Protokolle der OSI-Schicht 2 (Data-Link)

Michael Dienert

19. Oktober 2008

Inhaltsverzeichnis

1 Die Schichtenmodelle in der Netzwerktechnik **2**
1.1 Das OSI- und das DOD-Modell 2
1.2 Kapselung der Daten: Encapsulation 2
1.3 Die Protocol-Data-Units 3

2 Aufgaben der Schicht 2 (Data Link) **4**
2.1 Media Access Control und physikalische Adressen 4
2.1.1 Media Access Control bei gemeinsam genutzten Medien . . . 5
2.1.2 Media Access Control bei Punkt-zu-Punkt Verbindungen . . . 5
2.2 Richtungsabhängigkeit (Duplex) 6

3 Ethernet **6**
3.1 Wer hat das Ethernet wann erfunden? 6
3.2 Grundprinzip von Ethernet 6
3.3 Standards 7
3.4 Topologie von Ethernet 7
3.4.1 Bustopologie bei klassischem Ethernet 7
3.4.2 Übergang zur physikalischen Sterntopologie 7
3.4.3 Netze mit Switches 10
3.5 Anpassung an das OSI-Modell mit Sub-Layers 10
3.5.1 Logical Link Control (LLC) 10
3.5.2 Media Access Control (MAC) 11
3.6 Ethernet-Frame-Typen 12
3.6.1 Ethernet-II 12
3.6.2 Ethernet IEEE 802.3 12
3.7 Felder der Ethernet-Frames 13
3.7.1 MAC-Adressen 13
3.7.2 Type-Feld 14
3.7.3 Präambel und Start Frame Delimiter 14
3.7.4 Frame Check Sequence, FCS 15
3.7.5 Padding 15

3.8 CSMA/CD 15
3.9 CSMA/CA 16
3.9.1 Kollisionsdomänen und Slot-Time 17

1 Die Schichtenmodelle in der Netzwerktechnik

1.1 Das OSI- und das DOD-Modell

Bei der Übertragung von Daten über Netzwerke sind eine Vielzahl von unterschiedlichen Protokollen und Technologien beteiligt. Das hat zum Teil technische Gründe, wie die Trennung in hardwareabhängige und -unabhängige Teile aber auch historische Gründe, da die verschiedenen Technologien nicht gleichzeitig und auch nicht von den gleichen Firmen entwickelt wurden.

Damit nun die Anforderungen aller Beteiligten berücksichtigt und abgestimmt werden können, hat man zwei *Modelle* entwickelt, die die Teilaufgaben eines *weltweiten* Netzwerks darstellen.

Abb. 1 zeigt die beiden Modelle nach DOD und OSI.

Das DOD Modell wurde vor über 30 Jahren von der *Defense Advanced Research Projects Agency* (DARPA), einer Agentur des amerikanischen Verteidigungsministeriums (*Department of Defense*, **DOD**) vorgestellt.

Seine praktische Anwendung ist das heutige Internet.

Das *Open Systems Interconnection Basic Reference Model* (OSI-Model) wurde später entwickelt. Es dient der Veranschaulichung von Netzwerktechnologien und es gibt keine existierende Technologie, bei der das OSI-Modell genau 1:1 umgesetzt wurde. [1]

Für alle *hardwareabhängigen* Schichten ist das IEEE (*Institute of Electrical and Electronics Engineers* verantwortlich. Alle Schichten darüber werden von der IETF (*Internet Engineering Task Force*) definiert. Dazu werden sog. RFCs (*Request For Comments*) verwendet.

Die Dokumente des IEEE sind kostenpflichtig, die RFCs sind frei im Internet erhältlich.

1.2 Kapselung der Daten: Encapsulation

Die nächste Abbildung (Abb. 2) zeigt am Beispiel einer Dateiübertragung mit FTP (File Transfer Protocol), wie die Anwendungsdaten zunächst in kleine Einheiten unterteilt und dann von einer Schicht zur nächsten nach unten weitergereicht werden.

Zunächst werden die Datenstückchen in ein TCP-Segment *eingepackt*.

Den TCP-Header kann man sich nun als *Beschriftung* des Päckchens vorstellen. In diesem Fall trägt die Beschriftung Informationen darüber, an welcher Stelle am Empfangsort der Inhalt des Päckchens wieder in den Datenstrom eingeordnet werden soll und welcher *Dienst*, identifiziert über die **Portnummmer**, der Empfänger ist. Im Beispiel der Abb. 2 wäre das Nummer 21, FTP.

Dieses Päckchen wird dann wiederum eingewickelt. Diesmal enthält die Beschriftung u.a. die **IP-Adresse**, also die Information, an welches *Gerät* (Host, Drucker, ...) das Paket geschickt werden soll.

[1] Am nächsten von allen realen Netzwerktechnologien kommt dem OSI-Modell -wie kann es anders sein- Apple Talk. allerdings ist Apple Talk inzwischen auch auf Macs durch Ethernet und TCP/IP abgelöst worden.

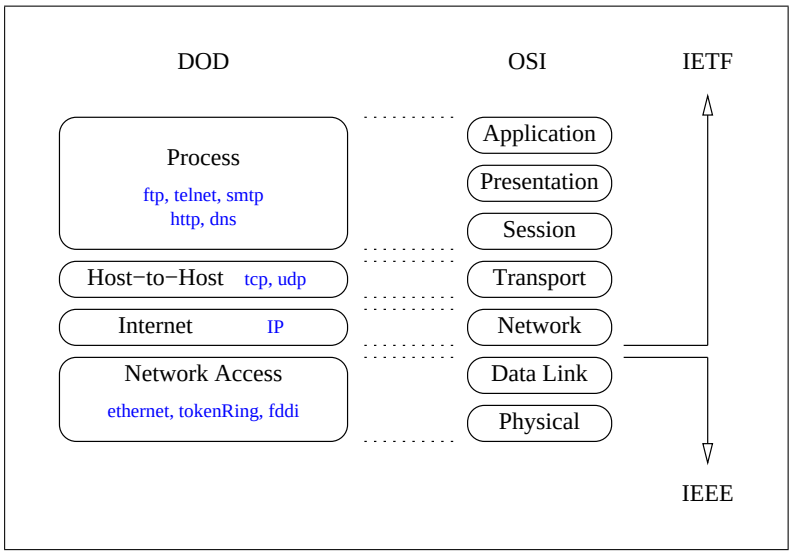

Abbildung 1: Gegenüberstellung von DOD- und Osi-Modell

Und zum Schluss nochmals eine Verpackung: das so entstandene Gebilde ist der *Ethernet-Frame*. Hier steht im Beschriftungstext endlich die **Harware-Adresse** des Empfängers, sowie eine Prüfsumme.

Dieses mehrfache Verpacken der Daten wird **Encapsulation** genannt.

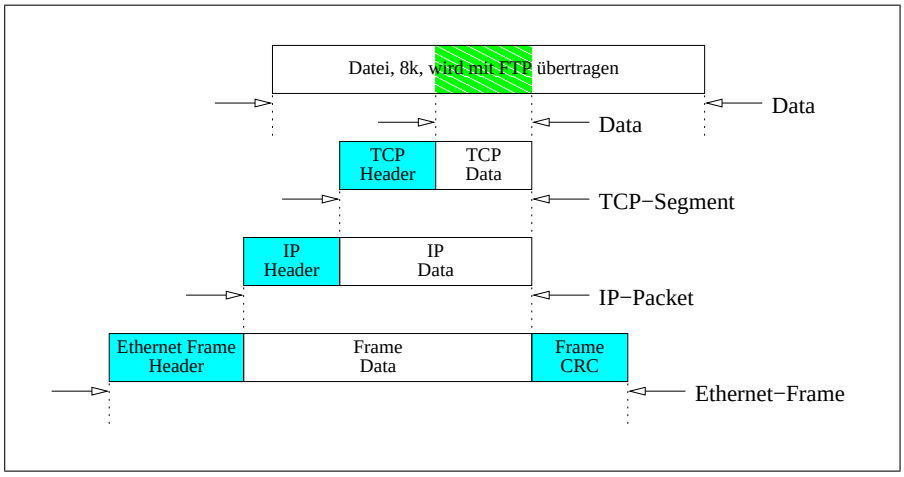

Abbildung 2: Die Ineinanderschachtelung (Encapsulation) der Daten

1.3 Die Protocol-Data-Units

Die Namen der Verpackungen und Unterverpackungen sind nicht zwingend normiert, können aber nach folgenden Regeln benannt werden. Im Englischen wird für (Protokoll)-

Dateneinheit der Begriff *Protocol Data Unit*, **PDU** verwendet.

- Die Dateneinheit der obersten Schicht (**Application**) heist einfach: **Data**.
- Die Dateneinheit der **Transport**-Schicht (Host-To-Host) ist das TCP-**Segment** oder das UDP-**Datagramm**.
  Adressiert werden die Segmente/Datagramme an die Dienste (Serverprozesse), auf dem Zielrechner. Die Dienste werden dabei über die **Portnummer** adressiert.
- In der **Internetworking**-Schicht heist die Dateneinheit **Packet**.
  IP-Pakete werden mit der weltweit eindeutigen IP-Adresse an einen bestimmten Empfangscomputer geschickt.
- Und auf der **Network-Access**-Schicht spricht man von **Frames**.
  Im lokalen Netz (z.B.) tragen die Frames Hardwareadressen.

Transportiert ein Ethernet-Frame ein IP-Paket, sind also 2 Adressen vorhanden: die hardwareunabhängige IP-Adresse **und** die Hardwareadresse der Ethernet-Schnittstellen.

2 Aufgaben der Schicht 2 (Data Link)

Die Data Link Schicht stellt für die höheren Schichten die Möglichkeit bereit, Daten über ein **gemeinsames Medium** austauschen zu können.
Hierfür muss die Data Link Schicht zwei Augaben erfüllen:

1. sie muss durch Kapselung der Daten (framing) den höheren Schichten Zugriff auf das Medium erlauben
2. sie muss den Zugriff auf das gemeinsam genutzte Medium verwalten

Wichtige Begriffe der Schicht 2:

Frame - Dateneinheiten der Schicht 2 heissen Frames (s.o.).

Node - Mit Node (Knoten) ist ein Teilnehmer eines gemeinsam genutzten Übertragungsmediums gemeint.

Media bzw. Medium - Das Übertragungs**medium**; z.B. eine elektrische Leitung (Koax, Twisted-Pair), ein Lichtwellenleiter, ein Hochfrequenz-Funkkanal, etc.

Network - Ein (physikalisches) Netzwerk entsteht, sobald zwei oder mehr Knoten (nodes) mit einem gemeinsamen Medium verbunden werden.

Mit diesen Begriffen kann man die Aufgaben der Schicht 2 nochmals definieren:

Der Data Link Layer ist dafür zuständig, Rahmen (Frames) zwischen den Teilnehmern (Nodes) über das physikalische Medium eines Netzwerks auszutauschen

2.1 Media Access Control und physikalische Adressen

In der Netzwerktechnik unterscheidet man zwischen **Bussystemen** und **Punkt-zu-Punkt** Verbindungen.

Bussysteme Bei Bussystemen sind mehrere Teilnehmer an einem **gemeinsam genutzten Medium** angeschlossen.

Mit speziellen Verfahren wird dabei sichergestellt, dass immer nur ein Teilnehmer senden kann (s.u.).

Ein Frame, den der Sender nun auf das Medium schickt, muss eine Ziel-Adresse haben, damit er bei mehreren angeschlossenen Empfangsknoten zum richtigen Empfänger gelangt. Zusätzlich tragen die Frames auch eine Absender–Adresse, auch als Quell-Adresse bezeichnet.

Die Quell-Adresse eines Frames stimmt dabei mit der eindeutigen Hardwareadresse des Senders überein. Entsprechend wird ein Frame auf dem Medium von dem Empfänger weiterverarbeitet, dessen Hardwareadresse mit der Ziel-Adresse des Frames übereinstimmt.

> Zur Unterscheidung dieser Adressen von den logischen (hardwareunabhängigen) Adressen der Schicht 3 und 4, nennt man sie **physikalische**-, **Hardware**- oder **MAC**-Adressen.

Punkt-zu-Punkt Verbindungen Besteht ein Netzwerk nur aus zwei Stationen, die mit einem Medium verbunden sind, spricht man von einer *Punkt-zu-Punkt Topologie* oder auch *Dual-Node*-Verbindungen.

2.1.1 Media Access Control bei gemeinsam genutzten Medien

Beim Zugriff auf das Medium unterscheidet man 2 Arten:

Kontrollierter Zugriff - beim kontrollierten Zugriff wird dafür gesorgt, dass immer nur ein Teilnehmer die Möglichkeit hat, auf das Medium zuzugreifen.

Bekannteste Beispiele: Token Ring (IEEE 802.5), FDDI (ANSI Standard X3T9.5)

Bei Token Ring und FDDI hat derjenige das alleinige Senderecht, der das sog. *Token* besitzt.

Das Token wird nach einer Sendung an die nächste Station weitergereicht und kreist auch bei Leerlauf (wennn niemand senden möchte) ständig von Station zu Station.

Wettbewerbszugriff - beim Wettbewerbszugriff (contention based access) hat jede Station **gleichbereichtigt** die Möglichkeit, bei Sendewunsch das (freie) Medium für sich zu beanspruchen.

Das Carrier Sense Multiple Access (CSMA) - Verfahren stellt dabei sicher, dass das Medium nicht im kompletten Chaos versinkt. Ein detaillierte Darstellung von CSMA ist im Abschnitt 3.8, im Kapitel über das Ethernet beschrieben.

2.1.2 Media Access Control bei Punkt-zu-Punkt Verbindungen

Bei einer Punkt-zu-Punkt-Verbindung ist die Zugriffssteuerung auf das Medium wesentlich einfacher als bei einem gemeinsam genutzten Bus: das Verbindungsmedium muss nicht mit anderen Teilnehmern geteilt werden und man benötigt keine Zieladressen, da es ja nur einen möglichen Empfänger gibt.

Beispiele für solche Layer-2-Protokolle sind:

PPP - Point-to-Point-Protocol (RFC 1661)

HDLC - High-Level Data Link Control (ISO 13239)

2.2 Richtungsabhängigkeit (Duplex)

Bei Punkt-zu-Punkt-Verbindungen und Bussystemen unterscheidet man in der Netzwerktechnik zwischen zwei Betriebsarten:

Halb-Duplex - Beide Knoten können senden und empfangen, aber sie tun das *niemals gleichzeitig*. D.h. Knoten1 sendet, Knoten2 empfängt **anschliessend** ist es umgekehrt usw. . Das klassische Ethernet (Bus-Topologie) arbeitet mit Halb-Duplex.

Voll-Duplex - Eine Übertragung kann *gleichzeitig* in beide Richtungen erfolgen. D.h. Knoten1 sendet und empfängt **gleichzeitig** die Sendung von Knoten2.

Voll-Duplex ist nur bei Punkt-zu-Punkt-Verbindungen möglich.

Voll-Duplex gibt es mit getrennten Medien für Senden- und Empfangen, z.B. bei moderner Twisted-Pair-Ethernet-Verkabelung.

Es ist aber auch technisch möglich, Voll-Duplex mit nur einem einzigen Adernpaar zu betreiben. Das alte, analoge Telefon ist ein Beispiel dafür. Man kann beim Telefonieren gleichzeitig sprechen und hören, obwohl das Gespräch nur über 2 Drähte übertragen wird. Auch bei 1000Base-TX (Gigabit-Ethernet auf Kupferleitungen) werden je zwei Drähte gleichzeitig für Senden und Empfangen verwendet.

3 Ethernet

3.1 Wer hat das Ethernet wann erfunden?

Das wohl weltweit erste LAN wurde von Robert Metcalfe und seinen Mitarbeitern vor über 30 Jahren am Xerox Palo Alto Research Center (PARC) [2] erdacht. Da die Entwicklung mehrere Jahre gedauert hat, kann man keine exakte Jahreszahl angeben. Ausserdem gründete Robert Metcalfe später die Firma 3Com und hat in Zusammenarbeit mit DEC, Intel und Xerox Ethernet zu einem Standard entwickelt. Dieser wird seit 1980 vom IEEE (Institute of Electrical and Electronics Engineers) in der **Arbeitsgruppe 802** weiterentwickelt.

Und wie gut diese Weiterentwicklung funktioniert sieht man daran, dass in der Zwischenzeit die Bitübertragungsrate von ursprünglich 3 MBit/s auf 10GBit/s gestiegen ist.

3.2 Grundprinzip von Ethernet

Das Grundprinzip von Ethernet ist ein von allen Teilnehmern gleichberechtigt genutztes Übertragungsmedium, das als Bus ausgeführt ist.

[2] Ebenfalls am PARC wurde das erste Betriebssystem mit GUI, Fenstern und Maus sowie die erste objektorientierte Programmiersprache *Smalltalk*, entwickelt.

3.3 Standards

Die für LANs zuständige Arbeitsgruppe trägt innerhalb der IEEE die Nummer **802**. Somit beginnen alle LAN-IEEE-Standards mit 802.

Der Standard für Ethernet ist IEEE802.3

3.4 Topologie von Ethernet

3.4.1 Bustopologie bei klassischem Ethernet

10Base5

> Das klassische Ethernet (10Base5) basiert auf einem **gemeinsam genutzten Bus** (shared medium) auf den alle *alle* Teilnehmer *gleichberechtigt* zugreifen können.

Das CSMA/CD-Verfahren (siehe Kap. 3.8) regelt den Zugriff auf den Bus.

> Übertragungen finden im Halb-Duplex-Betrieb statt.

Das Medium ist das sog. Yellow Cable, ein ca. 10mm dickes *Koaxialkabel* vom Typ RG 8. Aufgrund der sehr geringen Dämpfung der Signale auf diesem Kabel, sind bei 10Base5 Segmentlängen von 500m erlaubt.

Die Transceiver (Kombination aus RX und TX) wurden an dieses Kabel mit sog. *Vampire Taps* angeschlossen. Dabei wurde der elektrische Kontakt zur Koaxleitung mit einer Schraubklemme mit Metallspitzen hergestellt, die einfach durch den Kabelmantel hindurch bis zum Aussen- und Innnenleiter gebohrt wurden.

Die Einheit aus Kontaktierklemme und Sende- und Empfangselektronik wird *Medium Attachment Unit* genannt. Abb. 3 zeigt eine solche MAU.

Von der MAU verläuft je ein Adernpaar für TX (Senden) und RX (Empfangen) zum Hostcomputer. Ein Blockschaltbild eines Transceivers zeigt Abb. 4. An der MAU aus Bild 3 erkennt man einen Sub-D-Stecker für die Verbindung zum Host.

10Base2 Das YellowCable ist sehr teuer und mit den Vampire-Taps etwas unflexibel. Eine preiswertere Variante ist 10Base2: hier wird mit dem dünneren Koaxialkabel RG58 gearbeitet.

Da RG58 nicht so gute elektrische Eigenschaften besitzt wie das YellowCable (RG 8), ist bei 10Base2 die Segmentlänge auf 185m beschränkt.

Bei 10Base2 werden die Transceiver auch direkt als Einsteckkarten in die Hostcomputer eingebaut und mit BNC-Steckern mit dem Bus verbunden. Dazu muss beim Neuanschliessen eines Hosts der Bus kurz unterbrochen werden, was evtl. zu Datenverlusten führt.

> Bei 10Base5 und 10Base2 ist die logische und die pyhsikalische Topologie des Netzes ein Bus.

3.4.2 Übergang zur physikalischen Sterntopologie

Koaxialkabel sind zwar elektrisch haushoch überlegen (sehr hohe Bandbreite, geringe Dämpfung), wurden aus Kostengründen jedoch vom TwistedPair-Kabel verdrängt.

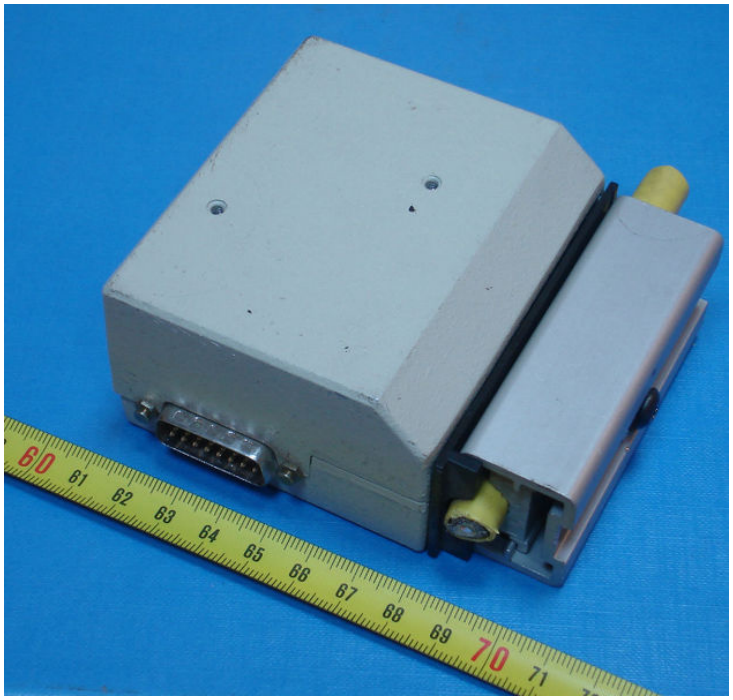

Abbildung 3: Medium Attachment Unit (Bildrechte: Creative Commons)

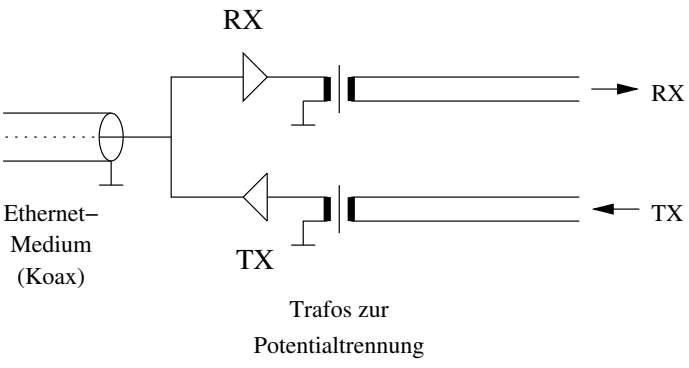

Abbildung 4: Blockschaltbild eines Transceivers

Dabei stellen die TwistedPair-Kabel einfach nur die Verbindung der RX- und TX-Ports eines Transceivers (Abb. 4) mit einem bis zu 100m entfernt stehenden Hosts dar.

Da man in einem Netz viele Hosts hat, benötigt man die entsprechende Anzahl an Receivern. Diese werden zusammen in ein spezielles Gerät, den sog. **Hub** eingebaut. Hier wieder das übliche Blockschaltbild: Abb. 5.

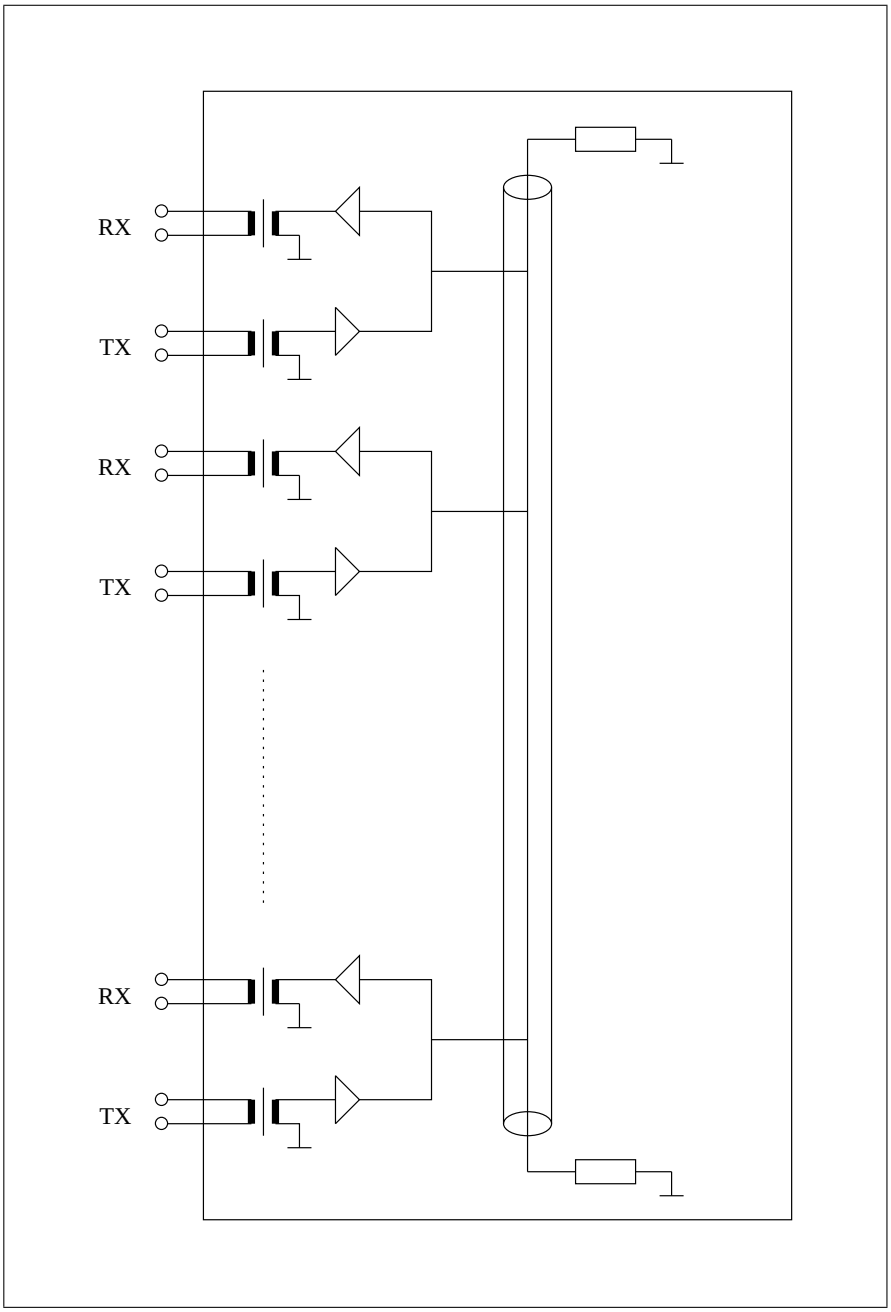

Abbildung 5: Blockschaltbild eines Hubs (Multiport-Transceiver)

Ein 10BaseT-Netz (T = TwistedPair) ist also nichts anderes als ein 10Base5-Netz, bei dem das eigentliche Ethernet-Medium auf die Grösse eines Hubs zusammengeschrumpft ist und bei dem dann alle Transceiver (MAUs) in einem Gehäuse, eben dem Hub, vereint sind.

Da wir im Inneren des Hubs nach wie vor eine Busleitung haben, ist die *logische Topologie* bei 10BaseT/100BaseT immer noch ein *Bus*.

Pysikalisch sind die Netzteilnehmer nun jedoch in Sterntopologie angeordnet.

Bei TwistedPair-Netzen **mit Hubs** hat man physikalisch eine Sterntopologie und logisch nach wie vor eine Bustopologie.

Die Übertragung findet ebenfalls noch im Halb-Duplex-Verfahren statt.

Hubs arbeiten im Halb-Duplex-Betrieb. Alle Ports eines Hubs bilden eine Kollisionsdomäne.

3.4.3 Netze mit Switches

Ersetzt man den Hub durch einen Switch, bleibt die physikalische Topologie eralten. Da ein Switch aber auf Layer2 arbeitet und die MAC-Adressen der Frames analysiert, kann er eine Punkt-zu-Punkt-Verbindung zwischen zwei Hosts aufbauen.

3.5 Anpassung an das OSI-Modell mit Sub-Layers

Da das Ethernet älter ist als das OSI-Schichtenmodell, passt es nicht hundertprozentig in das Schichtenmodell: Ethernet arbeitet gleichzeitig auf dem *Physical Layer* und dem *Data Link Layer*.

Nun ist es so, dass die Trennlinie zwischen den hardwareabhängigen Teilen und den rein in Software realisierten Komponenten des Schichtenmodells nicht zwischen Schicht 2 und 3 sondern *innerhalb* von Schicht 2 (Data Link) verläuft.

Abb. 6 zeigt das OSI (Open System Interconnect) - Schichtenmodell mit der Trennlinie zwischen Hard- und Software.

3.5.1 Logical Link Control (LLC)

Damit Ethernet an das OSI-Schichtenmodell angepasst werden konnte, wurde der Data-Link-Layer nochmals in zwei sog. **Sublayer** geteilt (vgl. Abb. 6).

Die LLC-Schicht ist als *Softwareanwendung* ausgeführt.

Bei einem Computer wird die LLC-Schicht von der *Treibersoftware* der Netzwerkkarte dargestellt.

Aufgaben der LLC:

- stellt Verbindung zu der darüberliegenden Schicht her
- verpackt die aus der darüberliegenden Schicht kommenden Datenpakete in einen Rahmen

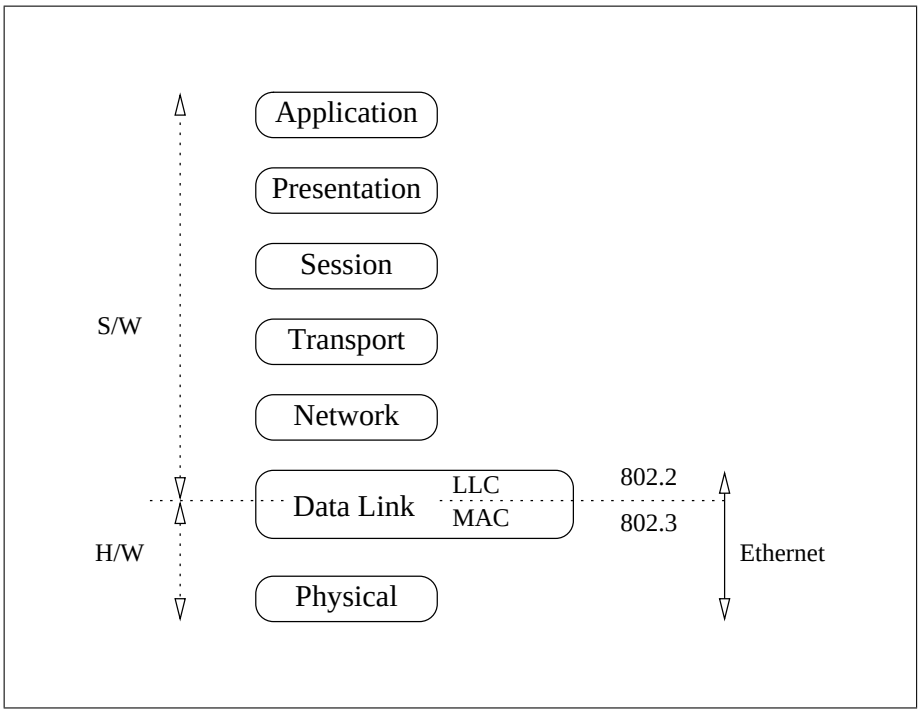

Abbildung 6: OSI-Schichtenmodell mit Ethernet

- identifiziert, mit welchem Protokoll die Pakete aus der darüberliegenden Schicht (Network) erstellt wurden (z.B. IP = 0800)
- macht alle darüberliegenden Protokolle **hardwareunabhängig**

3.5.2 Media Access Control (MAC)

Die MAC-Schicht ist für den Zugriff auf das physikalische Medium zuständig und wird *kommplett von der Hardware* ausgeführt. Diese Hardware ist z.B. die Netzwerkkarte oder die On-Board-Netzschnittstelle eines Rechners.

Bei einem Computer wird die MAC-Schicht von der *Netzwerkkarte* dargestellt.

Die zwei Aufgaben der MAC-Schicht:

Data Encapsulation - Daten Kapselung. Data Encapsulatin besteht wiederum aus folgenden 3 Teilen:

- Frame Delimiting: Begrenzung der Ethernet-Frames mit **Header** und **Trailer** (Framing der Layer 3 - PDU)
- Addressing: **Adressierung der Frames mit Quell- und Zieladresse**.

Ethernet verwendet je 48 bit (6Bytes) lange Adressen.

Um diese Adressen von der logischen IP-Adresse zu unterscheiden spricht man von **MAC-Adressen**. Auch der Ausdruck physikalische Adresse ist üblich.

Anhand der Adresse wird bereits in der MAC-Schicht entschieden, ob ein Frame überhaupt an die höheren Schichten weitergereicht werden soll.

- Error Detection: Fehlererkennung durch CRC-Prüfsumme. CRC = Cyclic Redundancy Check

Media Access Control - Zugriffskontrolle auf das Medium (Bus).

- Einleiten einer Übertragung
- Platzierung und Entfernen von Frames auf dem Bus
- Fehlerbehandlung nach Kollisionen

3.6 Ethernet-Frame-Typen

Insgesamt gibt es vier verschiedene Arten von Ethernet-Frames von denen nur noch 2 eine Bedeutung haben.

3.6.1 Ethernet-II

Ethernet-II wurde von DEC, Intel und Xerox entwickelt. Ethernet-II-Frames heißen desshalb auch **DIX-Frames**.

Die Bytes unmittelbar nach den je 6 Bytes langen MAC-Adressen (Byte 13 und 14) enthalten den *Ethertype*, d.h. den Typ des eingeschachtelten Layer3-Pakets. Ist dies z.B. ein IP-Paket, hat das Typ-Feld den Wert 0x0800.

Die Frame-Länge wird nicht mit übertragen. Das Frame-Ende wird dabei bitgenau signalisiert.

Abb. 7 zeigt einen solchen Rahmen.

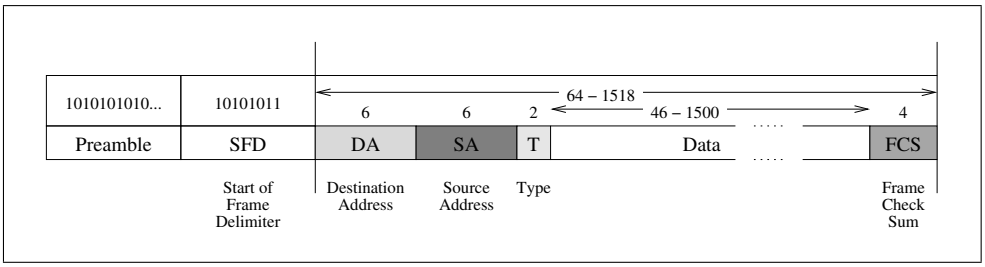

Abbildung 7: Ein Ethernet-II-Frame

3.6.2 Ethernet IEEE 802.3

Bei einigen Ethernet-Versionen enthielten die Bytes unmittelbar nach den MAC-Adressen nicht den Ethertype (s.o.) sondern die Länge des Frames.

Beim Standard IEEE802.3 kann dieses Feld (Bytes 13 und 14) die Länge **oder** den Ethertype enthalten.
Dabei wird so unterschieden:

- ist der Wert der in Byte 13 und 14 enthaltenen Zahl kleinergleich 1536 (Wert ¡ 0x0600), wird er als **Frame-Länge** interpretiert.
- ist der Wert grösser 1536=0x600, enthält das Feld einen *Ethertype*.

Die Bytes 13 und 14 bei Ethernet IEEE802.3 werden als Type/Length bezeichnet.

3.7 Felder der Ethernet-Frames

3.7.1 MAC-Adressen

Im Ethernet-Frame steht zuerst die 6 Bytes lange MAC-**Ziel**adresse, direkt danach folgt die 6 Bytes lange MAC-**Quell**adresse.

Dabei bestehen die MAC-Adressen wiederum aus zwei Teilen:

- dem Organization Unique Identifier, der OUI; dieser ist 3 Bytes lang
- einer Vendor Assigned Number, faktisch der Seriennummer des Netzwerkadapters; ebenfalls 3 Bytes lang

Beispiel:
MAC-Adresse = 00:1b:63:9f:a2:3c
OUI = 00:1b:63 = Apple
VA = 9f:a2:3c

Bitreihenfolge bei der Übertragung der MAC-Adresse Bei der Analyse eines seriellen Ethernet-Signals muss man beachten, dass die Adresse mit dem **niederwertigsten Bit beginnend** übertragen wird: das LSB = Least Significant Bit wird zuerst gesendet.
Die übliche Darstellung von MAC-Adressen wird **kanonisches Format** oder auch Ethernet-Format genannt. Bei diesem Format ist schon berücksichtigt, dass das LSB zuerst gesendet wird.
Hier ein Beispiel, wie eine Adresse im Ethernet-Format bitweise auf die Leitung gesendet wird:

Canonical:	12	34	56	78	9a	bc
Hex-Bin:	00010010	00110100	01010110	01111000	10011010	10111100
TX:	01001000	00101100	01101010	00011110	01011001	00111101

Unicast, Multicast, Broadcast Es gibt 3 verschiedene Arten von Adressen:

Unicast Unicast Adressen sind der Normalfall: der Frame enthaelt als Zieladresse die Hardware-Adresse der Netzwerkschnittstelle, an die der Frame geschickt wird.

Broadcast Ein Broadcast-Frame ist ein Frame, der an alle Teilnehmer des physikalischen Netzes adressiert ist.

Dazu wird als Ziel-MAC-Adresse die spezielle Broadcast-Adresse ff:ff:ff:ff:ff:ff verwendet.

Alle Netzwerkteilnehmer, die solche Frames empfangen bilden eine **Broadcast-domäne**.

Multicast Multicast-Frames werden an eine Gruppe von Empfängern adressiert.

Dazu wird eine MAC-Adresse verwendet, die im Bereich 01:00:5e:00:00:00 bis 01:00:5e:7f:ff:ff liegt.

Die niederwertigen 23 bit der MAC-Adresse enthalten die 23 niedrigsten Bit der zugehörigen Multicast-IP-Adresse, die von diesem Frame transportiert wird. Das übrigbleibende Bit 24 ist immer 0 (00:00:00 bis **7**f:ff:ff)

3.7.2 Type-Feld

Die genaue Bedeutung des Type-Felds wurde bereits in Kap. 3.6.2 dargestellt. Hier noch eine Liste wichtiger Ethertypes:

Type (hex)	Bedeutung	verantwortlich
0000-05DC	IEEE802.3 Length Field	XEROX
0800	Internet IP (IPv4)	IANA
0806	ARP	IANA
809B	Appletalk	XEROX
8100	IEEE 802.1Q VLAN-tagged frames	IEEE
814C	SNMP	JKR1
880B	PPP	IANA
8863	PPPoE Discovery Stage	RFC2516
8864	PPPoE Session Stage	RFC2516
9000	Loopback	XEROX

Tabelle 1: Wichtige Ethertypes

3.7.3 Präambel und Start Frame Delimiter

Präambel und Start-Frame-Delimiter bilden zusammen folgende Bitfolge:

```
10101010 10101010 10101010 10101010 10101010 10101010 10101010 10101011
```

also

Die Präambel besteht aus 7 Bytes mit dem Wert 10101010 gefolgt vom Start-Frame-Delimiter 10101011

Diese wechselnde Bitfolge war früher notwendig, damit sich die Empfangselektronik auf den Sendetakt synchronisieren konnte.

Durch die Unterbrechung des ständigen, monotonen 0-1 Wechsels am Ende des SFD wurde der Rahmenanfang bitgenau festgelegt.

Bei moderner Hardware (100BaseT, 1000BaseT) wird Präambel und SFD nur noch aus Kompatibilitätsgründen beibehalten.

3.7.4 Frame Check Sequence, FCS

Die Frame Check Sequence ist eine 32-Bit lange Prüfsumme, die über den Frame **ohne** Präambel und SFD berechnet wird.

Berechnet wird die FCS mit dem CRC-Verfahren (CRC = Cyclic Redundancy Check). Dieses Verfahren lässt sich leicht mit Hardware realisieren.

3.7.5 Padding

Padding-Bytes werden benötigt, wenn der Rahmen weniger als 46 Bytes Nutzdaten enthält und somit kleiner als die Minimalgrösse von 64 Bytes wird. Die PAD-Bytes stehen direkt vor der FCS.

3.8 CSMA/CD

CSMA/CD steht für *Carrier Sense Multiple Access / Collision Detection*:

Bei Ethernet als Bussystem sind alle Stationen parallel an einer Leitung angeschlossen. Dabei gibt es keinen Busmaster, d.h. alle Stationen sind gleichberechtigt: **Multiple Access**

Das Verfahren CSMA/CD sorgt nun dafür, dass immer nur eine Station Daten senden kann. CSMA/CD kommt dabei ohne zusätzliche Kommunikation zwischen den Stationen aus.

Das Prinzip: eine Station, die senden möchte, hört zunächst den Bus ab um festzustellen, ob dieser frei ist. Dazu wird einfach gemessen, ob ein Trägersignal auf dem Bus anliegt: **Carrier Sense** [3] Nun kann der Fall eintreten, dass zwei Stationen fast zeitgleich beginnen, ein Ethernet-Paket zu senden. Da sich die Signale auf dem Bus nur mit ca. $1.8 \cdot 10^8 m/s$ ausbreiten, werden sie sich nach einer bestimmten Laufzeit vermischen, wobei die Daten natürlich komplett verfälscht und damit unbrauchbar werden. So etwas wird als *Kollision* bezeichnet. Tritt so eine Kollision auf, wird sie von den sendenden Stationen erkannt: **Collision Detection**.

Erkennt eine Station eine Kollision, hört sie sofort mit der Sendung auf. Damit dann, wenn der Bus wieder frei ist nicht wieder die beiden Sender gleichzeitig beginnen, hat jeder Teilnehmer einen Zufallsgenerator, der eine zufällige Wartezeit bis zur nächsten Sendung erzeugt.

Die Kollisionserkennung erfolgt elektronisch: bei einer Kollision treten erhöhte Spannungspegel auf der Leitung auf, die von einer Schaltung erkannt werden.

CSMA/CD funktioniert nach folgendem Algorithmus:

[3] Dies war beim ursprünglichen Ethernet technisch einfach zu lösen, da zusammen mit den Daten immer der Bittakt von 10MHz mitgesendet wurde. Erreicht hat man dies durch Verwendung des **Manchester-Codes**, bei dem immer zur Bitmitte ein Pegelwechsel auf der Leitung auftritt.

Hauptprozedur

1. Der Frame ist zusammengestellt und absendebereit.
2. Ist das Medium frei? Wenn nein, warte bis es frei wird und warte den *Inteframe-Gap*-Zeitraum ($9.6\mu s$ bei 10MBit/s) ab.
3. Beginne mit der Übertragung.
4. Ist eine Kollision aufgetreten? Wenn ja, gehe zur Kollisions-Prozedur.
5. Setze die Zähler für Retransmission zurück und beende die Rahmenübertragung

Kollisionsprozedur

1. Fahre mit der Übertragung fort, bis die minimale Paketlänge (64 Bytes) erreicht ist. Sende zu diesem Zweck das *jam signal* um sicherzustellen, dass alle Stationen die Kollision erkennen.
2. Erhöhe den Retransmission-Zähler
3. Wurde die maximale Anzahl an Übertragungsversuchen erreicht? Wenn ja, starte keine weitere Retransmission.
4. Warte eine zufällige *Backoff*-Zeit ab und fahre anschliessend mit Punkt 1 der Hauptprozedur fort.

Hier noch ein Modell des CSMA/CD-Verfahrens:

Wir befinden uns auf einer Pary. Alle Gäste reden miteinander und teilen sich dabei ein *gemeinsames Medium* (Shared Medium), die Luft.

Bevor jemand anfängt zu reden, wartet er höflich, bis niemand anderes mehr etwas sagt (Carrier Sense).

Falls zwei Leute *gleichzeitig* zu reden beginnen, hören beide auf und warten eine kurze, *zufällige* Zeitspanne (Backoff Algorithmus). Das zufällig lange Warten stellt sicher, dass beide nach einer Kollision nicht wieder gleichzeitig losreden.

Die Backoff-Zeit wird *exponentiell* vergrössert, wenn mehr als ein fehlerhafter Übertragungsversuch auftritt (e-Funktion über die Anzahl der Fehlversuche).

3.9 CSMA/CA

CSMA/Collision Avoidance wird u.a. bei WLAN-Standards eingesetzt (802.11).

Bei CSMA/CA schickt eine Station bevor sie die eigentliche Nachricht auf das Medium sendet eine *Benachrichtigung* (notificaton) an die anderen Stationen.

Ist das Medium belegt, wird eine Sendung um eine zufällige Zeit *vertagt*. Damit wird vermieden, dass nach Freiwerden des Mediums zwei Stationen *gleichzeitig* versuchen, eine Benachrichtigung zu senden.

Ganz auschschalten lässt sich eine Kollision damit natürlich immer noch nicht: es könnte der unwahrscheinliche Fall eintreten, dass zwei sendewilligen Stationen ihre Sendung zufällig um die selbe Zeit vertagen.

CSMA/CA wird bei WLANs angewendet, da WLAN-Stationen *nicht gleichzeitig* senden und empfangen können. Während einer Sendung können sie also nicht das Medium abhören was notwendig ist, um eine Kollision zu erkennen.

3.9.1 Kollisionsdomänen und Slot-Time

Ein Ethernet-Paket muss minimal 64 Bytes = 512 bit lang sein. In diesem Beispiel soll es mit 100 MBit/s gesendet werden. Dann dauert die Aussendung des gesamten Pakets $t = 5.12\mu s$. Mit der Ausbreitungsgeschwindigkeit von $v = 1.8 \cdot 10^8 m/s$ ist das erste vom letzten Bit

$$s = t \cdot v = 921m$$

entfernt.

Im schlimmsten Fall (worst case) tritt die Kollision dicht bei einer weit entfernten Station auf. Die Zeit, die ein Signal braucht bis es dorthin und als Kollisionssignal wieder zurück läuft, nennt man *Round Trip Delay*. Dieser Round Trip Delay muss also immer kleiner sein als die Zeit $t = 5.12\mu s$, damit eine Kollision auch im schlechtesten Fall erkannt wird.

Oder anders ausgedrückt: die Stationen dürfen nicht weiter als $921m/2 = 460m$ voneinander entfernt ein.

In der Praxis hat man noch ca. den Faktor 5 als Sicherheit eingebaut: bei 100BaseT ist die **maximale Länge eines Segments 100 m**. Unter Segment versteht man hier den Teil eines Netzes, in dem Kollisionen erkannt werden müssen. Man nennt diesen Teil auch *Kollisionsdomäne*.

Z.B. bilden alle Ports eines Hubs eine Kollisionsdomäne. Bei einem Switch ist das anders: dieser trennt Kollisionsdomänen voneinander indem er die Pakete analysiert. Dazu später mehr.