

# **Programmieren für das Internet**

*Wintersemester 2005/2006*

Eric Knauel      Mandeep Singh

Universität Tübingen

# Ansprechpartner

Bei Fragen zur Vorlesung, Übungen, etc:

- *Eric Knauel*  
Raum 105, Sand 13  
**knauel@informatik.uni-tuebingen.de**
  
- *Mandeep Singh*  
Raum 104, Sand 13  
**singh@informatik.uni-tuebingen.de**

# Themen der Vorlesung

## Fokus: Programmierung von Web-Anwendungen

- Netzwerk-Grundlagen
- CGI-Programme
- PHP
- Webprogrammierung mit Continuations
- XML, XSLT und andere Sachen mit X
- JavaScript
- Sicherheit im Internet

# *Organisatorisches*

# Homepage zur Vorlesung

<http://www-pu.informatik.uni-tuebingen.de/pfi-0506/>

- Skript zur Vorlesung, Beispielcode, Handouts
- Folien als PDF
- Übungsblätter, Hinweise und Korrekturen dazu

Min. einmal pro Woche draufschauen!

# Skript zur Vorlesung

- Knappe Zusammenfassung des Stoffes
- Literaturhinweise
- Erscheint kapitelweise auf der Homepage
- Erst nach Abschluß eines Kapitels

# Folien zur Vorlesung

- Vorläufige Version am Montag Mittag auf der Homepage
- Für Ausdruck geeignet
- Evt. Korrekturen nach der Vorlesung

Empfehlung: ausdrucken, mitbringen, Notizen machen

# Übungen

Anders als gewohnt...

- Zuerst: zwei normale Übungsblätter (1 Woche)
- Dann: drei Projektblätter (3-4 Wochen)
- Alles in Teamarbeit (2-3 Leute)

# Projektblätter

- Größere zusammenhängende Aufgabe
- Arbeit für 3-4 Wochen und 2-3 Leute
- Unterstützung durch Tutoren
- Nach einer Woche: Treffen mit Tutor
- Abgabe: Vorführung bei Mandeep oder Eric
- Evt. Nachbesserung innerhalb einer Woche

# Übungsgruppen

- Finden nur zweimal statt
- Besprechung der Übungsblätter

Termine stehen (unverrückbar) fest:

- Do, 13-15 Uhr c.t., kleiner HS, Sand 6/7
- Do, 15-17 Uhr c.t., kleiner HS, Sand 6/7
- Fr, 10-12 Uhr c.t., kleiner HS, Sand 6/7

# Scheinkriterien

- Fristgerechte Abgabe
- Insgesamt 60% auf den Übungsblättern
- Alle drei Projekte bestanden

# Account am WSI

Noch kein Account am WSI?

- Formular in der Pause abholen
- Formular ausfüllen
- Bei Mandeep abgeben

# Wichtige Durchsage

*Betrifft: Bachelor- und Master-Studenten*

*In der Pause bei uns melden!*

*Fragen zur Organisation?*

# Programmieren für das Internet

Was ist Programmieren für das Internet?

- Netzwerk-Programmierung
- Nebenläufige Programmierung
- Programmierung interaktiver GUIs
- Programmieren für eine heterogene Umgebung

⇒ Wissen aus allen genannten Bereichen notwendig

# Netzwerk-Grundlagen

Protokolle sind Regeln für die Kommunikation

Protokolle definieren:

- die Syntax von Nachrichten
- den Ablauf der Kommunikation
- häufig: die Syntax für Adressen zur Identifikation des Kommunikationspartners

# Protokollschichten

Protokolle werden in Schichten organisiert.

ideales Schichtenmodell:

- Jede Schicht definiert eine Aufgabe/Eigenschaft
- Ein Protokoll löst die Aufgabe einer Schicht
- Die Eigenschaften drunterliegender Schichten können vorausgesetzt werden
- Protokoll muss die Übertragung beliebig strukturierter Daten erlauben

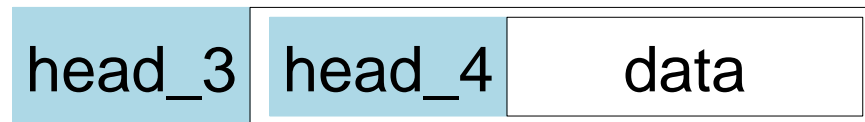
⇒ Modularer Aufbau, wiederwendbare Protokolle

# Geschachtelte Protokolle

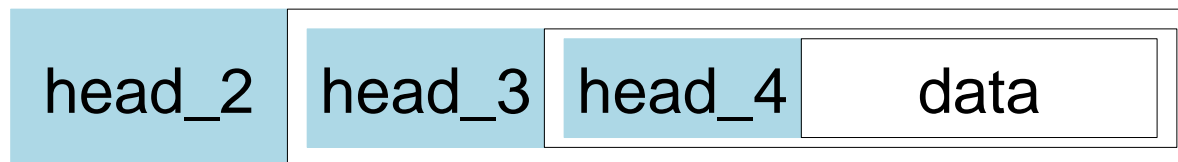
Ein Protokoll (Schicht 4) verpackt zu übertragene Daten in ein Paket:



Die drunterliegende Schicht verpackt diese Daten abermals:



Das fertige Paket:



# ISO/OSI Schichtenmodell

Ein Standard für den Aufbau Protokollhierarchie

7	Verarbeitungsschicht ( <i>application layer</i> )
6	Darstellungsschicht ( <i>presentation layer</i> )
5	Sitzungsschicht ( <i>session layer</i> )
4	Transportschicht ( <i>transport layer</i> )
3	Vermittlungsschicht ( <i>network layer</i> )
2	Sicherungsschicht ( <i>data-link layer</i> )
1	Bitübertragungsschicht ( <i>physical layer</i> )

# Schicht 1

## Bitübertragungsschicht (physical layer):

- Stellt Übertragungsmöglichkeit für rohe Bits bereit
- Implementiert durch Hardware
- Z. B. Kodierung von Bits durch Spannungsdifferenzen

Beispiele: Ethernet, Lichtwellenleiter

## Schicht 2

### Sicherungsschicht (data link layer):

- Stellt die Übertragungsmöglichkeit als Leitung dar
- Garantiert fehlerfreie Übertragung
- Daten werden als *Datenrahmen* bestimmter Größe versandt
- Korrekter Empfang wird bestätigt
- Im Fehlerfall: Rahmen wird erneut versendet

Beispiele: PPP, SLIP, (IEEE 802)

## Schicht 3

Vermittlungsschicht (network layer):

- *Routing*: Auswahl eines geeigneten Übertragungsweges
- Über mehrere Zwischenstationen hinweg
- Häufig verwendet: Wegbeschreibung durch *Routingtabellen*

Beispiele: IP, IPX, AppleTalk

## Schicht 4

Transportschicht (transport layer):

- Punkt-zu-Punkt-Verbindung von Quelle zum Ziel
- Garantiert fehlerfreie Übertragung
- Pakete kommen in Sendereihenfolge an
- Flußkontrolle: Anpassung der Geschwindigkeit der Kommunikation

Beispiele: TCP, (UDP)

# Schicht 5

## Sitzungsschicht (session layer):

- *Dialogue control*: regelt Duplexbetrieb
- *Token management*: exklusive Ausführung einer Operation
- *Check points*: Wiederaufnahme unterbrochener Übertragungen ab einem gesetzten Punkt im Datenfluß

Beispiel: UNIX RPC

# Schicht 6

## Darstellungsschicht (presentation layer)

- Darstellung der Daten in vereinbarter Weise
- Sender und Empfänger interpretieren Daten gleich
- Zum Beispiel: Gleitkommazahlen

Beispiel: XDR

# Schicht 7

## Verarbeitungsschicht (application layer)

- Kommunikation zwischen Anwendungen
- Vielzahl von Protokollen
- NFS greift auf alle Dienste unterliegender Schichten zu
  
- Beispiele: HTTP, FTP, SMTP, ...

# Internet-Protokoll (IPv4)

- ***Das Protokoll*** für Kommunikation im Internet
- Implementiert Schicht 3 (network layer)

Eigenschaften:

- Routing
- Fragmentierung
- *verbindungslos*: jede Nachricht wird einzeln versendet
- *nicht zuverlässig*: erfolgreiche Übertragung nicht garantiert

# IP-Pakete

Besteht aus Header (min. 20 Byte) und Datenteil (max. 64 KB)

Header eines IP-Paketes (Auswahl):

- Quelladresse
- Zieladresse
- Protokollbezeichner darüberliegendes Protokoll
- Checksumme
- Länge des gesamten Paketes
- u.v.a.m., insgesamt 13 Header

# IP-Adressen

Eine IP-Adresse ist eine 32-Bit-Zahl.

- Bestehen aus *Netz-Id* und *Host-Id*
- Die ersten Bits bestimmen die *Klasse*:

- Class A

0	7 Bits für Netz-Id	24 Bits für Host-Id
---	--------------------	---------------------

- Class B

10	14 Bits für Netz-Id	16 Bits für Host-Id
----	---------------------	---------------------

- Class C

110	21 Bits für Netz-Id	8 Bits für Host-Id
-----	---------------------	--------------------

# Versenden von IP-Nachrichten

## Versenden einer IP-Nachricht

- Befindet sich Ziel im selben Netz (Vergleich der Netz-Ids)?
- Ja: direkter Versand über Protokoll der Schicht 2
- Nein: Versand an Router
  
- Router betrachtet nur Netz-Id
- Erst im Zielnetzwerk wird die Host-Id wieder betrachtet

# Subnetze

Ein Netz kann in Subnetze aufgeteilt werden.

Die *Subnetz-Maske* zerlegt die Host-Id in

- Die *Subnetz-Id*
- Die *eigentliche Host-Id*

## Beispiel Subnetz-Maske

Die Netz-Id der Uni Tübingen ist 134.2. Binär:

10000110	00000010
----------	----------

5 Bits kodieren die Subnetz-Id. Die Subnetz-Id für das WSI ist 1.

Daher haben die IP-Adressen des WSI den folgenden Präfix:

10000110	00000010	00001xxx	xxxxxxxxxx
----------	----------	----------	------------

Gesucht: Eine Subnetz-Maske, die bei UND-Verknüpfung genau diesen Präfix erhält.

11111111	11111111	11111000	00000000
----------	----------	----------	----------

⇒ 255.255.248.0 dezimal

# Versenden von IP-Nachrichten und Subnetze

- Innerhalb eines Subnetzes werden Pakete direkt versendet
- Deshalb: Quell- und Zieladresse im selben Subnetz?

Berechne:

Quelladr. UND Subnetz-Maske = Zieladr. UND Subnetz-Maske

Im Beispiel: selbes Subnetz, wenn beide Seiten:

10000110	00000010	00001000	00000000
----------	----------	----------	----------