Kommunikationsnetze
- Zusammenfassung -

Gernot Fussan,
Ulf Wulfert

Technische Universität Berlin
Wintersemester 1998 / 99
Sommersemster 1999

Inhalt

Quellen

Skript Prof. Dr-Ing. Adam Wolisz
Unterlagen zu der Vorlesung Kommunikationsnetze 1998 / 99

Buch Andrew S. Tanenbaum
Computernetzwerke, 3. Auflage, Prentice Hall

1. Kommunikationsgrundlagen

1.1. Theoretische Grundlagen

Signalgeschwindigkeit / Baud-Rate
- Beschreibung Änderung der Spannung eines Signals innerhalb einer Sekunde
- Einheit Baud

Bitrate
- Beschreibung Anzahl der Bits die pro Sekunde übertragen werden.
- Einheit bps (Bits per second)

Bandbreite
- Beschreibung Frequenzbereich in dem Kommunikation stattfindet
- Einheit Hz (Hertz)

- Beschreibung Gesamtkapazität des Netzes
- Einheit bps (Bits per second)

Dämpfung / Rauschabstand
- Beschreibung Verlust bei Übertragungen
- Formel dB = 10 log(S / N) mit S ... Signalstärke, N ... Rauschstärke
- Einheit dB (Decibel)

1.2. Analoge Signale

Nyquist-Theorem

Wird ein Signal durch eine Tiefpaßfilter mit der Bandbreite H geführt, kann das Signal vollständig durch 2H Abtastungen pro Sekunde wieder hergestellt werden.
max. Datenrate = $2H \log_2 V$ $^{Bit}/_{Sekunde}$ mit V Anzahl von diskreten Stufen

Shannon

Kanal mit thermischen Rauschen
max. Bits pro Sekunde = $H \log_2 (1+ S / N)$ mit S/N Rauschabstand

1.3. Digital Signalcodierung

- Abbildung von Bits auf elektrische Signale

1.3.1. Kriterien zur Auswahl eines Verfahrens

- Signalspektrum
- Clock - Rekonstruktion beim Empfänger
- Fehlererkennung
- Unempfindlichkeit gegenüber Rauschen

1.3.2. Kodierungsverfahren

NRZ-L (Nonreturn-to-Zero-Level)	0 = high, 1 = low Polarität ist wichtig
NRZI (Nonreturn-to-Zero-Inverted	Pegeländerung gibt Wert an keine Änderung = 0, Änderung = 1
Bipolar-AMI	kein Signal = 0, Änderung = 1 (abwechselnd nach high und low)
Pseudoternary	kein Signal = 1 Änderung = 0 (abwechselnd nach high und low)
Manchester	0 = Änderung von high → low in Intervallmitte 1 = Änderung von low → high in Intervallmitte
differential Manchester	Änderung am Intervallanfang= 0, keine Änderung am Intervallanfang = 1

	Wechsel immer in Intervallmitte selbstsynchronisierend
Miller	1 = Änderung in Intervallmitte 0 = keine Änderung wenn danach 1 folgt 0 = Änderung am Ende gefolgt von 0

1.3.3. mBnL Codes

- m Anzahl der Bits
- n Anzahl der Änderungen der Level
- L Anzahl der Level
- Bsp.: 4B3T
  - 4 Inputbits
  - 3 Impulse
  - 3 (Tertiär) verschiedenen Leveln dargestellt

1.3.4. Entfernen von langen konstanten Signalen

- B8ZS: bei 8 Bit mit gleichem Wert wird andere Kodierung gewählt, so daß der Pegel wechselt
- HDB3: Unterscheidung nach gerader oder ungerader Anzahl von gleichen Bit

1.3.5. CMI Code

- coded mark inversion (CCITT G.703)
- zweistufiger NRZ-Code

1.3.6. 4b / 5b Code

- 4 Bits werden mit 5 Bits kodiert
- es stehen Data-, Line State- und Controlsymbole zur Verfügung

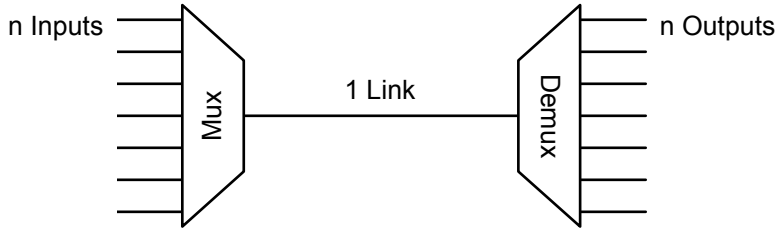
1.4. Bit-Synchronisation

- Empfänger muß Signal an richtiger Stelle abtasten → Synchron zum Sender
- *explicit clock signal*: Sender und Empfänger sind durch direkte Taktleitung verbunden
- *clock encoding*: Takt ist im Signal untergebracht, für Synchronisation wird Bitmuster (Präambel) gesendet

1.4.1. Digital phase-locked-loop concept

- Ziel: Synchronisierung im normalen Bitstrom unterbringen
- Vor.: stabile Uhr beim Empfänger, keine Start/Stop-Bits
- DPLL: Signal enthält periodische Signalwechsel, dadurch kann der Empfänger seine Uhr einstellen

1.5. Multiplexing



- verschiedene Kanäle werden über einen Link übertragen

1.5.1. Frequency division multiplexing FDM

- Aufteilung der gesamten Bandbreite in Kanäle → Frequenzbereiche (durch Filter)
- jeder Benutzer erhält einen exklusiven Kanal
- bei Glasfaser Wavelength-Multiplexing : Kanal entspricht Licht mit bestimmter Wellenlänge
- Übertragungsart: Breitband

1.5.2. Time division multiplexing TDM

- eingehender Kanal wird in Zeitscheiben zerlegt
- jeder Benutzer hat für kurze Zeit die volle Bandbreite
- Synchronous TDM : immer gleiche Reihenfolge, auch wenn auf Kanal nicht gesendet wird
- Statistical TDM : nur Kanal mit Daten bekommt Zeitscheibe zugeteilt, sonst wird er ausgelassen
- Übertragungsart: Basisband

1.6. Switching

- Ziel ist Herstellung einer Verbindung zwischen zwei Teilnehmern aus verschiedenen Kommunikationssystemen.
- Verbindung aller Teilnehmer durch direkten Draht ist nicht praktikabel

1.6.1. Circuit Switching (Leitungsvermittlung)

- zwischen Vermittlungen werden bei Bedarf Leitungen geschaltet.
- für die Zeit der Verbindung besteht eine physische Leitung
- einige der Leitungen können auch gemultiplext sein
- Aufbau der Verbindung benötigt Zeit (Einrichtung eines End-to-End-Paths)
- Verbindung muß explizit abgebaut werden
- keine Überlastung nach Aufbau, da Ressourcen reserviert sind

Space Division Switching

- direkter Pfad wird geschaltet durch a) Rotationsswitch oder b) Matrix-Switch
- non-blocking: jede Inputleitung kann Verbindung zu beliebiger Outputleitung bekommen (wenn diese nicht schon belegt ist)
- bei Matrix-Switch werden N*N Crosspoints benötigt → zuviel

Bus Time Division Switching

- Inputlines werden über Bus mit Ausgängen verbunden
- bei N Eingängen und N Ausgängen werden 2*N Schalter benötigt
- Busgeschwindigkeit muß N-fache Geschwindigkeit der Eingangsleitungen haben → non-blocking

Memory Time Division Switching

- gemeinsamer Speicherbereich wird benutzt
- pro Takt wird je einmal geschrieben und gelesen
- benötigte Geschwindigkeit: 1 / 2* Slotrate

→ bei großer Anzahl von Leitungen sind Space/Time Division Switching nicht praktikabel da

a) große Anzahl von Crosspoints bzw.
b) hohe Geschwindigkeit des Busses bzw. des Speichers benötigt wird

1.6.2. Multiple Stage Space Switches

- für Circuit Switching / Leitungsvermittlung
- 3-stage space switching matrix : 3 Stufen von je k Switchen werden hintereinander geschaltet mit n Eingängen
- Voraussetzungen damit 3S nicht blockiert
  - jeder einzelnen Switch muß non-blocking sein
  - Anzahl der Switches muß gleich 2n-1 sein
- Lee Graphs: Wahrscheinlichkeit dafür das Link blockiert ist
- Space-Time-Space Switch
- Time-Space-Time Switch

1.6.3. Message Switching

- keine permanente Verbindung, nur im Switch sind alle Input- und Outputleitungen direkt verbunden

- Nachrichten werden zwischengespeichert und erst weitergeleitet, wenn sie komplett eingegangen sind
- Nachrichten benötigen Header mit Routing-Informationen
- es werden große Zwischenspeicher benötigt
- keine Startverzögerung (Verbindungsaufbau)

1.6.4. Packet Switching

- Aufteilung der Nachrichten in Pakete mit begrenzter Größe
- können weitergeleitet werden bevor ganze Nachricht abgeschickt wurde
- Pipeline-Effekt : bessere Auslastung der Leitung da keine großen Wartezeiten
- Overhead entsteht bei Aufteilung und Zusammensetzung der Nachrichten
- beliebiger Weg für Pakete

1. Generation	CPU übernimmt Switching der Pakete
2. Generation	jeder Eingang hat eigene Queue und Prozessor und sendet Pakete direkt über einen shared bus an andere Ausgänge
3. Generation	Switching Fabric erlaubt parallelen Transport verschiedener Pakete

Siehe auch X.25 / Packet Switching

1.6.5. Vergleich

Circuit Switching	Packet Switching
festgelegte Verbindung	beliebiger Weg, verschiedene Routen
expliziter Verbindungsauf- und abbau	kein Auf- und Abbau notwendig
Überlastung nur bei Verbindungsaufbau	Überlastung bei jedem Paket möglich
keine Zwischenspeicherung	Zwischenspeicherung in Warteschlangen
feste verfügbare Bandbreite (Verschwendung möglich)	Bandbreite je nach Belastung des Netzes

1.7. Routing

- festlegen eines Weges (z.B. für ein Paket) durch ein Netz
- Source Routing: an der Quelle wird bereits der Weg ermittelt
- Hop-by-Hop Routing: an jedem Knoten wird der weitere Weg ermittelt
- Broadcast-Methoden: Flooding, Spanning Tree
- Flooding: Paket wird von jedem Node weitergeleitet
- Spanning Tree: Paket wird nur an die Nachfolgeknoten weitergeleitet, die es noch nicht bekommen haben
- Broadcast Tree: Jede Station kennt nur die Vorgänger- (Eltern-) Knoten. Broadcast : Flooding von Vorgänger zu Nachfolgern. Routing an ein Ziel: da jeder Knoten seine Vorgänger kennt kann Information zugestellt werden

1.7.1. Eigenschaften eine Routing-Algorithmus

- robust: kann auf Änderungen / neue Bedingungen im Netz reagieren
- stabil: eine kleine Änderung der Bedingungen, erzeugt nur eine kleine Veränderung der Routing-Entscheidung
- fair: kein Nutzer wird bevorzugt, gleiche Bedingungen wie Verzögerung und Übertragungsrate für alle
- optimal: maximiert die Leistung des Netzes
- Optimalität und Fairness können im Widerspruch stehen (z.B. Kosten)

1.7.2. Dijkstra's Algorithmus

- suche die kürzesten Wege von einem Knoten zu allen anderen Knoten → least cost routing
- gegeben ist ein Graph mit bewerteten Kanten (Kosten, Entfernung)
- D : Array von Entfernungen/Kosten, N: Menge der Knoten, a: Startknoten

INIT: D[a]=0, D[i]=∞ für alle i∈N \ {a}

wenn N=∅ dann Ende
    wähle Knoten i∈N mit geringsten Kosten
    für jede Kante zu allen Knoten in N {
        wenn Kosten durch neuen Weg kleiner sind
        → neue Kosten speichern
    }
    entferne Knoten i aus N (abgearbeitet)

1.7.3. Bellman-Ford Algorithmus

- suche den kürzesten Pfad von einem Quellknoten aus, und bestimme dabei die Kosten in Abhängigkeit von der Anzahl der Teilstücke (Links)
- bestimme den kürzesten Weg unter der Bedingung das der Weg aus einem Link besteht, danach aus 2 Links usw.
- es wird immer der Weg gespeichert, der am preiswertesten ist
- das Verfahren endet, wenn sich die Kosten nicht mehr ändern

1.8. Error Control

1.8.1. Fehler-Hypothese

Annahme: unabhängige Bitfehler mit Wahrscheinichkeit p, fehlerfreies Frame: $\mathbf{(1-p)^n}$, wenn **p<<n**, dann $\mathbf{1-(1-p)^n \sim p*n}$ (mind. ein Fehler im Frame).

In Wirklichkeit treten Fehler in Bursts auf, Gründe:
- elektrische Schwankungen
- Verlust der Bitsynchronsation
- cross-talk

Ist schwerer zu simulieren und zu erkennen, weil weniger Frames betroffen

1.8.2. Fehlererkennung und Fehlerbeseitigung

- offene Schleife: keine Rückmeldung vom Empfänger
- geschlossene Schleife: Rückmeldung vom Empfänger
- redundante Informationen einfügen (Block codes)
    - m Bits werden übermittelt, alle Kombinationen sind erlaubt → keine Erkennung
    - k Bits werden mit n Bits kodiert (n-k Bits sind redundant), nur 2^k von 2^n Codes sind erlaubt → Erkennung bei unerlaubtem Code
    - Korrektur: immer das erlaubte Codeword, was am dichtesten am empfangenen CW dran ist (wenigster Bitunterschiede)
    - simple Parity: Kontrolle auf Anzahl von „1“ (gerade oder ungerade) → Erkennung von ungerader Anzahl von Fehlern
- Hamming-Abstand: Anzahl der Bits, um welche sich die Codewords unterscheiden
    - Minimal-Hamming-Abstand (d_{min}): der kleinste über alle Paare, dann:
    - Detektion, wenn Anzahl der Fehler kleiner als d_{min}
    - Korrektur, wenn Anzahl der Fehler kleiner als $(d_{min} – 1) / 2$
- Polynomcode (CRC): Anhängen einer Prüfsumme (normiert: CRC-12, -16, -CCITT)

1.8.3. Polynomial Codes

- Generation Polynom $\mathbf{G_r(x)}$: Rank **r**, = Anzahl der Redundanz-Bits, muß Sender und Empfänger bekannt sein
- Klartext mit Länge **k** wird als Polynom dargestellt (Bits = Koeff’s: 11001 ~ $x^4 + x^3 + x^0$) = $\mathbf{V_{k-1}(x)}$
- $\mathbf{x^r V_{k-1}(x)}$, soll heißen: zusätzlich **r** Bits
- dieses wird durch $\mathbf{G_r(x)}$ mit Mod 2 geteilt, Ergebnis ist ein Quotient **Q(x)** und ein Rest **R(x) – a,** dessen Rank nicht größer als **(r-1)** ist
- Die Summe $\mathbf{T(x) = x^r V_{k-1}(x) + R(x)}$ wird gebildet, indem die letzten Bits von $\mathbf{x^r V_{k-1}(x)}$ durch **R(x)** ersetzt werden

- Diese zusätzlichen Bits werden CRC genannt
- Fehler: Wenn Empfänger den empfangenen Rahmen + Rest durch Generator Polynom teilen kann
- Wichtig: ein gutes $\mathbf{G_r(x)}$ definieren
- Eigenschaften für ein gutes G(x): (Detektion von)
  - Einzelfehler
  - Paare isolierter Fehler: G(x) nicht teilbar durch **x** oder $\mathbf{x^k+1}$
  - ungerade Anzahl von Fehlern: G(x) hat **x+1** als Faktor
  - alle Fehler-Bursts bis Länge = **r**
  - Fehler-Bursts der Länge exakt gleich to **r+1:** mit $\mathbf{P = 1- (1/2)^{(r-1)}}$
  - Fehler-Bursts der Länge größer als **r+1:** mit $\mathbf{P = 1- (1/2)^{(r)}}$
  - Realisierung über Shift-Register
  - Checksummen-Erstellung und Kontrolle on the fly, während Übertragung, Empfang auf Serieller Verbindung
  - bis 140MBits/Sec möglich

1.8.4. Vergleich open und closed loop

Open loop (Forward Error Correction FEC)	**closed loop** (Automatic Repeat Request ARQ)
+ keine Verzögerung bei der Übertragung	- zusätzlicher Austausch von Daten zwischen Sender und Empfänger
- höhere Datenmenge (Overhead)	- Verzögerung bis korrigierte Daten eintreffen
- schwierige Implementierung	+ einfache Implementierung
	+ wenig Overhead

1.8.5. Framing

- Grund: Physical Layer unterstützt nur bitsynchronisation → Daten bestehen aber aus mehreren Bits → Blöcke übertragen, Framing

Möglichkeiten

- Pause (time gaps): nicht gut, kann durch physical Layer „gequetscht“ werden
- Physical signalling: zusätzliche Bits neben 0 und 1 z.B. J (Start) und K (Ende), z.B. Manchester Extension im Token Ring
- Framelänge: vor Frame wird die Länge angegeben, bei fehlerhafter Übertragung des Längenbytes Verlust der Synchronisation
- Begrenzungszeichen: bei Char-basierter Übertragung (z.B. ASCII)
  - Zeichengruppen werden als Steuerzeichen benutzt
  - Synchronisation -SYN
  - Start of Header - SOH , Start of Text - STX, End of Text - ETX
  - Steuerzeichen werden durch DataLinkEscape (DLE)
  - DLE im Text, wird durch zusätzliches DLE markiert
- Begrenzungsbits: Bitfolge wird als Anfangs- und Endmarkierung benutzt, z.B. 01111110
  - bit stuffing: sicherstellen das Teile der Mitteilung auch Markierung enthalten können
  - wenn in Daten 11111 vorkommt, wird immer 0 angehängt
  - Empfänger entfernt immer 0 nach 11111
- Kombinationen davon werden oft verwendet um Framing-Power zu erhöhen (Delimiter + Frame-Count)

1.9. Flußsteuerung

1.9.1. Zweck

- Bekämpfung von Buffer-Overflow, Congestion
- dient dazu, sicherzustellen, daß der Sender den Empfänger oder ein Netzwerk nicht mit mehr als verarbeitbarem Traffic überlastet
- Verhindert Durchsatz- und Antwortzeitsverschlechterung durch Netzwerk- oder Userüberlast
- Geschwindigkeitsübereinstimmung zwischen User und Netzwerk
- Deadlock-Verhinderung
- Fairness

1.9.2. Sliding Window

- Frameübertragung, Empfang, Dest.-Entity sendet Permit → Sender sendet neues Frame
- Dest. kontrolliert Fluß, Permit meist im ACK
- die Pakete im S. W. werden solange verschickt, bis Windowgröße =1, wartet dann auf Permits (nicht ACKs!, darin enthalten, vergl. TCP)
- Permit-Generierung: immer gleich nach Empfang (ACK == Permit), gut für ACK
- empfangene Frames landen im Buffer, wenn User zu langsam die Frames entnimmt, dann Buffer voll, Empfang geht verloren , kein ACK (+ Permit)
- Problem: ACK abgeschickt, ohne Sicherheit, daß User dies auch aus Buffer entnimmt
- → Lösung: Permit erst wenn Empfang-Buffer neue Frames verkraftet, noch nicht voll ist, gut für Permits
- auch hier unnötig: retransmission in case of timer expiration
- ACK |= Permit: Empfangsquittung ohne Erlaubnis neuer Frames möglich (RNR-Frame (Receiver-Not-Ready)bei HDLC)
- Teilt Sender die „Schließung“ seines Übertragungsfensters mit. RR-Permit: „ReOpen“
- Credits im ACK, Sender weiß dann zu jeder Zeit wieviele Frames noch möglich

1.9.3. Deadlocks und Buffersize-Approach

- in Real: begrenzte Buffer: Deadlock-/Congestion-Vermeidung, max. Durchsatz
- Direct Store and Forward Deadlock: Input- und Output-Buffer ist der selbe
- → wenn Output-Buffer voll, kein Empfang mehr
- wird umgangen, in dem Output-Buffer nur Teil dieses Buffers alloziiert (Rest immer Input) = Channel queue limit
- Indirect Store and Forward: Structured Buffer-Pool, Buffer ist in Bereiche eingeteilt, die exklusiv von Frames mit entsprechenden Hops verwendet werden. Pakete, die i Hops gereist sind, können Buffer auf Stufe H < i (alle kleineren) verwenden. Channel queue limit auch keine Lösung
- Reassembly Deadlocks (ARPANET): nur bei Datagramm-Service. Virtual Circuits durch Reassembling der Pakete beim Empfänger
- Lösung: Buffer-Reservation für alle übertragene Nachrichten (~ Window-Flow-Control mit Reservierungsbestätigungen, die die Permits zum Übertragungsfenster ersetzen)
- Maximierung des Durchsatzes durch Input Buffer Limiting (IBL):
- N ist Anzahl der Buffer
- N_i ist die Anzahl der Buffer verfügbar für Input
- $N_t = N - N_i$ Buffer für Übertragung
- H ist ist durchschnittliche Anzahl der Hops pro Input-Paket
- Nachteil: H mal Übertragungs-Buffer wie Input-Buffer

1.9.4. Windowsgröße-Anpassung bei Congestion

- Pakete und Permits hängen fest (Buffer voll)
- Drosselung des Verkehrs im Netzwerk dazwischen (kleinere Fenster)
- Sender muß Netz-Situation kennen, um zu reagieren
- Queues werden je schneller voll, je mehr Sender, je größer die einzelnen Windows
- Verlust in Zwischensystemen verursacht noch höheren Verlust bei Empfänger
- AKTIV: alle Sender werden von Subsystemen informiert
- Subsysteme erkennen dies über Schwellenwerte, die bei Übertretung Stau signalisieren (z.B. Queue-Belegung)
- Indizierung durch spezielle ACk-Inhalte, Sender entscheidet über weiteres Verhalten (ignore oder react): Windowsgröße multiplikativ oder additiv verändert
- PASSIV: Sender muß Stau erkennen, → längere Response-Delays, Timeout, Rest wie bei Congestion Avoidance bei TCP

1.9.5. End-to-End vs. Hop-by-Hop Flußsteuerung

- EtE: Daumenregel: Windowgröße im Bereich (N, 3N) über einen N Hop-Pfad für kontinuierliche Datenübertragung → Übertragungen über lange Pfade verbrauchen mehr Bandbreite
- HbH: nicht für Datagram-Service geeignet (weil keine Vorhersage der Hops möglich), kürzere Delays → kleinere Fenstergrößen; nicht so schnell wie EtE (auf Congestion-Hop

beschränkt); Backpressure: Sender wird begrenzt, Gleichverteilung der Pakete, wenn Daumenregel befolgt wird

1.9.6. Isarithmetic Flow Control

- jedes Paket braucht eine Erlaubnis für den Eintritt ins Netz, um es nach Verlassen wieder abzugeben (wird neu generiert), für neue Pakete → Eintritts-Token
- Fragen: Wie Gleichverteilung der Permits im Netz? Wie genügend schnell neue belegen? Wie recover zerstörte Permits, wieviele sind insgesamt im Netz?
- Überlast ist möglich
- → wurde niemals implementiert

1.9.7. Steuerung der Rate

- auch Traffic Shaping oder Network Access Control
- **Leaky Bucket**: steuert Anzahl der Daten pro Zeiteinheit
- Vorteil: Open Loop Approach (kein Einfluß durch Round-Trip-Time)
- Angewandt in ATM und XTP

1.9.8. Andere Modelle

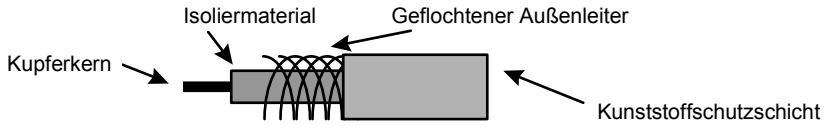
- Paketzerstörung: einfache Zerstörung des Zugriffspakete, Standard in Datagram-Services
- Gibt das Problem an höhere Protokolle weiter
- Routing: Flußbasierte Routing Abschätzung

2. Übertragungsmedien

2.1. Twisted Pair (Verdrilltes Kabelpaar)

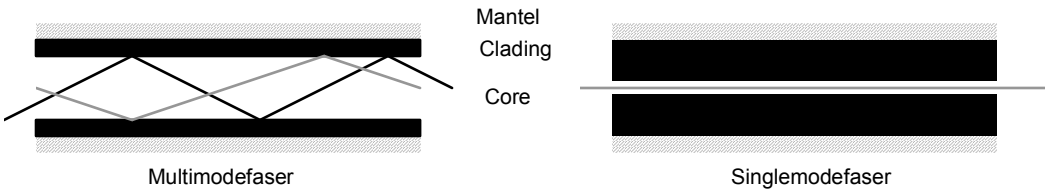
- besteht aus isolierten Kupferdrähten
- sind verdrillt um Abstrahlung / Aufnahmen von elektromagnetische Störungen zu reduzieren
- Kategorie 3: 4 Paar von je zwei isolierten, gewundenen Drähten in Kunststoffhülle
- Kategorie 5: stärke Windungen und bessere Hülle
- UTP - unshielded Twisted Pair: Kat 3 und Kat 5
- STP - shielded TP : einzelne Kabelpaare sind extra isoliert

2.2. Koaxialkabel



- 50 Ohm-Kabel für digitale Übertragung, 75 Ohm vorwiegend für analoge Übertragung
- hohe Bandbreite und ausgezeichnete Rauschbeständigkeit
- Breitband-Koax für TV-Netze

2.3. Fiber optics (Lichtwellenleiter)



- Multimodefaser : viele Lichtstrahlen werden in verschiedenen Winkeln gebrochen
  - multimode stepped Index: Core geht mit einem Schlag in Cladding über, bis 100MHz
  - multimode graded Index: Core geht langsam in Cladding über, bis 1GHz
- Singlemodefaser : nur parallele Lichtstrahlen in der Faser
  - singlemode (stepped) index: Core viel schmaler, bis 100GHz
- Die Umhüllung (Clading) hat einen niedrigeren Brechungsindex als der Glaskern, dadurch wird das Licht in der Faser gehalten.
- Das Licht wird durch LED's oder Halbleiterlaser erzeugt und durch Fotodiode ausgelesen
- Dämpfung des Lichtes ist abhängig von Wellenlänge
- benutzte Wellenlängen : 0.85μ, 1.30μ und 1.55μ Band mit ca. 25.000 - 30.000 GHz Breite
- Dispersion : Ausbreitung der Lichtimpulse in die Länge → Überlappung möglich, Gegenmaßnahme ist Abstände zwischen Impulsen erhöhen oder Solitons verwenden.

2.3.1. Übertragungsfenster

- 0.85µm: rel. hohe Dämpfung (2 dB/Km)
- 1.31µm: mittlere Dämpfung (0.45 dB/Km), geringere Streuung
- 1.55µm: kleinste Dämpfung (0.25 dB/Km), höhere Streuung

2.4. Vergleich Glasfaser und Kupferdraht

Vorteile Glasfaser	Vorteile Kupferdraht
höhere Bandbreiten	leicht zu installieren
niedriger Dämpfung	preiswerte Schnittstellen
Unempfindlich gegen elektromagnetische Störung	
dünner und leichter	
keine Abstrahlung, also sicherer	

2.5. Drahtlose Übertragung

Wellenlänge

Beschreibung : $\lambda f=c$
Einheit : m

- Datenrate eines Frequenzbandes : $\Delta f = \frac{c\Delta\lambda}{\gamma^2}$, Pro Herz können bis zu 40 Bits übertragen werden

2.5.1. Spread Spectrum Technologies

Stellt eine Möglichkeit dar, den Einfluß von frequenzabhängiger Interferenz zu begrenzen. Übertragung über ein weiten Frequenzbereich, um mit den Interferenzen auf sicheren Frequenzen umzugehen

Vorteile: Spread-Spectrum-Signale können über F.-Bänder gelegt werden, wo andere Schmalband Syteme schon arbeiten, Anti-Interferenz Charakter, bessere Sicherheit

Nachteile: Erhöhte Systemkomplexität, großes Frequenzband, um Baseband-Signal zu übertragen

2.5.2. Direct Sequence spread spectrum (DSSS) Systeme

- Jedes Bit (Dauer = T_b)wird multipliziert mit einer Sequenz von Schmal-Pulsen (chips) mit Dauer T_c
- Spreading Faktor N = T_b/T_c. Die Chip-Sequenz ist zufällig.
- Nachteil: größere Bandbreite als konventionelle Kommunikation, aber verschiedene Benutzer können sich dieselbe Bandbreite teilen, indem sie die Spreading-Codes verwenden, die rechtwinklig zu einem anderen sind (CDMA).

2.5.3. Frequency Hopping Spread Spektrum (FHSS) Systeme

- Die Trägerfrequenz von FSK-modulierten digitalen Signalen springt über viele Frequenzen, die von periodischen PN-Codes beschrieben werden. In FHSS/CSMA hat jeder Benutzer ein anderes Spring-Muster. Chip-Dauer ist die Zeit, die bei jeder Frequenz verbracht wird.
- Schnelle FHSS: mehr als ein Sprung / Bit
- Langsame FHSS: weniger als ein Sprung / Bit

2.5.4. Radio

- hohe Reichweite, gute Durchdringungseigenschaften
- Problem : Echoeffekt (multipath fading), Wellen werden reflektiert
- weitere Probleme : Rauschen, Doppler-Effekt (Frequenzänderung), Störung zwischen Stationen
- Dämpfung : $P_r = \frac{P_t\lambda^2}{\left(4\pi d^2\right)}$ mit P_r ... Empfangsstärke, P_t ... Sendestärke (in mW)

2.5.5. Mikrowellen

- über 100 Mhz, Wellen in geraden Linien - leicht zu bündeln
- Probleme : Multipath fading, Absorbierung durch Regen

2.5.6. Lichtwellenübertragung

- Verwendung von Lasern
- Probleme : Regen, Nebel, Ablenkung durch Wärmeströme in der Luft

2.6. Strukturierte Verkabelung

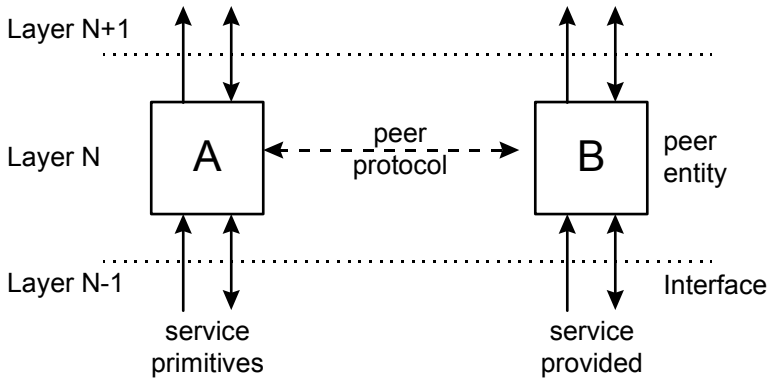
2.6.1. Problem

- in Gebäuden viele verschiedene Arten von Kabel und Steckern für verschiedene Kommunkationssysteme
- Umstellung auf andere Netzwerktechnologien ist aufwendig, teuer (z.B. Bus → Ring)
- gesamte Verkabelung muß getauscht, verändert werden, wenn Veränderung in Struktur

2.6.2. Lösung

- S.V. ersetzt alle durch einziges System
- Integration von digitaler und analogen Signalübertragungen
- billige Installation, Wartung, Austausch
- Strukturierte Verdrahtung, braucht wenig Änderung → Star-Topologie → zentrale Steuerung
- Patchkabel: Physikalische Verbindung, leicht verlegbar, bis 90m + 3m + 6m lang, UTP-5
- EIA/TIA 568:
- cat. 3: <10MHz
- cat. 4: <20Mhz
- cat. 5: <100MHz
- cat. 1 u. 2 nicht offiziell sichtbar
- nur optische Fasern
- definiert RJ45-Stecker
- Horizontale Verkabelung: vom Verkabelungsschrank zur Workstation
- Backbone-Verkabelung: Verkabelungsschrank und Geräteraum (z. B. verschieden Gebäude), besteht aus Backbonekabel, Kreuzverbindungen, Terminatoren, Patchkabel
- Verbindungshardware: Terminiert horizontale Verkabelung (Patchkabel im Schrank, ...)
- paßt BNC, Fibre, RJ-11 und andere Verbinder für Arbeitsplatz-Equipment an
- wichtig: strukturiertes Netzwerk
- Backbone: Verbindung von Routern, Gateways und zu anderen Netzen
- Routers: zwischen Gebäuden oder über Fluren
- Switch/Bridges: Arbeitsgruppen bis 200
- Repeaters/Hubs: Arbeitsgruppen bis 10 (Hub: 16)

3. Logische Architektur von Netzwerken



3.1. Data Units

- Interface Data Unit IDU: Daten die über einen SAP zwischen Schichten ausgetauscht werden, besteht aus ICI und SDU
- Interface Control Information ICI: Steuerdaten
- Service Data Unit SDU: Informationen die übermittelt werden sollen
- Protocol Data Unit N-PDU: SDU wird mit protokollspezifischem Header (Protocol Control Information) ausgestattet und eventuell aufgeteilt (Segmentierung) um an entsprechenden Layer auf anderem Entity übermittelt zu werden

3.2. Layer (Schicht)

- Netzwerkdesign: Schichten, Services, Protokoll
- Schicht bietet Service über SAP an darüberliegende Schicht an
- korrespondiert über Protokoll mit gleicher Schicht der anderen Seite
- n+1 überträgt IDU über SAP zu n
- n entnimmt daraus SDU und ICI
- n erstellt PDU für Schicht n auf der anderen Seite

3.3. Service

- der aktive Teil: Entity (Programm, Hardware, ein paar Codezeilen)
- verbindungsorientiert, verbindungslos (datagram-service)
- betätigt, unbestätigt
- zuverlässig, unzuverlässig
- Primitiven: Request (nach Service), Indication (Ereignis), Response(Reaktion), Confirmation (Antwort auf früheren Request)
- Indication und Confirmation nur bei bestätigenden Services
- jede Primitive besteht aus mindestens drei Operationen: Connect, Data, Disconnect

3.4. Datenübertragung

3.4.1. Verbindungslose Übertragung

- Datagramme werden unabhängig voneinander übertragen
- zwischen Endpunkten existiert keine explizite Verbindung
- nur zwei Service-Primitiven: request und indication
- request: enthält mindestens Ziel- und Quelladresse sowie Nutzdaten

3.4.2. Verbindungsorientierte Übertragung

- besteht aus drei Phasen

- Verbindungsaufbau: Vereinbarung der Parameter der Verbindung, Erstellung eines Kontextes
- Datentransfer:Übertragung der Daten
- Verbindungsabbau: Freigabe der belegten Resourcen

3.5. OSI-Referenzmodell

- Komm-Netzwerke bestehen aus Hard- und Softwarekomponenten
- Netzwerk-Topologie beschreibt deren Beziehungen untereinander
- Logische Architektur beschreibt die Struktur der Komm.-Protokolle und Software-Komponenten

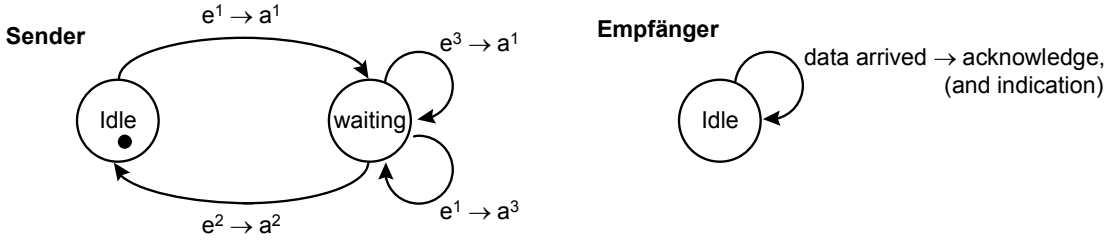
	Layer	**Beschreibung**
7	Application Layer (Verarbeitung)	Anwendungen für den Benutzer
6	Presentation Layer (Darstellung)	Kodierung der Daten in einheitliche Repräsentation
5	Session Layer (Sitzung)	Sitzung zwischen zwei Benutzern managen
4	Transport Layer (Transport)	End-to-End Verbindungen, Abstrahiert von Hardware
3	Network Layer (Vermittlung)	Routing durch Subnetze
2	Data link Layer (Sicherung)	Synchronisation, Fluß- und Fehlerkontrolle
1	Physical (Bitübertragung)	Bit-transparentes Interface zum Übertragungsmedium

4. Protokollspezifikation

4.1. Endliche Automaten

- zur Spezifikation von Protokollen kann man endliche Automaten (finite state machines) benutzen
- definiert durch (S, E, D, S_0) mit
  - S ... Zustände
  - E ... Ereignisse
  - D ... Überführungsfunktion
  - S_0 ... Anfangszustand
- außerdem
  - A ... Aktionen bei Übergang
  - $E^f(s)$ ... mögliche Ereignisse im Zustand s

4.2. Send-and-Wait



s^1	idle	e^1	data to send (request)	a^1	send data, set timer
s^2	waiting	e^2	get acknowledge	a^2	acknow. Transfer, clear timer (conf)
s_0	idle (s^1)	e^3	timer expired	a^3	response: busy (conf)

Übergangsfunktion (für Sender)

Ereignis \ Zustand	S^1 (Idle)	S^2 (Waiting)
e1	a^1, s^2	a^3, s^2
e2	0	a^2, s^1
e3	0	a^1, s^2

4.2.1. Reaktion auf verschiedenen Fehlerarten

Betrachtung aus Sicht des Senders

MsgLose	nach Ablauf des Timers wird erneut gesendet	erkannt
RespLose	Timer läuft ab, Paket wird wiederholt → Duplizierung	nicht erkannt
MsgDup	Paket kommt doppelt beim Empfänger an	nicht erkannt
RespDup	doppeltes ACK wird als Bestätigung für falsches Paket interpretiert	nicht erkannt
Msg/RespReord	können nicht vorkommen, da immer erst gesendet wird, wenn letztes Paket bestätigt ist	—

4.3. Automatic Repeat Request ARQ

- Idee: fehlerhafte Pakete werden erneut gesendet (nach Rückmeldung an Sender)
- Ack's werden benötigt weil
  - kein fehlerfreier, idealer Kanal existiert
  - verschiedene Fehlerarten : Verluste, Defekte, Dublizierungen, falsche Reihenfolge
- möglich nur bei verbindungsorientierter Kommunikation, Ausnahme: ackn. Datagrams
- Empfänger und Sender haben keine vollständigen Informationen über den Zustand der Übermittlung
- Sequenznummern: um Reihenfolge, Wiederholungen und Verluste feststellen zu können
- Timer: vermeiden von Deadlocks

- Sender: Informationen über Zustand des Empfängers durch Sequenznummern und Timer
- Empfänger: durch Sequenznummern Information über Zustand der Übertragung
- Übertragungsstrategien: Übertragung der Pakete
  - einzeln (individuelle)
  - kontinuierlich
- Wiederholungsstrategien
  - nur fehlerhaftes Paket wiederholen
  - fehlerhaftes Paket und alle folgenden wiederholen

4.3.1. Acknowledgments (Ack's)

- positive Ack's: bestätigen von Paketen → ausreichend
- negative Ack's: nur Fehler melden → nicht ausreichend
- Kombination möglich, als Optimierung
- kumulatives Ack's: bestätigen von mehreren Paketen mit einem Ack
- Ack's können extra Pakete sein → Control packets oder zu normalen Pakete hinzugefügt werden → piggybacked ack's

4.3.2. Timer

- für jedes Paket oder ganze Verbindung
- bei Sender und Empfänger
- festlegen des Timers: hop-by-hop oder end-to-end

4.4. Selective Repeat

- Selective Repeat Protokolle wiederholen nur die fehlerhaften Pakete
- WindowSize w: Größe des Sendebuffers
- Problem ist das Verhältnis von WindowSize w und Wertebereich der Sequenznummern R → R >= 2*w
- es kann gesendet werden, wenn weniger als w Pakete unbestätigt sind
- die Pakete werden zyklisch durchnummeriert
- Timeout beim Sender und kein Ack: Pakete wiederholen
- Empfänger schickt Ack mit Sequenznummer des zu bestätigenden Pakets
- Empfänger speichert Pakete wie sie kommen und liefert sie dann in der richtigen Reihenfolge ab
- Selective Repeat ist effizienter als Alternating Bit, da Sender nicht auf Ack warten muß, bevor neues gesendet werden kann
- es wird ein großer Empfangsbuffer benötigt

4.5. Go-back-N

- funktioniert wie Selective Repeat, solange keine Fehler auftreten
- wenn Sender Fehler bemerkt (Timeout), wird alles ab dem fehlerhaften Paket erneut gesendet
- dadurch muß Empfänger keine Pakete speichern, die außerhalb der Reihe ankommen (niedrigerer Speicherbedarf)
- Ack's bestätigen entsprechendes Paket und alle Vorgänger
- Pakete außerhalb der Reihenfolge werden vom Empfänger verworfen
- Effizienz liegt über Alternating Bit und unter Selective Repeat

4.6. Alternating Bit

- funktioniert wie Send-and-Wait, führt aber Sequenznummern ein (0 und 1)

4.6.1. Ablauf

- Sender: sendet Paket mit SeqNr 1, setzt Timer und wartet auf Ack
- Empfänger: prüft bei Empfang SeqNr mit erwarteter SeqNr (hier 1)
  - richtig: Ack mit SeqNr. 1
  - falsch: sendet Ack mit 0 (da Sender sonst immer Paket 1 wiederholen würde)

- Sender: bei Empfang von Ack überprüfen mit erwarteter SeqNo
    - richtig: nächstes Paket senden
    - falsch: Ack ignorieren und auf Timer oder richtiges Ack warten

Anm.: Im Praktikum bekam das Ack nicht die Nummer des empfangenen Pakets sondern die next_expected -SeqNr.

4.6.2. Reaktion auf verschiedenen Fehlerarten

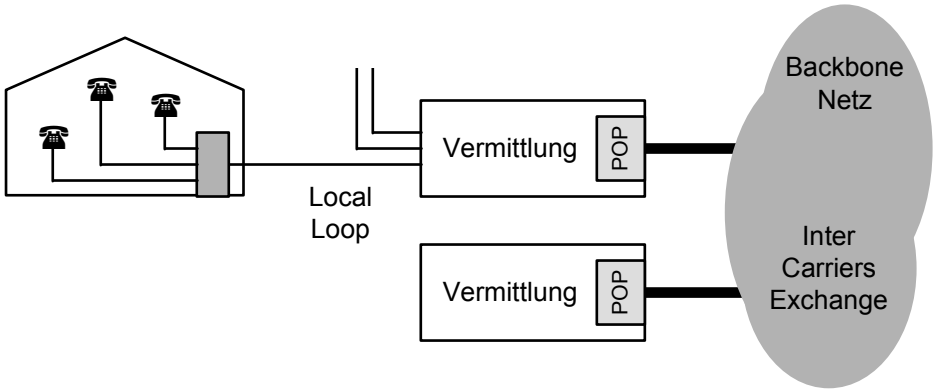
MsgLose	nach Ablauf des Timers wird erneut gesendet	erkannt
RespLose	Timer läuft ab, Paket wird wiederholt Empfänger erkennt Dupliziertes Paket anhand der SeqNr.	erkannt
MsgDup	Paket kommt doppelt beim Empfänger an, wird aber ignoriert	erkannt
RespDup	doppeltes ACK wird aufgrund der falschen SeqNr. Ignoriert	erkannt
Msg/RespReord	können nicht vorkommen, da immer erst gesendet wird, wenn letztes Paket bestätigt ist	—

4.6.3. Paketgröße

- kleine Pakete: wenn Bitfehler auftritt müssen nur wenige Daten wiederholt werden, aber hoher Overhead
- große Pakete: Wahrscheinlichkeit das Bitfehler auftritt wird größer (mehr Bits...), dadurch müssen viele Pakete wiederholt werden, aber wenig Overhead
- Optimum: je nach Bitfehlerrate
- Problem: wenn man die optimale Paketgröße für eine gemessene Bitfehlerrate wählt, gibt es einen Einbruch bei plötzlicher Erhöhung der Fehler

5. Plain old Telephon System P.O.T.S

- größtes Netzwerk der Welt mit gleichen Standards
- 10. März 1876: Vorstellung des ersten Telefons von Alexander Graham Bell und Thomas A. Watson



5.1. Funktionsweise Telefon

- Telefon bezieht Gleichstrom von Vermittlungsstellen
- wenn Hörer abgenommen wird, ist Stromkreis geschlossen
- Klingeln: 20 Hz und 75 Volt - Burst
- Signalisierung: Töne (Frei-, Besetzzeichen) für Endnutzer, Übertragung der gewählten Nummer, Klingeln
- Wählen: Unterbrechen des Stromes (bei abgehobenem Hörer) → Impulswahl oder Übertragung von Tönen → Frequenzwahl

5.2. Signalisierung

- Inchannel / Inband: Signalisierung in gleichen Frequenzband wie Sprache, einfache Technik
- Out-of-band: anderer Kanal für Signalisierung, aber die gleichen Geräte, während gesamter Verbindung können Kontrollinformationen ausgetauscht werden
- Common channel: gemeinsamer Kanal für Signalisierung von mehreren Sprachkanälen, reduziert Call Setup Time

5.2.1. Inband Interoffice Signalisierung

- für analoge Übertragungen / Telefonie
- Wechselstrom
- Einzelfrequenz-Ton wird zur Signalisierung benutzt, Adressinformationen (Telefonnummern) werden durch Kombinationen von 2 Tönen dargestellt mit Rate 10 Kombinationen pro Sekunde
- Frequenzen: 700, 900, 1100, 1300, 1500, 1700 Hz
  → multiple frequency key pulsing

5.2.2. American Digital Hierarchy

- jeder Sprach-Kanal ist mit 8000 Samples pro Sekunde á 8 Bit digitalisiert = 64.000 Bit/s
- 24 Kanäle + 1 Framing-Bit = 8000 * (24*8 +1) = 1.544 MBit/s → D3 Frame
- T1/D4 Superframe besteht aus 12 Frames (á 193 Bit)
- Framebits werden zur Synchronisation benutzt
- jedes 8. Bit eines Kanals im 6. und 12. Frame wird für Signalisierung benutzt → robbed Bit
- Problem: bei Sprache kein hörbarer Verlust, bei Daten nicht tragbar !
- Lösung für Datenübertragung: Jedes 8. Bit wird nicht benutzt → nur 56.000 Bit/s in Amerika
- Extended Superframe Format: 24 Kanäle,
- Framingbits werden für Framing (6 Bit), Error Checking (6 Bit) und Wartung (12 Bit) benutzt

5.2.3. Europäischer Standard

- 32 Kanäle á 64 KBit/s
- 2 ohne Nutzdaten (No. 0 und 16) für Framing und Signalisierung
-

5.2.4. Synchronisation

- Problem: Abtastung muß bei Sender und Empfänger übereinstimmen, sonst kommt es zu Verschiebungen z.B. wenn zu spät abgetastet wird, kann schon wieder die abfallende Flanke erreicht sein und statt 1 eine 0 gelesen werden
- Elastic Buffer: die gelesenen Daten werden in Puffer gelesen, anhand von Kontrollbits wird die Synchronisierung hergestellt und dann, mit richtigem Takt, wird der Puffer ausgelesen

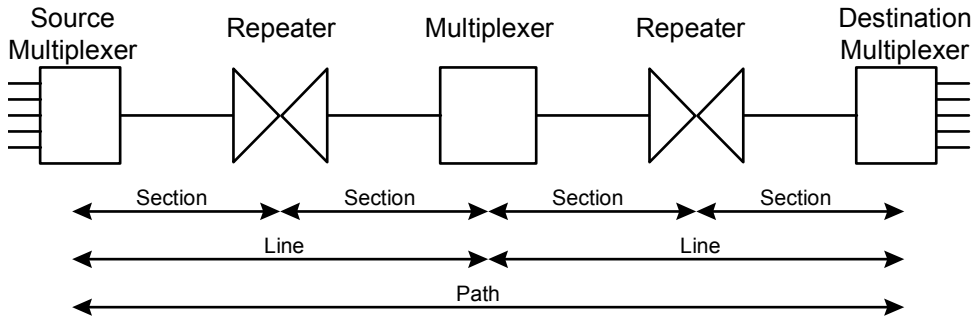
5.3. Sonet / SDH

- Standard: Sonet → Synchronous Optical Network
- CCITT-Empfehlungsreihe: SDH → Synchronous Digital Hierarchy
- TDM-System für Glasfaser-Verbindungen

5.3.1. Ziele

- Zusammenarbeit verschiedener Netzbetreiber ermöglichen (Zeitgabe, Wellenlänge, Rahmenstruktur)
- Gewährleistung des Multiplexings von unterschiedlichen digitalen Kanälen, hohe Bitrate erreichen
- bessere Operation, Administration und Maintenance

5.3.2. Aufbau



- Path Verbindung zwischen Quelle und Ziel
- Line Strecke zwischen zwei Multiplexern
- Section Verbindung zwischen zwei beliebigen Geräten (point-to-point)

- Teilschichten: Photon, Section, Line, Path (nicht im Skript)
- Photo physikalische Eigenschaften des Lichtes und der Fasern
- Section erzeugen und verarbeiten eines Rahmens (Point-to-Point)
- Line Multiplexing
- Path End-to-End Verbindung

5.3.3. Frame Struktur

- synchron: auch wenn keine Nutzdaten vorliegen wird gesendet
- Sonetrahmen. alle 125 μs eine 810 Byte Tabelle (90 Spalten, 9 Zeilen, je 1 Byte)
- 8000 Hz → 9 * 90 * 8 Bit * 8000 s^{-1} = 51.84 MBit/s (brutto)
- Spalte 1 bis 3: Management Informationen (Zeile 1-3: Section-Info, Zeile 4-9: Line-Info)
- restliche 87 Spalten: 50.112 MBit/s (netto)
- Datenblock kann an beliebiger Stelle beginnen, Zeiger wird mit Line-Overhead übermittelt
- Daten können auch über mehrere Sonet-Frames gehen

5.3.4. Multiplexing

- beliebige Eingabeströme werden zu STS-1-Kanälen zusammengefaßt (gegebenenfalls auffüllen)
- elektrisch: STS-12 ← 4 x STS-3 ← 3 x STS-1 (bis STS-48 definiert)
- optischer Träger: OC-*n*, wenn nicht gemultiplext OC-*n*c
- STS-3 paßt zu ATM-Übertragungsrate von 155 MBit/s

5.4. Modem

5.4.1. Aufbau und Funktionsweise

- Übertragung von Daten über die Sprachkanäle des Telefonsystems
- Daten werden digital vom Computer an Modem übermittelt
- Modem moduliert digitale Daten in analoge Signale und übermittelt diese über die normale Telefonleitung
- Modem an der Gegenstelle demoduliert analoge Daten wieder zu digitalen Daten und übermittelt diese an den angeschlossenen Computer

5.4.2. Modulationssysteme

- Gleichstrom nur für niedrige Geschwindigkeiten und kurze Entfernungen praktikabel → Dämpfung und Laufzeitverzehrung von digitalen Signalen sehr groß
- Wechselstrom → Sinuswellenträger (Dauerton zw. 1000 und 2000 Hz) wird moduliert
- Amplitudenmodulation: zwei verschiedene Spannungspegel stellen 0 bzw. 1 dar
- Frequenzmodulation (auch Frequency Shift Keying): Frequenz wird verändert
- Phasenmodulation (Phase Shift Keying): versetzen der Trägerwelle in gleichmäßigen Intervallen
- höhere Übertragungsgeschwindigkeit nicht durch höhere Abtastrate möglich (Nyquist Theorem) → also mehr Bits pro Sample übertragen (Bit/s ist größer als Baud)

5.4.3. Modemstandards

- V-Serie der CCITT → Standards für Datenübertragung

V.17 G3-Fax (14400 Bit/s)
V.23 1200 Bit/s über gute Leitungen sonst 600 Bit/s, FSK, langsamer Rückkanal
V.24 Definition für Schnittstellenleitungen zwischen DTE und DCE
V.25 Protokolle mit Hayes-Kommandos für Durchführung des Wählvorgangs
V.26 2400 (1200) Bit/s, DPSK
V.27 4800 (2400) Bit/s, DPSK, synchron, vollduplex (4 Drähte), halbduplex (2 Drähte)
V.29 9600 (7200,4800) Bit/s, Kombination Amplituden- und Phasenmodulation
V.32 9600 (4800) Bit/s, QAM, asynchron, synchron, vollduplex
V.33 14.400 Bit/s, TCM bei 2400 Baud, voll- u. halbduplex, 4 oder 2 Leitungen
V.42 Fehlerkorrektur für Modems
V.42bis Datenkompression mit BTLZ-Algorithmus für Modems
V.110 Unterstützung für Geräte mit V-Schnittstelle im ISDN

5.4.4. V.34 Standard

- enthält Eigenschaften von V.17, V.29, V.32, V.32bis, V.33
- 2 Drähte, Point-to-Point Telefonverbindungen
- funktioniert mit Dial-up- oder Standleitungen
- voll- und halbduplex, synchron, 2ter Kanal mit 200 Bit/s möglich
- Geschwindigkeiten bis 28.800 Bit/s
- benutzt V.24 Schnittstelle
- QAM
- verschiedene Symboleraten und Trägerfrequenzen möglich → wird bei Verbindungsaufbau ausgehandelt

5.4.5. Echosperren

- erreicht ein Signal über eine lange Leitung das Ziel, kann es am Ende reflektiert werden → Störung
- um bei Gesprächen diesen Effekt zu unterdrücken → Echosperren: Leitung ist nur in die Richtung freigeschaltet in die gesprochen wird, dadurch ist kein Volldublexbetrieb möglich
- um das zu umgehen wird ein 2.100 Hz-Ton gesendet, der Echosperren ausschaltet (In-Band Signalisierung)
- neue Technik → Echo Cancelers

5.4.6. Fehlerkorrektion und Datenkompression

- MNP (Microcom Networking Protocol): fehlerfreier Austausch von Daten in störanfälligen Netzen
- MNP 5: zusätzlich Kompression → identische Bytefolgen werden zusammengefaßt (bis 2:1)
- CCITT V.42: Fehlerkorrekturtechniken unabhängig von Übertragungsrate und Modulationsverfahren, LAP-M und MNP 2 bis 4
- CCITT V.42bis: zusätzlich Datenkompression bis 4:1

5.5. RS232 Interface

- RS-232-C dritte Überarbeitung des EIA-Standards, entspricht (fast) V.24 von CCITT
- DTE (Data Terminal Equipment): Terminal oder Computer
- DCE (Data Circuit-Terminating Equipment): Modem bzw. Gerät zur Datenübertragung
- mechanische Spezifikation: 25polige Stecker (DTE) bzw. 25polige Buchse (DCE)
- elektrische Spezifikation: kleiner 3 Volt → binär 0, größer +4 Volt → binär 1, Datenraten bis 20 Kbps, max. 15 m Kabellänge

5.5.1. funktionelle Spezifikation

die 9 wichtigsten Signale (Pinbelegung):

DTR (20)	Data Terminal Ready	Computer ist bereit
DSR (6)	Data Set Ready	Modem ist bereit
CD (oder RLSD) (8)	Carrier Detect	Modem hat einen Träger erkannt
RTS (4)	Request to Send	Computer möchte Daten senden
CTS (5)	Clear to Send	Modem ist bereit Daten zu empfangen
TD, TX (2)	Transmitted Data	Senden von Daten
RD, RX (3)	Received Data	Empfangen von Daten
GND (1)	Signal Ground	alle anderen Signallevel werden im Verhältnis dazu gemessen
RI	Ring Indicator	Modem informiert Computer über ankommenden Ruf

5.5.2. Verbindungen

- 2 Computer direkt → Nullmodemkabel, dabei sind jeweils TX/RX, RTS/CTS und DTR/DSR gekreuzt
- DTE zu DCE Timing: das Timing kann aus den ankommenden Daten gewonnen werden, intern vom Modem erzeugt werden oder durch eine externe Uhr vom Computer gesteuert werden
- local Loopback: Überprüfung der Verbindung von Computer zu Modem
- remote Loopback: Überprüfung der Verbindung der Modems

5.5.3. RS-449

- Weiterentwicklung von RS-232
- besteht aus drei Normen
- mechanisch, funktionell, verfahrenstechnisch → RS-449
- elektrisch, asymmetrisch → RS-423-A : gleiche Masseleitung für alle Schaltung (wie RS-232)
- elekrisch, symmetrisch → RS-422-A : jede Hauptschaltung hat eigene Masseleitung, dadurch bis zu 60 m Kabellänge und bis 2 Mbps

5.5.4. X.21 Interface und V.35

- Protokolel der Bitübertragungsschicht
- X.21 wird für die Verbindung von X.25-Netzen mit dem Kunden benutzt
- X.21 entspricht weitgehend RS-232-Interface
- V.35 Funktionen ähnlich RS-232, 34 Pins

5.5.5. USART

- Universal S??? Asynchronous Receiver /Transmitter
- Baustein im Modem zum Senden und Empfangen der Daten
- besteht aus:
    - Command Register
    - Mode Register
    - Status Register
    - Transmit buffer
    - Receive buffer
    - verbunden durch internen Datenbus
    - Taktgeber (Uhr)

6. Local Area Networks

6.1. LAN Topologien

6.1.1. Bidirektionaler Bus

- Bussystem auf dem in beide Richtungen gesendet werden kann
- ermöglicht Multipoint-Übertragungen
- Kollisionen sind möglich
- Kabellänge ist begrenzt; Segmentgröße, Anzahl der Stationen und minimaler Abstand werden fest definiert
- Propagation delay ist von der Länge des Segments nicht der Anzahl der Stationen abhängig
- am Ende des Busses verfällt Signal
- Signal breitet sich passiv aus, wird nicht von Stationen bearbeitet, nur mechanische Unterbrechung kann Ausbreitung des Signals verhindern
- Versagen einer einzelnen Station sollte Netz nicht beeinflussen
- schwierige Fehlersuche

6.1.2. Aktiver Stern

- nur Point-to-Point Verbindungen möglich
- verschiedene Medien bei Verbindung der einzelnen Stationen mit der Hub möglich
- für die Vernetzung von N Stationen werden 2*N Empfänger-Sender Paare benötigt
- Kabellänge kann groß sein, ist aber begrenzt
- Anzahl der verbundenen Stationen mit Hub sind begrenzt, aber hierarchische Konfigurationen möglich
- Propagation delay ist unabhängig von Anzahl der Stationen, aber abhängig von Anzahl der Hierarchien
- Hub ist zentraler Punkt, bei Ausfall versagt das gesamte Netz, Unterbrechung eines Links ist unkritisch
- leichte Fehlersuche und Konfiguration

6.1.3. Unidirektionaler Ring

- nur Point-to-Point Verbindungen möglich
- verschiedene Medien möglich
- Länge der einzelnen Point-to-Point Verbindungen ist begrenzt, Gesamtlänge kann groß sein
- jede Station verursacht Verzögerung → propagation delay wächst mit Anzahl der Stationen
- Signal wird von jeder Station bearbeitet, bei Ausfall muß Bypass möglich sein
- Daten müssen explizit von Ring entfernt werden, zum Beispiel vom Sender
- parallele Übertragung von verschiedenen Nachrichten kann möglich sein
- einfache Fehlersuche

6.2. Zugriffsmechanismen

6.2.1. Time Division Multiple Access TDMA

- Zuteilung von Zeitslots für jede Station
- wenn Station senden will, wartet sie auf den nächsten zugewiesenen Slot
- Verzögerung bis zum Senden wächst mit Anzahl der Stationen und steigendem Durchsatz

6.2.2. Polling

- zentrale Station erteilt Sendegenehmigungen

6.2.3. ALOHA

- jede Station sendet, wenn sie etwas zu übertragen hat
- wenn mehrere Stationen gleichzeitig senden, treten Kollisionen auf

- Station wartet positives Ack ab
- wird keine Bestätigung empfangen (timeout) wird nach einer zufällig gewählten Zeit erneut gesendet
- Durchsatz = $S = G \cdot e^{-2G}$ mit G ... Anzahl der zu übertragenden Rahmen (Poisson-verteilt)
- Slotted Aloha: Stationen sind getaktet, so daß immer nur zu bestimmten Zeitpunkten mit dem Senden bekonnen wird
- der Vorteil von Slotted Aloha ist, daß Rahmen entweder sofort kollidiert oder sonst ganz übertragen wird
- Durchsatz = $S = G \cdot e^{-G}$
- Anzahl der Kollisionen erhöht sich mit steigender Anzahl von zu sendenden Rahmen exponentiell

6.2.4. CSMA

- Carrier Sense Multiple Access (CSMA): Weiterentwicklung von ALOHA
- Station hört das Medium ab und beginnt erst zu senden wenn keine andere Station mehr sendet.
- tritt eine Kollision auf, weil mehrere Stationen gleichzeitig mit Senden begonnen haben, warten sie eine zufällige Zeitspanne und beginnen von vorn (abhören bis Kanal frei ist)
- mit steigender Ausbreitungsverzögerung wächst die Wahrscheinlichkeit, daß Kollisionen auftreten: eine Station beginnt zu senden, andere Station bemerkt dies aber nicht, da Signal noch nicht angekommen ist und beginnt auch zu senden → Kollision
- 1-persistent (ständiges) CSMA: sobald Kanal frei ist wird gesendet (Wahrscheinlichkeit = 1)
- nonpersistent (unterbrochenes) CSMA: Belegung des Kanals wird nicht fortlaufend überprüft, sondern wenn festgestellt wird, daß Kanal belegt ist, wird eine zufällige Zeitspanne bis zur nächsten Überprüfung gewartet
- p-persistent CSMA: es wird mit einer Wahrscheinlichkeit von p gesendet, wenn freier Kanal vorliegt

6.2.5. CSMA CD

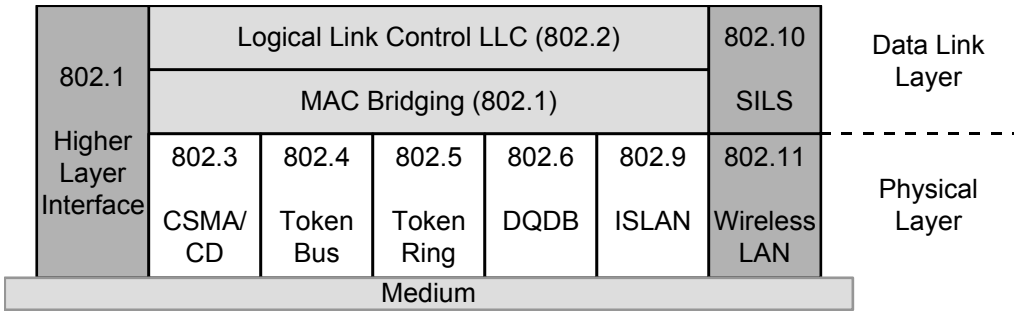
- Carrier Detect (CD): wenn eine Kollision auftritt, unterbrechen beide Stationen das Senden
- damit steht der Kanal schneller wieder zur Verfügung
- Erkennung einer Kollision durch Vergleich von übertragenem und empfangenen Signal (z.B. Leistung, Impulsbreite)

6.2.6. Weitere Zugriffsmechanismen

- CSMA/CR: es wird eine Reservierung vorgenommen, und nur in zugewiesenen Slots gesendet
- CSMA/CA: Collision Avoidance → sicherstellen das keine Kollisionen auftreten
- COMB: jeder Sender bekommt eine Schlüssel zugewiesen, der bestimmt wann gesendet werden darf
- EY-NPMA (Elemination Yield - Non-preemptiv Multiple Access)
- RTS/CTS mit ALOHA-Zugriff (Ready to Send, Clear to Send)
    - Sender schickt RTS mit Angabe von benötigter Zeit, Sender und Empfänger
    - alle Stationen warten solange, wie für CTS benötigt wird
    - Empfänger antwortet mit CTS
    - alle Stationen warten solange, wie für die Daten benötigt wird
    - Mode B: alle Stationen warten bei RTS und CTS jeweils die Zeit die für die Daten benötigt wird

6.3. IEEE 802.x

6.3.1. 802.x Architektur



Physische Schnittstellen:

- CSMA/CD: 10 Mbps Basisband-Koax, 10/100 Mbps UTP/STP, 100 Mbps STP, 10 Mbps Breitband-Koax, 10 Mbps Glasfaser
- Token Bus: 1, 5, 10 Mbps Breitband-Koax, 1, 5, 10 Mbps Carrierband Koax, 5, 10, 20 Mbps Glasfaser
- Token Ring: 4, 16 Mbps STP, 4 Mbps UTP
- Round Robin priority (802.12): 100 Mbps UTP
- FDDI: 100 Mbps Glasfaser, UTP
- DQDB: 100 Mbps Glasfaser
- CSMA polling (wireless): 1, 2 Mbps Infrarot, Spread Spectrum

6.3.2. Logical Link Control Sublayer

- des Data Link Layers ist aufgeteilt in LLC und MAC
- Kommunikation zwischen LLC-Layer der Endsysteme
- bietet verbindungslose und verbindungsorientierte Dienste
- mehr Info's im Kapitel über HDLC
- Frame besteht aus: Ziel und Quellen-Adresse des SAP's, CONT und Daten

6.3.3. MAC Sublayer

- MAC-Layer bietet verbindungslose Datenübertragung
- MA_DATA.request, .confirm, .indication, .status (nur DQDB)

6.4. Ethernet

- Norm: ANSI / IEEE 802.3 CSMA/CD
- z.B. 10Base5, 10Base2, 10BaseT, 100BaseXT, 100BaseVG, 100BaseXL
- erste Nummer gibt Übertragungsrate in KHz/s an
- **Base:** Baseband- und **broad**: Broadband-Übertragung
- letzte Ziffer oder Zeichen beschreibt Medium, Länge eines Segmentes

OSI/ISO Referenzmodell	IEEE 802 LAN	
Data Link Layer	LLC Logical Link Control MAC Media Access Control	
Physical Layer	PLS Physical Signaling AUI Attachment Unit Interface	
	MAU Medium Attachment Unit	Physical Medium Attachment MDI Medium Dependent Interface
Medium		

6.4.1. Physical Layer

- Senden und empfangen von Bitströmen
- Kollisionserkennnung
- Codierung / Decodierung

- Präamble-Generierung
- Uhr-Synchonisation
- Testen der Übertragung von Station hoch zu Medium-Access-Unit

6.4.2. MAC Frame Format

- Präambel: Synchronisation
- SFD: Start frame delimiter (Begrenzung)
- Destination und Source - Adresse: 48 oder 16 Bit
- Länge: Anzahl der Daten-Oktets
- LLC Data: Payload
- PAD (padding): auffüllen wenn zuwenig Daten
- FCS Frame Check Sequence: CRC32 über alle Felder zwischen SFD und FCS

6.4.3. MAC Funktionen

- verpacken bzw. entpacken der Daten in Frames
- Adressierung
- Fehlererkennung
- Kollisionserkennung und -behandlung

Transmission without contention
- überprüfen des CarrierSense-Signals
- wenn das Medium frei ist, wird die Übertragung initialisiert (nach einem Interframe Gap)

Reception without contention
- wenn CarrierSense-Signal anliegt, werden die Bits solange empfangen bis Signal aussetzt
- danach werden die Daten an den Decapsulation-Prozeß zur Fehlerkontrolle und entpacken gegeben

Kollisionserkennung
- Medium wird permanent abgehört
- wenn Kollision auftritt (also mehrere Stationen senden) ist die elektrische Spannung auf dem Medium höher

Kollisionsvermeidung
- durch aufschieben des Sendens, wenn CarrierSense-Signal festgestellt wurde
- wenn Station eine Kollision feststellt → JAM-Signal
- erneute Übertragung wird erst nach bestimmter Zeitspanne versucht
- Zeitspanne wird größer bei jeder Kollision und kleiner bei jeder erfolgreichen Übertragung

Collision handling
- nach Kollision wird solange erneute Übertragen bis Erfolg eintritt oder die maximal Anzahl von Versuchen erreicht wurde
- Scheduling of Retransmission: trucated binary exponential backoff
    - nach Kollision wird Zeit in Zeitschlitze unterteilt
    - Stationen bestimmt Zufallszahl, beim ersten Versuch 0 oder 1, und sendete im 0ten oder 1ten Zeitschlitz
    - tritt erneut ein Kollision auf (zum n-ten Mal) wird Zufallszahl zwischen 0 und 2^n-1 gewählt, bis zur erfolgreichen Übertragung
    - nach 10 Kollisionen max. Schlitzzahl wird auf 1023 begrenzt (und nicht z.B. 2^{12} = 4096)
    - nach 16 Kollisionen gibt Sender auf → Mitteilung an höhere Schichten
    - Vorteil: wenn wenige Stationen kollidieren → kurze Wartezeiten,
      wenn viele Stationen kollidieren → rasche Auflösung

6.4.4. Parameter für 10 MBit/s Netz

- interframe Gap: : Abstand zwischen Frames

- slottime und min Framesize: bei einer Netz mit maximaler Größe (2500 m Kabellänge) beträgt die maximale RTT 51,2 ms (also 512 Bits). Damit allle Stationen eine Kollision bemerken, muß ein Frame mdst. 64 Byte lang sein
- jam Size: Anzahl der Bits die Übertragen werden müssen, damit alle Stationen eine Kollision bemerken (32 Bit) (Größe des JAM-Signals)

6.4.5. PLS, AUI und MAU

Physical Signaling (PLS)
- logische und funktionale Kopplung von DTE und MAU
- ermöglicht Datenübertragung zwischen PLS-Entitäten
- definiert die Interaktion zwischen PLS und MAC sowie PLS und MAU

Attachment Unit Interface (AUI)
- spezifiziert das Interface zwischen PLC und MAU
- Festlegung von mechanischem Aufbau und elektrischer Belegung

Medium Attachment Unit (MAU)
- Empfangs- und Sendeeinrichtung zwischen AUI und Medium (Kabel)
- Medium Dependend Interface (MDI) : mechanische Komponenten (z.B Stecker)
- Physical Medium Attachment (PMA) : elektrische Funkionen

6.4.6. Hardwarekomponenten

- Koax-Kabel (yellow Cable, Cheapernet): max. 500m, 50Ohm Abschlußwiderstand, max. 100 Stationen
- Twisted Pair, Glasfaser
- Link cable segments: verbinden von Segmenten über 100m bis 1000m mit Glasfaser, je eine Leitung für Senden und Empfangen
- Tranceiver (AUI) Kabel: max. 50m, verbindet Sender mit Kontroller
- Tranceiver: Verbindung zum Kabel, Senden und Empfangen der Bits, Feststellen von Kollisionen, Jabber controll, max. 500m
- Repeater: verbinden von Kabelsegmenten, max. 4 erlaubt, regeneriert die Signale, bei Kollisionen erzeugen des Jam-Signals
- Controller: umfaßt LCC, MAC und PLS

6.4.7. Twisted Pair Ethernet 10Base-T

- Ethernet mit Twisted-Pair-Verkabelung
- jede Station ist direkt mit Hub verbunden
- Hub: empfängt Signale und sendete diese wieder auf jeden Anschluß
- Kollisionserkennung und -handling entspricht 10Base2/5 (jede Station)
- alle Stationen an einer Hub gehören zu einer Collision Domain
- max. Kabellänge Hub-Station bis 150m
- Stecker: RJ45 8polig

Vorteile
- preiswerter als 10Base5
- größere Reichweite als 10Base2
- Kabelstörungen, defekte Stecker u.ä. beeinflussen nicht das gesamte Segment
- leicht zu warten (z.B. Kabelbrüche finden)
- → Strukturierte Verkabelung
- einfache Installation von Station

6.4.8. Switched 802.3

- Problem bei großer Anzahl von Stationen: hohe Wahrscheinlichkeit für Kollisionen
- Lösung: Aufteilung des Netzes in Segmente → Collision Domains und Verbindung durch Switches

Aufbau 802.3-Switch

- mehrere Ports (in/out) verbunden durch Highspeed-Backbone
- anhand der Zieladresse wird festgestellt, an welchen Port ein Frame weitergeleitet werden muß
- an jedem Port kann wiederum eine Hub angeschlossen sein
- Switch muß wissen bzw. lernen, welche Stationen an welchem Port angeschlossen sind
- Kollisionen können nur noch an einem Port entstehen (innerhalb einer Collision Domain)

Switching	Shared Medium
• Puffer werden benötigt, Gefahr von Überlauf ohne das der Sender dies bemerken kann	• Sender bemerkt Verluste durch das Auftreten von Kollision und kann Retransmit starten
• dies muß durch die höheren Protokollschichten bearbeitet werden	• also keine Verluste, nur unterschiedliche Verzögerungen

6.5. Ring Topologien

6.5.1. Zugriffsarten

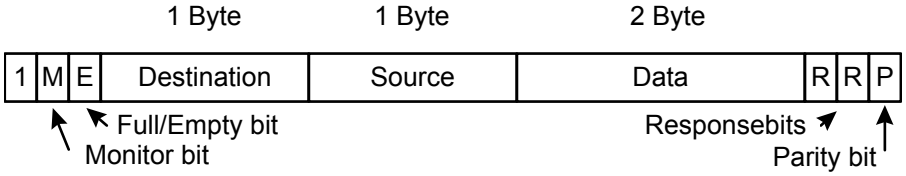
- Listen Mode: Station liest Daten vom Ring und schreibt sie wieder auf den Ring
- Transmit Mode: Station liest vom Ring und schreibt die eigenen Daten auf den Ring
- in Station findet Verzögerung statt

6.5.2. Slotted Rings

- Ring ist in feste Anzahl von Bits eingeteilt, die rotieren
- jede Station liest jedes Bit und gibt es dann zur nächsten Station weiter
- fester Anzahl von Bits wird zu Slot zusammengefaßt
- jedes Frame kann als leer oder belegt gekennzeichnet werden
- wenn Station senden will, wartet sie auf freies Frame, schreibt die Daten
- der Empfänger markiert die Frames als gelesen (response bits)
- der Sender markiert die Frames wieder als leer (automatische Ack)

6.5.3. Cambridge Ring

- 10 MBit/s, maximal 100 Stationen
- Frame → Minipackets: Länge 38 Bits, inklusive 2 Byte Daten, feste Anzahl von Minipackets (4 Stück)
- Sender kann feststellen, welche seine Pakete sind, durch Zählen der vorbeikommenden Pakete



- durch Auswertung der Responsebits kann Sender feststellen, ob Übertragung erfolgreich war
- Initialisierung und Wartung wird durch Monitorstation erledigt
- Monitorbit wird von Monitorstation in jedes benutzte Frame gesetzt und vom Sender entfernt. Dadurch kann festgestellt werden, ob Sender „vergessen“ hat ein Minipacket wieder freizugeben, dann übernimmt dies die Monitorstation
- jede Station darf nur ein Frame belegen und muß es außerdem danach freigeben → fair
- Probleme: ein zentraler Monitor (fehleranfällig), nur ein Slot (zu wenig)

6.5.4. Register insertion rings

- gleiche Station wie slotted Ring, nur anderes Station-Interface
- listen mode: Receive und Transmit Switch schalten auf Durchlaß
- senden: Daten werden in Transmitbuffer geladen, Ring wird unterbrochen und die Bits in den Insertionbuffer geladen, Daten werden aus Transmitbuffer in Ring „injiziert“, allerdings nur so viele, wie frei Plätze im Insertionbuffer festgestellt werden

- Empfänger lädt Daten in Receivebuffer und entfernt sie dann vom Ring
- Pakete können variable Länge haben, die maximale Größe ist durch die Größe der Puffer begrenzt
- gleichzeitige Übertragungen möglich, da Empfänger Daten entfernt
- größere Verzögerung in jeder Station, da Zieladresse analysiert werden muß

6.5.5. Token Ring Access

- single token operation: wenn Sender sein Busy-Token wieder empfängt und keine Daten mehr sendet, wird dies durch Free-Token ersetzt
- multiple token operation: Free-Token wird sofort nach letztem Bit der Daten auf den Ring geschrieben
- single packet operation: Free-Token wird sofort nach dem Empfang des letzten Bits der eigenen Daten geschrieben

1. Station wartet auf ein Free-Token
2. bei Sendewunsch nimmt Station Token vom Ring und beginnt die Übertragung
3. je nach Modus (siehe oben) wird gewartet und dann das Free-Token wieder auf den Ring geschrieben

6.5.6. IEEE 802.5 Token Ring

- spezifiziert 1985
- Medium: TP, 4 MBit/s und 16MBit/s, physikalischer Ring (wenn Ring unterbrochen wird → Totalausfall)
- Verbesserung der Verkabelung: wire center → alle Stationen sind mit Zentrum verbunden, dort wird erst die Verschaltung für einen Ring vorgenommen (wenn eine Leitung unterbrochen ist, wird im Zentrum Bypass geschaltet und Betrieb geht weiter)
- Differential Manchester Kodierung mit +/- 3,0 bis 4,5 Volt für high/low - Level
- jede Station hat eine 1 Bit Puffer (also 1 Bit Verzögerung pro Station)
- jede Station regeneriert das Signal, Aktiver Monitor korrigiert die Referenzphase
- wenn Jitter (unterschiedlich große Delays) zu groß, kann Station das Timing-Signal nicht wiederherstellen → Fehler

Token

- 3 Byte lang, max. einmal auf dem Ring vorhanden
- Token wird durch invertieren eines Bits von Station übernommen, es entsteht ein normaler Paketheader
- Token besteht aus Startdelimiter SD, Accesscontroll AC und Enddelimiter ED

Datenpaket

- besteht aus SD, AC, Framecontrol FC, Ziel und Quelladresse, Daten, Frame Check Sequenz FCS, ED und Framestatus FS
- Empfänger kopiert Daten und schreibt sie wieder auf den Ring
- bei erfolgreichem Empfang werden die FC-Bits gesetzt
- Sender entfernt Daten vom Ring und erzeugt neues Token
- Sender kann Token nur für bestimmte Zeit halten → token-hold-time
- Prioritäten können über AC-Feld geregelt werden : Frames mit höherer Priorität als aktuelle haben Vorrang

Ringwartung

- eine Station ist aktiver Monitor
- wenn Station ausfällt wird neuer Monitor bestimmt: wenn Station Ausfall bemerkt, sendet sie ein „Claim-Token“-Frame und wird Monitor wenn sie das Frame wiederempfängt
- Aufgaben der Monitorstation
    - Verluste feststellen (timer)
    - Paketfragmente nach Fehler entfernen
    - 1 Bit Verzögerungen einfügen um Token auf dem Ring zu halten (wenn nicht genügend Stationen eingeschaltet sind (je 1 Bit-Verzögerung), ist Ring nicht „lang“ genug um 24 Bit für Token zu enthalten)

7. HDLC

7.1. Warum HDLC

- High Level Data Link Control, Layer 2
- Single Hop
- Übertragungsmedien bieten unzureichenden Schutz votr Übertragungsfehler
- Multiplexing meist nicht möglich
- Flußsteuerung kaum vorhanden
- V.24: Restfehlerwahrscheinlichkeit zu hoch, Flußkontrolle: bei jedem Zeichen durch RTS/CTS oder XON/XOFF

7.2. HDLC bietet

- gesichert Datenübertragung (Aufbau, Überwachung, Abbau der Verbindung, Kontrolle von Fehlern und Fluß)
- auch ungesicherte Datenübertragung: Datagram-Service
- Multiplexing – Demultiplexing
- Splitten – Rekombinieren (SDU im mehrere PDUs)
- verschiedene Modi für verschiedene Geräteklassen

7.3. HDLC – Modi

- unbalanced Point-to-Point
- unbalanced Point-to-MultiPoint
- Async-balanced: exklusive Kanäle für jede Richtung

7.4. HDLC – Frames

- enthalten: Pole/Final-Bit
- Inhalt (Was?), Sequenznummern
- Rahmentyp
- Funktionen des Pakets
- Organisatorische Dinge

- Flag (Rahmenbegrenzer) „01111110“
- jeder Rahmen beginnt und endet mit Flag
- Bitfolge nicht im Address-, Control, oder Info- Feld sein → Bit-Stuffing
- Fehler wenn Empfänger in diesen Feldern 6 aufeinanderfolgende Einzen zählt, Rest siehe (Flußsteuerung)
- FCS (Frame Check Sequence): 16 Bit CRC über diese Felder
- immer noch Restfehler möglich

- Informationsfeld: Länge variiert, nicht fest (Daten)
- Adressfeld: Identifizierung der Station, Reservierung möglich (für Signalisierung, Broadcast, P-to-P - logische Verbindungen); +NRM , ARM: Secondary Station; +ABM(s. u.): SAP in anderer Station

7.5. HDLC – ABM

- Asynchronous Balanced Mode
- Balanced Mode P-to-P
- jede Station ist gleichzeitig Primary und Secondary Station (combined Station)
- werden in diesem Modus betrieben, beide Stationen können Kommandos senden und warten auf Antworten. Der Ablauf ist wie bei NRM
- beide Stationen können Verbindungen auf- und abbau
- Vollduplexbetrieb

- Quittungen enthalten: die nächste erwartete Sequenznummer, RRs (ReceiverReady), PiggyBacks
- auch NACKs: falsche Sequenznummer empfangen, erwartetete im NACK
- NACK(2): Empfänger-Buffer: out-of-order Frame → speichert dieses und schickt an Sender SREJ-Frame mit Sequenznummer des fehlenden Frames, Sender schickt ab diesem alle nachfolgenden (Go-back-n)
- Sender nutzt nur die Hälfte seines Sequenznummer-Raumes (a <= 2b)
- Bei schwerwiegenden Fehlern wird FRMR 1 verschickt (obwohl kein FC-Fehler): Dekodier-Fehler, falsche Länge des Info-Feldes, bei SREJ außerhalb des Wertebereichs; S-/U-Frame haben falsche Länge; Verhalten: Frame verwerfen, RESET-Frame
- Flußkontrolle (bei I-Frame (Info-Frame)): Window-Technik, Permits mittels RR (ACK und PER!) Empfänger kann RNR-Frame (ReceiverNotReady) an Sender schicken, der stoppt sein Senden bis RR vom Empfänger
- Piggy-Back: ACK in eigene I-Frames, nur wenn eigene Sendeabsicht (sonst sinnlos)

8. X.25

8.1. Der X.25 Service

- überdeckt die unteren 3 Layers
- erstes Packet-Switching Netzwerk
- ein Interface zwischen DTE und DCE (... Circuit terminating ...)
- Verbindungsorientierter Paket-Service
- Mehrere virtuelle Verbindungen

8.2. Virtual Circuit Packet Switching

- Verbindet die Vorteile von Circuit- und Datagram- Switching
- Circuit: pre-routing Verbindungssetup, kleines Delay
- Datagram: Link sharing, Bandbreite wird besser ausgenutzt

Ablauf:
- kurze Setup-Phase, kurze Identifier pro Paket, wenig Overhead
- während Setup-Phase: jeder Router Tabelle, wie er Pakete mit Verbindung-Identifier routen soll
- Datenratenanpassung mit Paket-Switching
- nicht-blockierend, sondern Acception wird mit höherem Delay delivered
- Prioritäten

8.3. Layers

- X.21 → LAPB → Packetlayer → User Prozess (to Remote User Process)
- Physical Layer: X.21 (oft RS-232-C)
- Benutzerdaten in Pakete aufgeteilt (Blocks, Layer3)
- Segmentierung der Pakete für LAPB

8.4. Service Primitiven

- N_CONNECT.(Request, Indication, Response, Confirm)
- N_DATA.(Request, Indication)
- N_DATA_ACK.(Request, Indication)
- N_EXPEDITED_DATA.(Request, Indication) – Daten außerhalb der Reihenfolge werden sofort nach vorn genommen, für Daten die schnell übertragen werden
- N_RESET.(Request, Indication, Response, Confirm)
- N_DISCONNECT.(Request, Indication)

8.5. Virtual Circuit

- multiplexing bis 4095 VCs
- Statisches Multiplexing über TDM, full-duplex
- Zwei Verbindungstypen:
- Permanent VC: Permanente Verbindung, keine Connect/Disconnect-Phase
- Switched VC: Dynamische, temporäre virtuelle Verbindungen mit CON/DISCON-Phase
- VC-Hierarchie:
- 16 logische Kanalgruppen (LCG)
- 256 LCG-Nummern (LCN) pro LCG
- innerhalb LCG sind alle VCs vom selben Typ
- LCN werden vergeben: DCE für ankommende Anrufe, DTE für ausgehende Anrufe
- Verteilung: 4095 ... 1: Outgoing, Two Way, Incoming, PVC
- HOC (Highest Outgoing Channel), LOC, HTC, LTC, HIC, LIC, (PVC), werden während Beschreibung festgelegt
- Quelle und Ziel nutzen NICHT dieselben LCN, Netzwerk mappt das ganze (diff. LCN)

- X.25 vergibt während Request Nummern an Kommunikationspartner (= VCI), je nach VC-Art
- Routing-Table-Eintrag: VCI-in bestimmt die Nachguck-Zeile; in Zeile dann Link-Out-Nummer und freier VCI-Out (10→2 an 15, 15→1 an 10, ...), markiere diesen VCI in Tabelle
- Also: Der Link (= physik. Verbindung) ist bekannt, nur welches Gerät (VCI) ist dran → Tabelle
- Dies wird bei Request festgelegt, alle nachfolgenden Pakete gehen dann diesen Weg

8.6. LAPB: a HDLC Variant

- (Link Access Procedure Balanced)
- ist Teil von HDLC P-to-P Fullduplex Verbindung, z.B. Computer ⇔ Packet-Switched-Netzwerk (wie X.25), siehe dort
- verwendet Asynchronous Balanced Mode, RR und REJ-Frames für Fehlerkontrolle, RNR für Flußsteuerung, SREJ, PiggyBack (wegen unverzüglicher Antwort) nicht
- Extended Mode kann für LFNs verwendet werden

8.7. X.25 Adressierung

- CCITT-Empfehlung X.121: 14 BCD's (14 Binärcodierte Zahlen) mit Format:
- 3 für Country Code (z.B. 49)
- 1 für Netzwerk (z.B. 1 für TK-Netz, was ist es?)
- 10 für DNIC (Data Network ID Code)
- DTE: Adresse eines DTE (Telekom: Typ, Stadt, Adresse)

8.8. Fast Select

- bei zu langem Verbindungsaufbau
- Direktes Mapping eines N_CONNECT.Request auf X.25 Call-Request-Paket
- keine separate Call-Setup-Phase
- Call-Request-Pakete könne bis 128 Byte Daten enthalten
- Angerufenes DTE muß sich im Fast-Select-Acceptance-Zustand befinden (auf Gegenseite Fast-Select aktiv)
- Angerufenes DTE antwortet mit Call-Accepted oder Clear-Request-Paket + Daten bis 128 Byte
- Transport-Layer wird sofort informiert, deshalb schneller als normal
- Also nicht erst Aushandlung zwischen Network-Layer, sondern gleich auch an Ziel-Transport-Layer

8.9. X.75

- ist vereinfachtes X.25, beschäftigt sich hauptsächlich mit der Signalisierung der Knoten untereinander
- Prozeduren für das Internetworking mehrerer X.25 PSPDNs (Packet Switched Private Data Networks)
- Funktionalität gleich X.25 (LCGN, LCN, PVC, SVC)
- Netzwerkverbindung über Signalling Terminal Exchanges (STE)
- wird verwendet für Herstellen und Lösen virtueller Anrufe
- transparent für beide DTEs
- X.75 PLP erfordert weniger Pakettypen (keine DCE-DTE-Kommunikation, aber STE-STE-direct)
- Backbone für Netzwerke mit großen Paketen

8.10. Packet Switching

- ~ switches Virtual Circuits
- innerhalb einer Box, Multiplexing
- Packet-Switches können in drei Kategorien geteilt werden: Shared Memory, Shared Medium, Space Division

8.10.1. Shared-Memory Packet-Switch

- Input Queues ⇔ Output Queues, zentralen Controller verarbeitet pro Zeiteinheit N Anrufe und wählt N in/out – Pakete aus

8.10.2. Shared-Medium Packet-Switch

- (Ring / Bus) hat eine Kapazität, die genauso groß ist, wie der Eingangd
- die Eingänge legen ihre Daten auf das Shared Medium. Danach kommt ein Filter, dieser leitet das Paket weiter

8.10.3. Space Division Packet-Switch

- Eingangs- und Ausgangslink werden für die Übertragung des Paketes physik. verbunden.
- Vorteil: Kaum Verzögerungen, also sehr schnell
- vor dem Switch werden Buffer vorgeschaltet, da es sonst zu Kollisionen kommen kann, wenn zwei Pakete gleichzeitig ankommen
- Konflikt, wenn 2 Pakete den gleichen Weg nehmen → Sorting Network davorschalten, welches die Pakete sortiert
- freie Switch-Blöcke werden genutzt (~ Routing)
- Banyan Network steuert sich selbst
- jeder Switch weiß, wo Paket hinsoll (Adressierung)

8.11. Frame Relay

- X.25 light, kein dynamischer Verbindungsaufbau
- verbindungsorientiert, paketvermittelt, variable Paketgröße, PVCs
- X.25: Multiplexing über Packet-Layer, Link-Layer nur Error-Control
- Frame Relay: Routing/Multiplexing über Link-Layer
- schnelleres Routing (Bitrate)
- gleichzeitig mehrere Ziele
- zuerst Setup: Identifierzuweisung (DLCI – Data Link Connection Identifier), jedes nachfolgende Paket nutzt DLCI → Routing -Schlüssel
- Congestion-Control mit FECN, BECN und DE-Bit
- DLCI ist ähnlich VCI → Änderungen entlang des Pfades → lokale Signifikatoren
- Alle Frames, die zusammen auf Link multiplext sind, verbinden das Customer-Interface-Equipment zu seinem nächsten Switching-Node

8.12. ATM

- Weiterentwicklung von Virtaul Circuit Packet Switching
- Virtual Circuits → Virtual Channel Connections (VCC)
- Virtual Path Connections (VPC): mehrere VCC mit demselben Endpunkt (reduziert Steuerungsaufwand
- Vorteile von virtuellen Pfaden:
- vereinfacht die Netzwerkarchitektur
- erhöht die Netzwerkperformance uund Zuverlässigkeit
- reduziert Rechenzeit und und verkürzt VErbindungsaufbau
- mehr Netzwerkservice

9. ISDN

9.1. Einführung

- End-zu-End digitale Verbindung (Hauptunterschied zu traditioneller Telefonie)
- n*64KBit/s Bitraten, bis 2MBit/s
- über Telefonnetzwerk: Sprache, Video, Bilder, MM
- Wähl- oder Standleitung
- definiert user/network-interface-Protokolle
- alle ISDN-Geräte verwenden die selben Stecker und Signalisierungen
- mit vielen Services (voice, non-voice)
- Zugriff über begrenzte Anzahl von Standard-Mehrzweck-User-Netzwerk-Schnittstellen

9.2. Was ist ISDN, was nicht

- Regeln für Benutzerinterface zum Netzwerk
- Servicedefinitionen, Angebot für Benutzer
- macht Telefon-Gesellschaft zu Service-Anbieter
- Globale Strategie, grundlegene, weitreichende Verbesserung der Netzwerk-Infrastruktur

- Beschreibung des darunterliegenden Netzwerkes
- keine Anwendungsdefinition für die Services
- Interesse von Forschung und Industrie wecken
- eine Revolution

9.3. Standardisierung von ISDN

- ITU-TSS (früher CCITT): <Farbe>buch, I.xxx, Q.7xx, Q.9xx
- ISO: IS 8802.3 (LAN – CSMA/CD)
- ETSI: Fernmeldenorm ETS 300 102/125
- DBP Telekom (uha!) TR3, ...

9.4. logische Kanäle (logical Channels)

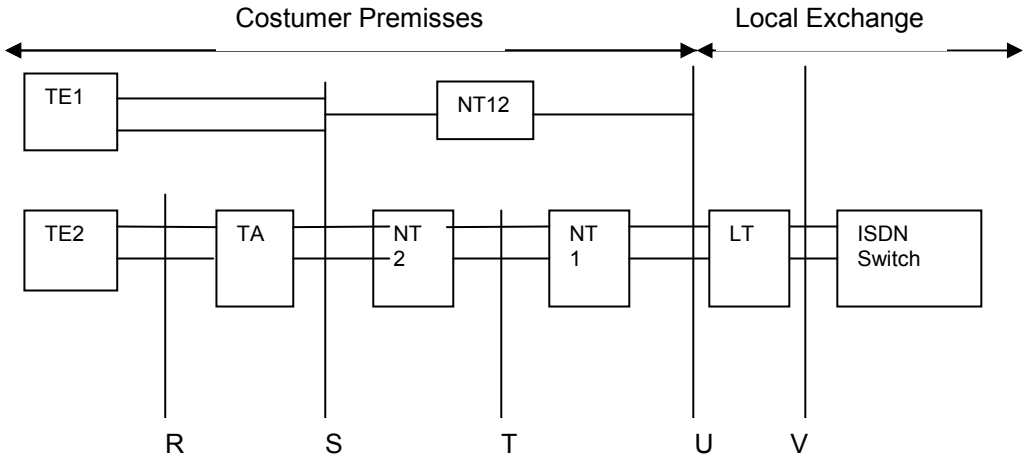
- Local: 2-Draht, Twisted Copper ist logisch in Kanäle zerteilt
- zwischen den Switching Offices: über E1 (Europa) oder T1(USA), 4-Draht → logische aufgeteilt
- mehrere Kanäle, mit TDM:
  - B (64KHz), digitaler PCM-Kanal für Sprache und Daten (US:56KHz), connection-oriented, virtual-circuit-switched
  - D (16KBit), out-of-band-Signalisierung, connectionless, Packet-switched, non-delay, auch für Userdaten
  - H (384, 1536, 1920 KBits) digital, same as D-Channel, but n*64kBps

9.5. Benutzerschnittstelle (User/Access Interface)

- digital pipes von verschiedener Größe
- zu verschiedenen Zeiten verwendet die Pipe eine variierende Anzahl von Kanälen
- ITU-T:
  - Basic Rate Interface (BRI, S_0) → 2B+D
  - Primary Rate Interface (multiplexed) (PRI, S_2M) → 30B+D in Europa (2.048 MBits, E1-Trunk)

    → 23B+D in USA/Japan (1.544MBits, T1-Trunk)

9.6. Physikalische Schnittstellen



(die Nummern stellen Referenzpunkte da)

- Terminal Equipment (TE):
  - TE1 ist für Geräte mit eingebautem ISDN-Interface (Card, Phone)
  - TE2 ist für analog-interface Geräte (analog-Phone, Modems)
- S ist 4drahtiges Interface für Entfernungen bis 590 Meter: passiver Bus für bis 8 Geräte
- Network Terminator 1 (NT1)
  - Local-Loop-Termination, verbindet Kunde mit ISDN-Switch
  - bietet S-Interface für Kunden
- Network Terminator 2 (NT2) == PABX
- NT12 ➔ NT1 und NT2
- für große Umgebungen, welche reale Verbindungen zu Telefone und Terminals brauchen kein Unterschied zu ISDN-Switch, nur kleiner
- LT (line termination)

9.7. U- Referenzpunkt:

- Aufgabe: (D, 2xB, Bittakt, Worttakt, Speisung, Akivierung, Deaktivierung, ...)
- verschiedene physikalische Layer, Firmen-(Anbieter-) abhängig
- Problem: Full-Duplex über 2 Leitungen
- Lösung: Echo Canceller with Hybrid (ECH):
  - setzt 4 zu 2 Leitungen um
  - Löscht das lokale Send-Data-Echo, um die Empfangsdaten zu extrahieren
  - Set-up Leitungsbalance mit initialem einseitigem Testen
  - kritisches Standleitungs-Balancing mit den Frameflags
  - Daten- , um Echolöschung zu verbessern
- Used Line Code: Zwei Binary, 1 Quaternary (2B1Q):
  - Framing: 18 Bit Synchronisation, 216 (12*(2B+D)) 12 Gruppen, 6 Overhead
  - 2B+D-Coding: 8-Bit-B1 + 8-Bit-B2 + 2-Bit-D = 18 Bit
  - 48 M Bits bilden den sogenannten M-Channel: Maintenance Messages, Signaling, Spannungsstatus, Fehlererkennung

9.8. Basic Rate S_0 Interface

- Funktionalität: zwischen TE/TA und NT:
  - 2B/D-Kanäle, Schrittakt, Farme-Synchronisation, Aktivierung/Deaktivierung, Speisung
  - 192 KBps, bis 8 TEs, P-to-P oder P-to-MP
  - bis 1km (P-P), 200m (P-MP)
- Konfigurationen: Bustermination ⇔ $(TE)^+$ -Bus ⇔ NT
- Physical Layer: abgewandelter AMI code (vgl. Manchester, Center Frequenz = 96KHz
- Framing: 48 Bit in 250ms

- Framesynchronisation: über Codeverletzung: Ausgleichsbits werden durch Synchronisation gesetzt (TE→ NT)
- Bus-Aktivierung/Deaktivierung: (INFO -)
  - S0: kein Pegel
  - S1: (nur TE→ NT) Dauersignal
  - S2: (nur TE ← NT) Rahmen + Kanäle B,D und D-Echo mit Dauer-“0”, A-Bit = “0”
  - S3: Rahmen + B+D transparent, logisch 1
  - S4: Rahmen + B+D transparent + E=D von TE→ NT, A-Bit=“1“

9.9. Primary Rate S Referenzpunkt

- gleiche Funktionalität wie BRI
- Phy-Layer ist unterschiedlich: Fullduplex, P-to-P, Synchron, 2 Phy Kanäle
- 1.544 (193 Bit bei 125µs) oder 2.048 (256 Bit bei 125 µs) MBps

9.10. D-Kanal Zugriff

- Beispielkonfig: TE1, TE2 und NT
- Zugriffskontrolle: Regel: Binäre „0“ (Startvon Beginflag) setzt sich gegeüber „1“ (Ruhesignal) durch
- Binäre „1“ werden von TE gezählt, Prioritätswert erreicht (z.B. x=8), HDLC von Layer 2 wird vom TE 2 gesendet
- Kollisionserkennung: (TE1, TE2, TE3 und NT)
  - TE1 sendete, dann Ruhe
  - „1“ werden von TE2 und TE3 mitgezählt
  - PrioriätswertTE2 und TE3 senden gleichzeitig Begin-Flag
  - TE2 sendet SAPI 0 und TE3 sendet z.B. SAPI 63
  - TE3 erkennt Kollision – da D-Bit <> E-Bit – und sendet Ruhesignal („1“)
  - TE2 sendet ungestört weiter
- → nicht für B-Kanäle wegen ihres exklusiven Gebrauchs

9.11. Protokoll Frames

- zwei Frame-Typen: A und B
- Bit-Stuffing von 0 bit wird in LAPD verwendet für:
  - Flag-Erkennung: wenn 6x „1“ in Daten, dann 0 darin
  - Abbruchsignal: Exakt 7x „1“ veranlaßt den Empfänger, das Interrupt-Frame zu ignorieren
  - Idle-State: 8x oder mehr „1“

9.12. LAPD Features

- Multiplexing, TE1 und SAPI
- Notwendigkeit eines Mechanismus um mehrere logische Links über D-Kanal zu multiplexen
- verschiedene Layer3 Prozesse über einen D-Kanal
- mit Service Access Point Identifier (SAPI) werden Layer3-Prozesse unterschieden
- mit TE-Identifier wird zwischen verschiedenen physikalischen TEs mit gleichem SAPI (= gleichem Layer3-Prozess) unterschieden
- TEIs sind Management-Objekte
- Broadcast-Logical-Link wird von LE verwendet, um an verschiedene TEs Daten zu senden
- D2-Adressenfeld-Format: SAPI, C/R (Unterscheidung zwischen Command/Response-Frame), TEI
- TEI-Management: (bekommt TE von Vermittlungsstelle)
  - jedes TE hat mind. ein eindeutigen TEI
  - ein TE kann mehrere L3-Services enthalten → pro TE mehrere TEI (→ verschied. X.25 Services)
  - → TEI Identity Message in LAPD UI-Frames mit SAPI=63 und TEI =127 (Broadcast) enthalten

LAPB	LABD
1-Oktet Adreßfield	2-Oktet Adreßfield
8-bit Adreßfeld wird verwendet, um Commands von Responses zu unterscheiden	C/R-Bit im Adreßfeld, um zwischen C und R zu unterscheiden
2 Timer (T1 und T3) und ein Timer-Parameter (T2)	4 Timer (T200 – T203)
3 Systemparameter (N1, N2 und k)	4 Systemparameter (N200, N201, N202, k)
nur P-to-P Konfiguration	P-to-MP und statistisches Multiplexing von verschiedenen Links
---	verwendet UI- und XID- Frames
Modulo 8(SABM) oder 128 (SABME) Sequenzen	hier: 128 (SABME)
Abbruchsignal sind 7-14 folgende „1“-Bits	hier: 7 aufeinanderfolgende „1“
Idle Channel: >=15x „1“-Bits	hier: >=8x „1“-Bits

9.13. ISDN D-Kanal Layer 3

- stellt den DSS 1 Netzwerk-Layer da
- D3-Layer Messages werden im Info-Feld der LAPD I-Frames übertragen, gleiches Format:
  - Protokoll-Diskriminator: Protokolltyp der Nachricht
  - Call-Referenz: identifiziert den betreffenden Anruf
  - Art der Message
  - die anderen Informationen hängen vom Messagetyp ab
- D3 tzu D2 Services: TE-D3 ⇔ TE-D2 ⇔ ET-D2 ⇔ ET-D3

9.14. ISDN X.25 Services

- CCITT X.31 definiert zwei Wege, X.25 mit ISDN zu verbinden: über speziellen X.31-TA-Adapter
- X.31 – A:
  - X.25 sind über ISDN-Netzwerk zu spezieller Access Unit (AU) geswitched, welche mit PSPDN verbunden ist
  - weil D-Kanal LAPD-Signalisierung am terminiert LE, wird es nur zur Netzwerk-Signalisierung verwendet
  - Paket-Mode-Transfer geht nur über B-Kanal
- X.31 – B
  - direkter Paket-Service durch ISDN-Netzwerk
  - LE erscheint für Benutzer als X.25-DCE-Node
  - ISDN stellt Paket-Handling für PSPDN zur Verfügung
  - dadurch kann X.25-LAPB und PLP in LAPD-Frames auf D-Kanal oder direkt über B-Kanal übertragen werden
  - so ist es in Deutschland

9.15. ISDN Frame Relay Abstract

9.15.1. Warum ein anderer Packet-Switching-Service

- zur Entwicklungszeit von X.25, digitale Übertragung war noch nicht zu verfügbar
- X.25 hat deswegen einen großen Protokoll- und Prozessing- Overhead zu Fehlererkennung, Flußsteuerung und Routing
- normalerweise wird das von HDLC-Controlern+(Nodes mit Microcomputern) übernommen
- dieses Problem wird mit der heutigen Verbreitung digitaler Übertragung schon von anderen gelöst

9.15.2. Frame Relay

- 64 KBps bis 2 MBps

- 2 Techniken eingeführt: Frame Relay und Frame Switching
- wurde auf die Erweiterung des ISDN-Signaling (Q.921/LAPD)aufgebaut, bekannt als LAPF/ Q.922 (LAP-Frame-Mode)
- Anruf-Steuerung-Signalisierung wird auf separaten logischen Verbindungen übertragen
- Multiplexing und Switching von logischen Verbindungen findet im Layer 2 statt
- ein DLCI (Data Link Connection Identifer) wird zu Identifizierung der logischen Verbindung verwendet
- keine Hop-by-Hop- Fluß- und Fehlersteuerung
- Reduziert die Funktionalität und Prozessing im Benutzer-Netz-Interface
- im Netz selber kommt es zu weniger Verzögerungen und höherem Durchsatz
- Unterschiede zu X.25:
  - X.25: Phy-Layer → LAPD/B → X.25 Packet-Level (alles im Netzwerk)
  - Frame Relay: Phy-Layer → LAPF-Core → LAPF-Steuerung (nur das nicht im Netzwerk)
  - Frame Switching: Phy-Layer → LAPF-Core → LAPF-Steuerung (alles im Netzwerk)
- LAPF:
  - EA-Bits: Identifizierung des Types / Länge des Adreßfeldes
  - C/R-Bit: Command/Response, verwendet von Anwendung, nicht von LAPF
  - FECN-Bit: Netzwerk-Benutzer-Interface-Information über Stau, Empfangssystem soll den Datenstrom aus höheren Layern reduzieren
  - BECN-Bit: Netzwerk-Benutzer-Interface-Information über möglichen Stau, Empfangssytem selber soll seinen Datenstrom reduzieren

9.16. ISDN BONDING

- QoS kann erhöht werden, wenn man mehr als einen B-Kanal verwendet → invers Multiplexer
- Standard: Bandwidth ON Demand INter-operability Group (BONDING) → viele audiovisuelle und Daten-Anwendungen
- BONDING-Vorgehen:
  - der Inverse-Multiplexer macht n individuelle Anrufe (jeder 64 KBps)
  - erster (Master-) Kanal dient dem Management während dem Setup, später für Daten
  - unterstützt Framing, CRC, Management-Kanal
  - FIFOs werden für die Kompensation der Zeitverzögerung zwischen den Kanälen verwendet
- BONDING-Modes:
  - werden während des Setups identifiziert, kleinster gemeinsamer Mode wird verwendet
  - mode 0: kein Kanal-Delay-Einigkeit, für Geräte, die eigene haben
  - mode 1: initiale Delay-Einigkeit. Danach wird Framing und Control-Overhead entfernt → volle Bandbreite verwendbar. Alle n*64 Anrufe müssen im Fehlerfalle neu hergestellt werden
  - mode 2: Framing und Control Overhead wird beibehalten. Zur dynamischen Kanal-Justierung, Synchronisation, Fehlererkennung und dynamische Bandbreiten (~ on demand)
  - mode 3: Mix aus 1 und 2: für feste Bandbreiten von n*64 KBps gibt es n+1 Verbindungen. die n+1 wird für Framing- und Control-Information verwendet

9.17. Jenseits ISDN: xDSL

- HDSL: billige T1/E1 Alternative, zwei Drahtpaare, POTS und ISDN nicht auf selben Draht verwendbar
- S-HDSL: ein Drahtpaar
- ADSL: Multimedia, asymmetrisch, flexible Datenraten, parallel zu POTS, ISDN in einem 160KBps-Kanal von ADSL
- CAP (Carrierless Amplitude/Phase Modulation): QAM-Variante, ohne Träger, FDM für up/down
- DMT (Discrete Multitone Modulation): Multicarrier-System, 32 Kanäle für upstream und 256 für downstream, 4KHz pro Kanal. Jeder Träger verwendet eigene Modulation und Steuerung
- VDSL: kürzere Reichweite, für kleine Street-Exchanges

9.18. B-ISDN

- mit ATM als Switching-Technologie
- ATM eben (zusätzlich noch Phy-Layer-Framing auf diesen Folien

10. Internetworking

10.1. Internetworking Probleme

- Welche Funktionen werden dem Benutzer angeboten?
- diese alle gleich zugreifbar?
- Sicherheitseinrichtungen?
- Transparenzlevel: Neue Ressourcen durch exsitierende Mechanismen
- Protokoll-Kompatibilität
- Anforderungen an Performance-Fähigkeiten
- Politische Erwägungen
- Effektives Management
- Effektive Fehlerisolation
- accross-Ressourcen-Nutzung
- wie werden neue Technologien aufbenommen
- **Address-Namens-Schemata**: meist hierarchisch, auch flach
- **Routing-Techniken**: meistens Unterschiede in Datagram-Netzwerken, abh. von Stau oder anderen
- **Information Quanta**: Informationen sind Pakete, Messages, Blöcke, ...Werden im Nummerierung-Schema für die Netzwerk-Steuerung (z.B. Sequenznummern) verwendet
- **Paketgröße**: verschied. Max-Größen, Zerstückelung, Wiederzusammensetzung, wenn zu groß
- **Verbindungsorient./-los**: manche nur eines. Internetworking: beides, unabh. vom Service
- **Fehlerkontrolle**: keine bis zuverlässig, Internet-Protokolle sollten eigene haben, wenn die der anderen unzureichend
- **Flußsteuerung**: basiert meist auf Paketverlust, oder Windows, Probleme wenn unterschiedlinche Quanta
- **Timing**: Schutz vor Deadlocks oder anderen Fehlern, entsprechende Algo's, abh. von Delays, Antwortzeiten
- **Unterbrechungen**: Signalisieren Ausnahmebedingungen, Fehler-, Flußsteuerung, Kompatibilitätsprobleme
- **Statusanzeigen**: unterschiedlich: sollte nur für autorisierte Parteien zugänglich sein, Billing/Accounting-Info, für den Internetworking-Service sind eigen Accountings notwendig
- **Zugriffskontrolle**: Polling, Selektion, Reservierungen, ..., manchmal an spezielle Begrenzer oder Flags im Datenstrom gebunden. Über das gesamte Netz notwendig
- **Close-Prozeduren**: Manche Netzwerke versuchen mit aller Macht sicherzustellen, das die Übertragungsdaten einer einseitig geschlossenen Verbindung trotzdem ankommen. Andere hoffen beim Schließen auf vorhergehene Ankunft der Daten

10.2. Netzwerk-Layer

- Layer 1: Repeater
- Layer 2: Bridges (zwischen Netzen, empfängt Pakete vom Netz, sep. Coll. Domains)
- Layer 3: Router
- Layer 4: Protocol converters (Gateway)

10.3. Bridges

10.3.1. Vorteile der Bridges

- verkleinert die LAN-Segmente: bessere Zuverlässigkeit, Verfügbarkeit, Services
- beseitigt die Phy. Beschränkungen der Interkonnektivität, erhöht dadurch Anzahl möglicher Stationen und Segmente → toll für große LANs
- Framebuffering: ermöglicht verschiedene MAC-Protokolle, Mix verschiedener LAN-Basen
- Funktionalität unterliegt der MAC-Subadresse im Frame, sie werden transparent für höhere Layer

10.3.2. Nachteile von Bridges

- Store-and-Forward-Delay, vergl. Repeater
- keine Flußkontrolle im MAC-Layer, Bridges könne durch viel Traffic überladen werden → noch mehr Frames speichern
- MAC-Inhalt muß dem anderen MAC-Layer angepaßt werden → neue Frame-Check-Sequenz in jeder Bridge, führt dazu, das Fehler, die beim crossing entstehen unentdeckt bleiben

10.3.3. Probleme beim Bridging

- z. B. fällt Token Ring-Priorität beim umwandeln in CSMA/CD weg
- ...

10.3.4. Routing mit FDB (forward data base)

- Bridges lernen durch Traffic-zuhören, wo Stationen sind, und welche Pakete sie wohin weiterleiten müssen
- FDB ist eine Liste von Stationen, mit denen die Bridge kommuniziert: Wenn eine Zieladresse im LAN eines aktiven Ports A in FDB eines anderen aktiven Ports B, dann Paket von Bridge dorthin gebroadcasted
- Wenn Bridge nicht weiß, wohin mit Paket → Broadcast über alle aktiven Ports, richtiger Empfänger meldet sich → FDB-Update. Für den Rückweg merkten sich die Bridges die Quelladresse
- Problem: Zwei Bridges, beide Enden verbunden, an den verbunden Enden je eine Station: Beide Bridges empfangen also ein Nachricht von A und leiten sie an B weiter. Bridge a empfängt die Nachricht von Bridge a und vice versa, beide glauben A befände sich bei B → Antwort von B nicht weitergeleitet, Lösung: baumartige Verbindung (Spanning Tree)
- **Spanning-Tree** (über BPDUs bestimmt): Bridges haben Root-Port → der, wo die Root-Bridge dranhängt
- Es gibt *designated* Bridges (haben höheren Anspruch) pro LAN, also jedes LAN hat einen *designated* Port, der mit den geringsten Kosten, dann mit der höheren Priorität: nur diese dürfen Frames weiterleiten, die anderen werden blockiert
- diese besonderen Bridge wird periodisch wiederholt bestimmt
- wenn Bridge BPDU empfängt:
- Ist der Port, wo sie empfangen wird, *designated,* hat der Transmitter einen Anspruch, um *designated* zu sein?
- Soll dieser Port mein Root-Port sein?
- dient im allgemeinen dazu, daß jede Bridge zu jeder Zeit weiß, wo die Root-Bridge ist, und wieviele Hops sie von ihr entfernt ist, dies wird diesen BPDU-empfang-änder-weiterleit-Algo sichergestellt. Die Bridge, die am nächsten an Root-Bridge ist, wird *designated* Bridge.

10.4. Netzwerke

10.4.1. Services

- Connectionless, -oriented
- Layer-Struktur (network layer)
- SNICP: subnetwork independent convergence protocol
- SNDCP: subnetwork dependent convergence protocol
- SNDAP: subnetwork dependent access protocol
- subnet layering
- Internet sublayer
- Subnet enhancement sublayer
- Subnet access sublayer
- Verbindungslose WAN-Verbindung von LANs über z.B. X.25 (Layer 1-3)
- Subnet enhancement sublayer
- Verbindungorientierte X.25: Network-layer-Mapping:

Network Service Primitive	X.25 Action
N-CONNECT.request N-CONNECT.indication	Send CALL REQUEST CALL REQUEST arrives

N-CONNECT.response N-CONNECT.confirm	Send CALL ACCEPTED CALL ACCEPTED arrives
N-DISCONNECT.request N-DISCONNECT.indication	Send CLEAR REQUEST CLEAR REQUEST arrives
N-DATA.request N-DATA.indication	Send Data Packet Data Packet arrives
N-DATA-ACKNOWLEDGE.request N-DATA-ACKNOWLEDGE.indication	No Packet No Packet
N-EXPEDITED-DATA.request N-EXPEDITED-DATA.indication	Send INTERRUPT INTERRUPT arrives
N-RESET.request N-RESET.indication N-RESET.response N-RESET.confirm	Send RESET REQUEST RESET REQUEST arrives none none

10.5. Sockets

10.5.1. Allgemein

- Socket-API von 1981 von BSD 4.1 Unix
- bietet Kommunikation zwischen Prozessen über Netzwerk, die Prozesse können auf verschiedenen Maschinen laufen
- vergleichbar mit Dateien–I/O-Programmierung mit großen Unterschieden
- Abstraktion aller möglichen Netzwerkprotokolle, -topologie
- Schnittstelle wird vom Betriebssystem erwaltet
- Transport-Services (haupt): unzuverlässig-Datagram, zuverlässig-Stream, raw-Daten
- basiert auf Client-Server-Modell

10.5.2. Client-Server-Modell

Eigenschaften
- einfachst
- bietet Ressourcen-Sharing
- einfaches Maintaining
- der Client kontaktiert Server, erfragt bestimmten Service
- Server stellt Service je nach Interesse des Client zu Verfügung
- Beispiele: Datenbanken, Webserver, Videoserver, FTP

Server
- Prozeß irgendwo, mehrere auf einer Maschine möglich
- öffnet Kommunikationskanal, zeigt OS Requestbereitschaft an
- wartet auf Client-Requests, wenn einer kommt, dann bedient und antwortet, kehrt in Wartemodus zurück

Client
- öffnet Kommunikationskanal
- stellt Verbindung zu bestimmten Serverprozeß her
- sendet Service-Request, erhält Antwort
- schließt Kanal

10.5.3. Die Socket-API

- jedes Socket hat separate send/receive-Puffer, ein Port-ID und von Anwendungen änderbare Parameter
- Socket-Operationen sind implementiert als System-Calls im OS
- User/Kernel-Bindung muß hergestellt werden
- Jeder Socket muß eine lokal eindeutige Port-Nummer zugeteilt bekommen haben

10.5.4. Socketoperationen

- **int socket (*int family, int service, int protocol*)**
- **int bind (*int socket, struct sockaddr *localaddr, int addrlen*)**
- localaddr: Socket-Verbindung zu wem → lokale Adresse, sockaddr ist Protokoll-abhängig
- **int connect (*int socket, struct sockaddr *destaddr, int addrlen*)**
- destaddr: Zieladresse
- **send, sendto, write**
- **recv, recvfrom, read**
- **listen (*socket, maxtime*)**
- **accept**: Gibt neue Socket-ID zurück, zu welcher der Client verbunden ist
- **close (*socket*)**

10.5.5. Winsock

- basiert auf BSD-Sockets
- unterstützt Stream/Datagram-Sockets
- gleiche API
- Unterschiede:
  - Erweiterung für asynchrone Programmierung
  - verschiedene Fehlercodes
  - Socket- und Datei-Identifier sind unterschiedlich
  - read/write/close sollten nicht verwendet werden, sondern spezielle Operationen

10.5.6. CAPI

- Common-ISDN-API
- API-Standard, um ISDN-Geräte anzusprechen, die mit BRI und PRI verbunden sind
- basiert auf Q.931
- einfache Anrufeigenschaften (call set-up, clear-down)
- unterstützt B-Kanal Daten-/Spracheverbindungen
- unterstützt verschiedene logische Datenverbindungen innerhaln einer phy. Verbindung (MP)
- Auswahl verschiedener Services und Protokolle während Verbindungs-Setup und auf ankommende Anrufe hin
- transparente Schnittstellen für Protokolle über Layer 3
- unterstützt mehrere BRIs und PRIs auf einem ISDN-Adapter
- mehrere Anwendungen
- Betriebssystem-unabhängige Nachrichten
- Asynchroner, Ereignisgesteuerter Mechanismus
- Operation: Eine Anwendung kann mehrere Controller verwenden, mehrere Anwendungen können sich einzelnen oder mehrere Controller teilen
- Message-Basiertes Interface: Nachrichten gelangen zur Anwendung über Queue, eine für jede Anwendung, Nachrichten gelangen zum CAPI über eine Queue

10.6. Connection Management

10.6.1. Allgemein

- die Protokolle auf beiden Seiten managen die Statusinformationen, um Fehler-, Flußkontrolle, usw. durchzuführen
- C. M. kann in drei Phae geteilt werden:
- Initialisierung der Verbindung auf beiden Seiten
- während Datentransfer Status zu entwickeln
- Statusinformationen zurücksetzen, wenn keine Daten mehr übertragen werden

10.6.2. Connection Establishment

- Verbindungsinformationen auf beiden Seiten in Records (Identifier, Timers, History, ...)
- Zwei-Wege-Handshake(Req, Ind, Res, Con) für sichere Verbindung, nur unter den Bedingungen darunterliegender Services

- unzuverlässiger Service: wegen langen Verzögerungen können Conf's falsch ausgelegt werden
- → delayed Duplicates, Lösung:
- wenn keine Verbindung existiert und der Responder empfangen kann, dann spätes Con.Res ignorieren
- wenn eine Verbindung existiert, kein Paket früher geschlossener Verbindungen akzeptieren innerhalb der aktuellen Verbindung
- begrenzte Lebensdauer der Pakete wegen Seq-wrapping-Gefahr, die Begrenzung auch für neue Verbindungen anwenden
- Paket-Identifier kann Seq.nr. oder Connection+Seq.nr sein
- dazu ist Drei-Wege-Handshake notwendig, synchronisiert Initiator und Responder
- jedes Paket hat also eine Seq.nr, im ACK ist diese als Verweis, Sender entscheidet aufgrund ACK, wie er weiter sendet
- bei Systemabsturz muß Initiator innerhalb der Lebensdauer aller PDUs die Verbindung wieder aufbauen (wie lang muß dann die Lebensdauer sein, um das zu garantieren?)
- oder Verbindungs-Identifier: Initiator und Responder bauen zusammen neuen Connection-Identifier
- wenn nicht, dann kann man Seq.nr. verwenden, die von einem Timer bestimmt werden:
- Timer muß unter allen Umständen laufen (Systemuhr)
- senden nur so schnell, wie Uhrtick
- Sender zu langsam: Resynchronisation nötig, wenn verbotene Regionen erreicht werden

10.6.3. Connection Releasing

- Abrupt: Datenverlust, und sonst? → DISCONNECT-Req.
- Nicht nötig, wenn jede Seite weiß, das die andere Seite alle Daten empfangen hat

Timer basiert

- Implizites Verbindungs-Setup, keine Setup-Verzögerung
- Sender erstellt Verbindungs-Record, startet Timer und sendet
- erste PDU enthält Begin-Flag, nach senden dieses wird der Timer gestartet
- wenn Timer ausläuft, bevor ACK, dann Retransmit dessen, nach n Retransmissions, Sender gibt auf
- Empfänger erstellt Verbindungs-Record, sobald er das PDU mit Begin-Flag erhält, nachfolgende werden nur anerkannt, wenn sie in-Sequenz sind
- beide Seite starten ihre Timer nach senden/empfangen einer PDU neu
- kumulative ACK: Verbindungserhaltung
- Verbindungs-Record wird gelöscht, wenn bestimmte Zeit abläuft (Sender: 3MPL + Retransmit-Time + Receiver-Hold-Time, Receiver: 2MPL + Retransmit-Time + Receiver-Hold-Time)
- Pakete ohne Verbindungs-Record werden ignoriert

10.6.4. Timeouts absschätzen

- RTT
- in heterogene Netzwerken variiert RTT stark
- RTT immer messen aus altem und neuen: Durchschnits-RTT: **RTT= p*Old + (1-p)*New**
- $0 < p < 1$ ($p \sim 0.8$)
- new **Timeout = b * RTT, b>1**
- Problem: ACKs von retransmittierten Paketen
- Lösung: nicht verwenden (nach KARN)
- Backoff-Strategie: Bei Retransmissions, Timeout erhöhen
- Timeout wird zurückgesetzt, wenn RTT wieder abgeschätzt werden kann (non Retransmit-ACK)

11. Internet

11.1. Architektur

- IP verwendet den kleinsten gemeinsamen Service, der von allen Netzwerken unterstützt wird: unzuverlässiger Datagramm-Service
- TCP hat zuverlässige Übertragung, UDP nicht
- Internet-World-View:
- subnetwork: ATM, Ethernet, ISDN
- network layer: IP, IPng, (CLNP?)
- transport layer: UDP, TCP, ...
- application l.: http, ftp, telnet, RTP
- control: RSVP
- management: SNMP
- Verzeichnis: DNS
- → no session, presentation!
- klein lokale Netze werden über Backbones verbunden (=Verbindung einer Ansammlung von Netzen)

11.2. Terminologie

- internet: Ansammlung von Packet-Switching-Netzwerken, die mit Routern untereinander verbunden sind
- (the) Internet: öffentliche Verbindung von Netzen
- Endsystem = Host: mit Netz ⇔ Router verbundener Computer
- Router = Gateway = Zwischen-System: leitet Pakete weiter, mehrere verschieden Schittstellen
- Sub-Netzwerk: Teil des internet (z.B. einzelnes Ethernet)
- Firewall: Internal- ⇔ External-internet-Verbindung, Sicherheit!
- Name: Wer?
- Adresse: Wo?
- Route: Wie hinkommen? (Nur im Network-Teil)
- IP-Adresse: Netzwerk-Name + -Adresse (unteilbar, unveränderbar), jeder Rechner hat eine eigene
- Class A-C

11.3. Adressierungs-Architektur

- Chaotischer Aufbau, nicht nachvollziehbar, wo Netz ist, → LISTE!
- Subnetting: Network-Subnet-Host
- Router kennen haben nur Network in ihrer Tabelle, zu Subnet lokal, zu Host noch lokaler
- Core-Router ⇔ Border-Router

11.4. Routing

- im Internet gibt es mehrere autonome Systeme
- interior Gateway verbindet Subnet mit a. System
- exterior Gateway verbindet a. System mit Core-Netzwerk
- Host kennen Routen für Ziele, die nicht in ihrem Netz sind
- Interior Gateways leiten Pakete ihn ihren angeschlossenen Netzen weiter, oder andere interiore Gateways innerhalb ihrer Netze
- exterior Gateways leiten Pakete an interiore GWs weiter oder durch das Core-Netz zu anderen
- exteriore GWs tauschen Request-Response-Messages aus
- Interiore GWs innerhalb eines a. Systems updaten ihre Tabellen untereinander
- Routing Information Protocol (RIP)
- Open Shortest Path First (OSPF) ~ Dijkstra
- Router bestimmen die Grenze der Übertragungskapazität, schneller mit LookUp-Tables

11.5. Namesauflösung

- DNS: Hierarchische Namesraum
- Hierarchie: Top-Down, Suchen: Bottom-Up
- lokaler Names-Server → diese kennen Root-Server, mind. 1 Parent-Server
- Anfrage: Name, Antworttyp, Flag: rekursiv oder iterativ
- Antwort: Antwort, oder Angabe über nächsten Hop
- Mapping (Names ⇔ IP): verteilt über mehrere Server, meist lokal, zuverlässig: keine Fehler, Autoritäts-Zonen
- DNS: Name → IP; ARP: IP → MAC
- Name: Human-suitable, IP: Routing, MAC: innerhalb des Netzes
- Wie MAC zu IP? → ARP:
  - nutzt Ethernet für Broadcast
- Hosts senden ARP-Requests, ARP antwortet mit IP- und Ethernet Adresse des Senders und Empfängers
- ARP Antworten werden von den Nodes gespeichert
- Wenn sich die Adresse ändert, muß sich auch der Name ändern

11.6. IP Header

11.6.1. Fixed

- Protocol Identifier
- Header Länge
- VErsion
- PDU Lifetime
- Flags: SP=1: Segmention permitted, MS=1: (middle Segment) nur im letzten Segment =0, ER: Error Report
- Typ (Error- oder Daten-PDU)
- Segmentlänge
- Checksum

11.6.2. Adressenfelder

- Dest Address
- Src Address
- Adressenlänge

11.6.3. Segmentation Felder (nur wenn SP gesetzt)

- Data Unit Identifier
- Segment Offset
- totale Länge

11.7. Connectionless Network Protocol, Options

- Source Routing (loose, strict)
- Route Recording
- Timestamps
- Security (?)

11.8. Error Reports (Spezielles Protokoll)

- übertragen mittels IP Abschätzung (verschiedene Paket-Codes)
- enthält Fehlerursache
- für Diagnosen, wie Überlastentdeckung (viele Verluste)
- Echo-Request/Response-Fähigkeit

11.9. Fragmentation

- Reassembling sollte an der Zieladresse erfolgen, da die Zwischenadressen nicht bei jedem Paket über den gleichen Weg erreicht werden, sind dort dann längeren Wartezeiten möglich

11.10. IP Version 6 (IPv6)

11.10.1. Ziele

- Löst IPv4-Problem mit der steigenden Anzahl von Subnetzen und Hosts, Sicherheitsfragen, QoS
- untzerstützt Milliarden von Hosts
- verkleinert die Routing-Tabellen
- mehr Sicherheit (Authentifizierung)
- bessere Unterstützung für Serviceklassen, bes. Realtime
- Multicast-Support
- ünterstützt beide Protokolle parallel
- Eigenschaften

11.10.2. Eigenschaften

- 16 Byte Addresse (4mal IPv4)
- vereinfachter Header aus 7 Elementen
- zusätzliche Parameter durch Optionen
- Pakete werden gleich behandelt (demokratisch)
- Gruppen von Pakete werden als zusammengehörig markiert

11.10.3. Header

- Version
- Priorität: 0-7 beste Bemühungen, 8-15 für konstante Raten (mit Verlusten)
- Flow Label: später, für pseude connections mit richtigen Eigenschaften
- Nutzlastlänge
- Nächster Header: Welcher Header ist der nächste. Dieser hier ist fixed, die nachfolgenden für Erweiterungen
- Hop-Limit: TTL in IPv4

11.11. Mobilität in IP-Netzwerken

11.11.1. Problem

- Netzwerke sind überall essential
- Tragbare Computer werden immer beliebter: Drahtlose Netzwerke beschleunigen deren Netzzugriff
- am besten Laptop gleich als Internet-Host, dann Netzwerkzugriff von überall möglich
- Wenn aber woanders?

11.11.2. Lösung für heute

- Schrittweise:
- besuchter Netzwerk-Administrator muß IP-Adresse vergeben (Annahme:sein Netz ist sicher)
- Manuell umkonfigurieren bezgl. neuer Umgebung, oder DHCP
- Natürllich verliert man die alte Identität (Home-Adresse), deswegen keine passive Erreichbarkeit mehr → kein Mobilitätsunterstützung seitens IP
- Also (Tunneling): Home-Agent erhält die alte Adresse und bekommt lokale Updates
- Pakete zum mobilen Host gehen zur Home-Adresse, Home-Agent encapsuliert dieses, schickt diese weiter zum visited Agenten (VA), dieser decapsulated diese und leitet sie zum mobilen Host weiter

11.12. UDP / TCP

- UDP hat die Fähigkeit, TCP die Bandbreite streitig machen, weil bei Netzüberlast TCP zurücksteckt, UDP aber nicht, so TCP Bandbreite „klaut“ und sie ungern wieder hergibt
- UDP: Datagramm, verbindungslos, Checksumme, wenn gewünscht, unzuverlässig
- TCP: siehe BBNT-Vortrag, genau dasselbe, bzw. weniger, noch einfacher

11.13. Representation Layer

- beschreibt Syntax und Semantik der übertragenen Daten
- Problem: verschiedene Computer benutzen unterschiedliche Darstellungen für gleichen Datentyp (Länge, Bitreihenfolge usw.)
- Definition von abstrakten Datenstrukturen
- zusätzlich können Verschlüsselung und Kompression stattfinden
- Möglichkeiten:
    - jeder Computer muß für die Umsetzung in die Darstellung jedes anderen Computers eine Routine implementieren
    - allgemeine Darstellung wird genutzt, die jeder versteht

11.13.1. ISO Abstract Syntax Notation 1 (ASN.1)

- allgemeine Darstellung, definiert komplexe Datentypen für Anwendungen
- zwei Regeln notwendig:
    - abstrakte Syntax: Definition der Datentypen die übermittelt werden sollen
    - Transfer Syntax: Definition, wie Datentypen übertragen werden
- für jeden Datentyp existiert globaler Name (type identifier)
- jeder Datentyp ist in Libary mit entsprechendem Namen gespeichert → beschreibt die Struktur des Typs (Pascal-artige Syntax)
- ein konkreter Wert wird mit Type-ID und gegebenenfalls zusätzlichen Informationen (z.B Stringlänge) übermittelt
- Einfache Type z.B. INTEGER, BOOLEAN, REAL usw.
- zusammenfassen von einfache Typen mit SET, SEQUENCE usw.
- Übertragung besteht aus: Type-ID, Länge des Datenfeldes, Datenfeld, End-Flag (bei dynamischer Länge)

11.14. Application Layer

- verschiedene Service-Protokolle die von vielen Anwendungsprogrammen benutzt werden
    - Services: virtuelles Terminal, E-Mail, Dateitransfer, Drucken, 3W
    - HilfsServices: Remote Procedure Call RPC, CCR

11.14.1. RPC

- Client/Server-Ansatz: Client will Aktion ausführen, auf anderem Rechner (Server) der den Dienst anbieten kann (z.B drucken)
- Client übermittelt Request-Message
- Server empfängt Request und führt in gegebenenfalls aus
- Das Ergebnis wird an den Client zurückgeschickt
- Ort der Ausführung ist für Client transparent
- Aufruf findet über spezifizierte Prozedure-Interface statt
    - local Call: Rechner führt Aufruf selbst aus
    - remote Call: stub-Prozedure wird lokal ausgeführt, diese ruft dann bei Server die entsprechende stub-Prozedur auf, die dann lokales Äquivalent ausführt
- RPC-Parameter: call-by-call, call-by-reverence (aber es muß gesamter Datenbereich übertragen werden, also auch die Größe bekannt sein), komplexe Datentypen mit Pointern können nicht übermittelt werden
- Client findet Server durch fest kodierte Adressen oder Server meldet sich an oder Client fragt nach bestimmter Prozedur

Probleme

- kein Server vorhanden, Server abgestürzt, Nachricht verloren gegangen

- Client kann erneute Anforderung senden (at least ones semantic)
- Fehler an Programm melden (at most ones semantic)
- oder irgendwas anderes (z.B. 37 Versuche unternehmen)
- idempotente Funktion: mehrmaliges Ausführen bringt gleiches Ergebniss → Semantik egal
- wenn nicht idempotent → exactly once semantic notwendig
- orphan (Waise): Aktion des Servers, die ins „Leere“ geht, weil Client abgestürzt ist
- Problem: Client ist schnell wieder online und wiederholt Call
    - Extermination: alle Call werden in Logfile gespeichert, orphans können erkannt werden → großer Aufwand
    - Reincarnation: nach bestimmter Zeit (Epoche) werden Call ungültig
    - gentle Reincarnation: Server fragt erst nach, ob Call gelöscht werden soll
    - Expiration: Client wartet bestimmte Zeit T bevor, er neuen Call tätigt
- Sun-RPC: benutzt TCP oder UDP, at least once semantic
- OSI-ROS (Remote Operation Service): stellt Serviceprimitiven zur Verfügung sowie verschiedene Operationsklassen

11.14.2. Transaction Processing

- Problem: wenn viele User auf gleiche Ressourcen zugreifen kann es zu „race conditions“ kommen
- Lösung: unteilbare Aktionen → wenn ein User Aktion ausführt, ist die Aktion für alle anderen blockiert (Semaphoren)
- atomic action: unteilbar, nur erfolgreich - sonst passiert nichts, kann aus normalen und atomic Aktionen bestehen

Ablauf

- Master schickt Anforderung an Slave
- Slave sichert Systemzustand und führt Aktion aus und schickt Ack
- wenn Master Ack erhalten hat, bekommt Slave eine „commit“-Nachricht
- Slave nimmt jetzt erst die Änderungen am Systemzustand vor

11.14.3. Virtual Services

- abstrakte, maschinenunabhängige Definition von Services
- Vorteil: nur der Anbieter braucht eine Abbildung von virtuellem Service auf eigenes System vornehmen
- Virtual Terminal Services: Anwendung greift auf virtuelles Terminal zu, entsprechender Treiber setzt die Funktionen dann auf die reale Hardware um
- Scroll Mode Terminals: ohne eigenen Prozessor, PAD (Packet Assember / Disassembler) übernimmt Kommunikation mit Host-Computer
- Telnet: stellt Terminal auf entferntem Computer per TCP/IP her (ASCII, zeilenweise)

11.14.4. File Managment

- Fileserver bieten Computern Filesystem über ein Netz an
- remote access: Operationen auf die Dateien werden auf dem Server ausgeführt
- unchangeable files: Client kann nur lesen, Daten lokal ändern und nur als neu Datei auf Server ablegen (uneffizient)
- Problem: gleichzeitiger Zugriff mehrere Clients und caching bei Clients
- OSI FTAM (File Transfer, Access and Management): Framework für Network File Systeme
- FTP: komplette Dateien transferieren, löschen, umbenenne, Verzeichnis wechseln und anzeigen
    - TCP Port 21 für Verbindungskontrolle (ASCII-Kommandos)
    - anderer Port für Datenübertragung

11.14.5. Directory Services

- stellt Informationen über verschiedene verfügbare Dienste / Geräte im Netzwerk dar
- jedes Objekt hat eindeutige Adresse und eine Klartext-Namen
- ISO 9594, CCITT X.500
- Directory System Agents (DSA): haben Informationen über alle verfügbaren Objekte

- Directory User Agents (DUA): holen sich Informationen vom zuständigen DSA
- Objekte sind hierarchische angeordnet, jeder DSA ist für einen Teil zuständig (verteiltes System)
- jedes Objekt hat verschiedene Attribute
- DS ermöglicht Zugriffskontrolle im Netz

11.14.6. E-Mail

- Mail User Agent: stellt Interface für Benutzer zum Lesen, Schreiben und Verschicken von e-mails zur Verfügung
- Mail Transfer Agent: transportiert Mails vom Sender zum Empfänger
- Mail besteht aus:
    - Envelope (Umschlag) → Informationen über Quell- und Zieladresse, Priorität u.ä.
    - Content (Inhalt) → Mail-Header mit Subject, Empfänger und Inhalt (Text, Attachments ...)
- Transfer Agent des Senders übermittelt Daten an Transfer Agent des Empfängers
- SMTP : Simple Mail Transfer Protocol RFC 812
- SMTP benutzt ASCII-text
- Empfänger muß nicht permanent online sein, da Transfer Agent E-Mail zwischenspeichert (mailbox)

11.14.7. World Wide Web

- Hypertext System → Links zu weiteren Texten
- 3W-Browser fordert Seiten vom 3W-Server per HTTP an
- HTTP: Hypertext Transfer Protocol (benutzt TCP)

11.14.8. SNMP

- Simple Network Management Protocol
- wird zum Austausch von Informationen benutzt
- besteht aus 4 Elementen:
    - managed node: Geräte (Router, Drucker, PC) das überwacht und konfiguriert wird
    - managment stations: steuert Management
    - managment information base (MIB): Baumstruktur mit Informationen über Gerät
    - managment protocol: Definition der Schnittstellen und Datenstrukturen
- Auf den Nodes läuft der SNMP-Agent
- wenn Station keinen eigenen Agent ausführen kann → Proxy-Agent auf anderem Node
- MIB wird von jeder Station geführt (RFC 1213)
- MIB benutzt Untermenge von ASN.1 für Objekte → SMI: Structure of Management Information

Ablauf

- Management Station sendet Request an Agent
- Agent antwortet mit den geforderten Informationen oder bestätigt update
- Get-Request → anfordern von Informationen
- Set-Request → Variablen updaten
- SnmpV1-trap → Agent schickt Meldung an Management Station, wenn bestimmter Fall eingetreten ist

Sonstiges

Quality of Service

- Durchsatz (mean, peak - batch length)
- Verzögerung / Delay (mean, maximal, jitter, skew)
- Fehlerrate (bit, PDU)
- Verlust, Dublizierung, Misordering -Rate von Paketen
- garantierte QoS: Parameter sind nie schlechter als vereinbart (Vertrag, technisch festgelegt)

erwartete QoS: Parameter sind in bestimmten Bereich (nach Erfahrung, Modellen usw.)