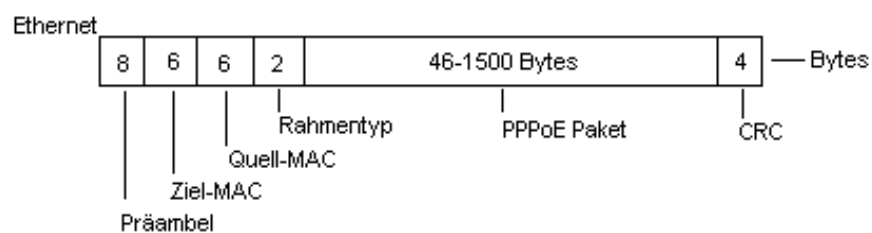
Verbesserungen von SLIP zu PPP

- Support von verschiedenen Protokoll Familien
- Dynamisches Aushandeln von Verbindungs-Daten möglich
(z.B. IP Adresse)
- Das LCP handelt die Verbindungs-Optionen aus

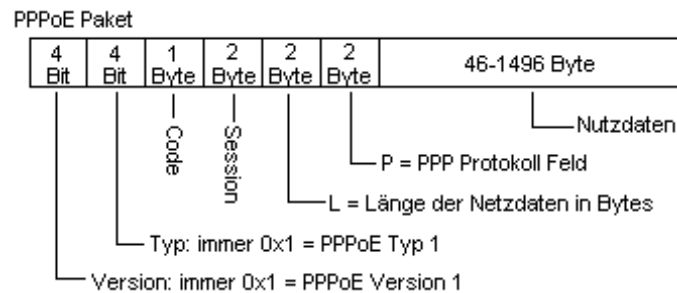
PPPoE – PPP over Ethernet RFC 2516

Wird in Deutschland (T-DSL/A-DSL) zwischen dem Home PC und dem Point of Presence des Internet Providers benutzt.

PPPoE Rahmen:



Rahmentyp: 0x8863 → PPPoE Discovery Paket (Auf, Abbau einer Verb.)
 0x8864 → PPPoE Session Paket



Version: immer 0x1 = PPPoE Version 1

Typ: immer 0x1 = PPPoE Typ 1

Code: 0x09 → PADI = PPPoE Active Discovery Initiation
 Wird gesendet, wenn sich ein Internet Nutzer (PC) via DSL Einwählen will.
 Zweck: Suchen eines Point-of-Presence, eines Einwahltones eines Internet-Providers
 Eth-Ziel-Adr: Broadcast Adresse
 Eth-Quell-Adr: die Adr. des anwählenden Gerätes (DSL-Router)

0x07 → PADO = PPPoE Active Discovery Offer
 Wird vom PoP gesendet, enthält die PoP Ethernet Adresse
 Pop Namen, sowie eine Dienstbezeichnung
 Es können mehrere PoP's antworten, das Home Gerät wählt einen PoP aus.

0x19 → PADR = PPPoE Active Discovery Request
 Sendet den Nutzer PC / den Nutzer DSL Router an den von Ihm gewählten PoP

0x65 → PADS = PPPoE Active Discovery Session Confirmation
 Sendet der PoP an den Nutzer, das Paket enthält eine SESSION-ID

0xA7 → PADT = PPPoE Active Discovery Termination
 Beendet die Verbindung zwischen Nutzer und PoP

Session:

L = Länge → Länge der Netzdaten in Bytes

P = PPP Protokoll-Feld

Repeater, Bridges und Switches

Die Größe von traditionellen LANs ist beschränkt und wird durch die maximal erlaubte Verzögerungszeit innerhalb eines Netzwerk-Segmentes bestimmt. ($\sim 10\mu s$) Die Verzögerungszeit muss wegen der Zugriffsverfahren CSMA/CD möglichst klein sein.

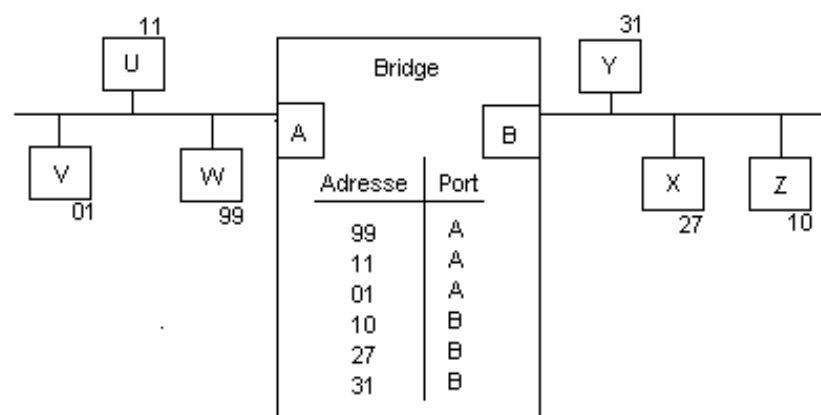
Ein **Repeater** ist ein Hardware-Gerät das zur Verbindung von je 2 LAN Segmenten benutzt wird.

- Er verstärkt und überträgt alle elektrischen Signale zwischen den Segmenten
- Quell- und Ziel-Netzwerk-Gerät erkennen NICHT, wenn sie über einen Repeater kommunizieren
- Maximal 4 Repeater dürfen zwischen 2 kommunizierenden Netzwerk-Geräten liegen

Nachteile:

- Repeater verstärken elektrische Signale, also auch Rauschen
- Repeater kennen keine Frames/Rahmen
- Es entsteht ein LAN, d.h. alle Netzwerk-Geräte bilden eine Collision-Domain, → d.h. ein Sender belegt das ganze Netzwerk

Brücken/Bridges sind einfach Computer mit CPU & Speicherplatz. Sie übertragen nur vollständige, fehlerfreie Rahmen/Frames. Weiterhin werten Sie die Frame-Header der ankommenden Rahmen/Frames aus.



W → X Brücke schickt Rahmen-Kopie über Schnittstelle B in Segment 2.
W → U Brücke tut nichts

- Die MAC/Port Tabelle wird automatisch aufgebaut. Die Brücke wertet die Sender-Adressen aller ankommenden Ethernet-Frames aus und ordnet sie einem Port zu. Man sagt die Brücke lernt.
- Die Tabelleneinträge werden von Zeit zu Zeit gelöscht, sie altern.
- Man spricht von Rahmenfilterung
- Natürlich werden Broadcast- und Multicast-Frames durch die Brücke weitergeleitet.
- Es ist parallele Kommunikation in den Teilsegmenten möglich
- Jedes Segment ist seine eigene Collision-Domain, das gesamte Netzwerk ist eine Broadcast Domain

Spanning Tree

Problem: zirkulierende Frames durch LAN-Schleifen. Brücken leiten fehlerfreie Frames weiter, sie wissen nicht ob sie diesen Frame schon einmal weitergeleitet haben.

Entstehung: Netzwerk sollte gegen Brücken-Ausfall immun werden.

Forderung: Brücken sollte LAN Schleifen erkennen, und verhindern.

Lösung: die Bridges bauen mit Hilfe eines Protokolles über einen verteilten Spanning-Tree Algorithmus ein Schleifenloses LAN auf.

Wie geht's? Die Brücken tauschen via BPDU (Bridge Protokol Data Unit) ihre Anschlussdaten aus und legen davon ausgehend ihre Ports auf AKTIV oder PASSIV, dadurch entsteht ein zyklenfreies LAN.

Spanning Tree Algorithmus – kurz, vereinfacht

Eine BPDU Nachricht enthält im wesentlichen folgende Informationen:

- Die Kennung der Bridge, die die BPDU Nachricht sendet
- Die Kennung der Bridge, welche die sendende Bridge für die Wurzel-Bridge hält
- Die Entfernung der sendenden Bridge zur Wurzel-Bridge

Anfangs halten sich alle Bridges für die Wurzel-Bridge und senden folgende Daten

(W=B2, E=0, SB=B2) (W=B3, E=0, SB=B3)

Alle Brücken erhalten von ihren Nachbarn solche Info-Pakete, die Informationen werden wie folgt ausgewertet:

Eine ankommende Nachricht ist besser und wird die vorher verschickte Nachricht ändern wenn

- Eine Wurzel mit kleinerer Kennung identifiziert
- Sie eine Wurzel mit gleicher Kennung jedoch kürzerer Entfernung identifiziert
- Die Wurzel Kennung und die Entfernung gleich sind, die sendende Bridge aber eine kleinere Kennung hat.

Die bessere Nachricht wird gespeichert, die Entfernungsangabe um 1 inkrementiert und versendet.

Beispiel B2 \rightarrow (W=B2, E=0, SB=2)
 B3 \rightarrow (W=B3, E=0, SB=3)

B3 empfängt die Nachricht von B2
B3 erkennt W=B2 ist $<$ W=B3
B3 versendet (W=B2, E=1, SB=B3)
Also hat B3, B2 als Wurzel anerkannt

Inzwischen hat B2, B1 als Wurzel anerkannt
B2 sendet (W=B1, E=1, SB=B2) an B3

B3 sendet in Zukunft (W=B2, E=2, SB=B3)

B5 hat B1 als Wurzel akzeptiert
und sendet an B3 (W=B1, E=1, SB=B5)

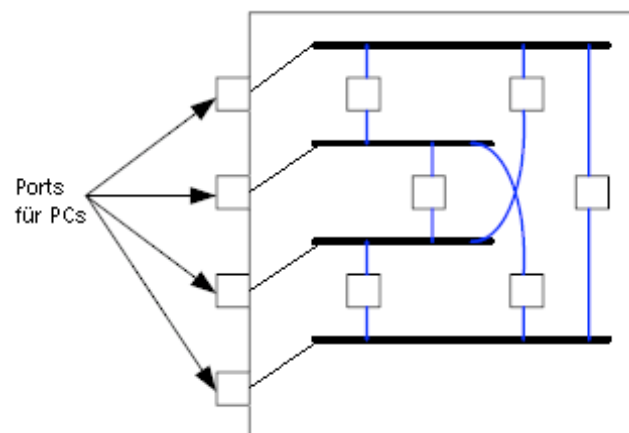
B3 (W=B2, E=2, SB=B3) * AKTIV
 (W=B5, E=2, SB=B3) PASSIV

Nach der Stabilisierung des Systems, sendet die Root Bridge periodisch ihre Daten, die die übrigen Bridges modifiziert weiterleiten. Fällt eine Bridge aus, so sendet sie keine Dateien weiter \rightarrow der Algorithmus beginnt von neuem.

Switches

Physisch ist ein Switch einem Hub ähnlich. An einem Hub werden mehrere Computer via Twisted Pair-Kabel angeschlossen. Ein Hub funktioniert wie ein Kabelkonzentrator.

Ein Switch simuliert ein mit Bridges aufgebautes LAN, wobei pro Segment ein Computer angeschlossen ist.



Ein Switch ist wie eine Bridge ein Gerät des Layer 2 – sie kennen Ethernet Pakete und können diese Auswerten.

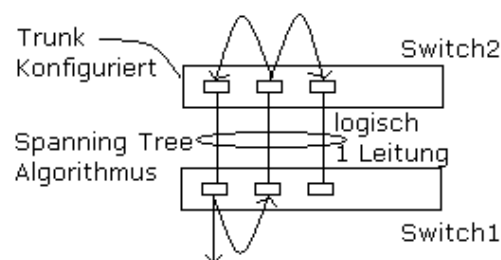
Der Netzwerktechnische Super-GAU – 27. Okt. 2000

Entstehung: Stadtwerke SB schalten um 7.45 eine 10KV Leitung in der Nähe der HTW

Folgerungen: kurze Spannungslose Zeit an der HTW

1. Die Telefonzentrale ist bis 11 Uhr gestört
2. HTW Backbone ständig überlastet

Der Fehler liegt in STL ! bei Switch STL2



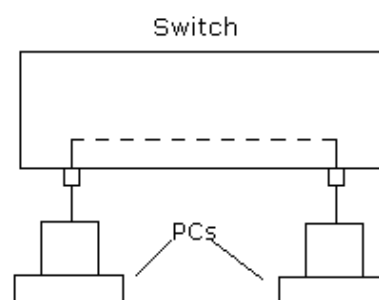
Durch den kurzen Stromausfall verfiel die Trunk Konfiguration und die 3 Physikalischen Leitungen wurden wie 3 Logische und nicht mehr als eine logische Leitung angesehen. So kreisten immer wieder Pakete vom Spanning Tree Algorithmus hin und her und überlasteten das System.

Switching Verfahren

Dynamisches Switching → Store and Forward Prinzip

Das empfangene Datenpaket wird ganz gespeichert [gepuffert] dann überprüft, bei Fehlerfreiheit weitergeleitet. Die Zeitspanne zwischen Empfang und Sendung wird „Latency Delay“ genannt.

Das Datenpaket kommt an einem Port an, die MAC Adresse des Senders wird mit dem Port verbunden. Aus der Header Analyse kennt die Switch Software die Empfängeradresse, darüber sucht sie den Empfänger Port und schaltet eine exklusive Verbindung zwischen Empfängerport und Senderport.



Cut through → die Datenpakete werden nicht mehr komplett gepuffert, eine Prüfsummenberechnung fällt weg.

Lediglich das Zieladressfeld wird ausgewertet, dann das Paket on-the-Fly weitergereicht.

VLAN – Virtuelles LAN

Was sind VLANs?

Warum sind VLANs notwendig?

Die HTW hat ein fast vollständig geschwitchtes Netzwerk

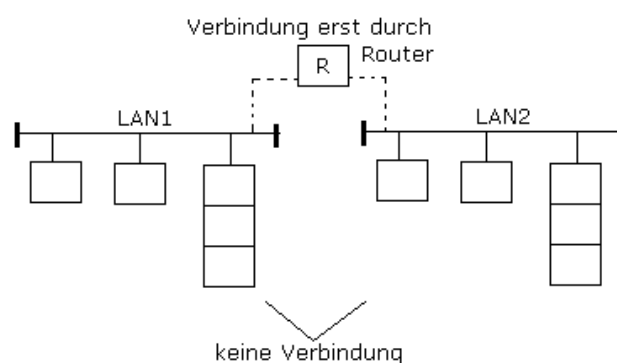
Vorteil: Jede an das Netzwerk angeschlossene Komponente (PC, Workstation, Drucker, Server, ...) könnte mit jeder anderen Komponente unabhängig vom Standort, direkt d.h. via Layer 2 (Ethernet), kommunizieren.

Nachteil (ohne VLAN)

- Broadcast Pakete belegen das gesamte Netz
- Zu viele Netzwerk Geräte
- Sicherheit

Lösung: Das physikalische EINE LAN wird durch die Switch Software logisch aufgeteilt oder segmentiert.
Logische LANs → Virtuelle LANs = VLANs

Wie? Netzwerkgeräte werden unabhängig von ihrem Standort zu VLANs zusammengefasst. (gleiche Broadcast Domain → direkter Kontakt auf Layer 2 [Ethernet]) Aber ohne Beschränkung der Standortwahl.



Beispiele Einsatz von VLANs

Bsp1: VLANs vereinfachen das wandern von Netzwerkgeräten

Szenario: Pauly wird Rektor der HTW, und will seine Arbeitsstation mitnehmen, wegen STL Verwaltung.

Traditionelle Lösung:

Im Rektorzimmer liegt anderes LAN! Arbeitsstation muss umkonfiguriert werden → neue IP, neuer Name, DNS Eintrag, STL Firewall muss umkonfiguriert werden.

VLAN Lösung:

Rektor Anschluss wird in das STL VLAN aufgenommen.

Bsp2: Abschottung von Verwaltungsrechnern

Szenario: die Verwaltungsbüros liegen im Bau 2 und Bau 8

Traditionelle Lösung:

Wegen der Distanz waren die Rechner in verschiedenen LANs untergebracht.

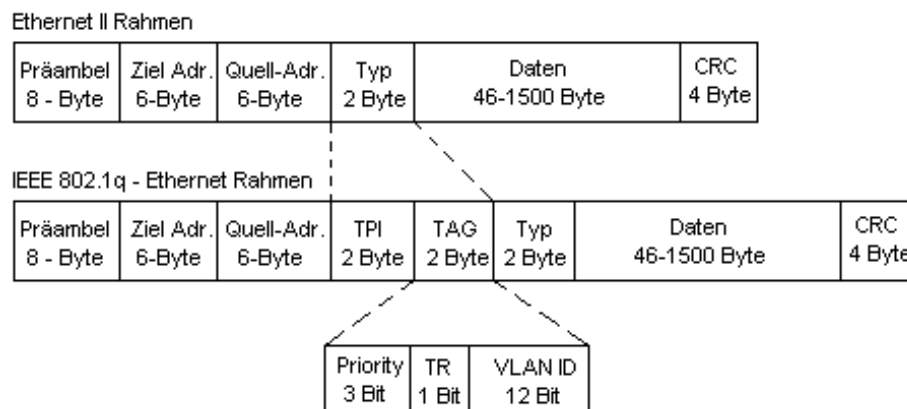
VLAN Lösung:

Alle Sicherheitsrelevanten Verwaltungsrechner gehören zu einem VLAN der hinter einer strengen Firewall liegt.

Wie entstehen VLANs in geschwichten Netzwerken?

- Es gibt Standardmethoden nach IEEE 802.1p und IEEE 802.1q, die 1998 veröffentlicht wurden und die es ermöglichen Herstellerunabhängige VLANs zu erzeugen
- Es wurde ein neues Ethernet-Rahmen-Format definiert (IEEE 802.1q) das um 4 Bytes länger ist als das Standard Ethernet Paket.
 $1518 \text{ Bytes} + 4 \text{ Bytes} = 1522 \text{ Bytes}$
- Beim IEEE 802.1q – Ethernet Rahmen werden 4 zusätzliche Bytes vor den Ethernet II Rahmentypfeld eingefügt
- Die Zugehörigkeit zu einem bestimmten VLAN wird über das TAG Feld bestimmt, so dass oft von VLAN-tagging gesprochen wird

Die zusätzlichen Felder im Einzelnen



TPI Tag Protocol Identifier Field
 0x800 → gibt an, dass es sich um ein Datenpaket mit VLAN Informationen handelt

TAG eigentlich TCI Tag Control Information Field
 Es besteht aus 3 Bereichen

- VLAN ID → 12 Bit lang → $2^{12} = 4096$ VLAN Ids
 - TR → 1 Bit – zeigt eine TokenRing Encapsulation an.
 - Priority → 3 Bit – legt die PRIORITY des Datenpaketes fest.
 Es gibt 8 Prioritätsstufen, die als CoS Class of Service bezeichnet werden.
- Jedes Datenpaket, das zu einem VLAN gehört trägt dessen eindeutige VLAN ID
 - Die VLAN ID wird vom Netzwerk Admin in Switch Verbund konfiguriert und je nach VLAN Typ in die Datenpakete eingeschrieben.

Bsp VLAN Typ Port grouping VLAN

Das am Switch Port angeschlossene Netzwerk Gerät sendet normale Ethernet Rahmen. Der Switch erzeugt einen IEEE802.1q Ethernet Rahmen, übernimmt alle Informationen von den Ursprungs Rahmen, füllt die zusätzlichen Felder auf und versendet den neuen Rahmen.

Moderne Netzwerk Karten können außer Ethernet II auch IEEE 802.1q Datenpakete versenden, wenn sie dafür konfiguriert werden. (Solaris 10 – Servernetzwerk im STL)

VLAN Typen

Port grouping VLAN – gängigster Typ

- Die einzelnen Ports eines oder mehrerer Switches werden einem VLAN fest zugeordnet
- Alle an diesen Ports angeschlossenen Geräte gehören zu dem mit dem Port verbundenen VLAN

Vorteile

- Einfach zu konfigurieren
- Einfache Fehlersuche, der Port gehört immer zum gleichen VLAN
- Keine neue Endgeräte Software
- Keine Konfiguration des Endgerätes
- Das VLAN ist für die Endgeräte transparent
- Eine Migration ist einfach möglich

Nachteile

- Keine dynamische Konfiguration auf Hinblick Port → VLAN zugehörigkeit
(mehrere LAN Anschluss Dosen pro Raum z.B. 8202 4 LAN Dosen HTW (gelb), STL, ..., ...)

Mac – Layer – Grouping – VLAN

Mac – Adressen – basiertes – VLAN

- Die Mac Adresse des Netzwerk Gerätes ist das Gruppierungsmerkmal
- D.h. der Netzwerk Admin pflegt eine Switch Übergreifende Datenbank mit einen Datenpaar: MAC-Adr ← → VLAN ID
- Der Switch packt anhand der Sender Adresse, des ankommenden Ethernet Paketes in ein 802.1q Paket des konfigurierten VLANs um.

Vorteile:

- Das Netzwerk Gerät gehört zu einem VLAN
(Laptop wandert von Hörsaal zu Hörsaal und ist immer im gleichen VLAN)
- Mehrere VLANs über ein Switch-Port, eine Netzwerk Anschlussdose erreichbar.

Nachteile:

- Hoher Konfigurationsaufwand
- MAC Adresse sind bei neuen Netzwerk Karten **kofigurierbar**
- Fehlersuche komplex

Network – Layer – Grouping VLAN

Schicht 3 Protokoll basiertes VLAN

Diese werden über sogenannte Layer 3 Switches realisiert, d.h. die Grouping Merkmale werden von den übergeordneten Protokollen bestimmt.

Bsp1 die Zuordnung geschieht über den Typ des Schicht 3 Protokolles

Port 6 → IP X Datenpaket in Ethernet Paket siehe Typfeld des Ethernet Paketes → Paket gehört zu VLAN X

Port 6 → IP Y ... → VLAN Y

Bsp2 die Zuordnung wird über die Adressen der Schicht 3 vorgenommen.

- Ist die IP Adresse des Senders an Port 6 = 134.96.216.* → VLAN X
- Ist die IP Adresse des Senders an Port 6 = 134.96.200.* → VLAN Y

Layer 2 Switching

- Zur Weiterleitung des Datenpaketes (Ethernet-Frames) werden nur die Sende- und Empfänger- MAC Adressen (Ethernet-Adressen) verwendet.
- Die Switch Software analysiert nur den Frame Header und verteilt die empfangenen Pakete an die entsprechenden Ports.
- Ein Switch führt intern eine MAC/Port Tabelle.

Beispiel

Ein Gbit Switch verwaltet ca 20 000 Adresseinträge und kann bis zu 100 000 MAC Adressen/Sekunde lernen

- Hat man VLANs Konfiguriert, so können Netzwerk Geräte aus verschiedenen VLANs nur über Router miteinander kommunizieren
→ Neues Gerät → mehr Aufwand → Idee Layer 3 Switching
- Ein reines Layer 3 Switching gibt es (noch) nicht, da die aufwändige Routing-Funktionalität die Switch Software langsamer machen würde und natürlich mit dem Layer 2 Switching gekoppelt sein muss

Layer 3 Switching

Flow-Based-Switching: Fast IP bei 3Com
 Secure Fast Cabletron

Fast IP geht von festen Zuordnungen IP Adr \leftrightarrow MAC Adr aus, basiert auf den NEXT HOP RESOLUTION PROTOCOL (NHRP) gemäß RFC 2332 (1998)

durch das NHRP ist eine IP Kommunikation zwischen Netzwerk Geräten in verschiedenen IP Subnetzwerken möglich OHNE STÄNDIG einen Router zu verwenden.

Layer 4 Switching

Problem: Welcher Dienst benötigt welche Netzwerkbandbreite?
 www – http, Videokonferenz. VoIP, ftp

Der Switch wertet die Header Daten der Layer 4 Protokolle aus
→ er priorisiert die Datenpakete anhand der Ports.

WAN Technologien und Routing

Ein LAN vernetzt eine kleine Anzahl von Computern innerhalb von Gebäuden. Es ist in Bezug auf Ausdehnung und Anzahl der Computer begrenzt, da nur ein Übertragungsmedium benutzt wird.

Ein WAN sollte beliebig skalierbar sein, d.h. bei Bedarf immer erweiterbar. Es sollte beliebig viele Computer an beliebig vielen Standorten, mit beliebigen Entfernungen verbinden. Ein WAN ist auch ein paketvermittelndes Netzwerk und soll im kleinen (STL Labor) so funktionieren wie im großen (Internet).

Wichtige WAN Eigenschaften

- Kein gemeinsames Übertragungsmedium
- Kein Einheitliches Übertragungsmedium
- Zentrales Element: Router [Paketvermittler]
- Alle an einem WAN angeschlossenen Netzwerk Geräte können parallel miteinander kommunizieren (parallel = durch das Store and Forward Prinzip)
- Eigenes Adressierungs-Schema

Store and Forward

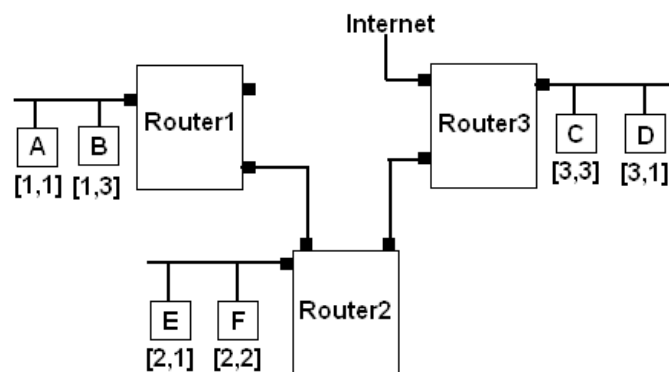
Prinzip:

- Ein Computer sendet Datenpakete an den Router
- Router empfängt das Paket, speichert es, wertet Paket-Header aus
- Router sucht weiteren Weg für das Paket (der nächste Router oder das Zielsystem)
- Wenn Weg frei, dann wird Paket auf den Weg geschickt
- Wenn Weg besetzt, dann Paket in Warteschlange

Aus der Sicht des Computers arbeiten WANs wie LANs. WANs verwenden hierarchische Adressierungs-Schematas.

Jedes Netzwerk Gerät besitzt 2 Adressen, eine LAN Adresse (Ethernet Adresse), und eine WAN Adresse (IP Adresse).

Kommunikations Beispiele



Ca → Cb Ca erkennt an WAN Adresse von Cb, dass Cb im gleichen WAN Netz
== gleichen LAN liegt → Ca benutzt LAN Adresse von Cb und
kommuniziert damit direkt mit Cb

Ca → Ce Ca erkennt an WAN Adresse von Ce. Ce in anderem WAN Netz. Ca
sendet Daten an Router 1 via LAN. Router 1 erkennt Router 2 ist
zuständig für Ce → Paket an Router 2 senden.

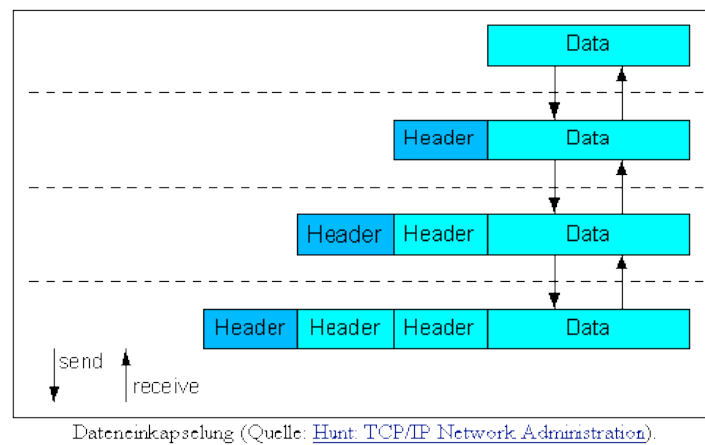
Ca → Cd Ca erkennt an WAN Adresse von Cd, Cd in anderem WAN Netz
→ sendet Paket an Router 1. Router 1 erkennt es gibt keinen Router
der direkt Cd ansprechen kann, er sendet das Paket an seinen Default
Router: Router 2

→ Teilstreckenverfahren [Next-Hop-Forwarding]

Routing Tabelle eines Computers anschauen: netstat -r[n]

III. TCP / IP im Unix Umfeld

Protokolle – Protokoll Familien – Protokoll Stacks



Internetworks & IP Adressen

Internetwork bezeichnet eine beliebige Anzahl zusammengeschalteter (LAN) Netzwerke, die Pakete zwischen beliebigen Hosts transportieren können.

Ein Internetwork ist ein virtuelles Netzwerk, es wird durch die IP Software erzeugt.

$$\sum_{N=1}^x \text{physischen Netzwerken}$$

Es löst das Problem der LAN Heterogenität, denn es gibt viele LAN Technologien.

Zu bewältigende Probleme:

1. Definition eines einheitlichen Adressierungs Schemas
 2. Weg finden – Routing
- IP ist die Vermittlungsschicht die TCP/IP Protokoll-Stacks und damit die entscheidende Schicht die die Netze verknüpft.
 - IP ermöglicht die Kommunikation 2er beliebiger Rechner in einem Netzwerk Verbund, es vereinheitlicht die Schicht auf das Gesamt Netzwerk.
 - IP erzeugt ein Internet
 - IP Software läuft auf allen Hosts und Routern im Internet
 - Host Rechner, die an ein IP Netzwerk angeschlossen sind, besitzen eine Netzwerk Karte, eine IP Adresse
 - Multihomed Hosts → Rechner, die an mehrere IP Netzwerke angeschlossen sind, mehrere Netzwerk Karten, mehrere IP Adressen aber OHNE Routing funktionallität

- Router sind dedizierte Rechner, die Netzwerke verbinden, also 2 oder mehrere Netzwerk Karten besitzen. Sie leiten IP Datenpakete (IP Datagramme, IP Datagram) zu verschiedenen IP Netzwerken weiter
- Gateway != Router
- IP ist ein verbindungsloser Datagramm Dienst → dieser Dienst wird als “Best Effort“ Dienst bezeichnet (IP bemüht sich die Datagramme auszuliefern)
- IP ist unzuverlässig, denn es fehlen Mechanismen um verlorene Pakete zu wiederholen, doppelte Pakete auszusondern, zu erkennen das Pakete außerhalb der Sendereihenfolge ankommen.

Die IP Adresse → 32 Bit = 4 Byte lange Adresse die jedem Netzwerk Gerät, das im Internet kommunizieren will/soll zugewiesen wird.

Zuweisung

statisch: /etc/hosts

dynamisch: DHCP

IP Adresse wird gewöhnlich in dem Punkt Dezimal Schema angegeben:

134.96.216.10

Jedes IP Datagramm enthält in seinem IP Header die IP Adresse des Senders und des Empfängers.

IP Adresshierarchie

Jede IP Adresse wird logisch in 2 Teilbereiche gegliedert

→ einen Präfix → Netzwerkadresse

→ einen Suffix → Host Adresse

Diese Aufteilung erlaubt:

1. Jedem Client eine weltweit eindeutige IP Adresse zuzuweisen
2. Die Netzwerk Adressvergabe geschieht Global
3. Die Host Adressvergabe geschieht Lokal

IP Adressklassen

Klasse A - 0.0.0.0 – 127.255.255.255
8 Bit Netzwerk Adresse, es gibt 128 Klasse A Netzwerke

Klasse B - 128.0.0.0 – 191.255.255.255
16 Bit Netzwerk Adresse, es gibt 16 384 Klasse B Netzwerke
z.B. 134.96.0.0

Klasse C - 192.0.0.0 – 223.255.255.255
24 Bit → $2,1 \cdot 10^6$ Adressen

Classless IP Adressen – Klasse B Adressen welche Teilbereiche als Klasse C Adressen verkaufen.

Ipv6 → 16 Bytes IP Adressen = 128 Bit

IP Adress Kategorien


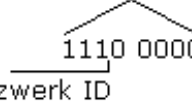
Unicast Adressen → für einen Host

Multicast Adressen → für eine bestimmte Gruppe von Hosts bestimmt
Multicastgruppe

Broadcast Adressen → für alle Hosts eines bestimmten IP Netzwerks

Spezielle IP Adressen

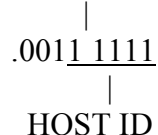
Netzwerk Adresse ist die Adresse eines IP Netzwerkes → f(Netzwerkmaske)

	Netzwerk Adresse	Netzwerk Maske
UniSB	134.96.0.0	255.255.0.0
STL	134.96.216.0	255.255.255.0
campus	134.96.208.32	255.255.255.224
		

Netzwerkteil auf 1 gestellt
Host Teil auf 0 gestellt

Gerichtete Broadcast Adresse

→ erreicht alle in einem IP Netzwerk vorhandenen Netzwerk Geräte
gerBroadcastAdr = f(Netzwerk Maske)

	Netzwerk Adresse	Netzwerk Maske	gerBroadcast
UniSB	134.96.0.0	255.255.0.0	134.96.255.255
STL	134.96.216.0	255.255.255.0	134.96.216.255
Campus	134.96.208.32	255.255.255.0	134.96.208. <u>63</u>
			

Ping -s 134.96.216.255 1 40

Begrenzte Broadcast Adresse

→ ist auf das lokale LAN Segment begrenzt, es gibt keinen Routerübergang

→ 255.255.255.255 ein Router leitet ein IP Paket mit dieser Ziel IP Adresse NICHT weiter

Ping -s 255.255.255.255 1 40

Loop Back Adresse

→ 127.0.0.1 – internes IP Netzwerk eines Rechners (localhost)

DHCP – Dynamic Host Configuration Protocol

Ein Protokoll das die Möglichkeiten der begrenzten IP Broadcast Adresse nutzt.

Was muss ein Host alles wissen, um im Internet kommunizieren zu können?

- Router – IP Adresse (default)
- DNS Server (IP Adressen $\leftarrow \rightarrow$ Namen)
- Host IP Adresse
- Netzwerkmaske
- ...

Konfiguration \rightarrow manuell oder Automatisch via DHCP

DHCP benötigt einen DHCP Server, der die Konfiguration Daten für den Client liefern muss.

Problem 1: Wie findet der Host den DHCP Server?

Lösung \rightarrow via gerichtete Broadcast IP Adresse

Siehe auch ~pauly/internet-vorlesung/DHCP

DHCP Kommunikations Ablauf – Protokoll

1. Client sendet DHCP DISCOVER via UDP
 \rightarrow IP begr. Broadcast \rightarrow Ethernet Broadcast
2. Server antwortet DHCP OFFER via UDP
 \rightarrow IP Broadcast \rightarrow Ethernet Broadcast
3. Client sendet DHCP REQUEST via UDP
 \rightarrow IP Broadcast \rightarrow Ethernet Broadcast
4. Server sendet DHCP ACK via UDP
 \rightarrow IP Broadcast \rightarrow Ethernet Broadcast
5. Client sendet zyklisch DHCP REQUEST via UDP
 \rightarrow IP Unicast \rightarrow Ethernet Unicast
6. Server sendet als Antwort DHCP ACK via UDP
 \rightarrow IP Unicast \rightarrow Ethernet Unicast

Multicast IP Adressen

- Klasse D \rightarrow 224.0.0.0 – 239.255.255.255
- IP Multicasting: Ansprechen von Rechner Gruppen mittels eines IP Paketes
- Via UDP

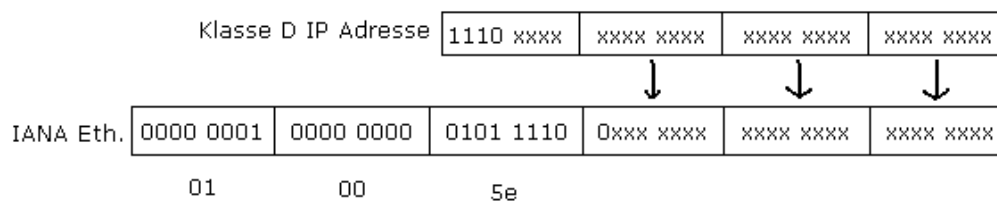
Merkmale

Im Ethernet Netzwerk ist Multicasting möglich, denn auch Broadcasting ist möglich. Die Ethernet Pakete können alle Ethernet Karten eines LANs erreichen.

- Die Datenquelle, der sendende Rechner will eine Gruppe von Hosts erreichen
- Hosts können Multicastgruppen beitreten oder verlassen
- Hosts können in mehreren Multicastgruppen aktiv sein
- Mittels IGMP (Internet Group Manage Protocol) stimmen sich Router über die Weiterleitung von Multicast IP Paketen ab
- Es gibt **permanente** Multicast Gruppen die von der IANA festgelegt sind
<http://www.iana.org/numbers.html>

IP – Multicast → Ethernet Multicast Adresse

- Die IANA besitzt ein Ethernet Adressblock 01:00:5e
- IP Multicast Adressen werden auf die IANA Ethernet Adressen abgebildet
01:00:FF:FF:FF



Aufgabe eines Netzwerk Interfaces / Netzwerk Karte

Eine Ethernet Netzwerkkarte filtert aus dem Datenstrom des Netzwerks folgende Datenpakete, wenn als Zieladresse

- Die eigene Ethernet Adresse steht
- Die Ethernet Broadcast Adresse steht
- Eine Multicast Adresse steht, zu einer Gruppe zu der der Rechner gehört

IP Adressen für private Netzwerke

Warum? Zum Aufbau von privaten/firmeninternen Netzwerken auf TCP/IP Basis

Adress Bereiche:

Klasse A	0.0.0.0	-	10.255.255.255
Klasse B	172.16.0.0	-	172.31.255.255
Klasse C	192.168.0.0	-	192.168.255.255

Wichtig:

- Keine Registrierung bei der IANA
- Nicht Routbar im Internet

IP Subnetting / IP Unternetzwerke

UNI-SB: 134.96.0.0 Netzwerk Adresse: 134.96.0.0
 Netzwerk Maske: 134.96.255.255

Ziel: einen gegebenen IP Adressbereich **dezentral** verwalten

Beispiel zu Subnetting:

134.96.208.0 ← Netzwerk Adresse
255.255.255.0 ← Netzwerk Maske
134.96.208.255 ← Netzwerk Broadcast
134.96.208.1 ← Router Adresse
134.96.208.2 – 134.96.208.254 ← Host Adressen

Gesucht sind 8 gleich große Netzwerke auf der vorhandenen IP Adresse 134.96.208.0

$$0 - 255 = 256 : 8 = 32$$

1. Netz 134.96.208.0 - 134.96.208.31
 134.96.208.0 - Netzwerk
 134.96.208.1 - Router
 134.96.208.31 - Broadcast

2. Netz 134.96.208.32 - Netzwerk
 134.96.208.33 - Router
 134.96.208.63 - Broadcast

Netzwerk Maske: 255.255.255.1110 0000

128 + 64 + 32 = 224

$2^3 = 8$ Mögliche Netzwerke

Übersicht:

Netzwerk Adresse	Netzwerk Maske	Router	Broadcast	Host Nr.
134.96.208.0	225.225.225.224	.1	.31	.2-.30
134.96.208.32	225.225.225.224	.33	.63	.34-.62
134.96.208.64	225.225.225.224	.65	.95	.66-.94
134.96.208.96				
134.96.208.128				
134.96.208.160				
134.96.208.192				
134.96.208.224				
	134.96.216.99 & 255.255.255.0		134.96.216.101 & 255.255.255.0	
	-----		-----	
	134.96.216.0	↔	134.96.216.0	
	!= → anderes LAN		== → gleiches LAN !	

Die Verbindung von IP zu Ethernet Adressen

Kennt ein Host die IP Adresse seines Kommunikationspartners, so kann er durch seine Netzwerk Maske erfahren, ob dieser im gleichen Netzwerk angeschlossen ist wie er. In diesem Fall muss er die Ethernet Adresse seines Kommunikationspartners herausfinden.

[gleiches IP Netzwerk == gleiches LAN / VLAN]

Bei TCP/IP wird im LAN eine Nachrichtenaustauschmethode in Verbindung mit Tabellensuchen eingesetzt → ARP/RARP – ARP Cache

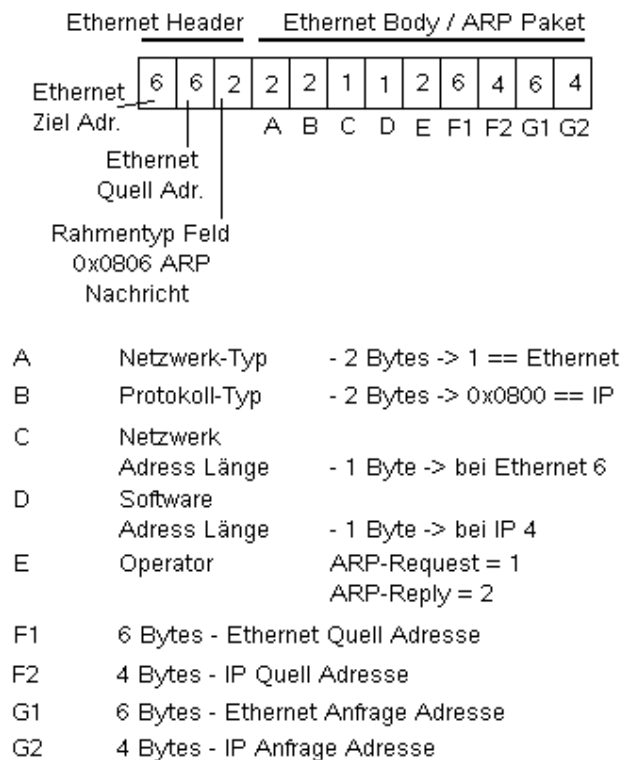
Im WAN Bereich benutzt man Tabellensuchen.

Problem: Host1 soll mit Host2 kommunizieren, Host1 kennt nur die IP Adresse von Host3. Zur Kommunikation testet Host1 die IP Adresse von Host 3 mittels Netzwerk Maske → Host3 in gleichem IP Netzwerk wie Host1
Wie lautet die Netzwerk Adresse, die MAC Adresse von Host3 ???

Lösung:

- Berechnung der MAC Adresse z.B. Multicasting
- Der Admin erzeugt auf jedem Host eine Übersetzungstabelle
IP Adresse ↔ MAC Adresse
- Host baut sich die Übersetzungstabelle selbst auf → ARP

Format des ARP Paketes



Ethernet Ziel Adresse ist bei einer ARP Anfrage
ARP Request == Ethernet Broadcast Adresse

Ablauf der ARP Anfrage / Antwort

1. Computer Y sendet eine ARP Anfrage mittels Ethernet Broadcast ins (V)LAN
2. Alle Computer des (V)LAN empfangen und verarbeiten das ARP Request Paket
3. Anhand des Operations Feldes erkennen sie die Nachrichtenart = ARP Anfrage und vergleichen ihre IP Adresse mit der Anfrage IP Adresse. Sind diese gleich (Computer A) wird ein ARP Antwort Paket generiert und direkt an den anfragenden Computer Y gesendet.
4. Um die ARP Kommunikation effektiver zu gestalten speichert Computer A die Computer Y Adressen in seinem ARP Cache == ARP Tabelle
5. Die ARP Cache Einträge haben eine Lebenszeit zwischen 2s und 1/2 h

Versuch:

1. ARP Tabelle ansehen: `arp -a`
2. Root → `snoop ziel computer`
3. Ping ziel computer

Überflüssiges ARP → Gratuitous ARP

Beim booten sendet ein Host einen ARP Request mit seiner IP Adresse als Anfrage Adresse. Kommt keine Antwort dann ist alles OK! Kommt eine Antwort so benutzt ein anderer Host die gleiche IP Adresse

→ Fehler, Meldung „Duplicate IP Address sent from Ethernet: MAC Adresse“

Angriffe aus dem lokalen Netz

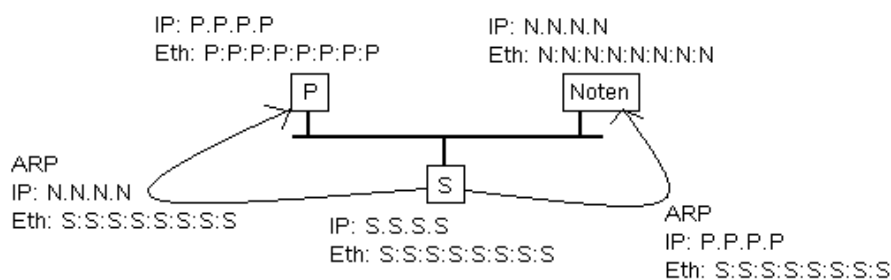
70% aller Angriffe erfolgen von innen.

- MAC Spoofing – reinlegen, verschleiern, manipulieren
Der Angreifer benutzt eine fremde MAC Adresse [konfigurierbare MAC]
(Stumpfe Abwehr: Switch Port Security)
- MAC Flooding / CAM Flooding
Der Angreifer versendet tausende MAC Adressen an den Switch dessen CAM Tabelle läuft über → alte oder schlecht konfigurierte Switches schalten auf Notprogramm um, sie spielen Hub.
- ARP Spoofing / ARP Poisoning, vergiften
Der Angreifer versendet gefälschte ARP Pakete

Gefälschte ARP Pakete?!

ARP ist zustandslos! Die ARP Software bearbeitet alle ankommenden ARP Reply's oder ARP Antworten.

Man in the middle attack



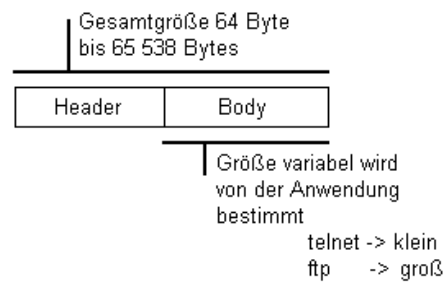
IP Datagramme – Routing Fragmentierung

IP ermöglicht es dem Sender einzelne Datenpakete == Datagramme == Datagram zu übertragen, wobei jedes Datagramm unabhängig mit seinen Informationen durch das Netz zum Empfänger wandert.

IP ist **unzuverlässig** – es gibt keine Garantie, dass ein IP Datagramm beim Ziel ankommt.

Verbindungslos – es verwaltet keine Statusinformationen über IP Datagramme
→ RFC 791

Aufbau eines IP Datagramm



IP Paket

VERS	H-LEN	SERVICE TYPE	TOTAL LENGTH	IDENTIFICATION	FLAGS	FRAGMENT OFFSET	TIME TO LIVE	P-TYPE	HEADER CHECKSUM	SOURCE IP ADDRESS	DESTINATION IP ADDRESS	IP OPTIONS	PADDING	Nutzdaten....
------	-------	-----------------	-----------------	----------------	-------	--------------------	-----------------	--------	--------------------	----------------------	---------------------------	---------------	---------	---------------

VERS → 4 Bit → IPv4

H-LEN → Header Länge in Anzahl der 32 Bit Zeilen-Worte

SERVICE
TYPE

→ Type of Service (ToS)
8 Bit → die ersten 3 Bit sind ausgenutzt
es folgen 4 ToS Bits
es folgt ein 0-Bit

4 ToS - Verzögerung zu minimieren
- Durchsatz zu maximieren
- Zuverlässigkeit zu maximieren
- kosten zu minimieren
→ RFC 1340/1349

TOTAL
LENGTH

→ Gesamtlänge des IP Datagramm ≤ 65535 Bytes
! kein Host muss IP Datagramme > 576 Bytes annehmen
! NFS Umgebung – IP 8192 Bytes -> 4096 Bytes

IDENTIFICATION →

FLAGS → Fragmentierung

FRAGMENT OFFSET →

TIME TO
LIVE

→ Obergrenze der möglichen Routerübergänge (32 bzw 64)

P-TYPE

→ Protokoll Typ der transportierten Daten → TCP (6), UDP (17)
Siehe /etc/protocols

HEADER

CHECKSUM → IP Header Checksumme

TCP/UDP haben eigene Checksums !!!
RFC 1071

SOURCE IP
ADDRESS → Senderadresse

DESTINATION
IP ADDRESS → Empfängeradresse

IP OPTIONS → kein Pflichtteil, wenig benutzt

PADDING → füllt Zeile auf 3 Bit auf.

Der Anwendungs Datenstrom wird von der TCP/IP Software in IP Datagramme aufgeteilt, diese werden eventuell stückweise via Ethernet verschickt.
Jeder Host muss eine erste Routing Entscheidung treffen.

Problem: ein IP Datagramm kann bis zu 64 kByte groß sein, aber jede Hardware Technologie hat ihre eigenen Vorstellungen über die maximale größe ihrer Rahmen → MTU
ifconfig -a → Ethernet → 1500

Lösung:

- a. Wähle IP Datagramm klein genug, so dass es von allen Hardware Technologien transportiert werden kann (ATM – nur 49 Byte !!!)
- b. IP Pakete aufteilen und **am Ende** beim Empfänger wieder zusammenfassen

Ein Host teilt, falls nötig, IP Datagramme gemäß seiner lokalen Hardware Technologie (Ethernet 1500 Bytes) auf, der Zielhost fügt das IP Datagramm wieder zusammen.

Ein Router fragmentiert (teilt auf) ein IP Datagramm wenn er Netzwerke mit verschiedenen Hardware Technologien verbindet und das Sender Netzwerk eine größere MTU als das Empfänger Netzwerk besitzt.

Ein Router fügt **nie** IP Datagramme wieder zusammen.

Jedes Fragment ist ein vollständiges IP Datagramm das unabhängig von den anderen Fragmenten durch das Netz wandert.

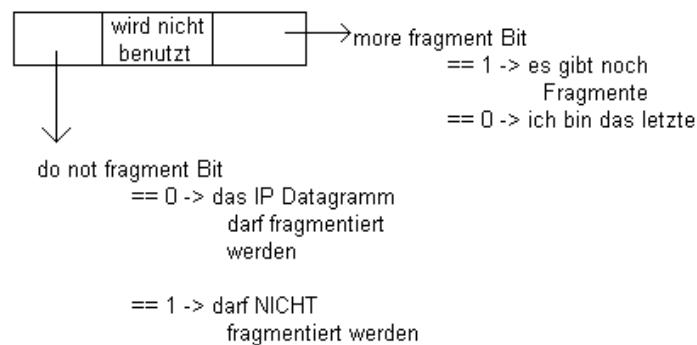
Heutzutage benutzen viele TCP/IP Implementierungen das sogenannte Path MTU Discovery zum finden der kleinsten MTU auf dem Weg von Sender zum Empfänger.

Fragmentierung – Technisch

Jedes Fragment hat das gleiche Format wie das ursprüngliche IP Datagramm. D.h. ALLE Header Felder des ursprünglichen IP Datagramm werden unverändert übernommen, außer das FLAGS und das FRAGMENT_OFFSET Feld.

IDENTIFICATION → eindeutige Nummer eines IP Datagramm, identifiziert ein IP Datagramm, alle Fragmente eines IP Datagramms enthalten das gleiche IDENTIFICATION Feld

Das FLAGS Feld – 3 Bit lang



Das FRAGMENT-OFFSET Feld

Bezeichnet die genaue Lage der im Fragment enthaltenen Daten im original IP Datagramm

Ein IP Datagramm kann nur dann reassembliert (zusammengesetzt) werden, wenn **alle** seine Fragmente angekommen sind.

Die IP Software berechnet aus den letzten Fragmente die Gesamtzahl der zu empfangenden IP Datagramm Bytes.

Fragment Offset + Aktuelle (TOTAL Lengs H LENGTH)
== Anzahl der Bytes, des original IP Datagramms

Die IP Software wartet bis alle Fragmente eingetroffen sind, jedoch nicht unendlich lange

Das ICMP – Internet Control Message Protocol

Es hilft der IP Software, Fehler während des Datenverkehrs zu erkennen und zu melden.

ICMP Paket

Typ	Code	Prüfsumme	Inhalt f. (Typ+Code)
-----	------	-----------	-------------------------

Eine ICMP Fehlermeldung enthält als Daten den IP Header und die ersten 8 IP Datenbytes (TCP/UDP Header) des auslösenden IP Datagrammes.

Durch diese Daten kann die IP Software dem sendenden Prozess mitteilen, dass seine Daten ihr Ziel nicht erreicht haben.

Die wichtigsten ICMP Nachrichten

ICMP Type(4) source quench – (Quelle dämpfen)

Ein Router sendet diese Meldung wenn seine Sendepufferwarteschlangen drohen überzulaufen. Der sendende Host wird veranlasst seine Senderate herunterzufahren (zu dämpfen)

Ein Host sendet diese Meldung, wenn er die IP Datagramme nicht so schnell verarbeiten kann, wie sie ankommen.

ICMP Type(11) time exceeded – (Zeit überschritten)

Ein Router sendet diese Meldung wenn er bei einem IP Datagramm das TTL Feld auf 0 dekrementiert hat.

Ein Host sendet diese Meldung, wenn sein interner Timer für das reassemblieren eines IP Datagramms abläuft bevor alle Fragmente des IP Datagramms angekommen sind.

ICMP Type(3) destination unreachable

Ein Router stellt fest, dass er ein IP Datagramm nicht an sein Ziel weiterreichen kann, dann sendet er diese ICMP Meldung

Ein Host sendet diese ICMP Meldung, wenn das im IP Datagramm angegebene höhere Protokoll auf diesem Host nicht ansprechbar ist.

ICMP Type(8) echo request

ICMP Type(0) echo reply

Hosts/Router sollten immer auf ein ICMP Echo Request mit einem ICMP Echo Reply antworten

ICMP Anwendungen

ping

ping Host is alive - Route existiert, IP Adresse gültig, IP Software läuft

no answer from – Route gestört, Rechner aus, IP Software aus

tracert

- Ist UDP basiert
- Wertet die ICMP Meldungen TIME EXCEEDED und DESTINATION UNREACHABLE aus.

Der Mechanismus:

Tracert versendet UDP Datagramme via IP Datagramm mit TTL=1..x, an den Zielrechner. Es sendet UDP Pakete an Port 33434 folgende.

Routing – Begriffe – Verfahren – Protokolle

Router \leftrightarrow Gateway

→ Router und Routing Protokolle arbeiten in/auf dem Layer 3 des ISO 7 Schichten Modelles (IP Ebene)

Man spricht von

- DIRECT Routing - wenn die kommunizierenden Hosts im gleichen IP Netzwerk (LAN) liegen. (Repeater/Hubs/Bridges/Switches sind transparente Elemente)
- INDIRECT Routing – wenn zwischen den kommunizierenden Hosts mindestens ein Router liegt. Wenn die kommunizierenden Hosts in verschiedenen LANs liegen.
- DEFAULT Routing – wenn eine Default Route benutzt wird um die IP Datagramme weiter zu leiten.

Routing – Statisch / Dynamisch

Bei uns genutzt: beim booten eines Systems wird die Routing Tabelle aufgebaut

Standard Route: eigenes Netz, Multicast Netz für lokales IP Netzwerk, loopback

Default Route: Solaris/Linux → /etc/defaultrouter
Win XP → Konfig des Standard Gateways

Statisches Routing wird benutzt in

- Kleinen Netzwerken
- Wenn Netzwerk nur einen Router besitzt

Dynamisches Routing \leftrightarrow Adaptives Routing (realisiert über Routing Protokolle)

- Es gibt mehrere Router im Netzwerk
- Es werden sogenannte Autonome Systeme miteinander verbunden

Routing Protokolle:

- RIP – Routing Information Protocol (RFC 1058)
- OSPF – Open Shortest Path First (RFC 1247)
- BGP – Border Gateway Protocol (RFC 1771)

Ein Routing Daemon baut mit Hilfe seines Routing Protokolls eine Routing Tabelle auf und verwaltet sie.

Routing Protokolle Einteilung

Nach Verfahren:

- Distanzvektor Routing (RIP)
- Link State Routing (OSPF)

Nach Reichweite:

- Intradomain Routing (RIP/IBGP)
- Interdomain Routing (RGP/BGP)

RIP – Routing Information Protocol

- Die Routingtabelle wird als Vektor interpretiert in dem die Entfernungskosten zu allen erreichbaren Netzwerken gespeichert sind.
- Alle 30 Sekunden verschickt ein Router eine Aktualisierungs Nachricht an seine Nachbarn, oder wenn sich seine Routingtabelle geändert hat.
- Ist auf kleine Netzwerke begrenzt, Max: 15 Hops – Routerübergänge
- Kosten 1 Hop = 1
- RIP benutzt UDP als Transportprotokoll

Beispiel siehe Zusatzkopien

OSPF – Open Shortest Path First

Prinzip:

- Jeder Router weiß wie er seine direkten Nachbarn erreicht.
- Jeder Router verschickt nur diese Informationen an alle Router des Netzes
→ zufälliges Fluten

Warum verwendet man Router

- IP Netzwerke verbinden
- Netzwerke voneinander Abschotten
- Router Firewall funktion
- Verschiedene Hardwaretechnologien verbinden

Die TCP/IP Transportschicht und ihre Ende-zu-Ende/Port-zu-Port Protokolle

Aufgabe der Transportschicht ist es mit Hilfe der Netzwerkschicht (Internetschicht) dem Host-zu-Host Paket Übermittlungsdienst, einen Prozess-zu-Prozess Kommunikationsdienst aufzubauen.

Der Kommunikations Endpunkt ist hier ein Prozess. Ende-zu-Ende Protokolle übernehmen diese Aufgabe.

Was erwartet der Programmierer von der Transportschicht?

- Garantierte Übertragung der Daten/Nachrichten
- Übertragung in der richtigen Reihenfolge
- Übertragung von mindestens einer Kopie der Daten, der Programmierer will nur eine sehen!
- Übertragung von beliebig großer Nachrichten
- Eventuell Synchronisation von Sender und Empfänger
- Unterstützung von mehreren Sendern und/oder Empfängern auf einem Host

Was leistet die Netzwerk Ebene nicht?

- Die Reihenfolge der Daten ist **nicht** gewährleistet
- Die Ankunft der Daten ist **nicht** gewährleistet
- Es können mehrere Kopien der Daten ankommen
- Sie können verzögert ankommen

Die Transportschicht überwindet die Einschränkungen der Netzwerk Schicht (IP)

Mögliche Transportdienste

- Ein asynchroner einfacher Demultiplex Dienst → UDP
- Ein zuverlässiger Byte-Strom-Dienst → TCP
- Ein Frage-Antwort Dienst → RPC Protokoll SunRPC DEC RPC

UDP – Der Verbindungslose Transportdienst

- RFC 768
- UDP erweitert den Host-to-Host Datagramm Dienst von IP auf einen Prozess-to-Prozess Datagramm Dienst
- UDP sorgt dafür, dass mehrere Prozesse die auf einem Host laufen, quasi parallel mit anderen Prozessen auf anderen Hosts über Netz kommunizieren können.
- Man sagt UDP Multiplext/Demultiplext die Prozesskommunikation eines Hosts über Netzwerk
- UDP gewährt Prozessen direkten Zugriff auf einen Datagramm Dienst mit minimalem Verwaltungsaufwand
- UDP ist unzuverlässig, es gibt keine Mechanismen zur Fehlerkontrolle, und verbindungslos → kein Verbindungsauf- Abbau, UDP Datagramme laufen unabhängig von einander durchs Netzwerk.
- UDP wird via IP transportiert
- UDP bringt nur Prozess-to-Prozess Kommunikation und UDP Multiplexing als Mehrwert mit.

Die Prozesse auf den Hosts werden über sogenannte Port Nummern identifiziert

Der Client Prozess sendet an den Server Prozess Daten.

Der Server Prozess besitzt im Internet normalerweise eine well known Port Nummer (IANA)

Jeder Prozess im Internet ist eindeutig durch das Tripel (Socket):

IP Adresse + Protokoll + Port Nummer

Adressierbar!

Die Begriffe Port und Socket

Der Port (Ziel, Hafen, Tor) eine 16 Bit Adresse zur Identifizierung eines eindeutigen Zugangspunktes den das TCP bzw. UDP Protokoll benötigt um mit einer übergeordneten Anwendung zu kommunizieren.

Well known ports: 1-1023

Server Standard

Ports der IANA: 80 → http
23 → telnet
25 → smtp

dynamically allocated ports – client ports 1024 – 5000 ... Solaris > 32 768

reservierte ports (UNIX/Linux) 1-1023 dürfen nur von root Prozessen belegt werden.
Ports 256-1023 – Unix Dienste (z.B. rlogin, rsh)

Socket → (PortNr. + Protokoll + IP Adresse)

UDP ist am häufigsten von Fragmentierung betroffen, denn ein UDP Datagramm wird in ein IP Datagramm verpackt.

Eine Anwendungs-Programm-Schreiboperation erzeugt ein UDP Datagramm, die Größe des UDP Datagramm hängt von der Anwendung ab.

Bsp: DNS verschickt 512 Byte Nachrichten via UDP
NFSv3 verschickt 8192 Byte Nachrichten via UDP

$$\begin{array}{rcll} \text{UDP Maxgröße:} & IP_{Max} & -IP_{Header} & -UDP_{Header} \\ & 65535 & -20 & -8 & = 65507 \text{ Bytes} \end{array}$$

UDP Datagramm

Port Nr. Sender	→ 16 Bit
Port Nr. Empfänger	→ 16 Bit
UDP Länge	→ Größe des UDP Datagramm, Anzahl Bytes
UDP Prüfsumme	→ UDP Header (ohne Prüfsumme)
	+ UDP Body
	+ IP – Quell – Adresse \
	+ IP – Ziel Adresse - Endpunkt
	+ IP – Protokoll Typ /

Warum wählen Anwendungsprogramme UDP als Transportdienst?

- Anwendungen benutzen Multicast oder Broadcast
(Ansprache mehrerer Kommunikationspartner, z.B. Video-Konferenz)
- „höhere“ Geschwindigkeit durch weniger Protokoll Overhead
- Anwendungen mit Frage – Antwort – Schema
Antwort == positive Bestätigung der Frage, kommt keine Antwort, so stellt man die Frage wieder.
- Es werden nur Nachrichten (d.h. geringe Datenmengen) ausgetauscht.
- Die Anwendung regelt die Zuverlässigkeit selbst (NFS)

Transmission Control Protocol – TCP – als zuverlässiger Bytestream

TCP befreit die Anwendung davon sich um fehlende oder ungeordnete Daten kümmern zu müssen.

RFC 793, 761, 675, ...

- TCP bietet einen zuverlässigen, geordneten, verbindungsorientierten Bytestromdienst.
- TCP ist ein vollduplex Protokoll, jede TCP Verbindung hat 2 Byteströme, einen in jede Richtung.
- TCP hat ein Fluss-Kontroll-Mechanismus für jeden der Datenströme. Flusskontrolle verhindert, dass ein Sender den Empfänger mit Daten überflutet. Der Empfänger kann den Sender bremsen.
- TCP implementiert einen Mechanismus zur Überlastkontrolle, des Netzwerkes. Dieser verhindert dass zu viele Daten in das Netzwerk eingespeist werden.
- TCP unterstützt wie UDP einen Multiplex / Demultiplex Mechanismus, d.h. es können mehrere TCP Anwendungen gleichzeitig auf einem Rechner laufen.
- Broadcasting / Multicasting funktioniert **nicht** denn via TCP kommunizieren **immer** 2 Kommunikationspunkte, Anwendungs Programme

Wie produziert TCP Zuverlässigkeit?

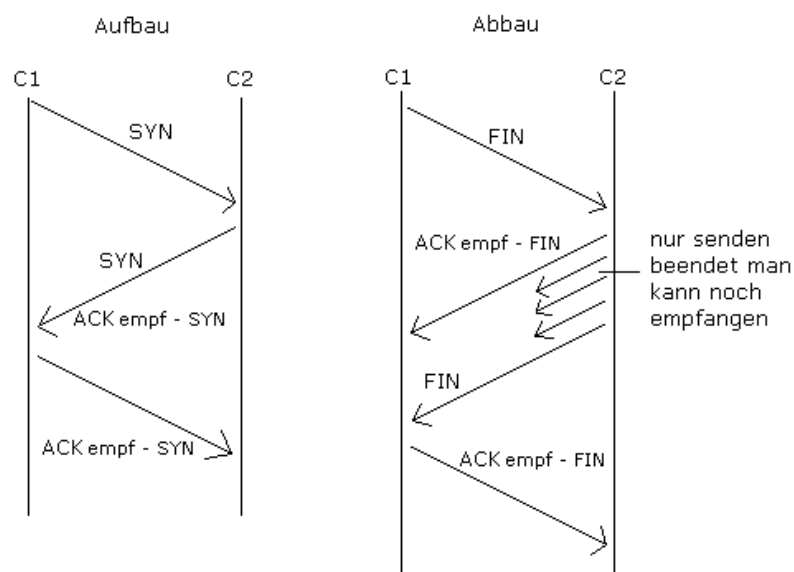
- Die Anwendungsdaten werden in Abschnitte aufgeteilt, als TCP Segmente bezeichnet und via IP Datagramme verschickt.
- Wenn die TCP Software ein Segment verschickt, wird ein Timer gestartet, trifft eine Empfangsbestätigung nicht vor Ablauf des Timers ein, so wird das Segment wieder verschickt.
- Empfängt die TCP Software Daten (Segmente), so sendet sie (leicht verzögert) eine Bestätigung (Acknowledgement)
- TCP errechnet eine Prüfsumme für Header&Daten. Geht ein Segment mit ungültiger Prüfsumme ein, wird es verworfen und **nicht** bestätigt.
- TCP Segmente werden via IP Datagramm versendet. → Die IP Datagramm Reihenfolge kann durcheinander geraten. → TCP löst das mit einer Sequenznummernfolge in den Segmenten
- TCP verwirft doppelte Segmente (gleiche Sequenz Reihenfolge Nummer)
- TCP regelt die Flusssteuerung über die Sende und Empfangspuffer.

TCP Header siehe Kopie!

Grundsätzliches zum Verbindungs Auf-/Abbau

- Verbindung einleiten → aktives öffnen der Verbindung (Client)
- Verbindung annehmen → passives annehmen der Verbindung (Server)
- Verbindungsaufbau ist Asymmetrisch, besteht aus 3 TCP Segmenten
- Verbindungsabbau ist Symmetrisch, besteht aus 4 TCP Segmenten
(Jede Seite kann ihre Verbindung (schreibende-output seite) beenden)

TCP Verbindungs Auf- / Abbau



Ethernet -> MTU = 1500 – 20 Byte IP-Header – 20 Byte TCP-Header = 1460 Byte

Wie erreicht TCP Zuverlässigkeit

→ Neuübertragung (adaptive) nach ablauf eines Timers

Wenn TCP ein Segment verschickt wird ein Timer gestartet, trifft eine Bestätigung nicht rechtzeitig, d.h. vor Ablauf des Timers ein, so wird das Segment wieder verschickt.

Wann kommt keine Bestätigung?

1. TCP Segment ging verloren
2. TCP Segment war fehlerhaft
3. Die Bestätigung ging verloren

Adaptive Neuübertragung – Berechnung der Timer Laufzeit

Geschätzte Neue RTT

$$\frac{0,8}{0,9} \cdot \text{Geschätzte alte RTT} + \frac{(1 - 0,8) \cdot \text{Gemessene RTT}}{0,9}$$

Time Out $\rightarrow 2 \cdot \text{Geschätzte RTT}$

TCP Datenflusskontrolle

Beim Verbindungsaufbau wird jedes Verbindungsende dem anderen eine WindowSize für den Datenempfang mitteilen. WindowSize = Datenempfangspufferspeicher

Bei jeder Bestätigung wird dem Sender die aktuell verfügbare, d.h. freie Windowgröße mitgeteilt.

Kann der Empfängerprozess, die Daten nicht so schnell verarbeiten, wie sie ankommen, so wird der Empfängerpuffer irgendwann gefüllt sein und dem Sender wird das durch ein ZERO-WINDOW mitgeteilt

Sliding Window (technisch) \rightarrow gewährleistet: (siehe Kopie)

- Die zuverlässige und geordnete Übertragung von Daten

Sendepuffer enthält:

- bereits gesendete Daten, deren Empfang noch nicht bestätigt ist.
- Daten die vom Anwendungsprogramm in den Sende Puffer geschrieben wurden aber noch nicht versendet wurden

3 Sendepuffer Zeiger

- LastByteAcked \rightarrow das zuletzt bestätigte Byte
- LastByteSent \rightarrow das zuletzt gesendete Byte
- LastByteWritten \rightarrow das zuletzt von der Anwendung in den Puffer geschriebene Byte

Empfangspuffer enthält:

- Außer der Reihe ankommende Bytes
- Daten die in der richtigen Reihenfolge angekommen sind, aber von der Anwendung noch nicht gelesen wurden

3 Empfangspuffer Zeiger:

- LastByteRcvd \rightarrow das zuletzt empfangene Byte
- LastByteRead \rightarrow das zuletzt von der Anwendung gelesene Byte
- NextByteExpected \rightarrow das aktuell erwartete Empfangs Byte

- Die Übertragung in der richtigen Reihenfolge (von Anwendung zu Anwendung)
- Datenflusskontrolle zwischen Sender und Empfänger

Wann sendet TCP Daten?

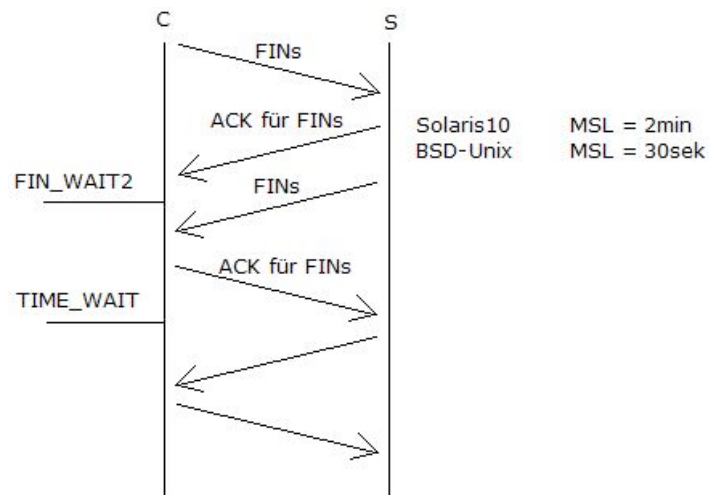
1. Wenn im Sendepuffer genügend Daten für ein volles TCP Segment vorliegen (MSS)
2. Wenn der sendende Prozess es ausdrücklich fordert (TCP-Push-Option)
3. Wenn Bestätigungs Timer abgeläuft
4. Wenn ein ZeroWindow empfangen wurde, sendet die TCP Sender Seite zyklisch 1 Byte Segmente um zu erfahren ob wieder Empfängerspeicher frei ist.

Das TCP Zustandsübergangsdiagramm (siehe Kopie)

Server startet	CLOSED	→	LISTEN
			Passives Öffnen
Client startet	CLOSED	→	SYN-Segment verschicken
			Aktiv

Die Zustandsübergänge bei einer Client Software (siehe Kopie)

MSL – Maximum Segment Lifetime



TCP / IP Software tuning unter SOLARIS

Welche tuning Parameter bietet Solaris TCP / IP ?

ndd /dev/ip ?	netstat -s -P ip
tcp ?	tcp
udp ?	upd

Übersicht verschaffen	docs.sun.com
-----------------------	--

DOS Angriffe

Denial of Service

- Flooding Angriffe
- Schwächen der Protokolle ausnutzen (IP, TCP)
- Fehlerhaften Programmcode ausnutzen

Ping of Death

- ICMP Paket (zu groß für IP)
- BlueScreen / System Absturz

Teardrop

- verschickt fehlerhaft fragmentierte IP Pakete
- System Absturz

Land Attack

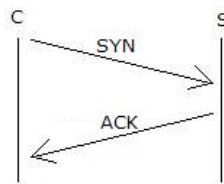
- IP Paket mit identischer IP Sendende und Empfangsadresse
- 100% CPU Last

Smurf Attack

- 1 PC benutzt viele PCs als Helfer
- `ping -s 134.96.216.255 50000 10` verursacht z.B. sehr viel Netzlast
- der Angreifer PC versendet sehr große ICMP Pakete an die Broadcast Adressen vieler Netzwerke. Dabei wird die Quell IP Adresse so manipuliert das sie der anzugreifenden IP Adresse entspricht.

SYN Storm

- verschicke tausende SYN Segmente möglichst schnell zum Ziel Host aber antwortet nicht auf die Acknowledgements. So hat der Ziel Host tausende halb offene Verbindungen eröffnet



Abhilfe:

- Erlaubte halboffene Verbindungen auf 100-200 setzen
- Wartezeit bis schließen einer halboffenen Verbindung 60-300sek

IV Netzanwendungen im UNIX Umfeld

Was leistet TCP / IP ?

Ein TCP / IP Internet bietet **nur** eine allgemeine Kommunikations Infrastruktur (genau wie das Telefonsystem)

Welche Dienste im Netz erreichbar sind, wer und wann sie von wem benutzt werden weiß das Netz nicht.

Um einen Dienst zu benutzen benötigt man dessen Adresse und entsprechende Software.

Man benötigt mindestens 2 Kommunikationsendpunkte.
TCP genau 2, sonst UDP.

Bsp: Stand alone Anwendung: cp datei1 datei2

cp ruft geschickt UNIX Kernel Funktionen, System Calls auf, um die Arbeit zu erledigen.

Client-Server Programm: rcp datei1 datei2 arbeitet mit rcpd zusammen
→ hier werden TCP/IP System Calls des UNIX Kernels benutzt

Einfache UNIX Standard Dienste

echo	<ul style="list-style-type: none">- RFC 862 / TCP-UDP Port 7- der Server sendet erhaltene Zeichenketten an den Client zurück
discard	<ul style="list-style-type: none">- RFC 863 / TCP-UDP Port 9- der Server wirft alle ankommenden Zeichen komentarlos weg
chargen	<ul style="list-style-type: none">- RFC 864 / TCP-UDP Port 19- TCP Server sendet eine ununterbrochene Folge von Zeichen, bis der Client die Verbindung abbricht- UDP Server sendet als Antwort auf ein Client UDP Paket ein UDP Paket beliebiger Größe gefüllt mit Zeichen
daytime	<ul style="list-style-type: none">- RFC 867 / TCP-UDP Port 13- Server gibt eine Zeichenkette mit Datum und Uhrzeit zurück
time	<ul style="list-style-type: none">- RFC 868 / TCP-UDP Port 37- Server liefert eine 32-Bit Zahl, die die Anzahl der verflossenen Sekunden, seit dem 01.01.1990 00:00 Uhr darstellt
telnet	<ul style="list-style-type: none">- universeller Client- z.B.: telnet host echo

Das Client Server Paradigma

Merkmale der Client Software

- Wird vom Benutzer aktiviert/deaktiviert
- Läuft meist auf dem Rechner des Benutzers
- Leitet die Kommunikation aktiv ein
- Benutzt einen temporär zugewiesenen Protokoll Port
- Benötigt keine spezielle Hardware, kein spezielles Betriebssystem
- Kann bei Bedarf eventuell mehrere Dienste ansprechen (http, https, file, wais, mail, ...)

Merkmale der Server Software

- Wird automatisch bei Systemstart gestartet
oder
bei Bedarf von einem SuperDaemon gestartet
- Läuft meist auf einem gemeinsam benutzten Rechnersystem (Server)
- Wartet passiv auf Verbindungswünsche von entfernten Clients
- Benutzt meist einen well known Port
- Benötigt eine gute Hardware, ein sicheres, stabiles Betriebssystem
- Bietet einen Dienst an

Client – Server Kommunikationsformen

1. Client stellt Anfrage (aktiv)
Server antwortet (passiv)
Bsp.: NFS, http 1.0
Transportprotokolle: TCP, UDP
2. Client stellt mehrere Anfragen im gleichen Kontext
Server liefert mehrere Antworten
Bsp.: rlogin, telnet, ssh, Datenbank Sitzungen, http 1.1
Transportprotokoll: TCP
3. Server bietet laufend Daten an
Client nimmt Verbindung auf
Bsp.: Videokonferenz Software, Radio Software
Transportprotokoll: UDP

Reale Server Beispiele STL

134.96.216.233 Stand 2001
stl-s-ad - NIS+ Master Server
stl-pop - POP
stl-mail - SMTP
www1 - http
stl-www - https

134.96.216.210 - stl-s-sie Stand 2006
stl-s-samba - SAMBA (virtueller Server)
stl-s-nfs - NFS (virtueller Server)
stl-s-nisp - NIS+ Secondary Server (virtueller Server)
stl-s-studwork - ssh (virtueller Server)

Protokolle, Ports und Sockets

Problem 1:

Auf stl-s-ad laufen folgende Server:

- NIS+
- POP
- SMTP
- http
- https
- alle r-Utilities
- DHCP

Frage:

- a. Wie verteilt IP die ankommenden Daten an die einzelnen Server?
- b. Woran erkennt IP welches Transportprotokoll benutzt wird?

Antwort

- A. IP kennt KEINE Server !
- B. IP kennt die empfangende Transport Software (TCP/UDP)
IP Header Feld: Protokoll-Typ

Problem 2:

Frage:

Wie identifiziert die Transport Software TCP/UDP den anzusprechenden Server?

Antwort:

In TCP/UDP Header existiert eine Port Nr, die den Serverprozess identifiziert!
Die Server Software hat einen einprogrammierten Port der beim Start der Transport Software mitgeteilt wird.

Problem 3:

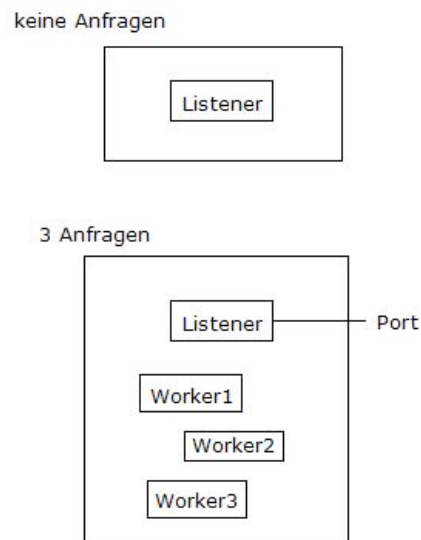
- Mehrere Clients kontaktieren gleichzeitig einen Server, denke an http Server, telnet Server

Frage:

Wie kann ein Server quasi parallel, Anfragen mehrerer Clients bearbeiten?

Antwort:

Eine Server Software besteht aus 2 Komponenten, einem Listener der auf Anfragen wartet und einem Worker, der die Anfragen bedient. Wobei sowohl Listener als auch Worker mehr als einmal laufen können.



Problem 4:

Folgt aus Lösung von Problem 3

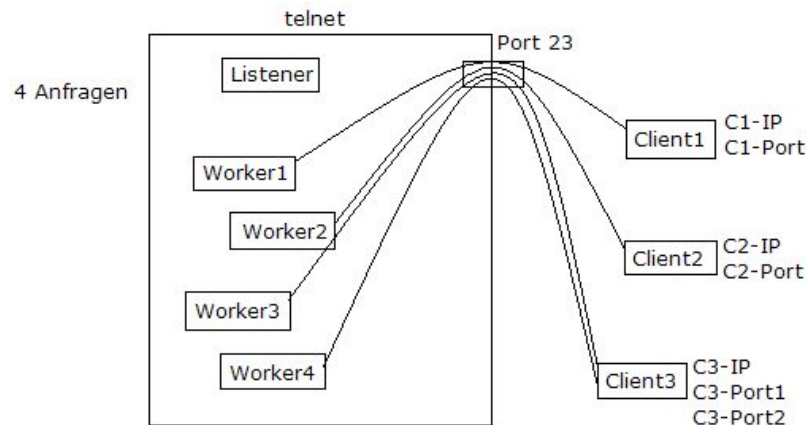
Es existieren parallele Worker Prozesse/Threads

Frage:

Wie werden die Daten auf die verschiedenen Prozesse verteilt?

Antwort:

Socket Paare identifizieren den Client-Server Kommunikations Kanal!



Server	Client
Server IP, TCP, 23	(Client 1 IP, TCP, Client 1-Port 1)
	(Client 2 IP, TCP, Client 2-Port 1)
	(Client 3 IP, TCP, Client 3-Port 1)
	(Client 3 IP, TCP, Client 3-Port 2)

netstat zum finden von aktiven Sockets

```
netstat    -a [n]  [-f inet]
           -s      [-f inet]
           -i      [-f inet]
           -i [a]  [-f inet]
```

Dämonen (Daemon) und ihr Meister

Welche Server (=Daemons) laufen standardmäßig auf einer Solaris Maschine?

```
/usr/ucb/ps -axc | grep „^.*d$“
```

Es fehlen die UNIX r-Utility Server! Alle UNIX r-Utilities sind Worker Server, sie werden bei Bedarf durch den Super Daemon „inetd“ gestartet!

Unter UNIX werden viele Dienste nur bei Bedarf gestartet.

Der SuperDaemon **inetd** – Internet Daemon

Der **inetd** muss alle Ports die er betreuen soll offen halten um ankommende Verbindungswünsche weiterzuleiten.

Solaris < 10 und Linux → /etc/xinet.conf -> /etc/xinetd.d oder /etc/rc.d

Solaris > 10 → SVC

Welche Sockets benutzt inetd

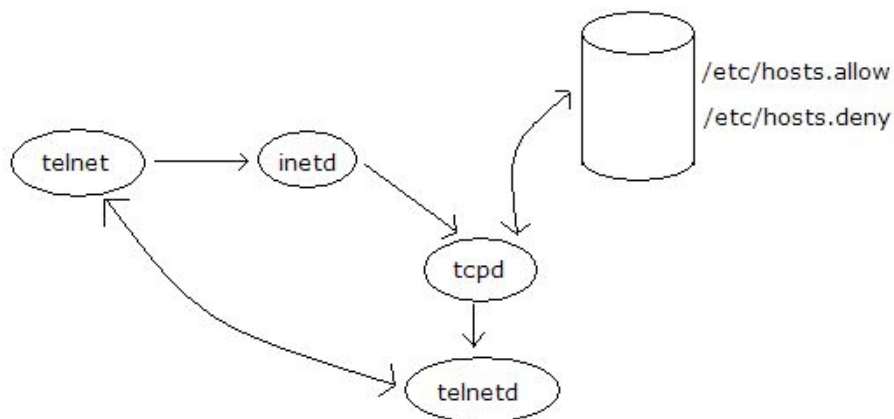
1. netstat -a -f inet -LISTEN
2. pgrep inetd
3. als root: pfiles inetd_PID > /tmp/xx
4. vi /tmp/xx → suchen AF_INET

top → CPU Last

iftop → Netzwerk Last

Exkurs: tcpwrapper

tcpd → prüft die Remote IP Adresse auf erlaubt/nicht erlaubt ab und entscheidet ob Dienst gestartet wird oder nicht.



Die Socket Programmier Schnittstelle

API – Application Programming Interface ermöglicht das Benutzen von tiefer liegender Software.

SOCKET-API → auf die Dienste der TCP/IP Software zuzugreifen

UNIX/Linux → Systemaufrufe

Win XP → Socket Bibliotheksaufrufe

Unix Ein-/Ausgabe- Konzept

Wie greifen Unix Programme auf Dateien oder Geräte zu?

1. open - öffnet die Datei/das Gerät zum lesen/schreiben,
liefert einen Datei DESCRIPTOR
2. read - Daten via Datei lesen
write - Daten via Datei schreiben
3. close - schließt die Datei/das Gerät

Dateien sind Byte Streams!!!

UNIX Netz Ein-/Ausgabe Konzept

1. socket - öffnet einen Socket (Kommunikationskanal)
liefert einen Socket Descriptor Handle
2. recv - Daten via Handle empfangen
send - Daten via Handle senden
3. close - Handle schließen

RPC – Remote Procedure Call

Socket Programmierung ist eigentlich merkwürdig, da nur String hin und her geschickt werden.

Normal beim Programmieren sind Funktionen, bzw Prozeduren Aufrufe und Daten wie: double, int, float, ...

Werkzeuge zur RPC Programmierung

Schnittstellen Beschreibungs Sprache: **RPCL**

Remote Procedure Call Language

Damit beschreibt man die RPC Funktionen mit ihren Parametern und Rückgabewerten. Die in RPC beschriebene Schnittstelle wird mit dem RPCL Compiler **rpcgen** übersetzt, es entstehen die Quell Programme für
den Client Stub
den Server Stub
und die XDR Funktionen

SUN-RPC

RPCL → rpcgen → „call by value“ – einfache Datenstrukturen
IDL → Interface Definition Language
 + Call by Reference
 - komplexe Datenstrukturen
XDR → External Data Representation



Die RPC Paketstruktur

RPC setzt auf der Socket API auf, ist kein eigentliches Protokoll, sondern ist eine API. Trotzdem hat eine RPC Nachricht eine gewisse Datenstruktur

Transaktions ID - identifiziert die Art der Client Server Interaktion
 RPC-Aufforderung = 0
 RPC-Antwort = 1

RPC Versions Nr - (2)

Programm- identifizieren eindeutig
Versions- die aufzurufenden
Prozedur- remote Prozedur
Nr

Identifikator des Clients

Prozedurparameter in XDR Format

Problem: Kontaktaufnahme, wie wissen Clients Server IP, Protokoll, Port?

Lösung:

Server auf RPC Basis melden sich beim Startup bei einem Service Server (rpcbind) [Port Mapper] an. Sie übergeben die Programm-, Versions, und alle Prozedurnummern die sie haben.

Der Portmapper weist ihnen einen Port zu

Frage: Welche RPC Programme sind auf einen Host aktiv?

```
rpcinfo -p  
-s  
-T tcp stl-s-ad 100005
```

Client / Server mittels RPC

Realisiert ein remote ls

```
rsh host kommando  
rsh host ls -a dir == rls host dir
```

RPC Programm Nummern

0x0000 0000-1FFF FFFF	von Sun vergeben
0x2000 0000-3FFF FFFF	für Benutzer
0x4000 0000-5FFF FFFF	“
0x6000 0000-FFFF FFFF	“

Internet Dienste

DNS

eMail – SMTP / POP / MIME

www – http

DNS – Domain Name System

IP funktioniert nun mit IP Adressen

Menschen merken sich Namen besser wie Zahlen

DNS übersetzt Rechner Namen → IP Adressen bzw. IP Adressen → Rechner Namen

→ in den entsprechenden Anwendungs Programmen wird die Übersetzung mit der C-Funktion: `gethostbyname()` aufgerufen.
`gethostbyaddr()`

3 Mögliche Datenquellen:

Rechner lokal → `/etc/hosts`

Netz lokal → name Service NIS / NIS+

Global → DNS

Steuerdatei: UNIX / LINUX → `/etc/nsswitch.conf`
`/etc/resolv.conf`

Am Anfang: `/etc/hosts`

NIC – Network Information Center USA verwaltet alle Server des Internets

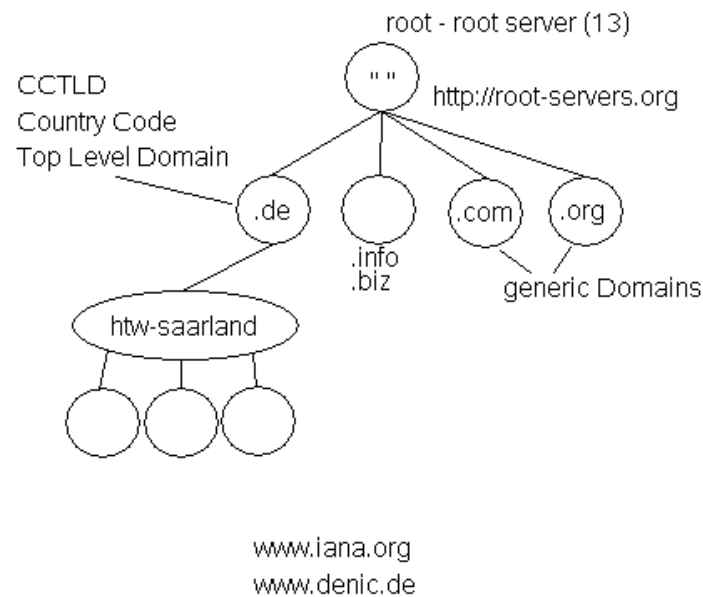
Die `/etc/hosts` musste man sich ab und zu herunterladen → UNIX: `gettable`, `htable`

Heute NICs in allen Ländern, Deutschland → www.deNIC.de

DNS → RFC 1034/1035

Beschreiben eine Client-Server Anwendung mit einer im Netz verteilten Datenbank unter UNIX sind die Datenbanktabellen in Textdateien abgelegt.

DNS Namenshierarchie



Die Iana verlangt immer 2 Nameserver, einen innerhalb und einen außerhalb der Organisation

Das DNS Client Server Modell

→ Eines der wichtigsten Merkmale des DNS ist die **Autonomie** seiner Teilbereiche (Domains)

→ Das System ist so ausgelegt, dass jede Organisation, die eine DNS Domain besitzt, unterhalb dieser Domain beliebige Sub Domains mit beliebigen Namen einrichten kann, **ohne** einer übergeordneten Instanz Informationen darüber zukommen zu lassen.

Bsp.: Domain: uni-sb.de

HTW → htw.uni-sb.de

Die Server Hierarchie entspricht im allgemeinen der Domain Hierarchie / Namens-Hierarchie.

Jeder Server ist für seine Domain zuständig, er kennt seine Sub Domain Server und normalerweise einen Übergeordneten Domain Server und/oder root Server.

Ein Root Server kennt alle Top Level Domain Server. Kommt eine Anfrage zu einer Second Level Domain, so wird der root Server dem anfragenden die Adresse des zuständigen Top Level Domain Servers mitteilen.

→ rekursives Abfragen der evt. näher am Ziel liegenden Name Server (saarländisches Prinzip)

Ineffizient:

Lösung1: root Server Replikation (13)

Lösung2: Caching

DNS Server Typen

Primary Server: ns.htw-saarland.de

- lädt seine Daten aus Daten Dateien von seiner Festplatte
- der Sysadmin managt die Dateninhalte
- wird autoritativer Server genannt, er gibt immer richtige Antworten

Secondary Server: ns1.htw-saarland.de

- laden von Zeit zu Zeit die Daten vom Primary Server, speichern sie in lokalen Dateien
- meist richtige Antworten (99,9%)

Caching Only

Besitzt keine eigenen Datendateien, lernt über Client-Anfragen derer Ergebnis er cacht

Das chaching Problem:

Hacker „vergiften“ den DNS Cache eines DNS Servers durch einschleusen von falschen DNS Antworten

Elemente des DNS unter UNIX / Linux

Client Software:

Besteht aus den Resolv Funktionen gethostbyname(), gethostbyaddr(), die von den Client Programmen bei Bedarf aufgerufen werden.

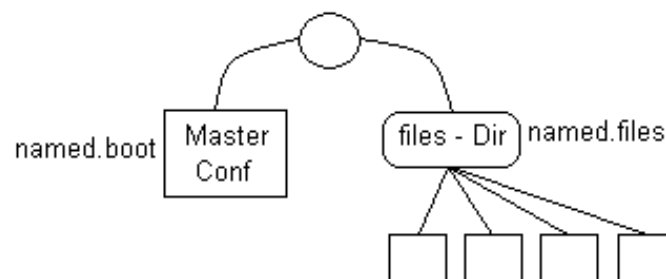
Die Resolv Funktionen benutzen die Datei /etc/nsswitch.conf um die Datenquellen Hierarchie zu erfragen und damit die Namensauflösung zu beginnen:

/etc/hosts
nis
/etc/resolv.conf

Server Software:

BIND – Berkely Internet Name Domain → named (Name Daemon)

Der named hat einen Satz von Konfigurationsdateien



DNS caching → Sicherheitsproblem, DNS poisoning

Resolver → /etc/nsswitch.conf
→ /etc/resolv.conf

Das DNS Nachrichtenformat

→ Software → bind() → Server → gethostbyname() → Client

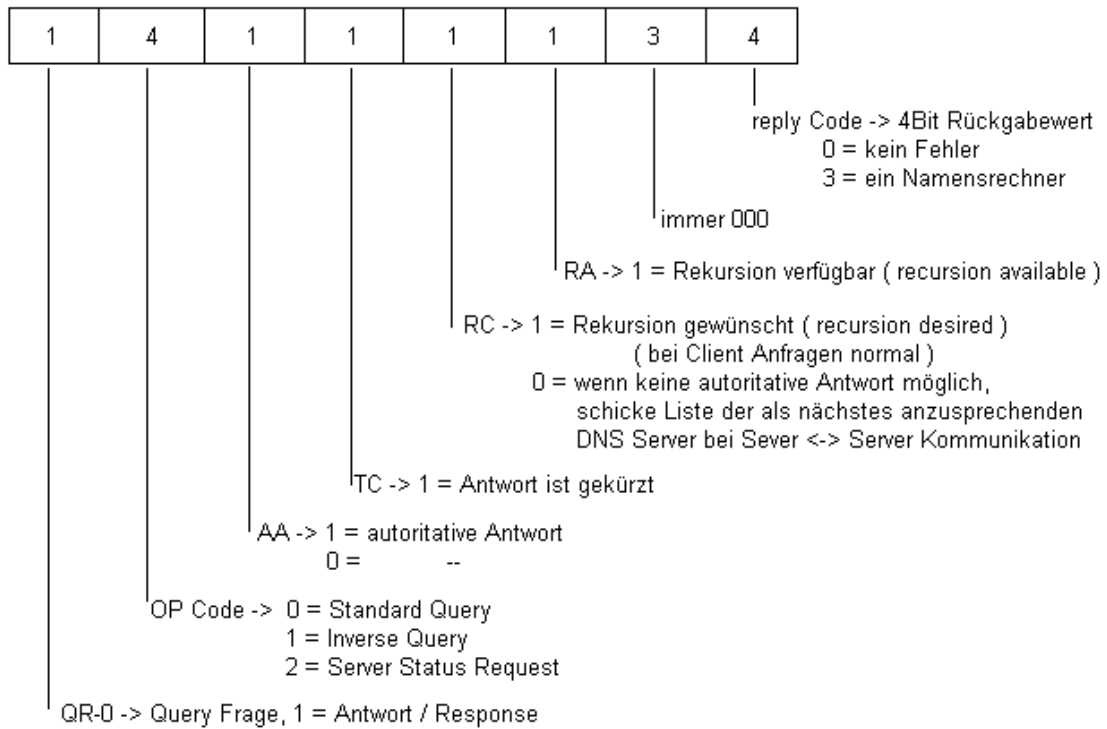
→ Port: UDP 53 (wird benutzt für „normale“ Anfragen)
—
TCP 53

Wird benutzt für „Nachfragen“, wenn die UDP Antwort gekürzt wurde! – DNS Antwort wird in **einem** UDP Datagramm versendet, mit einer MAX MAC Größe von 512 Byte.

Wird benutzt zum Zonentransfer – Laden der Daten vom primary zum secondary Server

Identifikation	Flags	12 Byte
Anzahl Fragen	Anzahl Antworten	
Anzahl auto Antworten	Anzahl Zusatz Infos	Resource Record
Frage		Kann Mehrmals Auftauchen Bel. Länge
Antwort		
Autorität		
Zusatz Infos		

Identifikation → wird vom Client gesetzt, vom Server zurückgegeben, hilft dem Client die Antwort mit der Frage zu verbinden



Informationen über Domain Eigentümer suchen & finden

whois → RFC 954

www.denic.de - whois

V.2 Elektronische Post – eMail

- eMail = „kurznachrichten Dienst“
- anfänglich nur 7 Bit ASCII Daten versendbar
- später – MIME Protokoll (Multipurpose Internet Mail Extensions)
um auch Binärdaten – Grafiken, ausführbare Programme, Word-/Exel Dateien,
... versenden zu können.
- sind auch Grundlage für Mailinglisten, News Foren

Wieso wird eMail so viel genutzt?

- Einfach benutzbar, bequem – es entfällt das Kuvertieren und zur Post bringen
- schnell
- asynchron
- nicht Ortsgebunden
- speicherbar, bearbeitbar
- billig

Der Aufbau des eMail Systems

User Agent (UA) – eMail Programm des Benutzers (UNIX: mail)

Message Transfer Agent (MTA) – smtp Mail Server

UNIX: sendmail

Linux: postfix

Versendet eMails auf Wunsch eines UA im Internet

POP3-Server -

IMAP Server – incoming Mail Server

Aufgaben des:

UA → Unterstützt den Benutzer beim lesen, schreiben, archivieren, suchen seiner eMails

MTA → wartet bis Nachrichten in seiner Warteschlange stehen, die er versenden soll.
An lokalen User: Server hängt die Nachricht an die Mail Box, die Datei des Benutzers unter /var/mail an.
Bsp.: /var/mail/pauly (mail: drwxrwxrwt) (pauly: -rw-rw----)

Ist die Nachricht für einen entfernten Benutzer, so wird der lokale SMTP Server zu einem Client, sucht via DNS den Ziel- oder Zwischen Mail Server für den Empfänger und sendet die Nachricht weiter.

Aufbau einer eMail Adresse

Name @ Ziel - Domain Name → generische eMail Adresse
Bsp.: wolfgang.pauly@htw-saarland.de

- Rechnername
Bsp.: pauly@stl-s-stud.htw-saarland.de

Weitere Standardisierte Form: Wolfgang Pauly pauly@htw-saarland.de

eMail versenden an andere eMail Systeme

AOL / t-online

name@aol.com
name@t-online.de

CompuServe

CompuServeID@compuserve.com

Mobildfunk Netzwerk – max 170 Zeichen

<TelefonNr>@sms.netcs.net

49172____@sms.netcs.net (D2)

Aufbau einer eMail [das eMail Datenpaket]

Eine eMail besteht, wie jedes andere Datenpaket im Internet aus 2 Teilen:

eMail Header: enthält Transportinformationen wie

- Sender
- Empfänger
- Verarbeitungsempfehlungen
- Beschreibung des Inhalts

eMail Body: enthält die eigentliche eMail Nachricht in 7-Bit ASCII Format

Der Header hat ein Standardformat für seine Zeilen, es wird weder die Zeilenanzahl noch die Art aller Zeilen beschrieben.

Genial: Kann ein UA und/oder MTA eine Headerzeile **nicht** interpretieren d.h. verarbeiten, so gibt er sie unverändert weiter !!!

Zeilenformat:

Schlüsselwort: Informationen zum Schlüsselwort

Bsp.:

FROM: Wolfgang.Pauly@htw-saarland.de

Einige Headerzeilen sind vorgeschrieben:

z.B.: FROM, TO

einige sind wahlweise auszufüllen:

z.B.: Subject, CC, BCC

einige werden automatisch erzeugt:

z.B.: Date

Wie kann man WinWord Dokumente, pdf oder ps Dateien oder Bilder, ausführbare Programme via eMail versenden?

- Alle diese Dateien benutzen 8 Bit Codierung, **aber** das eMail System benutzt 7 Bit Codierung
- Lösung: MIME
die IETF hat, da viele verschiedene Lösungen für das 8 Bit \leftrightarrow 7 Bit Problem erarbeitet wurden eine vereinheitlichung vorgeschlagen \rightarrow MIME

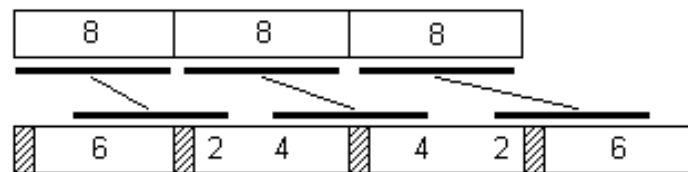
MIME ermöglicht einen Austausch von Binärdaten via eMail (RFC 1521/1522/1524)

Multipurpose Internet Mail Exchange diktiert **keinen** Standard der Codierung / Decodierung sondern eine einheitliche Kennzeichnung.

MIME benutzt dazu 5 zusätzliche Headerzeilen

Die MIME Headerfelder

- MIME Version 1.0
- Content Type: Typ/Subtyp; Seperator
[www.iana.org](http://www.iana.org/assignments/media-types) / assignments / media-types
RFC 1521 / 2045 / 2046
- Content Length: Länge des Abschnittes in Bytes
- Content-Transfer-Encoding: Verschlüsselungsart
Aus 8 Bit mussten nun 7 Bit gemacht werden
base64



- Content-ID: nähere Beschreibung
Content-Description: der übertragenden Daten

MIME wird auch im WWW als Seiteninhaltsbeschreibung benutzt.

SMTP – Simple Mail Transport Protocol

TCP Port 25, RFC 821/822

Kommunikation:

ASCII basiert, dem menschlichen Dialog nachempfunden

1. → Beim Server anmelden HELO
 ← 250 <pleased to meet you ...>
2. → Bezeichnen des Senders MAIL FROM
 ← 250 <OK>
3. → Bezeichnen des/der Empfänger RCPT TO
 ← 220 <OK> / 550 <ERROR>
4. → DATA
 ← 354 Enter mail
 Daten
5. Verabschieden vom Server QUIT

IMAP – Interactive Mail Access Protocol

RFC 1064

Wurde entwickelt um mehrere Client Rechner zu unterstützen, zwischen denen der User wandert.

Die Mail Folder des Users werden auf dem IMAP Server verwaltet.

POP3 – Post Office Protocol version 3

TCP / 110 – POP

TCP / 915 – POPoSSL

Vorgehensweise:

1. Abfrage von User-Kennung & Passwort
2. Anmelden am POP Server
3. Herunterladen aller anstehenden Mails auf die lokale Platte
4. Löschen der heruntergeladenen Mails auf dem POP Server
5. Beenden der Verbindung zum Server
6. Lesen und verarbeiten, der lokal gespeicherten Mails

POP ist wie das SMTP ein ASCII basiertes Protokoll, das einen „menschenähnlichen“ Dialog nachbildet.

eMail Weiterleitung unter UNIX/Linux

→ \$HOME/.forward

Zugriffsrechte: -rwxr-xr-x user group

Einfachster Inhalt: std-account@domain
Wolfgang_pauly@web.de

\pauly, | /usr/sbin/vacation

Mail wird in die Abwesenheitsmeldung
Mailbox von
Pauly
Weitergeleitet

pauly@t-online.de
/export/home_pw/pauly/Juli
| /usr/bin/myprog

SPAM und HAM

EU Definition: unverlangt zugestellte eMail, „Junk Mail“, „Balk Mail“

UBE unsolicited balk eMail

UCE unsolicited commercial eMail

SPAM → „Monty Python“ – Newsgroups

HAM → erwünschte eMail

Warum wird überhaupt gespamt?

Spam kostet den Spammer fast nichts!

Die Kosten tragen die Provider, der Empfänger (Speicherplatz, Downloadzeiten, -volumen, Zeit)

Die Rechtslage:

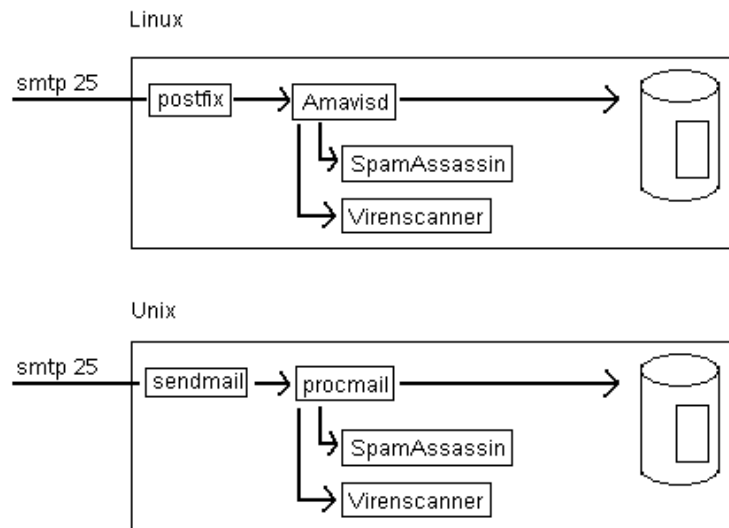
Das EU Parlament hat 12.07.2002 die Richtlinie RL 2002/58/EG verabschiedet

Richtlinie für den Schutz persönlicher Daten und der Privatsphäre auf dem Feld der elektronischen Kommunikation.

→ Opt-In-Lösung: E-Mail Werbung nur mit vorheriger Zustimmung des Empfängers.

Vorgehensmöglichkeiten gegen Spam:

- 1) Schütze deinen PC mit Firewall & Virens Scanner
- beides möglichst aktuell –
- 0) Der eigene Mail Server sollte kein Mail relaying erlauben.
Der Mail Server erlaubt nur Rechnern der lokalen Domain eMail zum Versenden zu übergeben.
Der Absender muss als User dem Mail Server bekannt sein.
- 1) Empfehlung für den User:
Benutze verschiedene eMail Accounts für die verschiedenen Arbeiten die auszuführen sind: Arbeits, Spiel, eBay - eMail Kennung
- 2) Einsetzen von Filter Programmen
 - 1. User: alle gängigen eMail Clientprogramme bieten eingebaute Filteroptionen
 - 2. Zentrale eMail Filterung
HTW → SpamAssassin <http://eu.spamassassin.org>
In Perl geschrieben, verursacht viel CPU Last



SpamAssassin → Black List, Liste der bekannten Spam Server
(~12 000)

<http://www.stearns.org/sa-blacklist/sa-blacklist.current>

+ Textuelle Filter → müssen HTML verstehen

Greylisting Verfahren zur SPAM Anwehr

Spammer Software nutzt das Prinzip „fire and forget“ sie sendet eine eMail genau ein mal.

Prinzip:

Speichere während des SMTP Verbindungsaufbaus folgende Merkmale:

- Die IP Adresse des Sender Hosts
- Die Envelope Adresse des Absenders
- Die Envelope Adresse des Empfängers

Beim ersten Auftreten des Tripels speichere die Daten in einer Datenbank und weise den Zustellungsversuch mit folgender Fehlermeldung ab.

Fehlermeldung: 450 you are greylisted – try again later

In den SMTP RFCs ist ein solches Server Verhalten erlaubt. Erfolgt innerhalb der nächsten 12 Stunden ein erneuter Zustellversuch, dann wird das Tripel für 36 Tage freigeschaltet. (es gibt eine Tripel Totzeit >= 10 Minuten, d.h. erst nach 10 Minuten darf wieder versucht werden die Mail zu zustellen)

Warum ist Spam möglich?

Brief → Umschlag + Briefbogen (Briefkopf + Text)

eMail → Envelope + Briefbogen (eMail Header + Body)

→ eMail Header frei bestimmbar !

siehe Deutsche Bank Spam Mail

Wird die Received Kette unterbrochen, erkennt man dass es eine Spam Mail ist!

Fallstricke für Spamfilter Betreiber

Strafgesetzbuch §206 Abs. 2 Nr. 2

verbietet Inhabern oder Beschäftigten eines Unternehmens, das geschäftsmäßig Post- und Telekommunikationsdienste erbringt, unbefugt eine dem Unternehmen zur Übermittlung anvertraute Sendung zu unterdrücken.

Freiheitsstrafe bis zu 5 Jahren

Sinn und Zweck: Schutz des Post-/Fernmeldegeheimnisses

Auch Firmen, die eine auch zeitweise, private Nutzung der Internet Dienste erlauben (eMail, Web) fallen unter diesen §. Sie sind dann TK-Provider für ihre Mitarbeiter. Man muss die private Nutzung ausdrücklich verbieten!

eMail Abweisungen via Blacklists sind rechtlich strittig.

Greylisting ist besser.

Lösung 1:

Klauseln im Vertrag mit dem User. (aol.com)

Lösung 2:

User Konfiguriert seine Mail Verarbeitung selbst. (web.de)

Lösung 3:

Spam Mail wird nur gekennzeichnet.

Strafgesetzbuch §303a

verbietet das Löschen und Unterdrücken von Daten.

Dieser § ist nicht auf TK-Dienstleister begrenzt, auch normale Firmen sind davon betroffen.

Auch das Löschen von Viren-Behafteten Mails fällt darunter! Man darf den Schad-Software-Teil entfernen!

V.3 www – World-Wide-Web

- 1989 – Cern – Tim Berners Lee
- März 1989 → „Link Technik“
- 1990 gab es den ersten Textbasierten Zeilenorientierten „Web Browser“
- Dezember 1991 Texas → im Rahmen Hypertext erste Demonstration
- Februar 1993 erster echter Browser (graphisches Tool): Mosaic
vorher: whois / gopher / veronica / ...
- Marc Andreessen – Netscape
- 1994 www Konsortium gegründet
- März 1994 HTW/STL 1. Web Server
Studenten PI 1. Studienjahr (5) + Pick & Pauly
experimenteller Web Server HTW
- 1996 im Rahmen einer Fallstudie PI Studenten verbessern Design
- April 1998 → Ranking (Spiegel) HTW beste FH Webseite
- Oktober 1998 → zweitbeste Webseite aller Hochschulen in Deutschland

Aufgaben die ein Internetdienst erfüllen muss:

- Adressierung und Verwaltung der Client – Server Kommunikation
- Anpassung der Daten und Kommando Formate
- Auf Basis des Anwendungs Protokolls Kommunizieren

Aufgabenerfüllung durch http

- Die Adressierung erfolgt auf Basis eines URI
(Uniform Resource Identifier) RFC 1030
bezogen auf http spricht man von einer URL (RFC 1738 / 1808)
- Die Anpassung und Bekanntgabe der Datenformate erfolgt durch eine „MIME Kommunikation“
d.h. die Client Software teilt der Server Software bei ihrer Anfrage mit,
welche Art von Daten sie darstellen kann.
(Accept: text/plain ... Accept: application/pdf ... UserAgent: Mozilla)
- Das Anwendungs Protokoll http ist einfachst gestrickt, es besteht aus einer
http Anforderung (http Request) vom Client zum Server und einer http
Antwort (http Response) vom Server zum Client.

URL → das universelle WWW Adressierungs Schema

Sie besteht aus 3 Teilen:

<Protokoll> : // <Server> [: PortNr] / Pfad

Protokoll:

ftp://	mailto://	file://
http://	gopher://	
https://	wais://	

[ftp://user@server/...](#)

user:passwd@server

Das http Protokoll

Ist ein einfaches, **zustandsloses** auf dem ASCII Zeichensatz basierendes Protokoll zur Übertragung von beliebigen Daten.

HTTP benutzt TCP.

Prinzipieller Kommunikationsablauf:

- TCP Verbindungsaufbau
- Übertragen der http Anfrage
- Übertragen der http Antwort
- [nachladen von Seitenelementen]
- TCP Verbindungsabbau

Vorteile der Zustandslosen http Kommunikation

- Die Server Software muss keine Sitzungen verwalten → einfacher Aufbau
- Server Software ist schnell
- Es ist nicht nötig viele Verbindungen parallel zu verwalten.
- Jede http Anforderung kann in einem Schritt beantwortet werden.

Nachteile

- Keine Sitzungen (Sessions) einfach möglich

Bsp.: telnet

- Anmeldung
- Austausch von Daten und Kommandos
- Abmeldung

Lösung

- Versteckte html Elemente
z.B. Username und Passwort werden als versteckte Elemente bei jedem Client/Server Austausch mitgeschickt
- Cookies