

# Wiederholungsfolien: Netzwerke

## 1. Router vs. Switch

- Router - mehr Intelligenz, Entscheidungen über längere Netzwerkabschnitte
- (Sicherungsschicht)-Switches – primitiv, haben Informationen nur über direkte Nachbarn

## 2. Paketvermittlung vs. Leitungsvermittlung

### Paketvermittlung vs. Leitungsvermittlung

Paketvermittlung	Leitungsvermittlung	Ineffizienz der LV?
Ressourcen werden nur nach Bedarf verwendet	Dedizierte Belegung und Reservierung der Ressourcen	Weniger gleichzeitige Nutzer
Pakete werden mit der vollen Übertragungsgeschwindigkeit der Leitung übertragen	Unterteilung der Bandbreite in Kanäle (pro Leitung ein Kanal)	Übertragung langsamer als ggf. möglich

## 3. 4 Arten von Verzögerungen:

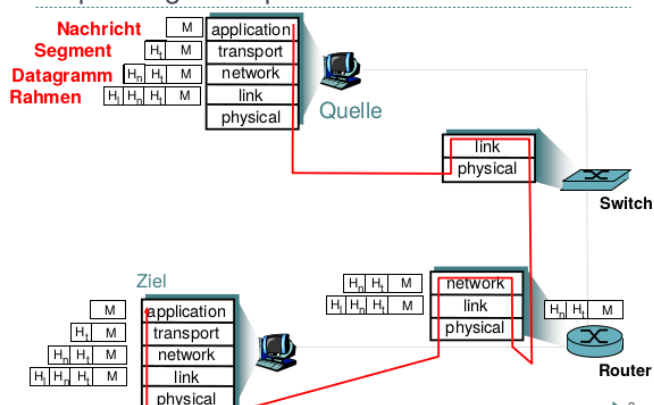
- Verarbeitungsverzögerung
- Warteschlangenverzögerung
- Übertragungsverzögerung
- Ausbreitungsverzögerung

## 5. Netzwerkprotokollschichten (Hybrides-Referenzmodell)

Layer:	Gekapselte Nachrichten	Endpunkte	Adressierung
Anwendungsschicht	Nachrichten	Nutzer, Accounts, auch vermittelnde Server	IDs, URL, Emailadresse
Transportschicht	TCP-Segment, UDP-Datagramm	Prozesse/Sockets	Protocol + Portnummer
Netzwerkschicht	Datagramm	Hosts	IP-Adresse (w.x.y.z)
Sicherungsschicht	Rahmen	Netzwerkknotten	MAC-Adresse (ff:ff:ff:ff:ff:ff)

## Beispiel für Kapselung (Hinweis zur Aufgabe 4.2 von der Altklausur 2016 )

### Kapselung - Beispiel



## 6. TCP vs. UDP-Protokoll

### Zwei Grundlegende Internet-Protokolle

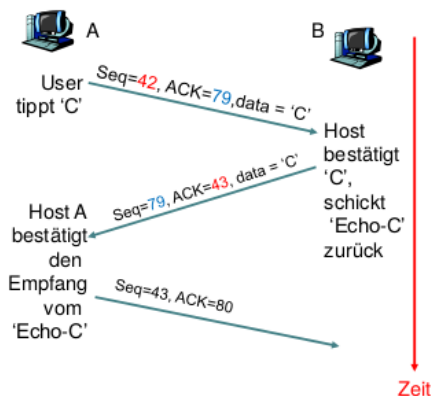
#### ► TCP Protokoll

- **Verbindungsorientiert:** Einrichtung (setup) der Verbindung durch ein **Handshake** ist nötig
  - Zustand nur an den „Enden“
    - => keine leitungsorientierte Verbindung!
- **Verlässliche Übertragung**
- **Überlastkontrolle:** Der Empfänger wird nicht überlastet; Drosseln der Ü-Geschwindigkeit, wenn Netzwerk überlastet
- **Keine:** Echtzeitgarantien, Sicherheit, min. Durchsatz, Mindestlohn

#### ► UDP Protokoll

- **Verbindungslos:** Kein Handshake nötig
- **Unzuverlässig:** Keine Garantien auf die Zustellung der Nachricht oder die richtige Reihenfolge ihrer Ankunft
- **Keine:** Überlastkontrolle, Echtzeitgarantien, Sicherheit
- Frage: Warum wird UDP überhaupt verwendet?

#### Ein telnet-Scenario



#### TCP-Protokoll mit Handshake

- 2 Typen von Sockets: **verbindungsorientiert (TCP)** und **verbindungslos (UDP)**

**verbindungslos:** - Sender fügt jedem Segment IP-Adresse + Portnummer des Empfängers hinzu  
 - Java-Server mit UDP braucht nur ein Socket

**verbindungsorientiert:** - 2 Sockets sind nötig → **welcome socket** (wird vom Client an einem lokalen Port erstellt) und **connection socket** (wird von dem Server nur für diese Verbindung erstellt)  
 - identifiziert durch: IP-Adresse der Quelle, Quellportnummer, IP-Adresse des Ziels, Zielportnummer

#### Wiederholung: **Selective Repeat**

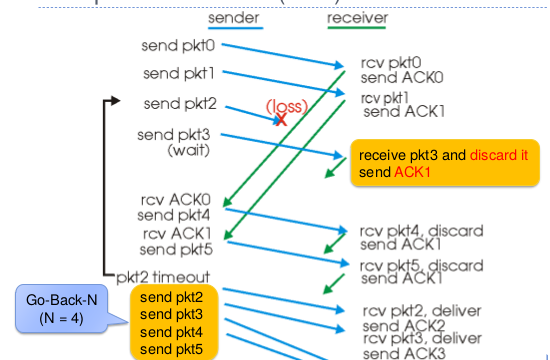
##### Empfänger

- Empfänger bestätigt er jedes korrekt empfangene Paket **individuell**
- Empfängt alle Pakete (im gewissen Bereich der Sequenznummer) und bringt sie in richtige Reihenfolge

##### Sender

- Sendet bis zu **N Pakete ohne ACKs (Pipelining)**
- Hat einen Timer für **jedes** nicht-bestätigte Paket
- Sender schickt erneut **nur diese** Pakete, für die die ACKs **nicht** empfangen wurden
- Daher der Name „**Selective Repeat**“

#### Beispiel Go-Back-N (N=4)



## 7. IP-Adressen

- **NetzID** identifiziert ein Netzwerk; obere Bits
- **HostID** identifiziert ein Interface innerhalb des Netzwerks, untere Bits

### Aufgabe 12.4:

Einer Firma stehen für ihr Intranet die IP-Adressen im Bereich 128.10.192.0 bis 128.10.199.255 zur Verfügung.

- Wie viele Adressen enthält dieser Bereich? Der Bereich soll in mehrere Subnetze mit jeweils 30 Hosts aufgeteilt werden. Geben Sie eine geeignete Subnetzmaske an!

192.0 = 1100 0000.0000 0000  
und 199.255 = 1100 0111.1111 1111

Dieser Bereich enthält  $2^{11} = 2048$  IP-Adressen

Berücksichtigen Sie, dass zwei Adressen in jedem Subnetz immer für Broadcast reserviert sind: 0..0 und 1..1. D.h. jedes Subnetz mit 30 Hosts benötigt 32 Adressen. Um 32 Adressen pro Subnetz zu adressieren, werden 5 Bit benötigt.

Die letzten 5 Bit der Subnetzmaske müssen 0 sein.

Subnetzmaske: 255.255.255.224 = 1111 1111. 1111 1111. 1111 1111. 1110 0000

- Wie viele Subnetze mit je 30 Hosts können adressiert werden?

Für Subnetz- und Host-ID stehen insgesamt 11 Bit zur Verfügung. Da 5 Bit für die Host-ID verwendet werden, stehen noch 6 Bit für die Subnetz-ID zur Verfügung. Damit lassen sich also:  $2^6 = 64$  Subnetze adressieren.

- Teilen Sie gemäß der oben gewählten Subnetzmaske die IP-Adresse 128.8.192.171 in Netz-ID und Host-ID auf. Geben Sie beides sowohl in binärer als auch in dezimaler Schreibweise an.

Net-ID: 1000 0000.0000 1000.1100 0000.1010 0000 = 128.8.192.160

Host-ID: 0000 0000.0000 0000.0000 0000.1011 = 0.0.0.11

- Eine andere Firma hat eine Class-C-Adresse bekommen und hat 180 Hosts, die angeschlossen werden sollen. Ist es möglich, das Netzwerk in ein Subnetz mit 40 Rechnern und 5 Subnetze mit je 28 Rechnern aufzuteilen? Wenn ja, wie? Wenn nein, warum nicht?

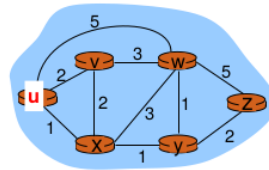
Eine Class-C Adresse hat 8 Bit zum Aufteilen in subnet und host ID. Um 5 oder 6 Subnetze zu adressieren, sind in jedem Fall 3 Bit nötig. Mit den übrigen 5 Bit können Sie aber maximal 32 Adressen darstellen. Diese Aufteilung ist daher mit starren Subnetzmasken nicht realisierbar.

Anmerkung: VLSM (variable length subnet masks) erlaubt Subnetzmasken verschiedener Größe pro Subnetz.

## 8. Routing-Algorithmen:

### Dijkstra-Algorithmus: Beispiel

Step	N'	D(v),p(v)	D(w),p(w)	D(x),p(x)	D(y),p(y)	D(z),p(z)
0	u	2,u	5,u	1,u	∞	∞
1	ux	2,u	4,x	2,x	∞	∞
2	uxy	2,u	3,y	3,y	4,y	∞
3	uxyv	2,u	3,y	3,y	4,y	∞
4	uxyvw	2,u	3,y	3,y	4,y	4,y
5	uxyvwz	2,u	3,y	3,y	4,y	4,y



$p(s)$ : vorheriger Knoten  
(Nachbar von s) entlang  
des momentan  
kostengünstigsten Pfades  
von der Quelle zu s

- Distanzvektoralgorithmus:  $d_x(y) = \min_{v \in \text{neigh}(x)} \{c(x,v) + d_v(y)\}$  (Bellman-Ford-Gleichung)