

## Übung 1 - Informationsübertragung

### 1. Signale

- Was unterscheidet Information und Signal?
- Wieviele Stufen muß ein Signal mindestens haben, um 3 bit zu übertragen?
- Wie stark müssen sich 2 Signalstufen mindestens unterscheiden, wenn Rauschspannungen 1mV (fast) nie überschreiten?
- Wozu führen Signalreflexionen in Kabeln, wie können sie verhindert werden?
- Beweisen Sie mathematisch, daß eine Welle und eine totalreflektierte Gegenwelle sich zu einer stehenden Welle auslöschten.

### 2. Nyquist/Shannon-Theorem

- Warum werden Rechteckimpulse in bandbreitenbegrenzten Kanälen verformt?
- Diskutieren Sie den Verlauf einer sinusförmigen Welle mit der Frequenz 1 MHz. In welcher Zeit erreicht die Schwingung ihr 1. Maximum?
- Kann ein Rechteckimpuls bei 1 Mhz Bandbreite schneller einschwingen? Welche Signalrate und welche Datenrate sind nach dem Nyquisttheorem erreichbar?
- Welche max. Datenrate hat ein Kanal mit einer Bandbreite 600kHz und einem Kodierungsverfahren mit 10 Signalzuständen?
- Wie hoch ist die maximal erreichbare spektrale Effizienz, wenn der Signal-Rauschabstand ca. 1000 beträgt.
- Welche max. Datenrate hat ein Kanal mit der Bandbreite 20 kHz, wenn die Sendeleistung 10 mW und die Rauschleistung im Kanal 0,1 mW beträgt?

Über einen digitalen Übertragungskanal soll eine Datenrate von 9600 bit/s übertragen werden.

- Wie groß sind Schrittgeschwindigkeit und minimale Bandbreite des (rauschfreien) Übertragungskanals, wenn pro Signalschritt 4 bit kodiert werden können?
- Auf welchen Wert erhöht sich die minimal erforderliche Bandbreite bei einem Signal-Rausch-Verhältnis auf dem Übertragungskanal von  $SNR = 15$  ?

### 3. Leistung/Dämpfung

Ein Gerät sendet Signale mit 100 mW Leistung und 1 V Spannungshub über eine Strecke mit einer Dämpfung von 40 dB.

- Wie hoch ist die Empfangsleistung in mW?
- Wie hoch ist die Empfangsspannung?
- Welche Rauschleistung bzw. -spannung ist noch akzeptierbar, wenn eine spektrale Effizienz von 2 bit/s pro Hz erreicht werden soll?

#### 4. Übertragungsmedien

- a. Welche Übertragungsmedien sind besonders geeignet (Begründung) für
  - lange Übertragungsstrecken und hohe Übertragungsraten
  - kurze Übertragungsstrecken und hohe Übertragungsraten
  - mobile Nutzer
- b. Berechnen Sie die Wellenlänge elektromagnetischer Wellen mit einer Frequenz von 2,4 GHz?
- c. Welche Frequenz hat ein Laserlichtstrahl mit einer Wellenlänge von 400 nm?
- d. Warum sind Daten-Cu-Leitungen verdrillt (oder verseilt) ?

#### 5. Multiplextechniken

- a. Wie funktioniert Echokompensation?
- b. Vergleichen Sie Frequenz- und Wellenlängenmultiplex?

#### 6. Basisbandübertragung

- a. Stellen Sie die Bitfolge "011010" im Manchester-Kode dar.
- b. Stellen Sie die Bitfolge "011010" im MLT-3-Kode dar.

#### 7. Modulation

- a. Wieviel Information trägt eine reine Sinuswelle?
- b. Stellen Sie die Übertragung der Bitfolge "011010" dar mit Amplituden-, Frequenz- und Phasenmodulation.

#### 8. Bandspreizung

- a. Welche Vorteile bietet das Prinzip des Spreizens und Entspreizens eines Spektrums?
- b. Erläutern Sie das Prinzip des „Direct Sequence Spread Spectrum“ (DSSS).
- c. Bei welchen Standards wird DSSS genutzt?
- d. Erläutern Sie das Prinzip des „Frequency Hopping Spread Spectrum“ (FHSS).
- e. Bei welchen Standards wird FHSS genutzt?

## 9. Pulscodemodulation

Ein analoges Audio-Signal soll mittels PCM digitalisiert und binär kodiert werden. Zur Amplitudendiskretisierung dient eine lineare Quantisierungskennlinie mit 1023 Intervallen.

- a. Auf welche obere Grenzfrequenz ist das analoge Signal zu begrenzen, wenn zur Übertragung des digitalen Signals nur eine Kanalbandbreite von 60 kHz zur Verfügung steht?
- b. Wie groß ist (näherungsweise) der maximale (absolute) Quantisierungsfehler, wenn die maximale Signalamplitude 5 V beträgt?
- c. Wie kann der relative (auf den Signalwert bezogene) Quantisierungsfehler reduziert werden?

## Übung 2 - Strukturierte Verkabelung

### 1. Standardisierung

- a. Warum ist die Bedarfs- gegenüber der anwendungsneutralen Verkabelung nachteilig?
- b. Welche Lebensdauer hat die Verkabelungsinfrastruktur im Vergleich zu Geräten?
- c. Welchen Anteil an den Gesamtkosten hat die Verkabelung im Vergleich zu Geräten?
- d. Was besagt die Norm EN 50173?
- e. Welche weiteren Spezifikationen zu strukturierter Verkabelung existieren?
- f. Was versteht man unter einem „Link“ und einem „Channel“ ?
- g. Welche Anforderungen muß ein Netzwerk nach Klasse D erfüllen?

### 2. Mediennutzung

- a. Vergleichen Sie die Eigenschaften von Lichtwellenleitern und TP-Kabeln.
- b. Ordnen Sie den folgenden Einsatzfällen sinnvolle Übertragungsmedien zu
  - 10 Gbit/s, 40 km
  - 100 Mbit/s, 80 m
  - 100 Mbit/s, 200 m
- c. Berechnen Sie die Signallaufzeit für ein TP-Kabel mit 50 m Länge und einem NVP-Wert von 0,66.
- d. In welchen Grenzen schwankt die Signallaufzeit, wenn herstellerbedingt der NVP-Wert um 10% schwankt.
- e. Wieviel Bitzeiten entspricht diese Schwankung bei 10 Gbit/s Übertragungsrate?

### 3. Dimensionierung/Messung

- a. Welche Empfangsleistung ergibt sich bei einer Übertragungstrecke (50 m) mit einer Dämpfung von 18 dB/100m bei einer Sendeleistung von 10mW?
- b. Welche der nachfolgenden Größen ist längenabhängig
  - a. NEXT
  - b. FEXT
  - c. ELFEXT
- c. Wann tritt Alien NEXT auf und wie kann es verringert werden?

#### 4. Wärmeableitung

- a. Beurteilen Sie UTP und S/FTP-Kabel bezüglich der Wärmefestigkeit.
- b. In Netzen mit strukturierter Verkabelung kann auch die Stromversorgung über die TP-Kabel realisiert werden. Die Speisespannung sei 45 V, die Leistung max. 15 W. Berechnen Sie, wieviel Wärmeenergie bei 50 Kabeln mit je 90 m Länge und einem Schleifenwiderstand von  $167 \Omega/\text{km}$  erzeugt wird.

#### 5. Lichtwellenleiter

- a. Vergleichen Sie die Eigenschaften von Multimode- und Monomode-LWL
- b. Nach welcher Streckenlänge (Faser mit Kategorie OS1; Wellenlänge von 1310 nm) darf die Sendeenergie frühestens auf 10% sinken ?
- c. Für welche Ethernetvarianten darf eine OM2e-Faser über eine Strecke bis 2000 m eingesetzt werden?

## Übung 3 – Ausbreitungsmodelle

Benutzen Sie ggf. die Mini Tools auf der CANDY-Projektseite  
(<http://candy.inf.tu-dresden.de/candy/tools/candy-prop-outdoor.htm>)

### 1. Ausbreitung/Freiraumdämpfung

- a. Wie hoch ist die Signallaufzeit im freien Raum über eine Distanz von
  - 30 m typisch für WLAN
  - 3 km WiMAX
  - 36000 km geostationäre Satelliten
  - 227,1 Mill. km mittlere Entfernung Erde-Mars
- b. Wie verändert sich die Freiraumdämpfung (in % und in dB) bei einer Verdoppelung des Abstandes zwischen Sender und Empfänger?
- c. Stellen Sie die Gleichung für den Pfadverlust bei Freiraumdämpfung  $L[\text{dB}] = 32,44 + 20 \lg(f/\text{MHz}) + 20 \lg(R/\text{km})$  so um, daß Frequenzen in *GHz* und Abstände in *m* angegeben werden.

In einem Unternehmen existiert ein WLAN mit der Sendefrequenz 2,4 GHz.  
Ein Access Point ist im Außenbereich installiert und hat eine Sendeleistung  $P_{\text{TX}} = 30 \text{ mW}$ .

- d. Wie groß ist die Empfangsleistung  $P_{\text{RX}}$  in 35 m Abstand?  
(Wenden Sie das Modell der Freiraumdämpfung an.)
- e. Ab welchem Abstand ist der Empfang praktisch nicht mehr möglich?  
(Empfangsleistung unter  $10^{-10} \text{ W}$ )

### 2. EIRP

Für WLAN 802.11b/g ist eine max. EIRP-Leistung von 100 mW gestattet.

- a. Wie hoch darf die max. Sendeleistung  $P_{\text{TX}}$  sein, wenn eine Sendeantenne mit einem Gewinn von 12 dBi eingesetzt wird?

### 3. Fresnelzone

Zwei Richtfunk-Stationen haben einen Abstand von 10 km. In der direkten Verbindungslinie befinden sich Gebäude mit einer Maximalhöhe von 20 m.

Die Sendefrequenz beträgt 6 GHz.

Die 1. Fresnelzone darf nur zu max. 20% durchdrungen werden.

- a. Berechnen Sie die minimal erforderliche Antennenhöhe der Stationen.
- b. Welche Antennenhöhen sind für einen Abstand von 50 km erforderlich?

## Übung 4 - Sicherungsschicht

### 1. Fehlerwahrscheinlichkeit

Ein Computer sendet Frames mit einer durchschnittlichen Größe von 1000 Byte und einer Datenrate von 100 Mbit/s. Die Bitfehlerwahrscheinlichkeit beträgt  $10^{-8}$ . Zusätzlich gibt es im durchschnittlichen Zeitabstand von 10 min burstartige Störungen mit einer Bitfehlerwahrscheinlichkeit von  $10^{-3}$  und 1 ms Dauer.

- Berechnen Sie den Einfluß beider Störarten.
- Welcher Störeinfluß dominiert?

### 2. Fehlererkennung durch Paritätskontrolle

Der ASCII-Basiskode ist ein Zeichendarstellungskode für Klein- und Großbuchstaben des englischen Alphabets, für Ziffern, für Sonder- und für Steuerzeichen

Auszug

A	B	...	O	P	Q	...	Z
0x41	0x42		0x4F	0x50	0x51		0x5A

- Notieren Sie zeichenweise untereinander die Zeichenkette „ABCDEFGH“ in binärer 7-bit-Darstellung.
- Fügen Sie ein Kontrollbit für gerade Parität hinzu.
- Welche Bitfehlersituationen können erkannt und welche korrigiert werden?
- Wie hoch ist bei einer Bitfehlerwahrscheinlichkeit von  $10^{-3}$  die Wahrscheinlichkeit eines fehlerhaften Zeichens und wie hoch ist die Wahrscheinlichkeit, daß der Fehler nicht erkannt wird?
- Fügen Sie ein Kontrollzeichen für gerade Blockparität hinzu.
- Welche Bitfehlersituationen können erkannt und welche korrigiert werden?
- Für welche Anwendungen wäre ein solcher Kode geeignet?

### 3. Fehlererkennung durch CRC

Prüfen Sie folgende empfangene Bitfolgen auf Fehlerfreiheit (Generatorpolynom  $x^3 + x + 1$ ).

- a. 110110101
- b. 100110011

### 4. Fenstersteuerung I

Ein Kanal habe eine Datenrate von 40 kbit/s und eine phys. Signallaufzeit von 20ms.

- a. Für welche Rahmengrößen ergibt sich für das Stop-and-Wait-Protokoll eine Effizienz von mindestens 50%? Die Quittungsgröße sei vernachlässigbar.
- b. Wie verändert sich das Ergebnis bei Nutzung eines Schiebefensterprotokolls mit Fenstergröße 4 ?
- c. Wie hoch muß die Fenstergröße mindestens sein, damit bei 100 byte Framelänge eine Kanalauslastung von 100% erreicht wird?

### 5. Fenstersteuerung II

- a. Skizzieren Sie ein Weg-Zeit-Diagramm für die Datenübertragung mit einem Schiebefensterprotokoll der Fenstergröße 3, wobei die Fehlerbehandlung durch go-back-n realisiert wird und das zweite gesendete Frame zunächst verloren geht, aber nach Wiederholung erfolgreich übertragen wird.
- b. Skizzieren Sie für jede Übertragung den Zustand des Sendefensters.
- c. Weshalb reicht es bei einem Schiebefensterprotokoll der Fenstergröße F nicht aus, die Frame-Laufnummern 0 ... (f-1) zu vergeben ?
- d. Erläutern Sie, wie die Bestätigung bei HDLC bei Duplex-Datenübertragung versendet werden.

## Übung 5 – Layer-2-Netzwerktechnologien

### 1. ALOHA

Das ALOHA-Verfahren realisiert eine sehr einfache Medienzugriffssteuerung.

- Erläutern Sie, warum das ALOHA-Verfahren nur bei geringen Netzauslastungen befriedigend funktioniert.
- Erläutern Sie, warum Slotted-ALOHA einen besseren Durchsatz erreicht als das einfache ALOHA-Verfahren

### 2. CSMA

CSMA-Verfahren reduzieren die Kollisionswahrscheinlichkeit.

- Erläutern Sie, warum bei p-persistent CSMA für kleinere Werte von p ein günstigeres Verhalten im Hochlastfall erreicht wird.
- Weshalb darf die Framelänge bei Ethernet einen bestimmten Wert nicht unterschreiten ?
- Wie groß ist dieser bei IEEE 802.3 ?
- Wie groß müßten die Frames mindestens sein bei einem CSMA/CD-Netz der Länge 200 m, einer Datenrate von 1 Gbit/s und einer Signalausbreitungsgeschwindigkeit von 200000 km/s ?
- Ist das Ergebnis von d) vereinbar mit dem Frameformat von IEEE 802.3 ?

### 3. Tokenverfahren

- Vergleichen Sie das Hochlastverhalten von CSMA/CD und Tokenverfahren
- Vergleichen Sie das Echtzeitverhalten von CSMA/CD und Tokenverfahren
- Vergleichen Sie Token-Ring- und Token-Bus-Verfahren?
- Ab welcher Ringlänge passen mehrere Frames zeitgleich auf den Ring bei Datenraten von 100 Mbit/s und Framegrößen von 1000 bit und einer Signalausbreitungsgeschwindigkeit von 200 000 km/s ?
- Berechnen Sie für einen Token-Ring-LAN nach IEEE 802.5 die maximale Wartezeit einer sendewilligen Station, bei 1 km Ringlänge, 200000 km/s Ausbreitungsgeschwindigkeit, 4 Mbit/s Datenrate, 10 Stationen und einer maximalen Framelänge von 4000 Byte!  
Zur Vereinfachung wird angenommen, dass alle Stationen sendewillig sind und nur einen Frame gleicher Priorität senden wollen. Die Stationsverzögerung wird nicht beachtet.

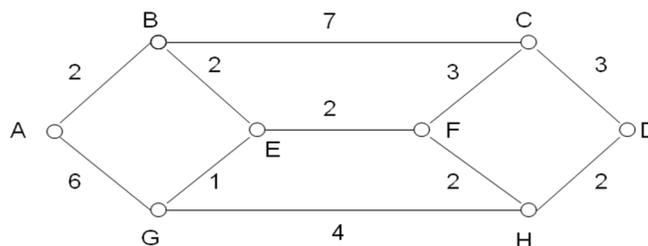
## Übung 6 - Vermittlungsschicht

### 1. ATM - Virtuelle Verbindungen

- Diskutieren Sie den Unterschied zwischen Pfaden/Kanälen und Netzwerkadressen.
- Wie viele Pfade und wie viele Kanäle können innerhalb des ATM-Netzwerks zwischen zwei Netzknoten (NNI) genutzt werden?
- Was versteht man unter einer virtuellen Verbindung?
- Ein Videostrom von 1,8 Mbit/s soll über ein ATM-Netz übertragen werden. Die physikalische Übertragungsrate sei 63,6 Mbit/s. Schlagen Sie eine sinnvolle Organisation des Zellstromes vor.

### 2. Link State Routing

Gegeben sei ein Netz mit folgender Topologie und Kostenbewertung der Übertragungspfade



- Wovon können die Kostenbewertungen im realen Netz abhängen? Diskutieren Sie drei Parameter!
- Bestimmen Sie schrittweise den kürzesten Pfad von B nach D nach dem Verfahren "Shortest Path Routing" von Dijkstra.

### 3. Distance Vector Routing

Gegeben sei das gleiche Netzwerk wie in Aufgabe 2.

Die Router sind in der Lage, die Übertragungskosten zu ihren Nachbarn zu bestimmen. Im 30-Sekunden-Rhythmus senden sie ihre Distanzvektoren an ihre Nachbarn.

- Wie ändert sich die Routingtabelle des Knotens E, wenn dieser nach einem Systemausfall folgende Distanzvektoren von seinen Nachbarn erhält?  
von B ( 2, 0, 7, 8, 2, 4, 3, 6 ) , von F ( 6, 4, 3, 4, 2, 0, 3, 2 ) und von G ( 5, 3, 6, 6, 1, 3, 0, 4 )
- Skizzieren Sie für alle Router die schrittweise Entwicklung der Tabelleneinträge bezüglich Knoten D vom Initialzustand bis zum stabilen Endzustand.
- Wie ändern sich jetzt die Weiterleitungstabellen nach Ausfall der Knotens C und H in den folgenden 5 Zyklen?

#### 4. IP-Adressierung

Eine Firma besitzt einen IP-Adreßbereich von 128.10.192.0 bis 128.10.199.255

- a. Wie viele Adressen enthält dieser Bereich? Berechnen Sie die Netzmaske. Der Bereich soll in mehrere Subnetze mit jeweils 30 Hosts aufgeteilt werden. Geben Sie eine geeignete Subnetzmaske an!
- b. Wie viele Teilnetze mit je 30 Hosts können adressiert werden?
- c. Teilen Sie unter Berücksichtigung der gewählten Subnetzmaske die IP-Adresse 128.10.192.70 in Netz- und Host-Anteil auf.

#### 5. IP - Fragmentierung

Ein TCP-Paket mit 2048 Byte Nutzdaten wird an IP zur Auslieferung übergeben. Der Übertragungsweg geht über zwei Netzwerke (Quellrechner → Router → Zielrechner).

Netzwerk 1: MTU = 1024 Byte

Netzwerk 2: MTU = 512 Byte

- a. Geben Sie für die beim Empfänger ankommenden Fragmente jeweils die Größe und den Offset an.
- b. Wie viele Fragmente würden erzeugt, wenn der Sender wüsste, dass die kleinste MTU auf den Pfad zum Empfänger 512 Byte beträgt?

#### 6. AS-Routing

Am 24.2.2008 um 18.47 Uhr startete der Administrator des AS17557 (Pakistan) eine falsche Netzanündigung (Präfix 208.65.152.0/24). Diese wurde über AS3491 (PCCW Global) weltweit verbreitet. Erst 21.01 Uhr wurde der Fehler durch AS17557 korrigiert. Weltweit wurde erst am 25.2. um 01.00 Uhr ein normaler Internetbetrieb wieder erreicht. (<http://www.ripe.net/news/study-youtube-hijacking.html>)

- a. Was kann bei einer solchen Fehlankündigung passieren?
- b. Finden Sie heraus, welche anderen AS durch die Fehlankündigung betroffen waren. (<http://www.ris.ripe.net/cgi-bin/riswhois.cgi>)
- c. Warum war die Reaktion des betroffenen Systems mit der Gegenwehr-Ankündigung (Präfix 208.65.152.0/24) um 20.07 Uhr nicht ausreichend?
- d. Finden Sie eine bessere Abwehrankündigung, original 20.18 Uhr erfolgt.

## Übung 7 - Transportschicht

### 1. Nutzerschnittstelle TCP/UDP

- a. Welche Aufgabe haben Portnummern?
- b. Welche Dienstqualität bietet UDP?
- c. Welche Dienstqualität bietet TCP?

### 2. Socketschnittstelle

Diskutieren Sie die Aufgaben der Socketprozeduren für die Nutzung von

- a. UDP
- b. TCP

### 3. Interna

- a. Wie erfolgt die Prüfsummenberechnung bei UDP?
- b. Wie erfolgt der Verbindungsaufbau bei TCP?
- c. Wie erfolgt der Verbindungsabbau bei TCP?
- d. Wie erfolgt die Quittierung von TCP-Segmenten?
- e. Was sind die Ziele der Fluß- und der Staukontrolle?  
Wie werden diese Ziele erreicht?
- f. Benennen Sie zwei unterschiedliche TCP-Implementationen.  
Können unterschiedliche Endsysteme über eine TCP-Verbindung zusammenarbeiten?

### 4. TCP – Fehlererkennung-/korrektur

- a. Sendewiederholung (Retransmission) bei ausbleibender Quittierung  
Welche Folgen hat ein zu langes RTO (Retransmission Timeout)?  
Welche Folgen hat ein zu kurzes RTO ?  
Kann RTO auf den RTT-Wert (Round Trip Time) gesetzt werden?
- b. Berechnen Sie die zeitliche Entwicklung des RTO-Wertes nach RFC 2988 für eine Serie von Übertragungen mit den RTT-Meßwerten (1s, 2s, 3s, 2s) und den Glättungsfaktoren  $\alpha=1/8$  und  $\beta=1/4$ .
- c. Sendewiederholung nach Serie duplizierter Quittungen (Fast Retransmission)  
Nennen Sie zwei Ursachen für duplizierte Quittungen.  
Wieviele duplizierte Quittungen werden ohne Sendewiederholung akzeptiert?

- d. Eine TCP-Verbindung hat eine Fenstergröße von 8 Segmenten und eine RTT von 800ms. Der Sender überträgt mit einer konstanten Rate von 1 Segment/100ms, der Empfänger sendet Quittungen ohne Verzögerung zurück. Segmentverluste werden durch Fast Retransmission erkannt.

Wieviel Zeit verliert der Sender bei einem Datenverlust?

## 5. TCP – Flußsteuerung

Ein Sender sendet TCP-Segmente (Daten 1 ...1024 Byte), zunächst mit 10 vollen Segmenten pro Sekunde, der Empfänger verarbeitet die Information byteweise mit 7 KByte/s. Der Empfangspuffer ist 30 KByte groß. Übertragungszeiten sind vernachlässigbar.

- Nach welcher Übertragungsdauer kommt es zu Verarbeitungstau im Empfänger?
- Was passiert danach?
- Wie kann die Übertragungseffizienz verbessert werden?

## 6. TCP Congestion Control

Eine TCP-Verbindung hat eine Fenstergröße von 64 KByte und eine RTT von 3 ms. Die maximale Segmentgröße MSS beträgt 1460 Byte, Überlastungen liegen nicht vor.

- Wie entwickelt sich (5 Schritte) die Datenrate bei linearem Anstieg des Sendefensters?
- Wie entwickelt sich (5 Schritte) die Datenrate bei der „Slow-Start-Methode“?
- Wie lange dauert es bei Slow Start mindestens, bis das erste volle Fenster gesendet werden kann?
- Warum kombiniert man bei TCP-Implementierungen Slow Start und linearen Anstieg?
- Vergleichen Sie TCP Tahoe mit TCP Reno.
- Interpretieren Sie nachfolgenden Verlauf der Größe des Überlastfensters (in Segmenteinheiten)

Zeit	0	+1	+2	+3	+4	+5	+6	+7	+8	+9
0	1	2	4	8	16	17	18	19	20	10
+10	11	12	13	14	15	16	1	2	4	8
+20	9	10	11	12	13	14	15	16	8	9

## 7. Socketprogrammierung - Programmierung eines einfachen WWW-Servers

Der untenstehende Programmtext enthält den C-Kode für einen einfachen WWW-Server, der auf der Socket-Schnittstelle aufbaut.

Das Programm kann mit dem frei verfügbaren Compiler Borland C++ Compiler Version 5.5 (<http://www.codegear.com/downloads/free/cppbuilder>) übersetzt werden.

Aufruf: `httpserv www-rootverzeichnis`

- a. Interpretieren Sie die Rolle der fett hervorgehobenen Prozeduren.
- b. Übersetzen und testen Sie die Funktionsfähigkeit des Servers.
- c. Entwickeln Sie einen WWW-Client (optional)

### Quell-Programm

```

/* httpserv.c
 * Demoprogramm zur Programmierung von Netzwerkservern
 * Es wird ein simpler http Server implementiert,
 * der ausschließlich GET requests bearbeiten kann */
#include <stdio.h>
#include <fcntl.h>
#include <errno.h>

#ifdef _WIN32
/* Headerfiles für Windows */
#include <winsock.h>
#include <io.h>

#else
/* Headerfiles für Unix/Linux */
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#define closesocket(s) close(s)
#endif

/* http requests werden auf Port 80 erwartet */

#define HTTP_PORT 80

static void serv_request( int in, int out, char* rootpath);

/***** MAIN *****/
int main( int argc, char **argv)
{
    struct sockaddr_in server, client;

```

```

int sock, fd;
int len;

#ifdef _WIN32
/* Initialisiere TCP für Windows ("winsock") */
short wVersionRequested;
WSADATA wsaData;
wVersionRequested = MAKEWORD(1, 1);
if (WSAStartup(wVersionRequested, &wsaData) != 0) {
    fprintf(stderr, "Failed to init windows sockets\n");
    exit(1);
}
#endif

/* Teste auf Kommandozeilenargument "documentroot" */
if (2 != argc) {
    fprintf(stderr, "usage: httpserv documentroot\n");
    exit(1);
}

/* Erzeuge das Socket */
sock = socket(PF_INET, SOCK_STREAM, 0);
if (sock < 0) {
    perror("failed to create socket");
    exit(1);
}

/* Erzeuge die Socketadresse des Servers
 * Sie besteht aus Typ und Portnummer */
memset(&server, 0, sizeof(server));
server.sin_family = AF_INET;
server.sin_addr.s_addr = htonl(INADDR_ANY);
server.sin_port = htons(HTTP_PORT);

/* Erzeuge die Bindung an die Serveradresse
 * (d.h. an einen bestimmten Port) */
if (bind(sock, (struct sockaddr*)&server, sizeof(server)) < 0) {
    perror("can't bind socket");
    exit(1);
}

/* Teile dem Socket mit, dass Verbindungswünsche
 * von Clients entgegengenommen werden */
listen(sock, 5);

/* Bearbeite die Verbindungswünsche von Clients
 * in einer Endlosschleife

```

```

* Der Aufruf von accept() blockiert solange,
* bis ein Client Verbindung aufnimmt */
for (;;) {
    len = sizeof( client);
    fd = accept( sock, (struct sockaddr*)&client, &len);
    if (fd < 0) {
        perror( "accept failed");
        exit(1);
    }

    /* Bearbeite den http Request */
    serv_request( fd, fd, argv[1]);
    /* SchlieÙe die Verbindung */
    closesocket( fd);
}
}

/*
* serv_request
* Bearbeite den auf in ankommenden http request
* Die zu sendenden Daten werden auf out ausgegeben
*/
static void serv_request( int in, int out, char* rootpath)
{
    char buffer[8192];
    char *b, *l, *le;
    int count, totalcount;
    char url[256];
    char path[256];
    int fd;
    int eoh = 0;

    b = buffer;
    l = buffer;
    totalcount = 0;
    *url = 0;
    while ( (count = recv( in, b, sizeof(buffer) - totalcount, 0)) > 0) {
        totalcount += count;
        b += count;
        while (l < b) {
            le = l;
            while (le < b && *le != '\n' && *le != '\r') ++le;
            if ('\n' == *le || '\r' == *le) {
                *le = '&#0;';
                printf ("Header line = %s\n", l);
                sscanf( l, "GET %255s HTTP/", url);
                if (strlen(l)) eoh = 1;
            }
        }
    }
}

```

```

        l = le + 1;
    }
}
if (eoh) break;
}

if ( strlen(url) ) {
    printf( "got request: GET %s\n", url);
    sprintf(path, "%s/%s", rootpath, url);
    fd = open( path, O_RDONLY);
    if (fd > 0) {
        sprintf( buffer, "HTTP/1.0 200 OK\nContent-Type: text/html\n\n");
        send( out, buffer, strlen(buffer), 0);
        do {
            count = read( fd, buffer, sizeof(buffer));
            send( out, buffer, count, 0);
            printf(".");
            fflush(stdout);
        } while (count > 0);
        close( fd);
        printf("finished request: GET %s\n", url);
    }
    else {
        sprintf( buffer, "HTTP/1.0 404 Not Found\n\n");
        send( out, buffer, strlen(buffer), 0);
    }
}
else {
    sprintf( buffer, "HTTP/1.0 501 Method Not Implemented\n\n");
    send( out, buffer, strlen(buffer), 0);
}
}
}

```

## Übung 12 - Planung

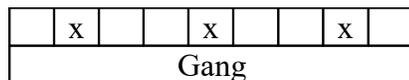
### 1. Sendeleistungen IEEE 802.11g

- Wie hoch ist die max. erlaubte Sendeleistung?
- Wie hoch ist die minimal erforderliche Empfangsleistung eines WLAN-Adapters für Datenraten von 54 Mbit/s ?
- Welcher Signal-Rauschabstand sichert eine gute Übertragungsqualität?
- In einer WLAN-Umgebung liegt überall eine Rauschleistung von -98 dBm an. Wie hoch muß die Empfangsleistung mindestens sein, damit eine ordentliche Empfangsqualität gewährleistet ist?  
Berechnen Sie nach dem Freiraumdämpfungsmodell wie hoch die Sendeleistung mindestens sein muß bei einem Senderabstand von 15m.

### 2. Frequenzplanung

Diskutieren Sie ein Grobkonzept für die WLAN-Versorgung der Zimmer (jeweils 5m x 5m) einer Gebäudeetage.

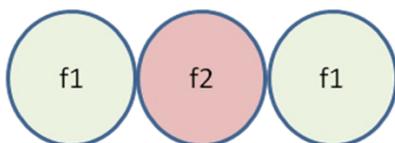
Für jeden Raum soll mindestens eine Empfangsleistung von -60 dBm zur Verfügung stehen. Benutzen Sie das Modell der Freiraumausbreitung plus Berücksichtigung der Wanddämpfungen (jeweils 15 dB).



- Wie viele Access Points würden Sie in welchen Räumen installieren?
- Welche Kanäle würden Sie den Access Points zuordnen?

### 3. Interferenzen

Ein WLAN besteht aus einer Aneinanderreihung von mehreren Zellen (Radius jeweils 8m)



- Wie hoch ist die Dämpfung am Rand der 1. Zelle?
- Wie wirkt sich dort die Sendeenergie der 2. Zelle aus?
- Welchen Einfluß hat die Sendeenergie der 3. Zelle (gleiche Frequenz)?
- Vergleichen Sie den Interferenzeinfluß im freien Gelände mit dem Einfluß in Gebäuden.

# Lösungen

**Nur für internen Gebrauch,  
unter Vorbehalt, keine Fehlergarantie !!!**

# Übung 1 - Informationsübertragung

## 1. Signale

- Was unterscheidet Information und Signal?
- Wieviel Stufen muß ein Signal mindestens haben, um 3 bit zu übertragen?
- Wie stark müssen sich 2 Signalstufen mindestens unterscheiden, wenn Rauschspannungen 1mV (fast) nie überschreiten?
- Wozu führen Signalreflexionen in Kabeln, wie können sie verhindert werden?
- Beweisen Sie mathematisch, daß eine Welle und eine totalreflektierte Gegenwelle sich zu einer stehenden Welle auslöschten.

### Zu a) Information

Beseitigte Unkenntnis

Maßeinheit 1 bit

z.B.

nicht(!)

(Unkenntnis auf 50% verringert)

männlich/weiblich

Rechtshänder/Linkshänder

### Signal

Zuordnung von Information

zu physikalischen (raum- und zeitveränderlichen) Eigenschaften eines Mediums

z.B. Form+Amplitude+Dauer elektrischer Impulse auf Kabeln

### Zu b) Signal: Zusammenhang Informationsgehalt-Stufung

$$I = \lg(S)$$

$$S = 2^I$$

$$I = 3 \text{ bit/Signal}$$

$$S = 2^3 = 8$$

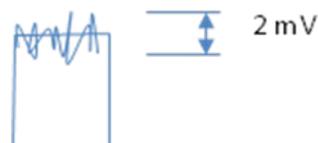


8 Stufen erforderlich

### Zu c) Rauschspannung 1mV

→ Meßwerte weichen ab vom ungestörten Signalwert

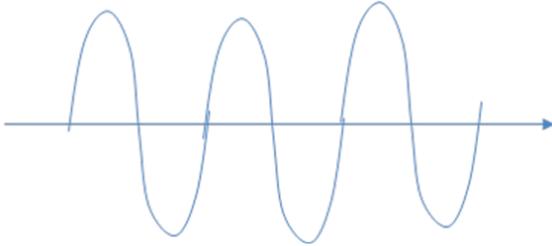
Meßwert - 1mV < Signalwert < Meßwert + 1mV



Korrekte Zuordnung in einem Bereich von 2mV möglich  
Min. Abstand zweier Signalpegel > 2mV

## Zu d) Signalreflexionen

Signale können als Wellenpaket dargestellt werden.  
Wellen bewegen sich mit konstanter Geschwindigkeit im homogenen Medium



Vereinfacht:  $y_{\text{ori}} = \sin(\omega t - 2\pi x/\lambda)$   
mit  $\omega = 2\pi f = 2\pi c/\lambda$

Ist das Medium inhomogen,  
z.B. Kabelende, Verbindung ungleicher Kabel, Kabelquetschung, ...  
gibt es Reflexionen  
(rücklaufende Welle, i.a. mit verringerter Amplitude und einer Phasenverschiebung)

Vereinfacht:  $y_{\text{refl}} = \sin(\omega t + 2\pi x/\lambda - \phi)$

speziell  $\phi = 0$  bei Reflexion am offenen Ende  
 $\phi = \pi$  bei Reflexion am kurzgeschlossenen Ende

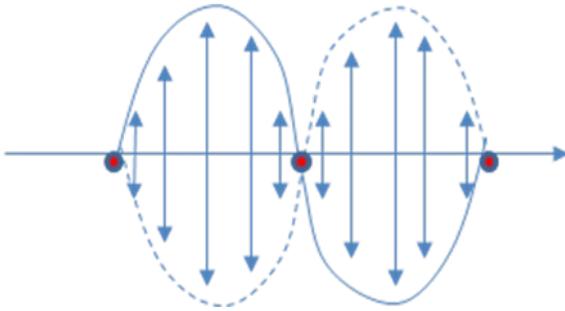
Auswirkung teilweise Auslöschung (Verschlechterung Empfang)  
evtl. stehende Wellen (kein Informationsaustausch möglich)

Verhinderung: gleicher Wellenwiderstand für alle Übertragungskomponenten  
(Kabel, Stecker, Dosen, ...)  
Kabelenden abschließen (Abschlußwiderstand=Wellenwiderstand)

Zu e) **Totalreflexion***Offenes Ende*

$$\begin{aligned}
 y &= \sin(\omega t - 2\pi x/\lambda) + \sin(\omega t + 2\pi x/\lambda) \\
 &= [\sin(\omega t) \cdot \cos(2\pi x/\lambda) - \cos(\omega t) \cdot \sin(2\pi x/\lambda)] \\
 &\quad + [\sin(\omega t) \cdot \cos(2\pi x/\lambda) + \cos(\omega t) \cdot \sin(2\pi x/\lambda)] \\
 &= 2 \cdot \sin(\omega t) \cdot \cos(2\pi x/\lambda)
 \end{aligned}$$

stehende Welle

an jedem Punkt  $x$  periodische Schwingung, aber keine Bewegung der Amplitude*kurzgeschlossenes Ende*

$$\begin{aligned}
 y &= \sin(\omega t - 2\pi x/\lambda) + \sin(\omega t + 2\pi x/\lambda - \pi) \\
 &= \sin(\omega t - 2\pi x/\lambda) - \sin(\omega t + 2\pi x/\lambda) \\
 &= [\sin(\omega t) \cdot \cos(2\pi x/\lambda) - \cos(\omega t) \cdot \sin(2\pi x/\lambda)] \\
 &\quad - [\sin(\omega t) \cdot \cos(2\pi x/\lambda) + \cos(\omega t) \cdot \sin(2\pi x/\lambda)] \\
 &= 2 \cdot \cos(\omega t) \cdot \sin(2\pi x/\lambda)
 \end{aligned}$$

ebenfalls stehende Welle

## 2. Nyquist/Shannon-Theorem

- Warum werden Rechteckimpulse in bandbreitenbegrenzten Kanälen verformt?
- Diskutieren Sie den Verlauf einer sinusförmigen Welle mit der Frequenz 1 MHz. In welcher Zeit erreicht die Schwingung ihr 1. Maximum?
- Kann ein Rechteckimpuls bei 1 Mhz Bandbreite schneller einschwingen? Welche Signalrate und welche Datenrate sind nach dem Nyquisttheorem erreichbar?
- Welche max. Datenrate hat ein Kanal mit Bandbreite 600kHz ist und Kodierungsverfahren mit 10 Signalzuständen?
- Wie hoch ist die maximal erreichbare spektrale Effizienz, wenn der Signal-Rauschabstand ca. 1000 beträgt.
- Welches max. Datenrate hat ein Kanal mit der Bandbreite 20 kHz, wenn die Sendeleistung 10 mW und die Rauschleistung im Kanal 0,1 mW beträgt?

Über einen digitalen Übertragungskanal soll eine Datenrate von 9600 bit/s übertragen werden.

- Wie groß sind Schrittgeschwindigkeit und minimale Bandbreite des (rauschfreien) Übertragungskanals, wenn pro Signalschritt 4 bit kodiert werden können?
- Auf welchen Wert erhöht sich die minimal erforderliche Bandbreite bei einem Signal-Rausch-Verhältnis auf dem Übertragungskanal von  $SNR = 15$  ?

Zu a) Fourieranalyse      periodische Rechteckimpulsfolge darstellbar als Summe von (unendlich vielen) Sinusschwingungen

Bandbreitenbegrenzung

Keine unendlich schnellen Veränderungen der Signalform möglich

Verschleifung der Rechteckform



Zu b) Frequenz 1 MHz  $\rightarrow$  Schwingungsdauer  $1 \mu\text{s}$

- Maximum      bei  $\frac{1}{4}$  der Schwingungsdauer ( $\pi/2$ )  
bei  $0,25 \mu\text{s}$

Zu c) nein, Einschwingdauer wird bestimmt durch Teilschwingung mit der höchsten Frequenz

Nyquist       $SR < 2 * B = 2 * 10^6$  Signale/s

Impulsdauer  $0,5 \mu\text{s}$

Datenrate abhängig von Signalstufung

Zu d)  $S=10 \rightarrow \text{Id}(S) = \lg(10)/\lg(2) = 3,3 \text{ bit/Signal} \rightarrow 3 \text{ bit/Signal}$

DR < Signalinformationsgehalt \* 2 \* Bandbreite

<  $3 \text{ bit/Signal} * 2 * 6*10^5 \text{ Signale/s}$

< 3,6 Mbit/s

Zu e) spektrale Effizienz gleich Datenrate pro Frequenzbereich

$\varepsilon = 3,8 \text{ Mbit/s} / 600 \text{ kHz}$

= 6 bit/s / Hz

Zu f) DR <  $\text{Id}(1 + \text{SNR}) * B$

<  $\text{Id}(1 + 10\text{mW}/0,1\text{mW}) * 20\text{kHz}$

<  $\text{Id}(101) * 20 \text{ kbit/s}$

<  $6,66 * 20 \text{ kbit/s}$

< 132 kbit/s

Zu g) DR = 9600 bit/s     $\text{Id}(S)=4 \text{ bit/Signal} \rightarrow (9600/4) \text{ Signale/s} = 2400 \text{ Bd}$

DR < Signalinformationsgehalt \* 2 \* Bandbreite

B >  $\text{DR}/(2*\text{Id}(S)) = 9600 \text{ bit/s} / 8 \text{ bit}$

> 1200 Hz

Zu h) DR <  $\text{Id}(1 + \text{SNR}) * B$

B >  $\text{DR} / \text{Id}(1+\text{SNR}) = 9600 \text{ bit/s} / \text{Id}(16) = 9600/4 \text{ Hz}$

> 2400 Hz

### 3. Leistung/Dämpfung

Ein Gerät sendet Signale mit 100 mW Leistung und 1 V Spannungshub über eine Strecke mit einer Dämpfung von 40 dB.

- Wie hoch ist die Empfangsleistung in mW?
- Wie hoch ist die Empfangsspannung?
- Welche Rauschleistung bzw. -spannung ist noch akzeptierbar, wenn eine spektrale Effizienz von 2 bit/s pro Hz erreicht werden soll?

Zu a) Dämpfung =  $10 \lg(\text{Sendeleistung}/\text{Empfangsleistung})$

$$40 = 10 * \lg ( 100\text{mW} / \text{PE} )$$

$$10^4 = 100\text{mW} / \text{PE}$$

$$\text{PE} = 100\text{mW} * 10^{-4} = 10 \mu\text{W}$$

Zu b) Leistung proportional zu  $U^2$

Dämpfung =  $20 \lg(\text{Sendespannung}/\text{Empfangsspannung})$

$$\text{UE} = 1\text{V} * 10^{-2} = 10 \text{ mV}$$

Zu c)  $\text{DR} < \lg(1 + \text{SNR}) * B$

$$\text{DR}/B < \lg(1 + \text{SNR})$$

$$2 \text{ bit/Hz} < \lg(1 + \text{SNR})$$

$$\text{SNR} = 3$$

Rauschen

$$\text{Leistung} < 10 \mu\text{W} / \text{SNR} = 3,3 \mu\text{W}$$

$$\text{Spannung} < 10 \text{ mV} / \sqrt{\text{SNR}} = 5,77 \text{ mV}$$

#### 4. Übertragungsmedien

- a. Welche Übertragungsmedien sind besonders geeignet (Begründung) für
  - lange Übertragungsstrecken und hohe Übertragungsraten
  - kurze Übertragungsstrecken und hohe Übertragungsraten
  - mobile Nutzer
- b. Berechnen Sie die Wellenlänge elektromagnetischer Wellen mit einer Frequenz von 2,4 GHz?
- c. Welche Frequenz hat ein Laserlichtstrahl mit einer Wellenlänge von 400 nm?
- d. Warum sind Daten-Cu-Leitungen verdreht (oder verseilt) ?

Zu a)

Glasfaserkabel	hohe Bandbreite ( $10^{14}$ Hz), geringe Dämpfung
Cu-TP-Kabel	hohe Bandbreite ( $10^9$ Hz), Dämpfung akzeptabel bei kurzer Strecke
Luft (WLAN)	keine Kabelinfrastruktur für Endnutzer erforderlich

Zu b)

Wellengeschwindigkeit = Frequenz \* Wellenlänge

Wellengeschwindigkeit = Lichtgeschwindigkeit ca. 300 000 km/s

Frequenz 2,4 MHz                      →      Wellenlänge 12,5 cm

Zu c)

Wellenlänge 400 nm                      →      Frequenz  $7,5 * 10^{14}$  Hz      (violette Licht)

Zu d) Störfelder induzieren Spannungen in Cu-Leitungen → Störströme  
 Durch Verdrehung Störströme abschnittsweise entgegengerichtet → (fast) Aufhebung

## 5. Multiplextechniken

- a. Wie funktioniert Echokompensation?
- b. Vergleichen Sie Frequenz- und Wellenlängenmultiplex?

Zu a)

Geeignet nur für 2-Punkt-Verbindungen  
 Vollduplexübertragung über einen Kanal  
 Hardware muß Pegel auf Übertragungsmedium messen können

Prinzip: Medienpegel messen („Summe“ von Eigen- und Partnersignal)  
 $\text{Partnersignal} = \text{Medienpegel} - \text{Eigensignal}$

Problem: Partnersignal ist gedämpft  $\rightarrow$  Subtraktion ergibt falsche Werte  
 $\text{Partnersignal} = \text{Medienpegel} - (\text{Korrekturfaktor} \cdot \text{Eigensignal})$

Bestimmung Korrekturfaktor nicht einfach  
 Verringerte Genauigkeit

Zu b)

Physikalisch vergleichbar

Bezeichnungstradition in	E-Technik	Frequenz/Hz
	Optik	Wellenlänge/m

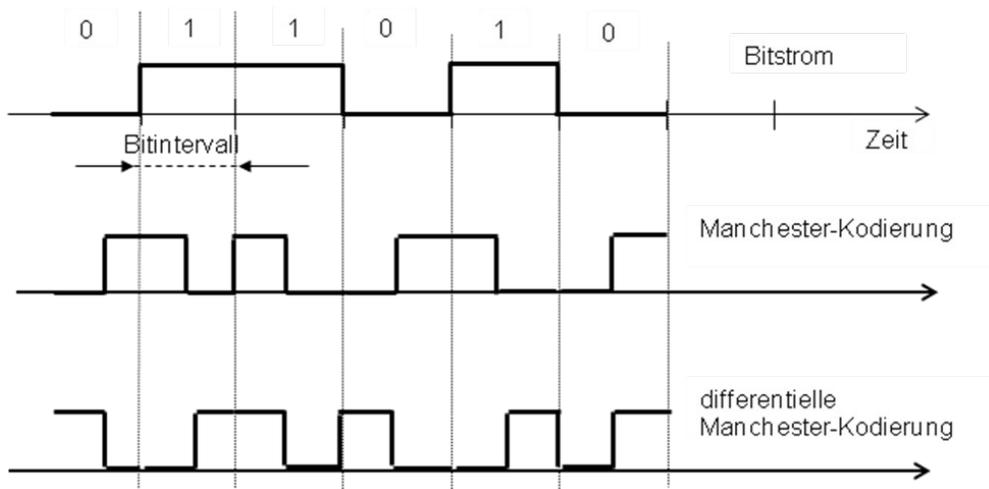
E-Quellen Frequenz und Intensität zeitlich veränderbar

Lichtquellen i.a. Frequenz(Wellenlänge) fest, nur Intensität zeitlich veränderbar

## 6. Basisbandübertragung

- Stellen Sie die Bitfolge "011010" im Manchester-Kode dar.
- Stellen Sie die Bitfolge "011010" im MLT-3-Kode dar.

Zu a)



Zu b)



## 7. Modulation

- Wieviel Information trägt eine reine Sinuswelle?
- Stellen Sie die Übertragung der Bitfolge "011010" dar mit Amplituden-, Frequenz- und Phasenmodulation!

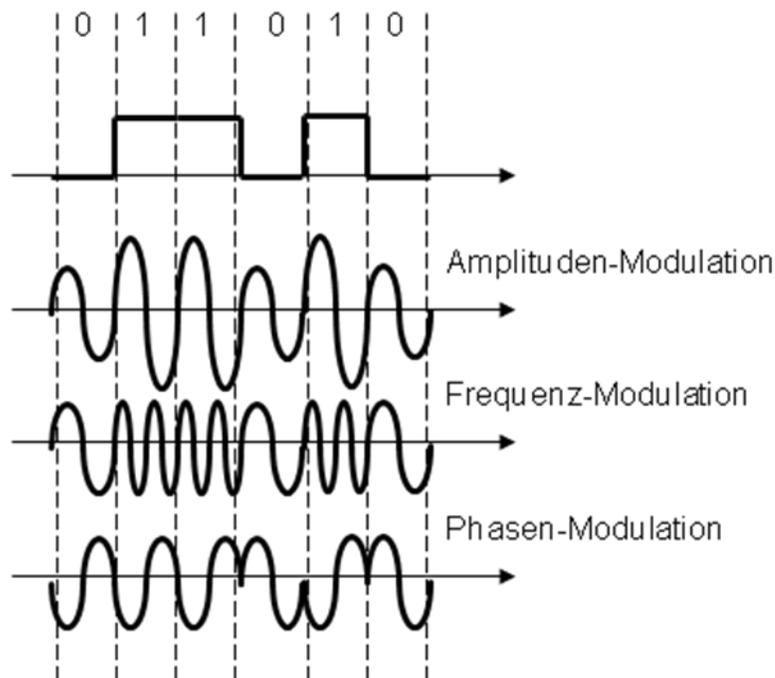
Zu a) Welle  $y = A \cdot \sin(\omega t - x - \phi)$

Konstanten	A	Amplitude
	$\omega$	(Kreis-)frequenz
	$\phi$	Phasenwinkel

Zeitverhalten für alle Ewigkeit vorhersagbar  $\rightarrow$  keine Information für Empfänger

Amplitudenmodulation	A	$\rightarrow$	A(t), gesteuert von Informationsinhalt
	$\omega$	$\rightarrow$	$\omega(t)$ ...
	$\phi$	$\rightarrow$	$\phi(t)$ ...

Zu b)



## 8. Bandspreizung

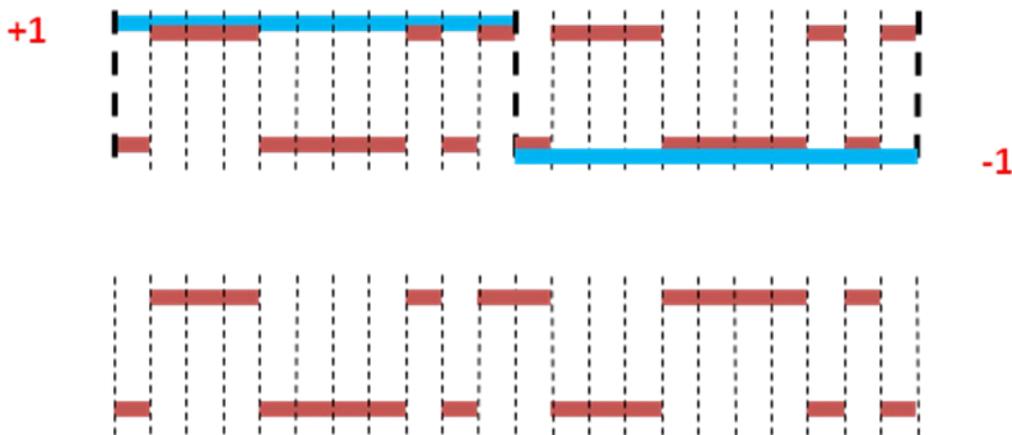
- Welche Vorteile bietet das Prinzip des Spreizens und Entspreizens eines Spektrums?
- Erläutern Sie das Prinzip des „Direct Sequence Spread Spectrum“ (DSSS).
- Bei welchen Standards wird DSSS genutzt?
- Erläutern Sie das Prinzip des „Frequency Hopping Spread Spectrum“ (FHSS).
- Bei welchen Standards wird FHSS genutzt?

Zu a) Vorteile

Robustheit gegen schmalbandige Störungen  
 Robustheit gegen Reflexionen  
 Evtl. Abhörsicherheit

Zu b) Multiplikation mit Spreizkode,  
 z.B.

-1 +1 +1 +1 -1 -1 -1 -1 +1 -1 +1



Senden „Chip“-Folge

Entspreizung durch erneute Multiplikation mit Spreizkode

Zu c) Anwendung z.B. bei CDMA

Zu d) Nutzsinal wird auf sprunghaft wechselnde Trägerfrequenzen moduliert

- Frequenzwechsel in vorbestimmter Reihenfolge (pseudozufällig)
- Vereinbarung der Reihenfolge bei Verbindungsaufnahme

Zu e) Nutzung z.B. bei WLAN IEEE 802.11

## 9. Pulsmodulation

Ein analoges Audio-Signal soll mittels PCM digitalisiert und binär kodiert werden. Zur Amplitudendiskretisierung dient eine lineare Quantisierungskennlinie mit 1023 Intervallen.

- Auf welche obere Grenzfrequenz ist das analoge Signal zu begrenzen, wenn zur Übertragung des digitalen Signals nur eine Kanalbandbreite von 60 kHz zur Verfügung steht?
- Wie groß ist (näherungsweise) der maximale (absolute) Quantisierungsfehler, wenn die maximale Signalamplitude 5 V beträgt?
- Wie kann der relative (auf den Signalwert bezogene) Quantisierungsfehler reduziert werden?

Zu a)

<b>Nyquist</b>	DR	$< 2 B \lg(S)$ $< 2 * 60 \text{ kHz} * 1 \text{ bit}$ $< 120 \text{ kbit/s}$	S=2, weil binäre Kodierung
----------------	----	--	----------------------------

1023 Intervalle	→	10 bit zur Wertdarstellung erforderlich
-----------------	---	---

Abtastrate	AR	$< DR/10 = 12000 \text{ Abtastwerte/s}$
------------	----	---

<b>Shannon</b>	$f_A$	$> 2 * f_G$
	$f_G$	$< f_A/2$ $< 12 \text{ kHz} / 2 = 6 \text{ kHz}$

Zu b) 1023 Intervalle im Bereich von -5V ... + 5 V  
Lineare Quantisierung ca. 10 mV Quantisierungsabstand (10 V / 1000)

5 mV max. Quantisierungsfehler

Zu c) rel. Fehler bei kleinen Quantisierungswerten sehr hoch

Lösung logarithmische Teilung der Quantisierungswerte  
rel. Fehler immer gleich  
bessere Wiedergabequalität

## Übung 2 - Strukturierte Verkabelung

### 1. Standardisierung

- a. Warum ist die Bedarfs- gegenüber der anwendungsneutralen Verkabelung nachteilig?
- b. Welche Lebensdauer hat die Verkabelungsinfrastruktur im Vergleich zu Geräten?
- c. Welchen Anteil an den Gesamtkosten hat die Verkabelung im Vergleich zu Geräten?
- d. Was besagt die Norm EN 50173?
- e. Welche weiteren Spezifikationen zu strukturierter Verkabelung existieren?
- f. Was versteht man unter einem „Link“ und einem „Channel“ ?
- g. Welche Anforderungen muß ein Netzwerk nach Klasse D erfüllen?

Zu a) Verkabelung muß zusammen mit aktiven Komponenten ausgetauscht werden  
Umzüge usw. erfordern Veränderung der Infrastruktur  
Netzausfall bei Erneuerung  
Unübersichtliche Topologien  
Schwieriges Management

Zu b) Verkabelungsinfrastruktur    Nutzungsdauer    10 ... 20 Jahre  
aktiven Komponenten            Abschreibungsdauer    3 ... 5 Jahre

Zu c) Verkabelungskosten sehr hoch wegen Baumaßnahmen  
Nur vertretbar bei sehr langen Nutzungsdauern

Zu d) EN 50173, ...

Baumförmige strukturierte Verkabelung

3 Ebenen	primär	- Geländeverteiler
	Sekundär	- Gebäudeverteiler
	Tertiär	- Etagenverteiler

Qualitätsklassen (bzw. -kategorien)

Bandbreite, Dämpfung, Kabelarten, Steckübergänge, Längenrestriktionen, ...

Meßverfahren, Zertifizierungsvorschriften

Zu e)	EN 50310	Gebäudemaßnahmen: Erdung, Potentialausgleich, ...
	EN 50173	Planung der strukturierten Verkabelung
	EN 50174-1	Spezifikation/Qualitätssicherung
	EN 50174-2	Installation in Bürogebäuden
	EN 50174-3	Installation im industriellen Bereich
	EN 50174-4	Installation in Wohnungen
	EN 50174-5	Installation in Rechenzentren
	EN 50288-X	Kabellnormen
	EN 60603-7-X	Steckverbinder (RJ-45, ...)
	EN 50346	Prüfvorschriften für installierte Verkabelung

- Zu f) Link            feste Verbindung  
                          zwischen Patchfeld im Verteilerraum und Anschlußdose  
                          (evtl. dazwischen ein Consolidation Point)
- Channel            Verbindung zwischen Switchport und Computernetzkarte  
                          Schließt flexible Verbindungskabel mit ein
- Zu g) EN 50173 Klasse D            entspricht etwa            EIA/TIA Kategorie 5
- 100 MHz            Bandbreite  
                          24 dB/100m        max. zulässige Dämpfung bei 100 MHz  
                          ...  
                          Geeignet für Fast Ethernet, Gigabit-Ethernet

## 2. Mediennutzung

- Vergleichen Sie die Eigenschaften von Lichtwellenleitern und TP-Kabeln.
- Ordnen Sie den folgenden Einsatzfällen sinnvolle Übertragungsmedien zu
  - 10 Gbit/s, 40 km
  - 100 Mbit/s, 80 m
  - 100 Mbit/s, 200 m
- Berechnen Sie die Signallaufzeit für ein TP-Kabel mit 50 m Länge und einem NVP-Wert von 0,66.
- In welchen Grenzen schwankt die Signallaufzeit, wenn herstellerbedingt der NVP-Wert um 10% schwankt.
- Wieviel Bitzeiten entspricht diese Schwankung bei 10 Gbit/s Übertragungsrate?

Zu a) TP-Kabel      preiswert: Kabel, Anschlußdosen, Montagearbeit  
 Günstige Verlegung, robust, kleine Biegeradien, ...  
 kurze Strecken - Bandbreite im GHz-Bereich, geeignet bis 10 GbEthernet

LWL                hohe Kosten für Kabel, Anschlußdosen, Montagearbeit  
 Verlegung aufwendig, Biegeradien, Zugbelastung, ...  
 Sehr geringe Dämpfung  
 Geeignet für höchste Übertragungsraten

Zu b)

- LWL
- UTP Kategorie 5
- LWL Multimode

Zu c) TL      = Länge-Kabel / (NVP \* Lichtgeschwindigkeit)

$$\begin{aligned} TL &= 50 \text{ m} / (0,66 * 300000 \text{ km/s}) \\ &= 50 \text{ m} / 200000 \text{ km/s} \\ &= 0,25 \text{ } \mu\text{s} \end{aligned}$$

Zu d) Annahme Fehler für Kabellänge vernachlässigbar, dann

10% Fehler für Signallaufzeit

$$TL = 0,25 \text{ } \mu\text{s} \quad \text{mit Genauigkeit } 25 \text{ ns}$$

Zu e) Bitzeit            = 1 / Datenrate                      Bitanzahl = Zeitschlitz / Bitzeit

$$\text{Bitzeit} = 10^{-10} \text{ s}$$

$$\begin{aligned} \text{Bitanzahl} &= 25 * 10^{-9} \text{ s} / 10^{-10} \text{ s} \\ &= 250 \text{ bit} \end{aligned}$$

### 3. Dimensionierung/Messung

- a. Welche Empfangsleistung ergibt sich bei einer Übertragungstrecke (50 m) mit einer Dämpfung von 18 dB/100m bei einer Sendeleistung von 10mW?
- b. Welche der nachfolgenden Größen ist längenabhängig
  - NEXT
  - FEXT
  - ELFEXT
- c. Wann tritt Alien NEXT auf und wie kann es verringert werden?

Zu a) 18 dB/100m → 9 dB auf 50 m

Dämpfung =  $10 \lg(\text{Sendeleistung}/\text{Empfangsleistung})$

$$9 \text{ dB} = 10 \lg(\text{PS} / \text{PE}) \quad \text{PS/PE} = 10^{0,9} = 7,94$$

$$\begin{aligned} \text{PE} &= \text{PS} / 7,94 = 10 \text{ mW} / 7,94 \\ &= 1,26 \text{ mW} \end{aligned}$$

Zu b) NEXT - längenunabhängig  
 FEXT - abhängig  
 ELFEXT - (relativ) unabhängig

Zu c) Alien NEXT Übersprechen zwischene Nachbarkabeln  
 Problem bei hohen Frequenzen bei UTP

Abstände erhöhen  
 Bessere Schirmung ( S/FTP bringt 100 dB !!!)  
 Installation Klasse F

### 4. Wärmeableitung

- a. Beurteilen Sie UTP und S/FTP-Kabel bezüglich der Wärmefestigkeit.
- b. In Netzen mit strukturierter Verkabelung kann auch die Stromversorgung über die TP-Kabel realisiert werden. Die Speisespannung sei 45 V, die Leistung max. 15 W. Berechnen Sie, wieviel Wärmeenergie bei 50 Kabeln mit je 90 m Länge und einem Schleifenwiderstand von 167  $\Omega$ /km erzeugt wird.

Zu a) UTP -schlechtere Wärmeableitung

$$\begin{aligned} \text{Zu b) } I &= P / U & I &= 15 \text{ W} / 45 \text{ V} & &= 0,33 \text{ A} \\ R &= 167 \Omega/\text{km} * 90 \text{ m} & &= 15 \Omega \\ \text{Wärme/Kabel} & 15 \Omega * (0,33 \text{ A})^2 & &= 1,63 \text{ W} \\ \text{Insgesamt} & 1,63 \text{ W} * 50 \text{ Kabel} & &= 81,7 \text{ W} \end{aligned}$$

## 5. Lichtwellenleiter

- Vergleichen Sie die Eigenschaften von Multimode- und Monomode-LWL
- Nach welcher Streckenlänge (Faser mit Kategorie OS1; Wellenlänge von 1310 nm) darf die Sendeenergie frühestens auf 10% sinken ?
- Für welche Ethernetvarianten darf eine OM2e-Faser über eine Strecke bis 2000 m eingesetzt werden?

Zu a)

Multimode-LWL	mehrere Strahlen (Moden) mit unterschiedlichen Weglängen Modendispersion – unterschiedliche Laufzeit- Impulsverbreiterung Preiswert: Montage, LED-Strahler möglich, ...
Monomode-LWL	nur 1 Strahl in optischer Achse, keine Modendispersion Teuer: Komplizierte Montage, Laserdiode erforderlich, ... Geeignet für sehr lange Ü-Strecken (70 km ...)

Zu b)

Tabelle	$d$	$= 1,0 \text{ dB/km}$	
Formel	$D$	$= d \cdot l = 10 \cdot \lg(\text{PS/PE})$	
	$l$	$= 10 \cdot \lg(\text{PS/PE}) / d$	mit PS/PE=10
		$= 10 \text{ km}$	

Zu c)

Tabelle	100Base-FX 1000Base-LX
---------	---------------------------

## Übung 3 – Ausbreitungsmodelle

### 1. Ausbreitung/Freiraumdämpfung

- a. Wie hoch ist die Signallaufzeit im freien Raum über eine Distanz von
- 30 m typisch für WLAN
  - 3 km WiMAX
  - 36000 km geostationäre Satelliten
  - 227,1 Mill. km mittlere Entfernung Erde-Mars
- b. Wie verändert sich die Freiraumdämpfung (in % und in dB) bei einer Verdoppelung des Abstandes zwischen Sender und Empfänger?
- c. Stellen Sie die Gleichung für den Pfadverlust bei Freiraumdämpfung  $L[\text{dB}] = 32,44 + 20 \lg(f/\text{MHz}) + 20 \lg(R/\text{km})$  so um, daß Frequenzen in GHz und Abstände in m angegeben werden.

In einem Unternehmen existiert ein WLAN mit der Sendefrequenz 2,4 GHz. Ein Access Point ist im Außenbereich installiert und hat eine Sendeleistung  $P_{\text{TX}} = 30 \text{ mW}$ .

- d. Wie groß ist die Empfangsleistung  $P_{\text{RX}}$  in 35 m Abstand?  
(Wenden Sie das Modell der Freiraumdämpfung an.)
- e. Ab welchem Abstand ist der Empfang praktisch nicht mehr möglich?  
(Empfangsleistung unter  $10^{-10} \text{ W}$ )

Zu a)	30 m	$l/c = 30/(3 \cdot 10^8 \text{ s})$	= 0,1 $\mu\text{s}$
	3 km	...	= 10 $\mu\text{s}$
	36000 km	...	= 120 ms
	227,1 Mill. km	...	= 757 s

Zu b)

$$F = P_S/P_E = (4\pi R/\lambda)^2$$

Verdoppelung von R	Faktor 4	400%
--------------------	----------	------

$$L = 32,44 + 20 \lg(f/\text{MHz}) + 20 \lg(R/\text{km})$$

Verdoppelung von R	+ 20lg(2)	6,0 dB
--------------------	-----------	--------

Zu c)

$$1 \text{ MHz} = 1 \text{ GHz} / 1000 \qquad 1 \text{ km} = 1000 \text{ m}$$

$$L[\text{dB}] = 32,44 + 20 \lg(1000 \cdot f/\text{GHz}) + 20 \lg(0,001 \cdot R/\text{m})$$

$$L[\text{dB}] = 32,44 + 20 \lg(f/\text{GHz}) + 60 + 20 \lg(R/\text{m}) - 60$$

$$= 32,44 + 20 \lg(f/\text{GHz}) + 20 \lg(R/\text{m})$$

Zu d)  $F = P_{tx}/P_{rx} = (4\pi R/\lambda)^2$

$$P_{rx} = P_{tx} / (4\pi R * f/c)^2 = 30 \text{ mW} / (4\pi * 35 \text{ m} * 2,4 * 10^9 \text{ s}^{-1} / (3 * 10^8 \text{ m/s}))^2$$

$$= 30 \text{ mW} / (3518,6)^2$$

$$= 2,4 * 10^{-9} \text{ W} = 0,24 \text{ } \mu\text{W}$$

Zu e)  $F = P_{tx}/P_{rx} = (4\pi R/\lambda)^2$

$$R = (c / 4\pi * f) * \text{sqrt}(P_{tx}/P_{rx})$$

$$= (3 * 10^8 \text{ m/s} / (4\pi * 2,4 * 10^9 \text{ s}^{-1})) * \text{sqrt}(0,03/10^{-10})$$

$$\text{sqrt}(0,03/(10^{-10})) = 1,73 * 10^4$$

$$(3 * 10^8 \text{ m/s} / (4\pi * 2,4 * 10^9 \text{ s}^{-1})) = 0,00995$$

$$R = 0,00995 * 1,73 * 10^4 \text{ m}$$

$$= 172 \text{ m}$$

## 2. EIRP

Für WLAN 802.11b/g ist eine max. EIRP-Leistung von 100 mW gestattet.

- a. Wie hoch darf die max. Sendeleistung  $P_{TX}$  sein, wenn eine Sendeantenne mit einem Gewinn von 12 dBi eingesetzt wird?

Zu a)

$10 \lg(100 \text{ mW}/1 \text{ mW})$	→ maximale Sendeleistung	20 dBm
Antennengewinn (12 dBi) abziehen	→ zulässige Leistung (EIRP) bzw. $10^{0,8} \text{ W}$	8 dBm 6,3 mW

### 3. Fresnelzone

Zwei Richtfunk-Stationen haben einen Abstand von 10 km. In der direkten Verbindungslinie befinden sich Gebäude mit einer Maximalhöhe von 20 m.

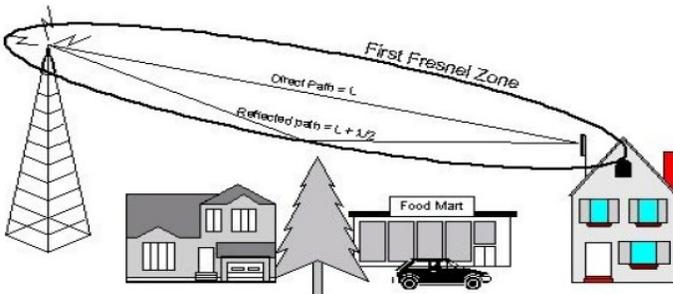
Die Sendefrequenz beträgt 6 GHz.

Die 1. Fresnelzone darf nur zu max. 20% durchdrungen werden.

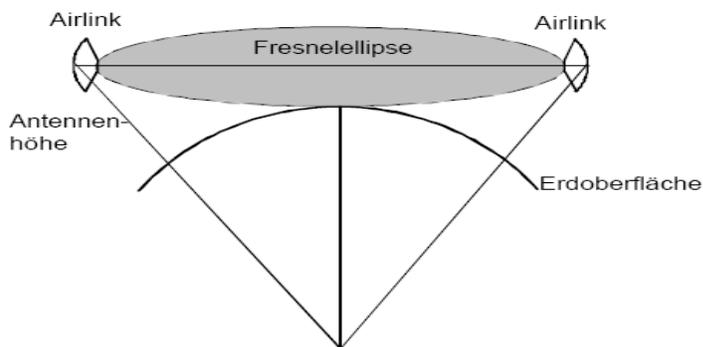
- Berechnen Sie die minimal erforderliche Antennenhöhe der Stationen.
- Welche Antennenhöhen sind für einen Abstand von 50 km erforderlich?

Zu a) 33,1 m

Zu b) 94,0 m



[Bild: Fresnel-Zone]  
Fresnelellipse



Bei 18 KM Reichweite, müssen die Antennen schon eine Höhe von 28 Metern haben =>

Clearance Modell

Praktisch immer Leistungseinbußen oder Verbindung über Täler (LOS)

Faktoren:

Abschirmung durch Erdkrümmung

Hindernisse in der Fresnelzone

10dB Singal - Rauschabstand

## Übung 4 - Sicherungsschicht

### 1. Fehlerwahrscheinlichkeit

Ein Computer sendet Frames mit einer durchschnittlichen Größe von 1000 Byte und einer Datenrate von 100 Mbit/s. Die Bitfehlerwahrscheinlichkeit beträgt  $10^{-8}$ . Zusätzlich gibt es im durchschnittlichen Zeitabstand  $D$  von 10 min burstartige Störungen mit einer Bitfehlerwahrscheinlichkeit von  $10^{-3}$  und 1 ms Dauer  $d$ .

- Berechnen Sie den Einfluß beider Störarten.
- Welcher Störeinfluß dominiert?

Zu a) Bei konstanter Bitfehlerwahrscheinlichkeit gilt für  $f_{\text{Bit}} \ll 1$  und für  $f_{\text{Bit}} \ll 1/n_{\text{Block}}$

$$\begin{aligned} f_{\text{Block}} &= n_{\text{Block}} * f_{\text{Bit}} \\ &= 8000 \text{ bit} * 10^{-8} / \text{bit} \\ &= 8 * 10^{-5} \end{aligned}$$

Für die burstartige Störung ist obige Formel nicht anwendbar

$$\begin{aligned} f_{\text{Block}} &= 0 \quad \text{falls Block nicht im Burstzeitraum gesendet} \\ f_{\text{Block}} &\approx 1 \quad \text{falls Block im Burstzeitraum gesendet} \\ f_{\text{Block}} &= 0 * (D-d)/D + 1 * d/D \\ &= d/D = 10^{-3} / 600 \\ &= 1,7 * 10^{-6} \end{aligned}$$

Zu b) Der Einfluß der konstanten Bitfehlerwahrscheinlichkeit dominiert gegenüber den burstartigen Störeinflüssen.

## 2. Fehlererkennung durch Paritätskontrolle

Der ASCII-Basiskode ist ein Zeichendarstellungskode für Klein- und Großbuchstaben des englischen Alphabets, für Ziffern, für Sonder- und für Steuerzeichen

Auszug

A	B	...	O	P	Q	...	Z
0x41	0x42		0x4F	0x50	0x51		0x5A

- Notieren Sie zeichenweise untereinander die Zeichenkette „ABCDPQRS“ in binärer 7-bit-Darstellung.
- Fügen Sie ein Kontrollbit für gerade Parität hinzu.
- Welche Bitfehlersituationen können erkannt und welche korrigiert werden?
- Wie hoch ist bei einer Bitfehlerwahrscheinlichkeit von  $10^{-3}$  die Wahrscheinlichkeit eines fehlerhaften Zeichens und wie hoch ist die Wahrscheinlichkeit, daß der Fehler nicht erkannt wird?
- Fügen Sie ein Kontrollzeichen für gerade Blockparität hinzu.
- Welche Bitfehlersituationen können erkannt und welche korrigiert werden?
- Für welche Anwendungen wäre ein solcher Kode geeignet?

Zu a) b) e)

1000001	0
1000010	0
1000011	1
1000100	0
1010000	0
1010001	1
1010010	1
1010011	0
0000100	1

Zu c)

- 1 Bitfehler pro Zeichen kann erkannt werden (gilt auch für das Paritätsbit)
- Korrektur ist nicht möglich
- Eine gerade Bitfehleranzahl pro Zeichen wird nicht erkannt

Zu d)

$$\begin{aligned} \text{Fehlerwahrscheinlichkeit}_{\text{Zeichen}} &\approx \text{Bitanzahl}_{\text{Zeichen}} * \text{Fehlerwahrscheinlichkeit}_{\text{Bit}} \\ &= 8 * 10^{-3} \end{aligned}$$

2 Bitfehler werden nicht erkannt

$$\text{Wahrscheinlichkeit} = 6,4 * 10^{-5}$$

Zu f)

- 1 Bitfehler pro Zeichen kann erkannt und korrigiert werden
- Bei mehreren Bitfehlern im Block Fehlererkennung und-Korrektur u.U. nicht möglich
- Eine gerade Bitfehleranzahl pro Zeichen bzw. Bitposition wird nicht erkannt

Zu g)

Die meisten Fehler können beim Empfänger erkannt und korrigiert werden.

Nutzung sinnvoll bei Echtzeitanwendungen mit Akzeptanz einer gewissen Restfehlerrate

### 3. Fehlererkennung durch CRC

Prüfen Sie folgende empfangene Bitfolgen auf Fehlerfreiheit (Generatorpolynom  $x^3 + x + 1$ ).

d. 110110101

e. 100110011

Zu a) 110110101 : 1011

1011

----

1101

1011

----

1100

1011

----

1111

1011

----

1000

1011

----

111

Rest ungleich Null

Zu b) 100110011 : 1011

1011

----

1010

1011

----

1011

1011

----

000

Rest gleich Null

#### 4. Fenstersteuerung I

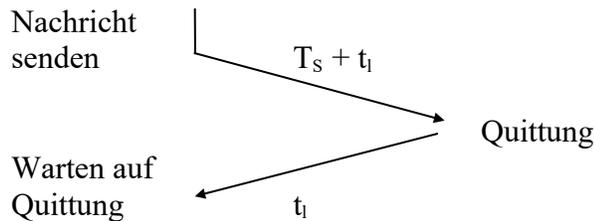
Ein Kanal habe eine Datenrate von 40 kbit/s und eine phys. Signallaufzeit von 20ms.

- f. Für welche Rahmengrößen ergibt sich für das Stop-and-Wait-Protokoll eine Effizienz von mindestens 50%? Die Quittungsgröße sei vernachlässigbar.
- g. Wie verändert sich das Ergebnis bei Nutzung eines Schiebefensterprotokolls mit Fenstergröße 4 ?

Zu a)

Datenrate DR = 40 kbit/s       $t_1 = 20 \text{ ms}$        $w > 0,5$       Rahmengröße F

Sendezeit       $T_s = F / DR$



$$w = \text{Kanalbelegungszeit} / (\text{Kanalbelegungszeit} + \text{Wartezeit})$$

$$= (F/DR) / (F/DR + 2 * t_1)$$

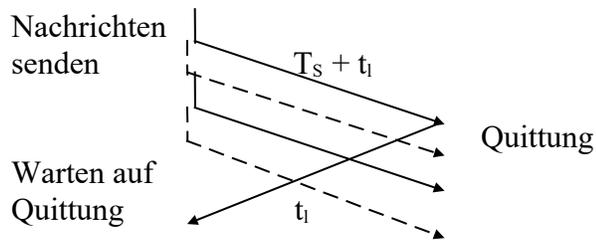
umgestellt nach F

$$F = DR * \frac{2 * t_1 * w}{(1 - w)} = 1600 \text{ bit}$$

Die minimale Rahmengröße muß 1600 bit betragen.

Zu b)

In der gleichen Wartezeit können 4 Rahmen gesendet werden.



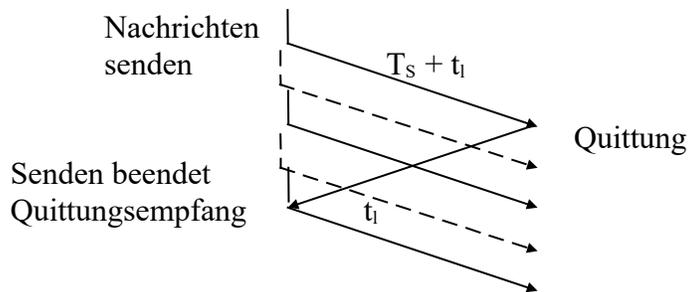
$$w = (4 F/DR) / (F/DR + 2 * t_l)$$

$$F = DR * \frac{2 * t_l * w}{(4 - w)} = 229 \text{ bit}$$

Die minimale Rahmengröße muß 229 bit betragen.

Zu c)

n Rahmen senden bis Quittung auf 1. Rahmen eintrifft



$$n * T_s = T_s + 2 * t_l$$

$$T_s = F/DR = 800 \text{ bit} / 40 \text{ kbit/s} = 20 \text{ ms}$$

$$n = (T_s + 2 * t_l) / T_s$$

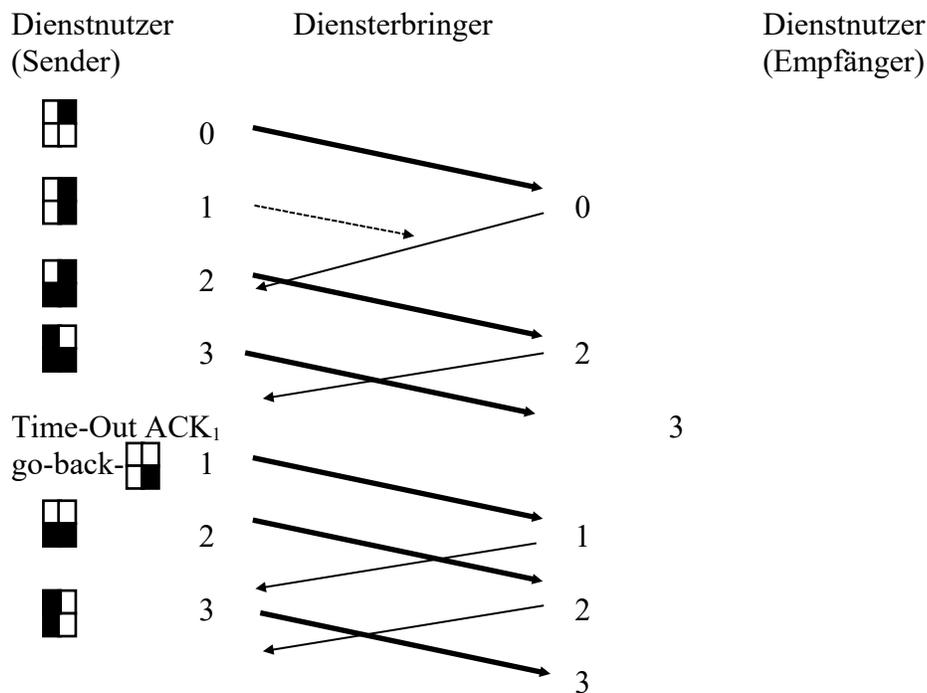
$$= 60 \text{ ms} / 20 \text{ ms} = 3$$

Ab Fenstergröße 3 wird eine 100%-ige Auslastung erreicht.

## Fenstersteuerung II

- Skizzieren Sie ein Weg-Zeit-Diagramm für die Datenübertragung mit einem Schiebefensterprotokoll der Fenstergröße 3, wobei die Fehlerbehandlung durch go-back-n realisiert wird und das zweite gesendete Frame zunächst verloren geht, aber nach Wiederholung erfolgreich übertragen wird.
- Skizzieren Sie für jede Übertragung den Zustand des Sendefensters.
- Weshalb reicht es bei einem Schiebefensterprotokoll der Fenstergröße  $F$  nicht aus, die Frame-Laufnummern  $0 \dots (f-1)$  zu vergeben ?
- Erläutern Sie, wie die Bestätigung bei HDLC bei Duplex-Datenübertragung versendet werden.

Zu a) und b)



Zu c) Es kann zu Verwechslungen beim Empfänger kommen. Würde im obigen Beispiel der Frame 3 die Nummer 0 erhalten, dann würde der Empfänger nicht wissen, ob ein neuer Frame vorliegt oder ein go-back-0. Ist die max. Laufnummer größer als die Fenstergröße, kann der Empfänger ein go-back-n immer erkennen (Reihenfolgeverletzung).

Zu d) In HDLC-Frames befindet sich ein Control-Feld (8 bit), welches jeweils 3 bit für einen Empfangs- und einen Sendezähler enthält (8 Laufnummern; Fenstergröße 7).

Huckepack-Verfahren, d.h. bei Duplexübertragung sind meist keine expliziten Quittungsnachrichten erforderlich.

## Übung 5 – Layer-2-Netzwerktechnologien

### 1. ALOHA

Das ALOHA-Verfahren realisiert eine sehr einfache Medienzugriffssteuerung.

- Erläutern Sie, warum das ALOHA-Verfahren nur bei geringen Netzauslastungen befriedigend funktioniert.
- Erläutern Sie, warum Slotted-ALOHA einen besseren Durchsatz erreicht als das einfache ALOHA-Verfahren

Zu a)

Das ALOHA-Verfahren verhindert nicht, daß Nachrichtenkollisionen auftreten.

Nachrichtensendedauer:  $T = \text{Framelänge} / \text{Datenrate}$

Das Überlappungsintervall ist bei ALOHA doppelt so groß  
(mögliche Kollision bei P-ALOHA letztes Bit Frame 1 / erstes Bit Frame2)

z.B.	Sender 1	DR= 10 Mbit/s und F= 1000 bit bei isochron 1000 Frames/s	→ T = 0,1ms 10 % Auslastung
	Sender 2	DR= 10 Mbit/s und F=1000 bit 1 Frame	→ 20 % Kollisionswahrscheinlichkeit

Bei hohen Netzauslastungen ist es fast unmöglich, daß eine Nachricht ohne Kollision übertragen werden kann.

Zu b)

Beim S-ALOHA gibt es vorgegebene Sendezeitpunkte (Zeitslots) mit dem Zeitabstand T

$T = \text{max. Framelänge} / \text{Datenrate}$

Kollision bei S-ALOHA nur beim ersten Bit beider Frames möglich)

Kollisionswahrscheinlichkeit im obigen Beispiel nur noch 10% !

## 2. CSMA

CSMA-Verfahren reduzieren die Kollisionswahrscheinlichkeit.

- Erläutern Sie, warum bei p-persistent CSMA für kleinere Werte von p ein günstigeres Verhalten im Hochlastfall erreicht wird.
- Weshalb darf die Framelänge bei Ethernet einen bestimmten Wert nicht unterschreiten ?
- Wie groß ist dieser bei IEEE 802.3 ?
- Wie groß müßten die Frames mindestens sein bei einem CSMA/CD-Netz der Länge 200 m, einer Datenrate von 1 Gbit/s und einer Signalausbreitungsgeschwindigkeit von 200000 km/s ?
- Ist das Ergebnis von d) vereinbar mit dem Frameformat von IEEE 802.3 ?

Zu a)

Extremfall  $p=1$  ist unnachgiebiges CSMA,  
nach einer Kollision sofortige Wiederholung mit erneuter Kollision,

bei Nachgiebigkeit (Senden nur mit Wahrscheinlichkeit  $p$ )  
sinkt die Folge-Kollisionswahrscheinlichkeit

Zu b)

Sender und Empfänger sollen Kollision erkennen !

Extremfall: 2 Sender an den Enden des Übertragungsmediums (Länge  $l$ )  
Sender 1 sendet zum Zeitpunkt  $t$   
Sender 2 stellt „Kanal besetzt“ erst fest nach Ablauf der Signallaufzeit  $t_l$ ,  
Bis zum Zeitpunkt  $t+t_l$  ist eine Kollision möglich.

Sender 2 stellt immer schnell fest, daß eine Kollision vorliegt,  
Sender 1 aber im Extremfall nur,  
wenn er mindestens die doppelte Signallaufzeit sendet ( $\rightarrow$  Framemindestgröße)

Sendezeit:  $T = F/DR$                       Framegröße/Datenrate

Laufzeit  $t_l = l/v$                       Länge/Ausbreitungsgeschwindigkeit

**Sendezeit > doppelte Signallaufzeit**

$$F > 2 * l * DR / v$$

Zu c)

IEEE 802.3 Festlegung auf

$F > 512 \text{ bit}$  (ohne Präambel)  
 $> 64 \text{ byte}$

Zu d)

Anwendung der Formeln von 2b)

$F > 2 * (200 \text{ m} * 1000 \text{ Mbit/s}) / 200\,000 \text{ km/s}$

$F > 2000 \text{ bit}$   
 $> 250 \text{ byte}$

Zu e)

Im Frameformat von IEEE 802.3 sind kürzere Frames als 250 byte zulässig, d.h. die Kollisionserkennung würde nicht immer funktionieren.

Damit ist CSMA/CD für Gigabit-Ethernet ungeeignet.

### 3. Tokenverfahren

- Vergleichen Sie das Hochlastverhalten von CSMA/CD und Tokenverfahren
- Vergleichen Sie das Echtzeitverhalten von CSMA/CD und Tokenverfahren
- Vergleichen Sie Token-Ring- und Token-Bus-Verfahren?
- Ab welcher Ringlänge passen mehrere Frames zeitgleich auf den Ring bei Datenraten von 100 Mbit/s und Framegrößen von 1000 bit und einer Signalausbreitungsgeschwindigkeit von 200 000 km/s ?
- Berechnen Sie für einen Token-Ring-LAN nach IEEE 802.5 die maximale Wartezeit einer sendewilligen Station, bei 1 km Ringlänge, 200000 km/s Ausbreitungsgeschwindigkeit, 4 Mbit/s Datenrate, 10 Stationen und einer maximalen Framelänge von 4000 Byte!  
Zur Vereinfachung wird angenommen, dass alle Stationen sendewillig sind und nur einen Frame gleicher Priorität senden wollen. Die Stationsverzögerung wird nicht beachtet.

Zu a)

Tokenverfahren realisieren eine deterministische Zugriffssteuerung → keine Kollisionen

100 % Auslastung erreichbar

CSMA/CD kann Kollisionen nicht restlos ausschließen

Max. 70% Auslastung

Zu b)

Tokenverfahren arbeiten mit Sendezeitbegrenzungen (THT - Token Holding Time) für die Teilnehmer

max. Wartezeit  $(n-1) \cdot \text{THT}$   
dh. echtzeittauglich

CSMA/CD kann Kollisionen und Folgekollisionen nicht restlos ausschließen

keine max. Wartezeit angebar (nur Wahrscheinlichkeiten)  
d.h. nicht echtzeittauglich

Zu c)

Token Ring Tokenweitergabe gemäß physikalischer Reihenfolge im Ring  
keine Adressierung erforderlich

Token Bus Tokenweitergabe durch adressierte Nachricht  
jede Station muß Vorgänger- und Nachfolgeradresse kennen  
flexibler Ringverkürzung möglich durch passive Stationen  
passive Stationen werden gepollt (Intervallzeit an Problem anpaßbar)

Zu d)

$$\text{Sendezeit } t_s = F/DR = 1000 \text{ bit} / (10^8 \text{ bit/s}) = 10^{-5} \text{ s}$$

$$\text{Framelänge } l = t_s * v = 10^{-5} \text{ s} * 2 * 10^8 \text{ m/s} = 2000 \text{ m}$$

Zu e)

Geg:  $l = 1.000 \text{ km}$   
 $DR = 4 \text{ Mbit/s}$   
 $n = 10$   
 $v = 200.000 \text{ km/s}$   
 $F = 4000 \text{ Byte}$

Ges: Signalumlaufzeit  $\tau$  (TokenRotation Time TRT)  
 Sendezeit  $t_s$   
 Wartezeit  $t_w$

$$\begin{aligned} t_s &= F / DR \\ &= 4.000 \text{ Byte} / 4 \text{ Mbit/s} \\ &= 8 \text{ ms (Sendezeit für Frame)} \end{aligned}$$

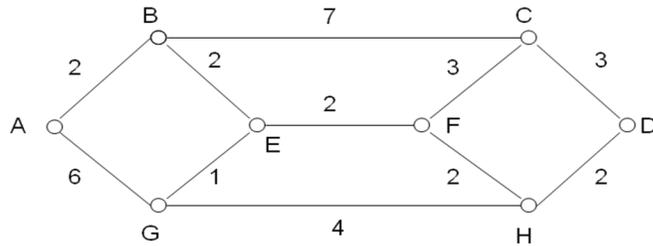
$$\begin{aligned} \tau &= l / v \\ &= 1 \text{ km} / 200.000 \text{ km/s} \\ &= 0,005 \text{ ms} \end{aligned}$$

$$\begin{aligned} t_w &= (\text{Anzahl Stationen} - 1) \cdot (\text{Sendezeit pro Station} + \tau) + \tau \\ &= 9 \cdot 8 \text{ ms} + 10 \cdot 0,005 \text{ ms} \\ &= 72,05 \text{ ms} \end{aligned}$$



## 2. Link State Routing

Gegeben sei ein Netz mit folgender Topologie und Kostenbewertung der Übertragungspfade



- Wovon können die Kostenbewertungen im realen Netz abhängen? Diskutieren Sie drei Parameter!
- Bestimmen Sie schrittweise den kürzesten Pfad von B nach D nach dem Verfahren "Shortest Path Routing" von Dijkstra.

Zu a)

- Datenrate
- Durchschnittliche Auslastung
- Zuverlässigkeit
- Mietkosten
- ...

Zu b)

s. Kap. 8. Animation: Folie 16

### 3. Distance Vector Routing

Gegeben sei das gleiche Netzwerk wie in Aufgabe 2.

Die Router sind in der Lage, die Übertragungskosten zu ihren Nachbarn zu bestimmen.

Im 30-Sekunden-Rhythmus senden sie ihre Distanzvektoren an ihre Nachbarn.

- Wie ändert sich die Routingtabelle des Knotens E, wenn dieser nach einem Systemausfall folgende Distanzvektoren von seinen Nachbarn erhält?  
von B ( 2, 0, 7, 8, 2, 4, 3, 6 ) , von F ( 6, 4, 3, 4, 2, 0, 3, 2 ) und von G ( 5, 3, 6, 6, 1, 3, 0, 4 )
- Skizzieren Sie für alle Router die schrittweise Entwicklung der Tabelleneinträge bezüglich Knoten D vom Initialzustand bis zum stabilen Endzustand.
- Wie ändern sich jetzt die Weiterleitungstabellen nach Ausfall der Knotens C und H in den folgenden 5 Zyklen?

Zu a)

Empfang Distanzvektor vom Nachbarrouter | Abstand zum Nachbarn addieren

Ziel	Tabelle nach Ausfall		Tabelle nach DV von B (+2)		Tabelle nach DV von F (+2)		Tabelle nach DV von G (+1)	
	über	Kosten	über	Kosten	über	Kosten	über	Kosten
A			B	4	B	4	B	4
B			B	2	B	2	B	2
C			B	9	F	5	F	5
D			B	10	F	6	F	6
E	-	0	-	0	-	0	-	0
F			B	6	F	2	F	2
G			B	5	B	5	G	1
H			B	8	F	4	F	4

Zu b)

Tabelle in	Initial-Tabelle		Tabelle nach Austausch 1		Austausch 2		Austausch 3		Austausch 4		Austausch 5	
	nach D über	K	... über	K	... über	K	... über	K	... über	K	... über	K
A							G	12	B	10	B	10
B					C	10	C	10	E	8	E	8
C			D	3	D	3	D	3	D	3	D	3
D	-	0	-	0	-	0	-	0	-	0	-	0
E							F	6	F	6	F	6
F					H	4	H	4	H	4	H	4
G					H	6	H	6	H	6	H	6
H			D	2	D	2	D	2	D	2	D	2

Zu c)

Quelle	Vor Ausfall C und H		danach Austausch 1		Austausch 2		Austausch 3		Austausch 4		Austausch 5	
	... über	K	... über	K	... über	K	... über	K	... über	K	... über	K
A	B	10	B	10	B	10	B	10	B	10	B	14
B	E	8	E	8	E	8	E	10	E	12	E	14
E	F	6	F	6	F	6	F	10	F	12	F	14
F	H	4	?		E	8	E	8	E	12	E	14
G	H	6	?		E	7			E	11	E	13

→ Regelschwingungen

u.U. kein sinnvoller neuer Endzustand

#### 4. IP-Adressierung

Eine Firma besitzt einen IP-Adreßbereich von 128.10.192.0 bis 128.10.199.255

- Wie viele Adressen enthält dieser Bereich? Berechnen Sie die Netzmaske. Der Bereich soll in mehrere Subnetze mit jeweils 30 Hosts aufgeteilt werden. Geben Sie eine geeignete Subnetzmaske an!
- Wie viele Teilnetze mit je 30 Hosts können adressiert werden?
- Teilen Sie unter Berücksichtigung der gewählten Subnetzmaske die IP-Adresse 128.10.192.70 in Netz- und Host-Anteil auf.

Zu a)

128	10	192	0
10000000	00001010	11000000	00000000
128	10	199	255
10000000	00001010	11000111	11111111
			
11 bits			

Adreßanzahl                     $2^{11} = 2048$   
 Maske:                            255.255.248.0

2 Host-Adressen sind in jedem Netz reserviert

00...00            für Netzadresse  
 11...11            für Broadcastadresse

30-er Netz erfordert 32-Adreßraum → 5 bit Hostanteil

Submaske                            255.255.255.224

Zu b)

6 bit übrig für Subnetzadressierung (11 bit – 5 bit)

$2^6 - 2 = 62$  Subnetze möglich (mit je max. 30 Hosts)

Zu c)

128.10.192.70	IP	.....	.....	11000000	01000110
	Maske			11111111	11100000
	Netz	128.10.192.64			
	Host-ID	6			

## 5. IP - Fragmentierung

Ein TCP-Segment mit 2048 Byte Nutzdaten wird an IP zur Auslieferung übergeben.  
Der Übertragungsweg geht über zwei Netzwerke (Quellrechner → Router → Zielrechner).

Netzwerk 1: MTU = 1024 Byte

Netzwerk 2: MTU = 512 Byte

- Geben Sie für die beim Empfänger ankommenden Fragmente jeweils die Größe und den Offset an.
- Wie viele Fragmente würden erzeugt, wenn der Sender wüsste, dass die kleinste MTU auf den Pfad zum Empfänger 512 Byte beträgt?

Zu a)

TCP-Segment (mit 20-byte-Header)			→	2068 byte	
IP-Paket (mit 20-byte Header)				2088 byte (zu groß, wegen MTU)	
Netz1 →	IP-Paket-1	1024 byte		1004 Byte Inhalt	Offset 0
	IP-Paket-2	1024 byte		1004 Byte Inhalt	Offset 1004
	IP-Paket-3	80 byte		60 Byte Inhalt	Offset 2008
Router →	IP-Paket-1-1	512 byte		492 Byte Inhalt	Offset 0
	IP-Paket-1-2	512 byte		492 Byte Inhalt	Offset 492
	IP-Paket-1-3	40 byte		20 Byte Inhalt	Offset 984
	IP-Paket-2-1	512 byte		492 Byte Inhalt	Offset 1004
	IP-Paket-2-2	512 byte		492 Byte Inhalt	Offset 1496
	IP-Paket-2-3	40 byte		20 Byte Inhalt	Offset 1988
	IP-Paket-3-1	80 byte		60 Byte Inhalt	Offset 2008

Zu b)

Größerer Overhead durch kurze Pakete (Headeranteil)  
Wenn kleinste MTU auf Weg bekannt, Verbesserung möglich

5 statt 7 Pakete       $4 \times (20 + 496) \text{ byte} + 1 \times (20 + 100) \text{ byte}$

Außerdem Vermittlungsaufwand für 5 Pakete kleiner als für 7

## 6. AS-Routing

Am 24.2.2008 um 18.47 Uhr startete der Administrator des AS17557 (Pakistan) eine falsche Netzankündigung (Präfix 208.65.152.0/24). Diese wurde über AS3491 (PCCW Global) weltweit verbreitet. Erst 21.01 Uhr wurde der Fehler durch AS17557 korrigiert. Weltweit wurde erst am 25.2. um 01.00 Uhr ein normaler Internetbetrieb wieder erreicht. (<http://www.ripe.net/news/study-youtube-hijacking.html>)

- a. Was kann bei einer solchen Fehlankündigung passieren?

CIDR mit „longest prefix matching“  
Herauslösen eines Teilnetzes aus einem größeren Adreßbereich  
Hijacking bei Fehl eingaben möglich

- b. Finden Sie heraus, welche anderen AS durch die Fehlankündigung betroffen waren. (<http://www.ris.ripe.net/cgi-bin/riswhois.cgi>)

Routing Information Service - Windows Internet Explorer

<http://www.ris.ripe.net/cgi-bin/riswhois.cgi?address=208.65.152.0%2F22&matchtype=Lt>

RIPE NCC | LIR Portal | RIPE

About RIPE NCC | Contact | Search | Sitemap

you are here: [home](#) -> [RIPE NCC Projects](#) -> [RIS](#)

RIS:

- [RIS Home Page](#)
- [RIS Raw Data](#)
- [Documentation](#)
- [Analysis using RIS](#)
- [Contact Us](#)
- [Send Feedback](#)

**RISwhois**

```
[Querying riswhois.ripe.net]
[riswhois.ripe.net]
% This is RIPE NCC's Routing Information Service
% whois gateway to collected BGP Routing Tables
% IPv4 or IPv6 address to origin prefix match
%
% For more information visit http://www.ripe.net/ris/riswhois.html
```

```
route:          208.65.152.0/22
origin:         AS36561
descr:         YOUTUBE - YouTube, Inc.
lastupd-frst:  2010-03-18 20:10Z 202.249.2.20@rrc06
lastupd-last:  2010-10-03 19:53Z 91.103.24.2@rrc00
seen-at:       rrc00, rrc01, rrc03, rrc04, rrc05, rrc06, rrc10, rrc11, rrc12, rrc13, rrc14, rrc15, rrc16
num-rispeers: 96
source:        RISWHOIS
```

rrc00 Query RIS Looking Glass

Address or Prefix

All more specific matches  All less specific matches

Exact match only

Short fast output  Return primary keys only

About RIPE NCC | Site Map | LIR Portal | About RIPE | Contact | © RIPE NCC. All rights reserved.

→ 25% des Netzes von YouTube entführt



## Übung 7 - Transportschicht

### 1. Nutzerschnittstelle TCP/UDP

- a. Welche Aufgabe haben Portnummern?
- b. Welche Dienstqualität bietet UDP?
- c. Welche Dienstqualität bietet TCP?

Zu a)

Identifikation von Nachrichtenübergabestellen (SAP) auf einem Rechner zwischen Anwendungsprozeß und Betriebssystem (TCP/IP)

Zu b)

UDP-Dienstqualität vergleichbar IP  
verbindungslos, keine Fenstersteuerung, ...eigene Fehlersteuerung erforderlich  
Reihenfolgeverluste  
evtl. Datenverluste  
zusätzlich  
Portunterstützung  
Multitasking

vorteilhaft für Multimediaströme

Zu c)

TCP-Dienstqualität bietet zusätzlich zu UDP  
Verbindungsorientierung  
Fehler-, Fluß-, und Staukontrolle

vorteilhaft für Filetransfer, ..., nachteilig für Multimediaströme

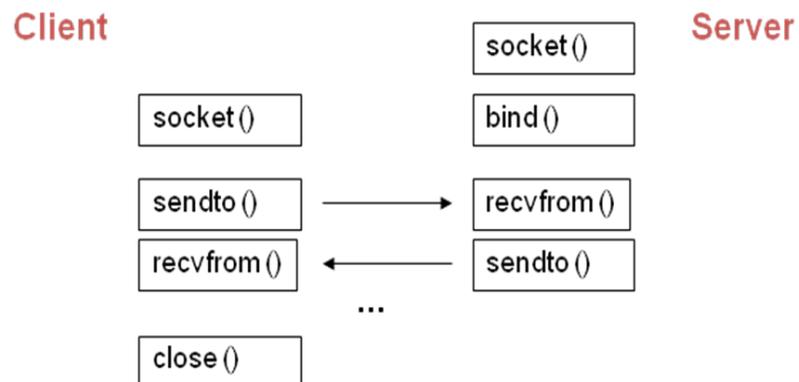
## 2. Socketschnittstelle

Diskutieren Sie die Aufgaben der Socketprozeduren für die Nutzung von

- a. UDP
- b. TCP

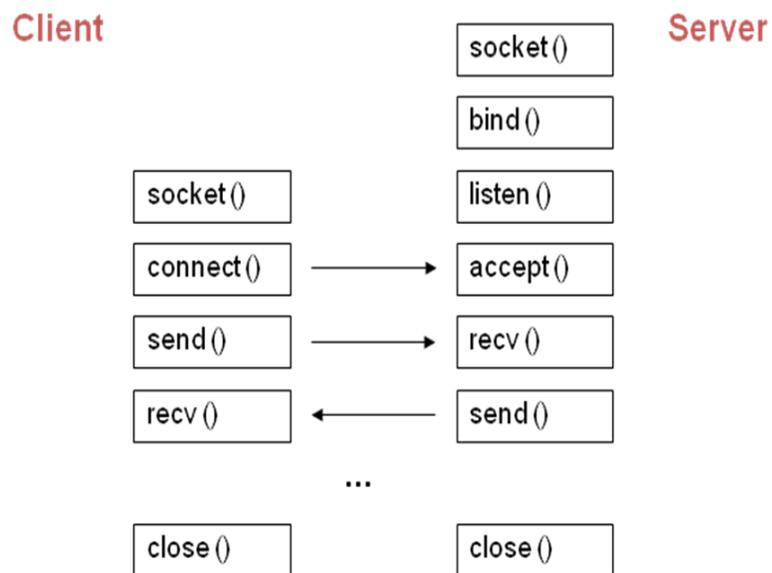
Zu a)

Funktion	Inhalt
socket	Anforderung Kommunikationsendpunkt
bind	Festlegen Portnummer
sendto	Senden
recvfrom	Empfangen
close	Schließen Socket



Zu b)

Funktion	Inhalt
socket	Anforderung Kommunikationsendpunkt
bind	Festlegen Portnummer
listen	Empfangspuffer und Warten auf Verbindungsaufnahme
accept	Annahme Verbindung
connect	Anforderung Verbindung
send	Senden
recv	Empfangen
close	Schließen Socket



### 3. Interna

- Wie erfolgt die Prüfsummenberechnung bei UDP?
- Wie erfolgt der Verbindungsaufbau bei TCP?
- Wie erfolgt der Verbindungsabbau bei TCP?
- Wie erfolgt die Quittierung von TCP-Segmenten?
- Was sind die Ziele der Fluß- und der Staukontrolle?  
Wie werden diese Ziele erreicht?
- Benennen Sie zwei unterschiedliche TCP-Implementationen.  
Können unterschiedliche Endsysteme über eine TCP-Verbindung zusammenarbeiten?

Zu a)

UDP-Prüfsequenz wird berechnet über

- Pseudoheader (Quell- und Ziel-IP, 0x00, Protokoll-Id, Paketlänge)
- UDP-Header (Prüfsequenzfeld mit 0x0000 initialisiert)
- UDP-Daten

**Sender**      **16-Bit-Worte**

	10110011	01011000	
	01001011	00101110	
	10001100	00010011	
	00000000	00000000	
<b>1</b>	10001010	10011001	$\Sigma_s$
	10001010	10011010	Übertrag zuaddieren
	01110101	01100101	Einerkomplement $\rightarrow$ Header

**Empfänger**

	10110011	01011000	
	01001011	00101110	
	10001100	00010011	
	01110101	01100101	
<b>1</b>	11111111	11111110	$\Sigma_E$
	11111111	11111111	Übertrag zuaddieren

Zu b)

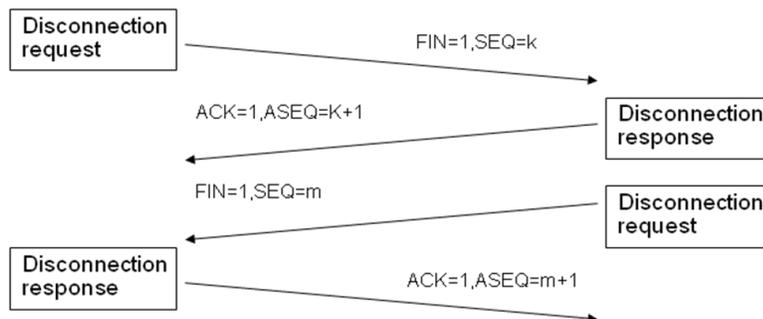
Client kennt Server-Socketadresse  
fordert freien Port vom Betriebssystem  
→ Client-Socketadresse

Server vereinbart mit Betriebssystem  
die Nutzung des Serverports  
wartet auf Nachrichten

Start-Sequenz-Nr.  $x$  wird zufällig gewählt



Zu c)



beide (!) Seiten müssen Verbindung abbauen  
evtl. Zusammenfassung von ACK und FIN zu einer Nachricht bei Partner-2  
(3-Wege-Handshake)

Zu d)

In jeder Nachricht wird die erwartete nächste Byte-Nummer mitgeschickt  
(Piggyback im TCP-Header).

Die Quittungsnummer ist kumulativ,  
d.h. alle logisch vorhergehenden Segmente werden mit quittiert.

Bei Segmentverlusten werden korrekt übertragene spätere Segmente nicht quittiert, sondern nur der Wert des zeitlich davor korrekt übertragenen Segmentes, d.h. es gibt sog. „duplizierte Quittungen“.

Zu e)

- Flußsteuerung - Vermeidung der Überlastung des Empfängers erreicht über „Sliding Window“ im Empfangspuffer Empfangsfenstergröße wird Sender mitgeteilt
- Staukontrolle - Vermeidung Überlast im Netz (Router-Wartezeiten steigen extrem) erreicht über Beobachtung des Netzverhaltens (Timeouts, duplizierte Quittungen)

Geführt werden beim Sender

Sendefenster  
(gemeldeter freier Platz im Empfangspuffer, evtl. veraltet)

Überlastfenster  
(je nach Netzverhalten vom Sender ermittelte sinnvolle Fenstergröße, vor allem abhängig von Umlaufzeit RTT)

Das kleinere der beiden Fenster gilt als Begrenzung !

Zu f)

TCP Tahoe, TCP Reno, TCP New Reno, ...

Zusammenarbeit zwischen verschiedenen Varianten möglich, da sie sich nur in der Staukontrolle unterscheiden.

#### 4. TCP – Fehlererkennung-/korrektur

- a. Sendewiederholung (Retransmission) bei ausbleibender Quittierung  
Welche Folgen hat ein zu langes RTO (Retransmission Timeout)?  
Welche Folgen hat ein zu kurzes RTO ?  
Kann RTO auf den RTT-Wert (Round Trip Time) gesetzt werden?
- b. Berechnen Sie die zeitliche Entwicklung des RTO-Wertes nach RFC 2988 für eine Serie von Übertragungen mit den RTT-Meßwerten (1s, 2s, 3s, 2s) und den Glättungsfaktoren  $\alpha=1/8$  und  $\beta=1/4$ .
- c. Sendewiederholung nach Serie duplizierter Quittungen (Fast Retransmission)  
Nennen Sie zwei Ursachen für duplizierte Quittungen.  
Wieviele duplizierte Quittungen werden ohne Sendewiederholung akzeptiert?
- d. Eine TCP-Verbindung hat eine Fenstergröße von 8 Segmenten und eine RTT von 800ms.  
Der Sender überträgt mit einer konstanten Rate von 1 Segment/100ms,  
der Empfänger sendet Quittungen ohne Verzögerung zurück.  
Segmentverluste werden durch Fast Retransmission erkannt.

Wieviel Zeit verliert der Sender bei einem Datenverlust?

Zu a)

RTO zu lang - Ineffizient, Datenrate zu gering  
RTO zu kurz - unnötige Doppelübertragungen

RTO  $\gg$  RTT wegen Schwankungen der Laufzeit

Ziel RTO anpassen an (veränderliche) Netzeigenschaften

RFC 2988

meist  $\alpha=1/8$  und  $\beta=1/4$

RTT messen, dann

SRTT = RTT (smoothed ...)  
RTTVAR =  $0,5 * RTT$   
RTO =  $SRTT + 4 * RTTVAR$

Danach zyklisch

SRTT =  $(1 - \alpha) * SRTT + \alpha * RTT$   
RTTVAR =  $(1 - \beta) * RTTVAR + \beta * |SRTT - RTT|$   
RTO =  $SRTT + 4 * RTTVAR$

Zu b)

RTT = 1s	SRTT	= 1 s	
	RTTVAR	= 0,5 s	
	RTO	= 3 s	
RTT = 2s	SRTT	= $(7/8 * 1 + 1/8 * 2)$ s	= 1,125 s
	RTTVAR	= $(3/4 * 0,5 + 1/4 *  1,125 - 2 )$ s	= 0,343 s
	RTO	= $(1,125 + 4 * 0,343)$ s	= 2,5 s
RTT = 3s	SRTT	= $(7/8 * 1,125 + 1/8 * 3)$ s	= 1,359 s
	RTTVAR	= $(3/4 * 0,343 + 1/4 *  1,359 - 3 )$ s	= 0,668 s
	RTO	= $(1,125 + 4 * 0,343)$ s	= 4,029 s
RTT = 2s	SRTT	= $(7/8 * 1,359 + 1/8 * 2)$ s	= 1,439 s
	RTTVAR	= $(3/4 * 0,668 + 1/4 *  1,439 - 2 )$ s	= 0,641 s
	RTO	= $(1,125 + 4 * 0,343)$ s	= 4,004 s

Zu c)

Bei Segmentverlusten werden korrekt übertragene spätere Segmente nicht quittiert, sondern nur der Wert des zeitlich davor korrekt übertragenen Segmentes, d.h. es gibt sog. „duplizierte Quittungen“.

Bei Nutzung alternativer Übertragungswege kann eine zeitlich späteres Segment eher beim Empfänger eintreffen, dann gibt es auch duplizierte Quittungen.

Mehrfach empfangene duplizierte Quittungen geben einen Hinweis auf Übertragungsfehler !

**Fast Retransmission** nicht auf Timeout warten, sondern  
Sendewiederholung nach Empfang von **3** dupl. Quittungen

Zu d)

Zeit (ms)	Sender			Kanal	Empfänger	
	S	E	Komm.		E	S
000	S0		F=7	Alle Übertragungen positiv  bis auf Verlust von S1, danach kumulative Qu.		
100	S1		F=6			
200	S2		F=5			
300	S3		F=4			
400	S4		F=3			S0 Q0
500	S5		F=2			
600	S6		F=1			S2 Q0
700	S7		F=0			S3 Q0
800	S8	Q0	F=0			S4 Q0
900			S- Pause			S5 Q0
1000		Q0	...	Sende-Wdhlg. von S1	S6 Q0	
1100		Q0	...		S7 Q0	
1200	S1	Q0	3. DQ		S8 Q0	
1300		Q0	...			
1400		Q0	...			
1500		Q0	...			
1600			...		S1-W Q8	
1700			...			
1800			...			
1900			...			
2000	S9	Q8	Fort- setzung F=7			

Im Zeitraum von 900 ... 1900 ms stockt die Übertragung, d.h. Zeitverlust = 1100 ms

Dies gilt für „selektive reject“

Bei „go back n“ wäre der Zeitverlust noch größer (1900 ms)

## 5. TCP – Flußsteuerung

Ein Sender sendet TCP-Segmente (Daten 1 ...1024 Byte), zunächst mit 10 vollen Segmenten pro Sekunde, der Empfänger verarbeitet die Information byteweise mit 7 KByte/s. Der Empfangspuffer ist 30 KByte groß. Übertragungszeiten sind vernachlässigbar.

- Nach welcher Übertragungsdauer kommt es zu Verarbeitungsstau im Empfänger?
- Was passiert danach?
- Wie kann die Übertragungseffizienz verbessert werden?

Zu a)

Datenrate DR = 10 \* 1024 byte/s  
Verarbeitungsrate VR = 7000 byte/s

Differenz pro Segment = DR/10 – VR/10  
= 324 byte/s

Puffergröße = 30 \* 1024 byte  
= 30720 byte

Anzahl der Segmente (bis Pufferüberlauf) = 30720 / 324 = 94,81      nach 94 Segmenten (9,4 s)  
ist Puffer fast voll

Zu b)

Nach 9,5 s      Senden Segment mit  
Größe = 30720 byte – (94\*324) byte  
=264 byte

Effizienz sinkt

- konstanter Overhead durch MAC/IP/TCP-Header wirkt sich relativ stärker bei kleinen Segmenten aus
- bei geringer Fenstergröße wirkt sich die Umlaufzeit RTT aus

Zu c)

Ausweg: „Zero Window“ melden,  
bevor Puffer voll, z.B. schon bei Pufferbelegung von 50 %

dann muß Sender warten, sendet aber später mit höherer Effizienz

## 6. TCP Congestion Control

Eine TCP-Verbindung hat eine Fenstergröße von 64 KByte und eine RTT von 3 ms.  
Die maximale Segmentgröße MSS beträgt 1460 Byte, Überlastungen liegen nicht vor.

- Wie entwickelt sich (5 Schritte) die Datenrate bei linearem Anstieg des Sendefensters?
- Wie entwickelt sich (5 Schritte) die Datenrate bei der „Slow-Start-Methode“?
- Wie lange dauert es bei Slow Start mindestens, bis das erste volle Fenster gesendet werden kann?
- Warum kombiniert man bei TCP-Implementierungen Slow Start und linearen Anstieg?
- Vergleichen Sie TCP Tahoe mit TCP Reno.
- Interpretieren Sie nachfolgenden Verlauf der Größe des Überlastfensters (in Segmenteinheiten)

Zeit		0	+1	+2	+3	+4	+5	+6	+7	+8	+9
0		1	2	4	8	16	17	18	19	20	10
+10		11	12	13	14	15	16	1	2	4	8
+20		9	10	11	12	13	14	15	16	8	9

Zu a)

Zyklus1	$DR = 1 * 1460 \text{ byte} / 3 \text{ ms}$	= 3,9 Mbit/s
Zyklus2	$DR = 2 * 1460 \text{ byte} / 3 \text{ ms}$	= 7,8 Mbit/s
Zyklus3	$DR = 3 * 1460 \text{ byte} / 3 \text{ ms}$	= 11,7 Mbit/s
Zyklus4	$DR = 4 * 1460 \text{ byte} / 3 \text{ ms}$	= 15,6 Mbit/s
Zyklus5	$DR = 5 * 1460 \text{ byte} / 3 \text{ ms}$	= 19,5 Mbit/s

Zu b)

Zyklus1	$DR = 1 * 1460 \text{ byte} / 3 \text{ ms}$	= 3,9 Mbit/s
Zyklus2	$DR = 2 * 1460 \text{ byte} / 3 \text{ ms}$	= 7,8 Mbit/s
Zyklus3	$DR = 4 * 1460 \text{ byte} / 3 \text{ ms}$	= 15,6 Mbit/s
Zyklus4	$DR = 8 * 1460 \text{ byte} / 3 \text{ ms}$	= 31,2 Mbit/s
Zyklus5	$DR = 16 * 1460 \text{ byte} / 3 \text{ ms}$	= 62,4 Mbit/s

Zu c)

Pufferkapazität	=	64 Kbyte/1460 byte	
	=	65535 / 1460 Segmente	= 44,9 Segmente
	=	44 volle Segmente + Rest	

Annahme: Empfänger verarbeitet Information sofort

Slow Start	1	1 Segment
	2	2
	3	4
	4	8
	5	16
	6	32
	7	Sendefenster kann voll genutzt werden

Zu d)

Slow Start	schnelle Annäherung an optimales Fenster Gefahr der Überschreitung wegen schnellen Anstiegs
lin. Anstieg	langsame Annäherung an optimales Fenster Geringere Gefahr der Überschreitung (Stauvermeidung)

Zu e)

TCP Tahoe (veraltet)	TCP Reno
-----	-----
Slow Start mit 1 MSS lin. Anstieg ab Schwellwert/2	Slow Start mit 1 MSS lin. Anstieg ab Schwellwert/2
bei Timeout Schwellwert := akt.Wert/2 danach SlowStart ab 1 MSS	bei Timeout Schwellwert := akt.Wert/2 danach SlowStart ab 1 MSS
	bei Fast Retransmit (3x Dupl.ACK) Schwellwert := akt.Wert/2 danach lin. Anstieg ab Schwellwert

Zu f)

Zeitbereich	Methode
-----	
0 ... 4	Slow Start (ab Staufenster=1, Schwellwert 16)
5 .. 15	lin. Anstieg ab Staufenster=17, Timeout, Schwellwert:=8
16...19	Slow Start
20...27	lin. Anstieg ab Staufenster 9, Fast Retransmit, Schwellwert:=8
28 ...	lin. Anstieg ab Staufenster 8, ...

## 7. Socketprogrammierung - Programmierung eines einfachen WWW-Servers

Der untenstehende Programmtext enthält den C-Kode für einen einfachen WWW-Server, der auf der Socket-Schnittstelle aufbaut.

Das Programm kann mit dem frei verfügbaren Compiler Borland C++ Compiler Version 5.5 (<http://www.codegear.com/downloads/free/cppbuilder>) übersetzt werden.

Aufruf: `httpserv www-rootverzeichnis`

- a. Interpretieren Sie die Rolle der fett hervorgehobenen Prozeduren.
- b. Übersetzen und testen Sie die Funktionsfähigkeit des Servers.
- c. Entwickeln Sie einen WWW-Client (optional)

### Quell-Programm

```

/* httpserv.c
 * Demoprogramm zur Programmierung von Netzwerkservers
 * Es wird ein simpler http Server implementiert,
 * der ausschließlich GET requests bearbeiten kann */
#include <stdio.h>
#include <fcntl.h>
#include <errno.h>

#ifdef _WIN32
/* Headerfiles für Windows */
#include <winsock.h>
#include <io.h>

#else
/* Headerfiles für Unix/Linux */
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#define closesocket(s) close(s)
#endif

/* http requests werden auf Port 80 erwartet */

#define HTTP_PORT 80

static void serv_request( int in, int out, char* rootpath);

/***** MAIN *****/
int main( int argc, char **argv)
{

```

```

struct sockaddr_in server, client;
int sock, fd;
int len;

#ifdef _WIN32
/* Initialisiere TCP für Windows ("winsock") */
short wVersionRequested;
WSADATA wsaData;
wVersionRequested = MAKEWORD(1, 1);
if (WSAStartup(wVersionRequested, &wsaData) != 0) {
    fprintf(stderr, "Failed to init windows sockets\n");
    exit(1);
}
#endif

/* Teste auf Kommandozeilenargument "documentroot" */
if (2 != argc) {
    fprintf(stderr, "usage: httpserv documentroot\n");
    exit(1);
}

/* Erzeuge das Socket */
sock = socket(PF_INET, SOCK_STREAM, 0);
if (sock < 0) {
    perror("failed to create socket");
    exit(1);
}

/* Erzeuge die Socketadresse des Servers
 * Sie besteht aus Typ und Portnummer */
memset(&server, 0, sizeof(server));
server.sin_family = AF_INET;
server.sin_addr.s_addr = htonl(INADDR_ANY);
server.sin_port = htons(HTTP_PORT);

/* Erzeuge die Bindung an die Serveradresse
 * (d.h. an einen bestimmten Port) */
if (bind(sock, (struct sockaddr*)&server, sizeof(server)) < 0) {
    perror("can't bind socket");
    exit(1);
}

/* Teile dem Socket mit, dass Verbindungswünsche
 * von Clients entgegengenommen werden */
listen(sock, 5);

/* Bearbeite die Verbindungswünsche von Clients

```

```

* in einer Endlosschleife
* Der Aufruf von accept() blockiert solange,
* bis ein Client Verbindung aufnimmt */
for (;;) {
    len = sizeof( client);
    fd = accept( sock, (struct sockaddr*)&client, &len);
    if (fd < 0) {
        perror( "accept failed");
        exit(1);
    }

    /* Bearbeite den http Request */
    serv_request( fd, fd, argv[1]);
    /* SchlieÙe die Verbindung */
    closesocket( fd);
}
}

/*
* serv_request
* Bearbeite den auf in ankommenden http request
* Die zu sendenden Daten werden auf out ausgegeben
*/
static void serv_request( int in, int out, char* rootpath)
{
    char buffer[8192];
    char *b, *l, *le;
    int count, totalcount;
    char url[256];
    char path[256];
    int fd;
    int eoh = 0;

    b = buffer;
    l = buffer;
    totalcount = 0;
    *url = 0;
    while ( (count = recv( in, b, sizeof(buffer) - totalcount, 0)) > 0) {
        totalcount += count;
        b += count;
        while (l < b) {
            le = l;
            while (le < b && *le != '\n' && *le != '\r') ++le;
            if ('\n' == *le || '\r' == *le) {
                *le = '&#0;';
                printf ("Header line = %s\n", l);
                sscanf( l, "GET %255s HTTP/", url);
            }
        }
    }
}

```

```

        if (strlen(l)) eoh = 1;
        l = le + 1;
    }
}
if (eoh) break;
}

if ( strlen(url)) {
    printf( "got request: GET %s\n", url);
    sprintf(path, "%s/%s", rootpath, url);
    fd = open( path, O_RDONLY);
    if (fd > 0) {
        sprintf( buffer, "HTTP/1.0 200 OK\nContent-Type: text/html\n\n");
        send( out, buffer, strlen(buffer), 0);
        do {
            count = read( fd, buffer, sizeof(buffer));
            send( out, buffer, count, 0);
            printf(".");
            fflush(stdout);
        } while (count > 0);
        close( fd);
        printf("finished request: GET %s\n", url);
    }
    else {
        sprintf( buffer, "HTTP/1.0 404 Not Found\n\n");
        send( out, buffer, strlen(buffer), 0);
    }
}
else {
    sprintf( buffer, "HTTP/1.0 501 Method Not Implemented\n\n");
    send( out, buffer, strlen(buffer), 0);
}
}
}

```

Zu a)

<b>socket()</b>	Service Access Point  IP-Adresse (32 bit) plus Portnummer (16 bit) Client-Port frei wählbar Serverport=80 entsprechend Konvention
<b>bind()</b>	Forderung an Betriebssystem nach Realisierung der Nachrichtenübergaben über Socket
<b>htonl/htons</b>	Beheben Integer-Darstellungsproblem in heterogenen Netzwerken
<b>listen()</b>	Warten auf Verbindungsaufnahme
<b>accept()</b>	Akzeptieren Verbindungswunsch
<b>closesocket()</b>	Schließen Verbindung

Zu b)

Download Borland C++ Compiler Version 5.5 von  
„<http://www.codegear.com/downloads/free/cppbuilder>“

Im Kommandofenster von Windows

Übersetzung → `htpserv.exe`  
➤ `bcc32 htpserv.c`

Anlegen WWW-Server-Rootverzeichnis  
➤ `mkdir www-root`

Start Server  
➤ `htpserv www-root`

Aufruf einer Testseite im Verzeichnis `www-root` mit WWW-Browser  
`http://127.0.0.1/test.htm`

Zu c) WWW-Client (optional)

```

/* httpget.c
 * Demoprogramm zur Programmierung von Netzwerkclients
 * Es wird ein GET request via http an einen Webserver
 * gesendet und das Ergebnis auf der Konsole ausgegeben. */
#include <stdio.h>
#include <errno.h>

#ifdef _WIN32
/* Headerfiles für Windows */
#include <winsock.h>
#include <io.h>

#else
/* Headerfiles für Unix/Linux */
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#define closesocket(s) close(s)
#endif

/* http requests werden normalerweise auf Port 80 vom Server entgegengenommen */
#define HTTP_PORT 80

/***** MAIN *****/
int main( int argc, char **argv)
{
    struct sockaddr_in server;
    struct hostent *host_info;
    unsigned long addr;
    int sock;
    char buffer[8192];
    int count;

#ifdef _WIN32
    /* Initialisiere TCP für Windows ("winsock") */
    short wVersionRequested;
    WSADATA wsaData;
    wVersionRequested = MAKEWORD (1, 1);
    if (WSAStartup (wVersionRequested, &wsaData) != 0) {
        fprintf( stderr, "Failed to init windows sockets\n");
        exit(1);
    }
#endif
#endif

```

```

/* Sind die erforderlichen Kommandozeilenargumente vorhanden? */
if (argc != 3) {
    fprintf( stderr, "usage: httpget server file\n");
    exit(1);
}

/* Erzeuge das Socket */
sock = socket( PF_INET, SOCK_STREAM, 0);
if (sock < 0) {
    perror( "failed to create socket");
    exit(1);
}

/* Erzeuge die Socketadresse des Servers
 * Sie besteht aus Typ, IP-Adresse und Portnummer */
memset( &server, 0, sizeof (server));
if ((addr = inet_addr( argv[1])) != INADDR_NONE) {
    /* argv[1] ist eine numerische IP-Adresse */
    memcpy( (char *)&server.sin_addr, &addr, sizeof(addr));
}
else {
    /* Wandle den Servernamen in eine IP-Adresse um */
    host_info = gethostbyname( argv[1]);
    if (NULL == host_info) {
        fprintf( stderr, "unknown server: %s\n", argv[1]);
        exit(1);
    }
    memcpy( (char *)&server.sin_addr, host_info->h_addr, host_info->h_length);
}

server.sin_family = AF_INET;
server.sin_port = htons( HTTP_PORT);

/* Baue die Verbindung zum Server auf */
if ( connect( sock, (struct sockaddr*)&server, sizeof( server)) < 0) {
    perror( "can't connect to server");
    exit(1);
}

/* Erzeuge und sende den http GET request */
sprintf( buffer, "GET %s HTTP/1.0\n\n", argv[2]);
send( sock, buffer, strlen( buffer), 0);

/* Hole die Serverantwort und gib sie auf Konsole aus */
do {
    count = recv( sock, buffer, sizeof(buffer), 0);
    write( 1, buffer, count);
}

```

```
while (count > 0);  
  
/* SchlieÙe Verbindung und Socket */  
closesocket( sock);  
return count;  
}
```