

## 4 Ethernet

- weltweit sehr verbreitete **LAN-Technologie**
- historische Entwicklung:
  - 1976 im Xerox Palo Alto Research Center entwickelt
  - 1980 erster Standard von Xerox, DEC und Intel
  - 1983 erster IEEE Standard 802.3
- leistet **ungesicherte Paketübertragung** (ohne Quittierung!) auf verschiedenen Übertragungsmedien
- verschiedene Standards:

Bezeichnung	Standard	Übertragungsmedium	Bitrate	Topologie
„ <i>Thickwire</i> “ Ethernet	10Base5	10 mm Ø Koaxialkabel	10 MBit/s	Bus (< 500m)
„ <i>Thinwire</i> “ Ethernet	10Base2	5 mm Ø Koaxialkabel	10 MBit/s	Bus (< 185m)
Ethernet	10BaseT	2 TP-Kabel ( <b>UTP</b> )	10 MBit/s	Stern
Fast Ethernet	100BaseT	2 TP-Kabel (UTP)	100 MBit/s	Stern
Gigabit Ethernet	1000BaseT	4 abgeschirmte TP-Kabel ( <b>STP</b> )	1000 MBit/s	Stern

### 4.1 Ethernet Protokoll

- Standard-Protokoll für Halb-Duplex Betrieb: **CSMA/CD**
  - 1) wenn ein Rechner A Daten senden möchte, prüft er zunächst, ob der Übertragungskanal frei ist (**CS** = *C*arrier *S*ense)
  - 2) sobald Kanal frei ist, beginnt Rechner A mit der Datenübertragung
  - 3) gleichzeitig hört Rechner den Datenkanal ab und vergleicht gesendetes Signal mit abgehörtem Signal
  - 4) hat ein anderer Rechner B fast gleichzeitig auch mit einer Übertragung begonnen (**MA** = *M*ultiple *A*ccess), kann Rechner A eine Kollision feststellen (**CD** = *C*ollision *D*etect)
- Verhalten im Falle einer **Kollision**:
  - 5) Rechner A bricht Übertragung ab und sendet spezielles „*Jamming*“-Signal, um die Ungültigkeit der Daten zu kennzeichnen
  - 6) Rechner A wartet eine zufällig gewählte Zeitspanne und beginnt Übertragung erneut mit Punkt 1)

## 4.1 Ethernet Protokoll (2)

---

- Aufbau eines **Ethernet-Paketes** (Größe: 64-1526 Bytes):

<i>preamble</i> (7 Byte)	10101011 (1 Start Byte)	<i>destination addr.</i> (6 Byte)	<i>source addr.</i> (6 Byte)	<i>length</i> (2 Byte)	<i>data</i> (38–1500 Byte)	<i>checksum</i> (4 Byte)
-----------------------------	----------------------------	--------------------------------------	---------------------------------	---------------------------	-------------------------------	-----------------------------

- *preamble* : 56-Bit Muster zur Synchronisation  
(gibt dem Empfänger Zeit, ein Signal zu erkennen und Lesevorgang zu starten)
- *source address* : eindeutige physikalische 6-Byte MAC-Adresse des sendenden Rechners; in den ersten 3 Byte ist Hersteller/Modell kodiert
- *destination address* : eindeutige physikalische 6-Byte MAC-Adresse des Zielrechners oder Broadcast (alle 48 Bit auf 1 gesetzt)
- *length* : Länge des Datenfeldes (in Byte)
- *data* : zu übertragende Daten (max. 1500 Byte); ggf. werden Null-Bytes eingefügt, um eine minimale Paketlänge von 64 Byte zu erreichen
- *checksum* : 32-Bit Prüfsumme  
( $\Rightarrow$  mehrere Bitfehler können mit hoher Wahrscheinlichkeit erkannt und zum Teil auch korrigiert werden!)

## 4.2 Ethernet Timing

---

- **Signallaufzeit  $\tau$**  (*propagation delay*)
  - maximale Zeit  $\tau$ , die ein Paket benötigen darf, um von einem Ende eines Netzwerks zum anderen Ende transportiert zu werden
  - legt maximale Entfernung im Ethernet Netzwerk fest
- **Rundlaufzeit  $2\tau$**  (*round trip time*)
  - minimale Zeit für die Erkennung einer Kollision mit dem am weitesten entfernten Rechners im Netzwerk
  - wurde für 10/100 MBit/s Ethernet festgelegt als die Übertragungszeit für 512 Bit, also 51,2  $\mu$ s bzw. 5,12  $\mu$ s
  - hieraus resultiert eine minimale Paketlänge von 64 Byte
- **Wartezeit** (*interpacket gap*)
  - minimale Zeit, die ein Knoten vor dem Versenden eines weiteren Paketes warten muss
  - Übertragungszeit von 96 Bit, bei 10/100 MBit/s also 9,6  $\mu$ s bzw. 0,96  $\mu$ s
  - ermöglicht auch anderen Knoten, ein Paket zu senden ( $\Rightarrow$  Fairness!)

## 5 Fallstudie: TCP/IP

- TCP/IP = *Transmission Control Protocol / Internet Protocol*
- ursprünglich vor über 30 Jahren im Rahmen des ARPANET entwickelt, heute **Standard-Protokolle** des **Internets**
- hohe Popularität durch **offene** Protokolle  
(zuerst in jeder Unix-Implementierung, heute auf jeder Plattform verfügbar)
- Funktionalität entspricht in etwa den ISO/OSI-Schichten 3 und 4

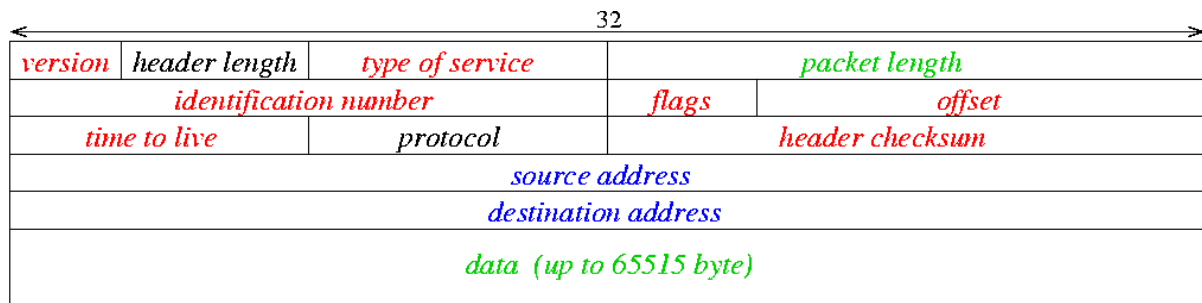
ISO/OSI	Anwendungsschicht	7	<i>application layer</i>					Internet
	Darstellungsschicht	6	email/smtp	ftp	http	DNS	telnet	⋮
	Steuerungsschicht	5						
	Transportschicht	4	TCP			UDP		
	Vermittlungsschicht	3	IP			ARP		
	Sicherungsschicht	2	<i>network layer</i>					
	Physikalische Schicht	1	Ethernet	FDDI	SLIP / PPP	⋮		

### 5.1 IP Protokoll

- Protokoll der Vermittlungsschicht für eine **paketvermittelte** Datenübertragung zwischen zwei Netzwerkknoten A und B
- von der Transportschicht übernommene Nachrichten werden ggf. in kleinere Pakete unterteilt und **unbestätigt** über ein Subnetzwerk übertragen  
(⇒ keine gesicherte Übertragung, Verlust von Paketen ist möglich!)
- Pakete können über Netzwerke **unterschiedlicher Technologie** transportiert werden (i.a. mit unterschiedlichen Paketgrößen)
- **dynamisches Routing** mittels Routing-Tabellen  
(Pakete einer Datenübertragung können unterschiedliche Wege nehmen)
- jeder Host und jeder Router im Internet verfügt über eine ihm zugewiesene **globale logische Adresse** (IP-Adresse)  
(32-Bit Adressen in IP Version 4, 128-Bit Adressen in IP Version 6)

## 5.1 IP Protokoll (2)

- Aufbau eines **IPv4-Paketes**:



- *version* : IP Versionsnummer (z. Zt. IPv4)
- *type of service* : angeforderte Service-Qualität (z.B. *normal*, *priority*)
- *packet length* : Länge des Gesamt-Paketes (in Bytes, max. 65535)
- *identification number* : Datenübertragung, zu der ein Paket gehört
- *flags* und *offset* : Angaben zum Paket (Position, noch weitere Pakete ?)
- *time-to-live* : max. Zeit, die ein Paket im Netzwerk verbleiben darf
- *header checksum* : Prüfsumme, nur für Header!

## 5.2 IPv4 Adressen

- 5 verschiedene **Netzwerkclassen** bei IPv4-Adressen:

A	0	<i>network ID</i>		<i>host ID</i>	1.0.0.0 bis 126.255.255.255
B	1 0	<i>network ID</i>		<i>host ID</i>	128.0.0.0 bis 191.255.255.255
C	1 1 0	<i>network ID</i>		<i>host ID</i>	192.0.0.0 bis 223.255.255.255
D	1 1 1 0	<i>multicast address</i>			224.0.0.0 bis 239.255.255.255
E	1 1 1 1 0	<i>reserved for future use</i>			240.0.0.0 bis 255.255.255.255

- jedes Netzwerk der Klassen A, B oder C kann in **Subnetzwerke** aufgeteilt werden, z.B. für Klasse B:

B	1 0	<i>network ID</i>	<i>subnetwork ID</i>	<i>host ID</i>
---	-----	-------------------	----------------------	----------------

- Unterteilung wird jedem Knoten eines Subnetzwerks durch eine binäre **Subnetzmaske** (0 für alle *host ID* Bits) angezeigt, z.B. für ein Subnetzwerk mit 64 Adressen: 255.255.255.196

## 5.2 IPv4 Adressen (2)

- spezielle IP Adressen:

0 0	<i>this host</i>		
0 0	<i>host ID</i>	<i>host in the local network</i>	
<i>network ID</i>		1 1 1 1 1 1 1 1	<i>network broadcast</i>
1 1			<i>local broadcast</i>
127	(anything)		<i>loopback test</i>

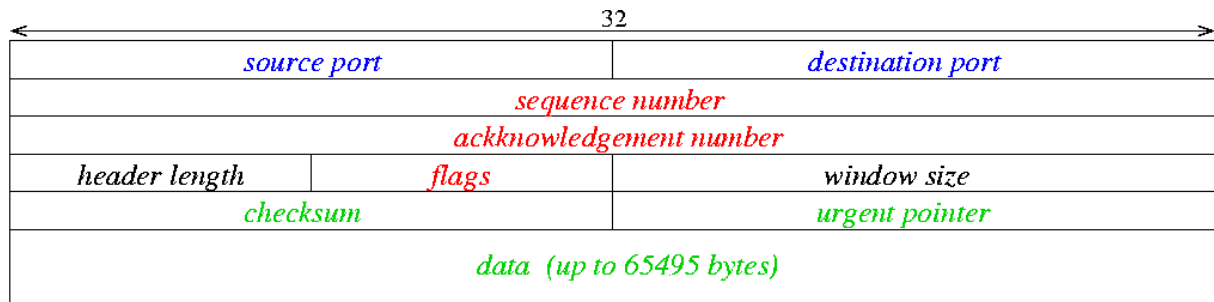
- den IP-Adressen werden **symbolische Namen** zugeordnet (z.B.: 134.60.73.1 = neuro.informatik.uni-ulm.de)
- mittels des **DNS-Dienstes** (*Domain Name Service*) kann die IP-Adresse eines Rechners anhand des symbolischen Namens ermittelt werden
- **ARP-Protokoll** dient der Abbildung von logischen IP-Adressen auf physikalische 48-Bit Ethernet-Adressen

## 5.3 TCP Protokoll

- Protokoll der Transportschicht **zwischen zwei Endknoten** einer Datenübertragung
- ermöglicht Übertragung eines byte-orientierten **Datenstroms** vom Sender zum Empfänger
- leistet **sichere Übertragung**: TCP erkennt und fordert fehlende Pakete erneut an, sortiert Pakete ggf. in korrekter Reihenfolge
- Schnittstellen zum TCP bilden von der Anwendung vergebene Kommunikationspunkte (**Ports**), die durch 16-Bit Portnummern angegeben werden
- Portnummern für Standard-Anwendungsdienste sind festgelegt (z.B.: 7 für echo, 21 für ftp, 23 für telnet, 25 für smtp, 80 für http)
- die Kombination aus IP-Adresse und Portnummer bezeichnet man als **TSAP** (*Transport Service Access Point*)

## 5.3 TCP Protokoll (2)

- Aufbau eines **TCP-Paketes**:



- *source/destination port* : End-Kommunikationspunkte auf beiden Seiten
- *sequence number* : Nummer des ersten im Paket gesendeten Bytes
- *acknowledgement number* : Nummer des nächsten erwarteten Bytes
- *flags* : Kennzeichnung spezieller Nachrichten, z.B. „*acknowledgement*“, „*urgent*“, „*final*“ oder „*connection reset*“
- *urgent pointer* : Zeiger (Offset) auf wichtige Information in den Daten
- *checksum*: 16-Bit Prüfsumme, für Header und Daten!

## 5.4 Sockets

- auf TCP/IP aufbauende „*stream*“-basierte **Kommunikations-schnittstelle**, mit der eine Punkt-zu-Punkt Verbindung zwischen zwei Prozessen implementiert werden kann
- Unterscheidung zwischen **Server** (Dienstanbieter) und **Klient**
- Sockets in Java: Klassen `Socket` und `ServerSocket` im Package `java.net`
- Ablauf einer Socket-basierten Kommunikation in Java:

auf Server-Seite:

```
ServerSocket s = new ServerSocket(1234);  
while (true) {  
    Socket c = sock.accept();  
    in = c.getInputStream();  
    out = c.getOutputStream();  
    query = in.read();  
    out.write(...);  
    c.close(); in.close(); out.close();  
}
```

auf Klienten-Seite:

```
...  
Socket s = new Socket("a.b.de",1234);  
out = s.getOutputStream();  
in = s.getInputStream();  
out.write(...);  
answer = in.read();  
s.close(); in.close(); out.close();  
...
```

## 6 Lernziele

---

- **Begriffe:** halb-/voll-duplex, CSMA/CD, Ethernet, TSAP, ...
- Vor- und Nachteile verschiedener **Netztopologien**
- Funktionalität der 7 Schichten des **ISO/OSI-Referenzmodells**
  - Einsatz zur Unterscheidung von Hub, Switch, Router
  - Einsatz zur Einordnung von Kommunikationsprotokollen
- Arbeitsweise einer **paketorientierten Datenübertragung**
  - **Aufbau** von Paketen (Bsp.: TCP/IP)
  - **Adressierung:**
    - physikalische Adressen (Bsp.: Ethernet)
    - logische Adressen für Hosts und (Sub-)Netzwerke (Bsp.: IPv4)
  - **Sicherung:**
    - Paritätsbits und Prüfsummen (Bsp.: HDLC, Ethernet)
    - Sequenz- und Quittierungsnummern (Bsp.: HDLC, TCP)

## 7 Prüfungen

---

- **Vordiplom**-Klausur zu Technische Informatik I+II findet statt am Dienstag, 30.03.04, von 9.30 bis 12.30 Uhr
  - Voraussetzung: Leistungsnachweis Praktikum Technische Informatik
- **Bachelor**-Klausur zu Technische Informatik II findet statt am Dienstag, 30.03.04, von 9.30 bis 11 Uhr
- Anmeldung: bis 23.03.04 im Studiensekretariat
- Ort: Hörsäle H4/5, H2, H3 und ggf. H1
- keine Hilfsmittel erlaubt!
- Ausweis mit Lichtbild erforderlich!
- Klausuren der letzten Jahre im WWW