Datenkommunikation
Ausgearbeitete Fragen
SS2008

REITER BERNHARD

Anmerkung:
Die Nummerierung der Fragen entspricht jener, die Hr. Lindner in seinen Fragebögen verwendet. Im Klartext bedeutet dies, dass die Fragen 9-37 bis 9-49 eigentlich zum Kapitel 10 (OSPF) gehören und die Fragen 10-1 bis 10-26 dem Kapitel 11 (TCP,UDP) zuzuordnen sind.

Manche wenige Fragen wurden mithilfe der Ausarbeitung vom letzten Jahr (von Betina MIKSCH) beantwortet. Vielen Dank an dieser Stelle.

Kommentare und Verbesserungsvorschläge sind gerne gesehen – an reiterbernhard@hotmail.com oder im ET Forum (http://www.et-forum.org/index.php?showtopic=3656)

Bei manchen Fragen habe ich als Anmerkung (Skizze x-x) geschrieben, da ich eine Skizze bei der Beantwortung für diese Fragen als sinnvoll empfinde. Bei einigen Gelegenheiten verweise ich auch auf die jeweiligen Folien.

Viel Erfolg beim Lernen und bei der Prüfung!

Kapitel 1 Transmission Principles

1-1)
Parallele Übertragungstechnik wird hauptsächlich innerhalb von Computersystemen (z.B. auf zwischen Motherboard und CPU) eingesetzt. Dabei werden – je nach Größe des Datenwortes (8bit, 16bit) für jedes Bit Leitungen zur Verfügung gestellt (sog. BUS). Weiters gibt es noch zusätzliche Leitungen für das „Clock“ Signal zur korrekten Abtastung und „signal reference“ als Kontrolleitung(en) (Empfänger bereit zur übertragung usw..). Es gibt also generell 3 BUS Systeme für die parallele Datenübertragung:

-Datenbus
-Adressbus (gibt an wohin Daten gesendet werden)
-Control-bus (Clock, Signal reference, Interrupt…)

Die Taktsynchronisation erfolgt über die Kontrolleitung “Clock” – Die Clock wird also ständing mitübertragen.

Skizze 1-1

1-2)
Serielle Übertragungstechnik wird meist zwischen 2 Endsystemen über größere Distanzen verwendet. Sämtliche Daten werden bit-für-bit über die Leitung übertragen und im Empfänger wieder in Nutzdaten umgewandelt. Serielle Übertragungstechnik kostet deutlich weniger als parallele Üb.Technik – dafür sind aber spezielle Mechanismen zur korrekten Übertragung und Wiedergewinnung der Daten erforderlich.
Eben solche Mechanismen ermöglichen es den Signaltakt direkt aus dem Übertragenen Datenstrom wiederzugewinnen, was eine zusätzliche Leitung für die „Clock“ und somit Mehrkosten nicht mehr erforderlich machen.

Skizze 1-2

1-3)
Bit Synchronisation ist bei Serieller Übertragungstechnik erforderlich, damit der Empfänger des Datenstroms die einzelnen Bits zum richtigen Zeitpunkt (im besten Fall in der Mitte der bit-cell) abtastet und somit die empfangenen Daten richtig interpretiert.
Aus Kostengründen wird bei Serieller Üb. meist keine seperate Clock Leitung verwendet um zu synchronisieren.
Der Signaltakt wird aus dem empfangenen Datenstrom generiert. Dafür wird eine sog. „Clock Recovery Circuit“ verwendet, die den Signaltakt über Signalflanken im empfangenen Datenstrom herstellt. Um die Funktion dieser Schaltung zu gewährleisten müssen Protokolle verwendet werden, welche sicherstellen, dass im Datenstrom nicht zu oft ‚0‘ oder ‚1‘ übertragen wird… es muss also für immer wieder auftretende Signaländerungen gesorgt werden.
Dabei gibt es 2 wesentliche Methoden..

-)asynchrone BitSynchronisation
-)synchrone ----„--------

1-4)
Bei Asynchroner BitSynchro wird nur jedes Datenwort einzeln Synchronisiert

Datenwörter können also einzeln gesendet werden und werden einzeln synchronisert (deshalb insgesamt asynchron… das Intervall zwischen 2 Datenwörtern ist variabel)

Für diese Art von Synchronisation wird die Technik des Start/Stop Bits verwendet – z.B. ist der „Idle Zustand“ der Leitung ‚1‘ dann gibt eine Signaländerung von 1→0 das Start Bit an – es folgt der Datenstrom(NRZ), wessen Ende durch ein Stop-Bit (einmal ‚1‘ oder zweimal ‚1‘ hintereinander) angegeben ist.

Ein einziges Datenwort aus z.B. 8 Bit enthält also 3 Bits allein für die Synchronisation, was sehr ineffizient ist.

1-5)
Oversampling heißt im Empfänger mit einer höheren Abtastrate abzutasten, als der eigentliche Datenstrom hat (im einfachsten Fall 2 mal höher) . Durch Mittelung der Samples entsteht ein Datenstrom höherer Frequenz als der ursprünglich gesendete.

Die eventuell auftretenden Alias Effekte können somit mit billigeren Filtern mit nicht all zu steiler Filterkurve herausgefiltert werden…

Oversampling wird bei asynchroner Bitsynchronisation betrieben um die Kosten geringer zu halten und die Fehlerrate zu senken.

1-6)
Bei der Synchronen Bit-Synchro wird der Empfänger für die Dauer von ganzen Datenblöcken (Datenstrom) Synchronisiert und verbleibt in diesem Zustand für längere Zeit – dies spart vor allem die redundanten Start/Stop Bits bei der Asynchronen Synchronisation ein, da Synchronisationsmechanismen nur am Anfang und am Ende eines Datenstroms benötigt werden.

Dazu wird eine sog. PLL (Phase-Locked-Loop) in Verbindung mit einem VCO (Voltage Controlled Oszilator) als kompletter Regelkreis verwendet um den Signaltakt im Empfänger zu erzeugen. Um eine dauerhaft Korrekte Frequenz am Ausgang dieses Regelkreises zu gewährleisten muss sichergestellt werden, dass die PLL nicht „ausrastet“ – es müssen genügend Signalflanken im Datenstrom vorkommen (zuviele ‚1‘ oder ‚0‘ hintereinander führen zum ausrasten der PLL). Dies erreicht man durch 3 Methoden:
-Scrambling der daten
-Codierung, sodass jedes Bit eine Signaländerung enthält (Manchester, Diff. Manchester, FSK) in LANs
-Codierung, sodass Signaländerungen regelmäßig im Datenstrom auftreten (HDB3, NRZI, RZ, AMI) in WANs

Skizze 1-6

1-7)
Die Phase-Locked-Loop in Verbindung mit einem VCO (Voltage Controlled Oscilator) wird bei der Synchronen BitSynchro. als Regelkreis verwendet, um aus Signaländerungen im Datenstrom den Signaltakt zu generieren. Dabei muss darauf geachtet werden, dass genügend Signaländerungen im Datenstrom vorhanden sind, damit der Regelkreis nicht „ausrastet“ und eine falsche Frequenz als Signaltakt an den Empfänger übergibt.

Skizze 1-7 = Skizze 1-6

1-8)

Manchester Code: in LANs verwendet.
-1 Bit wird in 2 Halbbits aufgeteilt.
-Das erste Halbbit ist das ursprüngliche Invertiert , das zweite hat den Wert des ursprünglichen
-Damit tritt eine Signaländerung in der Mitte eines jeden (ursprünglichen) Bits auf
-1→0 heißt ‚0‘ , 0→1 heißt ‚1‘

Skizze 1-8

NRZ: einfachste Art der Codierung – der Datenstrom enthält nur Nutzdaten.
-‚1‘ heißt logisch 1
-‚0‘ heißt logisch 0

Gleichstromanteil:
Ist bei NRZ evtl vorhanden
Bei Manchester immer 0

Dafür benötigt der Manchester Code die doppelte Bandbreite wie der NRZ Code (Doppelte Frequenz tritt im Manchester Signal auf)

1-9)

HDB3 (High Density Bipolar 3) Code: in WANs verwendet
-3 verschiedene Signalzustände
-aus logisch 1 werden alternierende Pulse erzeugt… logisch 0 erzeugt niemals Pulse (Signal bleibt 0)
-Für Folgen von ‚0‘ gibt es spezielle Vorschriften:
> 4 ‚0‘ werden mit speziellen A oder V Bits versehen
>V Bits stellen „Code Violations“ dar… da während eines 0 Bits ein Puls erzeugt wird
> durch bestimmte Regeln mit A und V Bits wird Gleichstromanteil im Signal verhindert

(Skizze)
Skizze 1-9

NRZ: einfachste Art der Codierung – der Datenstrom enthält nur Nutzdaten.
-‚1‘ heißt logisch 1
-‚0‘ heißt logisch 0

HDB3 braucht genausoviel Bandbreite wie NRZ, hat allerdings keinen Gleichstromanteil.

1-10)
Scrambler werden dazu verwendet um einen Nutzdatenstrom zu verändern, damit er bestimmte Eigenschaften für einen Übertragungskanal annimmt. Im Falle der Synchronen Bitsynchronisation wäre dies eine regelmäßige Änderung von Signalflanken.

Scrambler verwenden entweder sog linear rückgekoppelte Schieberegister oder fixe Tabellen um einen Nutzdatenstrom in ein „Pseudo-Zufalls-Signal“ mit derselben Information zu erstellen. Aufgrund der Einfachheit von Scrambling stellt es kein Kodieren eines Signals dar.

1-11)
Siehe Skizze 1-11

Beschreibung:
Sync: Bitmuster zur Rahmensynchronisation…oft 01010101…
SD: Starting Delimiter – markiert den Anfang des Datenrahmens.
Control: Informationen über das Verwendete Protokoll/Inhalt/Adresse/Länge des Rahmens
Data: Die eigentlichen Daten (auch Payload genannt)
FCS: Frame-Check-Sequence zur Erkennung und ggf. Behebung von Bitfehlern. (z.B. Parität)
ED: Ending Delimiter – markiert das Ende des Datenrahmens.

1-12)
Das Control Feld im Datenrahmen kann Informationen über:

-verwendetes Protokoll
-Rahmentyp (Daten, Acknowledge , Not-Ack, Verbindungsaufbau/ende usw…)
-die Prüfsumme des Rahmens für Fehlererkennung/korrekturen
-Adressierung des Rahmens
-Länge des Rahmens
u.a.

enthalten.

1-13 & 1-14)
Framesynchronization ist erforderlich um Anfang und Ende der übertragenen Rahmen (innerhalb des gesamten Datenblocks) zu erkennen, und somit Daten/Kontroll-Signale(Rahmen) voneinander zu unterscheiden.

Das SD (Starting Delimiter) und ED (Ending Delimiter) Bit darf dazu nur am „echten“ Anfang und am „echten“ Ende des Rahmens auftreten um Fehler in der Übertragung und Interpretation der Daten zu vermeiden. Es muss also Mechanismen geben um dies zu gewährleisten. (in dem Fall ist Data Transparency nicht gegeben)…

…Daten-Transparenz heißt, dass bestimmte Bitmuster (die gleich aussehen wie jene von Control Characters/Feldern) im Datenstrom auftreten dürfen.. Dies wird durch

-byte stuffing mit bestimmten „characters“
-bit stuffing
-code violations (Diff.Manchester Code, JK Symbole)
-byte count Technik im DDCMP Protokoll
-verschiedene Zustände der Leitung: „idle“ für „warten“ – danach Synchronisationsmuster 010101… vor speziellem SD Bitmuster ; und „idle“ Zustand nach Speziellem ED Bitmuster (Ethernet)

gewährleistet.

<u>1-15)</u>
Datentransparenz (bit-oriented) wird mit Bit-Stuffing erreicht. Nach einer langen Folge gleicher Bits wird automatisch ein anderes Bit eingefügt und beim Empfänger wieder herausgenommen.

Z.B. SD, ED entspricht der Bitfolge 01111110 (0, 6*1, 0)

Alles nach dieser bestimmten Bitfolge wird als der Anfang des Rahmens Interpretiert (Control+Data+FCS). Tritt innerhalb des Rahmens 5*‘1‘ auf wird automatisch eine ‚0‘ hinzugefügt - sodass erst gar kein ED+SD im Rahmen auftreten kann. Erkennt der Sender tatsächlich das Bitmuster ‚01111110‘ im Datenstrom – überträgt er dieses unverfälscht und der Empfänger erkennt SD/ED.

Skizze 1-15

<u>1-16)</u>
Charakter orientiertes Byte-Stuffing durch Einfügen eines bestimmten Kontroll-Zeichens(a) **vor** einem Kontrollzeichen(b) um (b) „wirklich“ zu übertragen.

Byte Stuffing mit ASCII Code:

NUR DLE+SOH Byte bedeuten „Anfang des Frames“ im Rahmen
NUR DLE+STX Byte bedeuten „Anfang des Textes“ im Rahmen
NUR DLE+ETX Byte bedeuten „Ende des Textes“ im Rahmen
NUR DLE+DLE (oder Formatzeichen) bedeuten „DLE“ oder „Formatzeichen“ im Datenblock

Somit kann auch ein einzelner Kontroll-Character ohne DLE im Payload auftreten, und wird dadurch nicht ausgeführt. (Data Transparency)

1-17)
Bei der Forward Error Controll Methode wird dem Datenstrom genügend redundante Information hinzugefügt um dem Empfänger ein Erkennen (detect) **und** Korregieren (correct) des Fehlers zu ermöglichen. Forward Error Control ist bei Netzwerken, wo eine extrem niedrige Bitfehlerrate (z.B. Weltraumverbindungen) gefordert ist, oder eine lange Verzögerung (Delay) zwischen Sender und Empfänger besteht, im Einsatz.

Dabei werden sog. Reed Solomon und Hamming Codes verwendet. (ECC Feld vor ED im Rahmen)

1-18)
Bei Feedback Error Controll wird dem Datenstrom genügend redundante Information hinzugefügt um dem Empfänger **nur ein Erkennen** eines Fehlers (mittels FCS – Frame Check Sequence) zu ermöglichen.

FCS wird nach einer bestimmten Regel im Sender und Empfänger unabhängig voneinander für jeden Rahmen erstellt und mit übertragen. Stimmt die FCS des Empfängers mit der des Senders überein, ist der Datenrahmen fehlerfrei.

Wenn der Empfänger einen Fehler erkannt hat – fordert er den Sender auf, den entsprechenden Rahmen/Block erneut zu senden.

Die FCS kann mit verschiedenen Methoden erstellt werden:
-parity even/odd
-Summe der datenwörter modulo 2
-CRC

Je nach Komplexität der Methode zur Erstellung der FCS ergibt sich eine
%-Zahl für nicht erkennbare Fehler in einem Rahmen oder eine
% Zahl für (erkannte Fehler) / (gesamte Fehler)

1-19)
Fourier – Prinzip : ein beliebiges periodisches Signal kann als eine Summe von Sinus und Cosinus Schwingungen dargestellt werden… Im Falle eines perfekten Rechteck Signals ist die Anzahl der Sinus und Cosinus Anteile unendlich.

Skizze 1-19

Auf einer realen physikalischen Leitung wird ein übertragenes Signal immer gedämpft (Attenuation). Wenn alle Frequenzanteile gleich stark gedämpft wären, ergäbe das Ausgangsignal dieselbe Form wie das Eingangssignal , nur in der Amplitude abgeschwächt. Im realen Fall werden die Frequenzanteile jedoch bis zu einer gewissen Frequenz (Grenzfrequenz) fast gar nicht gedämpft, alle höherfrequenten Anteile jedoch stark gedämpft…

…Eine Grenzfrequenz kann durch das Übertragungsmedium selbst auftreten, durch Filter in der Übertragungsstrecke implementiert sein oder durch einen Übertragungskanal mit bestimmter Bandbreite gegeben sein.

1-20)
Siehe 1-19.

Auf einer realen Leitung werden die verschiedenen Frequenzanteile eines Signals unterschiedlich schnell übertragen und kommen somit zu verschiedenen Zeitpunkten am Empfänger an (Verzerrung).

Dies kann durch die Angabe von ISI (Inter Symbol Interferenz) im sog. Augendiagramm visualisiert werden.

1-21)
Skizze 1-21

Ein Informationssignal wird auf einer realen Leitung gedämpft, gefiltert, verzerrt und verrauscht.
Bei der Wiedergewinnung erfolgt die Abtastung idealerweise in der Mitte einer bit-cell. Durch die auftretenden physikalischen Effekte kann es (bei gleichbleibender Abtastrate) zu Bitfehlern kommen.
Es besteht daher ein Zusammenhang zwischen maximaler (fehlerfreier) Bitrate, Leitungslänge, Grenzfrequenz auf einem Übertragungskanal.

1-22)
Das Nyquist Theorem gibt die maximal erreichbare Bitrate für einen Rauschfreien Übertragungskanal an

R = 2 B log(2) (V)

R = bitrate (bit/s)
B = Bandbreite für bandbreitenbeschränkte Übertragung
V = Signalstufen eines Symbols (=2 für binäre Üb.)

1-23)
Das Shannon Theorem gibt die maximale Bitrate für einen verrauschten Übertragungskanal an

Max R = B log2 (1+S/N)

R = bitrate (bit/s)
B = Bandbreite für bandbreitenbeschränkte Übertragung
S= Signalpegel
N = Rauschpegel

SNR = Signal to Noise Ratio = 10* log10 (S/N) [dB] als Kenngröße für Übertragungskanäle.

1-24)
Baseband Mode = Basisband Übertragung.

Die gegebene Leitung wird nur für **einen** Übertragungskanal zwischen Sender und Empfänger (im Basisband) genützt – die Bandbreite ist nicht begrenzt.

Signale sind meist Rechteckförmig

Bitrate wird durch Senderleistung / Empfängerempfindlichkeit , SNR und die phys. Eigenschaften der Leitung bestimmt.

Es ist lediglich Bitsynchronisation mit keinem Gleichanteil und niedriger EM Strahlung erforderlich.

1-25)
Narrowband Mode = Schmalbandübertragung

- Die Übertragungsstrecke ist in mehrere Kanäle aufgeteilt es wird ein Kanal mit begrenzter Bandbreite verwendet– deshalb muss das Signal entsprechend moduliert werden, um andere Kanäle nicht zu stören.

Die Modulation erfolgt mittels Modems (Modulator-Demodulator) – bestimmte Geräte, die bestimmte Signalparameter für den Kanal aufbereiten (Amplitude bei ASK, Frequenz bei FSK, Phase bei PSK – oder auch als Kombination QAM für heutige Modems)

1-26)
Broadband Mode = Breitbandübertragung
- Die Übertragungstrecke ist auf mehrere Kanäle aufgeteilt und es werden mehrere Kanäle geringerer Bandbreiten auf derselben Leitung übertragen.

-Für analoge Systeme heißt Breitband, dass jeder Kanal mit einem anderen Träger moduliert ist und somit ein gewisses Frequenzband abdeckt. – In Summe gesehen decken mehrere Kanäle ein breites Band ab.

-in digitalen Systemen heißt breitband oft nur „highspeed“ .. in dem Sinne, dass einem Kanal eine hohe Bandbreite zur Verfügung steht und somit hohe Bitraten erzielt werden können.

Kapitel 2 Protocol Principles

2-1)
Das 3 Schichten Modell verdeutlicht den Kommunikationsvorgang zwischen 2 Sytemen.

Skizze 2-1

Als Anwender kommuniziert man über die app.SW von A nach B. Um Informationen übertragen zu können gibt die app.SW Daten an die communication SW von A weiter, welche mit einem best. Leitungsprotokoll Daten über die Comm. HW überträgt. Im Empfänger (B) läuft dieser Vorgang beim Empfang andersrum ab.. B kann aber auch Daten senden. Vorteil dieses Systems ist, dass die Applicationsoftware quasi von dem Leitungsprotokoll ausgekapselt ist und sich somit um weniger Daten kümmern muss… Sie erhält die (für sie) notwendigen Daten über ein „Service“ der Communication Software.

Bei einem Connection-less Service sind in der Comm.SW lediglich die notwendigsten Aspekte integriert (Frame protection, error detection, Frame synchronization).

Es stellt daher das Service mit dem geringsten Aufwand (best effort) dar.
Die comm.SW muss nicht all zu komplex sein, keine speziellen Frames sind erforderlich…

dafür muss eine Fehlerkorrektur von der Applikation Software darüber betrieben werden.

„Der Sender sendet die Daten an den Empfänger ohne Rückmeldung“

2-2)
Siehe 2-1

Beim Connection-orientated Protokoll überwacht das Prot. den Status der Leitung und sendet/empfängt Daten in Abhängigkeit davon.

Vor der Übertragung wird eine „logische Verbindung“ mittels Connection-Request – connection acknowledged zwischen den beiden Systemen aufgebaut. Dafür sind aber auch spezielle Frametypen erforderlich.

Erst dann werden die Daten übertragen – wobei sich das line Protokoll der comm.SW um Fehlererkennung und Korrektur kümmert (mittels Feedback Error Controll). Fehlerhafte Datenblöcke werden mit sog. **A**utomatic **R**epeat Re**Q**uest (ARQ) Methoden angefordert.
Dadurch verringert sich der Aufwand für die app.SW – sie kann sich auf die comm.SW bzgl. Übertragung verlassen….

…Um diese Dienste bereitstellen zu können, ist die connection Software komplexer als beim CL orientated Service (flow controll, error recovery, conn.&line management).

2-3)

Automatic **R**epeat Re**Q**uest (ARQ) ist nur bei einem connection-orientated Protokoll möglich.. da nur diese Art Protokoll Feedback Error Controll betreibt.
ARQ ist also eine Anwendung von Error Controll. Es stellt den fehlerfreien Empfang aller Datenblöcke sicher.

Prinzipielle Eigenschaften:

Korrekt empfangene Frames werden dem Sender mittels spezieller Acknowledgement (ACK) Meldung (in entgegengesetzter Üb. Richtung) mitgeteilt.

Jedes üb. Frame wird im Sender in einem sog. „retransmission Buffer“ gespeichert, bis der Sender die ACK Meldung für die jeweiligen Frames empfangen hat.

Nach einem gewissen Timeout für ein best. Frame wird dieses erneut übertragen und auf eine ACK dafür im Sender gewartet.

Um diese Methode zu implementieren müssen Frames Nummeriert/markiert werden, um die richtige Reihenfolge, und doppelte Frames erkennen zu können.

Verwendet werden ARQ Methoden in…

TCP (Transmission Controll Protocol)
HDLC (High Level Data Link Control)
DDCMP (Digital Data Link Control Management Protocol)

2-4)

Idle RQ ist neben Continuous RQ eine der 2 ARQ Arten.
Ablauf einer Übertragung bei Idle RQ:
Skizze 2-4

Zwischen dem Senden bzw. Empfangen der einzellnen Frames vergeht sehr viel Zeit, da erst nach einem ACK der Sender das nächste Frame sendet.

Es kann also immer nur entweder der Sender oder der Empfänger Daten übertragen (half duplex). Eine full duplex Leitung kann also nicht genutzt werden. Die Performance kann durch eine zus. NACK(N) Nachricht vom Empfänger leicht verbessert werden.

Idle RQ ist daher im Vgl. zu C-RQ sehr langsam, benötigt aber nur wenig Ressourcen. Es wird heutzutage noch im TFTP (Trivial File Transfer Protocol) verwendet.

(2-4 forts.)

Ablauf1: Normale Übertragung

Skizze 2-4

Wie oben beschrieben – I(N) wird gesendet, I(N+1) erst dann gesendet wenn Sender ACK(N) empfängt.

Ablauf2: I Frame gestört

Skizze 2-4

Wenn I Frame nicht empfangen wurde, wird auch kein ACK(N) gesendet, Timeout beim Sender → erneute Übertragung

Ablauf3: Ack Frame gestört

Skizze 2-4

Wenn ACK(N) Frame den Sender nicht erreicht, hat für ihn dies diesbelbe Bedeutung wie vorhin → I(N) wird nach Timeout erneut gesendet. Empfänger muss das doppelt empfangene I(N) erkennen, erneut ACK senden, und verwerfen).

<u>2-5)</u>

Beim Continuous RQ kann eine Full duplex Leitung komplett ausgenutzt werden, da der Sender nicht auf ACK für bereits gesendete Datenrahmen wartet, sondern kontinuierlich weiterüberträgt .. sich aber gesendete Rahmen in einem retransmission buffer merkt. Wenn der Sender nun eine ACK(N) Nachricht erhält, wird das Frame N aus dem Buffer gelöscht. Weiters braucht der Receiver eine receive list um doppelte Frames zu erkennen und ggf. die Reihenfolge der Frames umzuordnen.

Man braucht also:

- Tansmission Buffers
- Receive Buffers
- Retransmission Timers

2-6)

Selective Ack. ist eine Art der Error Controll in Continuous RQ .

- Jedes Frame wird einzelln mit ACK bestätigt (kein multiple ACK für mehrere Frames)
- Frames werden bei nicht empfangenen ACK nach einem Timeout erneut gesendet und am Ende des Retransmission Buffers gespeichert.
- Dadurch kann es für den Empfänger erforderlich sein doppelte Frames zu erkennen (Empfang OK, aber ACK beim Sender nicht empfangen) bzw. Frames umzuordnen (Empfang Not OK , nach Timeout → inzwischen mehrere andere Frames empfangen)
- Jeses gesendete Frame startet einen Timer → bei Timeout erneutes Senden.

2-7)

Skizze 2-7

Skizziert sind 2 Methoden:

I Frame gestört, weitere I Frames gesendet… Sobald der Sender ACK (N+2) empfängt während N+1 noch im Retransmission Buffer gespeichert ist, sendet er Automatisch N+1 erneut und wartetauf ACK(n+1) … selbes Verhalten hat das System auch bei Szenario 3 (ACK Frame gestört, weitere gesendet)

I Frame gestört, keine Weiteren gesendet.. Sobald für das gestörte und nicht bestätigte I Frame der Timer im Sender ausläuft wird es erneut gesendet und auf ein ACK (N+1) gewartet. .. selbes Verhalten hat das System auch bei Szenario 4 (ACK frame gestört, keine weiteren Ack gesendet) .. Nur bei diesen beiden Szenarien wird ein Timer benötigt.

2-8)

Das Prinzip bei GoBackN ist „beginn von vorne“

- Das heißt, wenn Üb.Fehler auftreten wird jedes Frame seit dem letzten korrekt empfangenen (N) mittels NACK(N) neu angefordert..
- alle Frames (N+2…) die dazwischen empfangen werden, werden vom Empfänger verworfen, bis (N+1) korrekt empfangen wird → kein neu-ordnen im Empfänger erforderlich, erkennen von Duplikaten erforderlich…SW im Empfänger kann einfacher gehalten werden.
- Ein ACK(N+2) kann bei gestörtem ACK(N+1) , aber richtigem Empfang von N+1 auch N+1 bestätigen → multiple ACK.

(2-8 forts)

- Für jedes Frame N wird im Sender ein Timer gestartet, der beim Ablaufen und nicht Empfangen eines ACK oder multiple ACK das jeweilige Frame erneut sendet.

- GoBackN wird in HDLC verwendet

2-9)

Skizze 2-9

Das System verhält sich bei Variante 2+4 gleich: Nach Timeout für N+1 wird dieses erneut gesendet und auf ein ACK(N+1) gewartet. Bei Variante 4 muss ein Duplikat von N+1 verworfen werden.

Variante 1: Sobald der Empfänger N+2 empfängt, N+1 aber nicht sendet er NACK(N+1) und verwirft alle Frames (N+2,3, usw) .. Sender sendet alles ab N+1 erneut.

Variante 3: Empfänt der Sender ACK(N+1) nicht, aber N+2 bedeutet dies multiple ACK für N+1 und N+2…

Variante 5: Sobald der Empfänger nach Aussenden von NACK(N+1) weitere N+2 , N+3 bekommt, wird NACK(N+1) erneut ausgesendet – bis er N+1 empfängt.

Timer sind bei Varianten 2 und 4 sowie 5 erforderlich. Bei 5 hängt es davon ab, ob weiter Daten nach dem NACK(N+1) empfangen werden.

2-10)

Bei Positive Acknowledgement werden die Frames so lange bestätigt, wie sie in der richtigen Reihenfolge am Empfänger ankommen. (multiple ACK möglich, erkennen von Duplikaten erforderlich)

Sobald die Reihenfolge beim Empfänger nicht mehr stimmt, sendet er kein ACK mehr aus, empfängt aber alle weiteren Daten trotzdem (neuordnen erforderlich)

Jedes Frame löst im Sender einen Timer aus, der gestoppt wird wenn ACK für das Frame empfangen wird.. bei Timeout wird dieses Frame erneut gesendet.

2-11)

Skizze 2-11

Variante 1: Sobald empfänger ein Frame erhält, dass nicht der Reihenfolge entspricht hört er auf ACK zu senden, empfängt weitere Frames → Timout → N+1 erneut gesendet … alle bis dahin empfangenen Frames mit multiple ACK bestägtigt, neu-ordnen erforderlich

Variante2: So wie 1 nur ohne neu-ordnen

Variante 3: multiple ACK → Reihenfolge bleibt erhalten

Variante 4: Timeout für N+1 , da ACK nicht empfangen → N+1 erneut gesendet → Duplikat muss verworfen werden.

2-12)

Bei Selective Reject werden die Frames so lange bestätigt, wie sie in der richtigen Reihenfolge am Empfänger ankommen. (multiple ACK möglich, erkennen von Duplikaten erforderlich)

Sobald die Reihenfolge nicht mehr stimmt wird das fehlende Frame vom Empfänger explizit angefordert mittels SREJ(N) (dadurch Neuordnung von Daten mögl. Erforderlich)

Jedes Frame startet im Sender einen Timer ,der beim Empfang von ACK für das Frame zurrückgesetzt wird.. Erfolgt ein Timeout (ACK nicht empfangen) wird dieses Frame erneut gesendet.

2-13)

Skizze 2-13

Variante
1 – Nachdem N+1 nicht empfangen wurde, N+2 aber schon wird SREJ(N+1) gesendet → gleichzeitig ACK(2) für den Sender… N+1 wird erneut gesendet , und mit ACK(höchster Frame) bestätigt → multiple ACK, neuordnen

2- Timeout, gestörter Frame erneut gesendet, kein neuordnen

3- multiple ACK, kein Timeout,kein neuordnen

4- timeout & duplikat muss verworfen werden

5- timeout für das Frame, welches mit SREJ(n) angefordert wurde → neu ordnen

2-14)

Damit ARQ funktioniert werden 4 Größen für die Nummerierung der Frames benötigt:

N(S) send sequence number
N(r) receive sequence number

Register Variablen V(s) im Sender und V(r) im Empfänger.
Das Zusammenarbeiten dieser Größen wird Anhand von GoBackN erklärt.

V(S) gibt die Nummer des Frames an, welches als nächstes gesendet wird
V(R) gibt die Nummer des Frames an, welches als nächstes empfangen werden sollte
N(S) ist die Nummer für das Frame, welches gerade gesendet wird
N(R) ist die Nummer für das Frame, welches Acknowledged wird.

Vor dem Senden wird V(S)=N(S) gesetzt, danach wird V(S)+1 erhöht.
Bei GoBackN muss N(S) = V(R) sein, damit der Empfänger das Frame akzeptiert. Danach wird V(R) um 1 erhöht und ACK mit N(R) gesendet.
Das Frame 0 wird also mit ACK (1) bestätigt. Ein ACK(N) bei GoBackN heißt, dass alle Frames bis (N-1) bestätigt sind. Dies ist deshalb so, weil bie GoBackN das ACK sowohl für Bestätigung als auch für die Keepalive Msg. Verwendet wird (so kann der Sender zwischen keepalive und ack eines neuen Frames unterscheiden).

Skizze 2-14

2-15)

Um Full-duplex Leitungen besser ausnutzen zu können werden Acknowledgements nicht als eigene Frames übertragen sondern als Flag in einem Informationsrahmen.

Ein Inforahmen von A→ B enthält also Info für B + ACK für die Daten von B
Ein Inforahmen von B→ A enthält also Infor für A + ACK für die Daten von A

Man erkennt piggyback ACK daran, dass im Protokoll Header sowohl eine Send Sequence Numer (für A→B) als auch eine Receive Sequence Number (für B→A) auftritt. Es werden dabei deshalb 4 Register (2 im Empfänger und 2 im Sender) als auch größere Frames benötigt.

Dafür eröht sich die Performance einer Full-duplex Leitung deutlich.

2-16)

Bei Continuous RQ muss es ein Limit für die noch-nicht-bestätigten-Frames im Sendebuffer geben, da dieser sonst eine „unendliche“ Größe benötigen könnte (sowie eine sehr hohe Anzahl an Identifiers für die Frames). Deshalb gibt W bei Windowing die Anzahl der gespeicherten Frames im Sendebuffer an – wenn das Fenster voll ist, werden Daten nicht mehr gesendet, sondern auf ACK gewartet, sodass das neue Frames in das Window (Sendebuffer) hineingelangen können. Das Fenster wandert also mit dem ACK des Empfängers.

Windowing reduziert die Buffergröße auf W*maxFrameSize.. sowie die Anzahl der benötigten Identifiers (je nach ARQ Methode.. GoBackN W+1, Sel. ACK auf 2W)

Dadurch lässt sich die Nummerierung der Sequence Numbers mit Modulo(W+1) bzw. Modulo(2W) handhaben.

2-17)

Windowing wurd durch eine Limitierung des Sendebuffers auf W Frames realisiert. Wenn der Sendebuffer voll ist, „schließt“ sich das Window – es wird solange gewartet sprich keine neuen Daten gesendet, bis Daten im Window Acknowledged werden. Geschieht dies, wird das Window wieder „geöffnet“ und neue Daten werden gesendet.

Man kann sich das Window also als Ausschnitt des Datenstroms vorstellen, der mit dem ACK von einzellnen Frames den Datenstrom abwandert (sliding/jumping Window).

Vor dem Window sind die bereits akzeptierten Daten
Im Window stehen die gesendeten, aber noch nicht akzeptierten daten, sowie Daten, die alsbald gesendet werden (usable Window) … Danach stehen die Daten, die erst gesendet werden können sobald das Fenster weiterwandert.

Je nach verfügbarer Bandbreite, Bitrate und Distanz bestimmt die Windowgröße ob es sich um ein Sliding Window handelt (ACKs kommen genau rechtzeitig an um das Fenster immer offen zu halten) oder ein Jumping window.

Skizze 2-17

2-18)

Um einen Datenstrom bit für bit auf eine Leitung zu bringen (seriell zu übertragen) benötigt man eine gewisse Zeit (=Serialization Delay)

Ts = [(Number of Bytes)*8 / (Bitrate/s)]*1000

Signale auf Leitungen benötigen eine gewisse Laufzeit, um den Empfänger zu erreichen – dies spiegelt sich im Propagation Delay wieder

Tp = [(distance in m) / (velocity in m/sec)] *1000

Dadurch lässt sich eine „Länge“ errechnen… Sie gibt an- wie lange die Strecke ist, die ein Bit auf der Leitung einnimmt

L (m) = [(1/ (bitrate)] * [velocity]

Man kann also bei bekannten Daten errechnen wieviele Bits sich zu jedem Zeitpunkt auf der Leitung befinden (was für die Berechnung einer geeigneten Windowgröße erforderlich ist).

2-19)

Mittels Serialization Delay, Propagation Delay und response Time des Empfängers lässt sich die sog. „round-trip-time“ RTT errechnen – Sie geibt an wie lange ein Bit vom Sender zum Empfänger und wieder zurrück braucht.

RTT = 2*(Ts+Tp) + responsetime

Multipliziert mit der zur Verfügung stehenden Bandbreite erhält man das optimale (Volumen) für den Kanal (so viele Bits können zu jedem Zeitpunkt auf der Leitung sein) – was der Window größe in Byte entspricht.

Delay Bandwith Product = RTT*BW = Windowsize in Byte.

Ist das Window genau dem DBP kommen die ACKs genau zu dem Zeipunkt an, wo gerade das „letzte“ Byte im Window gesendet wurde → es kann also gleich nach dem „letzten“ Window Byte das nächste übertragen werden → sliding Window.

Wenn das Window kleiner als DBP ist, ist dies eben nicht der Fall.. Kapizitäten bleiben ungenützt → jumping Window

Deshalb sollte das Window mind. dem DBP entsprechen.

2-20)
Der Wert des Retransmission Timers wird durch:

Bitrate
Maximum Data Frame Size
„worst-case“ Zeit, die der Empfänger benötigt um ein ACK zu generieren
Größe des ACK frames

beeinflusst.

Dieser Wert kann fix sein, wenn die Übertragungsstrecke zu jeder Zeit konstant bleibt.
V.a. bei Drahtlosen Übertragunen mit variabler „Leitungslänge“ ist das nicht möglich.

2-21)
Wenn Daten am Empfänger schneller ankommen als die comm.SW die Daten an die app.SW übergeben kann bzw. die app.SW die Daten zu langsam verarbeitet werden diese automatisch verworfen (obwohl sie i.O. waren). Diese verworfenen Frames lösen evtl Timeouts aus, wodurch die Daten erneut gesendet werden – und erneut evtl wieder nicht empfangen werden können. Deshalb muss der Receiver den Datenfluss regulieren (flow control).

Prinzipiell wird flow Controll durch STOP und GO messages vom Receiver zum Sender (bzw. bei full-duplex zus. mit umgekehrter Richtung) realisiert.

Wenn es zu einem Datenstau kommt → Receiver sagt STOP
Sender hört auf Daten zu senden.
Sobald GO empfangen wird werden Daten erneut gesendet. Im GO frame ist evtl noch die Information über das letzte ACK enthalten -N(R).

Zusätzlich wird Windowing verwendet: Receiver generiert keine ACKs mehr sobald es zum Datenstau kommt → sendefenster wandert nicht weiter.

Windowing reicht alleine nicht aus, da durch das stoppen von ACKs Timeouts entstehen, die erneutes Senden von Daten zufolge haben… bzw. die Leitung evtl für tot befinden.

2-22)
Adaptive Windowing wird von TCP verwendet.

Dabei wird zu Beginn der Übertragung eine Start-Größe für das Sendefenster vereinbart.

Je nach Übertragungskanal und Datendurchsatz der app.SW wird diese Windowgröße während der Übertragung dynamisch optimiert…
Meldet receiver Windowgröße = 0 → STOP
Windowgröße >0 → GO

Kapitel 3 TDM Techniques

3-1)

In der heutigen Datenkommunikation sind einzellne Punkt-zu-Punkt Verbindungen weniger häufig. Es treten vielmehr Verbindungen zwischen mehreren Systemen am Standort A zum Standort B auf. Eine separate Leitung für jede dieser Verbindung wäre zu kostspielig – deshalb wurde Multiplexing entwickelt.

Multiplexer können mehrere Input Kanäle durch Interleaving zu einem „high-speed“ Datenstrom auf einer Leitung zusammenfassen und am anderen Ende der Leitung wieder aufsplitten. (mittels Demultiplexer)
Bei Zeitmultiplexing geschieht dies, indem jedem Kanal ein bestimmter Zeitschlitz zugeordnet wird – einzellne Zeitschlitze (den einzellnen Systemen zugeordnet) werden als Frames in einen high-speed Datenstrom umgewandelt.

Die Bandbreite auf der sog. Trunk Line wird zwischen den Kanälen aufgeteilt. Prinzipiell gibt es 2 Arten von Time-Multiplexing:

-Synchronous TDM (jedem Kanal steht ein konstanter timeslot periodisch zur Verfügung
-Asynchronous TDM (die Zeitschlitze werden vom MUX je nach Bedarf an die verschiedenen Kanäle vergeben.. ATDM nutzt die Bandbreite besser aus als STDM)

Skizze3-1

3-2)

Bei SynTDM wird jedem Kanal ein timeslot konstanter Länge im Frame auf der Trunk line periodisch zugewiesen. (z.B. timeslot 1 für Device 1). Diese Zuweisung ist konstant und bleibt dauerhaft bestehen.

Für die Framesynchro wird ein extra Flag dem Frame hinzugefügt.

Vorteile:

Verglichen mit reinen PTP Verbindungen hat SynTDM nur ein kleines zusätzliches delay (zum erstellen der Frames)

Das delay für jedes transportierte Byte ist konstant, die Zeit zwischen 2 (für einen Channel) übertragenen Bytes ist konstant → ideal für synchrone Übertragung

Dadurch vollkommen protocol-transparent (jedes line protocol kann verwendet Werden.
Von den Endsystemen aus sehen die Kanäle wie PTP Verbindungen aus. …

…
Nachteile:

Wird auf einem Kanal überhaupt nichts übertragen, bleibt der timeslot leer (idle) und Bandbreite wird verschwendet

Die Bitrate auf der Trunkline ist sehr hoch – Summe der einzelbitraten+framesynchro-Flag.. daher ist die Methode teuer.

3-3)

Bandbreite auf der Trunk-Leitung = Summe der Bandbreiten der Einzelleitungen + Framesynchro.

Adressierung wird nicht benötigt, da jede Leitung einen fixen timeslot konstanter Länge hat.

Wird innerhalb eines Kanals nichts übertragen, generiert der Multiplexer ein gewisses „idle“ Bitmuster.. der zugeordnete Slot bleibt quasi leer; Bandbreite wird verschwendet.
Da jeder Kanal einen fixen timeslot hat, kann keiner der Kanäle den timeslot eines anderen nutzen.

Bei SynTDM wird **kein** Flow-control zwischen Multiplexer und Device verwendet

3-4)

ATDM nutzt die verfügbare Bandbreite der Trunk-Line komplett aus. Je nach Bedarf werden dem jeweiligen Kanal mehr oder weniger (bis gar keine) timeslots zur Verfügung gestellt. 1 Kanal kann also theoretisch die komplette Bandbreite ausnutzen.

Daher müssen den Frames auf der Trunk-Line adressiert werden – damit der DEMUX erkennt an wen die jeweiligen Daten gehen.

Es entsteht nur Datenverkehr, wenn mind. einer der Inputs Daten zu senden hat. Jeder Kanal kann senden wann er möchte, ohne Bandbreite zu verschwenden.

Dadurch werden die delays für Datenbytes generell variabel (keine idealen Eigenschaften für synchrone Datenübertragung)

Für den Fall, dass die trunk-line gerade voll belegt ist und ein anderer Kanal etwas senden möchte, ist buffering erforderlich. Durch buffering enstehen evtl zusätzliche Delays.

3-5)

Bei ATDM wird die Bandbreite je nach Bedarf der einzelnen Leitungen aufgeteilt – es ist möglich, dass einer Leitung die komplette Bandbreite zur Verfügung steht. Dadurch wird Adressierung der Daten innerhalb des Trunk-Frames erforderlich.

Es sollte Flow-Control (zwischen End-devices.. und zwischen MUX-devices) betrieben werden, damit

a) Allgemein ein Endsystem nicht mehr Daten bekommt als es verarbeiten kann….

b) Ein Kanal nicht ständig die volle Bandbreite bekommt, wenn ein anderer etwas übertragen möchte – und damit es zu keinem buffer overflow innerhalb des MUX kommt und dadurch FCS errors enstehen.

<u>3-6</u>)

Nyquist Theorem:
Jedes analoge Signal mit limitierter BB fb muss mit mind 2*fb abgetastet werden um nach der Übertragung wieder hergestellt werden zu können.

Analoge Sprache im Bereich von 300-3400 Hz wird mittels PCM (Puls Code Modulation) übertragen und im Empfänger wieder hergestellt.

Das Sprachsignal von ca. 4kHz BB wird also ca. mit 8kHz abgetastet. Der Wert des Pulses wird mit 8 Bit kodiert – das ergibt 2^8 = 256 verschiedene Quantisierungsstufen.

Und für den Übertragungskanal : R = 2B*lb(256) = 2*4kHz*8 = 64kbit/s

Ein 8bit Rahmen für G.711 stellt ein Sample des Analogsignales dar. Er muss in einer TDM Hierarchie im konstanten intervall von 125µs (was 8000Hz, der Abtastfrequenz entspricht) übertragen werden, um dem ihm zugeordneten timeslot / Platz in einer übergeordneten Hierarchie zu belegen. Deshalb eignet sich hier SynTDM besonders gut, da der Abstand und das delay zwischen den Samples konstant bleibt.

<u>3-7)</u>

Analoge Signale können unendlich viele Werte annehmen. Digitale Signale bestehen aus endlichen Werten (je nach Anzahl der Bytes). Beim Quantisieren der Samples während der PCM kommt es daher zwangsläufig zu einem Fehler – die Differenz des tatsächlichen Wertes des Signals zur nächstgelegenen Quantisierungsstufe

Skizze 3-7

PCM samples werden logarrithmisch quantisiert (viele Stufen für niedrige Amplituden, wenige Stufen für hohe Amplituden) um das SNR des Signals zu verbessern (bei Störungen/Noise würden sonst die niederen Amplituden am ehesten verloren gehen – durch log. Quant. eben nicht)
µ-Law in USA
A-law in EU

Skizze 3-7 für PCM Sample (1 Byte)

3-8)

ADPCM ist ein Waveform Coder. (Adaptive differential PCM) –es wird oft nur die Differenz zwischen 2 Samples (PCM bytes) übertragen, was Bandbreite einspart, da weniger Bits benötigt werden.
Bei Waveform Coder wird eine analoge Waveform nicht-linear angenähert.

Vocoders

Sprache analysiert und mit einem Codebuch verglichen
Nur Codebuch Werte werden übertragen –der Empfänger reproduziert die Sprache mittels Synthesizer wieder

3-9)

Standardisierte Codec's:

PCM – G.711 (64kbps)
ADPCM – G.726 (16,24,32,40 kbps)
LD-CELP (low delay code excited linear predictor) – G.728 (16kbps)
CS-ACELP (conjugate structure algebraic code exited linar predictor) – G.729 (8kbps)

Dual Rate Speech Coding Standard – G.723 (5,3kbps – 6,3kbps)
Ist der standard für Sprachübertragun in IP Netzwerken … verwendet als Basis die CELP Technik von GSM.

3-10)

PDH (= plesiosynchronous digital hierarchy)

Plesio bedeutet nahezu. Jeder PDH Multiplexer gibt seinen eigenen Takt vor. Synchrone Übertragungszeitdifferenzen werden mit Bitstuffing ausgeglichen. Taktung erfolgt mittels Bitstuffing

Limitierung:

Kann für höhere Geschwindigkeiten nicht genutzt werden, da der Overhead dabei schnell ansteigt. Dieser overhead ist abhängig von den Bitraten der Übertragung. Er wird bedingt durch das notwendige Bitstuffing

Add/Drop Multiplexing ist wegen der unterschiedlichen Länge (je nachdem wieviele Bits gestopft wurden) unmöglich.

3-11)
SDH = synchronous digital hierarchy

Taktunk erfolgt prinzipiell durch Rückgewinnung des Taktes aus dem empfangenen Signal und Verwendung dieses rückgewonnenen Taktes als eigener Sendetakt auf der nächsten Teilstrecke.

Vorteile:
- Kleinerer Overhead
- Keine Stopfbits für Taktunterschiede
- Add/Drop Multiplexing möglich (als Vorraussetzung für SDH)

Kapitel 4 Network Principles

4-1)
Circuit Switching bruht auf SynTDM. Lokale Endsysteme sind mit einem TDM Switch verbunden, welcher wiederum mit anderen TDM Switches verbunden ist – auf der Leitung wird SynTDM betrieben, womit Circuit Switching die gleichen Basiseigenschaften hat:

-kleine, fixe Delayzeit
-hohe Bitraten auf der „Trunkline“ (Einzelbitraten addieren sich)
-wenn keine Daten gesendet werden → „idle“ Bitpattern (Bandbreite verschwendet)
-dafür komplett Protokoll transparent

Alle verschiedenen „Pfade“ für die Signale sind – je nach Quelle/Senke in sog. Circuit Switching Tabellen in den TDM Switches gespeichert.
Jeder Kanal ist also einem Port auf dem TDM Switch bzw. ein timeslot auf der Trunk-Line (zwischen den Switches) zugeordnet.

Skizze 4-1

4-2)
Die Circuit Switch Tabelle ist in den TDM-Switches gespeichert und gibt dabei an, **welchem Port** (am Switch) **welche logische Verbindung** (und damit auch **welcher timeslot**) zugeordnet ist.

Switches, die die Verbindungen in den Tabellen für einen **eingehenden trunk-timeslot entweder** auf ein lokales Endsystem **oder** eine weitere trunk Leitung „mappen“ (festlegen) werden als „transit switches“ bezeichnet.

Von den Endsystemen zum TDM-Switch hin (Access Ports) kann ebenfalls mit einer Trunk-Line TDM betrieben werden – was vor allem Leitungen einspart. Der TDM Switch selbst legt die logischen Verbindungen dann anhand der eingehenden und ausgehenden Timeslots fest.
Skizze 4-2

4-3)
Beim „permanent circuit service“ sind die Einträge für die logischen Verbindungen und Timeslots in den Switching Tabellen statisch und nur von Hand eingegeben (Aufgabe des Netzwerk Administrators).

Dieses System wird durch ein TDM Network Management Protokoll verbessert, welches neue „Pfade“ **automatisch** festlegt – sollte eine Trunk Leitung ausfallen („soft permanent circuit service“)

Dies wird von Providern oft als **digital leased line** bezeichnet. (angewandt in PDH oder SDH Netzwerken)

4-4)
Beim „switched circuit service“ erfolgt das Festlegen des Übertragungspfades komplett dynamisch:

Der Sender verwendet ein signalling prot. an den TDM Switch um eine Verbindung aufzubauen/zu beenden. Der TDM Switch legt daraufhin automatisch einen geeigneten Pfad/Timeslot für **diese** Verbindung fest. Nach Verbindungsabbruch ist der Pfad wieder gelöscht.

ISDN verwendet die switched circuit technologie

4-5)
ISDN – Integrated Services Digital Network ist eine Übertragungstechnologie für den Pfad

Endanwender → Netzwerk

Dabei ist das sog. (U)ser to (N)etwork (I)nterface definiert. Es gibt 2 verschiedene Arten: Basic Rate Interface und Primary Rate Interface.

BRI verwendet nur 2 Bearer (Nutzdaten) mit 64kbit/s und einen Daten („Controll-Daten“) Kanal mit 16kbit/s, die mit TDM auf eine Leitung multiplexed werden (+overhead)

PRI hat hingegen 30 Bearer Kanäle zu je 64kbit/s und einen Daten Kanal mit 64kbit/s – dabei werden ebenfalls alle Kanäle auf eine Leitung TDM multiplexed.

Der D-Channel dient bei beiden vor allem dazu um Verbindungen aufzubauen und wieder zu beenden.
PRI verwendet PDH Framing „E1“

4-6)
Packet Switching ist ein Vermittlungsverfahren das auf statistical/asynch. TDM beruht.

-Alle Verbindungen für „End-Device → Packet Switch“ sowie „Packet-Switch→Packet-Switch“ werden mit STDM betrieben

-So kann die Bandbreite viel effizienter ausgenutzt werden, das komplette System wird dynamischer, Leitungen werden eingespart.

-Allerdings muss jedes übertragene Datenframe nun Addressierungsinformationen enthalten, damit der jeweilige Switch die Daten richtig weiterleitet

-Deshalb ist das System nicht Protokoll Transparent… (Protokoll zwischen Endsystem und PS wird wg. Adressierung und evtl. Flow Control benötigt)

-Buffering evtl erforderlich

4-7)
Der komplette Vorgang für die Übertragung von Packeten wird als „store-and-forward“ bezeichnet.
-Der Sender splittet die Daten in kleinere Datenpackete auf, wobei jedes eine Source und Destination Adresse enthält

-Der PS empfängt die Pakete (evtl mit Buffer), liest die Adressinformation und entscheidet dann, an welchen PS er sie weiterleitet (evtl mit Buffer bzw. Queues)

-Der darauffolgende PS verfährt wie der vorige, bis das Datenpacket an der Destination angelangt ist.

4-8)
Packet Switches entscheiden anhand von Routing-Tabellen wohin Datenpakete weitergeleitet werden. Jedes Datenpaket benötigt daher eine Adressinformation. Der Prozess des „auswählen des Weges für das Datenpaket“ wird als **routing** bezeichnet.

Es wird zwischen Static Routing und Dynamic Routing unterschieden:

Static Routing:
-Tabellen vom Netzwerkadministrator vorkonfiguriert
-Einträge bleiben über lange Zeit fix (statisch) , kann u.U. zu Problemen führen, wenn Netzwerk erweitert wird, Leitungen ausfallen usw..

Dynamic Routing:
-Tabellen werden mittels eines Routing-Protocolls festgelegt, welches die Netzwerk-Topologie „scannt“ und somit alle Wege auswertet und sinnvolle Pfade speichert.
-besser als static routing.

4-9)
Die Weiterleitung (**forwarding**) der PS basiert auf Tabellen.. diese können f. connection oriented Service **switching tables** oder f. connection less Service **routing tables** sein.

CO- Routing Tabellen werden nur verwendet um Switching Tabellen zu unterstützen – und das auch nur beim Verbindungsaufbau.
Nach dem Verb.aufbau werden weiterhin nur switching Tabellen verwendet. **VIRTUAL CALL SERVICE**

CL – Routing Tabellen werden für Aufbau und forwarding für jedes Packet verwendet. **DATAGRAMM Service.**

4-10)
ROUTING-Prot : Protocoll für Dynamic Routing als Kommunikation zwischen den PS… legt die Netzwerkpfade (Einträge in den PS Tabellen) fest.

ROUTABLE-Prot: Verwendet eine **einmalige** und **strukturierte** Adresse um „Routing" im Netzwerk überhaupt zu ermöglichen.

4-11)
Routingprotokolle beinhalten einmalige, strukturierte Adresse – machen Routing überhaupt erst möglich.

Routingtabellen beinhalten die „Pfade" die die Datenpakete nehmen. (Dest. Adress, Incomming Line, Outgoing Line, Next PS).

Sie kommen entweder statisch (vom Admin eingestellt)…
Oder dynamisch durch routingprotokolle zustande.

4-12)
Switching Tabellen werden durch Routingtabellen beim Verbindungsaufbau eines Connection-Oriented Packet Switch Service am Anfang einer Verbindung festgelegt und anschließend (fix) verwendet…. Dh. ein fixer Pfad

4-13)
Beim (conn.less) Datagramm Service eines PS Systems. Werden Datenpakete ohne explizitem Verbindungsaufbau in das Netzwerk gesendet.

-Jedes Datenpaket enthält die komplette Adressinformation
-Datenpakete werden alle einzelln von den jeweiligen PSwitches behandelt (adress Info bleibt gleich im Paket)
-Je nach Zustand der Routing Tabellen werden die Pakete weitergeleitet
-Dadurch können Paketein ungeordneter Reihenfolge am Empfänger ankommen (verschiedene Pfade) …

…
-Pakete können gelöscht werden, wenn sie zu lange im Netzwerk „herumschwirren“ (kill-Mechanismus)
-Pakete können gelöscht werden, wenn transmission errors auftreten
→ Protokoll am End-System muss Pakete ordnen und Fehlerkorrektur durchführen (evtl. Conn.oriented)
-best effort service … Geschwindigkeit hängt nur von den Ressourcen des Netzwerks ab.

IP beruht auf PS - Datagramm Service.

4-14)
Die Änderung des Netzwerkes wird durch das Routingprotokoll erkannt und die Routing Tabellen werden entsprechend angeglichen. → alternative Pfade werden etabliert

Es kann jedoch passieren, dass während des Ausfalls Daten verloren gegangen sind.

4-15)
PS-Datagramm Service:

Vorteile:
-Wenig Ressourcen zur Implementierung für die PS und Endsysteme benötigt. (da kleiner Prot. Overhead)
-Daten werden schnellstmöglich übertragen

Nachteile:
-Fehlerkorrektur, Datenumordnen ist nicht gewährleistet und muss von einem anderen Prot. im Endsystem sichergestellt werden.
-flow control nicht möglich

4-16)
Es kann passieren, dass Pakete durch fehlerhafte Routing-Tabellen endlos zwischen den PS „herumschwirren“ und somit Buffermemory/Bandbreite verschwenden. Deshalb wird ein Kill Mechanismus für solche Frames beim Datagramm PS Service verwendet.

z.B. werden die Pakete nach einer best. Anzahl von Übertragungen zwischen PS verworfen (Hop-count). Es ist aber auch ein Mechanismus mit „max-time-to-live“ möglich.

IP verwendet den Hop-Count Mechanismus:
-Beim Senden eines Packets wird eine Variable auf einen gewissen Wert gesetzt, der pro PS um 1 verringert wird.
-Erreicht die Variable den Wert 0 , wird das Paket verworfen.

4-17)
Ablauf einer Datenübertragung bei (CO) Virtual Call Service Paket Switching:

-Spezielle Datenpakete werden zu Beginn der Verbindung verwendet um eine fixe virtuelle Punkt-zu-Punkt Verbindung über das Netzwerk zwischen den beiden Endsystemen aufzubauen. (call setup)
-ist diese Verbindung etabliert, werden Datenpakete ausschließlich über diesen „vorgegebenen Weg“ übertragen
-nachdem alle Daten übertragen wurden wird die virtuelle PTP Verbindung wieder geschlossen.

Die Datenpakete ansich benötigen dadurch nicht mehr die komplette Adressinformation, sondern lediglich den sog. „local connectionidentifier“ , welcher angibt von welchem device (zb. PS2) das Paket kommt und wohin (z.B. PS5) es gesendet wird… ist das Paket bei PS5 angekommen und wird zb. Zu PS7 gesendet ändert sich der local connection identifier erneut. **Die Adressinformation des Pakets ändert sich also von Device zu Device**

Bekannte Technologien: X.25, Frame Relay, ATM

4-18)
Ablauf eines Verbindungsaufbaus bei (CO) Virtual Call Sercie Paket Switching:

-ein call-setup Paket wird mit kompletter Adressinformation (zb. Device A→B) sowie einem „local conn. Identifier“ Feld als Datagramm gesendet.
-Es läuft daher über die eingestellten routing-tabellen bis zur Destination
-Dabei wird in den dafür verwendeten PS in der **seperaten** Switching Tabelle ein Eintrag für den Verlauf der connection angelegt. Der Connection identifier taucht dabei im „from“ Feld auf und wird vom Switch auf einen neuen identifier (zum jeweiligen Port) im „to“ Feld verknüpft.
-die Verbindung ist also in der Switching Tabelle für die Übertragung fix gespeichert. (**Virtual Circuit**)

Skizze 4-18

SVC: Switched Virtual Circuit – wie oben beschrieben … jede Verbindung wird durch call-setup aufgebaut und mit call-termination wieder beendet/gelöscht.
Ist etwas langsamer, dafür dynamisch

PVC:Permanent Virtual Circuit – Virtuelle Verbindungen sind im Netzwerk die ganze Zeit über fix vorgegeben… es ist also kein setup/termination erforderlich.
Ist dafür schneller aber statisch.

4-19)
Beim Ausfallen der Leitung einer virtuellen Verbindung bricht diese ab und muss neu aufgebaut werden (call setup). Dabei werden die Routing-Tabellen bei Ausfall einer Leitung evtl neu auf eine redundante Leitung festgelegt (dynamisches Routing). Der Verlauf der neu etablierten Virtual Circuit ist damit auch anders als der alte.

Die dazwischen Übertragenen Daten gehen dabei in der Regel verloren.

4-20)
Vorteile von (CO) Virtual Call Service Packet Switching:

-Ressourcen können im vorhinein reserviert werden, QoS kann dadurch angeboten werden.
-für das jeweilige End-System sieht das Netzwerk dadurch wie eine PTP Verbindung aus .. die höheren Protokolle können sich auf eine implementierte Error Recovery, Flow Control und geordnete Daten verlassen.
-die Bereitschaft des Empfängers wird bereits beim Verbindungsaufbau überprüft

Nachteile:

-call-setup benötigt einige Zeit
-Protokoll mit Virtual Call Service stellt höhere Hardwareanforderungen an die Endsysteme und Packet Switches, da es viel komplexer als Datagram Service ist.

4-21)
Grundidee von Open Systems Interconnection Reference Model:

7 „Layer“ – jeder Layer erhält bei der Datenkomm. eine bestimmte Aufgabe.. komplexe Prozesse werden so in kleinere überschaubare Aufgaben unterteilt. Der gesamte Prozess wird **hierarchisch** organisiert jeder Layer benötigt Daten von dem Layer darunter und liefert Daten an den Layer darüber. (Services)

Bei den Endsystemen kommunizieren damit alle Layer auf einer logischen Ebene miteinander. (Layer – spezifische Protokolle, protocoll specification standards)

Skizze 4-21

OPEN bei OSI bedeutet, dass dieses System nicht Herstellerabhängig gedacht ist, sondern übergreifend für alle Netzwerkhersteller gelten soll, damit quasi „alles mit allem“ kompatibel gemacht wird. Weiters ist die jeweilige **Realisierung** der 7 Layer in den **Endsystemen** nicht standardisiert, sondern lediglich **das Verhalten der Systeme untereinander**
Das OSI Modell kann also lediglich nur als Anhalt für einen Rechner-Internen Netzwerk Ablauf dienen.

4-22)
Bei OSI kommunizieren alle 7 Layer auf einer logsichen Ebene miteinander.

Dadurch werden mehrere Protokolle für einen Datenstrom verwendet.

Der Datenstrom hat daher einen Overhead für jedes Protokoll… Er wird dadurch u.U. sehr stark „aufgeblasen“ (großer Datenstrom, wenig Nutzdaten). **Encapsulation**

Beim Empfänger wird der Nutzdatenstrom Schicht für Schicht wieder extrahiert **Decapsulation**

Die Vorteile ergeben sich durch die allg. Vorteile des OSI Modells:

-standardisiert – „jeder ist mit jedem kompatibel“
-Aufgaben in Layer aufgespalten – erleichtert die Entwicklung von Netzwerk-Geräten
-Fehlerkontrolle/Korrektur schon auf den unteren Layern möglich (obere Schichten werden somit gar nicht erst „belastet“

Ein Intermediate System ist bei OSI Ein Netzwerksystem, welches lediglich die Layer 1-3 (also jene um Daten zu transportieren) enthält und wird dann verwendet, wenn 2 Endsysteme nicht direkt miteinander verbunden sind. Es enthält Paket Switches, muss routing/switching und store&forward beherrschen.

4-23)
Repeater – L1
SDH-Switch- L1
Bridge – L2
Ethernet Switch –L2
IP Router - L3
ISDN Switch - (L1 verbindung steht) , L3/7 (Signalisierung)
X.25 Switch – L3
Frame Relay Switch – L2
ATM Switch – L2

Für X.25, Frame relay und ATM Switch wird die Verbindung erst auf L3 aufgebaut, danach auf L2 gearbeitet.

Kapitel 5 LANs and Legacy Ethernet

5-1)

-Multipoint line (Mehrfachverbindung) **shared Media** , jede station ist gleichberechtigt, jeder kann mit jedem reden.

→ daher Access controll (Zugriffskontrolle) Mechanismus benötigt, wird durch MAC geregelt.

→ Addressierung notwendig – MAC Adressen (Adressierung kann an 1 Device oder eine Gruppe gehen.) Adressen sind unstrukturiert

-Broadcast behaveiour… gesendete Daten können prinzipiell von jedem empfangen werden.

- Layer 1+2 des OSI Models ist ausreichend für LAN

→ kein Packet Switching (kein L3),

→kein Layer 4, da „Transport“ auch als CO Protokoll in Schicht 2 realisiert werden kann.

5-2)

Für die Kommunikation innerhalb eines „alten“ LANs genügt L1 und L2… L3 ist obsolet da es kein PS gibt und „Transport“ als Layer 4 kann als CO Protokoll in L2 realisiert werden.

Der OSI L2 ist ursprünglich für PTP Verbindungen gedacht—LAN ist jedoch eine multipoint Verbindung mit shared Media.. deshalb wurde der OSI L2 beim IEEE 802 Standard in LLC und MAC aufgespalten.

802.3 spezifiziert z.B L1 und MAC von L2 – also die „Hälfte“ von L2

5-3)

LLC (logical link control) Layer erfüllt die ursprünglichen Aufgaben des OSI L2

-CL/CO Service

-Service Access Points (SAPs) **Felder** für höhere Protokolle

→ DSAP- Destination SAP

→ SSAP – Source SAP geben an, für welche Protokolle die jeweiligen Frames sind.

MAC Layer

-framing, adressing (MAC Adresse), error detection, Prioritäten

-Zugriffskontrolle

Netzwerkkarten, die Frames mit Destination Adresses Empfangen, die nicht gleich ihrer eigenen MAC Adresse sind, verwerfen diese automatisch.

5-4)
Jede Netzwerkkarte besitzt eine einmalige MAC Adresse, als sog. BIA – welche sie beim Senden jedes mal als „Source Adresse“ angibt.

BIA steht für „Burn in Adress“ – solche Adressen werden Netzwerkkarten bei ihrer Herstellung eingespeichert und sollten *eigentlich* nicht manipulierbar sein (um Netzwerkkarten genau festlegen/orten/adressieren zu können). De facto kann man die BIA Adresse einer NIC (Network Interface Card) jedoch ändern.

Eine MAC Adresse besteht aus 6 Byte (48bit). Sie tritt als SA(Source Adr) und DA (Destination Adr) auf.

Skizze 5-4

5-5)
Für einen Broadcast ist die komplette MAC Adresse auf ‚1‘ gesetzt.

Multicast heißt Broadcast für eine best. Gruppe von NICs - dabei ist das IG Feld auf 1 gesetzt (in den jeweiligen NICs muss eine Multicast adr. Definiert sein, sonst werden die Frames verworfen)

Mit L2 Multicast lassen sich Endsysteme gezielter ansprechen und ein „mithören“ dritter ist somit weniger wahrscheinlich als bei einem „echten“ Broadcast.

5-6)
Für die Destination MAC Adresse gibt es im Prinzip 3 Möglichkeiten…

Broadcast:
-Alle Bits der MAC Adresse werden auf ‚1‘ gesetzt, alle Stationen empfangen den Broadcast und leiten diesen an die höheren Layer in ihren Systemen weiter.

Multicast (Gruppen):
-Das I/G Bitfeld der MAC Adresse ist auf 1 gesetzt,
-das U/L Feld kann auf 1/0 sein, je nachdem ob es sich um lokal vom Admin eingestellt Adressen oder global von IEEE verwaltete Adressen handelt.
-in den jeweiligen Stationen für die Gruppen muss eine Multicast Adresse definiert sein.
-stimmt die Destination MAC Adresse mit der Multicast Adresse der jeweiligen Station überein, wird der Rahmen an die höheren Layer 3-7 weitergeleitet.. falls nicht wird er verworfen

Einzeladressierung:
-I/G Bitfeld auf 0,
-U/L Bitfeld auf 1 oder 0 - siehe oben
-Rahmen wird nur weitergeleitet, wenn Destination MAC Adresse = BIA der jeweiligen NIC der Station. …

…
Ein Broadcast bzw. Multicast sorgt für sog. Interrupts in allen/gruppenweisen Stationen… Broadcasts sollten für eine bessere Performance daher nur bei der Initialisierung des Netzwerks bzw. nur selten auftreten. Wird ein Rahmen empfangen, dessen Dest. MAC Adresse ungleich der BIA der NIC ist, wird er „still“ verworfen, ohne die höheren Layer zu unterrichten.

5-7)
Zu Beginn der Ethernet Zeit gab es 2 Standards für 10Mbit/s : 10Base5 und 10Base2

-Beide waren für eine **BUS-Topologie** mit **passiver Kopplung** ausgelegt , **shared Media**
-Es erfolgte dabei **Baseband Transmission, Manchester encoding (kein DC)**
-Der Betrieb war nur im half duplex Mode möglich.. „nur 1 bestimmter Sender zu einem Zeitpunkt“
-Reichweite : 2500-3000m (Max Abstand zwischen 2 Workstations)
-Um Reflektionen vorzubeugen, waren die Netzwerkleitungen an ihren Enden mit dem **Wellenwiderstand** abgeschlossen.

5-8)
CSMA/CD = Carrier Sense Multiple Access / Colision Detection

Beschreibt den Half-Duplex Mode bei älteren Ethernet Standards –„nur 1 Sender zu einem gewissen Zeitpunkt“

CS: Carrier Sense – Die einzellnen Stationen erkennen „Aktivität“ am shared Media und warten bis die Leitung frei ist um selbst ihre Daten übertragen zu können.
MA: Multiple Acess – shared Media, alle Workstations haben Zugriff auf das Medium und sind gleichberechtigt. Es kann allerdings öfter zu Kollisionen am Medium kommen (2 Sender starten gleichzeitig mit Übertragung)

CD: Collision Detection – Workstations sind dazu in der Lage eine oben beschriebene Kollision zu erkennen (das DC Level wird dabei überwacht)… Signale normalerweise DC-frei wegen Manchester Code

Wenn eine Kollision entdeckt wurde:
-alle Stationen, die die Kollision entdecken brechen die Übertragung ab
-ein 32 bit JAM Signal wird gesendet.. damit jede Station über die Kollision bescheid weiß
-Nach Abbruch der Übertragung startet in jeder Station ein Zufallsgenerator, der somit ein variables Timeout (für jede Station verschieden) bestimmt – „exponential backoff algorithm“
-nach dem Timeout versucht die Station erneut ihre Daten zu übertragen
-scheitert die Übertragung wieder – wird eine Variable erhöht… wenn eine Üb. 16 mal in Folge scheitert, wird der Rahmen verworfen und dem höheren Layer bescheidgegeben.

5-9)

Der Truncated Exponential Backoff algorithm ist ein Prozess, der versucht, Kollisionen am Ethernet möglichst zu dezimieren.

-Nachdem in einem Device eine Kollision erkannt wurde startet ein Zufallsgenerator.
-Werte : 0<=random < (2^k)
-dieser Wert wird mit der „slot time“ (=collision Window) des jeweiligen Ethernet Systems multipliziert.. und damit ein für quasi jede Station ein variables Delay erzeugt, bis diese neu sendet.. also delay=slot time * random… Die Slot time ist für verschiedene Ethernet Standards verschieden – sie wird bestimmt durch die doppelte max.signal propagation time.. für 10Base2/5 beträgt sie 51,2µs.
-der Wert k ist 10 – es können also höchstens 1024 verschiedene delay times auftreten, womit sich höchstens 1024 Einzelstationen für ein Ethernet Netzwerk mit 802.3 Standard ergeben.
- Durch das Zufallsverhalten, kann nicht vorrausgesagt werden, welcher Sender wann seine Daten überträgt → Ethernet ist daher **nicht deterministisch**

5-10)

Die Slot Time gibt den Zeitbereich an, in welchem eine Kollision überhaupt auftreten kann.

Sendet Station A Daten, benötigt dies tp (signal propagation delay) Zeit um zu Station B zu gelangen.
Sendet aber nun Station B auch Daten (etwas später als A), tritt eine Kollision auf.
Im worst-case erfährt A erst nach 2*tp von dieser Kollision.
2*tp ist also die Zeit, in welcher eine Kollision auftreten kann → Slot time

Die Stationen müssen also für den worst-case die slot time (2*tp) warten um eine Kollision sicher zu erkennen.

51,2µs slot time bei 10Mbit/s Bitrate entspricht 512 bits, was wiederum 64 byte entspricht

5-11)

Eine Festlegung für die maximale Rahmengröße wird aufgrund von „Fairness“ getroffen.. kein Sender sollte „unendlich“ lange Rahmen aussenden und somit das shared Media blockieren.

Das kontinuierliche Senden von Rahmen die größer als die max. frame size sind wird als „jabber“ (brabbeln) bezeichnet.

Um das zu verhindern, gibt es in jedem Transceiver eine Jabber-Control Schaltung.

5-12)
Bestandteile eines Ethernet Controllers:

Skizze 5-12

LLC – Logical Link Control - CL/CO Service
MAC – Media Access Control – Adressierung/Zugriffskontrolle
PLS – Physical Layer Signaling – Signalkodierung/dekodierung mittels Manchester

AUI – Attachment Unit Interface – Erlaubt einen getrennten Transceiver/End System

PMA- Physical Medium Attachment -- Transceiver mit Jabber Contr., Collision Detection
MDI – Medium Dependent Interface – siehe oben

5-13)

Die Manchester Kodierung bei PLS und AUI war ineffizient

5-14)
-Repeater = Verstärker zur Vergrößerung von Ethernet LAN Netzwerken
Signal am Eingang („1.LAN Segment“) wird mit minimaler Verzögerung verstärkt und am Ausgang („2. LAN Segment“) weitergegeben.

Diese Minimale Verzögerungszeit muss im Bezug auf die Slot-Time berücksichtigt werde → Repeater Rules.. Durch die fixe Slot Time lässt sich ein Ethernet mit Repeatern **daher nicht beliebig vergrößern**

Mit den Repeatern tritt erstmalig das „Link Segment“ auf – als (ursprünglich) PTP Verbindung (Fiber/Copper) zwischen 2 Repeatern.

5-15)
Im Laufe der Zeit wurde die Netzwerkverkabelung in Gebäuden gesetzlich definiert → Structured cabling
Die Festlegung entsprach einer STAR Topologie:
-Endsysteme an „1. HUB“ (Stockwerkverteiler-REPEATER) – haupts. copper
-1. HUBs an „HauptHUB“ (Gebäudeverteiler-REPEATER) – fiber
-evtl Gebäudeverteiler untereinander – fiber

Ethernet war aber ursprünglich als Bus-Topologie ausgelegt… Um dem Gesetz zu entsprechen wurden Link Segments nicht nur für Repeater-Repeater sondern auch für End-System-Multiport Repeater definiert → 10BaseT mit Unshielded Twisted Pair Kabeln, Half Duplex(…Ethernet..)

…

…
<u>Mögliche Verbindungen mit 10BaseT:</u>
Repeater-Repeater
End System – Multiport Repeater
End System – End System (Crossover Kabel)+

Skizze 5-15

<u>5-16)</u>
10BaseT als Link Segment Standard für End-System→ Multiport Rep. Veränderte die Topologie des Ethernet von BUS auf STAR– wegen gesetzlicher Verordnung des structured cabling wäre das Ethernet sonst ausgestorben.

Das HUB(Repeater) simuliert den BUS – CSMA/CD in a box

Die Multiport Repeater werden dabei als HUB bezeichnet, was mit Vorsicht zu genießen ist, da „HUB“ auch der Name für einen L2 Ethernet Switch ist.

<u>5-17)</u>
Es gibt 3 unterschiedliche Rahmentypen:

LLC (802.3)
Ethernet V2
Ethernet mit SNAP (Subnetwork Access Protocoll)

Aufbau → *Skizze 5-17*

Ethernet V2 ist das gebräuchlichste Format.

<u>5-18)</u>
Ethernet V2 unterstützt kein CO Service auf L2, weil es gar kein LLC gibt.

LLC SAP Felder mit 8bit können keine Ethernet V2 Protocol Type (16bit) enthalten, deshalb wird ein Anpassungsschicht eingeführt

➔ SNAP (Subnetwork Access Prot) ist das Protokoll, das die Übertragung von Ethernet V2 Daten über IEEE LANs ermöglicht

Skizze 5-18 - siehe Skizze 5-17 unten

Kapitel 6 Paket Switching on L2 (LAN Level)

6-1)

LAN war ursprünglich nur als shared Media Technologie designed – dies führt jedoch bei einer höheren Anzahl von Endsystemen zu Performance Problemen (collisions treten öfter auf)

Deshalb ist es sinnvoll das Netzwerk INTERN mit Bridges zu unterteilen :
-die „Last“ an Datenverkehr wird sinnvoll aufgeteilt
-höhere Sicherheit
-weniger Kollisionen
-Netzwerk wird leichter erweiterbar
-LANS können so leichter untereinander kommunizieren (remote networking über WAN)
→ deshalb TRANSPARENT BRIDGING

Arbeitsweise:
-Bridge auf L2 (MAC), --- LLC, L3-7 sind somit protocol transparent
→Dieser verwendet unstrukturierte MAC Adressen und „merkt“ sich in bridging tables welches Endsystem an welchem seiner Ports „hängt“ (Source Adresse im Frame wird dafür verwendet)

→Bridge entscheidet anhand seiner bridgint tables ob Adressen weitergeleitet oder gefiltert werden (Dest. Adresse im Frame)

--filtering: DA des Frames wird als ES der selben collision domain erkannt… frame wird nicht an den Ausgangsports weitergegeben

--forwarding: DA des Frames ist nicht in der selben collision domain.. frame wird an die Ausgangsport(s) weitergegeben.

--flooding: für Bridges mit selbst konfigurierbaren bridging tables- Frames mit unbekannter DA werden an alle anderen Ports weitergegeben, SA wird „gemerkt“

--Broad- und Multicasts werden an alle Ports weitergegeben.

-Bridging table kann statisch (vom Admin eingestellt) oder dynamisch (self-learning-mechanism) sein

Die Arbeitsweise ist vergleichbar mit Packet Switching.

6-2)

In der Bridging Tabelle ist jedem Port eine Auflistung seiner Endsysteme zugeordnet. Hängen z.B. die Endsysteme A,B,C der collision domain 1 an Port 1 und die Systeme D,E der coll. Domain 2 an Port 2 sieht die Tabelle folgendermaßen aus
…

Port 1	Port 2
A	D
B	E
C	

In dynamischen Systemen „lernen“ die Bridges automatisch, welche Endsysteme an welchen Ports hängen – für die Einträge werden immer die MAC Source Adressen verwendet.

Wenn Endsysteme ihre Standorte ändern (z.B. ES A von coll. Domain 1 → 2) ändert das auch die Bridging Table einträge des Bridges entsprechend

Port 1	Port 2
B	D
C	E
	A

Dies muss in einem „aging mechanism“ berücksichtigt werden- der Bridge, muss also immer wieder überprüfen welche Endsysteme an welche seiner Ports angeschlossen sind.
Ein gängiges Verfahren ist es auch Einträge aus der Bridging Tabelle zu löschen, falls eine SA in den Frames für eine gewisse Zeit (z.B. 5 min) gar nicht mehr am Bridge auftaucht.
„Steckt“ man also sein Endsystem vor dem Zyklus ab, verschwindet der Eintrag in den Bridging Tabellen und die Bridges müssen den neuen Standort des ES später lernen.

<u>6-3)</u>
Siehe 6-1
Arbeitsweise:
-Bridge auf L2 (MAC), --- LLC, L3-7 sind somit protocol transparent
→Dieser verwendet unstrukturierte MAC Adressen und „merkt“ sich in bridging tables welches Endsystem an welchem seiner Ports „hängt“ (Source Adresse im Frame wird dafür verwendet)

→Bridge entscheidet anhand seiner bridgint tables ob Adressen weitergeleitet oder gefiltert werden (Dest. Adresse im Frame)

--filtering: DA des Frames wird als ES der selben collision domain erkannt… frame wird nicht an den Ausgangsports weitergegeben

--forwarding: DA des Frames ist nicht in der selben collision domain.. frame wird an die Ausgangsport(s) weitergegeben.

--flooding: für Bridges mit selbst konfigurierbaren bridging tables- Frames mit unbekannter DA werden an alle anderen Ports weitergegeben, SA wird „gemerkt“

--Broad- und Multicasts werden an alle Ports weitergegeben.

-Bridging table kann statisch (vom Admin eingestellt) oder dynamisch (self-learning-mechanism) sein

…Der Bridge ist für die Endsysteme nicht sichtbar – für die Endsysteme wirken alle collision domains wie ein großes LAN

Ein Bridge muss stehts alle Frames seiner Collision domain empfangen um evtl Frames / Broadcasts und Multicasts weiterzuleiten – damit das ganze Netzwerk wie ein LAN wirkt

6-4)
(Bild Folie 20)
Frames mit Multicast/Broadcast Adressen werden von den Bridges an alle Ports weitergeleitet (alle Bridges flodden).

Remote Bridging: 2 collision domains über 2 Bridges über einen WAN Link (viel niedrigere Bitrate als LAN Link) gekoppelt.

Die lokale Bitrate stimmt mit der WAN-Link Bitrate nicht überein → WAN Link wirkt wie ein „Flaschenhals“ → Endsysteme verschiedener collision domains können nur schlecht miteinander kommunizieren.
Die hohe Anzahl an Multicasts/Broadcasts in einem LAN schränkt den Datenverkehr noch mehr ein.

6-5)
Ein Ethernet Switch ist im Prinzip ein Multiport Bridge (Bridge mit mehr als 2 Ports)

Ethernet Switching bedeutet schnelles Transparent Bridging. Es wird nur die L2 MAC Adresse verwendet, der Ethernet switch hält die Switchingtabelle auf aktuellem Stand (self-learning und aging mechanism) – Jedem Port in der Tabelle ist eine Auflistung seiner angeschlossenen Endsysteme zugeordnet… Pakete werden anhand der DA im Frame weitergeleitet.

Diese Switching (Bridging) Tabelle ist vergleichbar mit der Routingtabelle im CL Mode. Die Switches treffen die Entscheidung ob und wohin Pakete / Frames weitergeleitet werden.. es gibt aber nicht ein connection Setup mit local con. Identifiers (Virtual Call service, CO Mode).

6-6)
IP Routing arbeitet auf L3 mit strukturierten IP Adressen

In IP Routingtabellen steht der nächste Hop nächster Router.

Ein IP-Router empfängt nur Pakete, die an seine MAC Adresse(n) gehen.

IP Router ist für die Endgeräte sichtbar.

6-7)

Ein Ethernet LAN mit transparent Bridging ist durch die verschiedenen Bridges mit ihrer Filterfunktion in mehrere collision domains aufgebaut.. **Collisions bleiben innerhalb der Domain** – nach außen wirkt das Netzwerk aber wie ein zusammenhängendes LAN (eine Broadcast Domain).

Ein Ethernet LAN mit Repeatern hat **keine verschiedenen Collision Domains bzw. immer nur eine Collision Domain = Broadcast Domain**.. die Repeater haben keine Filterfunktion, sondern verstärken lediglich nur die empfangenen Signale.

6-8)

Treten „Parallelschaltungen" von Bridges im Netzwerk auf wird dies als „parallel Path" bezeichnet. Durch die Flooding Funktion der Bridges treten so Probleme der Kategorie „endless circling" und „Table rewriting" auf…

-Broadcast/Multicast Frames

-Frames mit unbekannter DA

Werden ununterbrochen von den Bridges gegenseitig weitergeleitet und verbrauchen somit unnötig Buffer-Platz im Speicher.

Diesem Problem wird mit dem STP (Spanning Tree Protocol) entgegengewirkt – stellt sicher, dass nur EIN aktiver Weg zwischen 2 Endsystemen auftritt, Bridges kommunizieren untereinander.

6-9)

Broadcast Storms ist ein Problemfall, der bei Parallel Paths von Bridges / Switches im Ethernet LANs auftritt („Parallelschaltungen von Bridges/Switches")

2 Arten

(Bild Folie 23,24)

Endless Circling

-Broadcast Frames laufen über alle Bridges immer wieder im „Kreis" , Verstärkungseffekt

(Bild Folie 31-33)

Table Rewriting

-Kennen die Bridges die collision Domains mit ihren Endsystemen noch nicht und erhalten Frames für unbekannte Endsysteme, werden diese an aller Ports immer wieder weitergeleitet (unknown destination).

Dadurch werden die Einträge in den Tabellen immer wieder geändert (Eintrag für System ‚A' springt von einer Seite zur anderen).

6-10)

Broadcast Storms treten bei Topologien mit „parallel paths" von Bridges/Switches im Ethernet LAN auf („Parallelschaltungen von Bridges/Switches")

Wird über solche Topologien ohne Spanning Tree Mechanismus ein Broadcast/Multicast Frame gesendet kommt es zum „endless circling" – das Frame wird ständig an allen Ports weitergeleiget und verbraucht somit unnötig Buffer der Bridges/Switches.

Ein Frame mit unbekannter Adresse verursacht Table Rewriting.. die Einträge bzgl. Eines „neuen" Endsystems ändern sich ständig.

6-11)
Mit dem STP Protokoll möchte man dem Effekt der Broadcast Storms (Endless circling, Table rewriting) entgegenwirken, welcher bei redundant Paths („Parallelschaltung von Bridges/Switches) auftritt.

-Das Protokoll sorgt dafür, dass zwischen 2 Endsystemen immer nur EIN aktiver Pfad exisitert – die 2 o.g. Effekte können somit nicht auftreten.

-Dazu wird ein eigenes communication protocoll für die Bridges untereinander eingeführt: Es verwendet sog BPDU Pakete mit MAC Multicast Adressen.

-bricht der „einzige“ aktive Pfad zusammen, wird einer der redundanten (alternativen) Pfade (falls existent) automatisch verwendet.

Spanning **Tree,** weil nach durchgeführtem Algorithmus die Netzwerkstruktur einem „Baum“ ähnelt.
Ethernet L2 Switches/Bridges unterstützen STP, wenn sie mit IEE801.2D beschriftet sind.

6-12)
Mit STP werden für Bridges/Switches auf einem Netzwerk 3 Parameter eingestellt:

Bridge ID = Priority# (2byte) + mac#(6byte)
Priority# kann von Admin konfiguriert werden, mac# ist üblicherweise die niedrigste MAC Adresse, die an einem Bridge/Switch angeschlossen ist.
→niedrigste Bridge ID hat höchste Priorität (im späteren Verlauf)

Port Cost = .. gibt an wie „schnell“ die jeweiligen Ports am Bridge sind… ist umgekehrt prop zur Bitrate.
Standardwerte:
10Mbit/s – PortCost=100
100 Mbit/s – PC = 19
1Gbit/s – PC = 4
10Gbit/s PC = 2 je höher die Bitrate („bessere Leitung“) umso niedriger ist PC

Port ID = port priority# , port#
Wird vom Admin eingestellt.

Pro Netzwerk mit STP gibt es nur einen ROOT BRIDGE – das ist jener Bridge mit der niedrigsten Bridge ID.
Von jedem ANDEREN Bridge wird nun der kürzeste Weg zum Root B. berechnet – die Ports, die als „Ergebnis“ für den kürzesten Weg errechnet werden sind ROOT Ports.
…

…
Nach Konvergenz des Spanning Trees werden die Root Ports und Designated Ports in den „forwarding state“ gesetzt, alle anderen Ports in den „blocking state“

6-13)
(siehe auch 6-12)

Nach Ermittlung des Root-Bridges und der Root Ports werden die Designated Bridges ermittelt.

Sind an einem LAN-Segment mehrere Bridges angeschlossen – wird einer zum designated Bridge – und zwar jener, bei dem die PortCost am ROOT PORT am geringsten ist („Schnellste Verbindung zum Root“)
Die Ports auf der anderen Seite dieses Bridges werden als designated Ports bezeichnet.

Nach Konvergenz des Spanning Trees werden die Root Ports und Designated Ports in den „forwarding state“ gesetzt, alle anderen Ports in den „blocking state“

6-14)

I. Root Bridge auswählen (Bridge mit niedrigster Bridge-ID)

II. Root Ports auswählen („kürzester Weg zum Root Bridge“)

III. Designated Bridges f. LAN Segmente mit mehreren Bridges auswählen (=Bridge mit niedrigster PortCost – also schnellste Leitung am RootPort)

IV. Designated Ports sind die Ports auf der „entgegengesetzten Seite“ zum RootPort der Designated Bridges.

V. Alle RootPorts und DesignatedPorts in den „forwarding state“ versetzen

VI. Alle anderen Ports in den „blocking state“ versetzen

➔ Erzeugt EINEN Pfad vom ROOT (Bridge) zu allen anderen LAN Segmenten.

6-15)
(siehe 6-12)

Der Netzwerkadministrator kann durch Konfiguration der 2 priority# und der PortCosts den genauen „Pfad“ des Spanning-Trees bestimmen und sein Netzwerk optimal an den Datenfluss (für den der „automatische“ Baum nicht unbedingt der ideale sein muss) auslegen.

6-16)
Der STP Algorithmus wird Periodisch mit der sog. „Hello Time“ (gespeichert im BPDU Frame) durchgeführt (standard= 2sec, vom Root Bridge ausgehend)

Zur Konfiguration des Baumes sendet der Root Bridge BPDU Frames aus, in denen das Feld „Message Age“ pro Passieren eines Designated Ports erhöht wird.
➔ Daraus ergibt sich für das Netzwerk ein „MAX Age“ , welches in allen folgenden BPDU Frames mitübertragen wird (Standard = 20sec). Dieser Wert gibt an, wie lange andere Bridges warten, wenn sie keine BPDUs mehr vom Root erhalten, bis sie den STP Algorithmus neu ausführen und ein neuer Baum entsteht

Forward Delay ist ein weiteres Feld im BPDU Frame, welches angibt wie lange es dauert, bis ein Port in den „forwarding state“ gesetzt wird. … praktisch tritt immer das doppelte Forward Delay auf .. 1xlisten (Zeit zum Konvergieren des BPDU – „Ausschnapsen, wer neuer Root Bridge wird) , 1xlernen (durch Mitlauschen Bridging Tabelle mit MAC Adressen befüllen) (Standard = 15 sec)

Der STP Algorithmus darf also max. 2*forwarding Time + Max Age dauern (standard 50sec)

Um zu verhindern, dass durch Frames mit unbekannter MAC Adresse intensives Flooding auftritt, werden die designated/root Ports erst nach der „Lernphase“ (15sec) in den Forwarding State gesetzt.

6-17)
Fällt der Root-Bridge aus, werden deshalb <u>keine zyklischen Hello MSG.</u> Gesendet.. alle Ports am Netzwerk „hören“ prinzipiell auf die Hello MSG. seitens des Root Bridges – entfällt die hello msg. , entdecken die anderen Bridges also, dass der Root-Bridge ausgefallen sein muss. Ein neuer Root Bridge wird ausgewählt und der komplette Baum nach dem STP Algorithmus neu festgelegt.

Im Standard Modus beträgt die Konvergenzzeit dafür MAX MSG AGE + 2*Forwarding Delay – also 20+2*15 = 50 sec.

6-18)
Fällt ein Designated Bridge aus , sendet dieser auch keine Bestätigung für die zyklischen HelloMSG mehr an den Root Bridge → andere Bridges entdecken dies, da sie (trotz eventuell geblockter Ports) die BPDU Msg. Empfangen..
Durch Ausfall eines Designated Bridges wird für „sein“ LAN Segment ein anderer Bridge (nach Regeln des STP Algorithmus) ausgewählt.
Im Standard Modus beträgt die Konvergenzzeit dafür MAX MSG AGE + 2*Forwarding Delay – also 20+2*15 = 50 sec.

6-19)

In Netzwerken kann es oft passieren, dass sich die Topologie dynamisch ändert (Endsysteme wechseln von einer collision domain in die andere).

Damit dieser Wechsel möglichst schnell von allen Bridges erkannt und deren Tabellen entsprechend angepasst werden, gibt es den Topology Change Mechanismus in STP. Dadurch werden evtl Daten nicht mehr bis zum nächsten „Hello-MSG Zyklus“ bzw. bis zum auftretenden Timeout an die falschen Ports gesendet. Das „FLAG“ Feld im BPDU Frame ist dafür reserviert.

Ablauf:
-Topolgieänderung tritt auf (Device a von LAN1→LAN2)
-Bridge an LAN 2 sendet solange BPDU Frames mit TCN (top.change notification) Flag **über seinen Root Port (also schnellstmöglich zum root)**, bis er selbst wiederum ein BPDU mit TCA (top.change ack) empfängt.
-Der nächste Bridge, der diesen BPDU TCN Rahmen erhält sendet TCA an den „vorherigen“ Bridge und sendet seinerseits den Rahmen solange weiter, bis er selbst einen BPDU TCA Rahmen erhält
-Sobald der Root Bridge erreicht ist erzwingt dieser eine Änderung in **allen** Bridging Tabellen mittels „Configuration“ BPDU Frames mit TC (top. Change) und TCA Flag gesetzt.

Dieser ganze Ablauf wird im Grunde durch das Endsystem hervorgerufen, welches seinerseits Daten in das Netzwerk sendet.

6-20)

Vorteile:

-bessere Performance, weil durch Hardware implementiert
-Arbeitet auf L1-L2 – darüber komplett Transparent
-verwendet lediglich MAC Adressen

Nachteile:

-kein Flow Control
-redundante Pfade können nicht zur load Balance ausgenützt werden.

6-21)

Die Konvergenzzeit im RSTP bewegt sich im Bereich von 1s – das ist quasi die Summe der Signallaufzeiten über das gesamte LAN.

Im RSTP wurden einige neue Mechanismen eingeführt, welche die „alten“ Parameter von STP – forwarding delay und max msg age obsolet machen.

- Bridges kommunizieren nun regelmäßig mittels BPDUs untereinander – dadurch ergeben sich neue Port Roles… für 2 Bridges an einem Segment hat B1 den root Port.. mittels einer proposal msg. teilt er dies B2 mit, welcher mittels agreement msg antwortet und seine Ports auf „**alternate**“ setzt.. B1 setzt den Weg zum Root sofort nach agreement auf **forwarding** – evtl noch andere Ports auf seiner Seite in „**backup**“.. der Root seinerseits bekommt davon gar nichts mit, weshalb der Vorgang relativ schnell abläuft.

- Die RSTP Bridges an sich können nun selbsständig entscheiden, ob sie manche ihrer eigenen Ports „sicher“ (ohne dass BC Storms auftreten können) in den forwarding state setzen können → **rapid transition into forwarding state**. Dies gelingt ihnen durch eine „Bewertung“ der Ports nach 2 neuen **Parametern**… ein Port wird entweder als *Edge Port* oder (gemessen durch evtl voll/halb duplex Verkehr) als *PTP bzw. shared Port* deklariert..
  PTP Ports mit full-duplex sind Verbindungen zwischen 2 Bridges..

  Edge Ports sind Verbindungen vom Bridge zum Endsysten. Diese beiden Ports können automatisch (und vor allem **schnell)** in den forwarding state gesetzt werden.

  Durch die Kommunikation der Brides untereinander ist der „listening“ Port state vom alten STP nicht meh

r erforderlich.

6-22)

Neue Port Rollen: *alternate Port , backup Port*

Die RSTP Switches kommunizieren nun mittels BPDU Frames untereinander und handeln sich untereinander alle Port Roles aus…
Der gesamte im folgenden beschriebene Mechanismus greift für jene Topologien, wo an einem LAN Segment mehr als 1 Bridge angeschlossen ist (redundant path)

-B1 wird zum Designated Bridge ausgewählt
-B1 teilt B2 mittels proposal BPDU mit, dass er nun Designated Bride ist und den root port hat.
-wenn B2 diese Nachricht auf seinem Root Port empfängt , antwortet er mit agreement
-B1 schaltet sofort nach agreement die ausgehandelten Ports in den forwarding state.
…

…
Port States:

STP	RSTP	Port in aktiver Topologie	Port „lernt“ MAC Adressen
Disabled	Discarding	No	No
Blocking	Discarding	No	No
Listening	Discarding	Yes	No
Learning	Learning	Yes	Yes
Forwarding	Forwarding	yes	yes

Disabled/Blocking State wird durch die neuen Port Rollen unter den RSTP Bridges obsolet, was zusätzlich Zeit einspart.

6-23)
Siehe 6-21, 6-22

Kapitel 7 – The Ethernet Revolution

7-1)
Ethernet mit Repeatern / Multiport Repeater:
Eine Collision = Broadcast Domain für das gesamte Netzwerk. Repeater sind lediglich Verstärker, die das Netzwerk nicht unterteilen könne, da sie keine Switching/Bridging Tabellen für ihre Ports besitzen. Das Netzwerk wird mittels CSMA/CD „nur“ im half-duplex Mode betrieben. (auch mit structured cabling und 10BaseT Technologie)

Ethernet mit Switches/Bridges (multiport):
Durch die Tabellen in den Bridges lässt sich eine Broadcast domain (=z.B. Firmennetzwerk) in mehrere collision domains aufspalten.. Die Bridges wissen, welche Endsysteme an welchen Ports hängen und können den traffic dementsprechend umleiten.. Endsysteme der coll. Domain 1 bekommen so z.B. vom traffic innerhalb von coll. Domain 2 gar nichts mit.

7-2)
Legacy Transparent Bridging (alt)
Netzwerk wird durch Bridges in kleinere coll. Domains unterteilt – collisions können somit nur innerhalb **einer** domain auftreten, traffic wird intelligenter als bei repeater technologie je nach Bedarf geblockt/weitergeleitet.
Die einzellnen coll. Domains haben **nachwievor** ein **shared media.**

…

…

Modern Ethernet Switching (neu)

Jede Verbindung (ES -> Switch ; Switch -> Switch) wird **einzelln** geführt.. im gesamten Netzwerk existiert somit kein shared media mehr, sondern **nurmehr PTP** Verbindungen. Ein Ethernet Switch ist ein Multiport Bridge (bessere Hardware)

Dadurch ist **full duplex mode** im gesamten Netzwerk möglich.. CSMA/CD wird somit oboslet.

7-3)

Siehe auch 7-2

Jede Verbindung wird einzelln geführt, Multiport Bridge, kein shared Media mehr vorhanden CSMA / CD obsolet.

7-4)

Collison Window/Slot Time – Zeit in welcher bei shared Media Ethernet Kollisionen auftreten können.

Da es beim Modern Ethernet Switching im full duplex Mode kein shared Media gibt, wird dadurch auch das gesamte CSMA/CD Verfahren mit slot time überflüssig… Jedes Endsystem kann somit „theoretisch“ ungestört senden, wann es möchte..

Die Switches untereinander sind mit High-Speed Fiberleitungen verbunden → ermöglicht lange Distanzen (mehrere km) → Wide Area Network

7-5)

Die Bandbreiten am Netzwerk sind natürlich immer begrenzt – ein Endsytem kann also nicht soviel und sooft Daten senden wie es möchte, da es somit das gesamte Netzwerk stören könnte. Somit ist Flow control am nähesten Netwerkknoten (L2 Switch) wünschenswert. Flow Control auf L4 ist für LANs ineffizient.

Dieses wird auch mittels einer MAC basierenden Flow Controll realisiert → „pause command“
Ein MAC Frame mit length feld = 8808h und pause Parametern wird gesendet.

Durch Flow Controll fällt somit auch die Begrenzung der maximalen Frame Größe (bei CSMA/CD weg).. Das ES kann solange Daten senden, bis es eine Flow Control Msg erhält.

7-6)

Autonegotiation („automatische Abmachung/Verhandlung) tritt für Fast (100Mbit/s) und Gigabit Ethernet (>= 1Gbit/s) als Mechanismus auf.
Mit diesem Verfahren tauschen die einzelnen Geräte am Netzwerk Informationen über ihre Verbindungsattribute aus.. Es ist hauptsächlich für Kupferverbindungen gedacht
Attribute sind:

-Datenrate (100BaseT, 100BaseTX, 1000BaseT)

-CSMA/CD oder full duplex (100BaseT, 100BaseTX, 1000BaseT)

-Acess Mode . full duplex/half duplex (1000Base T, 1000Base X)

-Flow Control Mode (1000BaseT , 1000BaseX)

Alle o.g. Parametern lassen sich bei Kupfertechnologie aushandeln. Für LWL Technologien kann man höchstens die letzten 2 genannten Parameter aushandeln.

7-7)

100Base T:

Schon bei den uralt Ethernet Technologien gab es die sog. „Normal Link Pulses“ (NLP), welche zur Überprüfung des Leitungsstatus verwendet wurden.

Die 100Base T Technologie verwendet FLP – f für fast, dh. codierte Pulse in kürzeren Zeitabständen. Um mit Autonegotiation nicht zu viel Datenrate zu verbrauchen wird es immer nur beim **neuen Aufnehmen einer Verbindung** betrieben.

Zuerst werden FLPs gesendet.. bekommt man aber nur NLPs zurück, weiß man, dass das Device am anderen Ende ein 10BaseT Device sein muss. Erhält man hingegen FLPs als Antwort verhandelt man zusätzliche Parameter (CSMA/CD – full duplex, Datenrate)

Bei 1000Base T kann man zusätzlich noch (für andere 1000Base T Devices) Access Mode und Flow control aushandeln

1000BaseX

Bei diesem Standard sind keine FLPs spezifiziert, da Autonegotiation nie für Fiber-Ethernet spezifiert wurde – 1000BaseX ist also nicht abwärtskompatibel und kann sich bei Autonegotiation nur mit gleich/höherwertigen Devices die Parameter AcessMode (full duplex, half duplex) und Flow controll Mode aushandeln.

7-8)

Die höheren Datenraten benötigten einen komplexeren Code als den „simplen“ Manchester Code… allerdings sollten die neuen Standards alle weitestgehend abwärtskompatibel sein – und so wurden Interfaceänderungen auf L1 durchgeführt, damit Endsysteme mit altem 10BaseT mit 100BaseT oder 1000BaseT Endsystemen kommunizieren können.

Codierverfahren:

-Das Datensignal wird 4 bitweise in eine sog 5bit „code group“(lauflängenbegrenzt) Codiert.

-Error detection und clock synchro sind leicht realisierbar

-leichter DC Anteil

-tatsächliche Datenrate wird dadurch (prinzipiell ähnlich wie bei Manchester) erhöht, da mehr bits übertragen werden.. tatsächliche Datenrate liegt für 100Mbit/s bei 125Mbit/s auf der Leitung.. das Verhältnis ist also 4/5 = 80% Effizienz – bei Manchester waren es nur 50%.

7-9)

Möchte man Fast/Gbit Ethernet über Repeater Technologie betreiben muss das CSMA/CD Verfahren angewendet werden – eine Slot time / collision window ist daher erforderlich. Damit Fast Ethernet/Gigabit Ethernet abwärtskompatibel mit den älteren Ethernet-Standards bleibt, behält man die minimale Rahmenlänge bei..

durch die schnelle propagation time bei den hohen Frequenzen ergibt sich durch die Abhängigkeit: Bitrage-Kabel-PropagationTime-CollisionTime-Länge

Eine extrem kurze Länge (10-20m) für Gbit Ethernet über Repeater… um größere Längen zu erreichen betreibt man carrier extension und frame bursting.

7-10)

Gigabit Ethernet sollte noch immer abwärtskompatibel bis zu 10BaseT Ethernet sein um die Technologie als solche Attraktiv zu halten.. Deshalb wurde die Minimale Rahmenlänge Ebenfalls auf 64Byte definiert.

Im Repeatermodus ergibt dies mit den zusammenhängen : Propagation Time – Collision Time – Bitrate – Länge – Kabeltyp eine **maximale Länge von lediglich 20m** .. was natürlich untragbar gewesen wäre. Zur Vergrößerung des max. Abstandes wurde carrier extension und frame bursting eingeführt.

Bei 10Gigabit Ethernet hat man CSMA/CD endgültig verworfen – es gibt dabei nur PTP full duplex Verbindungen… deshalb ist auch kein Repeaterbetrieb bzw. shared Media mit carrier extension und frame b. möglich.

7-11)

Die beiden Methoden sind lediglich im CSMA/CD Mode (half duplex)von Gigabit Ethernet erforderlich.

Carrier extension:

-Ein Frame wird künstlich länger auf einer Leitung gehalten, indem auf L1 extension bytes dem Ethernet Frame hinzugefügt und anschließen wieder abgenommen werden

Frame bursting:

-Um die unnützen extension bytes zu minimireen werden mehrere Frames „aneinandergereiht/kettet“ und als ein „frame burst“ (der dann auch eine passable Lebensdauer auf der Leitung hat) ausgesendet und vom Empfänger wieder dementsprechend zurrückgewandelt.

7-12)

Motivation: örtlich verteilte Endsysteme in einem LAN gehören derselben workgroup an und sollen nur die für SIE erforderliche Umgebung sehen → VLAN

-multiplexing mehrerer LANs über switches bzw. trunks-switches

(Bild Folie 22)

Für die Netzwerkgeräte heißt das:

-moderne Switches haben eigene Tabellen für VLANs

-dadurch können Broadcasts für jedes VLAN separat angewendet werden (jedes VLAN hat eine BC domain)

-separater Spanning Tree für jedes VLAN, bzw. RSTP für alle VLANs möglich

-VLAN Zuordnungsmethoden (port based, MAC based)

7-13)

VLAN Tags sind Markierungen für Frames, damit Switches diese dem richtigen VLAN zuordnen können.

Der VLAN Tag befindet sich im Feld „VID – VLAN Identifier“ .. erreichbar über

Rahmen → 802.1Q fields → TCI (Tag Contr. Info) → VID

7-14)

UP field gibt die User Priority an

Frames mit hoher UP werden am Switch zuerst abgearbeitet.

Das UP Feld befindet sich in den 802.1Q Feldern (nach preamble , DA ,SA) → TCI → UP

7-15)

Beim Multihoming gehört EIN Endsystem MEHREREN VLANs an. Die Frames müssen aber vom Endsystem sofort beim Absenden mit einem VLAN Tag versehen werden.

Dadurch kann ein Endsystem ohne zusätzlichen Switch auf seiner „Seite“ mit allen Endsystemen im Netzwerk kommunizieren, dessen VLAN es angehört. Ein Router zur Kommunikation zwischen 2 VLANs wird sich dadurch erspart.

7-16)

L2 Switches erlauben **keine** Kommunikation zwischen 2 VLANs

→ Es müssen Router verwendet werden.

Diese können durch mehrere physische Anschlüsse teil beider VLANs sein oder

Sie liegen auf Trunkleitungen zwischen den Switches und erkennen/ändern VLAN Tags der einzelnen Frames und leiten sie, falls erwünscht in andere VLANs weiter.

7-17)

Fast/Gigabit Channeling ist ein Verfahren von Cisco, bei welchem die Datenrate auf den full duplex Trunk Leitungen zwischen den L2 Switches optimiert wird.

Dies wird durch „bundling“ einzellner physischer Verbindungen zu einer logischen Verbindung ermöglicht.

Das dafür standardisierte Verfahren heißt LACP (IEE 802.3) – link aggregation control protocol.

7-18)

Die IEEE Working Group hat die Nummer 802.11 – bis dato gibt es 4 Standards:

-802.11 – „legacy“ , also urspr. 1 – 2 Mbit/s am 2,4GHz Band mit FHSS, DSSS Modulation

-802.11a - 54Mbit/s auf 5GHz Band mit OFDM Modulation

-802.11b - 5,5 und 11Mbit/s auf 2,4GHz Band mit DSSS Modulation – *zurzeit am meisten verbreitet*

-802.11g – neuester standard >20Mbit/s auf 2,4GHz Band

7-19)

Für WLAN wurde das lizenzfreie ISM Band auf den Frequenzbereichen ~2GHz und ~5GHz als Betriebsbereich festgelegt.

Auf diesem Band liegen z.B. aber auch die Mikrowellengeräte… andere Störeinflüsse auf diesen Frequenzen sind ebenfalls wahrscheinlich, da sie im Prinzip von jedem verwendet werden können.

Mittels intelligenter und störrobuster DSSS und FHSS Modulation werden WLAN Signale fehlerfrei übertragen. Eine gewisse Abhörsicherheit bietet FHSS ebenfalls. Weiters ist die Energiedichte von Signalen auf dem lizenzfreien Band limitiert, damit sich mehrere Benutzer nicht gegenseitig stören.. die beiden Modulationsverfahren stellen auch das sicher.

7-20)

WLAN benutzt eine Topologie, welche shared Media Charakteristiken hat…

Das Netzwerk ist in Zellen unterteilt,

innerhalb einer Zelle herrscht shared Media Verhalten vor

Jeder Zelle ist ein Access Point (WLAN Switch) zugeordnet… AP+Endsysteme = 1 BSS(Base Service Set)

Die WLAN Switches sind evtl mittels Fiber Ethernet miteinander verbunden.

Um nun sinngemäß kommunizieren zu können werden von allen Geräten 4 Adressen verwendet. 2 zusätzliche Steuerbits im Frame (sog. DS Bits – Distribution System) geben an, wie diese Adressen zu interpretieren sind.

Fall	*To DS*	*From DS*	Adr1	Adr2	Adr3	Adr4
1	0	0	Receiver	Sender	Cell	-
2	0	1	Receiver	Cell	Sender	-
3	1	0	Cell	Sender	Receiver	-
4	1	1	Cell	Cell	Receiver	Sender

Fall 1: Für Ad-hoc Netzwerke… Sender→ Receiver.. bzw. Broadcasts in Infrastructure

Fall2: Komm. Innerhalb eines BSS AP→Receiver… Frame kommt von Sender

Fall3: wie oben Sender→AP .. Frame geht an Receiver

Fall4: 2 APs kommunizieren miteinander Adr1:Sender→Adr:2 EmpfängerAP.

Infrastructure Mode: Cell Adresse = MAC Adresse des Access Points

Im Ad-hoc Mode gibt es nur 1 BSS.. daher sind auch keine Cell Adresses notwendig.

7-21)

Shared Media bei WLAN führt dazu, dass das CSMA/CD beim Ethernet zu CSMA/CA für WLAN wird.

CA steht für Collision Avoidance, was mittels Clear Channel Assesment Technologie (CCA) gelöst wird.

Diese stellt fest, ob ein Übertragungskanal benützbar ist – dabei werden die Signalstärken von WLAN (Modulations)Trägern bzw. von WLAN Trägern und dem kompletten Rauschen gemessen … übersteigt dieser Wert eine gewisse Schwelle, werden keine Daten übertragen.

Die ist im L1 implementiert.

7-22)

Die Zugriffsmethoden bei WLAN sind in Distributed Coordination Function (DCF) und Point Coordination Function (PCF) aufgeteilt. Ersteres unterstützt asynchronen Datenverkehr, letzteres synchronen, damit zeitkritische Services wie Voice over IP (VoIP) auch realisiert werden kann.

Synchrones und Asynchrones Zeitmultiplex wird gemischt… Ein „Superframe“ beinhaltet eine sog. „contention free“ Period – PCF Regime für Zeitkritische Anwendungen… sowie eine „contention Period“, in welcher asynchroner Datenverkehr erlaubt ist… die Dauer dieser Zeitfenster hängt von dem Beacon Intervall ab – der Zeit zwischen 2 Beacon Frames.

Auf dem Shared Media im WLAN senden Access Points ständig Beacon Frames aus- welche Informationen über die Modulationsparameter, den Zeitabstand, ihre adresse und die unterstützen Datenraten enthalten.. solche Frames werden als Service Set Management Frames bezeichnet.

Weiter Frames dieser Art wären:

Probe Request/Response: Endsystem Sendet an alle verfügbaren AP eine Probe Nachricht um seine „Umgebung“ im Netzwerk zu erkunden

Authentication/Deauthentication: AP erlaubt/verweigert Verbindung zum ES, bzw. bricht Verbindung ab

Association (response) : ES versucht Daten über AP herauszubekommen

Reassosiation/Disassosiation: ES wechselt Zelle, bzw. schaltet ab und verlässt Zelle

7-23)

Durch das shared Media bei LAN kann Network sniffing nicht verhindert werden.. es wird nicht einmal ein physischer Zugang (Stecker) erforderlich.

Deshalb müssen die Daten verschlüsselt übertragen werden um für dritte (unerwünschte) Teilnehmer im Netzwerk unverständlich zu sein.

…

…

Methoden:

WEP (Wired Equivalent Privacy, shared Secret key) – SEHR UNSICHER

TKIP (wi-fi, Temporal key integrity prot, shared secret key) .. WEP basiert, daher auch eher unsicher.

AES (advanced encryption sandard)

WPA (Wi-fi protecded Access) verwendet dynamische WEP Keys.. sicherer als alle bisher genannten.

In Entwicklung:

WPA2 Personal/Enterprise … gilt als große Security Hoffnung für WLAN.

Kapitel 8 IP Technology

8-1)

IP (Internet Protocol):

-Befindet sich im OSI Layer 3 (Network)

-Packet-Switching

-Connection Less

-Verwendet eigene IP Adressen für Packete.. strukturierte Adressen mit Topologie Information

-IP Host = Endsystem: Sieht das Netzwerk ab TCP (Transmission control Protokol) nicht

-IP Router verwendet eigene Routing Protokolle (auf OSI nicht einordbar) – kennt das Netzwwerk.

8-2)

IP ist ein L3 Protokoll… Conn. Less Packet Switching Technologie

TCP ist ein conn. Orientated L4 Protokoll, dass die Qualität der Übertragung gewährleistet.

IP und TCP sind in den IP Hosts implementiert… Die Router hingegen benötigen nur IP – welches die Netzwerkkomponennten komfortabel vereinfacht und den Endsystemen die Fehlerkorrektur überlässt. Die Router kümmern sich damit nur mit IP um die Übertragung über das Netzwerk, während die Endsysteme mit (IP) übertragen und mit TCP die Fehlerkorrektur vornehmen.

8-3)

-IP over everything… IP ist Technologie unabhängig und passt sich den unter L3 liegenden Protokollen laufend an. Kommt ein neues L2/1 Protokoll auf den Markt wird bald darauf eine IP Adaption dafür entwickelt.

-IP ist conn. Less packet switching.. mehr brauchen die Netzwerkkomponennten nicht zu können, d.h. es ist einfach zu realisieren… ein CL Service liefert TCP, welches in den Endsystemen realisiert wird (end-to-end principle).

- TCP ansich ist ein sehr gutes CO Protokoll

-das WWW wurde auf IP Basis erfunden und trägt einen großen Teil des Erfolges von TCP/IP.

8-4)

Skizze 8-4 (Bild folie 18)

Version- Version des IP Prot – zurzeit IPv4, IPv6 im Kommen

HLEN – Länge des Headers.. gibt an ab „wann“ Payload anfängt

ToS – Type of Service… damals wollte man IP auch QoS Ambitionen mitgeben.. das Meiste dieses Feldes wird jedoch nicht verwendet, da es sich nicht durchsetzte.. lediglich die „Precedence“ Felder, die als Klassifizierung von Frames dienen und in separaten Warteschlangen in den Routern (1 für jede Klasse) durchgearbeitet werden

Fragment ID – ID des IP Rahmenfragments, wenn Rahmen fragmentiert wurde

Flags – für Fragmentierung.. da IP unabhängig von Netzwerk ist, kann es sein, dass ein 5kByte Rahmen für ein max. 1,5kByte Ethernet fragmentiert werden muss.. enthält Fragmentierung JA/NEIN und „more fragments“ – ob noch mehrere Fragmente kommen

Fragment offset – gibt positon des Fragments im ursprünglichen Rahmen an

TTL – Time To Life „kill Mechanismus“ für das IP Protokoll

Prot- gibt an, ob es sich beim Payload um Infos für höhere Prots handelt

Header Checksum – für Fehlerkontrolle

SA – IP Adresse des Senders (strukturiert)

DA – IP Adresse des Empfängers (strukturiert)

IP Options – Netwerkoptionen … von denen jedoch aufgrund von Sicherheitsmängel die meisten heutzutage verboten sind.

PADs- Füllfelder, um den Header auf eine einheitliche Länge zu bringen

8-5)

Time To Live ist ein Kill-Mechanismus für das IP Protokoll … Somit wird sichergestellt, dass Datagramme nicht endlos im Netzwerk herumschwirren

TTL beschränkt die Lebensdauer eines Datagrams am Netzwerk, jeder Router dekrementiert den TTL wert um *seine* processing/waiting time.. ist diese relativ kurz wird der Wert (0-255 sec) um 1 dekrementiert.

Bei 0 wird das Datagramm verworfen

Endsysteme verwenden den TTL Wert des ersten ankommenden Datagramms um einen „reassembly timer“ zu setzen.

ICMP (Internet Contr. Msg Prot) generiert eine Fehlermeldung wenn TTL = 0

8-6)

IP Datagramme können z.B. eine Länge von 5kByte haben… da IP aber unabhängig vom verwendeten Netzwerk ist, kann es vorkommen, dass das Datagramm durch ein Netzwerk übertragen werden soll, dessen Parameter aber nur eine max. Länge von 1,5kByte (altes Ethernet) erlaubt → Datagramme werden „zerstückelt“ über das Netzwerk übertragen

Datagramme werden vom Router fragmentiert, wenn die Länge des Datagramms > MTU (max trans. Unit size) des darauffolgenden Netzes ist

Das Endsystem übernimmt reassembly für fragmentierte Datagramme… einmal fragmentierte Datagramme können wiederum unterschiedliche Wege über das Netzwerk nehmen – das Endsystem sollten jedoch alle erreichen – deshalb muss Buffer im Endsystem dafür reserviert werden. Das TTL Feld des ersten ankommenden Datagramms mit ‚MF=1‘ setzt den Buffer für die Rekonstruktion… läuft der Timer ab, wird der Buffer geleert und das Datagramm verworfen.

8-7)

Folgende Felder sind bei der Fragmentierung involviert:

Fragment ID, Fragment Offset, Flags, TTL

Fragment ID: Nummerierung der einzelnen Datagramme

Fragment Offset: Gibt die Position des Fragments im Bezug auf das urspr. Datagramm an..

Flags: More Fragments –gibt an ob nach dem Rahmen noch mehr Fragmente kommen, DF – gibt Anweisung an die IP PS das Datagramm NICHT zu fragmentieren

TTL – TTL Wert des ersten Fragments am Empfänger setzt den Timer zur Rekonstruktion des gesamten Datagramms.

8-8)

Die geforderte Methode heißt MTU Discovery

Bei DF (dont Fragment) Flag gesetzt erzeugt der Router eine ICMP Msg an den Sender, wenn das Datagramm zu groß für das nachfolgende Netz ist (Länge Datagramm>MTU des nachfolgenden Netzes). Der Sender verringert dann die Länge des Datagramms solange, bis keine ICMP Msg dieses Routers mehr zurrückkommt – die Länge des Frames ohne ICMP Antwort ist <=MTU des nachfolgenden Netzes.

ICMP – Internet Control Msg. Prot.

8-9)

Mit dem Type of Service Feld wollte man dem IP Ambitionen für Quality of Service geben. Das ToS Feld besteht aus 2 Teilen: Precedence, DTRC bits

Durch ToS – PRECEDENCE kann man die Datagramme Klassifizieren – Router haben für jede Klasse eine eigene Warteschlange… wird heute noch verwendet.

Durch die DTRC Bits sollte eine Routingentscheidung getroffen werden, falls das Datagramm verschiedene Weg zur Verfügung haben sollte.. Aspekte für den Pfad wären: min delay, max throughput, max reliability, min. cost.

Die Routingoptionen im ToS Feld haben sich allerdings nicht durchgesetzt und werden heute nicht mehr verwendet. QoS wird mit Protokollen auf einem höheren Layer geboten.

8-10)

Das ToS Feld wurde auf das „differentiated service codepoint“ Feld umdefiniert… Im Prinzip dient dieses Feld nun dazu, den Datenverkehr zwischen 2 IP Hosts zu Klassifizieren

Die Internet Provider können dieses Feld nun dazu nutzen den Datenverkehr ihrer Kunden zu Klassifizieren und Ihnen **über ihr eigenes Netzwerk eine sichere Performance zu bieten**. Alles was über diese Performance hinausgeht, wird mit einer niedrigeren Klassifizierung versehen… Die Kunden der Provider bekommen somit mind. genau die Leistung, die sie bezahlen.

Das Internet der Zukunft entwickelt sich wahrscheinlich in diese Richtung.

8-11)

Mögliche Funktionen, welche durch das IP Options Feld bereitgestellt werden:

-Bereitstellung für timestamps (Datagramme werden mit Zeiteinträgen versehen)

-Spezielles Routing:

-Record Route: Der „Weg“ des IP Datagramms wird aufgezeichnet (jeder Router trägt sich bei der Übertragung ein)

-Loose Source Routing: Im Datagramm wird eine Liste von Routern bereitgestellt, durch die das IP Datagramm infolge durchgehen MUSS – andere Wege sind ebenfalls möglich

-Strict Source Routing: Im Datagramm wird eine Liste von Routern bereitgestellt, durch die das IP Datagramm infolge durchgehen MUSS – alle anderen (möglichen) Wege werden ignoriert – **security Problem mit Hackern**

Fast alle IP Optionen werden Heute von Firewalls blockiert – aufgrund der security Probleme, die dadurch leicht auftreten können.

8-12)

Eine IP Adresse ist strukturiert (enthält Topologieinformation – vgl. „Postadresse“ Land-Stadt-Straße-Hausnummer)

Sie besteht aus 4Byte (32 bit) – jedes Byte wird (dezimal) separat mit einem Trennpunkt angeschrieben: *0-255* . *0-255* . *0-255* . *0-255* - z.B. 192.34.2.1

Prinzipiell besteht sie aus 2 Teilen – einer NET ID und einer HOST ID .. je nach Klasse sind die ersten 1-3 Bytes NET ID und die letzten 3-1 Bytes HOST ID.

Die NET ID muss weltweit einmalig sein, über die HOST ID kann der jeweilige Netzwerk Admin / Provier frei verfügen.

8-13)

Die verschiedenen Adressklassen bei IP unterscheiden sich je nachdem wieviel Byte der Adresse die NET ID und die HOST ID ausmacht. Weiters sind die Klassen durch bestimmte Anfangsbits speziell gekennzeichnet:

Klasse A-C sind unicast Adressen, D multicast , E experimentell

Class A: ‚0‘ NET ID (7bit) . host ID . host ID . host ID

Class B: ‚10‘ NET ID (6bit) . NET ID . host ID . host ID

Class C: ‚110‘ NET ID (5bit) . NET ID. NET ID. host ID

Class D: ‚1110‘ , Rest Multicast …

…

Class E: ‚1111‘ , Rest experimentell

Die Anfangsbits müssen immer einer speziellen Klasse entsprechen – dadurch ergeben sich die Adressierungsbereiche des ersten Bytes für die Klassen. (**first octet rule**)

A: 0-127

B:128-191

C:192 – 223

D:224-239

E:240-255

8-14)

Der Router selbst hat eine IP Adresse – Zusammen mit den 5 angeschlossenen Netzen kann man den Router also mit 6 verschiedenen IP Adressen erreichen.
In seinem Routing Table ist jedes angeschlossene Netzwerkinterface mit entsprechendem Port aufgelistet.

Router Benötigen IP Adressen um für IP-Hosts als Default Gateway für alle Packete die an eine fremde NET ID adressiert sind agieren zu können (IP Hosts senden all ihre Pakete an den Def. GW).

Jeder Router benötigt eine IP Adresse um von anderen Routern in deren Tabellen beim dynamischen Routing als Wegweiser für seine angeschlossenen Netzwerke eingetragen zu sein.

8-15)

Limited Broadcast ist ein Broadcast nur auf einem IP Netzwerk (auf einer NET-ID). Getriggert auf dem selben Netzwerk

Die Adressierung dafür ist ‚1‘ für alle Bits d. IP Adresse

Ein Datagramm mit limited Broadcast beinhaltet eine Broadcast MAC Adresse (DA Feld auf ‚1‘)

Das ARP (Adress Resolution Protokoll) , welches die Brücke zwischen IP und MAC Adressen ist verwendet limited BCs auf jeden Fall

8-16)

Adressierung für IP directed BC: {<net –id > ; ‚1‘ für alle restlichen Bits}

Damit ließe sich ein Broadcast auf ein fremdes Netzwerk von einem anderen Netzwerk triggern. Der Router, der den BC für sein Netz empfängt würde ihn in einen limited BC umwandeln.

IP directed BC ist heute wegen Securitygründen verboten .. der Router leitet IP Datagramme mit dieser Adressierung also erst gar nicht weiter.

8-17)

Zu Beginn des Internetzeitalters reichten 2 Hierarchien für IP Adressen komplett aus (NET ID , HOST ID) .. durch die massive Nutzung und die Verbreitung von LANs wurde mit Subnetting eine 3. Hierarchie eingeführt, um den Adressraum deutlich auszuweiten.

Innerhalb einer NET ID können durch Subnetting ‚interne Netze' etabliert werden… von außen betrachtet sieht man hingegen immer noch ‚nur' die NET ID

Die IP Adresse (untersch. Klasse!) wird mit einer Subnetmask verknüpft – Bits der HOST ID werden dadurch zu SUBNET ID Bits. Die Subnetmask selbst darf nur bestimmte Werte annehmen.

Folie 47

Um die Adressierung nachvollziehen zu können wird folgende Notation verwendet:

<IP Adresse> / **X** (Class A-C)

X ist die Anzahl der Bits (von links nach rechts angewendet), die in der Subnetmask ‚1' sind.

Die Subnetmask selbst hat das Format einer IP Adresse.

z.B. X = 8 Subnetmask : 1111 1111. 0000 0000 . 0000 0000. 0000 0000 bzw. 255.0.0.0

z.B. X=16 Subnetmask: 255.255.0.0

Dadurch ergibt sich eine ‚neue' Notation für IP Adressen: {net –id , subnet – id, host id}

8-18)

Anhand eines BSP verdeutlicht:

10.0.0.0 / 16 (Class A)

Heißt:

10.0.0.0 ist das erstmögliche Subnet (subnet ZERO) – sollte jedoch nicht verwendet werden.

10.1.0.0 erste „echte" subnet

10.2.0.0

10.3.0.0

10.255.0.0 subnet broadcast… sollte auch nicht als Subnet Adr. verwendet werden.

Das Problem mit der Subnet Zero und der Subnet Broadcast adresse ist, dass Doppeldeutigkeiten auftreten. Ein IP Datagramm an 10.0.0.0 adressiert könnte an das gesamte Netz 10 oder an das Subnet 0 adressiert sein. Ein Broadcast 10.255.255.255 könnte an das Gesamte Netz 10 oder nur an das Subnetz 255 (im Netz 10) gerichtet sein… um diese Schwierigkeiten zu umgehen wird Subnet Z. und Subnet Broadcast niemals als echte Subnetadressen verwendet.

8-19)

Der IP Host kennt in der Regel nur sein eigenes (sub)netz und einen Default-Gateway (Router) an welchen er IP Datagramme mit „fremden“ NET-IDs sendet. Sichtweise der Hosts ist also lokal.

Konfiguriert werden muss:

-Einmalige IP Adresse am Subnetz

-Subnetmask

8-20)

Direct delivery: Pakete werden innerhalb eines (sub)Nets versendet … DA enthält die NET/subnet- ID des Senders… Die IP hosts kümmern sich eigenständig um den Datenaustausch

Indirect delivery: Pakete enthalten in der DA eine NET/Subnet ID, die außerhalb der des Senders liegt. Ein „Default Gateway“ (Router) wird bei jedem IP-Host konfiguriert, an den er die Pakete zur Weiterleitung sendet.

Router melden sich mittels best. Protokolle bei Endsystemen an – somit wissen die IP Hosts von deren Existenz auf deren (lokalen) Netzwerk.

8-21)

-Für lokale Komm: (Router untereinander)

IP Adressen der Router/Zielnetze mit Zuordnung der Ports am eigenen Router
(ohne MAC Adressen der IP Hosts)

-Damit Router global kommunizieren können benötigen sie über das o.g. hinaus noch einen Speicher, der die IP Adressen ihrer angeschlossenen IP Hosts mit deren MAC Adressen gegenüberstellt… so kann jedes Paket überstellt werden.

Die Router bringen MAC Adressen z.B. durch ARP (Adr. Resolution Prot.) in Erfahrung und speichern diese im ARP Cache.

Sichtweise der Router ist global.

8-22)

Routing Tabelle mit Einträgen für jede Zeile der:

NET-ID / (subnetmask) ,
Next Hop (wohin muss das Paket dafür gesendet werden) ,
metric (hops) – wie ‚weit' ist es bis dorthin,
port (an welchem Port ist das),

Zeitstempel (dynamisches Routing)

Zusätzlich für die lokal angeschlossenen Endsysteme: ARP Cache mit einer IP Adresse → zugeordneter MAC Adresse pro Zeile

8-23)

-Destination Based Routing:

Verwendet nur DA bei der Weiterleitung. SA wird nicht betrachtet → evt. Security Probleme

-Hop-by-Hop Routing:

Datagramme werden immer nur zum nächsten Router weitergeleitet (kann für dieselbe Übertragung untersch. sein, Pakete kommen evtl. nicht in richtiger Reihenfolge an.

-Least Cost Routing:

Der ‚schnellste' Weg wird bei der Routing Entscheidung für die Datagramme gewählt… ist dieser überlastet wird er dennoch verwendet (Autobahn-Landstraße Eselsbrücke) d.h. bei überlasteter „Autobahn" wird die freie „Landstraße" trotzdem nicht verwendet.

8-24)

ARP : Adress Resolution Protocol , Layer 2 – Versorgt Router/IP Hosts mit der Verknüpfung von MAC ←→ IP Adressen.

ARP wird auf LAN Netzwerken **mit shared Media Topologie** benötigt… der Sender muss die MAC Adresse des Empfängers kennen und bezieht diese mittels ARP.
ARP läuft lokal auf den jeweiligen LANS ab.

Prinzipieller Ablauf:

Gerät A kennt IP Adresse von Zielgerät B (auf selbem Netz), benötigt jedoch MAC Adresse

Gerät A sendet ein ARP Request als Ethernet Broadcast (MAC DA = ‚1') … mit den ARP Parametern und seiner IP/MAC Adresse und der Ziel IP Adresse im ARP Protokoll Feld

Gerät B empfängt ARP Request und speichert die ARP Daten von A in einem Cache ab, sendet ein ARP Response, womit A die gewünschten Daten hat.

8-25)

ARP Requests sind L2 Ethernet Broadcasts… alle Empfänger haben somit die IP/MAC Adresse des Senders. Alle Geräte werden durch den BC geweckt

Es reagiert lediglich das durch ARP angesprochene Gerät mit einer Antwort → ARP Response ist direkt an das fragende Gerät gerichtet (unicast)

Das Gerät, das mittels ARP Request eine MAC Adresse eines anderen Geräts angefragt hat, erhält diese von diesem Gerät in Form eines ARP Replys im sog. ‚Src HW' field des Protokolls. Weiters ist noch dessen IP Adresse enthalten, die aber ohnehin schon bekannt war.

8-26)

Gratuitous ARP („grundloser" , „kostenloser" ARP)

Dies ist eine ARP Request Message, die im LAN als Broadcast an die **eigene** IP Adresse adressiert ist… das Device richtet einen ARP an sich selbst.

Dadurch (Broadcast) erhalten aber alle anderen Endsysteme die Information des Senders (IP und MAC Adresse). Falls ein Gerät die SELBE IP Adresse wie der Sender hat antwortet es auf diesen ARP, was auf einen Fehler bei der IP Adress vergabe hinweist.

Anwendungen:

Manuelle Konfiguration von IP Adressen → Sicherstellung der Einmaligkeit mittels Gratuitous ARP „wer hat diese IP Adresse noch?"

Refresh Mechanismus für alle anderen ARP Caches der Endsysteme am LAN… falls sich IP/MAC Adresse des Senders ändern.

8-27)

Im ARP Cache der Endsysteme sind die bekannten IP Adressen mit den zugehörigen MAC Adressen verknüpft.

Ein Refresh erfolgt durch gratuitous ARP durch ein anderes Endsystem am shared Media LAN bzw. durch ARP Request Broadcasts auf L2 (von ANDEREN Systemen). (Router leitet diese NICHT weiter) Eine weitere Möglichkeit ist eine einfache ARP Anfrage eines Endsystems am LAN für eine best. IP Adresse.

Der ARP Reply kann nicht authentiziert werden – im ARP Protokoll findet sich kein Mechanismus dafür – ein Endsystem „glaubt“ somit jeden ARP Reply bzgl. einer IP Adresse (ist ein große Security Problem -> ARP Spoofing, ARP Cache Poisoning, Netzwerktool Cain&Abel nutzt diese Eigenschaft von ARP sogar)

8-28)

ICMP = Internet Control Message Protocol

Durch die „best effort“ Natur des IP wird versucht durch ICMP Nachrichten/Mechanismen die Qualität der Datenübertragung zu verbessern. Jedes mal wenn Übertragungsprobleme auftreten wird an den Sender vom Gerät, welches die Probleme zuerst entdeckt eine ICMP Msg. geschickt. (als IP Datagramm).

ICMP Messages werden nur EINMAL ausgesendet, wenn die ICMP Message selbst wieder Störungen erleidet, wird der Sendevorgang nicht wiederholt , da es sonst zu „Lawineneffekten“ von ICMP Messages auf dem Netzwerk kommt → „ICMP mustn’t invoke ICMP“ – außer PING commands… „Echo Request“ – „Echo Reply“.

8-29)

-PING Funktion – Echo-Request , Echo Reply

-Destination unreachable (Host, Network, Prot…)

-Source Quench – drosseln des Senders (generiert von einem Router auf dem Weg zur Dest.)

-Redirect – dem Host wird ein anderer Pfad zugewiesen

-Time Exceeded … TTL (time-to-life) der Datagramme bei der Übertragung abgelaufen

-Parameter Problem im IP Header

8-30)

PING = Packet Internet Groper

Misst Verfügbarkeit und Verzögerung (round-trip-time… hin-und zurrück) des angepingten Ziels

Wird mittels ICMP Echo Request (ausgesendet) und ICMP Echo Reply (Antwort) realisiert …

…

Der Benutzer kann damit überprüfen ob er selbst andere Stationen erreicht, bzw. ob andere Stationen ihn empfangen können… Damit weiß man also, ob der Netzwerkpfad zum gewünschten Ziel aktiv ist.

8-31)

Destination unreachable hat Type 3 im ICMP Frame – durch 6 verschiedene Codes wird dem Empfänger zusätzliche Information gegeben:

0 – Network unreachable – das adressierte Netzwerk ist down oder es existiert kein Router der es kennt

1 – Host unreachable … der adressierte IP Host im Netzwerk ist nicht bekannt / down (das Netzwerk existiert aber)

2- Prot. unreachable … das im IP Header angegebene Prot. ist nicht verfügbar (geschickt von Endsystem … Empfänger)

3- Port unreachable – Port in L4 nicht verfügbar (geschickt vom Endsystem/Empfänger)

4- Fragmentation needed, but D(ont) F(ragment) Bit=1 … Paket für Netzwerk, durch das es durchtransportiert werden soll ist zu groß und darf nicht fragmentiert werden – vom jeweiligen Router generiert.

5 – Source Route failed angegebener Pfad in den IP Options (strict source routing) konnte nicht befolgt werden – vom jeweiligen router Generiert.

8-32)

Router, die schon relativ ausgelastet sind (Buffer fast voll) „wollen“ keine wirklich zusätzliche Last und senden daher, im Falle von Auslastung eine ICMP MSG mit „Source Quench“ – bedeutet, der Sender soll weniger Daten auf einmal senden (Bitrate verringern).

Damit wäre prinzipiell flow control auf L3 mittels IP möglich – allerdings ist dieser Befehl auch ein großes Security Problem, weshalb er nicht sofort strickt befolgt wird (Grundregel: „bei IP traut niemand niemanden“).

TCP sollte auf diese Nachricht auf jeden Fall Rücksicht nehmen, da es connection orientated ist.

8-33)

Traceroute ist ein Befehl, welcher mittels ICMP/ TTL Mechanismen den Pfad zu einem gewünschten Ziel aufzeigt. Dabei wird wiefolgt vorgegangen:

Packet mit TTL=2 ausgesendet , 2. Router zum Ziel sendet ICMP mit TTL Exceeded

→ Packet mit TTL = 3 ausgesendet , 3. Router zum Ziel sendet ICMP mit TTL Exceeded

Usw….

→ Packet mit TTL = X ausgesendet , X. Router zum Ziel sendet ICMP mit TTL Exceeded

→ Packet mit TTL = X+1 ausgesendet… keine Antwort vom Router – Endsystem reagiert, weil der Inhalt des Packets einen Port >30 000 (existiert nicht) von UDP Prot. adressiert.. das Endsystem sendet zuletzt eine ICMP MSG mit „Dest. Unreachable → Port unreachable“ und der Sender weiß, dass er X Sprünge bis zum Ziel braucht (IP Adressen der verwendeten Router sind ebenfalls aufgelistet).

Traceroute zeichnet nur den „besten“ Weg zum Ziel auf – da die Routing Tabellen nur diesen Weg enthalten.

8-34)

Router, die für ein Ziel einen besseren Weg kennen, als den des „default Gateways“ , senden die Packete auf diesem Pfad und geben dem Sender durch eine ICMP Redirect MSG bescheid, sodass dieser seine Packete mit dem gewissen Ziel von vorhin nurmehr an den angegebenen Router sendet.

Dies ist allerdings ein sehr großes Security Problem, weil Endsysteme mit unechten ICMP Redirect Nachrichten „geleimt“ werden können, damit sie ihre (vertrauensvollen) Daten an einen anderen (evtl. von Hackern kontrollierten) Router senden.

8-35)

First Hop Redundancy Problem:

Oft sind an Endsystemen (IP Hosts) mehrere Router angschlossen – aber <u>wie</u> können die IP Hosts die verschiedenen lokalen Router erkennen, um ihre IP Messages an den richtigen Router zu senden? Normalerweise hat der IP Host nur einen Router als „default Gateway“ – fällt dieser aus, kann das System gar nicht mehr kommunizieren → es sollte selbstständig ein Ersatzgateway gefunden werden

Lösungsansätze:

Auf Ebene der IP Hosts … Abhängig vom Operation System am Host (Proxy ARP, IDRP, DHCP , IP Routing)

Auf Ebene der IP Router (Netzwerk) .. OS des IP Host braucht nur Basisfunktionen (also nur 1 Default Gateway)… Router müssen Lösungsfkt. anbieten (Prots: HSRP,VRRP , GLBP)

8-36)

Durch folgende Protokolle (OS der IP Hosts muss diese supporten)

-Proxy ARP

Router gibt seine MAC Adresse für „fremde“ Netze an seinen IP Host

-IDRP (ICMP Router Discovery Msg Prot)

Router senden ihre IP Adressen durch ICMP periodisch aus – IP Hosts sehen dass und merken sich alle IP Adressen der Router als mögliche Default Gateways

-DHCP (Dyn. Host config Prot)

-IP Routing (RIPv2 , OSPF)

8-37)

Durch folgende Protokolle (Router müssen diese supporten , Hosts nicht)

-HSRP (Hot Standby Router Protocol)

Ein virtueller Router erscheint für alle angeschlossenen Hosts; in Wirklichkeit sind jedoch mehrere Router vorhanden, die den Verkehr managen… 1 ist Active, andere Standby. für die Hosts erscheint immer nur eine virtuelle IP und MAC Adresse.

-VRRP ähnlich wie HSRP .. HSRP von CISCO lizensiert.. VRRP ist Alternative

-GLBP

8-38)

Proxy ARP: Der Router (am LAN hängend) beantwortet stellvertretend (proxy) einen ARP Request für eine IP Adresse eines anderen – aus seiner Sich über seine RT Tabelle erreichbaren – IP Subnetzes.

IP Host und Router müssen Proxy ARP für Funktionalität unterstützen. (ARP Caches vorhanden, Host muss ARP Requests für IP Adressen, die nicht im eigenen Subnet liegen senden können – Trick: DAFÜR default gateway Eintrag auf eigene IP Adresse zeigen lassen.)

Funktionsweise:

-ARP Cache von allen Routern vorhanden , Routingtabellen ausgefüllt

-IP Host sendet ARP Request für FREMDE Net ID (normalerweise ARP nur für gleiche NET ID)

-IP Router, der die fremde NET ID kennt antwortet mit EIGENER MAC Adresse an den fragenden Host

→ Host trägt diese MAC Adresse in seinen ARP Cache ein und hat somit den Weg für die fremde NET ID gefunden. – IP Router übernimmt mit der Angabe seiner MAC an den Requestor die „Verantwortung“ für weiteren Datenverkehr in die fremde NET ID …

…

Methode ist aufgrund hoher Konvergenzzeit der ARP Caches bei Netzwerkänderungen anfällig (Black Hole). Ändert sich die Topologie (andere Router für LAN Segment) , haben die IP Hosts falsche proxy ARP Infos, und senden ihre Daten u.U. TROTZDEM an einen (mittlerweile möglicherweise inaktiven) alten Router.

8-39)

Bei Hot Standby Router Protocol (HSRP) sind mehrere Router für IP Hosts als ein einziger „virtual Router“ sichtbar…. Alle IP Hosts haben als Default Gateway die IP/MAC Adresse des Virtual Routers. Die Router untereinander bestimmen, wer der aktive Router ist (über den der Traffic läuft) und wer als Standby Router im Störfall für den active Router einspringt.

Für Netzwerke mit HSRP benötigen die IP Hosts lediglich die Basisfunktionalität (1 Default Gateway) um von der Technologie zu profitieren.

8-40)

HSRP ist von Cisco lizensiert – VRRP ist die alternative und funktioniert ähnlich wie HSRP.

HSRP funktioniert einwandfrei auf shared Media Topologien – in Wirklichkeit existieren jedoch L2 geswitchte LANs (Stern Topologie)viel häufiger.

d.h. es hängt z.B.

(active) Router A auf P1 des L2 Switches (mit virtueller MAC_V1)
(stdby) Router B auf P2 des L2 Switches (mit eigener MAC_B1)

Alle IP Hosts adressieren ihre Daten an MAC_V1 (default Gateway) → Signal wird auf P1 geswitcht

Wenn Router A ausfällt übernimmt Router B den active Router (er hat nun MAC_V1)
allerdings steht MAC_V1 im L2 Switch immer noch auf Port 1 und nicht auf Port 2.. durch einen Gratuitous ARP des neuen active Routers B wird die Switching Tabelle des L2 Switches aktualisiert und die Frames der IP Hosts werden wieder richtig geswitcht.

8-41)

Das PPP Protokoll wurde ursprünglich dazu entwickelt, damit Router verschiedener Hersteller auf L3 miteinander kommunizieren können (auf einer seriellen Leitung Router zu Router).

Der Hauptfokus von PPP heute ist die Verwendung als Dial-In Protokoll für die Einwahl ins IP Netzwerk von Kunden→Provider

8-42)

HDLC (high lvl data link control) für framing und encapsulation (Verschachtelung)

Bitstuffing für synchrone seriellen Leitungen

Modifiziertes bytestuffing auf asynchronen seriellen Leitungen

LCP (Link control Prot)

„Einwahl“ und „Abmeldung“ aus dem IP Netzwerk

Konfiguration und Test der Verbindung

NCPs (Network control Prot)

Für Interoperabilität am Netzwerk

8-43)

-Verbindungsaufbau , Kompression ja/nein , Form der Encapsulation, multilink ja/nein , Authentifizierung ,

-Verbindungsende

Bei PPP wird kein Error Recovery durchgeführt

8-44)

Für eine PPP Verbindung gibt es 4 verschiedene Phasen:

Phase 1 + 2: Verbindungsaufbau und Konfiguration mittels LCP… Kompression ja/nein, Form der Encapsulation, multilink, Authentifizierung

Phase 3: Network layer Protokoll (Layer 3) wird ausgehandelt und konfiguriert – mittels verschiedener Network Control Protokolle (NCPs)

Phase 4: Verbdinungsabbruch – üblicherweise durch höheren Layer ausgelöst – mittels LCP

8-45)

Mit NCP werden die L3 Protokolle für die Verbindung konfiguriert.

IP Control Protokoll (IPCP) regelt die Vergabe von IP Adressen und Def.GW, DNS an den eingewählten und authentizierten IP Host.

IPCP lässt sich mit DHCP für LAN (First Hop Redundancy Problem) vergleichen.

8-46)

RAS – Remote Access Server ist im Prinzip ein IP Router, der am POP (Point of Prescence) , also am gesamten Netzwerk des ISP /INTRANET hängt. Er ist ein Einwahlserver für die Kunden eines ISP mit möglichst vielen Verbindungen zu den Modems. Damals wurde diese Technik für ISDN Einwahl-Verbindungen eingesetzt … Heute zur Einwahl in ein Firmeninternes Netzwerk (Intranet)

Über ihn läuft die Einwahl und Authetifizierung – Verifikation wird evtl. über einen separaten Security Server gehandhabt, auf welchem heutzutage accounting (speichern des Datenvolumens pro User, Onlinezeit) und speichern der Passwörter für Auth. betrieben wird.

8-47)

CHAP: Challenge Authentication Protocol

Dient zur Authentifizierung des (neu eingewählten) Users am IP Netzwerk … z.B. User wählt sich bei ISP ein.

„two way authentication“ ist möglich… User autentiziert Netzwerk <←→> Netzwerk authentiziert User, allerdings wird meist nur one way verwendet… derjenige der CHAP startet authentiziert den Anderen. CHAP läuft in 3 Phasen ab:

(PPP Link durch LCP steht)

-„Prüf“ Station sendet eine challenge msg. nur neuen Station; diese enthält deren user ID sowie eine Zufallszahl

-neue station antwortet mit einem Wert , errechnet durch eine crypto die für die empfangene User ID ausgehandelt wurde

-„Prüf“ station vergleicht den Wert mit eigenem (nach dem selben Verfahren) kalkulierten Wert und senden ein success/failure an die neue Station.

8-48)

Bei der Entwicklung der ADSL Übertragungstechnik spezialisierte man sich auf die Verwendung von ATM – ATM ist ein CO PS Service mit Virtual Circuits und sehr komplex.

Die Entwickler hatten urpsrünglich ein reines ATM Netzwerk im Sinn – Fakt ist, dass viele ISP heute IP Netze verwenden, und ATM so gut wie ausgestorben ist. Deshalb muss eine ADSL Modem mittels PPP (genauer PPPoA PPPoE PPTP) betrieben werden… je nach Technologie des ISP.

Darüber hinaus sind ADSL Modems derart komplex, dass sie sich nicht gut in Stand-PCs (noch schlechter in Notebooks) integrieren lassen… Es wird also eine Ethernet Verbindung von ADSL Modem zum PC/Notebook mittels PPPoE / PPTP auf der einen Seite und eine PPPoA Technologie zum ISP hin betrieben. Das ADSL Modem tritt dabei als Packet switch auf, welches (je nach Anwendung) die PPPoE , PPPoA , PPTP Signale übersetzt.

Kapitel 9 Introduction to IP Routing

9-1)

Beim Zustellen von IP Datagrammen gibt es 2 verschiedene Möglichkeiten: direct delivery und indirect delivery

Direct delivery ist dann der Fall, wenn die NET-ID der DA gleich wie die NET ID des Senders (IP Host) ist – in diesem Falle sendet der IP Host die Pakete direkt über das Netz an den Empfänger (ohne Router)… ist die NET ID nun nicht gleich der eigenen NET ID kommt indirect delivery zum Tragen: Der IP Host sendet das Datagramm an einen Router, der sich um die Weiterleitung des Datagramms kümmert. Meist ist für jeden IP Host EIN Router (**sog. Default Gateway**) als standard-Router festgelegt, an welchen Datagramme mit „fremden" NET-IDs zur Weiterleitung übermittelt werden

9-2)

Scheint für ein IP Datagramm als DA eine NET ID auf, die nicht im Routing Table eines Routers gespeichert ist, verwirft er das Datagramm normalerweise und generiert eine ICMP Nachricht an den Sender.

Durch Konfiguration einer Static Route (an den Routern) wird dieses o.g. Verhalten (welches vielleicht oft nicht erwünscht ist) umgangen: „Ein Router sendet alle Datagramme mit unbekannten NET IDs über einen bestimmten Port an einen best. Router" .

Oft wird die Default Route im Router mit 0.0.0.0 gekennzeichnet.

9-3)

In einem separaten IP Netzwerk ist es ausgeschlossen, dass jeder Router sämtliche Ziel NET-IDs im Routing Table gespeichert hat.. Normalerweise verwirft ein Router das IP Datagramm mit unbekannter NET-ID in der DA und generiert eine ICMP Msg. an den Sender. Eine Kommunikation eines IP Hosts im angesprochenen Netzwerk mit einem anderen Host über das Internet wäre somit nicht möglich.

Durch die Default Route wird das Datagramm aber kontinuierlich bis zum ISP-Netzwerk weitergeleitet, wo die Chance sehr hoch ist, dass einer der Router im ISP Netzwerk die Ziel-NET-ID kennt → oben angesprochene Kommunikation wird dadurch ermöglicht.

Dabei Spielt das „Hop-by-Hop" Paradigma eine entscheidende Rolle – „der jeweilige Router, muss nur den Weg für das Datagramm zum nächsten Router (next hop) kennen"

9-4)

Router erhält Datagramm mit unbekannter Ziel-NET-ID und hat keine Default Route:

Datagramm wird verworfen, ICMP Message mit „Destination unreachable : network unreachable“ wird an den Sender generiert

Router erhält o.g. Datagramm und kennt Default Route:

Datagramm wird entlang der Default Route weitergeleitet – der weitere „Weg“ des Datagramms am Netz ergibt sich durch die Routing Tabelle des nächsten Routers.

9-5)

Static-Routing : Eigenschaften

-sämtliche Einträge in den Routing Tabellen sind vom Admin vorkonfiguriert… komplexe Netzwerke mit Topologieänderungen können daher für den Admin einen sehr hohen Arbeitsaufwand verursachen

-für best. Netzwerktopologien ist static Routing gut geeignet: „Hub-and-Spoke“ und „Stub-Networks“ … dabei gibt es keine network redundancy, aufwendige Protokolle sind somit nicht nötig

-Performance ist gut, da keine zusätzlichen Ressourcen an Bandbreite (Routing msg. für dyn. Routing) und CPU Rechenzeit (ändern der Routing-Einträge) benötigt wird.

-bietet gewisse Security-Vorteile (durch statische Einträge in den Routing Tables)

9-6)

Verwendung von Statischen Routen bei Wegredundanz – es existiert nur ein Router für ein LAN auf einer Seite. „Hub-and-Spoke“ , „Stub“ - Networks

Folie 11+12

Bietet Security Vorteile

Statische Routen müssen/sollten verwendet werden, wenn die Netzwerktechnologie ansich kein dyn. Routing erlaubt…. Z.B. dial on demand (Netzwerkverbindungen sind nicht dauerhaft aktiv) (X.25, Frame Relay, ATM)

9-7)

-Die Einträge in den Routing-Tables der Router werden durch Routing Protokolle laufend auf dem neuesten Stand gehalten – Änderungen in der Topologie sind somit i.d.R. kein Problem. Router erkennen die Topologie selbstständig

-Der ständig laufende Algorithmus der Routing Protokolle verbraucht allerdings Bandbreite auf der Leitung (durch Routing msg.) und CPU Rechenzeit am Router (Änderung der Tables, Generierung und Weiterleitung von msg.)…

…

-Durch Routing Protokolle werden anhand von best. Informationen (hops, Bandbreite einer gewissen Leitung) die „best Paths“ zu bestimmten Netzwerken gespeichert – welche dann auch für IP-Datagramme an diese Netzwerke verwendet werden… oft leider auch, wenn der „best“ path überlastet ist („Autobahn-Stauprinzip“ von IP: Trotz Staus auf einer Autobahn (best path) wird diese weiterhin verwendet, obwohl eine Landstraße (alternate path) existiert, welche in dieser Situation die Daten viel schneller übermitteln könnte)

-2 verschiedene dyn. Routing Technologietypen: Distant Vector , Link State

9-8)

Siehe auch 9-7

-Sammeln von Informationen über die derzeitige Topologie und Updates im Falle von Änderungen.

-Ermitteln der „best paths“ am Netzwerk anhand statischer Daten wie z.B. Interface Cost, Delay, Bandbreite, hops … Update der best paths

-2 Technologietypen: Link State, Distant Vector

9-9)

Die Routing Metrik schlägt sich bei der Ermittlung des best paths zu best. Netzwerken nieder … unter der Metrik versteht man den hop-count … also die Anzahl der Router bis zum Zielnetzwerk.

Die meisten Parameter bei der Ermittlung des best paths sind statisch : Bandbreite einer Leitung, Interface Delay/Cost.

Durch dynamische Routing Protokolle können Fehlerfälle wie

-Open Links (Leitung tot) → alternativer Link

-Verschlechterung des aktuellen best paths → neuer best path

gut behandelt werden.

9-10)

Konvergenzzeit:
Änderung in der Netzwerktopologie → wann sind alle Routing Tables wieder auf dem neuesten Stand?

Je nach Netzwerkgröße sollte man unterschiedliche Routingprotokolle verwenden, da die Konvergenzzeit maßgebliche vom Routingprot. Abhängt.

RIP (Routing Information Prot) : für kleine Netzwerke, hohe Konvergenzzeit (Minuten)

OSPF (Open Shortest Path First): für mittlere – große Netzwerke, Konvergenzzeit im Sekundenbereich. …

… Wenn die Konvergenzzeit relativ hoch im Vergleich zum aktuellen Traffic ist, können Daten verloren gehen oder im Netzwerk „herumschwirren“ (Endless circling).

9-11)

Routing Loops entstehen durch fehlerhafte Einträge in Routing Tables. Durch Rooting Loops schwirren Daten (im Extremfall) endlos lang im Netzwerk umher, ohne je an ihrem Ziel anzukommen.

Kleines Beispiel für das Entstehen und die Effekte einer Routing Loop:

Router A erreicht Router C über Router B… Daten von A nach C nehmen also den Weg A-B-C

Plötzlich bricht die Verbindung B-C ab.. Router A bekommt davon nichts mit. Router A sendet die Daten für C weiterhin an B.

Router B weiß , dass C über den direkten Weg nicht mehr erreichbar ist, und versucht die Daten über A an C zu schicken. Router A hingegen, der die Daten bekommt und immer noch nicht über den Ausfall der Leitung bescheid weiß, sendet die Daten wieder zurück an B… Die Daten Kreisen also „endlos“ im Netzwerk umher.

Der Effekt ist offensichtlich: Durch Loops werden Buffer in den Routern belegt, was zu einem massiven Einbruch der Netzwerkperformance führen kann („Verstopfung“). Ältere Routingprotokolle haben keine ausreichenden Mechanismen um Routing Loops zu verhindern (z.B. RIP) – neue , komplexere Protokolle hingegen schon (z.B. OSPF, IS-IS)

9-12)

-Router senden periodisch Informationen über direkt angeschlossene andere Router und Netzwerke an ihre benachbarten Router (IP – Limited Broadcast)

-Empfänger dieser Informationen überprüfen diese auf Neuheiten im Vgl. mit ihrem bisherigen Routingtable und erneuern diesen gegebenenfalls.

-Jeder Router hat zwar genaue Informationen über seine Netzwerke und direkten Nachbarn, allerdings ist die (eventuelle) Topologie dahinter komplett unbekannt) – „Wegweiser Methode“

-Dadurch ergeben sich Probleme mit Routing Loops und Konvergenzzeit, gegen die zusätzliche Mechanismen wie max. Hop-Count, split horizon, poison reverse, hold down usw.. erforderlich sind

-Beispiele für Dist. Vector Prots: RIP, RIPv2 (Routing Info Prot) , IGRP (Interior Gateway Routing Prot), IPX RIP.

9-13)

Es werden lediglich Informationen aus Routing Updates dritter Router weitergegeben, wenn diese „bessere“ Wege zu bestimmten Netzwerken als die eigene Routingtabelle enthalten. – *Schlechte ‚umständliche‘* , Wege müssen den anderen Routern nicht mitgeteilt werden. Weiters werden nur Informationen über die eigenen angeschlossenen Netzwerke weitergegeben.

Der Hop Count für jedes Netzwerk wird bei der Weitergabe um 1 erhöht , damit der Wert aus der Sicht des anderen Routers richtig ist.

Router mit DV Protokollen kennen die direkt angeschlossenen Netzwerke und Router genau, wissen allerdings nicht über eventuelle Topologie dahinter bescheid – haben also eine relativ eingeschränkte Sichtweise.

9-14)

Durch die eingeschränkte Sichtweise der Router mit DV Protokollen ergeben sich Probleme bei

-Konvergenzzeit (relativ hoch, da jede Nachricht von Router zu Router weitergegeben, überprüft, modifiziert und erneut weitergegeben wird)

-Count to infinity, Routing loops entstehen, da Router nicht über Verbindungen und Netzwerke <u>hinter</u> ihren benachbarten Routern bescheid wissen.

Lösungsmechanismen:

Max. Hop count, Split horizon, poison reverse, hold down, triggered update, route poisoning

9-15)

-Bei Link State Protokollen hat jeder Router eine globale Sichtweise des Netzwerks – der jeweilige Router kennt die Verbindungen und „link costs“ zu Zielnetzwerken und verwendet diese Information um den „besten“ Weg dorthin zu berechnen (Shortest Path First Algorithmus)… nur dieser shortest path wird im Routing Table gespeichert

→ roadmap prinzip

-Veränderungen in der Topologie werden von den betroffenen Routern mittels LSA (Link State Advertisments) an alle anderen Router „geflooded“ – kontrollierter Multicast… dieser kann dann zu eventuellen Änderungen in den Routing Tabellen anderer Router führen

-Gibt es keine Veränderungen passieren auch keine floods, sondern nur kurze „hello“ messages, um die Verfügbarkeit der angeschlossenen Netzwerke zu überprüfen – erfordert wenig Bandbreite als die periodischen DV Updates

-treten jedoch Änderungen auf, kann der Overhead an Traffic zur Konvergenz mehr Bandbreite belegen als bei DV Protokollen

-Beispiele: OSPF (open shortest path first), IS-IS (Version für IP World, OSI world) , PNNI (ATM Netze)

9-16)

Ein Router-LSA enthält immer nur die **lokalen Nachbarschaftsbeziehungen** des Routers, der den LSA aussendet (IP Adresse, Maske, OSPF Kosten zu lokalen Stub Netzwerken, zu direkten Nachbar PTP Netzwerken und im Falle eines Transit Netzwerks die Adresse zu diesem).

Die ANDEREN Router, die ein solches LSA (mit einer Sequence No. gebrandmarkt), geben dieses **unverändert** weiter, wenn die Nachricht für sie neu ist (Hot-Potatoe). Kennt man die LSA bereits (Sequence No. Bekannt), wird die LSA nur bestätigt, aber nicht weitergeleitet. Das LSA muss in jedem Fall bestätigt werden.

Oft werden im Netzwerkbetrieb aber immer nur „Änderungen der derzeitigen Topologie“ (Link-State-Updates) weitergegeben, um den Traffic durch das Routing Protokoll am Netzwerk gering zu halten.

Link-State-Router kennen das gesamte Netzwerk, in welchem das Link-State Protokoll eingesetzt wird. Jeder Router hat durch die unveränderten LSA MSG. die gleiche „roadmap“ (Netzwerkkarte) der gesamten OSPF Domain.

9-17)

„Flooding“ von LSAs geschieht lediglich bei Topolgieänderungen in der Netzwerkumgebung. Kennt ein Router eine best. LSA Msg. schon, dann bestätigt er diese zwar, leitet sie aber nicht weiter (Hot-Potatoe).

Die Verbindungen (link states) zwischen den Routern und ihrer Umgebung werden regelmäßig mit „hello“ msg. überprüft (von den jeweiligen beteiligten Routern) → Treten keine Topologieänderungen auf, gibt es also nur sehr wenig Overhead im Netzwerktraffic durch das Link-State-Protokoll (Gute Eigenschaft)

Es tritt jedoch enormer overhead auf, wenn 2 (vorher getrennte) Netzwerkteile miteinander verbunden werden (Datenbasen in allen Routern müssen sich synchronisieren, bis jeder alle Topologie Infos hat)

9-18)

Bei RIPv1 wird prinzipiell der gesamte Routing Table eines Routers an seine Nachbarn ausgesendet.

Bei der Initialisierung eines Netzwerkes enthält der Routing Table eines Router nur die direkt angeschlossenen Netze… eine RIP Nachricht bei Initialisierung enthält also nur diese Information – nach einer gewissen Zeit (Konvergenz) hat jeder Router am Netz die Information seiner direkt angschlossenen Netze ausgesendet und alle anderen Router speichern den Weg zu einem best. Netzwerk anhand der besten Metrik (niedrigster Hop Count).
Router senden weiterhin ihre Routingtabellen (mit allen Einträgen) periodisch aus. Der Hop Count wird dabei automatisch um 1 erhöht, damit der Wert aus Sicht des Empfängers richtig ist andere Router führen Änderungen in ihren eigenen Tabellen durch, wenn sie „bessere“ Pfade als bisher bekannt erkennen-> durch 1 gespeicherten Pfad für ein Netzwerk in den Routern werden redundante Pfade umgangen. Updates sind erforderlich um die fortbestehende connecitivty zu überprüfen.

9-19)

Good News: Routing Updates enthalten für best. Netzwerke eine *bessere* Metrik als bisher bekannt → der Empfänger übernimmt dies sofort in seine eigne Routingtabelle (unabhängig vom Sender, Source Adress…. Good news wird einfach „geglaubt“ ohne auf den Sender zu achten… Trusted news)

Bad News: Router empfangen Informationen über Pfade für Netzwerke mit einer *schlechteren* Metrik wie bisher bekannt. Dies wird im Regelfall ignoriert (da man ja ohnehin selber einen besseren Pfad kennt), jedoch wird es nicht ignoriert, wenn der Sender dieses Updates gleichzeitig als nächster Hop für das entsprechende Netzwerk (welches nun schlechtere Metrik hat) ist.

9-20)

Durch das Count to infinity Problem bei RIP entstehen Routing Loops, welche die Netzwerkperformance stark belasten.

Beim RIP Prinzip gilt folgendes:

„good news“ – also Nachrichten über eine Verbesserung eines Pfades werden durch die periodischen Updates (30sec) schneller weitergegeben und gespeichert als bad news (Verschlechterung eines Pfades, link down) , welche erst nach mind. 180 sec weitergegeben werden.

Weiter „glauben“ RIP Router good news, unabhängig vom Sender
Bad News führen nur dann zu Veränderungen in der eigenen Routingtabelle, wenn der Sender der derzeitige eingetragene Next Hop zu dem betroffenen Netzwerk ist.

Dadurch tritt folgendes Szenario auf:

Folie 45-49

Netz 3 fällt aus – Router B akzeptiert die good news über den Hop-Count 2 von A sofort. Beim nächsten Update sendet er wiederum an A einen Hopcount von 3 bzgl Netz 3.

A akzeptiert diese Nachricht wiederum ebenfalls sofort, weil B der Vektor zum Netz 3 ist und trägt Hop count 3 ein. Diese Update sendet er an B(hop count 4), welcher A jetzt für den Vektor von Netz 3 hält und trägt 4 für Netz 3 ein…. Dadurch entsteht ein Routing loop.

Während dieser Zeit schwirren Daten für das Netz 3 endlos zwischen den beiden Routern umehr.

9-21)

Maximum Hop Count wird bei RIP eingesetzt um dem Count To Infinity Problem entgegenzuwirken:

-der maximale Wert für den Hop count in den Rootingtables beträgt nun 15… ein Router der einen Hop-Count Wert mit 16 aussendet signalisiert den anderen Routern nun, dass das entsprechende Netzwerk nicht mehr von ihm erreichbar ist.

-durch den max-hop-count Mechanismus alleine können Routing loops nicht verhindert werden.

-die Beschränkung auf max. Hop von 15 in der Tabelle schränkt auch die Topologie des Netzwerkes dramatisch ein… zwischen 2 Netzwerken dürfen dabei dann nur noch 16 hops liegen. (1 Hop zum ersten Router, 15 weitere zum entferntesten Netzwerk)

9-22)

Die Split Horizon Methode führt eine neue Regel für das Aussenden von Routing updates ein:

Router senden keine Informationen über bekannte Netzwerke in **die Richtung, aus welcher sie die Informationen erhalten haben**, außer sie kennen einen besseren Weg als bisher bekannt dorthin.

Folie 53+54

Sendet Router A Updates an Router B über das Netz 2, enthalten diese Updates keine Infos über Netz 2+3, sondern nur 1 – selbes gilt für Router B , wenn er über Netz 2 Router A ein update sendet (enthält nur Net 3 Infos).

-Gemeinsam mit Max Hop Count lassen sich temp. Routing loops einschränken (innerhalb von 180s), allerdings nicht ganz verhindern.

-Prinzip als Eselsbrücke: „Dont tell me, what i have told you“ ,sag mir nicht was ich dir schon gesagt habe…

9-23)

Durch Split Horizon/Poison Reverse + Max. Hop Count können Routing loops in kleineren Netzwerken weitgehend unterbunden werden.

Allerdings treten dieselben Probleme trotz der o.g. Methoden u.U. auch in größeren Netzwerken auf – deshalb verwendet man „Hold down“ als zusätzliche Methode.

-Router die Updates über Network failurse (Hop count =16) erhalten gehen diesbzgl. In einen „Hold down state“ und ignorieren für eine gewisse Zeit alle anderen Informationen bzgl. Dieses Netzwerkes. Dadurch erreicht man, dass **alle** Router über den Ausfall eines Netzwerkes bescheid wissen, ohne dass während der Konvergenzzeit Routing-Loops entstehen.

-Dafür erhöht sich die Konvergenzzeit allgemein, und Netzwerke mit Hold-Down arbeiten allgemein eventuell schlecht als Netzwerke ohne Hold-Down …

…

-Temporäre Rooting Loops lassen sich mit Split Horizon, Triggered Update, Max Hop count und Hold down in kleinen Netzwerken immer verhindern. Dafür sind die Konvergenzzeiten evtl. relativ hoch

9-24)

RIPv2 ist vom Prinzip her identisch mit v1 mit einigen Verbesserungen – ungenutzte Felder im RIP Header werden nun ausgenutzt

Wichtigste neue Features:

-unterstützt classless Routing (unabh. Von Klasse der IP Adresse)

-mehr Sicherheit durch authentication : Updates nurmehr von „trusted Routers“ akzeptiert

-RIPv2 benutzt class D / multicast Adressen anstatt Broadcasts wie bei RIPv1 – Updates werden also effizienter zugestellt (nicht jeder empfängt alle updates)

9-25)

-Beim Classful Routing enthalten Routing Updates keine genauen Informationen über die Subnetzmaske, d.h. in einem Netzwerk mit einer best. Klassenadresse (A,B,C) muss die Subnetzmaske im gesamten Bereich konstant sein (VSLM – variable length subnet mask kann **nicht** verwendet werden)

-deshalb wird an der „Klassengrenze“ , dem Ort an dem das „subgenettete“ Netz an die Außenwelt angeschlossen ist nur die Hauptklassenadresse bekannt gegeben – „route summarization on class boundries“

-deshalb ist es **nicht** möglich , dass Subnets, die zu einem bestimmten Netzwerk gehören „irgendwo“ angeschlossen werden… die subnettet area muss „conitguous“ (zusammenhängend) sein, da durch die route summarization an der Klassengrenze ein verteiltes Netz Infos für **andere** Subnetze erhalten würde (oder in manchen Fällen sogar gar keine)

9-26)

Routing Table Lookup ist der Vorgang des „Nachsehens und Weiterleitens“ in und anhand der Routingtabelle in einem Router

-als Routing Updates empfangene IP Adressen werden als „Hauptnetze“ interpretiert (Class A,B,C usw)

-bei einkommenden IP Datagrammen wird **zuerst die DA mit den eingetragenen Hauptnetzen verglichen** und verworfen, wenn unbekannt

-wenn die IP Adresse mit einem Hauptnetzeintrag übereinstimmt wird die Adresse mit allen eingetragenen Subnetzen verglichen und dementsprechend weitergeleitet, oder verworfen.

Folie 91 …

…

-beim nicht-zusammenhängenden Aufbau (discontiguous) von Subnetzen kann es daher passieren, dass jetzt eben der R3 im Bild nur 1 Update für das Netz 10.0.0.0 /16 in seine Tabelle einträgt und das andere angeschlossene Subnetz 10.2.0.0 entweder ganz ignoriert oder (wenn im Protokoll implementiert) dorthin Load balancing betreibt.

Dadurch werden Daten für das Netz 10.2.0.0 entweder ganz verworfen oder nur zum Teil übertragen.

9-27)

-beim classless routing arbeiten die Router mit zusätzlichen Informationen über Subnets in den Routing updates… in einem Routing Update kommt für die Adresse ein **prefix (relevanter Teil der IP Adresse) + „länge“ (=Teil der Adresse die als Subnet verwendet wird)** vor.

-dadurch ergeben sich mehrere Vorteile:

VSLM kann problemlos verwendet werden – die Subnets können von der Größe her der Anzahl ihrer IP Hosts angepasst werden → effiziente Adress Nutzung – durch classless Routing wird jedes Subnet richtig erreicht

Durch die 2 neuen Attribute für eine Adresse können Subnets nun auch discontiguous aufgebaut werden, da die „route summarization“ an einem beliebigen Ort stattfinden kann (nicht bei der Klassengrenze wie beim classfull routing). IP Subnetze können also an beliebigen Orten zusammengefasst werden.

9-28)

Routing Table Lookup ist der Vorgang des „Nachsehens und Weiterleitens“ in und anhand der Routingtabelle in einem Router

-Anders als beim classfull routing werden die IP Adressen nicht streng nach Klasse unterteilt

-die DA eines eingehend IP Datagramms wird Bit-für-Bit von „links nach rechts“ mit den Routing-Tabelle Einträgen verglichen

-Das Datagramm wird dorthin weitergeleitet wo der Vergleich die meiste Übereinstimmung brachte: „longest match routing rule“

-dadurch lassen sich Subnets unabhängig von der Netzwerktopologie aufbauen (discontiguous) – die Datagramme erreichen immer ihr Ziel (wenn tatsächlich im jeweiligen Subnet vorhanden)

9-29)

VLSM – Variable Length Subnet Mask

Ist eine Technik, die nur bei classless Routing eingesetzt werden kann

Innerhalb eines Netzwerkes mit einer gewissen Klassenadresse können dadurch Subnetze unterschiedlicher Länge existieren – je nach Anzahl der angeschlossenen IP Hosts … dadurch wird der vorhandene Adressraum effizienter ausgenutzt

9-30)

Beim Supernetting werden Subnets nach außen hin (z.B. sämtliche Subnets eines ISP ins Internet) unter einer Adresse mit Subnetzmaske zusammengefasst. Diese Subnetmask nach außen hin ist dann u.U. kleiner als die Subnetzmasken der einzellnen Adressen im angeschlossenen Netzwerk.

-ist natürlich nur für classless routing möglich

-Spart jede Menge Einträge für den Router, der an die „Außenwelt“ angeschlossen ist.

-Für Internet-Routing werden so auch wertvolle Adressen eingespart.

9-31)

IP-Adressen sind mit der Entwicklung des Internet zu einem „wertvollen Gut“ geworden.

Die von den Internet Core Routern (= ISP Router) verwendeten Adressen müssen also so sorgfältig wie es geht gehandhabt werden. Durch Supernetting kann man hinter einer „Supernet“ Adresse mehrere Netzwerke unterbringen – und spart somit Adressraum für weitere Internet Core Router ein.

Classless Routing allgemein bietet nun aber auch die Möglichkeit die Subnetze topologisch nicht zusammenhängend (discontiguous) aufzubauen – dafür würde man allerdings wieder mehr Einträge in den (Core) Routern benötigen und Supernetting könnte für ein discontiguous (Sub)Netzwerk nicht dermaßen viele Adressen einsparen wie Supernetting bei einem contiguous Netzwerk.

Deshalb sollten auch Subnetze Topologisch zusammenhängend (zumindest Richtung Internet) aufgebaut werden, um am Core Router Supernetting zu betreiben.

9-32)

CIDR – Classless Interdomain Routing

…ist eine Strategie zur Adressvergabe und Zusammenfassung von IP Adressen. CIDR wurde eingeführt um den IP Adressraum im Internet so effizient wie möglich auszunutzen, damit nicht eines Tages die verfügbaren Adressen ausgehen.

-classless routing

-supernetting

…

…

-Adressvergabe nach international bestimmten Richtlinien (z.B. best. IP Adressen für best. Kontinente; Zusammenhängender Block an IP Adressen für ISPs um route summarization in supernets zu ermöglichen)

Für Kunden (große Firmen mit vielen Netzwerken) bedeutet das aber auch, dass bei einem Wechsel zu einem anderen ISP sich alle IP Adressen (nach außen hin) ändern.

9-33)

Im Zuge von CIDR wurden 3 Adressblöcke für private Adressierung (z.B. Heimnetzwerke) festgelegt:

10.0.0.0 – 10.255.255.255 (10/8 prefix)

172.16.0.0 – 172.31.255.255 (172.16 /12 prefix)

192.168.0.0 – 192.168.255.255 (192.168 / 16 prefix)

Innerhalb privater Netzwerke können diese Adressen ohne weiteres verwendet werden. Möchte man mit solchen Adressen hingegen ins Internet gehen ist dies nicht möglich, da diese nicht eindeutig geroutet werden können. Die Übersetzung privater Adressen in globale einmalige Adressen geschieht durch NAT (Network Adress Translation)

9-34)

NAT – Network Adress Translation

Motivation:
-Verfahren um wertvolle IP Adressen effizienter auszunützen (durch private Adressen INNEN → globale IP Adressen AUßERHALB eines NAT Devices
-TCP load sharing
-Hide internal topology (security)

NAT Device:

Übersetzt (je nach Mapping statisch oder dynamisch) intern vergebene Adressen (aus dem privaten IP Adress Bereich) nach außen hin als globale (einmalige) IP Adressen.

Durch diese Änderung der IP Adresse muss auch die IP Checksum (und auch die TCP Checksum) für jedes Datagramm neu berechnet werden.

Statisches Mapping: einzelne interne IP Adressen sind im NAT Table des Devices explizit einer globalen IP Adresse zugeordnet

NAT Binding Tabellen mit zeitlichen Informationen werden im Falle von dynamischem Mapping benötigt… mehrere interne Hosts haben eine begrenzte Anzahl an globalen IP Adressen zur Verfügung → u.U. verwendet nicht jeder Host die Internetverbindung dauerhaft – spart teure IP Adressen nach außen ein.

9-35)

Dynamisches Mapping von NAT Tabellen:

mehrere interne Hosts haben eine begrenzte Anzahl an globalen IP Adressen zur Verfügung → u.U. verwendet nicht jeder Host die Internetverbindung dauerhaft – spart teure IP Adressen nach außen ein.

Möchte ein Host nach außen hin kommunizieren wird ihm (wenn verfügbar) im NAT Table eine globale Adresse zugeordnet. Zeitgleich wird diese temporäre Verbindung zwischen interer Host Adresse und globaler IP Adresse im NAT Table zeitlich überwacht… dem Host wird seine globale IP Adresse aberkannt, wenn er z.B. für einen gewissen Zeitraum die Verbindung nicht verwendet oder die Verbindung von sich aus beendet. Danach steht diese globale IP Adresse anderen Hosts wieder zur Verfügung. **Beim dynamischen Mapping muss zur Kommunikation eine TCP Verbindung von innen nach außen aufgebaut werden**

Verschiedene Szenarien für Datagramme von außen ohne Eintrag in der Binding Tabelle:

DA ist im NAT derzeit nicht vergeben : IP Datagramm wird verworfen

DA ist im NAT zwar vergeben, aber an ein anderes Device: Falsches Device empfängt IP Datagramm

DA ist dem NAT Device gänzlich unbekannt : IP Datagramm wird verworfen

9-36)

Virtuelle Server Tabellen werden bei NAPT (Network Adress Port Translation) verwendet, wo mehrere Hosts von innen über eine globale Adresse (mit unterschiedlichen TCP / UDP Ports nach außen) kommunizieren.

Möchte man einen Server HINTER einem Device erreichen geht das nur, wenn das NAT Device eine Zuordnung der Adresse kennt (derjenige von außen muss das NAT Device mit der richtigen Adresse ansprechen) – er kennt diese Adresse u.U. jedoch nicht, weil der Server IHN direkt noch nicht ansprechen konnte und das NAT Device noch kein Binding zuordnen konnte.

Deshalb müssen für diese Fälle solche Einträge manuell im NAT Device angelegt werden –> in einer Virtuellen Server Tabelle

---

Kapitel 10 Open Shortest Path First (OSPF)

9-37)

OSPF ist ein Link-State Routing Protokoll, welches die Nachteile von Distance Vector Routing Protokollen mit seiner komplett anderen Funktionsweise meidet.

-Jeder Router besitzt und updatet eine sog. „topology database, die das gesamte Netzwerk beschreibt (Network Roadmap)

-Dadurch kann für jede Verbindung in einem gegebenen Netzwerk mittels Djikstar Algorithmus der kürzeste Weg (geringste Kosten – beste Bitrate) errechnet werden.

-Durch LSAs (Link State Advertisment) Nachrichten halten die Router sich **untereinander** ständig über etwaige Netzwerkänderungen in der Topologie auf dem Laufenden.

9-38)

Der Router führt ausgehend von seiner Topology Database den Djikstra Algorithmus durch und erhält dadurch alle kürzesten Wege. Diese trägt er dann in seine Routing Tabelle ein.

Ablauf:

2 Bereiche vorhanden: „Selected“ , „Boundary“

Folie 13

1 Root auswählen (**selected**)

2 Nachbarn des Routers zur Boundary hinzu

3 Von der Boundary Router(V) mit lowest Cost (=kürzerster Weg dorthin) auswählen(**selected)**

4 Nachbarn von V zur boundary hinzufügen → für diese Kosten mit V als Vorgänger kalkulieren

5 mit Schritt 3 solange fortfahren, bis gesamtes Netzwerk abgedeckt ist.

Der Bereich „Selected“ wird in die Routingtabelle eingetragen.

Das Shortest Path First Verfahren ist ähnlich dem STP Algorithmus auf Layer 2 – der jeweilige OSPF Bridge entspricht dort dem „Root Bridge“ , der Tree entspricht dem „kürzesten Weg“ , der mittels Djikstra herausgefunden wurde (Wege vom Root → Designated Bridges)

9-39)

Link States bedeuten eine direkte Verbindung zwischen 2 Routern.
Ein Link State wird von 2 benachbarten Routern etabliert – und mittels LSAs den anderen Routern im Netzwerk weitergegeben. (andere LSAs über andere Link States werden empfangen) → generiert Topology Database.

Ein Link State wird dann auf UP gesetzt, wenn die Start-Up Prozedur zwischen 2 Routern für deren direkte Verbindung zueinander durchlaufen ist („hello“ – „Database descr.“ – „LS request“ – „LS update“ – „LS ack“ ….. in beide Richtungen)

Link States werden alle 30 Minuten aktualisiert, alle 60 min. werden sie verworfen und müssen neu initialisiert werden.

9-40)

Wird ein Router an ein Netzwerk angehängt sendet er „hello msg.“ und empfängt „hello msg.“ von direkt benachbarten Routern.

R1 an Netz angehängt, erkennt R2 als direkten Nachbarn (R2 erkennt auch R1) mittels hello msg.

R1 sendet R2 seine bekannte Database – database description

R2 vergleicht die Nachricht mit seiner database und sendet evtl eine Anfrage auf genauere Daten – LS request

R1 antwortet mit LS Update

R2 wiederum mit LS ACK.

Danach passiert derselbe Prozess in Richtung R2→R1 (database descr. Usw..)

Zuletzt senden beide Router LSA UPDATES an ihre benachbarten Router, um den neuen Link State mitzuteilen (Broadcast)

9-41)

Änderungen in der Topologie werden mittels LS Updates (Broadcasts, „Flooding“) mitgeteilt.

LSA werden immer nur von Routern ausgesendet, welche eine Veränderung bemerkt haben, bzw. jene die eine Änderung in ihren Topology databases getätigt haben.

LSAs bewirken bei andern OSPF Routern evtl. eine Änderung in deren Datenbasen, die Empfänger von LSAs müssen immer mit ACKs darauf antworten … tun sie das nicht (regelmäßig) wird der Link-State zwischen den jeweiligen 2 Routern DOWN gesetzt.

9-42)

LSAs werden von den jeweiligen Routern an alle Nachbarn gebroadastet (LSA Broad, Flooding)

In der LSA steht neben der Information über die lokalen Verbindungen des Routers eine Sequence Number zur Unterscheidung der jeweiligen LSAs. Sobald ein Router eine LSA empfängt, bestätigt er diese explizit und leitet sie an seine benachbarten Router weiter. Kennt ein Router beim Empfang die LSA Sequence Number bereits, bestätigt er zwar, leitet aber anschließend **nicht weiter** (er kennt das Update in dem Fall schon)

Die LSA Sequence Number ist erforderlich , um LSAs voneinander unterscheiden zu können (zusätzlich wird noch jedem Router eine einmalige ID zugeordnet).

Ein LSA Refresh für einen Link State hat eine neue Sequence Number. (damit dieser auch von allen weitergeleitet wird)

9-43) Danke an „tipi“ aus dem ET Forum

-hello
-database descr.
-LS RQ
-LS Update (Router LSA, Network LSA)
-LS Ack

Die LSAs werden über IP (L3) transportiert.

Ein ACK ist erforderlich, damit sichergestellt wird, dass jeder die LSA erhalten hat.

Adressierung

Alle OSPF fähigen Router bekommen die IP Adresse 224.0.0.5, im shared Media mit DR und BR Routern bekommen diese 224.0.0.6. DR und BR werden Netzwerkinitialisierung bestimmt.

224.0.0.5 wird als Zieladresse von ALLEN OSPF Routern verwendet für „hello Msgs“

224.0.0.6 wird als Zieladresse nur für die BR und DR verwendet – für LSA Datenverkehr

L2 – MAC Multicast

9-44)

Bei Shared Media Netzwerken treten durch die ständigen hello und LSA Messages der Router hohe overheads auf… any-to-any Verbindungen hätten viel zu viele redundante Daten.

Deshalb kümmert sich (NUR FÜR DAS ROUTING PROTOKOLL, nicht für IP Forwarding) ein sog. Designated Router (DR) für die Verteilung und Bestätigung aller LSAs , welcher zur Unterstützung einen Backup Router (BR) hat.

9-45)

Der DR ist dafür verantwortlich die (im shared media) virtuellen ptp Verbindungen zu den Routern zu managen und Verteilen (hello, database descr, ls update, ls rq, ls ack …) – *transit network*

Er ist weiters dafür verantwortlich sog. Network LSAs auszusenden, mit welchen er den anderen Routern den Zustand und ihre Zugehörigkeit zum Broadcast Network mitteilt.

Der BR überwacht den Verkehr des DR und übernimmt dessen Rolle, sobald er merkt, dass der DR ausfällt (fehlende ACKs des DR als Indikator)

Der DR übernimmt sämtliche Aufgaben für das Routingprotokoll, das hat jedoch KEINEN Einfluss darauf wie IP Datagramme Weitergeleitet werden (IP Forwarding) – dies wird in den Routing Tables bestimmt.

9-46)

- Bandbreite durch Routing Prot Nachrichten ist bei OSPF deutlich geringer als bei RIP (hello msg, bzw. LSA wo nur Änderungen beschrieben werden im Gegensatz zu kompletten Routing Tabellen bei RIP)

-Größere Netzwerke als bei RIP möglich

-SPF Algorithmus hat eine Fülle von Vorteilen:

Jeder Router weiß über das komplette Netzwerk bescheid
Dadurch kann jeder Router für sich den idealen Weg bestimmen → schnelle Konvergenz
LSAs werden nicht modifiziert, wenn sie über das Netzwerk gesendet werden
LSAs hängen nicht von Subnets ab

-Bester Weg wird durch Port Costs bestimmt (Bandbreite,Delay) und nicht „nur“ durch den hop count

-VLSM möglich

-externe Verbindungen werden explizit angegeben

-Security durch authentizierte OSPF Msg.

-OSPF Nachrichten werden als Multicast gesendet → erreichen Schneller ihr Ziel

-PTP Verbindungen brauchen keine eigenen IP Adressen – damit werden IP Adressen eingespart

9-47)

RIP wird durch Max Hop Count stark beschränkt. (wegen Routing loops)

OSPF stößt an seine Grenzen, wenn große Netzwerkteile miteinander Verbunden werden (sehr lange Konvergenzzeiten zur Datenbasissynchro)

→ Deshalb sind beide eher ungeeignet für Internet Routing.

9-48)

Beim Internet Routing wählt man folgendes Konzept:

Eine Ansammlung von Netzwerken wird als „Autonomous System“ (AS) bezeichnet. Innerhalb dieses AS wird ein Interior Gateway Protokoll zum Routen verwendet (RIPv2, OSPF, IGRP usw..) ZWISCHEN diesen AS wird ein EGP (Exterrior Gateway Prot) zum Routen verwendet (EGP-2, BGP-3, BGP-4).

Ein EGP Router weiß, über viele AS bescheid… EGP Router sind sozusagen das Herzstück des Internets.

Jedes AS ist mit einer einmaligen Nummer gekennzeichnet – die ähnlich den Internetadressen bei einer zentralen Stelle vergeben werden.

9-49) ***Danke an „Tipi“ und „Etolkri“ aus dem ET-Forum!***

BGP 4 ist das aktuelle EGP (Exterior Gateway Protokoll), welches zzT. für das Internet verwendet wird. BGP Router sind an den „Grenzen“ von AS (Autonomous Systems) platziert und tauschen in sog. BGP-Sessions Informationen über die Erreichbarkeit von Netzwerken aus.

Eine solche Session wird als **External BGP Session** bezeichnet, wenn 2 BGP Router aus verschiedenen AS Informationen austauschen.

Internal BGP Session ist der Informationsaustausch zwischen 2 BGP Routern, die an ein und dasselbe AS angeschlossen sind.

Diese Sessions werden über TCP abgehalten – das TCP Protokoll kümmert sich dabei um das CO Service für die Übertragung der Information. (well known port 179)

Beim Austausch der Routing-Infos in Sessions wird der jeweiligen Nachricht die ID von **jenen AS hinzugefügt, durch welche die Nachricht läuft** – damit können BGP Router Graphen von AS konstruieren (verhinder routing-loops, weitere Topologie unterhalb ist irrelevant) → **BGP ist ein path Vector Prot.** – der „beste pfad“ ergibt sich durch die AS-hop-number.

Nach einmaliger Konstruktion des Graphen an einem Router, werden in weiterer Folge in den BGP Sessions nurmehr **Änderungen** der Netzwerktopologie übertragen (ähnlich OSPF). Die **einzelnen Routen zu Netzwerken** werden durch viele verschiedene Attribute beschrieben (AS Hop count ist…

…lediglich eines davon) – somit kann man beim BGP Protokoll **verschiedene Routing Policies („Regeln“) aufstellen** nach denen der Datenverkehr geroutet werden soll.

-unterstützt VSLM

-unterstützt CIDR (BGP-4 ist daher classless) und supernetting

---

Kapitel 11 Internet Transport Layer (TCP,UDP)

10-1)

Transport Control Protocol (TCP)

Connection Oriented Layer 4 Protokoll… Daten werden von höheren Layern in sog. TCP Segments eingebettet, welche wiederum in IP Datagramme eingebettet werden. Fragmentation für die Segments ist möglich

-ARQ Technik: Continuous RQ mit piggy backed ACKs

-Error Recovery mittels sequence Numbers .. jedes Segment kann mit positive&multiple ACKs bestätigt werden, jedes Segment initiiert einen Timer (Timeout – Mechanismus)

-Flow Controll mittels Sliding Window und dynamisch änderbarer Window Größe

-Three way handshake beim Verbindungsaufbau zwischen Server und Client.

-TCP kapselt die Funktionen der unteren 3 Layer für alle Protokolle höherer Layer aus, sodass diese (L5,6,7 Protokolle) sich nicht darum kümmern müssen.

10-2)

TCP ist ein L4 Connection Oriented P(acket) S(witching) Protokoll

Error Recovery durch ARQ (automatic Repeat Request) → Continuous RQ → Positive ACK (siehe 2-10, 2-11) mit Piggy backed ACK … d.h. der Datenstrom enthält ACKs, diese werden nicht jedesmal separat gesendet, multiple ACK.

Flow Control durch Sliding Window (2-17 ff), Größe des Fensters kann vom Empfänger automatisch angepasst werden, weiters: slow start congestion avoidance (Sender verwendet am Beginn einer Übertragung nicht gleich die größtmögliche Bitrate, sondern steigert sich langsam).

10-3)

Client ist beim TCP Protokoll immer derjenige, der eine Anfrage an den Server sendet, bzw. die Verbindung als erster initialisiert.

Server ist beim TCP Prot. immer derjenige der auf eine Verbindungsanfrage antwortet… so kann es sein, dass „echte“ Server selbst zu Clients werden, wenn sie mit anderen Servern kommunizieren…

TCP als L4 Protokoll stellt Dienste für höhere Protokolle zur Verfügung – dies geschieht durch die sog. TCP Ports… so können **verschiedene Kommunikationsprozesse parallel auf unterschiedlichen TCP Ports abgehandelt werden** (z.B. Browsen, Chatten, usw..) Man kann sich TCP also als eine Art Multiplexer für die Applications Software der höheren Layer vorstellen (*Folie 9*)

Die sog. Well-known Port Numbers sind „standardisierte“ Ports – die eine festgelegte Funktion haben. (z.B. Port 80 wird immer für http verwendet). Clients, welche eine Verbindung zum Server aufbauen möchten, müssen bei einer well-known Port number anfangen, da sie ja nicht wissen können, welche TCP Ports am Server bereits belegt sind.

Der Verbindungsaufbau (im Zusammenhang mit Ports) funktioniert also folgendermaßen:

Client bestimmt einen **seiner TCP Ports als Source Port**
Er baut die Verbindung zum Server mit einem **well known Port als Dest. Port auf**
der Server weist dem Client evtl. eine andere Port No. für die weitere Verbindung zu.

10-4)

Oftmals erhält ein Server über seine well known Ports viele Anfragen auf einmal… er daher zwischen den einzelnen Verbindungen zu einem Port unterscheiden können (Sys A→ Server auf Port 80 und SysB → Server auf Port 80)

Die Kombination aus IP Adresse und Port wird als Socket für eine Verbindung bezeichnet.

z.B:
Server hat 11.0.0.1
- ES A hat 12.0.0.3 macht Verbindung von Port 1234 auf Port 80 des Servers
-ES B hat 13.0.0.6 macht Verbindung von Port 544 auf Port 80 des Servers

Socket 1:

Port 80, 11.0.0.1
Port 1234, 12.0.0.3 beschreibt Verbindung von A(Port 1234)→Server (Port80)

…

…
Socket2:

Port 80, 11.0.0.1
Port544, 13.0.0.6 beschreibt Verbindung von B(Port 544) → Server (Port 80)

Ein Socket Paar beschreibt also **EINDEUTIG** eine Verbindung.

<u>10-5)</u>

Der Verbindungsaufbau bei TCP erfolgt durch den sog. „Three Way Handshake“ Prozess.

Er stellt sicher, dass TCP Segmente aus älteren Verbindungen zwischen den Systemen die neue Verbindung nicht stören.

Folie 24

<u>Ablauf:</u>

1
Client baut Verbindung auf… wählt eine Zufallszahl aus dem Bereich 2^32 – sendet diese als Sequence No. und **gesetztem SYN Flag** an den Server.

2
Server empfängt Frame, trägt Seq. No. +1 in sein ACK Feld ein, generiert eine Sequence Zufallszahl und sendet diese im Sequence No. Feld gemeinsam mit der Sequence No. des Clients im ACK feld bei **gesetztem SYN und ACK Flag** zurrück an den client

3
Client empfängt Frame, kontrolliert ACK, trägt Sequence No. In sein ACK Feld ein → Verbindung aufgebaut.

Ack No. Und Sequence No. Sind jeweils 4 Byte (32 Bit) lang und werden beim 3-Way-Handshake mit einem guten Verfahren als Zufallszahl generiert – dadurch können Störungen durch Frames aus alten TCP Verbindungen (selber Socket) eingedemmt werden.

Reale Bsp für ACK und SEQ number:

Zufallszahl im Bereich 0 bis 4.2*10^9 (Dezimal)

10-6)

Siehe auch 10-5

Der Bereich ist deshalb so groß, damit Störungen durch alte Verbindungen (und deren Sequ. No.) noch weiter umgangen werden können…. Die Zufallszahlen können stark variieren.

Weiters muss durch diesen hohen Bereich selten ein sog. Wrap-around erfolgen – falls ACK oder SEQ 2^31-1 überschreiten – dadurch können durch die kurze Lebensdauer der IP Datagramme auch selten alte Pakete in neuen Verbindungen auftreten.

10-7)

Beim alten TCP wurde das Positive ACK Verhalten genau angenommen:

Frames so lange mit ACK bestätigen, wie sie in richtiger Reihenfolge eintreffen

Falls ACK verloren geht, jedoch weitere Frames ankommen (in Reihenfolge) → Weitere ACK senden… Sender sendet durch diesen Multiple ACK (für mehrere Frames) weiterhin Daten

Falls Frames gestört werden keine ACK senden, warten bis Timeout beim Sender eintritt und dieser den Frame wiederholt….Daten allerdings weiter empfangen und dann (bei Eintreffen des fehlenden Frames → multiple ACK) **Genau hier geht beim alten TCP viel Performance verloren, da der Sender nicht genau weiß was los ist, wenn er plötzlich keine ACK mehr bekommt.** … es könnte sein , dass die Verbindung komplett gestört ist, und überhaupt keine Frames mehr am Empfänger ankommen ODER, dass nur das EINE Frame gestört war, und der Empfänger weiterhin gut Daten empfängt…. Der Sender weiß es durch den alten TCP Mechanismus (ohne duplicate ACKs) aber nicht.

10-8)

Bei der neuen Error Recovery wurde ein wichtiges Detail geändert:

Es werden erst dann überhaupt keine ACKs mehr gesendet, wenn auch wirklich überhaupt keine Daten empfangen werden. Fehlt EIN Frame (z.B. 22), dann wird weiterhin bei Empfang jedes Frames ein ACK (21) gesendet – so kann der Sender viel schneller erkennen, dass etwas nicht passt und darauf reagieren (früher: Timeout abwarten… was stark auf die Performance geht)

10-9)

Schließen von Half-Sessions – durch die Full Duplex Mode Beschaffenheit der Verbindungen wird zuerst eine Verbindung A→B und danach die andere B→A geschlossen.

Im Prinzip ist der Verbindungsabbau sehr ähnlich dem Three Way Handshake:

A (möchte Verbindung beenden): sendet aktuelle Sequ. No. und Ack. der zuletzt empfangenen Daten mit gesetztem ACK und FIN Flag. …

…

B sendet Sequence No. Von A + 1 im ACK Feld mit gesetztem ACK Flag → Verbindung von A-B beendet.

<Selber Vorgang von B→A> schließt die Verbindung auf der anderen Seite → Verbindung ganz geschlossen.

Dadurch wird sichergestellt, dass beide Systeme alle Daten erhalten haben.

<u>10-10)</u>

Bsp durchgeführt: weiters wird ein Wert für die Strecke B→ A mit „x“ angenommen.

A→B

Werte A(vor dem Senden): Seq=250, Ack = x
Werte B(vor dem Empfangen): Seq=x, Ack = 250

A sendet Block 1 an B mit Seq:250, 550Byte , ACK x

Werte A(nach Block1 gesendet): Sequ = 800, Ack =x
Werte B(nach Block1 empfangen): Sequ: x , ACK = 800

B sendet A leeren Block mit Sequ=x, Ack=800

A empfängt Block und vergleicht Werte… davor sendet A evtl. schon Block 2

Werte A(vor dem Senden Block2): Sequ= 800, Ack=x
Werte B (vor dem Empfangen von Block2): Sequ=x, Ack=800

A sendet Block 2 an B mit Seq:800, 340Byte , ACK x

Werte A(nach Block2 gesendet): Seq = 1140, Ack=x
Werte B(nach block 2 empfangen) Seq= x, Ack= 1140
B sendet A leeren Block mit Seq=x, Ack=1140

Die Sequence Number gibt also genau an, welches Byte zuletzt gesendet wurde (vor der neuen Übertragung)

Die ACK Numer gibt genau an, welches Byte zuletzt empfangen wurde.

10-11)

Timeouts sind als Indikator für nicht korrekt empfangene Frames zur Retransmission notwendig. (conn. Oriented service)

Der Wert des Timeouts für ein gesendetes Frame bestimmt die Performance maßgeblich – das Timeout muss bei jeder Verbindung angepasst werden. Ist es zu hoch, führt das zu langen idle Phasen, ist es zu niedrig treten timouts oft unnötig auf.

→ deshalb werden die Werte für ein Frame-Timeout für jede Verbindung mittels KARN Algorithmus neu bestimmt.

10-12)

Die TCP Checksum bezieht den kompletten Header und Datenbereich des TCP Segements mit ein – sowie Bereiche des IP Headers (IP adress, Prot type, segment length) – „pseudo IP Header“

Dadurch erfolgt eine Error detection nicht nur über den Port… sondern für das gesamte Socket der Verbindung.. TCP kann dadurch auch Fehler erkennen, die IP nicht erkennen kann (IP Übertragung eines „korrekten“ TCP Segments an einen falschen IP Host)

10-13)

Die Länge eines TCP Segements wird über die MSS (Max Segment Size) Option im TCP Options Feld bestimmt. Darin erklärt ein Sender dem Empfänger, wieviel (**Nutzdaten**) der Empfänger dem Sender in EINEM TCP SEGMENT zurücksenden kann.

Der MSS Wert wird als Basis für das Congestion Window verwendet.

Ein TCP Segment kann mit Maximal möglicher MSS ~ 65Kbyte groß sein.

Die Länge des TCP Datenstroms wird maßgeblich über das Window Feld bestimmt (+congestion avoidance zu Beginn)

Die Größe lässt sich weiter durch Window Scaling erhöhen.

10-14)

Flow Control bei TCP wird mittels Sliding Window und dynamisch regelbarer Windowgröße realisiert.

Der Empfänger signalisiert dem Sender die Windowgröße mittels Window Feld im TCP Header.

Die Stelle des ACK-1 gibt die „linke Kante“ des Fensters an (bis dahin wurden Daten gesendet und akzeptiert)… ACK-1+W bestimmt die rechte Kante, bis zu welcher der Sender Daten senden darf. Im Fenster selbst sind u.U. schon einige Daten gesendet, aber noch nicht akzeptiert… der „Rest“ an „sendbaren“ Daten wird als useable Window bezeichnet. …

…

Falls der Empfänger mit dem eingehenden Datenstrom überfordert ist, sendet er ACK und bestimmt somit die linke Kante des Fensters, und sendet weiters einen Wert W, womit er die rechte Kante des Fensters bestimmt – und dieses so komplett schließen kann…. Mit W=0 + Ack für letztes empfangene Byte. Bzw. wieder öffnen kann.

Siehe Folie 40,42,43,44

<u>10-15)</u>

Das URG(urgent =dringend) Flag ermöglicht mit dem Urgent Pointer eine Möglichkeit wichtige Daten schnell zu übertragen.

Ist URG Gesetzt, gibt Sequence Number+ Urgent Pointer das **letzte** gesendete Urgent Byte an (für Bestätigung der Urgent Daten)

URG versetzt die darüberliegende SW evtl auch in einen „Urgent-Mode“

PSH – falls gesetzt wird ein TCP Segment sofort an den nächst-höheren Layer weitergeleitet ohne vorher zwischengespeichert zu werden. –> wird aber heutzutage oft nicht verwendet.

<u>10-16)</u>

Die „delayed ACK“ Funktion bei TCP versucht das Senden von unnötig vielen Daten zu vermeiden. Normalerweise würde ja nach jedem empfangenen Segment **sofort** ein ACK ausgesendet werden – unabh. davon ob dieses ACK im gesendeten Segment Daten enthält oder nicht … es würden dadurch also viele Segmente entstehen, die eigentlich gar keine Daten sondern nur ACKs enthalten.

TCP verwendet piggy-backed ACKs… d.h. Datenblöcke, die ACKs für vorherige Daten enthalten. Deshalb wartet TCP nach korrektem Empfang eines Segments 200ms um **erst dann** ein delayed ACK auszusenden (während dieser Zeit kann die App.SW im höheren Layer evtl. noch Daten in das TCP Segment „hineinpacken“)

<u>10-17)</u>

Der „Nagle Algorithmus“ versucht das ständige Senden sehr kleiner Datenblöcke (sog. Tinygrams) zu unterbinden, da es sonst zu Netzwerküberlastungen kommen könnte.

Vorgehensweis:

Wenn das TCP eines Endsystems auf ein **ACK wartet, sendet es solange keine tinygrams aus, bis das ACK ankommt.** In der Zwischenzeit werden die Tinygrams gesammelt und **nach dem ACK in einem Segment ausgesendet.**

10-18)

Ungenützte Ressourcen / Performanceverlust ohne „slow start congestion avoidance“ :

Segmente gehen verloren, Sender kann nur noch das ihm zur Verfügung gestellte Window ausnützen (rechte Kante wandert wegen fehlender fortlaufender ACKs nicht mehr weiter) und muss dann auf sein retransmission Timeout warten – während dieser Zeit bleibt Bandbreite am Netzwerk ungenützt und TCP liefert nicht die optimale Performance

Die ersten Überlegungen, die zu dem neuen Flow-Controll Verfahren führten betrafen die „Herkunft“ solcher Packetverluste:

Damals nahm man an (und es stellte sich im Nachhinein auch als ca. richtig heraus) , dass Segmentverluste in den Netzwerken hauptsächlich durch Überlastungen (Flaschenhalseffekt, wenn Router Queues haben usw.) **und nicht durch Bitfehler** auftreten.

Um die Verfügbare Bandbreite also optimal ausnutzen zu können erfand man den „slow start congestion avoidance algorithm“ – eine quasi **senderseitige Flow Controll.**

10-19)

Die Grundidee ist, eine senderseitige Flow Controll zu implementieren, um die am Netzwerk vorhandene Bandbreite effizient ausnutzen zu können. Der Sender soll Datenstaus (congestion) frühzeitig erkennen und seine Sende-Bitrate entsprechend drosseln.

Zum Erkennen von „leichten“ Datenstaus wird das **Auftreten von Duplicate ACKs** (Empfänger fehlt ein Packet, sendet aber weiterhin ACKs für jeden korrekten Empfang , mit dem Wert des letzten Packets in richtiger Reihenfolge) berücksichtigt – die Bitrate wird **etwas gedrosselt**. (congestion Window =ssthress = ½ current window size)

Schwere Staus erkennt ein TCP Sender beim Auftreten von Timeouts von Segmenten – dann drosselt er die **Bitrate stark** (Congestion Window = 1 , slow start)

Congestion Window und ssthresh sind für slow start und congestion avoidance neue Parameter um die Flow Control zu ermöglichen.

10-20)

Congestion Window : cwnd
slow start threshold: ssthresh

<u>Congesiton Window: „SLOW START“</u>

Die Anzahl wird vom Sender aus bestimmt (gemessen in Segmenten)
Die Größe EINES congestion Windows (=1 Segment) wurd durch die MSS (max. segment size) des Empfängers in den TCP Options übermittelt

…

…

Zu Beginn der Übertragung sendet der Sender nur 1 cwnd. Für jedes weitere ACK erhöht er cwnd um 1 – dadurch Verdoppelt sich das cwnd pro RTT (round trip time), bis eine congestion auftritt.
Oftmals „wächst“ cwnd auch nur linear… siehe weiter unten.
Congestion tritt auf →
-Bei duplicate ACKs wird cwnd = ssthresh (Hälfte der derzeitigen window Größe) gesetzt,
-Bei timeout wird cwnd wieder auf 1 zurrück gesetzt.

<u>Slow start threshhold: „CONGESTION AVOIDANCE“</u>

Um congestion Avoidance wirklich gewährleisten zu können wurde eine weitere Variable eingeführt: ssthresh. Wie oben erwähnt wird cwnd auf ssthresh gesetzt, wenn ein leichter Stau detektiert wird. Davor wurde ssthresh allerdings auf ½ der vom Empfänger bestimmten Window Size gesetzt… so ist der Sender nach einem leichten Stau wieder garantiert bei einer sicheren Bitrate (ohne mögliche Staus) - und muss dafür nicht wieder bei 0 anfangen.

Weiters bestimmt ssthresh das WACHSTUM von cwnd.
Ist cwnd < ssthresh wächst es pro ACK um 1 an – **exponentielles Wachstum**
Ist cwnd > ssthresh **wächst** es pro ACK **linear** an mit MSS^2/cwnd

<u>10-21)</u>

Für die Datenrate einer Session zwischen A→ B stellt sich ein „wellenförmiges“ Bild über die Zeit dar – jedes Mal wenn ein duplicate ACK auftritt, wird die Bitrate leicht gedrosselt; bei einem Timeout stark. – ergibt Sägezahnförmiges Y/T Diagramm

Die Performance wird dadurch bestimmt, wie gut der Kanal ausgenutzt werden kann – dazu sollte die advertised Window size vom Empfänger **mindestens** dem Delay-Bandwith-Product entsprechen, da sonst diese Ressourcen einfach verschwendet werden.

<u>10-22)</u>

MSS = Maximum Segment Size , wobei die Nutzdaten (ohne Header) eines TCP Segments gemeint sind. Der Wert wird im Options Feld des TCP Headers konfiguriert.

Max possible MSS ist 2^16 Byte.

MSS bestimmt den Wert für EIN cwnd. (=Größe eines congestion windows)
MSS bestimmt den Startwert für ssthresh, bevor ssthresh nach der ersten congestion auf die halbe derzeitige window size reduziert wird.

10-23)

Durch Fast Retransmit und Fast Recovery wird der slow-start-congestion-avoidance Mechanismus weiter verbessert.

Duplicate ACKs signalisieren dem Sender normalerweise einen leichten Stau – bzw. Packet loss… oftmals ist es aber so, dass durch rerouting die Packete nicht verloren gehen, sondern „nur“ in der falschen Reihenfolge am Empfänger ankommen – deshalb enstehen in der Regel 1-2 duplicate ACKs, bis die Daten wieder alle in Reihenfolge sind, und der Empfänger wieder „richtig“ acknowledged.

Diesen Erfahrungswert implementierte man ins TCP : Der Sender wartet 2 Duplicate ACKs ab und sendet erst beim **dritten** das geforderte Frame erneut → fast retransmit

Dadurch wird automatisch die fast recovery Prozedur eingeleitet: ssthresh wird auf ½ der derzeitigen Windowgröße gesetzt ; cwnd wird auf ssthresh+3*segmentgröße gesetzt, da man davon ausgehen kann, dass der Empfänger nach 3 duplicate ACKs schon 3 zusätzliche Segmente empfangen hat.

Fast Recovery liefert damit congestion avoidance und verhindert gleichzeitig einen slow start – da das Netzwerk aufgrund der empfangenen duplicate ACKs noch in Ordnung zu sein scheint.

… dh. auch nach dem ersten +zweiten duplicate ACK wird Cwnd um 1segmentgröße erhöht und erst beim 3. gedrosselt.

10-24)

Siehe 10-23

10-25)

Durch Window Scale kann man das mögliche Sendefenster weiter als MSS vergrößern (bis zu 2^30 Byte) – dadurch lassen sich bessere Verbindungsstrecken effizienter ausnützen

Durch SACK Option erlaubt man dem Empfänger selective ACK zurrückzusenden, auch wenn Lücken im empfangenen Datenstrom vorhanden sind.

10-26)

-User Datagram Prot = L4 CL PS Service – zu Layer3 kommt lediglich Port adressing und eine checksum dafür hinzu

-benützt selben Portnummern wie TCP

-weniger komplex, und leichter zu implementieren

Anwendungsgebiete:

-überall dort, wo zusätzliche Daten eines CO Service unerwünscht sind (z.B. short DNS)

-überall dort, wo ein CO Protokoll keinen Sinn ergibt (Voice over IP – Audiodaten neu zu senden ergibt keinen Sinn)

-überall dort, wo die Endsysteme wenig Ressourcen benötigen sollten.

ENDE