



## Motivierende Fragen

- Welcher Netzmanagement-Standard wurde für das Internet entwickelt?
- In welcher Darstellung werden dabei Daten übermittelt?
- Wie funktioniert e-mail?
- Wie funktioniert das Web?
- Wie funktioniert die Abbildung von Namen auf Adressen im Internet?



## Übersicht

- |  |   |
|--|---|
| <ol style="list-style-type: none"> <li>1. Einführung und Motivation <ul style="list-style-type: none"> <li>▪ Bedeutung, Beispiele</li> </ul> </li> <li>2. Begriffswelt und Standards <ul style="list-style-type: none"> <li>▪ Dienst, Protokoll, Standardisierung</li> </ul> </li> <li>3. Direktverbindungsnetze <ul style="list-style-type: none"> <li>▪ Fehlererkennung, Protokolle</li> <li>▪ Ethernet</li> </ul> </li> <li>4. Vermittlung <ul style="list-style-type: none"> <li>▪ Vermittlungsprinzipien</li> <li>▪ Wegwahlverfahren</li> </ul> </li> <li>5. Internet-Protokolle <ul style="list-style-type: none"> <li>▪ IP, ARP, DHCP, ICMP</li> <li>▪ Routing-Protokolle</li> </ul> </li> <li>6. Transportprotokolle <ul style="list-style-type: none"> <li>▪ UDP, TCP</li> </ul> </li> <li>7. Verkehrssteuerung <ul style="list-style-type: none"> <li>▪ Kriterien, Mechanismen</li> <li>▪ Verkehrssteuerung im Internet</li> </ul> </li> </ol> | <ol style="list-style-type: none"> <li>8. <b>Anwendungsorientierte Protokolle und Mechanismen</b> <ul style="list-style-type: none"> <li>▪ <b>Netzmanagement</b></li> <li>▪ <b>DNS, SMTP, HTTP</b></li> </ul> </li> <li>9. Verteilte Systeme <ul style="list-style-type: none"> <li>▪ Middleware</li> <li>▪ RPC, RMI</li> <li>▪ Web Services</li> </ul> </li> <li>10. Netzsicherheit <ul style="list-style-type: none"> <li>▪ Kryptographische Mechanismen und Dienste</li> <li>▪ Protokolle mit sicheren Diensten: IPSec etc.</li> <li>▪ Firewalls, Intrusion Detection</li> </ul> </li> <li>11. Nachrichtentechnik <ul style="list-style-type: none"> <li>▪ Daten, Signal, Medien, Physik</li> </ul> </li> <li>12. Bitübertragungsschicht <ul style="list-style-type: none"> <li>▪ Codierung</li> <li>▪ Modems</li> </ul> </li> </ol> |
|--|---|



## Grundlagen: Rechnernetze und Verteilte Systeme

### Kapitel 8: Anwendungen

Netzmanagement, SNMP,  
SMTP, HTTP, DNS

Prof. Dr.-Ing. Georg Carle  
Lehrstuhl für Netzarchitekturen und Netzdienste  
Technische Universität München  
carle@net.in.tum.de  
http://www.net.in.tum.de



## 8. Anwendungen - Kapitelgliederung

- 8.1 Netzmanagement:
  - 8.1.1 Arten und Ursachen von Netzwerkproblemen
  - 8.1.2 Aufgaben und Ziele für das Netzwerkmanagement
  - 8.1.3 SNMP (Simple Network Management Protocol)
  - 8.1.4 Managementobjekte
  - 8.1.5 Management Information Base (MIB)
  - 8.1.6 Structure of Management Information (SMI)
  - 8.1.7 ASN.1
  - 8.1.8 Basic Encoding Rules, BER (Übertragungssyntax)
- 8.2 E-Mail
  - 8.2.1 SMTP, UA, MTA
  - 8.2.2 Beispielablauf
  - 8.2.3 MIME
- 8.3 FTP
- 8.4 WWW
  - 8.4.1 Uniform Resource Locator (URL)
  - 8.4.2 HTTP (HyperText Transport Protocol)
- 8.5 DNS

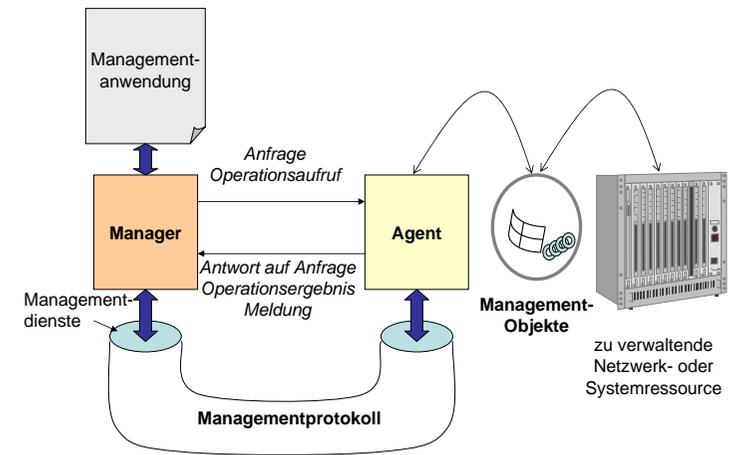


## Netzmanagement: Arten und Ursachen von Netzwerkproblemen

- **Ausfall** von Zwischen- oder Endsystemen, phys. Medien
  - Ursache: Totalausfall, Technischer Defekt
- **Fehlfunktion** von Zwischen- oder Endsystemen, Medien, Medienanschluss
  - Ursache: Teilausfall von Funktionen, intermittierende Fehler
- **Überlastung** von Zwischensystemen oder (Sub-)Netzen
  - Ursache: Fehldimensionierung, steigendes Datenaufkommen
- **Fehlkonfiguration** von Zwischen- oder Endsystemen
  - Ursache: mangelnde Erfahrung, Fehleinschätzung, Flüchtigkeitsfehler
- **Angriffe** auf Netze oder Netzkomponenten
  - Ursache: mutwillig oder fahrlässig



## Netzmanagement in der Übersicht



## Aufgaben und Ziele für das Netzwerkmanagement

- Zustandsüberwachung einzelner Netzkomponenten
- Steuerung des Netzbetriebs
- Sicherung eines effizienten und effektiven Betriebs
- Abrechnung der Netzressourcen-Benutzung
- gesicherter (geschützter) Netzbetrieb
- einfache Modellierung eines Netzwerks
- Gewinnung von Planungsdaten für das Netz und Netzwerkmodifikationen
- Planung „managebarer“ (verwaltbarer) Netze

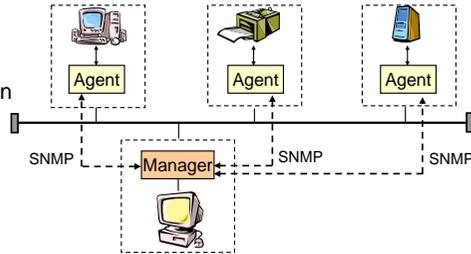


## Management im Internet

- Managementstandard im Internet (OSI-Standards nicht kompatibel)
  - benutzt das *Simple Network Management Protocol (SNMP)*
  - verbindungslos, basiert auf UDP
  - definiert eine einfache Struktur zur Modellierung von zu verwaltenden Ressourcen mit Hilfe der *Structure of Management Information (SMI)*
  - umfasst mehrere standardisierte Sammlungen von *Managed Objects*, sogenannte *Management Information Bases (MIB)*
- SNMP ist insbesondere in lokalen Netzen weit verbreitet
- Erste Version wurde 1990 erarbeitet, die aktuelle Version ist SNMPv3 nach RFC 2570 (1999). Neu ist hier vor allem die Unterstützung für Sicherheit (viele Produkte basieren noch auf SNMPv1).

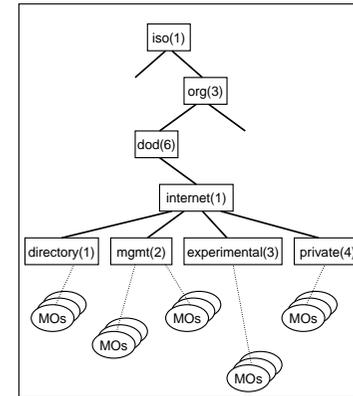
## SNMP (Simple Network Management Protocol)

- SNMP dient der Verwaltung (dem Management) beliebiger Netzwerk-Ressourcen
  - z.B. Drucker, Brücken, Router, Endsysteme usw.
- Verwaltete Ressourcen integrieren SNMP-Agenten (Software-Prozess)
  - Die Agenten verwalten die Managementinformationen der Komponente
    - z.B. Anzahl eingegangener/verlorener Pakete
- Der Manager (Software-Prozess) dient der Kommunikation mit den Agenten
  - Protokoll: SNMP (verwendet UDP)
- Basis der Kommunikation zwischen Manager und Agent: Managementobjekte



## Modellierung von Managementinformation: MIB und SMI

Management Information Base (MIB) Structure of Management Information (SMI)



**Beispiel: Object sysDescr**

sysDescr OBJECT-TYPE

SYNTAX DisplayString (SIZE (0..255))

ACCESS read-only

STATUS current

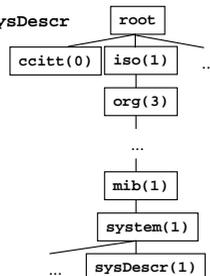
DESCRIPTION

"A textual description of the entity. This value should include the full name and version identification of the system's hardware type, software operating-system, and networking software. It is mandatory that this only contains printable ASCII characters."

::= { system 1 }

## Managementobjekte (Managed Objects)

- Managementobjekt (Managed Object):
  - Modell (Abbild) einer/mehrerer Eigenschaft(en) einer Netzwerkressource
  - Ein Agent verwaltet die Managementobjekte „seiner“ Ressource
  - Bestandteile eines Managementobjekts im Internet:
    - Eindeutiger Name, z.B. `iso.org.dod.internet.mgmt.mib.system.sysDescr`
    - Syntax: verschiedene einfache Datentypen, z.B. Integer, String, Array
    - Zugriffsrechte, z.B. read-only, read-write
    - Status, z.B. verpflichtend (mandatory), optional
- Management Information Base (MIB):
  - Gesamtheit aller Managementobjekte
  - verteilte, virtuelle Datenbank
- Management Information Tree (MIT):
  - Jedes Managementobjekt hat eindeutige Position im MIT
  - Somit eindeutige Bezugnahme möglich



## RMON – Remote Monitoring

- MIB mit speziellen Fähigkeiten (RFC 1757 u.a.)
  - Sammeln von Statistiken, Alarmen, Ereignissen
  - Teilweise Auswertung, Filtern, Packet-Capture, ...
  - ➔ Verlagerung von „Intelligenz“ von der Managementplattform weg zum Agent

□ Beispiel SMI:

etherStatsOversizePkts OBJECT-TYPE

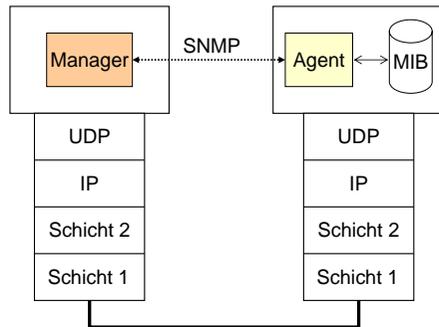
SYNTAX Counter

ACCESS read-only

STATUS mandatory

DESCRIPTION "The total number of packets received that were longer than 1518 octets (excluding framing bits, but including FCS octets) and were otherwise well formed."

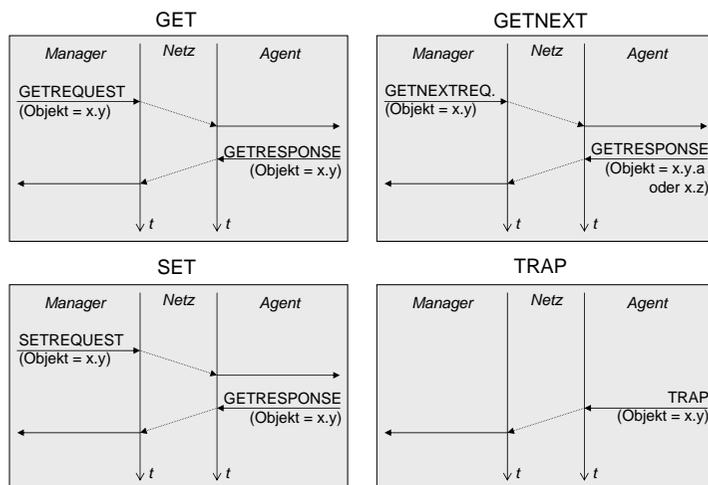
::= { etherStatsEntry 10 }



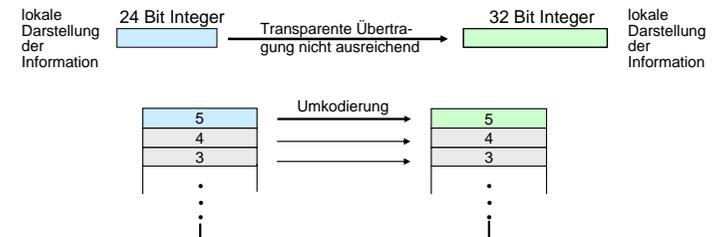
- Public Domain Tools (diverse Versionen):
  - Befehle: *snmpget*, *snmpnext*, *snmpwalk*, *snmpset*, evtl. weitere
  - Erzeugung und Dekodierung von SNMP-Dateneinheiten
  - teilweise auch mit Unterstützung für MIB-Dateien

```
snmpget -v 1 129.13.35.239 public .1.3.6.1.2.1.1.1.0
system.sysDescr.0 = "GIGAswitch Network Platform"

snmpwalk -v 1 129.13.35.239 public .1.3.6.1.2.1.1
system.sysDescr.0 = "GIGAswitch Network Platform"
system.sysObjectID.0 = OID: enterprises.DEC.2.15.3.3
system.sysUpTime.0 = Timeticks: (456990767) 52 days, 21:25:07
system.sysContact.0 = "wiltfang@telematik.informatik.uni-karlsruhe.de"
system.sysName.0 = ""
system.sysLocation.0 = ""
system.sysServices.0 = 10
```



- **Problem:**
  - Unterschiedliche Rechnersysteme besitzen verschiedene interne Darstellungen (Little/Big Endian, 16-/24-/32-Bit usw.)
  - Konsequenz: Umkodieren der zu übermittelnden Daten ist erforderlich ⇒ Austauschstandards notwendig
- **Aufgaben:**
  - Behandeln der Darstellung (Syntax) von Informationen
  - Bewahren der Bedeutung (Semantik) der Informationen



## ASN.1: Definition

- ASN.1 (abstrakte Syntax-Notation eins) ist eine von der ISO genormte Beschreibungssprache zu darstellungsunabhängigen Spezifikation von Datentypen und Werten
  - findet z.B. zur Definition von Managementobjekten bei SNMP Verwendung
- Elementare Datentypen:
  - Boolean, Integer, Bitstring, Octetstring, IA5String, ...
- Strukturierte Datentypen:
  - SEQUENCE: Geordnete Liste von Datentypen (Record in PASCAL)
  - SET: Ungeordnete Menge von Datentypen
  - SEQUENCE OF: Geordnete Liste von Elementen des gleichen Datentyps (Array in C)
  - SET OF: Ungeordnete Menge von Elementen des gleichen Datentyps
  - CHOICE: Ungeordnete Menge von Datentypen, aus der einige Datentypen ausgewählt werden können (Union in C)

Beispiel: `Mitarbeiter ::= SET { Name IA5String, Alter Integer, Personalnr Integer }`

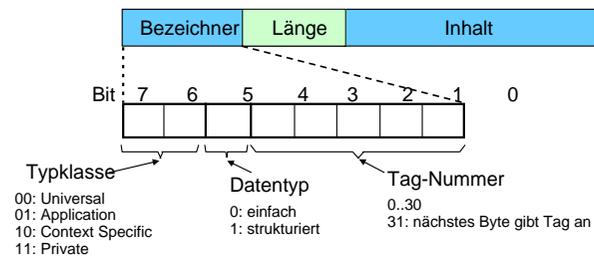
## Elektronische Post (E-Mail)

- Hauptziel:
  - Internationaler Austausch elektronischer Mitteilungen zwischen Personen
- Wesentliche Charakteristik:
  - Unterstützung eines asynchronen Verhaltens von Sender und Empfänger
  - Speichervermittlung
- Allgemeine Basisfunktionen:
  - Erstellen von E-Mails
  - Übertragung zum Ziel
  - Benachrichtigung im Erfolgs-/Fehlerfall
  - Anzeige erhaltener Nachrichten
  - Speicherung von Nachrichten
- Realisierung:
  - Simple Mail Transfer Protocol (SMTP) im Internet
  - Ehemals X.400 bei OSI
  - X.400: C=DE, A=DBP, P=BWL, O=BWLMWK, S=Wissenschaftsministerium

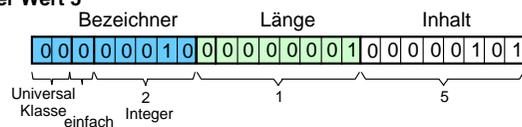


## ASN.1: Übertragungssyntax

### Basic Encoding Rules, BER (Übertragungssyntax):

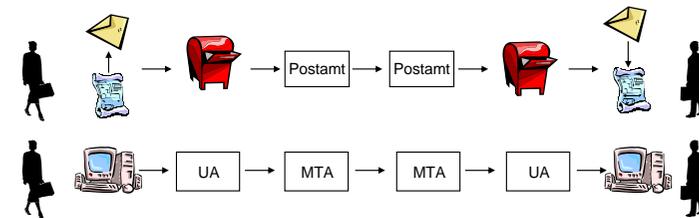


### Beispiel: Integer Wert 5



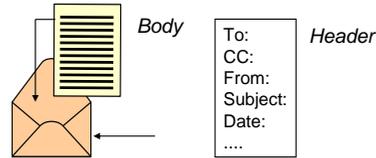
## E-Mail: Allgemeines Modell

- User Agent (UA)
  - Lokales, grafik-/textorientiertes Programm
  - Ermöglicht Lesen und Versenden von E-Mail vom lokalen Rechner
  - z.B. Elm, Mail, Outlook, Thunderbird
- Message Transfer Agent (MTA)
  - Hintergrundprozess
  - Zuständig für das Weiterleiten von E-Mails zum Zielrechner



## SMTP: Format einer E-Mail

- Umschlag (Envelope)
  - Enthält alle Informationen für den Transport der Mitteilung zum Empfänger
  - Adressierung erfolgt mithilfe von DNS, z.B. g.carle@[ieee.org](http://www.ietf.org)
  - Wird interpretiert von den MTAs
- Kopfteil (Header)
  - Optional, aber meist vorhanden
  - Enthält Informationen zu Absender und Empfänger („From:“, „To:“) sowie zusätzliche Felder wie z.B. Betreff, Kopie an („Subject:“, „CC:“)
  - Zusätzliche Informationen, die von den MTAs hinzugefügt werden
  - Interpretiert von den UAs
- Hauptteil (Body)
  - Enthält den eigentlichen Inhalt der Mitteilung (ursprünglich nur ASCII)



## SMTP: Beispielablauf

```

> telnet smtpserv.uni-tuebingen.de 25
Trying 134.2.3.3...
Connected to smtpserv.rr.uni-tuebingen.de.
Escape character is '^]'.
220 mx06.uni-tuebingen.de ESMTP Sendmail 8.13.6/8.13.6: Fri, 2 Feb 2007 10:58:49 +0100
HELO metz.informatik.uni-tuebingen.de
250 mx06.uni-tuebingen.de Hello rouen.informatik.uni-tuebingen.de [134.2.11.152], pleased
to meet you
MAIL FROM carle@informatik.uni-tuebingen.de
501 5.5.2 Syntax error in parameters scanning "FROM"
MAIL FROM: carle@informatik.uni-tuebingen.de
250 2.1.0 carle@informatik.uni-tuebingen.de... Sender ok
RCPT TO: g.carle@ieee.org
500 5.5.1 Command unrecognized: "RCPT TO: g.carle@ieee.org"
RCPT TO: g.carle@ieee.org
250 2.1.5 g.carle@ieee.org... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
From: carle@informatik.uni-tuebingen.de
To: g.carle@ieee.org
Subject: Test

Hallo, dies ist eine Testmail.

.
250 2.0.0 1129wnPo027024 Message accepted for delivery
QUIT
221 2.0.0 mx06.uni-tuebingen.de closing connection
Connection closed by foreign host.
    
```

**Angabe des (falschen) Rechnernamens wird ignoriert und stattdessen DNS-Name verwendet.**

**Fehlermeldungen nach falscher Eingabe**

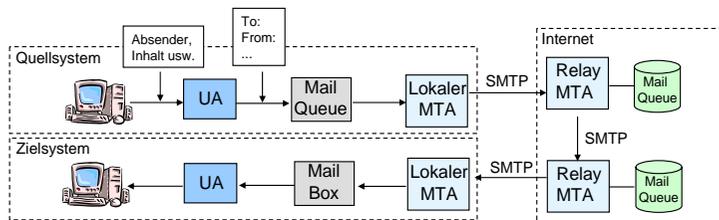
**Optional Mail-Header.**

- Falls „From:“ fehlt, wird die Absenderangabe aus „MAIL FROM:“ verwendet.
- Falls „To:“ fehlt, bleibt das Feld leer.

**Wie kommen Blind Copies (BCC:) zustande?**

## Internet Mail: Das SMTP-Modell

- SMTP dient der E-Mail-Übermittlung
  - zeichenorientiertes Protokoll, basierend auf 7-Bit-ASCII
  - nur wenige Kommandos, z.B. HELO, MAIL, RCPT, DATA, QUIT
- UA erhält alle notwendigen Angaben vom Benutzer
  - Mitteilung wird über Mail-Queue zum lokalen MTA übertragen
- MTAs übertragen die Mitteilung zum Zielrechner
  - Auslieferung einer E-Mail erfolgt über eine TCP-Verbindung (Port 25) zum Ziel-MTA (MTA unter UNIX: sendmail)
  - Relay-MTAs dienen als zentrale E-MAIL-Verteiler (z.B. Informatik-Institut)

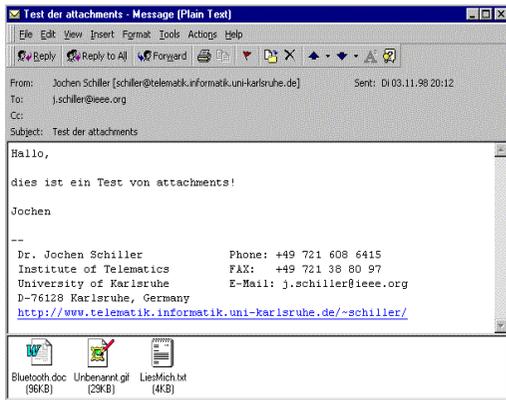


## MIME (Multipurpose Internet Mail Extensions)

- SMTP sieht nur einfache ASCII-Texte als Nachrichten vor (im Hauptteil)
- MIME erweitert den Hauptteil einer Nachricht um Formatinformationen. Hierzu werden neue Datenfelder für den Kopfteil einer Nachricht definiert:
  - Content-Type: definiert den Typ des Hauptteils.
    - Text, Multipart, Message, Application (Binary), Image, Audio, Video und X-private
  - Content-Transfer-Encoding: definiert die Transfer-Syntax, in der die Daten des Hauptteils übertragen werden.
    - Base 64, Quoted Printable, 7 Bit, 8 Bit und Binary
- Weitgehende Kompatibilität zur herkömmlichen Internet-Mail:
  - Mit der Transfersyntax Base 64 ist es möglich, Binärdaten durch Subnetze zu leiten, die nur die Übertragung von 7-Bit-ASCII-Texten erlauben.
  - Die Transfersyntax Quoted Printable erlaubt nationale Sonderzeichen. Wird eine solche Mail von einem „normalen“ Mail User Agent (Mail-Client-Programm) angezeigt, so werden nur diese Erweiterungen verstümmelt.

## MIME - Beispiel

- Email mit Text, Word-Dokument, Bild und Text-Anhang
- Ansicht in MS-Outlook:



## MIME - 1. Teil: Textnachricht

```
-----=_NextPart_000_0018_01BE0766.2BE9E500
Content-Type: text/plain;
  charset="iso-8859-1"
Content-Transfer-Encoding: 7bit
```

Hallo,

dies ist ein Test von attachments!

Jochen

--

Dr. Jochen Schiller                   Phone: +49 721 608 6415  
 Institute of Telematics               FAX: +49 721 38 80 97  
 University of Karlsruhe               E-Mail: j.schiller@ieee.org  
 D-76128 Karlsruhe, Germany  
<http://www.telematik.informatik.uni-karlsruhe.de/~schiller/>

## MIME - Standard Email-Kopf: Adressen & Co

**From:** "Jochen Schiller" <schiller@telematik.informatik.uni-karlsruhe.de>  
**To:** <j.schiller@ieee.org>  
**Subject:** Test der attachments  
**Date:** Tue, 3 Nov 1998 20:11:41 +0100  
**Message-ID:**  
 <001701be075d\$ca257d00\$732a0d81@tpc15.telematik.informatik.uni-karlsruhe.de>

```
MIME-Version: 1.0
Content-Type: multipart/mixed;
  boundary="-----_NextPart_000_0018_01BE0766.2BE9E500"
X-Priority: 3 (Normal)
X-MSMail-Priority: Normal
X-Mailer: Microsoft Outlook 8.5, Build 4.71.2377.0
X-MimeOLE: Produced By Microsoft MimeOLE V4.72.2120.0
Importance: Normal
```

MIME  
 proprietär

This is a multi-part message in MIME format.

## MIME - 2. Teil: Word-Anhang

```
-----=_NextPart_000_0018_01BE0766.2BE9E500
Content-Type: application/msword;
  name="Bluetooth.doc"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
  filename="Bluetooth.doc"
```

```
0M8R4KGxGuEAAAAAAAAAAAAAAAAAAAAAPgADAP7/CQAGAAAAAAAAAAAAAAAAAAAAAAgAAAA
AAAAA
pcEAWQAHBAACBK/AAAAAAAAEAAAAAAAABAAAP2sAAA4AYmpIavNX81cAAAAAAAAAAAAAAAA
AAAAAA
```

...

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```



## S/MIME – Sichere und signierte E-Mail

- Integration von Signatur und Verschlüsselung in E-Mail-Clients



- Neue Content-Types, z.B. Signatur:

```
Content-Type: multipart/signed; micalg=SHA1;
protocol="application/x-pkcs7-signature";
boundary="-----_NextPart_000_0017_01C01BD5.
D86ED4E0"
```

- Einfache Überprüfung von
  - Gültigkeit
  - Aussteller des Zertifikats
  - Besitzer des Zertifikats

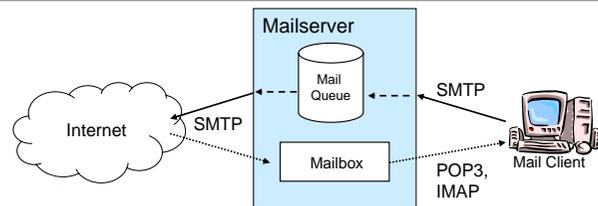


## FTP (File Transfer Protocol)

- Aufgabe: Dateiübertragung zwischen entfernten Rechnern
  - Ziel: Datenaustausch zwischen heterogenen Rechnerarchitekturen
  - FTP-Protokoll erlaubt, dass sich die Rechner auf einen geeigneten Übertragungsmodus einigen
  - Dateizugriff wird NICHT unterstützt (Attribute ändern, löschen, etc.)
  - FTP-Instanz auf TCP-Port 21
  - Dateiübertragung über kurzzeitig zugewiesene Ports
  - ASCII-Kommandos zur Ablaufsteuerung (z.B. GET, PUT)
- FTP-Optionen:
  - Datentyp (7-Bit-ASCII, EBCDIC, Image/Binary (Bitstrom), Local)
  - Dateistrukturen (File (Bytestrom), Record, Page)
  - Übertragungsmodus (Stream, Block, Compressed)
- FTP-Dienste:
  - Verbindungsaufbau mit Authentifizierung (Passwordeingabe)
  - Dateiübertragung (z.B. put, get)
  - Operationen auf Dateisystem (z.B. cd, dir)
  - Hilfefunktionen (z.B. Kommando-Auflistung inkl. Parameter)
  - Weitere implementierungsabhängige Dienste möglich



## Internet-Mail: Verwaltung durch Mailserver



- E-Mail wird meist zentral über einen Mailserver abgewickelt (Relay-MTA)
- Mittels **POP3** (Post Office Protocol 3) holt der Client die vom Mailserver empfangenen und in der Mailbox gespeicherten Meldungen ab
  - einfache Funktionalität
- IMAP** (Internet Message Access Protocol) dient dazu, die Nachrichten zentral auf einem Mailserver zu verwalten
  - IMAP erlaubt erweiterte Kommandos (z.B. Filterung)
- Beispiele für Mail-Client-Programme
  - Outlook Express (Microsoft)
  - Messenger (Netscape)



## Zur Entwicklung des World Wide Web (WWW)

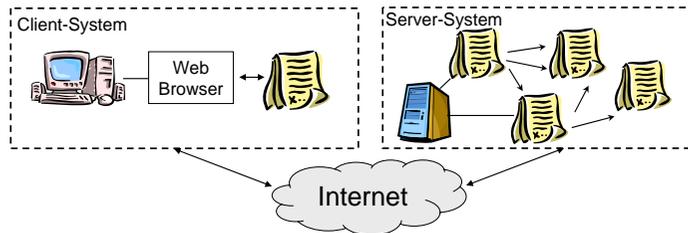
- Hervorgegangen aus Arbeiten des britischen Informatikers Tim Berners-Lee am europäischen Forschungszentrum CERN (Genf)
  - Ziel: Einfacher weltweiter Austausch von Dokumenten zwischen Wissenschaftlern
- Erster Prototyp Ende 1990
  - grafisch (auf NEXTStep) und zeilenorientiert
- Durchbruch des WWW durch den WWW-Client Mosaic
  - entwickelt von Marc Andreessen und Eric Bina (NCSA at UIUC: National Center for Supercomputer Applications at Univ. of Illinois Urbana-Champaign)
  - ursprünglich für X-Windows-Systeme
  - als Quellcode per FTP kostenlos verfügbar ⇒ schnelle Verbreitung
  - Marc Andreessen gründete 1995 die Firma Netscape
- Gründung des W3-Konsortiums im Juli 1994
  - vorrangiges Ziel: Weiterentwicklung des WWW, z.B. durch Standardisierung von HTML
  - Vorsitzender: Tim Berners-Lee
  - Infos unter <http://www.w3.org>





## Client/Server-Architektur des WWW

- Client-Server-Architektur
  - Web-Browser zur Anzeige von Hypertext-Dokumenten/Hypermedia-Objekten
  - Hyperlinks erlauben Navigation
- Lösungen zu folgenden Fragestellungen erforderlich:
  - Eindeutige Adressierung einer Web-Seite
  - Transport einer Web-Seite
  - Beschreibung des Inhalts (insbes. der Hyperlinks) einer Web-Seite



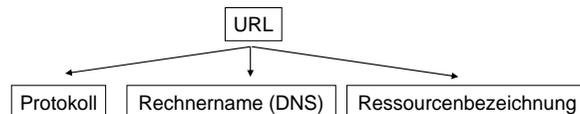
## Das WWW-Anwendungsprotokoll: HTTP

- HTTP (HyperText Transport Protocol)
  - Version 0.9 und 1.0 im RFC1945 beschrieben
  - seit Januar 1997 existiert eine Version 1.1 (RFC2068)
  - dient im Wesentlichen der Übertragung von Web-Seiten
- Wesentliche Eigenschaften
  - ASCII-Anwendungsprotokoll
  - setzt auf eine (sichere) TCP-Verbindung auf
  - Default-Port: 80
  - kurzlebige Verbindung, da der HTTP-Server nach Beantwortung einer Anfrage durch den HTTP-Client die Verbindung sofort schließt
- Beispiele von Befehlen:
  - **GET:** Anfordern eines bestimmten Dokuments
  - **HEAD:** Anfordern von Informationen im Kopfteil eines Dokuments
  - **POST:** Anhängen von Daten an ein existierendes Dokument
  - **PUT:** Anlegen eines Dokuments



## Adressierung eines Web-Dokuments

- Uniform Resource Locator (URL)
  - weist der Client-Software den Weg zu einer bestimmten Ressource
  - auch für Inhalte anderer Server (USENET, FTP, E-Mail) verwendbar
  - z.B. `http://www.informatik.uni-tuebingen.de/index.html`



- Durch die Ressourcenbezeichnung wird das Objekt, auf das im jeweiligen Server zugegriffen werden soll, identifiziert
  - bei WWW: abgerufene Web-Seite
  - bei FTP: zu übertragene Datei
  - bei Mail: Empfänger der Mail
- Web-Browser unterstützen eine Vielzahl von Protokollen
  - z.B. `http://`, `ftp://`, `mailto://`, `telnet://`, `soap://`



## Beispiel einer HTTP-Anfrage und HTTP-Antwort

HTTP-Client → HTTP-Server:

```
GET /index.html HTTP/1.1
Host:www.informatik.uni-tuebingen.de
Pragma: no-cache
....
```

- Befehlszeile: `<Befehl> <URL> <Version>`
- Client wünscht nicht zwischengespeicherte, d.h. aktuelle Version des Dokuments

**Hinweis: Verbindung zwischen Client und Server wurde bereits zuvor aufgebaut**

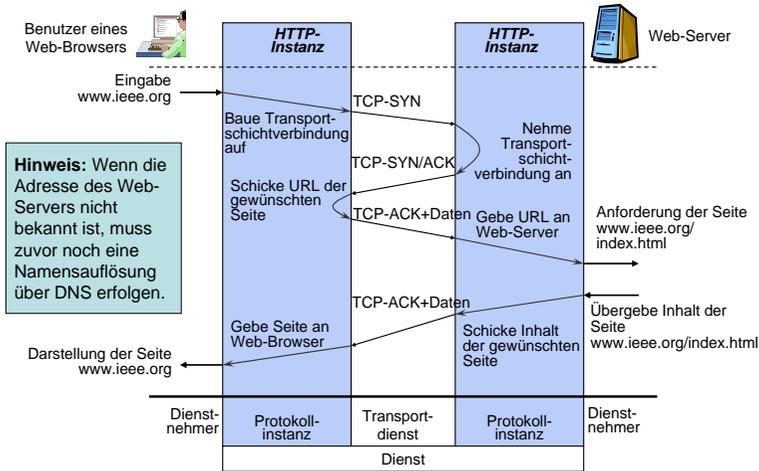
HTTP-Server → HTTP-Client

```
HTTP/1.1 200 OK
Date: Fri, 24 Sep 1999 09:45:51 GMT
Server: Apache/1.3.6 (Unix)
Transfer-Encoding: chunked
Content-Type: text/html

<HTML>
Gemäß HTML-Konventionen
strukturiertes Dokument
</HTML>
```

- Antwort-Zeile
- Datum
- Server
- Angaben zur Kodierung
- Art des Inhalt
- Hauptteil

## Beispiel: Surfen im Internet

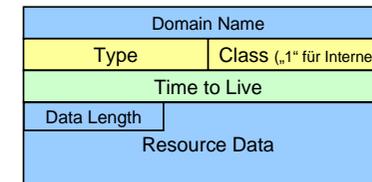


## DNS Resource Records

- Das DNS basiert auf dem Austausch sog. *Resource Records*:

Typ	Beschreibung
<b>A (Address)</b>	Abbildung: Name auf IP-Adresse
<b>MX (Mail Exchange)</b>	E-Mail-Server einer Domäne
<b>NS (Nameserver)</b>	Nameserver einer Domäne
<b>CNAME (Canonical Name)</b>	„Alias“-Namen für Rechner/Domäne
<b>PTR (Pointer)</b>	Abbildung: IP-Adresse auf Name
<b>HINFO (Host Info)</b>	zusätzliche Informationen (CPU,...)

- Format:



## DNS (Domain Name System)

- DNS stellt eine weltweit verteilte Namensdatenbank dar
- DNS besitzt eine hierarchische Namenstruktur
- DNS bildet Namen auf Information (z.B. IP-Adressen) ab
  - www.ietf.org ⇒ 199.172.136.40
  - Vorteil: Information/Adresse, auf die abgebildet wird, kann sich ändern
  - Bsp: Mailserver von IETF (z.B. für g.carle@ietf.org)

```

$ nslookup -q=mx ietf.org
Server: sioux.telematik.informatik.uni-karlsruhe.de
Address: 129.13.35.73

   ietf.org       preference = 0, mail exchanger = gemini.ietf.org
   ietf.org       nameserver = auth01.ietf.org
   ietf.org       nameserver = dns.ietf.org
   ietf.org       nameserver = ns.uu.net
   ietf.org       nameserver = krypton.ietf.org
   ietf.org       nameserver = deeptthought.ietf.org
gemini.ietf.org internet address = 199.172.136.14
auth01.ietf.org internet address = 199.172.136.2
dns.ietf.org    internet address = 199.172.136.6
ns.uu.net      internet address = 137.39.1.3
krypton.ietf.org internet address = 199.172.136.2
deeptthought.ietf.org internet address = 199.172.136.6
    
```

**MX-Record:** gemini.ietf.org

**NS-Records:** auth01.ietf.org, dns.ietf.org, ns.uu.net, krypton.ietf.org, deeptthought.ietf.org

**A-Records:** gemini.ietf.org, auth01.ietf.org, dns.ietf.org, ns.uu.net, krypton.ietf.org, deeptthought.ietf.org

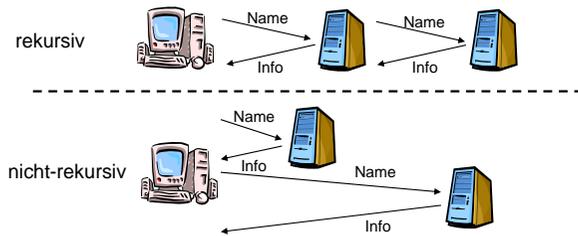
## DNS: Paketformat



- DNS verwendet UDP
  - effizient, da kein Verbindungsaufbau/Datensicherung notwendig
- DNS-Header:
  - enthält Identifikationsnummer der Anfrage, Anzahl der Einträge in den folgenden Feldern, weitere Steuerinformationen (z.B. für Rekursion)
- Anfragen:
  - bestehend aus DNS-Namen (z.B. www.amazon.de) und Typ der Anfrage (z.B. A, MX, PTR)
- Antwort-/Authority-/Zusatz-RRs
  - ein/mehrere Resource Records mit gewünschten DNS-Informationen
  - Authority-RRs: Name(n) des(r) verantwortlichen (d.h. für diese Anfrage zuständigen) Nameserver(s)

## DNS: Nameserver (NS)

- Jede Zone hat einen primären und beliebig viele sekundäre Nameserver
  - Jeder NS kennt nur einen Ausschnitt des gesamten Namensraums
  - Jeder NS kennt die IP-Adressen der NS seiner direkt untergeordneten Sub-Domains
  - Jeder NS führt Caching bereits bekannter Einträge durch
  - Sekundäre NS führen ein periodisches Update („Zonentransfer“) ihrer Datenbasis durch (basierend auf den Daten des primären NS)
- Anfragen können rekursiv oder nicht-rekursiv beantwortet werden:



## DNS: Reverse Lookup

- Aufgabe des Reverse Lookup:
  - Bestimmung des logischen Namens zu einer gegebenen IP-Adresse
  - hierzu Verwendung der PTR-Records
- Vorgehensweise:
  - Spezieller Teilbaum des DNS „in-addr.arpa“
    - dient der Zuordnung von IP-Adresse ⇒ Name
    - jede IP-Adresse entspricht Eintrag unterhalb in-addr.arpa
    - jede Stelle einer Adresse entspricht genau einem Knoten
  - eine Anfrage an das DNS enthält somit die IP-Adresse in „invertierter“ Form ⇒ hierarchische Strukturierung
- Beispiel:
  - IP-Adresse: 207.171.168.16
  - DNS-Name in Anfrage: 16.168.171.207.in-addr.arpa
  - Ergebnis (Antwort RR): Resource Data = www.amazon.de



## DNS: Beispiele (A-/MX-Records)

