



---

# Rechnernetze II

SoSe 2019

Roland Wismüller  
Betriebssysteme / verteilte Systeme  
roland.wismueller@uni-siegen.de  
Tel.: 0271/740-4050, Büro: H-B 8404

Stand: 12. März 2020



---

# Rechnernetze II

SoSe 2019

## 12 Zusammenfassung, wichtige Themen



### 1. Wide Area Networks (WANs)

- ➔ Zusammenhang Übertragungsrate / Bandbreite
  - ➔ Fourier-Analyse
  - ➔ **Nyquist-Theorem** (Abtasttheorem)
  - ➔ **Shannon'sches Theorem**  $B = H \cdot \log_2(1 + S/N)$
- ➔ Telefonnetz
  - ➔ synchone Netze (8000 Abtastungen/s), Multiplexing
- ➔ Modulationsverfahren mit mehreren Bits pro Baud
  - ➔ Änderung der Phase und Amplitude
- ➔ PPP: Zweck, Aufgaben
- ➔ *Frame Relay*, ATM: virtuelle Leitungsvermittlung
- ➔ ADSL: *Discrete MultiTone*, mehrere Frequenzkanäle



### 2. Schnelles Ethernet

- ➔ **Techniken zur Bandbreitenerhöhung**
  - ➔ 8B6T, 4B5B, 8B10B etc. statt Manchester-Codierung
  - ➔ mehrere Aderpaare
- ➔ Switches und **Vollduplexbetrieb** (keine Kollisionen)
  - ➔ keine max. Leitungslänge / min. Paketgröße



### 3. Drahtlose Netze

- ➔ **Spreizbandtechniken** (Motivation, FHSS, OFDM, DSSS)
- ➔ **MAC im WLAN:** DCF, PCF
  - ➔ DCF: kein CSMA/CD möglich, **CSMA/CA, MACAW**
    - ➔ Funktionsweise, **Hidden/Exposed Station-Problem**
  - ➔ PCF: Zeitmultiplex (TDMA)
- ➔ **WLAN-Sicherheit**
  - ➔ Funktionsweise und Schwächen von WEP
  - ➔ Verbesserungen durch WPA und WPA2
- ➔ **Bluetooth:** Funkschicht (FHSS), MAC (TDMA), Übertragungssicherung, Sicherheit



### 4. IP-Routing: Spezielle Aspekte

- ➔ **Multicast**
  - ➔ Multicast-Adressen
  - ➔ **IGMP:** Gruppenmanagement
  - ➔ Routing: (gemeinsame oder quellenspezifische) spannende Bäume
    - ➔ *Link-State-Routing:* Gesamtnetz bekannt
    - ➔ Distanzvektor-Routing: **Reverse Path Broadcast / Multicast** (*Flooding* mit *Pruning*)
    - ➔ **PIM:** explizite *Join*-Nachrichten bauen Multicast-Baum auf



### 4. IP-Routing: Spezielle Aspekte ...

- ➔ Mobile IP
  - ➔ Weiterleitung durch Heimatagen über IP-Tunnel
- ➔ **MPLS: virtuelle Leitungsvermittlung**
  - ➔ *Label-Edge-Router* fügt Label an Paket an, restliche Router betreiben Label-basierte Weiterleitung



### 5. VPN, IP-Tunnel und IPsec

- ➔ **Secure IP**
  - ➔ Vor-/Nachteile der Sicherheit auf Vermittlungsschicht
  - ➔ **AH und ESP-Protokoll, Transport- und Tunnel-Modus**
    - ➔ was wird jeweils verschlüsselt, authentifiziert?
  - ➔ **Security Association**, Konfiguration

### 6. Überlastkontrolle und Ressourcenzuteilung

- ➔ **Überlastkontrolle**
  - ➔ Basismechanismus „**Additive Increase / Multiplicative Decrease**“
  - ➔ Erweiterungen „*Slow Start*“, „*Fast Retransmit / Fast Recovery*“



### 6. Überlastkontrolle und Ressourcenzuteilung ...

#### ➔ Überlastvermeidung

##### ➔ Router-zentrisch: mittlere Warteschlangenlänge

- ➔ DECbit: Warnbit im Header
- ➔ RED: Verwerfen einiger zufälliger Pakete

##### ➔ Host-zentrisch: Latenz, Durchsatz

- ➔ TCP Vegas: Vergleich erwarteter und erreichter Durchsatz
  - ➔ Ziel: belege konstante, kleine Zahl von Puffern im Router

#### ➔ Quality of Service

- ➔ Dienstklassen mit verschiedenen Garantien (Bandbreite, Latenz, Jitter)



### 6. Überlastkontrolle und Ressourcenzuteilung ...

#### ➔ Quality of Service

##### ➔ feingranular: Datenflüsse einzeln betrachten

- ➔ Mechanismen: *Flow Specs* (geforderter Dienst, Datenfluß), Zugangskontrolle, Ressourcen-Reservierung, Paket-Scheduling
- ➔ Beispiel *IntServ*: GS, CLS
- ➔ *Token-Bucket*-Filter, RSVP, WFQ

##### ➔ grobgranular: nur Klassen von Datenflüssen betrachten

- ➔ Paketklassifizierung an der Peripherie, fest reservierte Ressourcen
- ➔ Beispiel *DiffServ*: EF, AF (4 Klassen á 3 Prioritäten)

##### ➔ Skalierbarkeit vs. echte Garantien



### 7. Anwendungsprotokolle

- ➔ Netzwerkmanagement
  - ➔ Aufgaben
  - ➔ SNMP: Lesen/Schreiben von Objekten, MIB: Aufbau, Inhalt
- ➔ **Multimedia-Anwendungen**
  - ➔ RTP/RTCP: Synchronisation (Zeitstempel), Multiplexing, Sequenznummern, *Sender-Feedback*
- ➔ Overlay-Netze: *Routing-Overlays*, *Peer-to-Peer-Netze*
  - ➔ *Peer-to-Peer-Netze*: verteilte Speicherung von Daten / verteiltes Suchen (Basis: *Hashing* und Weiterleitung)



### 8. Netzwerkprogrammierung

- ➔ Sockets: Netzunabhängiges API für Kommunikation
  - ➔ *Stream* und *Datagram Sockets*
  - ➔ `bind()`, `connect()`, `listen()`, `accept()`
- ➔ Server-Design: Prozeß / Thread pro Client, `select`



### 9. Netze für Cluster und Hochleistungsrechner

- ➔ Bisektionsbandbreite, Verbindungsgrad, Durchmesser
- ➔ Netztopologien, *Crossbar*, Clos-Netz (Idee)
- ➔ *Virtual-Cut-Through-Routing*, *Remote DMA*, *OS Bypass*

### 10. Netze für Automatisierungssysteme

- ➔ Anforderungen, Echtzeit, Merkmale von Feldbussen
- ➔ MAC für Echtzeit-Aufgaben: *Token Passing*, *Master-Slave*, CSMA/CA (prioritätengesteuerte Arbitrierung)

### 11. Drahtlose Sensornetze

- ➔ Spezielle Anforderungen: Energieeffizienz, *data centric*