Zusammenfassung - Advanced Communications

Marc Meier, CD

9. Januar 2016

Korrektheit und Vollständigkeit der Informationen sind nicht gewährleistet. Macht euch eigene Notizen oder ergänzt/korrigiert meine Ausführungen!

Inhaltsverzeichnis

1 Grundlagen

1.1 Grundprinzipien und Entwicklung des Internets

Das Internet entwickelte sich ab den 1960er Jahren. Es ging aus dem am Ende des Jahrzehnts entstandenen, vornehmlich militärisch und akademisch geprägten ARPA-Net hervor. Heutzutage wird es international kommerziell, industriell und auch akademisch (Katzenbilder) genutzt. Bei seiner Entstehung war vor allem eine dezentrale Struktur ohne zentrale Verwaltung von Interesse. Grund hierfür war die Angst des amerikanischen Department of Defense, dass eine atomarer Angriff zentrale Kommunikationspunkte außer Kraft setzen könnte. Die Kommunikation findet über hochgradig vernetzte Knoten mithilfe von Paketen statt.

Literatur: [1, 3]

1.2 Packet Switching

Paketvermittelte Übertragung bedeutet die Abkehr von der leitungsbasierten Vermittlung. Dabei werden längere Nachrichten in Datenpakete aufgeteilt und voneinander unabhängig versendet. Dies ermöglicht eine faire Verteilung der Leistungskapazität und redundante Wege bei einem Ausfall von Knoten oder Verbindungen. Im Gegenzug können konstante Bandbreiten nicht ohne Weiteres (Abschnitt 8) gewährleistet werden, ebenso ergeben sich unterschiedliche Laufzeiten von Paketen.

1.3 Dezentrale Verwaltung des Internets

1.3.1 Prinzipien

- Keine zentrale Verwaltung oder Behörde (trotz Einflussnahme)
- Demokratisches Zusammenwirken der Beteiligten / Wahlen
- Selbstorganisation
- Standards dort, wo sie erforderlich sind
- Dynamisch, offen für Neuigkeiten

1.3.2 Organisationen

ICANN: Vergibt IP-Adressen und betreibt die DNS-Rootserver.

IETF: Standardisierung von Protokollen in RFCs [38]

RIPE: Administration und technische Koordination

RIPE NCC: Adressvergabe in Europa und Zentralasien, Verwaltung der WHOIS-Datenbank.

DENIC eG: Domain-Verwaltung für die Zone .de

1.4 Standards

Standards ermöglichen die Kooperation im Netzwerk, nur durch sie können Geräte verschiedener Hersteller miteinander kommunizieren. Sie können textuell, mithilfe einer Referenzimplementierung oder anhand von Automaten (meist für zustandsbehaftete Protokolle) festgelegt werden.
Sie müssen verschiedenen Ansprüchen genügen: Vollständig, eindeutig (widerspruchsfrei) und stabil

Protokoll Standardisierte Regeln (Vorschriften) und Vereinbarungen zu Form, Ablauf, Steuerung und Sicherung (Fehler) der Datenübertragung in und zwischen Rechnernetzen, zwischen Einzel-Rechnern und zwischen Rechnern und Peripheriegeräten.

Standard Ein Standard wird von den verschiedensten internationalen und nationalen Organisationen sowie von großen Firmen erstellt. Ein Standard wird als verbindliche oder unverbindliche (empfohlene) Festlegungen schriftlich niedergelegt.

Ablauf einer Standardisierung bei RFCs:

1. **Proposed Standard**: Vollständige, konsistente Spezifikation vorhanden
2. **Draft Standard**: Mindestens 2 unabhängige, interoperable Implementierungen
3. **Standard**: Operationell stabil

Weitere Status: *Experimental*, *Informational* und *Historic*.

1.5 Netze, Autonome Systeme und Schichten

Große Teile des Internet-Backbones werden von wenigen Firmen bereitgestellt (Tier-1). Diese werden an einigen Knotenpunkten verbunden. Wichtiger Knotenpunkt in Deutschland ist DE-CIX in Frankfurt/Main. Man unterteilt folgende **Netzwerk-Schichten**:

Tier 1 : Ein Netzwerk, das mit allen anderen Tier-1-Netzwerken verbunden ist; *Internet-Backbone*; z.B. ATDN, GX, AT&T...

Tier 2 : Netzwerk, das mit vielen Netzwerken verbunden ist, aber Transit *einkauft*, um einige Bereiche des Internets zu erreichen; z.B. Deutsche Telekom

Tier 3 : Ein Netzwerk, das ausschließlich Transit *einkauft*, um das Internet zu erreichen

Autonome Systeme sind Ansammlungen von IP-Netzen, die als Einheit verwaltet werden. Innerhalb kommt ein einheitliches Routing-Protokoll zum Einsatz. Autonome Systeme sind untereinander verbunden und bilden das Internet.

1.6 Begriffe

Datendurchsatz Bla

Datenrate Bla

Routing Bla

2 Protokolle

Protokolle können als Vorschrift betrachtet werden, wie sich verhalten werden soll. Zur Interaktion wird beschrieben, welches Datenformat und wann etwas geschickt werden darf.

2.1 Zustandslose und zustandsbehaftete Protokolle

Bei **zustandslosen Protokollen** wird jede Anfrage in einer eigenständigen Transaktion ausgeführt, es existieren keine Vorbedingungen oder Sitzungsinformationen (UDP, HTTP, TFTP). **Zustandsbehaftete Protokolle** hingegen merken sich den aktuellen Zustand mithilfe einer Sitzung. Nachfolgende Anfragen können auf die Sitzungsinformationen zugreifen. Diese Zustandsübergänge können durch endliche Automaten dargestellt werden. Beispiele sind FTP, TCP und SMTP.

2.2 OSI-7-Schichten-Modell

1. Physical Layer / Bitübertragung
2. Data Link Layer / Sicherungsschicht / Datenübertragungsschicht
3. Network Layer / Vermittlungsschicht
4. Transport Layer
5. Session Layer / Sitzungsschicht
6. Presentation Layer / Darstellungsschicht
7. Application Layer / Anwendungsschicht

Gute **Eselsbrücken** sind:

- Alle deutschen Studenten trinken verschiedene Sorten Bier (deutsche Bezeichnungen, 7-1)
- An dem Samstag trug Verena 'nen String in Blau (deutsche Bezeichnungen, 7-1)
- Alle poppen Susis Tante nach der Party (deutsche Bezeichnungen, 7-1)
- Physiker, die nicht trinken sind potentielle Attentäter (deutsche/englische Bezeichnungen, 1-7)
- Alibaba präsentiert sich täglich nackt dem Personal
- Please Do Not Throw Salami Pizza Away (englisch, 1-7)

Jede Schicht n nutzt die darunterliegende Schicht $n-1$ um mit dem Kommunikationspartner zu kommunizieren. Daten höherer Schichten werden in niederen Schichten umkapselt. Die Bezeichnung der Pakete ist je nach Schicht unterschiedlich:

Data Link Layer : (Ethernet-)Frame

Network Layer : Paket

Transport Layer : Fragment

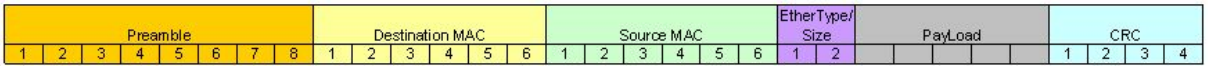
2.3 Ethernet

Das Ethernet-Protokoll wirkt auf den Layern 1 + 2 und wird im Standard **IEEE 802.3** definiert. Es kümmert sich um

- Elektrokrams (Physikalische Eigenschaften, Stecker, Stromversorung, Kabel etc.),
- Zugriffsverfahren auf das Medium,
- Adressierung (MAC),
- Protocol-Multiplexing,
- Flow Control (Logical Link Control),
- Fehlererkennung (CRC).

Es ähnelt den Standards **802.11** (WLAN), **802.15.1** (Bluetooth) und **802.16** (WiMAX).

Ein **Ethernet-Frame** hat eine Größe von 64 - 1518 Byte. Davon ausgenommen sind die Präambel und der SFD. Wird das VLAN-Tag genutzt, sind 1522 Byte möglich. Das **Ethernet-Paket** (Offensichtlich Präambel + SFD + Ethernet-Frame) umfasst folgende Felder:



Präambel : Zum Synchronisieren von Sender und Empfänger, *Einschwingphase* (8 Byte)

SFD : Festgelegte Sequenz 10101011 (1 Byte)

Ziel-Mac-Adresse : Adresse des Empfängers (8 Byte)

Quell-Mac-Adresse : Adresse des Senders (8 Byte)

VLAN-Tag : Nach IEEE 802.1q, optional (4 Byte)

Typ-Feld : Identifiziert die Art des nachfolgenden Inhalts, z.B. IP, ARP, etc...

Nutzlast

PAD-Füllfeld : Wird optional benötigt, um die Mindestlänge von 64 Byte einzuhalten [1]

CRC-Prüfsumme : Zur Fehlererkennung (4 Byte)

2.3.1 CSMA/CD

CSMA/CD regelt den Zugriff auf ein von mehreren Teilnehmern genutztes Medium (Kabel). Dazu prüft der sendende Host, ob das Medium frei ist, bevor er sendet. Beim Übertragen von Daten können Kollisionen erkannt werden. Der Sendevorgang wird dann nach einer zufälligen Zeit wiederholt. Aufgrund der verbreiteten Nutzung von Switches sind echte geteilte Medien inzwischen eher die Ausnahme.

⇒ Jeder Port am Switch bildet eine eigene *Kollisionsdomäne*. Die Bustopologie mit Koaxialkabeln (aber auch mit Hubs) wird nicht mehr genutzt.

2.3.2 Duplex / Half Duplex

Beim **Full Duplex** sind beide Seiten in der Lage, gleichzeitig zu Senden und zu Empfangen. Im Falle von **Half Duplex** ist dies nur wechselseitig möglich (vgl Walkie Talkie). Es sind verschiedene Realisierungen einer geteilten Nutzung eines Mediums möglich:

Zeitduplex (TDD) : Übertragung in verschiedenen Zeitschlitzen

Frequenzduplex (FDD) : Übertragung auf verschiedenen Frequenzen

Codeduplex : (nicht im Skript)

[1] Rausfinden, warum mindestens 64 Byte nötig. Vermutung: Kollisionserkennung

2.4 Switching

Switches sind Geräte auf dem OSI-Layer 2. Sie empfangen Ethernet-Frames und leiten sie anhand ihrer Empfänger-MAC-Adresse weiter. Im Gegensatz zum Hub wird dabei nur über den Port ausgegeben, hinter dem sich der Empfänger befindet. Die Ausnahme ist hierbei, wenn der Port des Empfängers nicht bekannt ist. Anhand der empfangenen Frames lernt ein Switch, wo sich Geräte befinden.

2.4.1 Realisierungsmöglichkeiten

2.4.2 Cut-Through und Store-and-Forward

Beim **Cut-Through** (auch *On The Fly Forwarding*) werden Pakete sofort nach Empfang der Empfängeradresse auf dem entsprechenden Port weitergeleitet, sofern dieser Frei ist. Diese Methode ist sehr schnell (Verzögerung ca. 40μs), leitet jedoch gegebenenfalls auch fehlerhafte Frames weiter, da CRC umgangen wird.

Store-and-Forward hingegen empfängt zuerst den gesamten Frame, prüft diesen und leitet ihn anschließend weiter. Offensichtlich werden keine fehlerhaften Pakete mehr in benachbarte Segmente weitergeleitet, dies wird jedoch durch erhöhte Latenz erkauft.

In der Praxis arbeiten Switches häufig im Cut-Through-Modus und schalten bei erhöhter Fehlerrate in den Store-and-Forward-Modus.

2.4.3 VLAN

Ermöglicht die Aufteilung von Switches in mehrere virtuelle LANs. Den Ports werden dabei einzelne VLANs zugeordnet. Auf diese Weise kann Hardware eingespart werden. Realisiert wird dies mit einem 4 Byte langen Feld im Ethernet-Frame:

- 2 Bytes **TPID** - Tag Protocol Identifier – Fester Wert 0x8100. Frame trägt die 802.1q/802.1p-Tag-Information
- 3 Bit **Priorität** (user_priority) – Benutzer-Prioritätsinformationen
- 1 Bit **CFI** - Canonical Format Indicator – Gilt für alle vorhandenen MAC-Adressinformationen im MAC-Datenpaket des Frames. Wert 0 das Format ist kanonisch (am wenigsten signifikante Bit zuerst); Wert 1 Format nicht-kanonisch. Benutzung im Token Ring/Source-Routed- FDDI-Media-Zugang, um die Bit-Order der Adressinformationen des verkapselten Frames festzulegen
- 12 Bit **VID** - VLAN Identifier – Identifizierung des VLANs zu dem der Frame gehört

Erleichtert die Arbeit eines Administrators, da es viele Probleme von physikalischen Verbindungen umgeht. (bspw. Viel Hardware, unflexibel, Anpassungen nur mit hohem Aufwand)

„Faulheit ist die Mutter der Ingenieurswissenschaften“

2.4.4 Trunking / Link Aggregation

Ermöglicht die Zusammenfassung mehrerer Ports zur Erhöhung des Datensatzes.

2.5 Asynchronous Transfer Mode

2.6 ATM

- ATM kann als Protokoll für Internettelefonie eingesetzt werden und bietet eine geringe Latenz (von unter 200ms).
- Im Gegensatz zu Ethernet bietet ATM Garantien(!) und besitzt einen geringen Header von 5Byte. Es wird eine Leitung für den Datenstrom geschaltet.

2.7 Internet Protocol

Beim Internet Protocol handelt es sich um ein Layer-3-Protkoll, weches auf die Layer-2-Protokolle Ethernet, ATM und FDDI aufsetzen kann. Es verwendet globale, logische Adressen. Aufgrund der Erschöpfung des IPv4-Adressraumes[2] (32 Bit) wird nach und nach IPv6 eingeführt (128 Bit)

[2] Weitere Maßnahmen, dem entgegenzuwirken sind etwa: NAT, CIDR, DHCP, Private Adressräume

2.7.1 IPv4

Wurde im RFC 791[60] definiert. Der Header eines IPv4 Paketes ist insgesamt 20 Byte lang. Davon sind insbesondere die folgenden von Interesse:

Version : In diesem Fall *4*, bei IPv6 offensichtlich *6* (4 Bit)

Header Length : Gesamtlänge des Headers kann 20 Byte überschreiben, wenn zusätzliche Optionen gesetzt werden. Angabe in 32-Bit langen Blöcken (4 Bit)

Total Length : Gesamtgröße des Pakets. Nach RFC muss jeder Host in der Lage sein, mindestens Pakete mit einer Länge von 576 Bytes zu verarbeiten. (16 Bit)

Type of Service : Type of Service nach RFC791(ursprünglich für Quality-of- Service-Anwendungen gedacht)

- bits 0-2: precedence
- bit 3: 0 = Normal Delay, 1 = Low Delay
- bit 4: 0 = Normal Throughput, 1 = High Throughput
- bit 5: 0 = Normal Reliability, 1 = High Reliability
- bits 6-7: Reserved for future use

Heute anders verwendet zur Servicebeschreibung durch Dienstklassen (DiffServ, 8 Bit)

Identification : Falls ein Paket fragmentiert wird, haben alle Fragmente die selbe Identification.

Flags : Reserved[4], Don't Fragment, More Fragments (3 Bit)

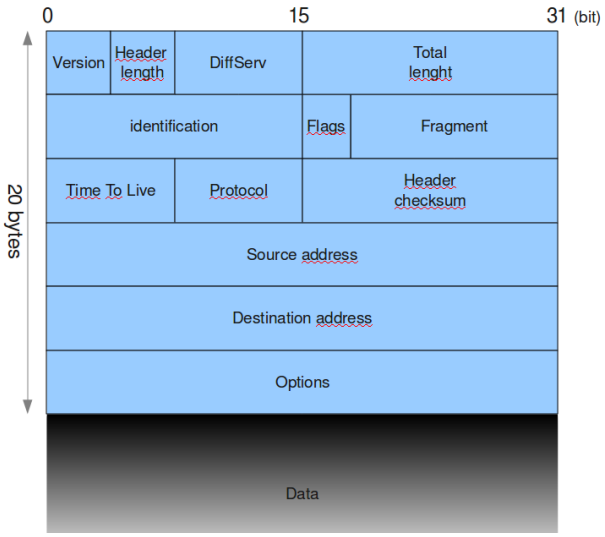
Fragment Offset : Kann ein Paket nicht auf einmal übertragen werden (z.B. bei kleinerer Maximum Transfer Unit, MTU), wird es fragmentiert. FO gibt an, ab welcher Stelle (gemessen in Blöcken von 8 Byte) dieses Paket die Daten enthält (MF Flag ist gesetzt) (13 Bit)

TTL : Anzahl der Hops, bis Paket verworfen wird (wird bei jedem Routingvorgang reduziert)

Protocol : Enthält für die darüber liegenden Layer Informationen, „sodass diese etwas damit anfangen können"

Checksum wird selbst als 0 betrachtet, und fließt so nicht in die Kalkulation mit ein

Options : Beispielsweise für Source Routing (Route ist im Paket vorgegeben); Sehr selten verwendet, häufig blockiert oder ignoriert



Nutzung von Adressklassen[3] aufgrund der Verknappung der Adressen durch CIDR[33] abgelöst. Dies ermöglichte Super- und Subnetting. Adressangabe bei CIDR im Format a.b.c.d/x, wobei x angibt, wie viele Bits zum Netz-Anteil der Adresse gehören. Subnetze dienen zur Aufspaltung von Netzen in Teile, um diese besser handhaben zu können (Broadcast-Domains, Logische Strukturierung, Dezentrale Verwaltbarkeit)

[3] Class A 1.x.y.z-126.x.y.z; Class B 128.0.y.z-191.255.y.z; Class C 192.0.0.z-223.255.255.z

2.7.2 IPv6

Die auffälligste Änderung von IPv4 zu IPv6 ist die vergrößerte Adressgröße (128 Bit). Damit ergeben sich $3,4 \cdot 10^{38}$ Adressen. IP - Adressen werden im Hexadezimalsystem zu je acht Word-Gruppen á 2 Bytes dargestellt. Verkürzte Darstellung möglich durch Verzicht auf „Nullen“ in einer Gruppe (einmal je Adresse). Es existieren IPv4-kompatible Adressen und die CIDR-Darstellung für Subnetze bleibt erhalten. Weiterhin wurde die Anzahl der Felder im Header reduziert und (optionale) Erweiterungsheader hinzugefügt. Wichtige Felder sind:

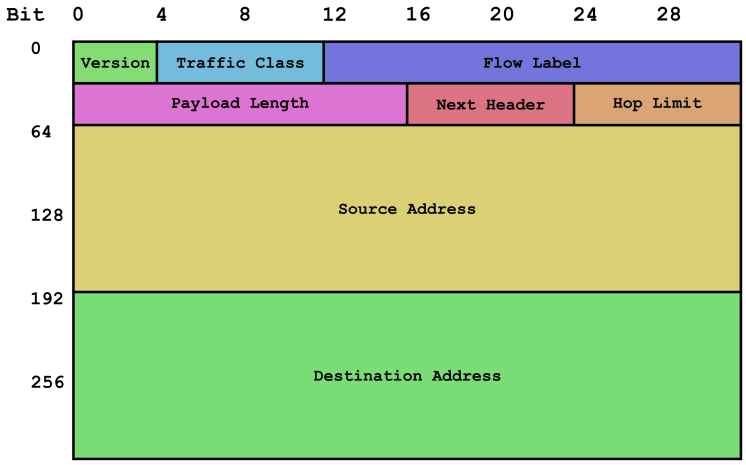
Traffic Class :

- 0 uncharakterisierter Verkehr
- 1 „Füllmaterial”, z.B. Newsgroups
- 2 zeitunkritischer Verkehr, z.B. EMail
- 3 reserviert
- 4 Mengendaten, z.B. FTP, NFS
- 5 reserviert
- 6 Interaktive Anwendungen, z.B. telnet
- 7 Steuerung, z.B. SNMP

Flow Label : Anwendung kann Datenstrom mit einem Flow-Label versehen, z. B. bei Streaming-Anwendungen. Flow nicht notwendigerweise an Verbindung gebunden (logisch, da IP nicht verbindungsorientiert arbeitet). Empfänger kann Datenstrom am Flow Label erkennen. [64, 63]

Next Header : Gibt an, dass ein weiterer Header folgt. In IPv6 sind viele Felder weggefallen. Next Header ermöglicht das Anfügen eines weiteren Headers. RFC 2460[16] bietet beispielsweise:

- Hop – by – Hop Options Header
- Routing Header
- Fragmentation Header
- Authentication Header
- Encapsulated Security Payload (ESP) Header
- Destination – Option – Header



2.7.3 Migration von IPv4 nach IPv6

Wie migriert man Millionen von Hosts im Internet auf IPv6? Langsam, nach und nach. Alle Hosts auf einmal sind nicht realisierbar. RFC 1933[34] schlägt drei Migrationsstrategien vor.

Tunneling : Zwischen zwei IPv6-Knoten wird ein virtueller Link aufgebaut. Die IPv6-Pakete werden (als Payload) in IPv4-Pakete verpackt und *normal* über das Internet geroutet.

Dual Stack : Auf Hosts und Routern werden sowohl IPv4 als acuh IPv6 eingerichtet. DNS kann A- oder AAAA-Records zurück geben, entsprechender Stack wird dann genutzt. Wird häufig mit Automatic Tunneling[4] genutzt.

[4] “IPv6-over-IPv4 tunneling where the IPv4 tunnel endpoint address is determined from the IPv4 address embedded in the IPv4-compatible destination address of the IPv6 packet.”

Außerdem besteht die Möglichkeit von **Assignment of IPv4 Global Addresses to IPv6 Hosts** (AIIH). Dabei wird eine IPv4-kompatible IPv6-Adresse genutzt. Diese entspricht dem 96-bit Präfix 0:0:0:0:0:0, gefolgt von der IPv4-Adresse. Ist der Client der Dual-Stack-Hosts und der Server IPv4-only, verlangt der Client für die Dauer der Kommunikation eine temporäre IPv4 Adresse beim AIIH Server (Kooperation von DNS und DHCPv6). Andernfalls (Client IPv4-only; Server Dual-Stack): DNS verlangt bei DHCPv6 eine temporäre IPv4 Adresse für Dual-Stack Host, welcher mit dieser rekonfiguriert wird.

Übersetzung der Header (Header Translation): Hierbei wird die IPv4 Unterstützung auf Systemen entfernt. Die IPv4-Pakete werden in IPv6-Pakete übersetzt; ein Translator übersetzt IP und ICMP Meldungen. Erweiterungsheader werden nicht, oder nur bedingt übersetzt. Probleme entstehen, weil sich einige Felder nicht immer übersetzen lassen und Adressumwandlung Datenbanken-Lookups erfordern.

2.8 User Datagram Protocol

Bei UDP handelt es sich um ein Layer-4-Protokoll.[58] Es dient zur Übermittlung kurzer Nachrichten an andere Systeme und garantiert weder Zuverlässigkeit noch Einhaltung der Reihenfolge der Pakete beim Empfänger. Die Adressierung geschieht über Ports (16 Bit). Der Header enthält 4 Felder (je 16 Bit): Quellport, Zielport, Datagram-Länge und eine Checksumme.

2.9 Transmission Control Protocol

TCP ist ebenfalls ein Layer-4-Protokoll.[59] Es garantiert eine Ankunft der Pakete in korrekter Reihenfolge. Clienten sehen die Verbindung als bidirektionalen Datenstrom, tatsächlich findet die Kommunikation über Pakete statt. Die Kommunikation erfolgt durch TCP zustandsbehaftet. TCP arbeitet byte-orientiert.

2.9.1 Paketstruktur

Der Header des TCP-Fragments ist 20 Byte groß. Wichtige Felder sind:

Sequence Number / Acknowledgement Number : Zerlegung des Datenstroms in nummerierte Blöcke. Größe der Blöcke ist variabel (Nagle Algorithmus). Verwerfen von Segmenten mit fehlerhafter Prüfsumme. Bestätigung empfangener Segmente. Nicht unbedingt für jedes Segment einzeln (Windowing). Erneuter Transport unbestätigter Segmente. Zusammensetzung des Datenstroms auf Empfängerseite

Flags : dienen unter anderem zur Steuerung des Verbindungsauf- und –abbaus.

URG - Urgent Flag (Urgent Pointer enthält Sequenznummer, die bevorzugt übertragen werden soll)

ACK – Acknowledgement

PSH – Push (Paket wird sofort an Anwendung weitergeleitet, ohne Zwischenpuffer)

RST – Reset (Unterbrechung der Verbindung)

SYN – Synchronized (Aufbau der Verbindung)

FIN – Finish (Beenden der Verbindung)

Window Size : Anzahl der Daten, die gesendet werden können bis ein Acknowledgement gesendet werden muss (in Bytes oder mit speziellem Option Header nach RFC1323 [39] auch bis zu 1GB, dann Linksverschiebung um bis zu 14 Bits, $2^{14} \cdot 64k = 1G$).

2.9.2 Zustände

Zum Aufbau einer Verbindung wird der **3-Way-Handshake** durchgeführt:

1. → SYN
2. ← SYN + ACK
3. → ACK

Ein Timeout findet typischerweise nach 75 Sekunden statt. Zum Abbau der Verbindung genügt das Senden und Quittieren eines FIN.

2.9.3 Sliding Window & Nagle-Algorithmus

Hinweis: Konnte in den Folien hierzu nix groß finden, Infos stammen aus dem Tanenbaum[5][72].

Die Flusskontrolle von TCP funktioniert im Groben folgendermaßen: Nach Erhalt der Daten quittiert der Empfänger das letzte Empfangene Segment im `ACK`-Feld. Empfängt er beispielsweise eine 2kB große Nachricht mit den Segmenten 0 - 2048, sendet er `ACK=2048`. Gleichzeitig ermöglicht das **Window Size** (WIN) Feld eine Angabe, wie viele Segmente der Empfänger noch annehmen kann, etwa weil er nur einen begrenzt großen Puffer hat, der erst gelesen und geleert werden muss.

Bei `WIN=0` wird dem Sender signalisiert, dass der Puffer des Empfängers voll ist und keine Daten empfangen werden können. Ausnahmen sind hier priorisierte Nachrichten und *window probes*[6]. Letztere fordern den Empfänger dazu auf, wiederholt das nächste erwartete Byte und die Fenstergröße zu übermitteln, um Deadlocks vorzubeugen.

Es muss nicht zwingend jedes TCP-Segment einzeln bestätigt werden. Anwendungen, wie Telnet oder SSH etwa, könnten für jeden Tastaturanschlag ein einzelnes Segment verschicken, was eigentlich zu bestätigen ist. Stattdessen werden hier häufig verzögerte Bestätigungen eingesetzt. Dabei kann bis zu 500ms auf weitere eintreffende Daten gewartet werden, die dann gleich mitbestätigt werden.

Da der durch viele kurze Pakete erzeugte Overhead die Effizienz der Kommunikation negativ beeinflusst, wird häufig der **Nagle-Algorithmus** eingesetzt. Hierbei wird, wenn die Daten vom Sender byteweise kommen, immer nur das erste Byte gesendet und die nachfolgenden Nachrichten im Puffer gehalten. Wird das erste Byte bestätigt, so werden alle sich im Puffer befindlichen Daten gesendet. So werden viele kleine Daten in großen Segmenten zusammengefasst. Aufgrund der Verzögerung kann diese Technik auch unerwünschte Folgen haben, etwa bei Spielen oder Konsolenanwendungen.

2.10 SCTP

Das **Stream Control Transmission Protocol**[71] ist ein zuverlässiges (Reihenfolge, Duplikate, Veränderungen, Fehlende Daten) Transportprotokoll für Unicast-Kommunikation. Es arbeitet nachrichten-orientiert, die Nachrichten[7] haben eine interne Struktur. SCTP ermöglicht multi-streaming, Daten können in parallelen Streams übertragen werden. Dies ist beispielsweise dann hilfreich, wenn Websiten mit mehreren Ressourcen übertragen werden. Streams werden einzeln nummeriert (Retransmits müssen nur für den jeweiligen Stream erfolgen). Daten-Chunks können einzeln bestätigt werden. (**Selective ACK**) Retransmits nur für fehlende Pakete.

Es bietet **Schutz gegen SYN Flooding** (Cookie-Mechanismus) - Anfrager bekommt Cookie und muss das nachfolgend wieder vorweisen, dies bedeutet keinen Ressourcenverbrauch beim Server: Der Server speichert dazu bei einer Verbindungsanfrage (`INITPaket`) keine Zustandsinformationen, sondern schickt diese in Form eines Cookies (`INIT-ACK-Paket`) an den Client. Der Client muss dieses Cookie in seine Antwort (`COOKIEECHO`-Paket) einfügen und wird damit vom Server als zum Verbindungsaufbau berechtigt erkannt. Eine Bestätigung geschieht durch ein `COOKIE-ACK`-Paket.

SCTP ermöglich **Multi homing**. Dabei haben beide Kommunikationspartner eine primäre und potentiell mehrere sekundäre (IP-)Adressen. Auf diese Weise ist es möglich, eine stabile Verbindung aufrecht zu erhalten, auch wenn ein Interface inaktiv wird. Offensichtlich können hierbei die Nachrichten unterschiedliche Pfade nutzen.

2.10.1 Paketaufbau

Ein allgemeines Paket besteht aus einem **Common Header** und mehreren nummerierten **Chunks**. Der Header besteht aus **Source-** und **Destinationport**, einem **Verification Tag** und einer **Checksum**. Chunks bestehen aus einem **Typen**, einer Reihe von **Flags**, einem **Längenfeld**, sowie dem **Wert (Value)**. Letzteres setzt sich, je nach Chunk-Type, aus mehreren Feldern zusammen. Weiter unten wird zur Veranschaulichung ein Payload Data Chunk beschrieben.

Der Typ des Chunks wird durch das 8-Bit lange Type-Feld bestimmt. Dafür sind die Werte 0-254 verfügbar. Wert 255 ist für eine zukünftige Erweiterung des Feldes reserviert. Im Folgenden einige Beispiele für Typenfelder. Weitere Werte sind verfügbar oder bereits reserviert.

0. Payload Data `DATA`
1. Initiation `INIT`
2. Initiation Acknowledgement `INIT ACK`
3. Selective Acknowledgement `SACK`[sic!]
4. Heartbeat Request `HEARTBEAT`
5. Heartbeat Acknowledgement `HEARTBEAT ACK`
6. Abort `ABORT`
7. Shutdown `SHUTDOWN`
8. Shutdown Acknowledgement `SHUTDOWNACK`
9. Operation Error `ERROR`

[5] Das Kapitel heißt in der deutschen Fassung ernsthaft *TCP-Schiebefenster*. Barbaren!
[6] Fenstersondieranfragen
[7] Auch als Chunks bezeichnet

10. State Cookie `COOKIE ECHO` 14. Shutdown Complete `SHUTDOWN COMPLETE`

Das **Verification Tag** im Header wird vom Empfänger genutzt, um den Sender des Pakets zu validieren. Es entspricht dem Wert des Initiate Tag[8]. **Ausnahmen:** Beinhaltet das Paket einen `INIT Chunk`, ist das Verification Tag Null. Wenn ein Paket einen `SHUTDOWN COMPLETE Chunk` enthält, bei dem das T-Bit gesetzt ist, so wird das Verification Tag vom `SHITDOWN ACK chunk` kopiert. Zu guter Letzt wird das Verification Tag eines `ABORT chunks` vom ABORT-verursachenden Paket übernommen.

Payload Data Chunks sollen im Folgenden kurz besprochen werden. Sie haben den Type-Wert 0. Im Flag-Feld sind die letzten 3 Bits von besonderem Interesse, gefolgt vom Längen-Feld. Die Flags haben haben die folgende Bedeutung:

U — Ist das *Unordered* Bit gesetzt, wird die Stream Sequence Number ignoriert. Das bedeutet, dass empfangene Chunks, ohne vorher sortiert zu werden, an den darüberliegenden Layer weitergereicht werden. Ist eine Nachricht fragmentiert, so wird dieses Bit bei jedem Fragment gesetzt.

B — Das *Beginning* fragment Bit gibt an, dass es sich um das erste Fragment einer Nachricht handelt

E — Das *Ending* fragment Bit gibt an, dass es sich um das letzte Fragment einer Nachricht handelt.

Das Value-Feld ist beim DATA Chunk wie folgt aufgebaut:

Transmission Sequence Number (TSN) — enthält einen Wert zwischen 0 und $2^{32} - 1$. Sie identifiziert einen Chunk und wird zur Bestätigung des Erhalts genutzt. Nach erreichen des maximalen Wert beginnt die Nummerierung wieder bei 0.

Stream Identifier S — identifiziert den Stream, zu dem der Chunk gehört. Dies ist notwendig, wenn parallele Streams versendet werden.

Payload Protocol Identifier — identifiziert das genutzte Protokoll. Das Feld wird von SCTP selbst nicht genutzt, sondern nur an höhere Schichten weitergereicht, beziehungsweise von Zwischenstationen interpretiert.

User Data — enthält letztendlich die zu transportierenden Daten.

3 Adressierung

3.1 MAC-Adressen

- Layer-2-Adressen für Ethernet
- 6 Byte / 48 Bit groß
- *Eigentlich* weltweit eindeutig
- *Eigentlich* in Hardware gegossen, trotzdem fälschbar
- Darstellung: Hexadezimal, Bits getrennt durch [.:-]
- Bit 3 bis Bit 24 an Hersteller gebunden (z.B. 00-60-2F-xx-xx-xx für Cisco)
- Broadcast: FF-FF-FF-FF-FF-FF

3.2 VPI und VCI bei ATM

ATM beruht auf Verbindungen, die sowohl fest eingerichtet oder nur für eine bestimmte Zeit geschaltet werden können. Für diesen Zweck wurden Virtual Paths (VPs) und Virtual Channels (VCs) definiert. Jede ATM-Zelle enthält im Header einen Virtual Path Identifier (VPI, 8 bzw. 12 Bit) und einen Virtual Channel Identifer (VCI, 16 Bit).
Ein ATM-Paket besteht aus 5 Byte Headerund 48 Byte Nutzlast. Es gibt zwei Arten von ATM-Paket-Formaten:

1. NNI (Network-Network-Interface)
   Wird von öffentlichen ATM-Netzen verwendet.
2. UNI (User-Network-Interface)
   Wird für private ATM-Verbindungen genutzt. UNI besitzt ein GFC-Feld für eine (bis heute undefinierte) lokale Flusskontrolle zwischen Netz und User.

[8] Das Initiate Tag ist ein Feld im `INIT Chunk`.

3.3 IP-Adressen

Siehe Abschnitt 2.7.

3.4 Ports

Für die Layer-4-Protokolle UDP und TCP werden 16-Bit-Adressen verwendet. Diese werden als Ports bezeichnet. Ports 0-1023 sind dabei standardisiert. Wichtige Portnummern sind beispielsweise:

Port	TCP	UDP	Beschreibung
20	Ja	Nein	FTP - Datenübertragung
21	Ja	Nein	FTP - Kontrolle
22	Ja	(Ja)	SSH
23	Ja	Nein	Telnet
25	Ja	Nein	SMTP
53	Ja	Ja	DNS
80	Ja	Nein	HTTP
110	Ja	Nein	POP3
123	Nein	Ja	NTP
443	Ja	Nein	HTTPS

3.5 URIs, URNs und URLs

Uniform Resource Identifier : String; Benennt oder identifiziert eine Ressource z.B. \\139.30.3.23\iukp\beispiel.txt

Uniform Resource Name : Sonderform der URI, die eine Ressource in einem bestimmten Namespace benennt — urn:ip-addr:139.30.3.23

Uniform Resource Locator : Sonderform der URI, die zusätzlich das Protokoll angibt, über das die Ressource erreichbar ist z.B. http://139.30.3.23/beispiel.txt

3.6 *cast

Unicast : Einer sendet an einen, wie beispielsweise in einer TCP-Verbindung für HTTP. Adressierung durch Angabe des Empfängers.

Broadcast : Einer sendet an alle, wie beispielsweise in ARP oder DHCP. Hierzu werden spezielle Adressen (MAC: FF-FF-FF-FF-FF-FF; IP: höchste Adresse im Subnetz) verwendet. Broadcastbereiche sollten möglichst klein gehalten werden, um möglichst wenig Teilnehmer zu belästigen.

Multicast : Einer sendet an viele Mitglieder einer Gruppe. Siehe Abschnitt 9.

Concast : Viele senden an einen, Nachrichten werden so früh wie möglich zusammengefasst. Verhindert Implosion, Überlastung des Empfängers. Beispielswiese sinnvoll, wenn Fehlermeldungen zusammengefasst werden können. Keine direkte Unterstützung in IP, MAC, Ethernet

3.7 ICMPv6 Neighbor Discovery Protocol (NDP)

- Dient als Ablösung von ARP
- Mechanismus von IPv6 für Auflösung (Übersetzung) von Netzwerkadressen in Hardwareadressen[53].
- Es existieren verschiedene Typen:
  - Router Solicitation – Type 133
    Alle Router im selben Netz werden aufgefordert, sich zu melden und senden an Router-Multicast-Adresse
  - Router Advertisement – Type 134
    Router verkünden Anwesenheit im Netz, periodisch oder als Antwort auf *Router Solicitation*
  - Neighbor Solicitation – Type 135
    Zur Auflösung von IPv6-Adressen zu Link-Layer-Adressen. Umgesetzt als Broadcast an Link-Layer-Broadcast-Adressen.
  - Neighbor Advertisement – Type 136
  - Redirect – Type 137
    Router teilt mit, dass es besseren next hop gibt

3.8 Domainverwaltung

- Adressierung mit Hilfe von IP-Adressen ist für menschliche Nutzer sehr unpraktisch. Daher werden Namen zur Adressierung verwendet.
- Namen sind in einem paketvermittelten Netzwerk unpraktikabel. Daher ist eine Umwandlung der Namen in IP-Adressen notwendig.
- Domain Name Service (DNS) erledigt diese Aufgabe.
- Verteilung der Verantwortlichkeiten.
  - Keine einzelne Institution verwaltet alle Namen im Baum.
  - NIC verwaltet die Top-Level Domain.
  - Die Verwaltung der unteren Ebenen wird entsprechend verteilt.
- Wird die Verantwortung für eine Zone delegiert, dann ist die entsprechende „Person“ für diese Zone verantwortlich (Zerlegung der Aufgabe in kleinere Teilaufgaben).
  - Zuordnung von Namen und IP-Adressen.
  - Betrieb von Primary Name Server und mindestens ein Secondary Name Server.

3.9 Informationen in der Domain Registry

- IPv4-Adressen (A-Records)
- IPv6-Adressen (AAAA-Records)
- Adresse des Mail-Servers (MX-Records)
- Referenzen auf andere Einträge (CNAME)

4 ARP, RARP

ARP wandelt Layer-3-Adressen in Layer-2-Adressen um. Es beschränkt sich nicht auf das Erfragen von MAC-Adressen zu IP-Adressen, sondern ermöglicht beispielsweise auch ATM ARP, IP over FDDI und IP over Token Ring.

- Erfragt für gegebene IP-Adresse eine MAC-Adresse.
- Host erzeugt Broadcast.
- Der angefragte Host darf darauf antworten (Unicast).
- Nachdem der Sender der Broadcast-Nachricht die Antwort erhalten hat, kann er die IP-Adresse der Ethernet-Adresse zuordnen.
- Diese Ethernet-Adresse wird dann für alle folgenden Pakete an diese Internet-Adresse verwendet, solange bis die Cache-Zeit abgelaufen ist.
- RFC 826 [56]

4.1 Einsatzfälle

1. Zwei Hosts möchten im selben Netzwerk (ausschließlich Layer 2) miteinander kommunizieren und kennen nur die Layer-3-Adresse (z.B. IP-Adresse) des Empfängers.
2. Ein Host benötigt die Layer-2-Adresse des Gateways, um andere Netze zu erreichen (Sonderfall des zuvor genannten)
3. Zwei Gateways wollen kommunizieren.

4.2 Paketstruktur

Wichtige Felder eines ARP-Requests sind:

Hardware type : Kennzeichen für die verwendeten Hardware-Adressen (1 für Ethernet, 2 Byte)

Protocol type : Kennzeichen für die verwendeten Protokoll-Adressen (L3) (0x0800 für IPv4, 2 Byte).

Hardware length : Länge der Hardware-Adresse in Bytes (6 für Ethernet, 1 Byte)

Protocol Length : Länge der Protokoll-Adressen (L3) (4 für IPv4, 1 Byte)

Operation : Unterscheidung von 1 Request und 2 Reply, da für beides gleiche Pakete verwendet werden (2 Byte, warum zur Hölle reserviert man für 2 Werte 2 Byte?)

Es folgen Sender hardware address, Sender protocol address, Target hardware address und Target protocol address

4.3 ARP-Caching

Um nicht für jedes IP-Paket einen neuen ARP Request zu stellen, werden die Ergebnisse zwischengespeichert. Typischerweise 10 Minuten lang. Kommt es während der Cache-Zeit zu einem Fehler (Host nicht erreichbar), wird erneut ein ARP-Request ausgeführt.

4.4 ARP-Announcements

Der anfragende Host sendet bekanntlich einen Broadcast, alle Host im gleichen Netz können folglich diese Information ebenfalls verwenden, und auf diese Weise eigene Anfragen sparen. Anwendungen:

- Unterbrechungslose Übernahme einer Protokoll-Adresse (z.B. IP-Adresse) in hochverfügbaren Systemen.
- Der anfragende Host kann auf diese Weise feststellen, ob eine Protokoll-Adresse bereits vergeben ist (Gratuitous ARP). Dieser Mechanismus wird bei IP-Autoconf verwendet.

IP-Autoconf [10] ist eine einfache Möglichkeit zur Adresskonfiguration ohne Server (DHCP, RARP).

- Host wählt zufällig eine Adresse.
- Überprüft mittels Gratuitous ARP, ob diese Adresse bereits vergeben ist.
- Falls ja, andere Adresse zufällig wählen und erneut überprüfen.
- Falls nein, Adresse benutzen und andere ARP Requests für diese Adresse beantworten.

4.5 ARP-Spoofing

Da ARP nicht kryptografisch abgesichert ist, kann jeder Host auf ARP Requests antworten oder ARP Announcements erzeugen. Ziel ist das Umleiten von Datenpaketen. Es gibt Programme, die Änderungen der Hardware-Adressen erkennen (z.B. arpwatch). Der Administrator muss dann die Ursache überprüfen

4.6 Proxy ARP

Wird von speziellen Routing-Protokollen verwendet. Jeder Nachbar (One-Hop-Neighbor), der auf der Route zum Ziel liegt, beantwortet einen eintreffenden ARP-Request mit seiner Hardware-Adresse. Pakete werden so von Host zu Host weitergeleitet

4.7 Reverse RP

Erfragt eine IP-Adresse bei bekannter MAC-Adresse und ist nicht unbedingt für die Funktionsfähigkeit von IP über Ethernet notwendig. RARP wird in RFC 903 [31] standardisiert. Haupteinsatzzweck ist die automatische Konfiguration der Protokoll-Adresse. Dabei sendet der Host einen RARP-Request mit der eigenen Hardware-Adresse (z.B. MAC). Ein RARP-Server beantwortet diesen Request und liefert die hinterlegte Protokoll-Adresse (z.B.IP). Da RARP Link-Local-Broadcasts verwendet, ist ein RARP-Server pro Netz notwendig. Benutzt gleiche Paketstruktur wie ARP, lediglich anderen EtherType[9] (0x8035)

[9] Feld im Ethernet-Header, siehe Abschnitt 2.3

5 DNS und WHOIS

Die Aufgabe des Domain Name Systems ist die Übersetzung von (Layer-4) IP-Adressen in, für den menschlichen Gebrauch besser verwendbare Namen. Der Name wird dabei aus hierarchischen Domains, getrennt durch Punkte (.) zusammengesetzt.
Direkt unter der Wurzel steht hierbei die sogenannte Top-Level-Domain. Bei Top-Level-Domains wird zwischen **länderspezifischen** (ccTLD) und **generischen** (gTLD) Domains unterschieden. Während erstere, bestehend aus zwei Buchstaben, durch lokal verantwortliche Institutionen[10] verwaltet werden, werden gTLD durch die **ICANN** oder beauftragte Institutionen vergeben. Bei der Vergabe gilt das *first come, first serve*-Prinzip. Für die Namensbildung gelten gewisse Regelungen, z.B. Ziffern 0-9, Bindestriche, Buchstaben, sowie einige ausgewählte lokale Buchstaben. Letztere müssen ASCII-kodierbar sein (ACE)[22]. Jede (Teil-) Domain ist maximal 63 Zeichen lang, insgesamt ist der vollständige Pfad jedoch 255 Zeichen lang. Folgende Daten sind bei der Domainanmeldung von Interesse:

Domaininhaber (Holder): Person, der die Domain *gehört*

Administrativer Ansprechpartner : Vom Inhaber Bevollmächtiger, darf alle Entscheidungen treffen (Admin-C)

Technischer Ansprechpartner, Zonenverwalter : Typischerweise Kontaktadresse der Firma, die die Domain betreibt (Tech-C, Zone-C)

Technische Daten : z.B. Nameserver

Hoster tragen oft nicht den Inhaber selbst als Admin-C, sondern sich selbst ein. Außerdem stellen sie in der Regel den Nameserver zur Verfügung. Die Registrierung einer Domain gilt immer für einen bestimmten Zeitraum. Ein Wechsel erfordert eine Freigabe beim alten Provider durch Inhaber oder Admin-C.

5.1 Weiteres Bla zu DNS

Beim Domain Name System handelt es sich um ein **sehr großes**, **hierarchisch strukturiertes**, **verteiltes**, **repliziertes** und **lokal verwaltetes System** [49, 50]. Aus dem hierarchischen Benennungsschema resultiert das verteilte Datenbanksystem des DNS. Jede Domain bestimmt selbst, wie die unter ihr liegenden Domains zugewiesen werden, hat dementsprechend eigene Verantwortlichkeiten. Ist ein Namensserver für eine solche Zone verantwortlich, kann er Anfragen autoritativ beantworten(Autoritäskonzept). Andernfalls wird die Anfrage aus dem Cache beantwortet oder an einen anderen Nameserver delegiert (Delegationskonzept). Das kann der Nameserver einer Subdomain oder (default) ein Root Name server sein.
Die Aufteilung in die Zonen erfolgt anhand von Punkten - Bsp: www.spiegel.de

5.2 Servertypen

Primary Server : Ist Hauptserver einer Domain und autorisiert, Anfragen zu seiner Domain verbindlich zu beantworten. Er verfügt über alle Daten dieser Domain, welche in Zonen-Dateien abgelegt sind, die der Verwalter des Servers erstellt

Secondary Server : Ist ebenfalls autorisiert, verbindliche Antworten zu seiner Domain zu liefern, lädt die Domain-Datenbank von einem Primary Server und aktualisiert sie bei Bedarf

Caching-only Server : Verfügt über keine eigenen Domain-Informationen. Fragt bei dem für die Domain zuständigen Primary oder Secondary Server nach und speichert die Antwort zwischen. Ist zur Beantwortung von Anfragen zu einer Domain „nicht autorisiert“ (kann die Genauigkeit und Aktualität nicht gewährleisten)

Slave Server : Reicht alle Anfragen, die er selbst nicht aus seinem eigenen Cache beantworten kann, an eine zuvor festgelegte Liste von anderen Servern (Forwarders) weiter. Forwarders fragen ihrerseits den zuständigen Server und speichern Ergebnis zwischen (Caching)

6 Timeouts, ACK, Bestätigungen

6.1 Begriffe

- **Positive Bestätigung**
  - Bestätigung, wenn ein Protokollschritt erfolgreich ausgeführt wurde.

[10] Network Information Centers z.B. DENIC für .de

  - Wartezeit ist ein Nachteil.
  - Reaktion auf verloren gegangene Bestätigungen muss vorgesehen werden.

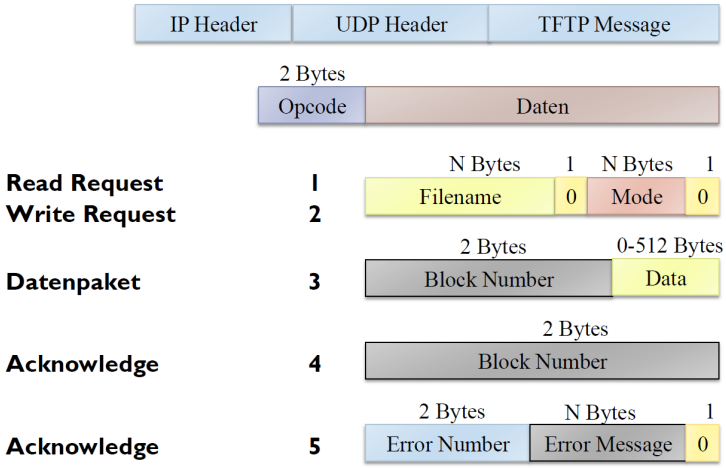
- **Negative Bestätigung**
  - Fehlermeldung, wenn ein Protokollschritt nicht erfolgreich war.
  - Ausbleiben der Fehlermeldung ist nur eine notwendige Bedingung, keine hinreichende, dafür, dass Protokollschritt erfolgreich war.

- **Timeouts**
  - Ein Kontrollsignal in einem technischen System, das die Überschreitung der normalerweise für eine Aktion benötigten Zeit anzeigt. Damit sollen unkontrollierte Zustände (Warten, Endlosschleifen) verhindert und die Systemressourcen geschont werden.[73]

- **Acknowledgements**
  - Ein ACK dient der Bestätigung eines empfangenen Datenpakets.

6.2 TFTP - Trivial File Transfer Protocol

- TFTP ist ein vereinfachtes Datei-Übertragungs-Protokoll, das auf UDP aufbaut (FTP verwendet TCP) [70].
- Das Ziel von TFTP ist zum einen ein einfacher **Datentransfer** und zum anderen das Ermöglichen des **Bootens einer Workstation** ohne Diskettenlaufwerk.
- An TFTP gestellte Anforderungen sind, dass es kompakt sein soll (um ins ROM zu passen), keine Authentifikation.
- TFTP ist ein stop-and-wait Protokoll.
  - Kein neuer Protokollschritt, solange nicht der vorhergehende erfolgreich (korrekt) abgeschlossen ist.
  - Niedrige Protokoll-Leistung (Datendurchsatz).
  - Dafür einfache Implementierung.
  - Geänderte Reihenfolge stellen kein Problem dar, da stopand-wait Protokoll nur jeweils ein Paket hängig hat.
- Alternative
  - Positive Bestätigung kann verzögert werden (sliding window).
  - Übersendung mehrerer Pakete und dann erst warten auf eine Bestätigung.
  - Höherer Datentransfer (Beispiel TCP).
- Doppelte Pakete werden aufgrund der Blocknummer erkannt, während verlorene Pakete über Timeout beider Partner erkannt werden.
- TFTP Nachricht hat keine Checksumme, Datenveränderungen müssen durch die UDP-Checksumme abgefangen werden.
- Retransmission Timer
  - Wert sollte je timeout um bestimmten Faktor multipliziert werden (sog. Exponential backoff).
  - Retransmission timeout je Paket = 5 s.
  - Retransmission timeout für „Verbindung“= 25 s.



7 Routing

Routing optimiert den Weg zum Ziel einzelner Pakete. Dabei gewährt es Dienstgüten (z.B. Konstante Latenz, Bitrate, Durchsatz, sowie Priorisierung von Paketen) und minimiert Kosten.

7.1 Begriffe

Router - Gerät, welches mindestens zwei unterschiedliche Netzwerke verbindet. Ist in der Lage, den besten Weg für eine Nachricht durch ein Netzwerksystem bis zum Empfänger zu bestimmen.

Routing - bezeichnet das Transportieren Daten innerhalb eines Netzes

Passives/Source Routing - Route ist durch das zu versendende Datenpaket vorgegeben (z.B. im Header enthalten)

Aktives Routing - Router kann sich den Versandweg selbst aussuchen

Globales Routing - Ein Knoten hat vollständige Informationen zu Topologie und Kosten. Informationen über das komplette Netz sind vorhanden. Optimierung läuft an einer Stelle oder repliziert an mehreren.

Dezentrales Routing - Keine vollständigen Informationen in einem Knoten. Jeder Knoten kennt zunächst (initial) nur die Kosten verbundener Links. Berechnung erfolgt iterativ und dezentral.

Distanz-Vektor-Algorithmen - Kein Knoten hat Wissen über das komplette Netz und kennt daher nie die komplette Route von einer Quelle zu seiner Senke.

Link-State-Algorithmen - Knoten haben Wissen über die gesamte Netztopologie. Daher haben im stabilen Zustand alle Knoten die gleiche Sicht auf das Netz.

Statisches/nichtadaptives Routing - Ist beim Starten/Hochfahren des Systems fixiert.

Dynamisches/adaptives Routing - Routen ändern sich im laufenden Betrieb und passen sich so an Änderungen von Topologie, Netzlast, Kosten etc. an. Setzt zwingend ein Protokoll voraus.

7.2 Anforderungen

Die Anforderungen an Routing-Algorithmen sind:

Einfachheit - Algorithmus muss in kurzer Zeit mit wenig Ressourcen berechnet werden können.

Robustheit - Bewältigung von Änderungen der Topologie.

Stabilität - Erreichung eines stabilen Zustandes.

Fairness - Alle Knoten werden gleich bedient.

Optimalität - Gemäß der Kostenfunktion günstigster Weg wird gefunden.

Anmerkung: Ohne weiteres können sich diese Anforderungen widersprechen! So kann etwa die Kommunikation der Knoten $x \to x'$ die Kommunikation $a \to a', b \to b' undc \to c'$ stark negativ beeinflussen und damit den Gesamtdurchsatz (Optimalität) verringern.

7.3 Spannbäume

Optimalitätsprinzip: *Wenn ein Router J auf dem optimalen Pfad von Router I zu Router K liegt, dann gehört der optimale Weg von J nach K zum gleichen Pfad*[72]. So lässt sich eine allgemeine Aussage über optimale Pfade treffen, ohne dass konkrete Kenntnisse der Topologie des Netzwerkes vonnöten sind. Alle optimalen Routen von den Knoten zu einem bestimmten Ziel (Senke), bilden einen **Spannbaum**[11].

- Senke muss nicht zwangsläufig eindeutig sein (auf den gleichen Pfadstrecken kann es auch andere Bäume geben).
- Da Senke ein Baum ist, enthält sie keine Schleifen.
- Pakete werden auf einer endlichen, begrenzten Anzahl von Teilstrecken übertragen.

7.4 Algorithmen

Eingabe: Graph G des Netzes, wobei jeder Knoten im Graphen einen Router und jede Kante eine Übertragungsleitung darstellt.
Problemstellung: Finde für einen Startknoten s und einen Endknoten e eines gewichteten Graphen $G = (V, E)$“ und der Kostenfunktion $k(E)$, einen Weg zwischen s und e mit minimalen Kosten bezüglich k.

7.4.1 Dijkstra-Algorithmus und Link-State-Routing

Dijkstra arbeitet nur für nichtnegative Kantengewichte korrekt. Die Zeitkomplexität[12] beträgt $\mathcal{O}(n^2)$, bei der Verwendung eines Fibbonacci-Heaps kann diese auf $\mathcal{O}(m + n \log n)$ verbessert werden.

Dieser Algorithmus wird beim *Link-State-Routing* verwendet. Zur Durchführung werden komplette (Topologie- und) Kosteninformationen benötigt Dazu propagieren die Knoten die Kosten zu ihren Nachbarn mittels Broad-/und Multicasts im gesamten Netzwerk.

Beschreibung Dijkstra-Algorithmus nach [7]:

$d(v_1) \leftarrow 0$
for j := 2 to n **do** $d(v_j) \leftarrow \infty$
end for
$OFFEN \leftarrow V$
while $OFFEN \neq \emptyset$ **do**
  wähle ein $v \in OFFEN$ mit $d(v) = min\{d(w) : w \in OFFEN\})$
  $OFFEN \leftarrow OFFEN \backslash \{v\}$
  for all $w \in OFFEN$ mit $(v, w \in E)$ **do** $d(w) \leftarrow min\{(d(w), d(v) + k(v, w))\}$
  end for
end while

7.4.2 Oszillation

Oszillation ist ein Problem, welches bei zu häufiger Ausführung und der Berücksichtigung der Netzauslastung entsteht. Dabei schaltet der Verkehr ständig zwischen verschiedenen alternativen Pfaden hin und her. Grund hierfür ist die Reaktion auf die Auslastung der einzelnen Verbindungen: Wird der Verkehr über Alternative A geleitet und Alternative B hat eine geringere Auslastung, wird der Pfad entsprechend geändert. Dadurch ist die Auslastung von Alternative A nun geringer und es findet wieder ein Wechsel statt.
Diesem Phänomen lässt sich entgegenwirken, indem man a) die Netzauslastung nicht zur Berechnung der Routen mit einbezieht, wobei dies die Verwendung von Alternativrouten bei Lastspitzen verhindert, b) schnelle Änderungen der Routen verbietet (Hold-down), c) die durchschnittliche Auslastung (oder anderweitig aggregierte Werte) heranzieht[12] oder d) verhindern, dass Router den Algorithmus gleichzeitig durchführen, z.B. durch zufällige Wartezeiten.

7.5 Bellmann-Ford-Algorithmus und Distance-Vector-Routing

Beim Distance-Vector-Routing verwaltet jeder Knoten eine Routingtabelle mit Informationen zu bekannten Zielen. Zu jedem Ziel wird der zu wählende Ausgang[13] mit den damit verbundenen Kosten gespeichert. Es wird jeweils angenommen, dass es sich dabei um die beste (= günstigste) bekannte Weiterleitung handelt. Die Informationen werden an die direkten Nachbarn weitergeleitet. Nachbarn können so neue Ziele kennenlernen oder Routen zu bekannten Zielen verbessern. Errechnen lassen sich die Kosten zum Ziel als Summe der Kosten

[11] Im Tanenbaum[72], welcher für die Entstehung des Skripts offensichtlich Modell gestanden hat, wird dieser als **Quelle-Senke-Baum** (sink tree) bezeichnet
[12] Abhängig von Kantenzahl m und Knotenzahl n
[13] Interface, Next Hop, whatever

zum Erreichen des Nachbarn und dessen Kosten zum Erreichen des Ziels.
Distance-Vector-Algorithmen sind **verteilt**, **iterativ** und **asynchron**. Bei **Änderung von Kosten** läuft der Algorithmus neu an, bis er sich auf allen Knoten stabilisiert hat. Haben sich die Kosten verbessert, findet eine schnelle Adaption (**Good news travels fast**) statt, bei Verschlechterungen ist die Reaktionszeit relativ hoch ((**Bad news travels slow**)).

Beschreibung Bellmann-Ford-Algorithmus nach [7]:

$d(v_1) \leftarrow 0$
for j := 2 to n **do** $d(v_j) \leftarrow \infty$
end for
for $i \leftarrow 1$ to $|V| - 1$ **do**
  for all $(u, v) \in E$ **do**
    $d(v) \leftarrow min(d(v), d(u) + k(u, v))$
  end for
end for
for all $(u, v) \in E$ **do**
  if $d(v) > d(u) + k(u, v)$ **then return** FALSE
  end if
end for

7.5.1 Count-to-infinity-Problem

Hinweis: Zur Vereinfachung wird in diesem Abschnitt angenommen, dass als Metrik die Anzahl der Hops zum Ziel gewählt wird.
In bestimmten Fällen kann es bei Distance-Vector-Algorithmen dazu kommen, dass das Verfahren nicht konvergiert. Dies ist beispielsweise der Fall, wenn ein Router (oder jede Verbindung zu ihm) ausfällt. Die verbleibenden Router sind vom Ausfall nicht informiert, stattdessen erhalten sie weiterhin von ihren Nachbarn veraltete Informationen zur Entfernung.
Der vormals direkte Nachbar, welcher zuvor von einer Entfernung von 1 kannte, erhält nun von einem anderen Nachbarn die Information, dass dieser den Ausgefallenen mit Kosten von 2 erreichen kann. Dies führt dazu, dass die fehlerhafte Information übernommen wird und sich die Kosten — zumindest theoretisch — bis unendlich *hochschaukeln*.
Gegenmaßnahmen sind a) **Split Horizon** - hierbei werden Updates nie an den Nachbarn gesendet, von dem die beste Route gelernt wurde. Stattdessen gibt es Updates erst nach einem Timeout. Variante b) **Split Horizon with Poisoned Reverse** funktioniert, wie folgt: Ein Router A glaubt, dass er über einen Router B zum Ziel gelangt. Dann teilt A B mit, dass er selbst keine Route zum Ziel kennt. Dies kostet allerdings unnötig Bandbreite und funktioniert wohl in der Praxis nicht ganz so gut. Kein Plan.

7.6 Routing in MANETs

Ein Mobile Ad Hoc Network (MANET) ist ein drahtloses, selbstkonfigurierendes Netzwerk mobiler Knoten. Zumindest einige dieser mobilen Knoten müssen auch als Router arbeiten können. Alle Knoten bewegen sich unkoordiniert und nur schlecht vorhersagbar. MANETs können Übergänge in andere Netze haben, z.B. in das Internet.

7.6.1 Herausfoderungen und Probleme

:

- Netzwerke ohne feste Infrastruktur.
- Knoten sind in der Regel mobil, daher können keine statischen Routen verwendet werden.
- Zielfunktion des Routings enthält weitere Faktoren
  - Energieeffizienz
  - Durchsatz einzelner Strecken
  - Gesamtdurchsatz
  - QoS für bestimmte Dienste

Beim Routing in MANETs bestehen insbesondere Probleme bei der **Konvergenz** (Routen müssen gefunden werden, bevor Knoten wieder außer Reichweite sind) und der **Skalierbarkeit** (Overhead durch Routing).

7.6.2 Lösungsansätze

- Einfache Ansätze
  - Reaktiv (bei Bedarf), Suche nach dem Ziel
  - Proaktiv (auf Vorrat), Regelmäßige Verbreitung aller Pfade
  - Flooding
    - Nachricht wird in alle Richtungen verschickt
    - Bisheriger Weg wird gespeichert
    - Nachricht, die zuerst am Ziel ankommt enthält die günstigste Route
- Erweiterte Ansätze
  - Auswahl besonderer Knoten als Router
  - Geografisches Routing
  - Vorausberechnung der Knotenbewegung
  - Scalable Source Routing[32] (Nicht im Skript)

7.6.3 Backbone-Netze

Auswahl bestimmter Knoten als Router. Jeder Knoten hat genau einen Backbone-Nachbarn. Berechnung des Minimum Dominating Sets[14] ist schwieriges Problem (NP-vollständig). Jeder Knoten außerhalb des Backbones hat ein *default gateway* im Backbone. Routing im Backbone nach den üblichen Routing-Verfahren. Erhaltung des Backbones bei wegfallenden Verbindungen ist ebenso ein schwieriges Problem.

7.6.4 Proaktives und Reaktives Routing

	Vorteile	**Nachteile**
Reaktiv	Geringe Latenz durch Routing und Zustandsinformationen über alle Verbindungen liegen vor	Overhead und Reparatur von Routen hängt von der Aktualisierungshäufigkeit ab
Proaktiv	Kein Overhead für das Vorausberechnen der Routen	Latenz durch Routenberechnung

Geometric Routing

- Alle Knoten kennen ihre Position z.B. GPS bei mobilen Knoten
- Verschiedene Varianten
  - Wahl des Knotens, der am nächsten an der „Luftlinie“ liegt.
  - Wahl des Nachbarknotens, der möglichst nahe am Ziel liegt.
  - Wahl des Knotens, der in möglichst derselben Richtung wie das Ziel liegt.

7.7 Routing-Protokolle

8 Quality of Service

8.1 Qualitätsparameter

- Datendurchsatz (Throughput, Bandbreite (minimale, mittlere, maximale)
- Paketverzögerung / Latenzzeit (inklusive zeitlicher Schwankungen / Jitter)
- Zuverlässigkeit (gegenüber Störungen, Ausfällen und Übertragungsfehlern)
- Zeitbedarf (Antwortzeit, Verbindungsaufbauzeit, Haltezeit)
- Verfügbarkeit (Fehlerraten, Paketverlustrate)

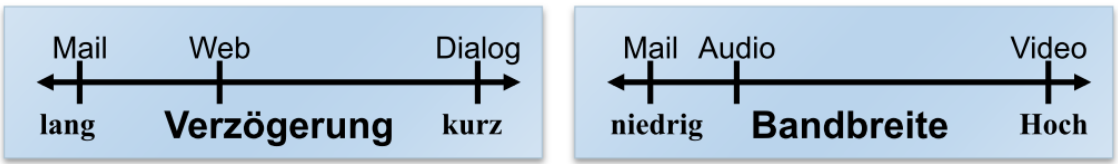
[14] Eine dominierende Menge eines Graphen ist eine Teilmenge D von V, sodass jedes $v \in V \setminus D$ einen Nachbarn in D hat.

8.2 Möglichkeiten zur Qualitätsverbesserung

1. Reduzierung der Komplexität
   (Schaffung einer Struktur mit kurzen Wegen zwischen den Komponenten und die Vermeidung von unnötigen Routern.)
2. Bandbreite erhöhen
   - Neuere Netzwerktechnologie
     (Höhere Frequenzen und bessere Modulationsverfahren)
   - Leistungsfähige Technik
   - Full-Duplex
   - Alternative Verbindungen
3. Geschickte Verteilung der vorhandenen Bandbreite des Internets.
4. Priorisierung des Datenverkehrs
5. QoS Systeme managen die zur Verfügung stehende Bandbreite
6. Leistungsüberwachung durch folgende Datenquellen:
   - Reaktion der Benutzer
   - Eigene Tests (Durchsatzmessungen, Latenzmessungen, Messungen wiederholen)
   - Ausgaben der Geräte (Geschwindigkeit, Latenz, Paketverlust)

Diese Maßnahmen lassen sich nicht immer umsetzen!

8.3 Beispiele für die Priorisierung



8.4 Strategien

Für jedes Interface sind verschiedene Strategien möglich.

8.4.1 FIFO (First In First Out) Queuing

- Abarbeitung nach der Reihenfolge des Eintreffens
- Vorteil: Einfach zu implementieren und zu modellieren
- Nachteile:
  - Nur eine Serviceklasse möglich, erlaubt keinerlei QoS.
  - Keine Priorisierung von Paketen gegenüber anderen.

8.4.2 Weighted Fair Queuing (WFQ)

- Mehrere Queues mit fairem Multiplexing
- Aufteilen der Bandbreite nach Gewichtung.
- Keine Queue „verhungert“, weil die anderen Queues eine hohe Last haben.
- Bewertung
  - Korrekte Zuordnung von Gewichten ist oft schwierig.
  - Datenstrom soll nicht höhere Gewichtung anfordern können als erforderlich.
  - Keine „Leer-Reservation“– sobald eine Queue leer wird, wird dessen Bandbreite für die anderen Queues genutzt.

8.4.3 Priority Queuing - Prioritätswarteschlangen

- Mehrere Queues mit Priorisierung
- Zuweisung der gesamten Bandbreite an jeweils höchstpriorisierte Queue (bis diese leer ist)
- Wichtiger Verkehr erhält stets genügend Bandbreite, kann allerdings hierdurch weniger wichtigen Verkehr (dauerhaft) blockieren.
- Ermöglichen die Bevorzugung einzelner Übertragungen
- Weniger wichtige Übertragungen werden eventuell verdrängt
- Bewertung
  - Einfach zu implementieren
  - Die Blockierung weniger wichtigen Verkehrs kann problematisch sein.
  - Verfahren ist sinnvoll, wenn höchstpriorisierter Verkehr die geringste Datenmenge anliefert und die zugehörige Queue immer wieder geleert wird.

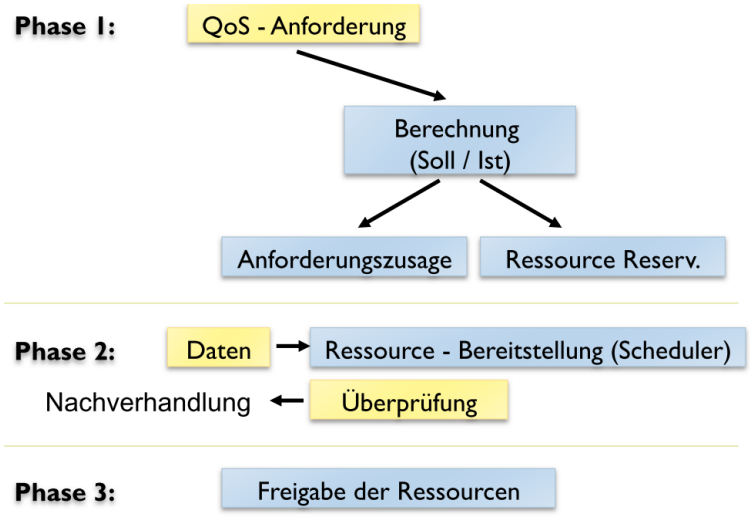
8.4.4 Custom / Class Based Queuing (CBQ)

- Mehrere Queues mit unterschiedlichen Prioritäten.
- Datagramme werden entsprechend ihrer Priorität / TOS in die Queues eingereiht.
- Benutzer gibt explizit die Aufteilung der Bandbreite an.
- Angabe als Anzahl Bytes, die zu übertragen sind, bevor die nächste Queue gewechselt wird (transmission window size).

8.4.5 Random Early Detection / Discard (RED)

- Lässt an bestimmten Punkten im Netzwerk wird überwacht (bspw. zentraler Router).
- RED wurde hauptsächlich für TCP-Anwendungen entwickelt.
- Funktionsweise
  - Zufällig werden Pakete verworfen, wenn Überlast auftritt.
  - Dabei werden Pakete aller Verbindungen gleichermaßen verworfen, damit nicht einzelne Verbindungen „absterben“.
- Absenderknoten erkennen verworfenen Pakete
  - TCP reduziert Window Size wenn die Verbindungsqualität schlechter wird.
  - Dann wird schrittweise die Datenrate erhöht, bis wieder Überlast auftritt.

8.5 QoS Ablauf



9 Multicast

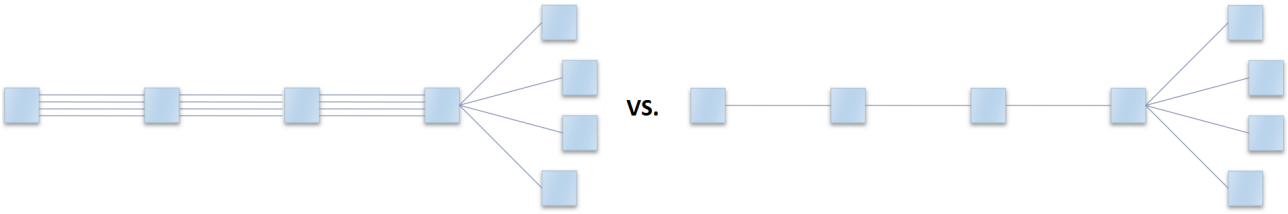
Multicasts stellen höhere Anforderungen an Router und es sind zusätzliche Protokolle notwendig.

9.1 Motivation

Mulitcasts bieten sich dann an, wenn es das Ziel ist, mit so geringem Aufwand wie möglich so viel wie möglich Beteiligte zu erreichen. Wenn es bspw. das Ziel ist, dass mehrere Hörer ein Radioprogramm empfangen gibt es zwei Möglichkeiten:

- Möglichkeit A – Daten werden für jeden Empfänger einzeln geschickt.
- Möglichkeit B – Daten werden möglichst weit nur einmal übertragen und dort, wo sie unterschiedliche Wege gehen müssen, werden sie vervielfacht und verteilt.

Multicasts können als Möglichkeit zum Sparen von Bandbreite durch Zusammenfassen von Datenübertragungen betrachtet werden.

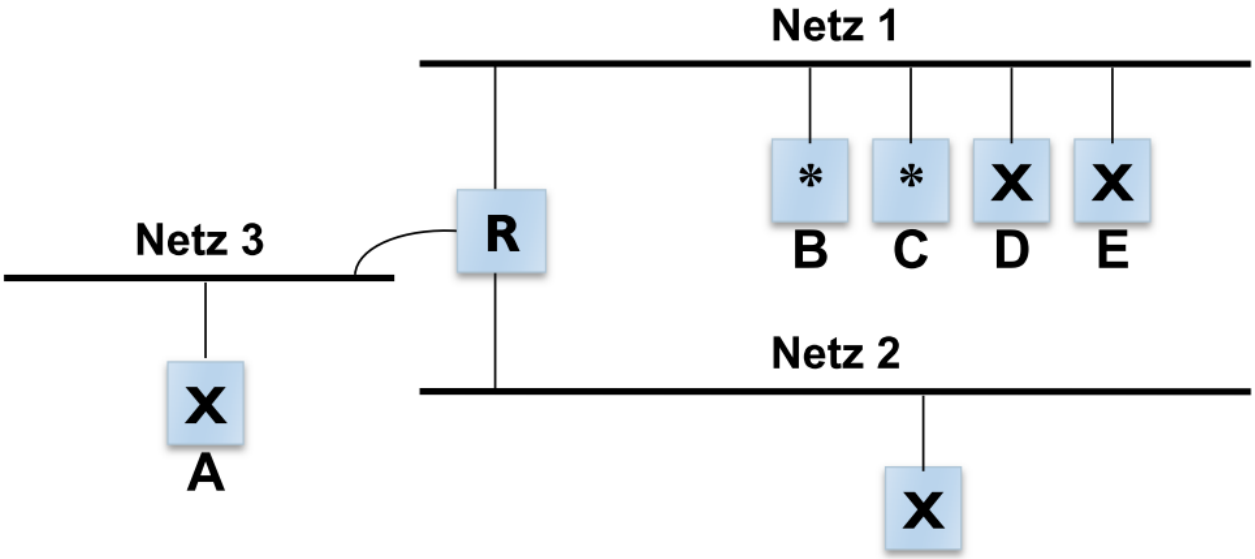


9.2 Voraussetzungen

- Für Multicasts werden spezielle **Router und Routing-Protokolle** benötigt. Diese werden wie üblich durch Berechnung eines Spannbaumes ermittelt. Es kann problematisch werden, wenn mehrere Sender an die Gruppe senden.
- Es ist notwendig die **Gruppenzugehörigkeit** zu übermitteln. Dazu sind spezielle Protokolle notwendig. Ein Rechner, der sich bei einer Gruppe an- oder abmelden mlchte, teilt dies nur seinem lokalen Router mit. Es wird nur die Gruppenadresse übergeben (keine weiteren Informationen). Anschließend muss der Router sich selbst darum kümmern, woher er die Multicastdaten im Internet bekommt.
- Es findet eine angepasste Flusskontrolle statt. In der Regel erfolgt keine Neuübertragung bei negativen Bestätigungen. Außerdem ist eine positive Bestätigungen unpraktikabel.

9.3 Komplexeres Multicast-Routing

Zu komplexeren Formen des Multicastings kann es kommen, wenn mehrere Sender existieren:



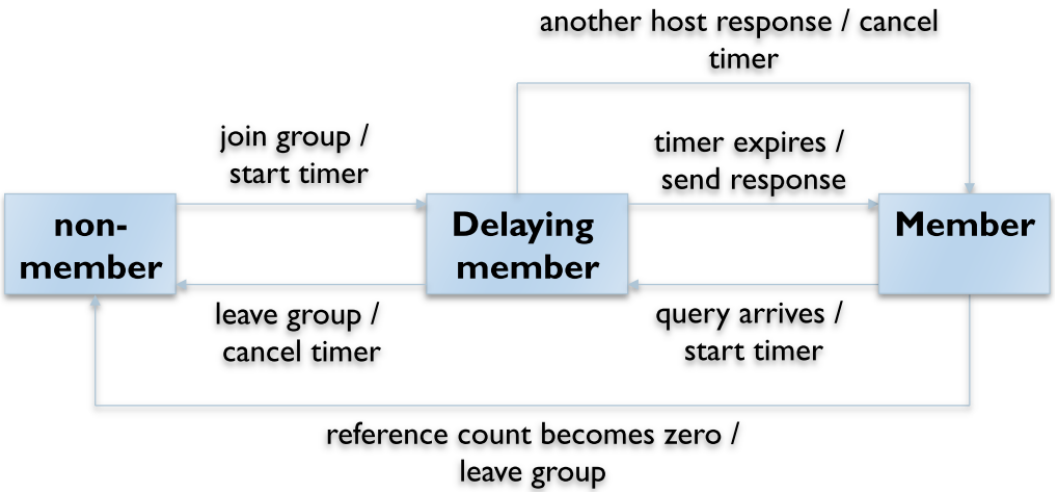
- Sendet A an die Gruppe X, muss Paket in die Netze 1 und 2 weitergeleitet werden.
- Sendet B an die Gruppe X, muss das Paket in die Netze 2 und 3 weitergeleitet werden.

9.4 IGMP

IGMP (Internet Group Management Protocol) ist ein Protokoll zur Steuerung von Multicast-Gruppen. IGMP ist im RFC 3376[9] in der aktuellen Version 3 beschrieben. Mit Hilfe von IGMP teilen Hosts und Router dem Multicast-Router ihre Mitgliedschaft in Multicast-Gruppen mit. Außerdem fragen Router in „ihren“Netzen nach Gruppenmitgliedern.

9.5 Ablauf

- Router sendet allgemeine IGMP Query an seine Interfaces. Zu Beginn ist jeder Router anfangs Querier. Entdeckt ein Router einen Querier mit niedrigerer IP-Adresse, so wird der Router mit der höheren IP-Adresse zum Non-Querier.
- Empfängt ein Host eine Query, startet er einen Timer mit zufälligem Interval zwischen 0 und dem vom Querier vorgegebenen Maximum für jede Gruppe, in der er Mitglied ist.
- Wenn einer Timer abgelaufen ist, sendet der Host einen Report mit TTL=1 an die Gruppe.
- Andere Hosts senden dann keine weiteren Reports, da der Router dieses Interface (Netz) bereits „versorgen“wird.
- Empfängt ein Router einen Report, startet er einen Timer für die jeweilige Gruppe.
- Erneute Reports starten den Timer neu.
- Solange der Timer nicht abläuft, bleibt die Gruppenmitgliedschaft für das jeweilige Interface erhalten.
- Hosts sollen Anfangs Reports ohne Query senden, um sofort die Mitgliedschaft anzuzeigen.



IGMP Ablauf (vollständig für v.1)

9.6 Nachrichten

IGMP v.1

- Kennt lediglich die Abonnierung einer Gruppe.
- Membership Reports.

IGMP v.2

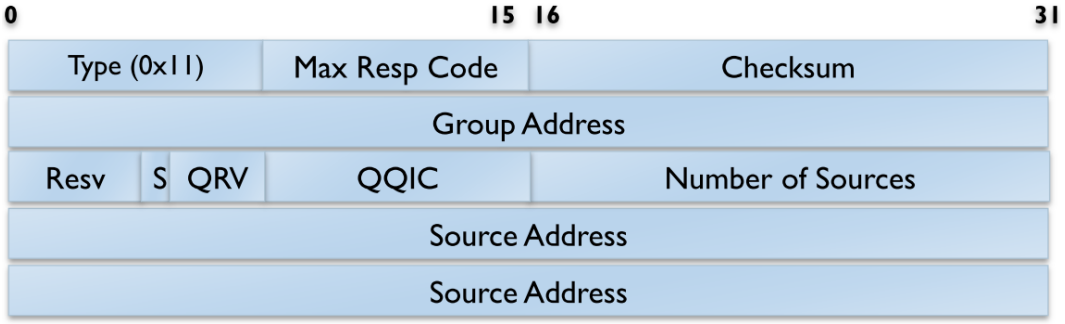
- Explizites Verlassen der Gruppe möglich.
- Leave

IGMP v.3

- Quelle der Multicasts kann angegeben werden.

9.7 Paketaufbau

9.7.1 Membership Query



Max Resp Code Maximum der zufällig zu wählenden Zeit in Zehntelsekunden, bevor ein Report gesendet werden muss.

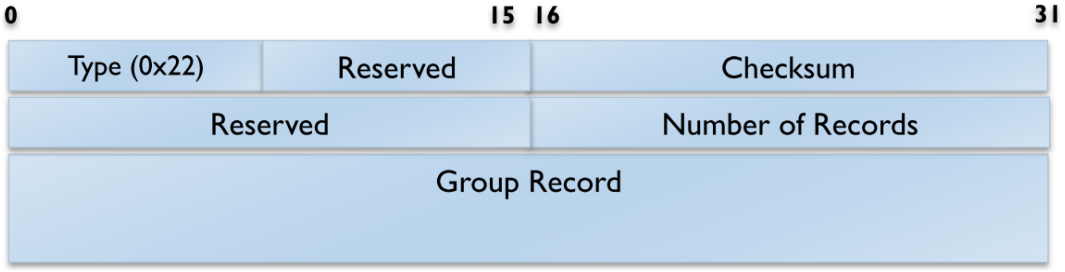
Group Address Multicast-Gruppe, auf die sich die Query bezieht. 0, falls es eine allgemeine Query ist.

S Flag, das den Multicast-Router auffordert, seinen Timer für Queries nicht neu zu starten.

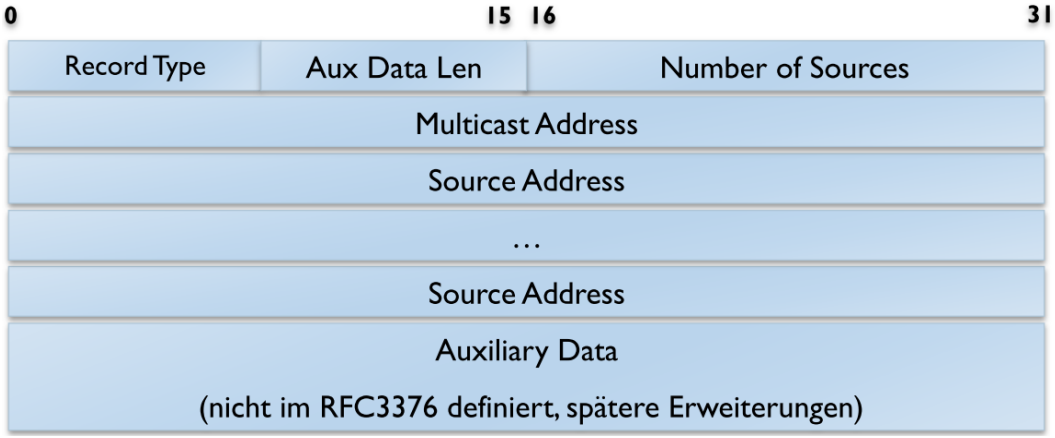
QQIC „Querier's Query Interval Code"- Zeit, nach der der Multicast-Router eine Query absendet. Wird verwendet, um festzustellen, ob der aktuelle Querier im Netz noch aktiv ist.

9.7.2 Membership Reports

In Version 3 wird dieses Paket an 224.0.0.22 oder an die entsprechende Multicast-Gruppe gesendet.



Dabeu enthält der **Group Record** nähere Informationen zum Report. Insbesondere die Information, ob eine Gruppenmitgliedschaft noch erwünscht ist.



RecordType Umgang mit empfangenen Source Adresse (Multicasts nachfolgender Source Adressen werden empfangen/ignoriert; Hinzufügen/Entfernen von akzeptierten Source Adressen)

Source Address ist ein gültiger Absender eines Multicasts

9.8 Multicasts und Switches

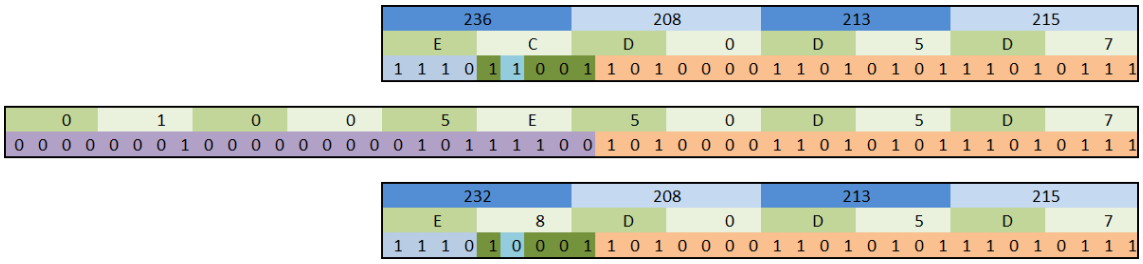
Ein Switch muss Multicasts an alle Mitglieder der Gruppe weiterleiten. Ein regulärer Switch müsste Multicasts wie Broadcasts behandeln. Um die Netzlast zu reduzieren, wird eine Technik namens **IGMP Snooping** angewendet.

- Der Switch belauscht (snoop - schnüffeln) den IGMP-Traffic an seinen Ports.
- Switch empfängt IGMP Membership Reports (Pakete auf Layer 3).
- Switch fügt den Port (im Sinne von Steckdose) zur Multicast-Gruppe hinzu.
- Nur Ports die einer Multicast-Gruppe beigetreten sind (join) werden in die Forwarding-Table für diese Multicast-Adresse eingetragen. Ports die die Gruppe verlassen (leave) werden aus der Tabelle gelöscht.

9.9 Multicast-Adressen

Auf der Vermittlungsschicht werden **Class-D-IP-Adressen** verwendet. Diese beginnen mit der Bitfolge `1110`, die Teilmenge `224.0.0.0/24` ist für Multicast in lokalen Netzen verfügbar[15]. Jede Adresse stellt dabei eine Multicast-Gruppe dar, folglich sind $2^{28} \approx 250$ Mio. Gruppen möglich.

Bei lokaler Kommunikation via Ethernet werden besondere **Multicast-MAC-Adressen** verwendet. Diese beginnen mit der festen Bytefolge `01005E` (3 Byte). Das vierte Byte beginnt mit dem Bit `0`. Die übrigen 23 Bit werden von den 23 untersten Bits der Multicast-Adresse übernommen. 5 Bits aus der Multicast-Gruppe können dabei nicht im Ethernet-Frame abgebildet werden.



Die Multicast-Adressen 236.208.213.215 und 232.208.213.215 werden auf dieselbe Mac-Adresse abgebildet.

10 Zeitsynchronisation

10.1 Notwendigkeit der Zeitsynchronisation

Authentisierungsverfahren müssen die Gültigkeit von Schlüsseln überprüfen können. Ebenso haben einige Abläufe eine beschränkte zeitliche Dauer. Andere Verfahren benötigen Time-Stamps, Vorher-Nachher-Beziehungen von Ereignissen müssen bekannt sein.

Uhren in verteilten Systemen sind nicht zwangsweise synchron. Sie können unterschiedliche Zeiten aufweisen, je nach Art der Uhr *ticken* sie unterschiedlich schnell (Quarzuhr, Atomuhr). Zusätzlich kann die physikalische Umgebung diesen Effekt verstärken (z.B. Temperatur einer Quarzuhr). Eigenschaften von Uhren nach Mühl sind: Drift, Auflösung und Abweichung (von der Realzeit).

10.2 Logische Zeit

Manchmal reicht es, nicht den genauen Zeitpunkt von Ereignissen, sondern lediglich die Vorher-Nachher-Beziehung zu kennen. Man spricht von **Lamport-Zeitstempel** oder **Vektoruhren**. Empfangene Lamport-Zeitstempel werden einfach bei jedem Kommunikationsschritt um eins inkrementiert und dann als eigener Zeitstempel versendet. Bei der Vektor-Zeit werden zusätzlich die Zeitstempel der anderen Kommunikationspartner mitgeführt. Dies hat den Vorteil, dass ein kausaler Zusammenhang immer erkennbar ist.

10.3 Herausforderungen

- Logische Zeitstempel (Lamport-Zeitstempel und Vektor-Zeit) genügen den Anforderungen nicht immer.
- Viele Uhren müssen synchronisiert werden.
- Die Übertragung eines Zeitstempels kostet Zeit – Latenz.
- Die Latenz ist nicht konstant – Jitter.
- Die Latenz ist nicht symmetrisch.

[15] Es wird kein Routing-Protokoll benötigt[72]

10.4 Algorithmen

Anforderungen an Algorithmen sind symmetrische oder bekannte Laufzeiten und eine konstante Latenz. Letzteres ist in paketvermittelnden Netzen nicht erreichbar.

10.4.1 Algorithmus von Cristian

Clients fragen beim Server nach der korrekten Zeit. Server kennt die Zeit aus zuverlässiger, externer Quelle.

10.4.2 Berkeley-Algorithmus

Server fragt alle verfügbaren Clients nach der Zeit und bildet den Mittelwert. Zeit wird anschließend an Clients verbreitet.

10.4.3 Marzullo's Algorithmus

Versucht den Einfluss des Jitters durch mehrfache Messungen zu eliminieren. Messungen werden solange durchgeführt, bis das Konfidenzinterval den Anforderungen entspricht.

10.5 NTP

NTP[47, 46] verwendet Marzullo's Algorithmus. Es löste den ICMP Timestamp Request (Abschnitt 10.7) ab und ist mittlerweile in Version 4 aktuell. NTP-Server lauschen auf UDP-Port 123. Im Internet sind Genauigkeiten von $\leq 10ms$ erreichbar.

10.5.1 Paketaufbau

Anmerkung: Paketaufbau von Version 4 scheint sich recht stark von Version 3 zu unterscheiden. Ich bleibe bei der in der Vorlesung besprochenen Version 3.

Leap Indicator : Gibt an, ob Schaltsekunde in dieser Minute hinzugefügt oder entfernt wird (2 Bit)

Status : Gibt mögiche Fehler an.

0. clock operating correctly
1. carrier loss
2. synch loss
3. format error
4. interface (Type 1) or link (Type 2) failure

Type : Typ[16] der Referenzuhr: 1 — Primärreferenz (z.B. Atomuhr) bis 4 — *Eyeball and wrist watch*

Precision : Angabe der Genauigkeit der Uhr (als Exponent zur Basis 2)

Estimated Error : Geschätzter Fehler zum Zeitpunkt der Synchronisation in Sekunden.

Estimated Drift Rate : Geschätzte Drift der Uhr (ohne Einheit).

Reference Timestamp : Zeit, auf den die Uhr bei der letzten Synchronisation eingestellt wurde.

Originate Timestamp : Zeit beim Senden des Pakets

Receive Timestamp : Lokale Zeit bei Ankunft des Pakets

Transmit Timestamp : Lokale Zeit beim Senden der Anwort

10.6 SNTP

Vereinfachung von NTP, bei der auf eine Wiederholung der Zeitsynchronisation verzichtet wird. So sind etwa Synchronisationen über Multi- und Broadcasts möglich. Es werden dasselbe Nachrichtenformat und derselbe UDP-Port, wie bei NTP genutzt[48].

[16] Neuere Versionen bezeichnen dies als Stratum; Es gibt quasi eine Hierarchie bei den (Genauigkeiten der) NTP-Server

10.7 ICMP - Timestamp Request und Reply

Timestamp Request[61] ermöglicht die Anfrage eines anderen Systems nach der aktuellen Zeit. Es handelt sich um ICMP-Nachrichten mit den Type-Werten 13 (Request) oder 14 (Reply)[17]. Empfohlener Rückgabewert sind die Millisekunden seit Mitternacht (Coordinated Universal Time - UTC). ICMP-Data-Bereich kennt Felder **Originate Timestamp**, **Receive Timestamp** und **Transmit Timestamp** (je 32 Bit); diese scheinen dieselbe Bedeutung wie bei NTP zu haben. Außerdem **Identifier** und **Sequence Number**, diese gestatten dem Sender eine Zuordnung von Replies bei mehreren Requests. Letztere werden vom Sender eingetragen und vom Empfänger in die Antwort kopiert.

11 Internet Control Message Protocol

ICMP ist ein Layer-3-Protokoll zur Steuerung des Nachrichtenaustausches im Internet[61][18]. Es dient häuptsächlich dem Austausch von Status- und Fehlermeldungen zwischen Gateways und Hosts. Die Nachrichten werden über IP-Datagramme übertragen, das Protokoll ist verbindungslos.

11.1 Paketaufbau

Eine ICMP-Nachricht besteht aus den Feldern **Type**, **Code**, **Checksum** und **Data**, angeführt von einem 20 Byte IP-Header. Der IP-Header hat häufig vorbestimmte Feldwerte, z.B. Type of Service = 0, weitere Infos im RFC.

Type : Bestimmt das Format der weiteren Felder. (1 Byte)

Code : Inhalt Abhängig vom Type (1 Byte)

Checksum : Prüfsumme[19] (2 Byte)

Data : Nicht immer genutzt; Enthält beispielsweise bei Type *Redirect Message* (5) die *Gateway Internet Address* oder die in Abschnitt 10.7 beschriebenen Felder für Timestamps.

11.2 Beispiele

Im Skript existiert eine umfangreiche Tabelle von Funktionen und entsprechender Werte für Type- und Code-Felder. Diese auswendig zu lernen entspräche nicht 80-20. Daher beschränke ich mich hier nur auf zwei Beispiele (+ Zeitsynchronisation in Abschnitt 10.7)

11.2.1 Destination Unreachable

Wird beispielsweise gesendet, wenn er Empfänger laut den Routing-Tabellen eines Gateways nicht erreichbar ist, z.B. weil die Distanz zum Zielnetzwerk unendlich ist. Empfänger im IP-Header ist der ursprüngliche Absender einer (nicht zustellbaren) Nachricht. Der ICMP-**Type** ist 3, der Code gibt genauere Informationen zur Fehlerursache (z.B. 3 — port unreachable)

11.2.2 Ping

Beispiel nicht aus dem Skript. Beim Request wird ein **Echo-Request** (Type 8) versendet. Das Feld **Code** enthält den Wert 0[20]. **Identifier** und **Sequence Number** helfen bei Unterscheidung mehrerer Requests/Replys. Im **Data** Feld kann scheinbar noch unbestimmter Payload mitgesendet werden.

Bei der Antwort werden Sender- und Empfängeradresse im IP-Header vertauscht. Der **Type** ist 0 (Echo-Reply). Weitere Felder werden aus dem Request übernommen.

11.3 Regeln

ICMP-Fehler-Nachrichten werden nie als Folge auf

- eine ICMP Fehlermeldung („Teufelskreislauf“) erzeugt.
- ein Datagram, das an eine IP Broadcast Adresse oder eine IP Multicast - Adresse (class D) gesendet wird, erzeugt.

[17] Skript sagt hier 17 und 18, im RFC stehen allerdings 13 und 14. Habe Thomas drauf hingewiesen.

[18] Laut Folien RFC 950[51], aber das ist Quatsch (*Internet Standard Subnetting Procedure*)

[19] Scheinbar abhängig vom Type; häufig: *The checksum is the 16-bit ones's complement of the one's complement sum of the ICMP message starting with the ICMP Type.*

[20] Laut RFC scheint die Möglichkeit zu bestehen, dass hier etwas anderes steht. Warum das so ist, kann ich allerdings nicht finden

- ein Datagram, welches als Link - Layer Broadcast gesendet wird, erzeugt.
- ein anderes Fragment als des ersten erzeugt.
- ein Paket erzeugt, dessen Quelladresse nicht einen einzelnen Host definiert.

12 Internet Group Message Protocol

13 Voice over IP

VoIP ist kostengünstig, da taugliche IP-Geräte verbreitet und auf Grund der hohen Stückzahlen billig zu produzieren sind. Außerdem sind die Betriebskosten gering.
Des Weiteren gibt es eine Vielzahl an Gründen für VoIP:

- Integration von Sprach- und Datendiensten.
- Leichtere Implementierung komplexerer Anwendungen (z.B. Call Center).
- Potentiell geringerer Bandbreitenbedarf.
- Verfügbarkeit von IP-Netzen.

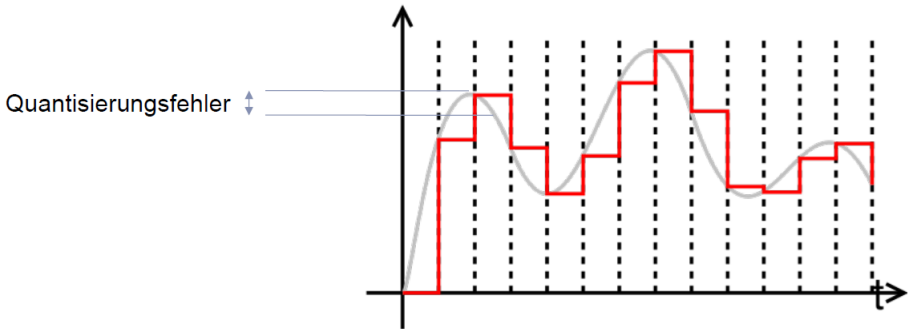
Wichtige Aspekte sind die Digitalisierung der Sprache (Bandbreite, Gesprächsqualität) und die eigentliche Übertragung der Sprache (Quality of Service / Priorisierung).

13.1 Digitalisierung

Zur Digitalisierung der Sprache bestehen verschiedene Möglichkeiten:

- Zählverfahren - „Hochzählen“ um jeweils einen Spannungsschritt und Vergleichen mit dem Input.
- Sukzessive Approximation – schrittweise Annäherung der Spannung entsprechend der Bitwertigkeit mittels D-A-Wandler und Vergleich (Komperator).
- Parallel-Umsetzer – ein Komperator pro möglichem Eingangswert.

Die Digitalisierung erfolgt mittels Quantisierung. Durch begrenzte Auflösung des A-D-Wandlers kommt es zu einem Quantisierungsrauschen. Dadurch entstehen Quantisierungsfehler:



13.2 Sprachqualität

Eine Aussage über die subjektive Sprachqualität wird in der Regel über „Testhörer“getroffen. Andere Methoden vergleichen das Eingangs- und das Ausgangssignal algorithmisch, z.B. Perceptual Speech Quality Measurement.
Die Sprachqualität hängt von verschiedenen Faktoren ab:

- Latenz
  (ITU-T G.114 empfiehlt maximal 300ms, andere Empfehlungen unter 150ms.)
- Jitter
- Packet Loss
- Quantisierung
- Abtastfrequenz
- Art der Sprachkodierung

Die Sprachqualität wird in MOS (Mean Opinion Score) angegeben.

13.3 Sprachkodierung

Die folgenden Kodierungen können durch verschiedene „Trick“verbessert werden:

- Silence Detection (auch Abschaltung des Senders DTX bei GSM).
- Comfort Noice (teilweise mit unterschiedlichen Rauschmodellen).
- Ein Problem bei der Mehrfachkodierung ist, dass falls mehrfach der Codec gewechselt wird, die Sprachqualität stark abnimmt (z.B. bei Netzübergängen).

13.3.1 Pulse Code Modulation (PCM)

- Waveform Codec - Abtastung und Erzeugung unkomprimierter, unveränderter Daten.
- PCM ist ein Pulsmodulationverfahren, das ein zeit- und wertkontinuierliches analoges Signal in ein zeit- und wertdiskretes digitales Signal umsetzt.
- G.711 zum Beispiel mit 8000Hz, 8bit (64kbit/s netto)
- Funktionsweise:
  1. Abtastung des analogen Signals mit einer zeitlich konstanten Abtastrate. Dabei wird aus dem zeitkontinuierlichen Signalverlauf eine zeitdiskrete Signalfolge gebildet. Zur Erhaltung der Information in der zeitdiskreten Folge ist die Erfüllung des Nyquist-Shannon-Abtasttheorems notwendig. Dies bedeutet, dass die Abtastrate mehr als doppelt so groß sein muss, wie die im Signalverlauf höchste vorkommende Frequenzkomponente ist.
  2. Quantisierung auf diskrete Werte mit endlich vielen Stellen.
  3. Erzeugung des Digitalsignals mittels Codierung.

13.3.2 Differential Pulse Code Modulation (DPCM)

- Die Kodierung wird teilweise auch Delta PCM genannt und es handelt sich um einen Waveform Codec.
- Wegen der relativ langsamen Änderungen des Eingangssignals lassen sich einige Werte mit annehmbarer Genauigkeit vorhersagen.
- DPCM überträgt Differenz zwischen Sample n und Sample n+1.
- Bei einigen Varianten von DPCM werden lediglich die Differenzen zwischen einer Vorhersage und Messwert übertragen.
- Gleicher Informationsgehalt, wie PCM, allerdings reduzierte Datenübertragung.

13.3.3 Adaptive Differential Pulse Code Modulation (ADPCM)

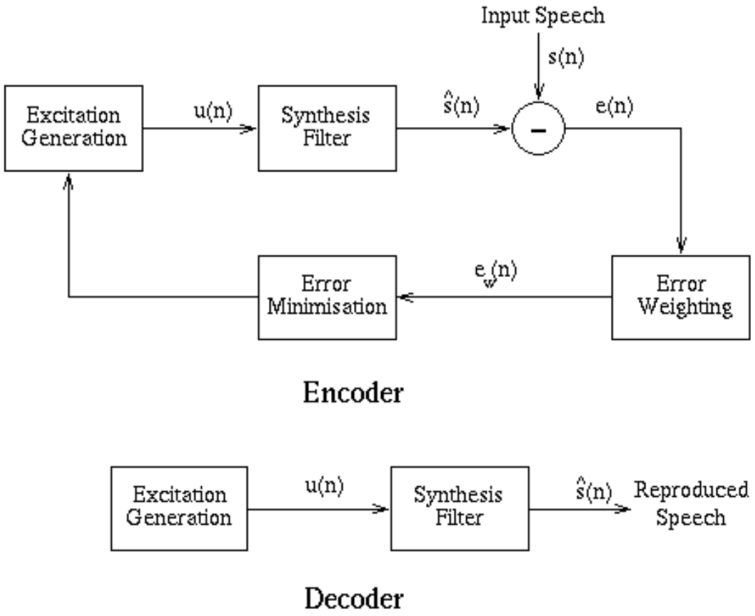
- Wie DPCM, allerdings mit dynamischer Anpassung der Quantisierungsschritte.
- Verlustbehaftet
- G.726 mit 32kbit/s hat MOS von 4,0.

13.3.4 Vocoder

- Einige Codecs versuchen, ein mathematisches Modell der an der Spracherzeugung beteiligten Vokaltrakts zu erstellen.
- Es werden die „Einstellungen“des Vokaltrakts übertragen.
- Es ist jedoch keine natürlich klingende Ausgabe möglich.

13.3.5 Hybride Codecs

- Hybride Codecs kombinieren beide Varianten und benutzen Teilinformationen aus Waveform Codecs und von Vocodern.
- Erzeugen immer kleine Verzögerung durch blockweise Bearbeitung der Daten.
- G.728 – Low Delay Code-Excited Linear Predictive, MOS 3,9 bei 16kbit/s.
- G.723 – Algebraic Code-Excited Linear Prediction – 30ms Verzögerung.
- GSM-Codec.



13.4 Aufbau eines VoIP-Systems

Es ist notwendig, dass neben dem Austausch der reinen Sprachdaten (oder Audio-Video-Daten) eine Möglichkeit zur Steuerung besteht. Die folgenden Punkte sind dabei zu beachten:

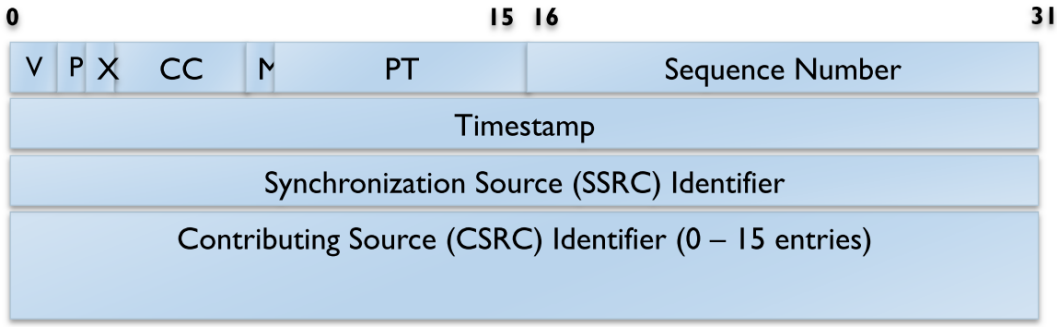
- Gesprächsauf- / -abbau.
- Umleitungen.
- Ansagen / Anrufbeantworter.
- Konferenzschaltung
- Übergang in andere Netze.
- Ressourcenverwaltung / QoS-Steuerung.
- Abrechnung / Logging / „Vorratsdatenspeicherung“.

13.5 Migration

Zur Zeit existieren unterschiedliche Telefonnetze, Anbieter und Standards. Deshalb ist eine Umstellung auf VoiIP nicht instantan möglich. Die Umstellung erfordert eine schrittweise Migration: Zeitweiser Parallelbetrieb; Spezielle Telefon-Router zur Kostenoptimierung; Umsetzer zwischen verschiedenen Standards und ein Austausch von Telefonanlagen.

13.6 RTP

- RTP (Real Time Transport Protocol) ist ein Protokoll zum Streaming von Daten.
- RTP dient der Übertragung von Daten und optional kann RTCP genutzt werden, um Feedback für QoS zu erhalten.
- Basiert auf UDP, Port frei wählbar, default 5004 und 5005.



V - Version 2

P - Padding Gibt an, ob an Ende des Datenpakets Padding-Bytes enthalten sind. Falls gesetzt, enthält das letzte Byte im Payload die Anzahl der Padding-Bytes.

X - Extension Header-Erweiterung vorhanden.

CC - SRC-Count Anzahl der CSRS-Felder

M – Marker Verschiedene Einsatzmöglichkeiten, gemäß RFC1890 bspw. „Silence im Datenstrom möglich“.

PT – Payload Type Typ der Daten (G.726-32, GSM, 9 – G.722, QCELP, JPEG Video)

Timestamp Zeitpunkt des ersten Samples.

Synchronization Source Normalerweise der Sender der Nachricht, derjenige, der Timestamps setzt.

CSRC Im Falle, das verschiedene Signale gemischt werden (Mixer, z.B. für Konferenzgespräche), werden die einzelnen Quellen hier aufgeführt.

- RTP wird häufig auch als Multicast versendet. RTCP Reports werden dann in der Regel nicht ausgewertet.

13.7 RTCP

- das RTCP (RealTime Control Protocol) definiert verschiedene Kontrollnachrichten. Dabei können mehrere Kontrollnachrichten (RTCP Packets) in einem Compound Packet zusammen versendet werden.
- Sender Report (SR) kann bspw. enthalten:
  - NTP Timestamp
  - RTP Timestamp
  - Fraction Lost (Anteil der verlorenen Pakete seit dem letzten Report) – in n von 256.
  - Cumulative Number of Packets Lost
  - Interarrival Jitter.
  - Last SR Timestamp.
- Receiver Report (RR) enthält bspw. Interarrival Jitter
- Source Description (SDES)
- BYE
- APP (Application Specific Function)

13.8 Protkolle zur Übertragung von Steuerungsinformationen

13.8.1 SIP

SIP (Session Initiation Protocol)[36] dient nicht zur Übertragung multimedialer Daten. Das Protokoll steuert den Aufbau einer Verbindung zwischen Kommunikationspartnern. Die Verbindung wird hierbei über HTTP-ähnliche Nachrichten gesteuert.
Dabei beteiligte Entitäten sind:

- Proxy Server
- Redirect Server

- User Agent Server
- Registrar

Es kann aus den folgenden Gründen zu Problemen beim Verbindungsaufbau kommen:

- **Überwinden von NAT Gateways**
  - Nutzer registrieren sich bei Proxy Servern.
  - Invitations und Requests werden über Proxy Server weitergeleitet.
  - Häufig kennen User Agents ihre IP-Adresse nicht, dann häufig STUN-Server.
    *Ein STUN-Server (**S**imple **T**raversal of **U**DP Through **N**etwork Address Translators [NATs]) ermöglicht es NAT-Clients (z. B. Computern hinter einer Firewall), die Kommunikation mit einem VoIP-Provider außerhalb des lokalen Netzwerks aufzubauen.*
- **Discovery des Kommunikationspartners**
  - IP Nummern sind untauglich, da veränderlich.
  - Proxy Server enthalten Registrierungsdaten der Nutzer.

13.8.2 Erinnerung: ASN.1

Die Abstract Syntax Notation One ist eine Beschreibungssprache zur Definition von Datenstrukturen sowie Festlegungen zur Umsetzung von Datenstrukturen und Elementen in ein netzeinheitliches Format.

```
FooProtocol DEFINITIONS ::= BEGIN
    FooQuestion ::= SEQUENCE {
        trackingNumber INTEGER,
        question       IA5String
    }
    FooAnswer ::= SEQUENCE {
        questionNumber INTEGER,
        answer         BOOLEAN
    }
END
```

13.8.3 H.323

- Bei H.323 handelt es sich um einen übergeordneter Signalisierungsstandard für Multimedia-Dienste, inkl. VoIP. Die Beschreibung erfolgt in ASN.1.
- Der Standard definiert eine Reihe von Protokollen für die Steuerung von multimedialer Kommunikation.

  H.225.0 - Registration, Admission, and Status (RAS)

  H.245 - Steuerung, beschreibt Nachrichten zum Öffnen und Schließen logischer Kanäle für Audio, Video und Daten.
- Es werden außerdem eigene Codecs definiert (H.261, H.263, H264 - letzterer für HD DVD und Blu-Ray)
- H.323 ist Transport-Protokoll-unabhängig

13.8.4 H.225-0 (RAS)

Definiert eine Reihe von Nachrichten für die Registrierung, Zugangssteuerung und Statusübertragung.
(Discovery, Registration, Unregistration, Admission – ARQ (Request) / ACF (Confirm) / ARJ (Reject), Bandwidth Change, Endpoint Location, Disengage, Status)

14 World Wide Web und HTTP

Durchbruch des Internets erfolgte erst nach dem Erscheinen des ersten graphischen WWW-Browsers („Mosaic“). Dieser ermöglichte erstmals den komfortablen und effizienten Zugriff auf die Ressourcen des Internets. Das zuvor „beliebte“ Gopher wurde relativ schnell verdrängt. Bis zum Ende der 90er-Jahre vervielfachte sich die Zahl der Websites dramatisch (1993 ca. 50; 1994 ca. 800, 2000 über 22 Millionen)

14.1 Anforderungen an Internet-Protokolle

- Weitestgehende Unabhängigkeit von der Netzwerk-Hardware.
  - Übertragungsmedium (Kupferkabel, Lichtwellenleiter usw.)
  - Typ des lokalen Netzwerks (Ethernet, Token Ring usw.)
- Erreichbarkeit aller Rechner im gesamten Internet.
- Fehlererkennung und Fehlerkorrektur.
- Logische Adressierung (Adresse eines Rechners sollte nicht von der Netzwerk-Hardware abhängen, da bei Defekt einer Ethernet-Karte beispielsweise eine ausgetauschte Karte zu einem Adresswechsel führen würde).
- Verbindungsherstellung.
- Datenübertragung (Versendung von Datenpaketen /Datagrammen).

14.2 Hypertext Transfer Protocol

> **Anmerkung:** Die Folien sind hier nicht so geil. Ich schreibe das Interessanteste zu HTTP aus dem Tanenbaum[72] und den Folien hier zusammen, sodass der Leser danach hoffentlich einen kurzen, zusammenhängenden Schnack über das Thema hinbekommt.

HTTP 1.1 wurde in RFC 2616[24] definiert und später um TLS(RFC 2817 [41]) erweitert; 2014 wurde die gesamte Protokollsuite in den RFCs 7230 - 7235 neu formuliert[25, 26, 27, 28, 29, 30]. Es handelt sich um ein**generisches**[21], **zustandsloses Request-Response-Protokoll**, Daten werden im Klartext übermittelt, wenn keine explizite Verschlüsselung gefordert ist. Ressourcen werden über URIs[6] adressiert.[22]

14.2.1 Verbindungen

HTTP nutzt den TCP-Port 80 (bzw 443 für HTTPs). Während bei HTTP 1.0 für jede Anfrage und Antwort eine eigene Verbindung aufgebaut wurde, bleibt diese in HTTP 1.1 eine Weile erhalten (*persisten connections*). Auf diese Weise wird der Overhead zum Auf- und Abbau der Verbindung, sowie der Leistungsverlust durch **TCP-Slow-Start** verringert, wenn weitere Inhalte (z.B. Bilder, CSS, Skripte) in Folge der originalen Anfrage nachgeladen werden. Hier ist auch **Pipelining** erlaubt, das heißt, dass mehrere Anfragen gesendet werden können, bevor entsprechende Antworten empfangen wurden.

14.2.2 Methoden

HTTP kennt mehrere Methoden[23]. Eine Anfrage beginnt immer mit dem Namen der angeforderten Methode. Einige Methoden sind:

GET : Lesen einer Website

HEAD : Lesen des Headers einer Website, zum Beispiel genutzt von Suchmaschinen

POST : Anhängen von Daten an eine Website, beispielsweise genutzt, um Formulardaten an den Server zu übermitteln.

Weitere sind beispielsweise **PUT**, **DELETE**, **TRACE**, **CONNECT** und **OPTIONS**. Eine einfache Anfrage wäre etwa `GET index.html HTTP/1.1`. Diese fordert offensichtlich die Datei `index.html` unter Verwendung von HTTP 1.1 an.

Die Antwort auf eine Anfrage besteht aus einer Statuszeile und gegebenenfalls weiteren Informationen. Zu Letzterem zählt insbesondere der angeforderte Inhalt, sofern verfügbar. In der Statuszeile ist ein **Statuscode**, bestehend aus drei Ziffern zu finden. Diese Statuscodes werden, entsprechend der ersten Ziffer, in fünf Gruppen geordnet:

1xx : Informational - Request received, continuing process

2xx : Success - The action was successfully received, understood, and accepted

[21] Es ermöglicht die Übertragung jeglicher Daten und beschränkt sich nicht nur auf Hypertext-Dokumente.
[22] Angeführt durch die Protokollangabe http:// oder https://, dabei handelt es sich dann allerdings um eine URL.
[23] Laut Tanenbaum[72] scheint hier tatsächlich die Methode aus dem objektorientierten Sprachgebrauch gemeint zu sein.

3xx : Redirection - Further action must be taken in order to complete the request

4xx : Client Error - The request contains bad syntax or cannot be fulfilled

5xx : Server Error - The server failed to fulfill an apparently valid request

Klugscheißerwissen: Neuer Statuscode 451 für zensierte Inhalte[8, 69]

14.2.3 Header

An die im letzten Abschnitt gezeigten Anfrage können weitere Informationen in **Request Headern**[24] angefügt werden. Analog dazu kann die Antwort mit **Response Headern** versehen werden. Einige dieser Felder können nur bei Anfragen (z.B. User-Agent), einige nur bei Antworten(z.B. Content-Encoding), einige bei beidem (z.B. Date) übermittelt werden.

Der Client kann beispielsweise in seiner Anfrage angeben, welche Zeichensätze (*Accept-Charset*, z.B. ISO-8859-1) und Codierungen (*Accept-Encoding*, z.B. gzip, compress oder deflate[35, 17]) er akzeptiert . Entsprechend enthält die dann Antwort (im Allgemeinen) die Felder *Content-Encoding* und *Content-Type*[25]

14.3 HTTP/2

HTTP/2 wird in RFC 7540[5] definiert. Es erweitert seinen Vorgänger um einige neue Funktionen:

- Möglichkeit des Zusammenfassens mehrerer Anfragen,
- Weitergehende Datenkompressionsmöglichkeiten,
- Binär kodierte Übertragung von Inhalten und
- Server-initiierte Datenübertragungen (push-Verfahren).

HTTP/2 überträgt sämtliche Nachrichten in Streams. Ein Stream ist eine unabhängige, bidirektionale Sequenz von Frames, die zwischen Client und Server in einer HTTP/2 Verbindung übertragen werden. Es handelt sich weiterhin um ein Klartext-Protokoll, die Syntax ist hierbei jedoch stark verändert worden. Es basiert in weiten Teilen von Googles **SPDY**, welches wiederum auf HTTP aufsetzt und dieses verbessern sollte. Ziele bei der Entwicklung von SPDY waren unter anderem:

- SPDY-Übertragung wird immer mittels TLS verschlüsselt (HTTP/2 optional, aber Browser-Hersteller verlangen Verschlüsselung)
- Multiplexen der Übertragungen
- Über eine einzelne TCP-Verbindung können beliebig viele Dokumente parallel übertragen werden
- Möglichkeit zur Priorisierung einzelner Anfragen

14.4 File Transfer Protocol

FTP[62] ist ein Protokoll zum Kopieren von Dateien von einem System auf ein anderes. Es wurde so konzipiert, dass verschiedene Hosts (betriebssystemunabhängig), Dateitypen (ASCII, binary, etc.) und Dateistrukturen (byte stream oder record oriented) unterstützt werden. Zum Übertragen der Daten wird in der Regel der TCP-Port 20 verwendet, für Steuernachrichten Port 21. Der prinzipielle Ablauf ist wie folgt:

1. FTP - Clientprozess wird gestartet.
2. Client baut eine TCP Verbindung zum Server auf.
3. Durch TCP werden zwischen den beiden Prozessen zwei Verbindungen aufgebaut (Daten und Kommunikation).
4. Interaktiver Anwender muss Zugangsberechtigung auf Server eingeben (Login/Passwort bzw. anonym).
5. Datentransfer kann erfolgen (Textdateien und Binärdateien).

[24] Möglicherweise eine Sache der Übersetzung, aber im Tanenbaum wird ein Feld als einzelner Header bezeichnet. Ich switche gleich zu Feldern statt Headern.

[25] Eigentlich für den MIME-Typ, enthält aber auch den charset.

14.4.1 Steuersignale und Return Codes

Steuersignale werden im Klartext vom Client an den Server gesendet.

USER (User Name) - Eingabe einer UserID für den Server.

PASS (Password) - Passwort an den Server senden.

CWD (Change Working Directory) - Wechseln des Arbeitsverzeichnisses.

PORT - Angabe der Client Callback-Adresse (IP und Port).

LIST - Auflisten von Dateien / Verzeichnissen.

RETR (Retrieve) - Hole eine Datei vom Server.

STOR (Store) - Speichere eine Datei auf dem Server.

SYST (System) - Erhalte vom Server den System - Typ.

TYPE - Angabe der zu nutzenden Datendarstellung (A / I).

ABOR (Abort) - Beendet den Datentransfer bzw. den letzten Befehl.

QUIT - Ausloggen vom Server / Verbindung abbauen.

MKD (Make Directory)

RMD (Remove Directory)

DELE (Delete)

PWD (Print Working Directory)

HELP

Analog zu den Statuscodes bei HTTP sendet der Server Return Codes.

1yz – Positiver Beginn eines Befehls.

2yz – Positive Beendigung eines Befehls.

3yz – Positiver Zwischenbericht.

4yz – Transientes Problem. Der Befehl wurde nicht ausgeführt.

5yz – Definitives Problem.

x0z – Syntaxfehler.

x1z – Allgemeine Information.

x2z – Verbindungszustand.

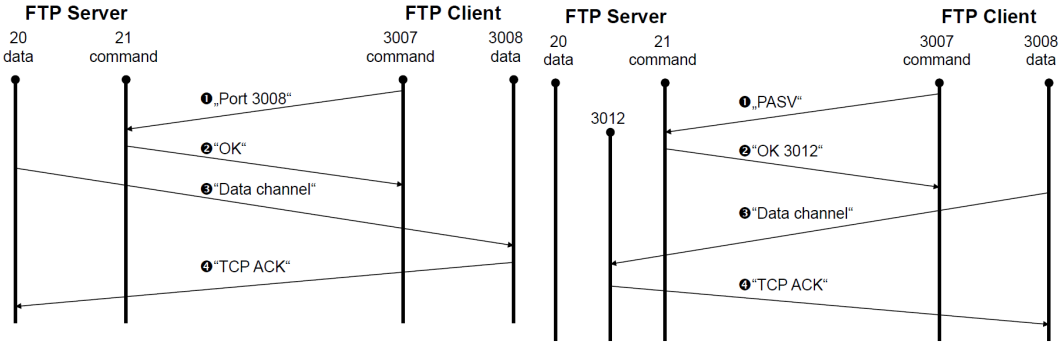
x3z – Authentisierung und Accounting

x4z – Nicht spezifiziert.

x5z – Status des Dateisystems.

14.4.2 Aktiver und Passiver Modus

Beim **aktiven** Modus baut der Client eine den Steuerkanal auf und öffnet einen TCP-Port zur Datenübertagung. Dieser teilt er dem Server mit, welcher sich darauf verbindet und die Datenübertragung kann beginnen. Beim **passiven** Modus teilt der Server dem Client einen Port mit. Auf diesen Verbindet er sich dann zur Datenübertragung.



15 Peer-to-Peer

Der Tausch von Dateien und Streams jeder Art wurde zur Anwendungen des Internets, die die meiste Bandbreite verbraucht (geschätzt bis 75 Prozent). Bekannte P2P-Beispiele sind:
eDonkey, eMule, Gnutella, LimeWire, Napster (bis 2001), Bit-Torrent, FastTrack

P2P Netze stellen eine Alternative zu zentralisieren Strukturen dar und bringen eine Reihe von Herausforderungen mit sich:

- Finden der gesuchten Information
- Geschickte Verteilung der Downloads
- Kein Verlass auf Peers und Verbindungen
- Wie kann die Funktionstüchtigkeit eines P2P Netzwerkes sichergestellt werden?

15.1 Rechtliche Aspekte

- Überwiegende Resistenz gegen staatliche Eingriffe.
- Störungssicherheit durch chaotische Organisation.
- Zensur von Informationen ist nahezu unmöglich.
- Legislative „Störungen“– Beschränkung der Macht von Gesetzgebern – Echte P2P Systeme sind von keiner Regierung zu kontrollieren.
- Urheberrechte wurden in den meisten Ländern verschärft.
- Einführung neuer Straftatbestände.

15.2 Organisation der Daten

Einige P2P Netzwerke benutzen **zentrale Server** (Tracker), an denen sich alle beteiligten Peers anmelden. Andere implementieren **verteilte Suchstrategien**, um ohne zentralen Server auszukommen.

15.3 Suchstrategien

„Alle“fragen (Broadcast)

- Funktioniert nur in kleinen Domains
- Zahl der Anfragen steigt mit der Zahl der Peers
- Je mehr Peers vorhanden sind, desto ungünstiger wird das Verhältnis zwischen Anfragebearbeitung und „eigentlicher“Diensttätigkeit

Super Peers

- Einige Peers haben eine Sonderrolle
- Normale Peers melden sich bei Super Peers an
- Super Peers kennen weitere Super Peers
- Anfragen werden an Super Peers gestellt und von diesen verteilt

Flooding oder Web Crawling

- Jeder Peer fragt ihm bekannte Peers.
- Diese leiten die Frage im Nichterfolgsfall weiter.
- „Verebben"der „Frageflut"nach einer vorgegebenen Anzahl von Peers (TTL).
- Vermeidung von Kreisen durch Informationen, wer schon gefragt wurde.
- **Flooding bekannt aus Verteilte Algorithmen!**
- Im Erfolgsfall wird mit einem Antwortpaket geantwortet, dieses enthält die Netzwerkadresse des Hosts, der die Antwort bereitstellt. Die eigentliche Anfrage erfolgt anschließend über spezielle Anfrage-Pakete.

15.4 Beschreibung von Informationen

Die Daten, die in einem P2P-Netzwerk angeboten werden, können durch zwei Möglichkeiten beschrieben werden:

- Eindeutige Kennzeichnung von Informationen, z.B. durch Hash-Werte, IDs etc.
- Umfangreiche Beschreibungen der eigentlichen Informationen mit Meta-Informationen.

15.5 Eigenschaften von P2P

Verfügbarkeit wird gewährleistet durch:

- Exzessive Replikation.
- Redundanz der Dienste.
- Redundanz der Kommunikationspfade.

Performance in Hinblick auf

- Den Aufwand eine gestellte Frage zu beantworten
- Den Overhead für die Kommunikation und zur Steuerung

→ Dezentral organisierte P2P-Netzwerke eignen sich nicht für Aufgaben, die in einer vorgegebenen Zeit erfüllt sein sollen

Anonymität

- Vollständige Anonymität lässt sich mit P2P Technologie relativ einfach erreichen.
- Dabei bleiben sowohl Dienste-Anbieter als auch Dienste-Client anonym.
- Tunneln aller Nachrichten über verschiedene Peers.

15.6 Beispiel BitTorrent Protocol

P2P Protokolle stellen besondere Anforderungen an die zugrundeliegenden Algorithmen, damit die Verteilung der Daten vorteilhaft für alle geschieht.

Das BitTorrent Protocol wurde von BitTorrent, Inc. spezifiziert. Eine Distribution besteht aus:

- Web Server – Download der Torrent-Files
- Metainfo – Torrent-File
- Tracker – Verwaltet die aktuell angemeldeten Clients
- Client – Verteilt Teile der Nutzdatei

15.6.1 Aufbau

Die einzelnen Torrent-Files sind wie folgt aufgebaut:

- Strings, denen die Länge vorgestellt wird
  announce44:http://tpb.tracker.thepiratebay.org/announce
- Enthält alle Tracker
- Enthält alle „Pieces“
  - piece_length legt Länge der Teile fest, typisch sind 256KB
  - Jedes „Piece“wird durch 20 Zeichen SHA-1repräsentiert.

15.6.2 Nachrichtenaustausch

Nachrichten zwischen Tracker und Client werden über HTTP GET ausgetauscht:

info_hash - SHA-1 Hash aus dem Metainfo-File wird an Tracker geschickt

peer_id – 20 Zeichen lange zufällige ID, vom Client erzeugt

ip – IP-Adresse oder Name des Clients

port – Port, auf dem der Client arbeitet

uploaded – Anzahl der hochgeladenen Bytes

downloaded - Anzahl der heruntergeladenen Bytes

left – Anzahl der Bytes, die Peer noch runterladen muss

event – Optional, wird zum Beispiel zu Beginn und zum Ende gesendet.

15.6.3 Datenaustausch

Der eigentliche Datenaustausch erfolgt dann P2P:

Request – Piece anfordern

Cancel – Abbruch der Anforderung

Choke – Drosseln des entsprechenden Clients

Unchocke - Freigabe des Clients (anderer Peer entscheidet, wer download starten darf)

Have – Gibt an, dass das entsprechende Piece vorhanden ist.

Interested – Client zeigt Interesse an.

Piece – enthält das gewünschte „Piece“

16 E-Mail

Protokolle zum E-Mail-Versand sind POP3, IMAP und SMTP. Diese bauen auf Transport Transportprotokollen auf, wie

- TCP auf Port 25 mit 7 bit ASCII
- NCP (Netwok Control Program) auf Port 25. „Gibt es überhaupt noch ein laufendes NCP? “
- X.25 könnte direkt benutzt werden, vorgeschlagen wird aber TCP over X.25

16.1 POP3

Das POP3 (Post Office Protocol - Version 3)[52] dient zum Abrufen eingehender Mails und wird typischerweise eingesetzt, wenn direkter Mail-Empfang nicht möglich ist. Dies kann dann eintreten, wenn bspw. keine permanente Verbindung besteht, zu wenig Ressourcen beim Client bzw. Host zur Verfügung stehen oder auch wenn „postlagernde“Mails aberufen werden.
Obwohl POP3 eigentlich unabhängig vom Transport Protkoll ist, ist es nur für TCP auf Port 110 spezifiziert.

16.1.1 Arbeitsweise

Kommandos bestehen aus einem Schlüsselwort und eventuell aus Argumenten. **Antworten** bestehen aus Status, Schlüsselwort und eventuell nachfolgenden Informationen

16.2 Zustände und Kommandos des Servers

AUTHORIZATION - Phase der Authentisierung

- **USER** - Übergabe des Benutzernames
- **PASS** - Übergabe des Passworts (nur nach USER)
- **APOP** (Optional) - Übergabe es MD5 Hashes anstatt eines Passworts
- **QUIT** - Schließen der Verbindung

TRANSACTION - Phase der Datenübertragung

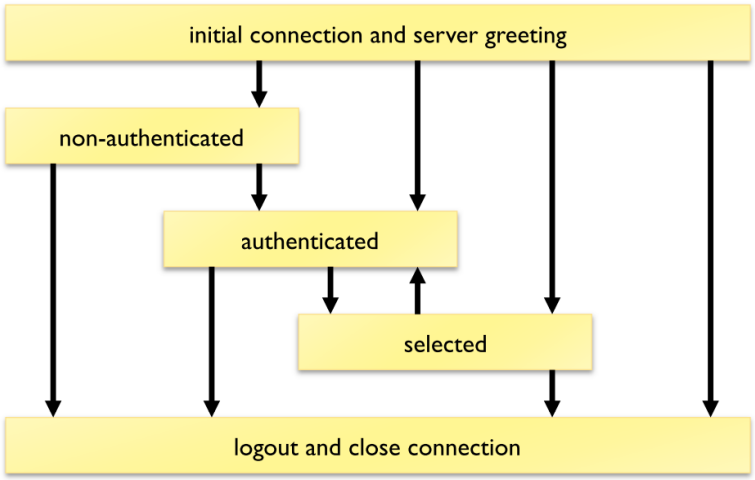
- **LIST** [**msg**] - Listet die vorhandenen Nachrichten auf. Zum einfacheren Parsen der Resultate ist das Ergebnis-Format streng vorgeschrieben
- **RETR msg** - Abholen der Nachricht mit der Nummer msg, wobei das Nachrichtenende mit . ¡CR¿¡LF¿ gekennzeichet ist.
- **TOP msg n** - Mit diesem optionalen Befehl werden die ersten n Zeilen der Nachricht mit der Nummer msg abgeholt
- **DELE msg** - Löschen der Nachricht mit der Nummer msg
- **RSET** - Reset des Servers, alle zum Löschen markierten Nachrichten werden demarkiert
- **STAT** - Liefert die Anzahl der Mails und die Gesamtgröße zurück
- **UIDL** [**msg**] - Gibt für jede bzw. die Nachricht mit der Nummer msg einen Unique Identifier aus
- **NOOP** - Tue nichts - sinnvoll, um die Session am Leben zu halten
- **QUIT** - Schließen der Anwendung und Übergang in UPDATE State

UPDATE - Phase beim Beenden und alle zum Löschen markierten Mails werden gelöscht

16.3 IMAP4

Das IMAP4[13] (Internet Message Access Protocol - Version 4) ermöglicht, dass Nachrichten auf dem Server verbleiben und dort effektiv verwaltet werden. Das Protokoll ist nur für das Empfangen von Nachrichten zuständig, aber Nachrichten können auch in einzelnen Mailboxen erzeugt werden.

16.3.1 Zustände des Servers



16.3.2 AUTHENTICATE

Die Anmeldung mittels IMAP4 kann bspw. mittels Challenge Response, wie Kerberos, erfolgen. Alternativ kann der CLient verschiedene Authentisierungsverfahren vorschlagen. Als Ausweg, wenn Authentisierung über Challenge-Response nicht funktioniert, kann ein Login genutzt werden.

16.3.3 SELECT

Im Zustand **selected** wurde eine Mailbox ausgewählt. Dieser Zustand wird nach einem erfolgreichen Auswählen betreten. Erreicht wird dieser Zustand mit dem Select-Befehl, der eine Mailbox auswählt. Nachfolgende Kommandos beziehen sich dann auf die selektierte Mailbox.

16.3.4 Befehle

EXAMINE Als Statusanfrage zum Überprüfen einer Mailbox auf ungelesene Mails.
Laut RFC: *The EXAMINE command is identical to SELECT and returns the same output; however, the selected mailbox is identified as read-only.*

CREATE - Anlegen einer Mailbox

RENAME - Unbenennen einer Mailbox

DELETE - Löschen einer Mailbox

SUBSCRIBE - Abonnieren einer Mailbox

UNSUBSCRIBE - Kündigen eines Abonnements

LIST path selector - Anzeige der verfügbaren Mailboxen unterhalb des path gemäß des selectors

SEARCH - Suche innerhalb von Mailboxen auf dem Server

FETCH - Liefert Nachrichten vom Server durch eine Auswahl nach Bedingungen bzw. Flags. Außerdem können Nachrichtenteile angegeben werden, die man haben möchte.

PARTIAL - Wie FETCH, aber mit Einschränkung der Länge der zu liefernden Teile

STORE - Ablegen einer Nachricht in einer Mailbox, z.B. nach Änderungen

COPY - Kopieren einer Nachricht von einer Mailbox in eine andere

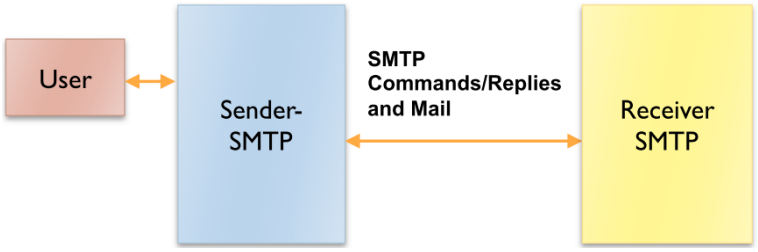
UID - Ausgabe eines eindeutigen Identifiers für jede Nachricht

16.3.5 Mögliche Auswahlkriterien

- Ungesehene Mails
- Nach Datum
- Nach Absender
- Nach TAG („Wichtig“„Unwichtig“etc.)

16.4 SMTP

Das SMTP (Simple Mail Transfer Protocol)[57] ist von der zugrundeliegenden Übertragungsart unabhängig. Es benötigt lediglich einen zuverlässigen, geordneten Datenstream. Zwischen den Kommunikationspartnern (Sender-SMTP und Receiver-SMTP) wird ein Zwei-Wege-Kommunikation-Kanal aufgebaut. Dabei kann der Receiver-SMTP entweder das Ziel oder ein Zwischenglied sein. Vom Sender werden Commands an den Receiver gesendet, dieser antwortet mit Replies.



16.4.1 Arbeitsweise

- Once the transmission channel is established, the SMTP-sender sends a **MAIL** command indicating the sender of the mail.
- If the SMTP-receiver can accept mail it responds with an **OK** reply.
- The SMTP-sender then sends a **RCPT** command identifying a recipient of the mail.
- If the SMTP-receiver can accept mail for that recipient it responds with an **OK reply**; if not, it responds with a **reply rejecting** that recipient (but not the whole mail transaction).
- The SMTP-sender and SMTP-receiver may negotiate several recipients. When the recipients have been negotiated the SMTP-sender sends the mail data, terminating with a special sequence.
- SMTP Commands:

```
S: MAIL <SP> FROM:<reverse-path> <CRLF>
S: RCPT <SP> TO:<forward-path> <CRLF>
S: DATA <CRLF>
```

16.4.2 Beispiel

```
S: RCPT TO:<Green@Beta.ARPA>
R: 550 No such user here

S: RCPT TO:<Brown@Beta.ARPA>
R: 250 OK

S: DATA
R: 354 Start mail input; end with <CRLF>.<CRLF>

S: Blah blah blah...
S: <CRLF>.<CRLF>
R: 250 OK
```

16.4.3 Weiterleitung

Wenn ein Sender eine Daten an einen Receiver sendet, um sie daraufhin an einen Empfänger weiterzusenden, gibt es zwei Reaktionsmöglichkeiten zum Forwareding:

1. Receiver kennt den Pfad zum Empfänger und übernimmt die Verantwortung für die Zustellung

   ```
   R: 251 User not local; will forward to <forward-path>
   ```

2. Receiver kennt den Pfad, aber will Mail nicht selbst zustellen

   ```
   R: 551 User not local; please try <forward-path>
   ```

Was passiert, wenn sich nach Annahme der Mail durch einen Relayer herausstellt, dass Empfänger doch nicht erreichbar ist? Beispielsweise so wie in Möglichkeit 1.

- Receiver muss eine Fehlermeldung generieren.
- Wohin wird diese Fehlermeldung gesendet?
- Was passiert bei mehrstufigem Forwarding?

Um dies zu Lösen wird beim Forwarding jeweils der Pfad mitaufgenommen. Die Fehlermeldung gelangt auf dem umgekehrten Pfad zurück zum Empfänger. Typischerweise existiert für eine Gruppe von Usern ein Relay Host, der Mails für beliebige Empfänger entgegennimmt und weiterleitet.
Es wird zwischen Zwei Konzepten unterschieden:

- Adresse (Wer ist der Empfänger?)
- Route (Wie gelangt man zum Empfänger?)

Des Weiteren gibt es eine Reihe von Befehlen:

- **VRFY** - Fordert den Server auf eine Adresse zu überprüfen
- **EXPN** - Dieser Befehl fragt nach einer Namensauflösung. Es werden je nach Berechtigung die Mitglieder einer Mailingliste ausgegeben.
- Nachricht am Terminal des Users ausgeben

  `SEND <SP> FROM:<reverse-path> <CRLF>`

- Nachricht am Terminal ausgeben oder in die Mailbox legen

  `SOML <SP> FROM:<reverse-path> <CRLF>`

- Nachricht am Terminal ausgeben und in die Mailbox legen

  `SAML <SP> FROM:<reverse-path> <CRLF>`

- HELO, QUIT, TURN, HELP, RSET, NOOP

17 Autokonfiguration

Dieser Abschnitt soll sich mit Technologien zur automatischen Konfiguration von Geräten und Diensten beschäftigen. Die automatische Konfiguration von Geräten kann zentral von einem Server oder dezentral durch Abstimmung zwischen den Geräten erfolgen. Dabei bieten sich zwei Strategien an:

- Geräte bieten permanent ihre Dienste an (Advertisements)
- Dienstenutzer fragen nach dem gewünschten Dienst

17.1 Motivation

- Wachsende Anzahl vernetzter Geräte
- Mobilität (PDA, Notebook, Handy)
- Vordringen rechnergestützter Systeme in neue Einsatzbereiche (Home/Office Automation)
- Neue Medien und Inhalte
- Vereinfachung der Administration

17.2 Discovery

1. Anfragebasiert
   - Gerät oder Client fragt in den Raum nach einem Anbieter von Konfiguration oder Diensten.
   - Anbieter lauscht und antwortet auf Anforderung.
2. Mittels Advertisements
   - Anbieter von Diensten oder Konfigurationsdaten macht von sich aus auf sich aufmerksam.
   - Sendet Bekanntmachungen an alle.
   - Clients lauschen einige Zeit.
3. Broadcasts / Multicasts für Anfragen in den Raum.

Dies lässt sich **optimieren**: Broadcasts / Multicasts zum Finden des Konfigurationsservers sind nur einmal notwendig. Danach ist wiederholtes Anfragen beim gefundenen Server möglich.
Ebenso ist eine Konfiguration ohne zentralen Server möglich. Dabei kann es jedoch zu Kollisionen kommen, die erkannt werden müssen.

Eine Alternative stellen **Registries** da: Dienste können bei einem Verzeichnisdienst registriert werden, den der Client bei der Konfiguration sucht. Der Client muss darauf hin nur einmal den gefundenen Verzeichnisdienst nach Diensten fragen.

17.3 Leases

- Fehler können Abbruch der Kommunikation bewirken.
- Vergebene Ressourcen würden dann nicht wieder freigegeben, weshalb sie nur auf Zeit vergeben werden.
- Vor Ablauf der Zeit (Lease Time) muss Ressourcen-Reservierung verlängert werden.
- Nach Ablauf der Lease Time wird Ressource wieder freigegeben. Client muss wissen, dass er die Ressource dann nicht mehr nutzen darf.

17.4 Geräte-Integration

- Herstellung der Kommunikation
- Einstellen von Geräte- und Netzwerkparameter
- Sicherheit
- Protokolle: RARP, BOOTP, DHCP, IP-Autoconfiguration, IPv6

17.4.1 RARP

17.4.2 BOOTP

Ursprünglich war BootP (Bootstrap Protocol) [14] für Diskless Workstations gedacht. Es liefert IP-Adresse und weitere Informationen, wie bspw. die Adresse eines TFTP-Servers. Die Kommunikation erfolgt via UDP.

Ein BootP-Protokoll-Paket enthält die IP-Adress-Informationen von BootP-Client, -Server und Router, sowie die vom BootP-Server vorgeschlagene IP-Adresse für den Client. Darüber hinaus hat das BootP-Frame Datenfelder für den Client-Hardwareadresse, den Server-Host-Namen, den Boot-File-Namen und ein herstellerspezifisches Feld.

17.4.3 DHCP

Das DHCP (Dynamic Host Configuration Protocol) [18, 2] wird zur Übertragung von Konfigurationsparametern zu Knoten in TCP/IP-Netzen verwendet. DHCP ist abwärtskompatibel zu BOOTP.

DHCP wird zur automatische Vergabe von IP-Adressen verwendet - und damit mehr als nur eine statische Zuordnung. Es erfolgt eine temporäre (dynamische) Zuweisung der Adressen, was DHCP zu einem aufwändigerem Protokoll macht.
Es gibt verschiedene Nachrichtentypen:

- Client: DHCPDISCOVER
- Server: DHCPOFFER
- Client: DHCPREQUEST
- Client: DCHPDECLINE
- Server: DHCPACK
- Server: DHCPNACK
- Client: DHCPRELEASE
- Client: DHCPINFORM

Außerdem lassen sich bestimmte Parameter anpassen:

Reqested Address - Kann der Client beim DHCPDISCOVER angeben, um sich eine bestimmte Adresse zuweisen zu lassen.

Lease Time - Zeitraum für die Gültigkeit der Adresse. Angabe in Sekunden (32 Bit → max. 99 Jahre)

Option Overload - Die Felder Server Hostname und Boot Filename des können für Optionen genutzt werden.

Server Identifier - Erlaubt die Unterscheidung zwischen verschiedenen Serven.

Parameter Request List - Aufzählung der Tags der Optionen, an denen der Client interessiert ist.

Max Message Size - Maximale Größe der vom Client akzeptierte DHCP-Nachricht. Kleinster zugelassener Wert ist 576.

Client Identifier - Eindeutige ClientID. Wird vom Server zur Verwaltung der Leases verwendet (bspw. MAC-Adresse).

Die wichtigsten DHCP-Vorgänge sind:

Adressanforderung - Initiale Anforderung von Parametern (z.B. während des Bootvorganges) aus einem unkonfigurierten Zustand heraus

Lease-Erneuerung - Verlängerung der Lease-Dauer der Adresse, Netzwerkschittstelle ist konfiguriert

Abfrage von Informationen - Anforderung von zusätzlichen Parametern (außer Adresse), Netzwerkschittstelle ist konfiguriert, Nachricht hat keinen Einfluss auf den Lease-Status im DHCP-Server

Rückgabe der Adresse - Adresse vom wird Client nicht mehr beötigt, daher Information an den Server, dass Adresse zur Wiederverwendung freigegeben ist.

17.5 Dienste-Integration

- Verfügbarkeit sicherstellen
- Registrierung
- Sicherheit
- Protokolle: UPnP, HAVi, SLP, JetSend, Jini

17.5.1 UPnP

UPnP (Universal Plug and Play) vereint Geräte- und **Dienstekonfiguration** auf der Basis von bewährteer Technologie als offene Service-Architektur. Es definiert netzwerkseitige Interaktion, die tatsächliche Implementierung ist herstellerspezifisch.
UPnP verwendet bekannte Technologien wie IP, TCP, UDP, ARP, HTTP, DHCP und XML.
Es stellt eine Sammlung von Protokollen dar:

- SSDP (Simple Service Discovery Protocol)
  - Protokoll zur Lokalisierung von UPnP-Diensten
  - Unterstützt Client Requests sowie Service Announcements
- GENA (General Event Notification Architecture)
  - Beschreibt eine HTTP-Benachrichtigungs Architektur
  - Eine HTTP-Ressource kann ein beliebiges Objekt sein, das Benachrichtigungen senden oder empfangen kann.
  - Die Event-Registrierung muss vor ihrem Ablauf erneuert werden, wenn sie andauern soll
  - Soll die Event-Registrierung vor Ablauf des Timeouts aufgehoben werden, muss sie explizit beendet werden
  - Notification wird vom Notifier an den Arbiter geschickt, der Arbiter prüft, ob es Subscriber für den Notification Type mit passendem Scope gibt, wenn ja, passt der Arbiter den SID-Header für den Subscriber an und leitet die Notification weiter.
- SOAP (Simple Object Access Protocol)

17.5.2 automatische Konfiguration in UPnP

Addressing - Konfiguration der Netzwerk-Interfaces über DHCP oder IP-Autokonfiguration

Discovery - Lokalisierung von Services durch Control Points oder Service Advertisement

Description - Akquisition der Service-Beschreibung durch den Control Point

Control - Benutzung des Dienstes (Interaktion Control Point-Service)

Eventing - Asynchrone Benachrichtigung von Control Points über Zustansänderungen im Service

Presentation - Darstellung Services-UIs im Browser (Statuspräsentation oder Service-Steuerung)

17.5.3 HTTPU und HTTPMU

HTTP wird von TCP als Transportschicht spezifiziert. HTTPU und HTTPMU sind Variationen von HTTP via Uni-/Multicast mittels UDP. Diese werden von SSDP und GENA verwendet, bilden aber keinen Ersatz für HTTP über TCP.

18 Dateien und Drucken

18.1 Herausforderungen

Files
Das Ziel ist ein Zugriff auf Verzeichnisstrukturen über ein Netzwerk, als wären sie lokal. Wünschenswert hierbei sind: Transaktionskontrolle auf verteilte Ressourcen, Fehlermeldungen, Rechtekontrolle, Sicherheit.
Daten kennzeichnen sich durch den Inhalt der Files und ihre Metadaten (Name, Größe, Datumsangaben, Eigentümer, Rechte, Flags, etc.)

Drucken
Hier ist das Ziel das Ansprechen unterschiedlicher Drucker mit unterschiedlichen Druckerprotokollen und im Fehlerfall das Erhalten von Fehlermeldungen.

18.2 Konzepte für Filesysteme

Transparenz
Es sollen sowohl die Anzahl der Server und dort vorhandener Geräte als auch die Verteilung der Dateien auf unterschiedlichen Servern unsichtbar bleiben. Das Client-Programme benötigen keine Kenntnis darüber, ob ein Filesystem lokal oder remote ist. Die Geschwindigkeit des Datenzugriffs ermöglicht die Illusion eines lokalen Filesystems.

Flash File Systeme
Eine gleichmäßige Abnutzung durch Verteilung erreichen.

Transaktionale Filesysteme

Schnittstelle zum Betriebssystem

Temporäre Filesysteme
Häufig im RAM realisiert: Schneller, gemeinsamer Zugriff über übliche Dateisystem-Schnittstelle
Schreibbare Filesysteme für eigentlich nicht schreibbare Medien (Union Mount)
Bspw. CD-ROM / DVD-ROM mit dynamisch eingeblendetem schreibbarem Bereich.

RAID

Komprimierte Filesysteme

Filesysteme mit unterschiedlich schnellen Medien

18.3 empty

Qualitätsfaktoren
Das Ziel ist ein gering halten der Latenz beim Bedienen von Anfragen:

- Lokale Systeme: Laufwerkszugriffe und Verarbeitungsgeschwindigkeit (CPU)
- Zusätzlich bei remote Zugriff: Netzwerklatenz für Anfrage, Netzwerklatenz für Ergebnis, Bandbreite

Nebenläufigkeit
Nebenläufigkeit ist bereits bei Mehrbenutzermaschinen lokal notwendig, also eigentlich kein spezifisches Problem von Netzwerk-Dateisystemen. Es besitzt nur eine etwas höhere Komplexität durch zwischengeschaltetes Netzwerkprotokoll.
Des Weiteren ist eine unterschiedliche Granularität von Locks notwendig (File-Systeme, Files, Blocks)

Zugriffssteuerung
Transparente Umsetzung der lokalen Mechanismen, wie Zugriffsmatrix und ACL.

18.4 Probleme

- Zeichenkodierung
- Netzwerkprobleme während offener Transaktionen
- Timeouts notwendig, um mit Netzwerkproblemen umzugehen
- Replikation und Fehlertoleranz.
- Historisierung.
- Archivierung.

18.5 Protokolle

18.6 NFS

Das NFS (Network File System) [67] ist in Version 4 und unterstützt UDP, TCP und Dateien, die größer als 4GB sind. Eine Authentisierung über IP-Adresse wird unterstützt.

- **zustandsbehaftetes Protokoll**
- Kryptografisch abgesichert
- Verteiltes Filesystem, bei dem Zugriffe nicht die Übertragung der gesamten Datei erfordern.
- NFSv4 bietet **Authentisierung** von Server und Client über Kerberos 5 – der eigentliche Benutzer wird nicht authentisiert.

Arbeitsweise

- Client meldet sich beim Server an
- Client erhält Client-ID als Lease
- Client kann File Handles anfordern (persistent oder volatile - flüchtig)
  Für volatile Handles ist nicht garantiert, dass sie gültig und persistent für die Dauer der Lebenszeit des Dateisystems auf das Dateisystemobjekt (File, Directory) verweisen. Server kann zu einer beliebigen Zeit entscheiden, dass ein volatile FH ungültig wird.
- Client führt Dateioperationen auf dem Handle au

NFS kann optimiert werden in dem mehreren Requests und Responses zusammengefasst werden. (Beispielsweise LOOKUP, OPEN, READ) Operationen können mit AND oder OR verknüpft werden und die Auswertung erfolgt in vorgegebener Reihenfolge.

Es gibt Procedures (*NULL, COMPOUND*)und eine Vielzahl von Operationen (*ACCESS CLOSE, COMMIT, CREATE, GETATTR, LINK, LOCK, LOOKUP, OPEN, PUTFH, REMOVE, READ, READDIR, SAVEFH, SECINFO, WRITE,...*)

Retransmission: Server darf Requests über zuverlässige Protokolle (TCP) nicht ignorieren.
Filehandles: Server wandelt Filehandles in Repräsentation des lokalen Filesystems um. Nahezu alle RPC Calls benötigen ein Filehandle. Daher erfragt der Client zuerst das Root-Filehandle. Der Server liefert Attribute des FH.

File Attributes: es gibt 3 Gruppen

- Mandatory Attributes müssen von jedem Client und Server unterstützt werden.
  Bspw. filetype, change, leaseTimem...
- Recommended Attributes: Client kann jederzeit danach fragen, muss aber damit rechnen, keine Antwort zu erhalten. Server müssen Anfragen korrekt oder nicht beantworten.
- Named Attributes sind nicht im NFS Protokoll vorgesehen, können aber über Strings angefragt werden.

Beim Server **Pseudo Filesystem** ersetzt der Server fehlende Teile im Namespace, die nicht exportiert werden, durch Pseudo FS.

Für das **Locking** identifiziert sich der Client mit einer Client ID. Wenn Client abstürzt, wird neue Inkarnation des Clients SETCLIENTID aufrufen. Dadurch werden alle alten Leases der alten Inkarnation auf dem Server ungültig.

OPEN ist immer vorgeschrieben, auch wenn keine Operationen oder exklusiver Zugriff gewünscht sind. Vor **CLOSE** sollen alle Locks freigegeben werden.

18.7 SMB / CIFS

Ursprünglich wurde SMB (Server Message Block) für Windows und OS/2 entwickelt. Es ist aber kein Standard-Dokument verfügbar, da einige Features geheimgehalten werden sollen. (Reverse Engineered für Samba Entwicklung)
Umbenennung 1996 von SMB in Common Internet File System (CIFS), neue Features. CIFS baut dabei auf NetBIOS over TCP/IP und SMB auf und bietet neben der Datei- und Druckerfreigabe weitere Dienste wie zum Beispiel den Windows-RPC- und den NT-Domänendienst an.

18.8 WebDAV

- HTTP Extensions for Web Distributed Authoring and Versioning [19]
- Erweiterung von HTTP durch neue Requests.
  (**COPY, MOVE, LOCK / UNLOCK, DELETE, MKCOL, PROPFIND / PROPPATCH**)
- Properties als XML Dokument
- Datei muss vollständig übertragen werden, dann geändert, dann zurück übertragen.

18.9 iSCSI

Das ISSCSI (Internet Small Computer Systems Interface)[66] benutzt TCP als Transport für SCSI Kommandos (quasi Tunneln) typischerweise Port 860 und 3260.Das SCSI-Gerät verhält sich für das Client-System Initiator) wie ein lokales Gerät.

- Der Initiator (Client) kann entweder aus Software oder Hardware erstellt werden.
- Target (Server) kann ebenfalls als Software oder Hardware realisiert werden.

Die iSCSI Adressierung erfolgt durch:

- Hostname oder IP Adresse
- Port
- iSCSI Name

19 Telnet, SSH und rlogin

19.1 Telnet

Telnet wird genutzt, um eine allgemeine, bidirektionale, 8-bit-orientierte Kommunikation zu ermöglichen und so mit entfernten Terminalen und terminal-orientierten Prozessen zu kommunizieren. Es verwendet den TCP-Port 23, wobei es ursprünglich NCP genutzt hat. Auf beiden Enden der Verbindungen werden Network Virtual Terminals (NVT) angelegt. Diese handeln *Options* aus (z.B. Character Set, Binary Transmission, Echo,Status, Timing Mark Option usw.)

Jedoch: Keine Verschlüsselung der Daten. Keine Authentisierung auf Paket-Ebene. Tunnel durch TLS natürlich möglich.

Einzige sinnvolle Anwendung für Telnet heutzutage ist `telnet towel.blinkenlights.nl` und die Vorort-Konfiguration von Geräten (z.B. Switches oder Router).

19.2 Secure Shell

SSH[74] ermöglicht eine kryptografisch abgesicherte Terminal-Verbindung. Es nutzt RSA und DSH[26] Außerdem wird eine Authentisierung über Public Key ermöglicht. SSH-Server lauschen für gewöhnlich auf TCP-Port 22.

19.3 rlogin

rlogin[40] lauscht auf TCP-Port 513. Nach Verbindungsaufbau sendet der Client vier Strings:

- <null >
- client-user-name<null>
- server-user-name <null>
- terminal-type/speed <null>

Server antwortet mit Null-Byte. Flusskontrolle über START- und STOP-Zeichen. Dimensionen des Ausgabefensters werden übertragen. Sicherheit wie telnet, hinzu kommt, dass viele Implementierungen Passwort-Eingabe für einzelne Clients (nach IP-Adresse) abschalten können.

20 Extensible Messaging and Presence Protocol (XMPP)

XMPP ist aus dem Jabber-Protokoll hervorgegangen und wird von diesem als Grundlage genutzt. Google Talk nutzt es ebenfalls.

- Offener Standard,kostenfrei nutzbar, nicht an einen Hersteller gebunden, freie Libraries
- Kommunikation Client-Server-Client, nicht direkt Client-Client.
- Kein zentraler Server notwendig (jeder kann einen Server betreiben), Adressierung ähnlich wie bei E-Mail
- XMPP-Server sind untereinander verbunden.
- XMPP basiert auf XML als Steam (der kein vollständiges XML-Dokument enthält)
- TCP Port 5222
- Übertragung von Binaries über andere Protokolle (http) oder notfalls als base64 (quasi Binary als ASCII)
- Verbindungen zu anderen Protokollen sind möglich (Translation durch Gateways)
- XMPP kann über HTTP getunnelt werden - Entweder durch regelmäßige Requests (Polling) oder durch Requests mit stark verzögerter Antwort (Binding)
- Benutzung von TLS und Simple Authentication and Security Layer (SASL) ist standardisiert
- Fehlermeldungen werden ebenfalls als XML Standard übertragen

21 LDAP

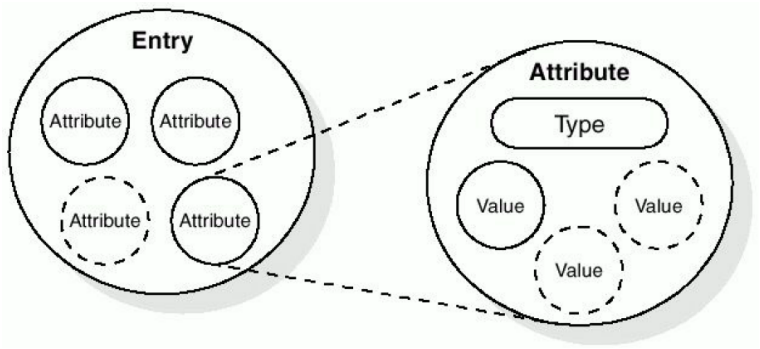
Das Lightweight Directory Access Protocol ist ein Netzwerkprotokoll zur Abfrage und Änderung von Informationen verteilter Verzeichnisdienste. Es beschreibt die Kommunikation zwischen dem LDAP-Client und dem Verzeichnis-(Directory-)Server. Aus einem solchen Verzeichnis können objektbezogene Daten, wie zum Beispiel Personendaten oder Rechnerkonfigurationen, ausgelesen werden. Die Kommunikation erfolgt auf Basis von Abfragen. [*Wikipedia*]

[26] Distributed Shell, warum steht das hier?

21.1 Structure

- Nutzt einen verbindungsorientierten zuverlässigen Transport: TCP auf Port 389
- Connection Oriented Transport Service (COTS) - mapped LDAP Nachrichten PDU direkt auf T-Data (???)
- Es wird X.500 zur Organisation von Verzeichniseinträgen in einem hierarchischen Namespace genutzt - ermöglicht Suchen.
- Used to access and update information in a directory built on the X.500 model
- Specification defines message format between client and the server
- Includes session handling (connect, disconnect etc.)

Information: Structure of information stored in an LDAP directory.
Naming: How information is organized and identified.
Functional / Operations : Describes what operations can be performed on the information stored in an LDAP directory.
Security: Describes how the information can be protected from unauthorized access.



Bei der LDAP Informationsspeicherung beschreibt jeder Eintrag ein Objekt einer Klasse

- Beispiel Eintrag: *InetOrgPerson(cn, sn, ObjectClass)*
- Beispiel Attribute: *cn (cis), sn (cis), telephoneNumber (tel), ou (cis), owner (dn),jpegPhoto (bin)*

(*cn common name, dn distinguished name, sn surname*)

21.2 Naming

Distinguished names consist of sequence of Relative DN *cn=John Smith,ou=Austin,o=IBM,c=US (Leaf 2 Root)*
Ein Directory Information Tree (DIT) folgt einem graphischen oder organisiertem Schema.
Aliases can link non-leaf nodes.

Schema:

- Definiert welche Objekt-Klassen erlaubt sind
- Wo diese gespeichert sind
- Welche Attribute sie besitzen bzw. optional sind
- Von welchem Typ/welche Syntax jedes Attribut besitzt
- Das Schema muss für den Client lesbar sein

21.3 Functions/Operations

Zur Verteilung von strukturierten Informationen in einem Netzwerk (Authentifikation, Registry Informationen)

- Authentifikation(BIND/UNBIND, ABANDON)
  - BIND-Request includes LDAP version, the name the client wants to bind as, authentication type
  - Server responds with a status indication

  - UNBIND: Terminates a protocol session
  - ABANDON: MessageID to abandon
- Query (Suche, Einträge vergleichen)
  - Request includes baseObject, Scope, derefAliases, sizeLimit, timeLimit, attrsOnly, Filter, Attributes
  - Read and List implemented as searches
  - Compare: similar to search but returns T/F
- Update (Einträge hinzufügen/löschen, modifizieren)
- Other Requests: Search/modify/delete/change requests can include maximum time limits (and size limits in the case of search). There can be multiple pending requests.

21.4 Client and Server Interaction

- Client establishes session with server (BIND)
  - Hostname/IP and port number
  - Security(authentication, Encryption/Kerberos supported)
- Client performs operations ()Read/Update/Search)
- Client ends the session (UNBIND)
- Client can ABANDON the session

21.5 Security

- Clear text passwords (version 1)
- KERBEROS version 4 authentication
- SSL

21.6 Protocol Model

Clients performing protocol operations against servers.

- Client sends protocol request to server
- Server performs operation on directory
- Server returns response (results/errors)

Asynchronous behavior

21.7 Search Request Parameters

- base
  The base is the DN of root of the search. A server typically serves only below some subtree of the global DN namespace.
- scope
  - base – search only the base element.
  - onelevel – search all elements that are children of the base.
  - subtree – search everything in the subtree base
- size
  Limit on the number of entries to return from the search. A value of 0 means no limit.
- time
  Limit on number of seconds the search can take. Value of 0 means no limit.
- attributes
  A list of attributes that should be returned for each matched entry. NULL mean all attributes. Attribute names are strings.

- attrsonly
  a flag that indicates whether values should be returned
- search_filter
  - a search filter that defines the conditions that constitute a match.
  - Filters are text strings.
  - There is an entire RFC that describes the syntax of LDAP filters.
    (RFC 1558 oder 1960 → Widerspruch in den Folien)

21.8 Search Filters

- Restrict the search to those records that have specific attributes, or those whose attributes have restricted values.
- You can combine simple filters with boolean &,— and !

Beispiele:

- objectclass=* *match all records*
- cn=*dave* *matches any record with dave in the value of cn*
- (&(cn=*da)(email=*hotmail*))

Search Reply

- Each search can generate a sequence of Search Response records:
  Distinguished Name for record, list of attributes, possibly with list of values for each attributem, Result code
- LDAP includes an extensive error/status reporting facility.

LDAP API

1. Open connection with a server
   - int ldap_bind(...) -¿ LDAP_SUCCESS or ldap_errno
   - There are actually a bunch of ldap_bind functions
   - Synchronous calls all end in **s** - Asyn. without **s**
2. Authenticate
   -
3. Do some searches/modification/deletions
4. Close the connection

22 Authentication Protocols

Authentifikation ist der Prozess, bei dem versucht wird die Identität von jemanden nachzuweisen. Authentifikation ist ein wichtiger Bestandteil für Login-/Zugriffs-Prozesse, Security-Protokolle oder das Web of Trust.

22.1 Prinzipien

Es gibt eine Vielzahl von Protkollen, die unverschlüsselte Passwörter verwenden: sogenannte Plain-Text-Passwörter.
Telnet, FTP, POP3(teilweise, auch wenn MD5 als replay-anfällige Alternative möglich ist), IMAP4 (teilweise)

22.1.1 Wissensbasiert

Es muss eine gemeinsame geheime Information dem Authentifizierenden gezeigt werden.
Probleme: ein zu schwach gewähltes Passwort (Wörterbuch-Angriff, Brute-Force), der Nutzer vergisst sein Passwort oder das Passwort wurde kompromittiert.
Beispiele: PIN, Passwort, key phrase, personal information

22.1.2 Besitzbasiert

Es muss über das Netzwerk nachgewiesen werden, dass jemand den Nachweis besitzt. Es muss eine geheime Information innerhalb des Tokens sein. Diese darf jedoch nicht über das Netzwerk übertragen, sondern nachgewiesen werden.
Probleme: Der Nutzer kann das Token verlieren oder jemand anderes kopiert es sich.
Beispiele: SmartCard / SIM card, credit card, key, RFID token, phone, transponder, PC, DRM module (inside audio player, TV set etc.)

22.1.3 Biometrisch

Biometrische Eigentschaften können nicht vergessen, gestolen, belauscht oder gelernt werden. Der Umgang kann mit unter kompliziert sein. Üblicherweise ist dies die einzige Möglichkeit eine Person an ein Gerät zu binden. Alle anderen Verfahren können von jemand anderem ausgeführt werden. Retina, Iris, finger print, DNA, voice, signature

22.1.4 Zwei-Faktor-Authentifizierung

Erhöht die Sicherheit um ein weiteres, wenn zwei Verfahren kombiniert werden, auch bekannt als „strong authentication“

22.2 Probleme

- Wie kann man sich sicher gegenüber jemandem ausweisen, der sich selbst nicht ausgewiesen hat?
  Es könnte ein Man-in-the-middle Angriff sein, Replay-Attaken müssen verhindert werden, weshalb eine gegenseitige Authentifikation notwendig ist.
- Wie kann man sich durch Wissen oder Besitz authentifizieren, ohne das eigentliche shared secret zu übertragen.

Es gibt standardisierten Protokollen für eine Vielzahl von Anwendungen.

22.3 Angriffe

Man in the middle
Es befindet sich jemand zwischen zwei Kommunikationspartnern und wirkt für alle wie das Gegenüber. Es ist ein vertrauenswürdiger Dritter von Nöten, um solche Angriffe zu verhindern.

Replay
Bei diesem Angriff zeichnet der Angreifer eine verschlüsselte Authentifikations-Nachricht auf. Um Zugriff zu erhalten, spielt er sie erneut ab. Es ist notwendig für jede Authentifikations-Nachricht einen einzigartigen Wert zu verwenden. Idealer Weise ist dieser nur einmal nutzbar.

Denial of service
Hierbei handelt es sich gewisser Maßen um einen Akt des Vandalismus. Jemand verursacht die Kommunikation zwischen Server und Client zu stören bzw. zu blockieren. Dieser Angriff ist meist schwer zu vereiteln.

22.4 Challenge-Response

Dieses Verfahren ermöglicht es, dass das Wissen über ein geteiltes Geheimnis bewiesen werden kann, ohne dass es versendet werden muss. Dadurch sollen Replay-Angriffe verhindert werden.

23 Simple Network Management Protocol

SNMP ermöglicht das Verwalten und Überwachen von Netzwerkressourcen. Wichtige Teilnehmer sind **Agenten** und **Manager**. Agenten sind Applikationen, die auf Netzwerkteilnehmern (Hosts, Router, Drucker etc.) laufen und Informationen über Konfiguration und Status verwalten. Manager kontaktieren Agenten um Informationen abzufragen oder zu verändern.

Datenobjekte (z.B. *Tonerstand Cyan*) werden als Object Identifier (OID) bezeichnet. Diese sind baumartig-hierarchisch geordnet und können entsprechend auf zwei Weisen dargestellt werden: Zum einen numerisch, etwa `1.3.6.1.2.1.4.6.` und zum anderen als Zeichenkette `iso.org.dod.internet.mgmt.mib-2.ip.ipForwDatagrams`. Eine OID entspricht dabei einem Knoten im Baum. **Management Information Bases** (MIB) sind Textdateien, welche diese Objekte beschreiben. Sie werden häufig von Software eingelesen, um dem Nutzer weitere Informationen zur Verfügung zu stellen oder bekannte OIDs anzuzeigen.

Manager und Agenten kommunizieren über das SNMP. Agenten lauschen auf UDP-Port 161. An diesen senden Manager eine Anfrage (`get-request`) und erhalten eine Antwort (`get-respond`). Über einen **SNMP-Walk** kann eine Reihe von aufeinanderfolgenden OIDs abgefragt werden (`get-next-request`). Modifikationen können mit (`set-request`) vorgenommen werden.

Ebenfalls kann die Kommunikation vom Agenten initiiert werden. Bei bestimmten Ereignissen (z.B. Tonerfüllstand $< 10\%$ oder Last auf Switch $> 95\%$) sendet der Agent eine Nachricht an den Manager (UDP-Port 162). Dieses Verfahren wird **SNMP-Trap** genannt.

Der **Paketaufbau** für Get/Set-Nachrichten ist relativ simpel gehalten. Sie beginnen mit der Versionsnummer und der Community. Die Communities stellen eine Art Passwort dar und werden im Klartext übermittelt. Häufig akzeptieren viele Geräte `public` als Standard-Community. SNMPv3 hat Verbesserungen der Sicherheit adressiert.

Es folgt eine **SNMP PDU**. Wesentliche Bestandteile hierbei sind der Type (z.B. v1Get, v2Get etc), eine einmalige Request ID und eine Folge von Objekt-Wert-Paaren, die angefragt werden.

24 Mac-Sublayer

25 Mobile Netzwerke

26 Thomas Fragestunde

26.1 08.04

- Merkmale von Verbindungen
  1. Latenz
  2. Datenrate
  3. Jitter
     (Änderung der Latenz über die Zeit)
  4. Verfügbarkeit
  5. Fehlerrate
- Bandbreite wird in Hz und nicht in Mbit/s angegeben (Wlan bspw.)
- Ursachen für Jitter
  - Es ist notwendig zu warten, bis gesendet werden kann
  - Sich ändernde Routen, durch bspw. Wegänderungen, Datenaufteilung durch den Router
  - Zeit, Route, Warteschlangenproblematik, gemeinsam genutzte Medien, CSMA
- Warum Begrenzung der Übertragungsraten?
  - Physikalisch - Frequenzen
- Was begrenzt den max. Durchsatz eines Glasfaseranschlusses
  - unterschiedlich lange Wege
  - Kapazität (Zeit bis Ladung angekommen ist)
  - Transistoren schalten nicht so schnell
  - Signal-Rausch-Verhältnis

26.2 09.04

- Was begrenzt die Bandbreite?
  - Kapazität $\rightarrow$ Grenzfrequenz
- CSACD
  - Warten auf freies Medium
  - Es gibt jedoch keine Garantien
  - Verursacht Jitter

- Jitter Gegenmaßnahmen
  - Puffer (Warteschlange, FiFo)
  - Pakete werden in gleichmäßiger Reihenfolge erhalten
- Leitungs- vs. Paketorientiert
  - ?
- Möglichkeiten zur Beschreibung eines Standards
  - textuell (schwer)
  - Referenzimplementation (bspw. Bittorrent - aber berücksichtigt ggf. nicht alle Fälle)
  - Automaten (5-Tupel, Mealy/Moore)
- Zustandsbehaftete und zustandslose Protokolle
  - Beispiel TCP (closed, SYN, etc.) besitzt einen Zustand
  - Ein Zustand merkt sich die „Vorgeschichte “und damit was passiert ist. Somit ist ein Login immer zustandsbehaftet.
  - Werden die Zwischenschritte nicht gespeichert, ist es zustandslos.
  - Cookies bieten die Möglichkeit einen Zustand nachzurüsten.
- Autonome Systeme sind „Provider “, die miteinander über *Peering Points* verbunden sind
- Minimierung der Aufreihungslatenz
  - Entsteht dadurch, dass das Paket erst versendet wird, wenn es vollständig da ist
  - Lösung, Weiterleiten, sobald die ersten 6 Byte (Adresse) bekannt sind
- Problem von CRC (cyclic redundancy check)?
  - Es entsteht eine unnötige Belastung, wenn fehlerhafte Pakete weitergeleitet werden
- Wann sollte umgeschaltet werden zwischen den verschiedenen Switch-Modi?

26.3 22.04

- Implemenationen von Switches
  - Shared Memorie
  - Schaltmatrix (crossPoint → komplexe Schaltungen)
  - Gemeinsamer Bus
- Welche Geräte arbeiten auf welchen Layern?
  - Layer 3: Router
  - Layer 2: Switches
- Welche Dienste werden von welchen Layern erbracht?
  - Layer 3: Routing, logische Adressierung
  - Layer 2: Media Access
- –

26.4 04.05

- Routing?
  - Erfolgt auf dem Network-Layer
  - Dient der Kommunikation zwischen Geräten (IP kann als Protokoll genutzt werden)
  - Ziele sind das Verbinden von Netzen miteinander und das Finden des richtigen Weges.
  - IP bietet die Möglichkeit Netze logisch einzuteilen.
- DNS (Domain Name System)
  - Zum Auflösen einer Domain zu einer IP-Adresse
  - Es gibt verschiedene Ebenen ...
  - Beschreibung der Struktur: hierarchisch, mit Root, etc.
  - Anfragen werden mitunter gecached
- Migrationsstrategien von IPv4 nach IPv6
  (Tunneling, Dual Stack, Header Translation)
- Nachteile von Tunneling?
  - mehr Header-Daten
  - äußeres Protokoll bspw. nicht von Admins kontrolliert werden
- Funktionsweise eines Routers
  - Statisch (fest eingegebene Liste)
  - Dynamisch (Routingprotkolle, Topologieermittlung, Pfaderkennung)
- Möglichkeit zur Topologiebestimmung
  - Es werden Testnachrichten (TTL=1) im Netzwerk versendet
  - Auf diese Weise werden die Nachbarn bestimmt
  - Anschließend erfolgt die Weitergabe dieser Informationen: n-Hob-Nachbarn
  - Eine wichtige Bedingung hierbei ist, dass kein häufiger Wechsel erfolgt

26.5 13.05

- Wie lange Routingverfahren brauchen um zu konvergieren hängt von der Größe der Netzwerke ab. Mögliche Lösungen sind
  - Näherungsverfahren (Es muss nicht alles bekannt sein)
  - pro- bzw. reaktives Routing: Abhängig davon, ob Kenntnisse über das Netz vorhanden sind, oder nicht
  - Reduzierung der Graphenkomplexität - Einführen eines Backbones als „minimal dominating set“. Eine Menge, aus der alle Knoten erreichbar sind
- Das Count-to-infinity-Problem kann durch einen Distanzvektor behoben werden
- Netzwerkparameter, die die Güte beschreiben: Datenrate, Datensatz, Fehlerrate
- Was ist zu tun, wenn mehr Bedarf als Ressourcen bestehen
  - Flusskontrolle
  - Prioritäten festlegen
    (Ein Gespräch sollte eine Latenz von unter 1/45 Sekunden haben, 64 kbit/s + Header)

26.6 18.05

- Gegen was wirkt die FiFo-Queue?
  → Zur Jitter-Bekämpfung, bewirkt dafür aber Latenz
- Ursachen für Verzögerungen auf Layer 2 (Ethernet)
  - Shared Medium
  - Alle Teilnehmer senden, dadurch kommt es Kollisionen, die Teilnehmer müssen warten/lauschen/nach einer zufälligen Zeit erneut senden
- Beispielhafte Übertragungsraten
  - Audio CD
    - 44,1 kHz Abtastfrequenz, 16 Bit Abtastung (geringer Quantisierungsfehler), *2 (Stereo)
    - 1,5 Mbit/s ohne Fehlerkorrektur
  - Video 10-100 Mbit/s
  - Tippgeschwindigkeit: 200-300 Anschläge die Minute
- Ursachen für Latenz
  - Entfernung, Router/Switches, Aufreihung, Pakete erst abgesendet, wenn voll

26.7 25.05?

- RTP (Real-Time Transport Protocol)?
  - zur kontinuierlichen Übertragung von audiovisuellen Daten
  - Normalerweise über UDP
  - ? Kommt es zu einem zu hohen Verlust, erfolgt ein Kodecwechsel

26.8 03.06

- SMTP/POP3
  - Via TCP (Layer 4)
  - Port 25,584
  - Thomas erwartete detailiertes Wissen (bspw. befindet sich am Ende ein Punkt, etc.)
  - Sicherheit, Authentizität,... gibt es nicht
- Zustandsbehaftete Protokolle sind: POP3, IMAP, TCP, FTP
- IMAP4
  - Port 143
  - neuer und bietet Verwaltung, Ordner, etc.
  - „Besser“als POP3 und arbeitet auf dem Server (suchen, kopieren, etc.)
- Sinn von Subnetzmasken?
  - Anhand der Ziel-, der eigenen Adresse und der Netzmaske kann entschieden werden, ob sich der Empfänger im eigenen Netzwerk oder außerhalb befindet
- Nutzung von UPnP: Fernseher, Smartphones, Playstation, DSL-Router, Drucker, etc.
- ICMP?
  - Steuerungsnachrichten für Geräte im Internet
  - Meldungen, wenn Nachricht nicht zugestellt wurde (Host, Port, etc. unreachable)
  - TTL abgelaufen (bei IP-Paketen)
  - EchoRequest & EchoReply
  - „Mach-mal-langsam“Nachrichten
- Routingprotokolle

  - Topologieerkennung
  - Topologieverbreitung
  - günstigste Wege finden (Kostenfunktion)
  - Distanzvektorproblem
  - CountToInfinity $\rightarrow \infty = 16$
- typische Protokollfragen
  - VLAN
  - HTTP
  - FTP
  - Die zwei Modi von Switches
  - Routing Protokoll

26.9 15.06

- Wie groß sollten LAN-Segmente gewählt werden?
  - Nachteile von groß: Broadcast Domain
  - Nachteile von klein: Mehr Routing, mehr verpacken, Aufreihungslatenz, Prüfsummen, mehr Overhead
- Anforderungen an das Networkfilesystem (NFS): sicher, transparent, Mehrbenutzerbetrieb, schnell, Latenz
- RFC Lebenszyklus: ?
- XMPP
  - basiert auf XML zum Nachrichtenaustausch
  - Ende-zu-Ende chatten
  - Geringe Datenübertragungsrate (Chat, ohne Bilder)
  - Character-Encoding
  - Anwesenheitserkennung (on, off, tipping)

26.10 17.06

- Was ist P2P?
  - direkte Kommunikation via Vermittlungsstellen
  - Verfügbarkeit: auf einem PC (Lösung: Replikation)
  - dezentral: keine legislativen/judikativen Entscheidungen
  - Lastverteilung
- WebDav
  - Web-based Distributed Authoring and Versioning
  - Standard zur Bereitstellung von Dateien im Internet
  - setzt auf HTTP auf - und fügt hinzu
  - Mit WebDAV können ganze Verzeichnisse übertragen werden.
  - Einsatz für CMS
- Angaben von RoutNameServern für Topleveldomains
  - Country: de,...
  - General: edu, org,...
- LDAP?
  - hierarchische Datenbank
  - Aktives Verzeichnis: Nutzergruppenverwaltung
- Aufgabe des Sessionlayers

  - Zusammenfassung der verschiedenen Kommunikationspfade
  - Nutzung der Zustände (-behaftete: SSH,FTP,TCP,IMAP4,POP3)
- Begriff Authentisierung
  - "Dem System klar machen wer man ist"
  - Biometrisch, Token, Schlüssel, wissensbasiert

26.11 24.06

- Kerberos
  - Kerberos ist ein verteilter Authentifizierungsdienst
  - Kerberos soll eine sichere und einheitliche Authentifizierung in einem ungesicherten TCP/IP-Netzwerk auf sicheren Hostrechnern bieten.
- Medienzugriff auf Layer 2
  - Aloa (Bei Zugriff wird einfach zugerufen)
  - Slotted Aloa (Zugriff nur zu bestimmten Zeiten, dadurch geringere Kollisionen)
  - Bei Ethernet: CSMACD
    * Carrier Sense (Lauschen auf dem Medium)
    * Multiple Access
    * Collision Detection
  - Pure Aloa
  - GSM als kollisionsfreies Zugriffsverfahren (sobald die Verbindung besteht - auf Grund des exklusiven Medienzugriffs??)
- Multiplexverfahren
  - Methoden zur Signal- und Nachrichtenübertragung, bei denen mehrere Signale zusammengefasst und simultan über ein Medium übertragen werden.
    (Raum, Frequenz, Zeit, Code)

26.12 29.06

- Faktoren die in eine Kostenfunktion beim Routing miteinbezogen werden können
  - Topologiekenntnisse
  - Durchsatz/Datenrate
  - Latenz
  - Reale Kosten
- Dijkstra Algorithmus
- Hidden Station Problem
  - RTC/CTS lohnt sich, wenn die Pakete klein genug sind - verglichen zu den Nutzerdaten
  - OLSR (Optimized Link State Routing)
    (Reduzierung der Graphenkomplexität - Backbone Bildung)

26.13 08.07

- MIB (bei SNMP)
  - Management Information Base (deutsch: Verwaltungsinformationsbasis) beschreibt die Informationen, die über ein Netzwerk-Management-Protokoll abgefragt oder modifiziert werden können.
  - Das Simple Network Management Protocol ist ein Netzwerkprotokoll, um Netzwerkelemente von einer zentralen Station aus überwachen und steuern zu können.
  - Es werden Schlüssel beschrieben, die zur Speicherung des „Gesundheitszustandes"von Geräten dienen.
  - Eine MIB-Datenbank erklärt, für welche Information ein Schlüssel steht.

- Hemming-Distanz
  - Der Hamming-Abstand ist ein Maße für die Unterschiedlichkeit von Zeichenketten.
  - Die Distanz zweier Blöcke mit fester Länge ist dabei die Anzahl der unterschiedlichen Stellen.
  - HD wird zur Fehlererkennung und zur Fehlerkorrektur benutzt.
  - Ob eine Fehlererkennung oder -korrektur stattfinden kann, hängt von der Hamming-Distanz ab.
  - FEC - forward error correction
    - Am einfachsten: 2-3 mal senden
    - Fehler erkennen: Distanz von 2
    - Fehler korrigieren: Distanz von 3
- Aktives vs. passives Scanning
  - Passiv: Der AP sendet Beacons mit Daten aus
  - Aktiv: Der Client fragt beim AP nach
  - Wenn ein Client wechseln möchte, wäre er solange offline bis er ein Beacon erhalten würde. Durch aktives nachfragen kann die „Hand off“Zeit reduziert werden.
- RTS/CTS (Ready to send / clear to send)
- Bluetooth
  - Musik, Staubsauger, Eingaben, Smartwatches, Datenübertragung,...
  - Kopfhörer Datenübertragung
    - Bits pro Sekunde = Samplerate * Samplebreite * Kanäle
    - 2*48 kHz (Nyquist) - Audio-CD: 44,1 kHz
    - 16 Bit
    - 2 Kanäle
    - $\approx$ 140/150 kBit/s

Literatur

[1] Janet Abbate. *Inventing the internet*. MIT press, 2000.

[2] S. Alexander. Rfc 2132-dhcp options and bootp vendor extensions. 1997. http://tools.ietf.org/html/rfc2132.

[3] Paul Baran et al. On distributed communications. *Volumes I-XI, RAND Corporation Research Documents, August*, pages 637–648, 1964.

[4] S Bellovin. The security flag in the ipv4 header. Technical report, RFC 3514, 2003. https://tools.ietf.org/html/rfc3514.

[5] M Belshe and R Peon. M. thomson,"hypertext transfer protocol version 2. Technical report, RFC 7540, May, 2015. http://tools.ietf.org/html/rfc7540.

[6] Tim Berners-Lee, Roy Fielding, and Larry Masinter. Rfc 2396: Uniform resource identifiers (uri): Generic syntax, august 1998. *Status: Draft Standard*. http://tools.ietf.org/html/rfc2396.

[7] Andreas Brandstädt. *Graphen und Algorithmen*. B.G. Teubner Stuttgart, 1994.

[8] T Bray. An http status code to report legal obstacles draft-ietf-httpbis-legally-restricted-status-04. 2015. https://www.ietf.org/id/draft-ietf-httpbis-legally-restricted-status-04.txt.

[9] B. et al Cain. Rfc 7736-internet group management protocol, version 3. 2002. http://tools.ietf.org/html/rfc3376.

[10] Stuart Cheshire, B Aboba, and E Guttman. Rfc 3927: Dynamic configuration of ipv4 link-local addresses. *IETF standard*, 2005. https://tools.ietf.org/html/rfc3927.

[11] T Clausen and P Jacquet. Rfc 3626. *Optimized link state routing protocol (OLSR)*, 2003. http://tools.ietf.org/html/rfc3626.

[12] Douglas Comer. *TCP-IP: Konzepte, Protokolle, Architekturen*. mitp Verlags GmbH & Co. KG, 2011.

[13] M. Crispin. Rfc 1730- internet message access protocol - version 4. 1994. http://tools.ietf.org/html/rfc1730.

[14] B. Croft. Rfc 951- bootstrap protocol (bootp). 1985. http://tools.ietf.org/html/rfc951.

[15] Leslie Daigle. Whois protocol specification. 2004. http://tools.ietf.org/html/rfc3912.

[16] Stephen E Deering. Internet protocol, version 6 (ipv6) specification. 1998. https://tools.ietf.org/html/rfc2460.

[17] LP Deutsch. Rfc 1951: Deflate compressed data format specificaiton v1. 3, 1996. http://tools.ietf.org/html/rfc1951.

[18] R. Droms. Rfc 2131- dynamic host configuration protocol. 1997. http://tools.ietf.org/html/rfc2131.

[19] L. Dusseault. Rfc 4918-http extensions for web distributed authoring and versioning (webdav). 2007. http://tools.ietf.org/html/rfc4918.

[20] H Eidnes and G de Groot. P. vixie,"classless in-addr. arpa delegation. Technical report, BCP 20, RFC 2317, March, 1998. http://tools.ietf.org/html/rfc2317.

[21] Kevin R Fall and W Richard Stevens. *TCP/IP illustrated, volume 1: The protocols*. addison-Wesley, 2011. Verfügbar auf http://tinyurl.com/tcp-ip-illustrated (aus Uni-Netz).

[22] P Faltstrom, P Hoffman, and A Costello. Rfc 3490: internationalizing domain names in applications (idna). *Network Working Group, IETF*, 2003. http://tools.ietf.org/html/rfc3490.

[23] Adrian Farrel. *The Internet and its protocols: a comparative approach*. Morgan Kaufmann, 2004.

[24] R Fielding, J Gettys, J Mogul, H Frystyk, L Masinter, P Leach, and T Berners-Lee. Rfc 2616. *Hypertext Transfer Protocol–HTTP/1.1*, 2(1):2–2, 1999. http://tools.ietf.org/html/rfc2616.

[25] R Fielding and J Reschke. Rfc 7230-hypertext transfer protocol (http/1.1): Message syntax and routing." h ttp. 2014. http://tools.ietf.org/html/rfc7230.

[26] R Fielding and J Reschke. Rfc 7231-hypertext transfer protocol (http/1.1): Semantics and content. 2014. http://tools.ietf.org/html/rfc7231.

[27] R Fielding and J Reschke. Rfc 7232-hypertext transfer protocol (http/1.1): Conditional requests. 2014. http://tools.ietf.org/html/rfc7232.

[28] R Fielding and J Reschke. Rfc 7233-hypertext transfer protocol (http/1.1): Range requests. 2014. http://tools.ietf.org/html/rfc7233.

[29] R Fielding and J Reschke. Rfc 7234-hypertext transfer protocol (http/1.1): Caching. 2014. http://tools.ietf.org/html/rfc7234.

[30] R Fielding and J Reschke. Rfc 7235-hypertext transfer protocol (http/1.1): Authentication. 2014. http://tools.ietf.org/html/rfc7235.

[31] Ross Finlayson, Timothy Mann, Jeffrey C Mogul, and Marvin Theimer. A reverse address resolution protocol. Technical report, 1984. https://tools.ietf.org/html/rfc903.

[32] Thomas Fuhrmann. Performance of scalable source routing in hybrid manets. In *Wireless on Demand Network Systems and Services, 2007. WONS'07. Fourth Annual Conference on*, pages 122–129. IEEE, 2007.

[33] Vince Fuller, Tony Li, Jessica Yu, and Kannan Varadhan. Rfc 1519: Classless inter-domain routing (cidr): an address assignment and aggregation strategy, 1993. https://tools.ietf.org/html/rfc1519.

[34] Network Working Group et al. Rfc 1933, 1996. http://tools.ietf.org/html/rfc1933.

[35] Network Working Group et al. Rfc 1950: Zlib compressed data format specification version 3.3. *ZLIB Compressed Data Format Specification. Version*, 3, 1996. http://tools.ietf.org/html/rfc1950.

[36] M. et al Handley. Rfc 2543-sip: Session initiation protocol. 1999. http://tools.ietf.org/html/rfc2543.

[37] C Hedrick, " Network Working Group Routing information protocol, et al. Rfc 1058. 1988. https://tools.ietf.org/html/rfc1058.

[38] Paul Hoffman and Scott Bradner. Defining the ietf. 2002. http://www.ietf.org/rfc/rfc3233.txt.

[39] Van Jacobson, Robert Braden, Dave Borman, M Satyanarayanan, JJ Kistler, LB Mummert, and MR Ebling. Rfc 1323: Tcp extensions for high performance, 1992. https://tools.ietf.org/html/rfc1323.

[40] B Kantor. Rfc 1258: Bsd rlogin, 1991. http://tools.ietf.org/html/rfc1258.

[41] Rohit Khare and Scott Lawrence. Rfc 2817: upgrading to tls within http/1.1, 2000. http://tools.ietf.org/html/rfc2817.

[42] Ed Krol. Hitchhikers guide to the internet. 1989. http://www.ietf.org/rfc/rfc1118.txt.

[43] Cricket Liu and Paul Albitz. *DNS and Bind*. Ö'Reilly Media, Inc.", 2006.

[44] G Malkin. Rfc 2453: Rip version 2. *Request for Comments*, 2453, 1998. https://tools.ietf.org/html/rfc2453.

[45] G Malkin and R Minnear. rfc 2080: Ripng for ipv6, 1997. https://tools.ietf.org/html/rfc2080.

[46] D Mills, J Martin, J Burbank, and W Kasch. Rfc 5905: Network time protocol version 4: Protocol and algorithms specification. *Internet Engineering Task Force*, 2010. http://tools.ietf.org/html/rfc5905.

[47] David L Mills. Rfc 1305: Network time protocol (version 3) specification. *Implementation and Analysis*, 1992. http://tools.ietf.org/html/rfc1305.

[48] David L Mills. Simple network time protocol (sntp) version 4 for ipv4, ipv6 and osi. 2006. http://tools.ietf.org/html/rfc4330.

[49] Paul Mockapetris. Rfc 1034: Domain names: concepts and facilities (november 1987). *Status: Standard*, 2003. https://tools.ietf.org/html/rfc1034.

[50] Paul Mockapetris. Rfc 1035—domain names—implementation and specification, november 1987. *URL http://www. ietf. org/rfc/rfc1035. txt*, 2004. https://tools.ietf.org/html/rfc1035.

[51] Jeffrey Clifford Mogul and Jon Postel. Internet standard subnetting procedure. Technical report, 1985. http://tools.ietf.org/html/rfc950.

[52] J. et al Myers. Rfc 1939-post office protocol - version 3. 1996. http://tools.ietf.org/html/rfc1939.

[53] T. et al Narten. Rfc 4861-neighbor discovery for ip version 6 (ipv6). 2007. http://tools.ietf.org/html/rfc4861.

[54] I Nazar. The hyper text coffee pot control protocol for tea efflux appliances (htcpcp-tea). 2014. http://www.ietf.org/rfc/rfc2324.txt.

[55] Charles Perkins, Elizabeth Belding-Royer, Samir Das, et al. Rfc 3561-ad hoc on-demand distance vector (aodv) routing. *Internet RFCs*, pages 1–38, 2003. http://tools.ietf.org/html/rfc3561.

[56] David C Plummer. Rfc 826: An ethernet address resolution protocol. *InterNet Network Working Group*, 1982. http://tools.ietf.org/html/rfc826.

[57] J.B. Postel. Rfc 821- simple mail transfer protocol. 1982. http://tools.ietf.org/html/rfc821.

[58] Jon Postel. Rfc 768: User datagram protocol, august 1980. *Status: Standard*, 1980. https://tools.ietf.org/html/rfc768.

[59] Jon Postel. Rfc 793: Transmission control protocol, september 1981. *Status: Standard*, 88, 2003. https://tools.ietf.org/html/rfc793.

[60] Jon Postel et al. Rfc 791: Internet protocol. 1981. http://tools.ietf.org/html/rfc791.

[61] Jon Postel et al. Rfc 792: Internet control message protocol. *InterNet Network Working Group*, 1981. http://tools.ietf.org/html/rfc792.

[62] Jon Postel and Joyce Reynolds. Rfc 959: File transfer protocol, 1985. http://tools.ietf.org/html/rfc959.

[63] Jarno Rajahalme, Shane Amante, Sheng Jiang, and Brian Carpenter. Ipv6 flow label specification. 2011. https://tools.ietf.org/html/rfc6437.

[64] Jarno Rajahalme, Alex Conta, Brian E Carpenter, and Steve E Deering. Rfc 3697: Ipv6 flow label specification, mar 2004. https://tools.ietf.org/html/rfc3697.

[65] Y Rekhter, T Li, and S Hares. Rfc 4271: Border gateway protocol 4, 2006. http://tools.ietf.org/html/rfc4271.

[66] J. et al. Satran. Rfc 3720-internet small computer systems interface (iscsi). 2004. http://tools.ietf.org/html/rfc3720.

[67] S. et al. Shelper. Rfc 3530-network file system (nfs) version 4 protocol. 2003. http://tools.ietf.org/html/rfc3530.

[68] M Slavitch. Definitions of managed objects for drip-type heated beverage hardware devices using smiv2. Technical report, 1998. http://www.ietf.org/rfc/rfc2325.txt.

[69] Daniel Sokolov. Neuer Status Code 451 zeigt Zensur an. 2015. http://www.heise.de/newsticker/meldung/Neuer-Status-Code-451-zeigt-Zensur-an-3052138.html.

[70] K Sollins. Rfc 1350: The tftp protocol (revision 2). *Network Working Group, MIT*, 1992. http://tools.ietf.org/html/rfc1350.

[71] R. Stewart. Rfc 4960-stream control transmission protocol. 2007. http://tools.ietf.org/html/rfc4960.

[72] Andrew S Tanenbaum. Computer networks, 4-th edition. *ed: Prentice Hall*, 2003.

[73] wissen.de. Timeout. *Lexikon*. http://www.wissen.de/lexikon/timeout-informatik.

[74] Tatu Ylonen and Chris Lonvick. The secure shell (ssh) protocol architecture. 2006. http://tools.ietf.org/html/rfc4251.

[75] Robert H Zakon. Hobbes' internet timeline. 1997. http://www.ietf.org/rfc/rfc2235.txt.

Glossar

ACE ASCII Compatible Encoding.

ARPA Advanced Research Projects Agency.

ATDN AOL Transit Data Network.

ATM Asynchronous Transfer Mode.

ccTLD Country Code Top Level Domain.

CFI Canonical Formati Indicator.

CIDR Classless Inter-Domain Routing.

CRC Cyclic Redundancy Check.

CSMA/CD Carrier Sense Multiple Acces / Collision Detection.

DE-CIX German Commercial Internet Exchange.

DENIC Deutsches Network Information Center.

DHCP Dynamic Host Configuration Protocol.

DNS Domain Name System.

FDD Frequency Divsion Duplex.

FTP File Transfer Protocol.

gTLD Generic Top Level Domain.

GX Global Crossing.

HTTP Hypertext Transfer Protocol.

ICANN Internet Corporation for Assigned Names and Numbers.

IETF Internet Engineering Task Force.

MAC Medium Access Control.

NAT Network Address Translation.

NIC Name Information Center.

NTP Network Time Protocol.

NVT Network Virtual Terminal.

RIPE Réseaux IP Européens.

RIPE NCC RIPE Coordination Centre.

RSA Rivest, Shamir und Adleman.

SCTP Stream Control Transmission Protocol.

SFD Start Frame Delimiter.

SMTP Simple Mail Transfer Protocol.

SNMP Simple Network Management Protocol.

SNTP Simple Network Time Protocol.

SSH Secure Shell.

TCP Transmission Control Protocol.

TDD Time Division Duplex.

TFTP Trivial File Transfer Protocol.

TLD Top Level Domain.

TLS Transport Layer Security.

ToS Type of Service.

TPID Tag Protocol Identifier.

TTL Time to Live.

UDP User Datagram Protocol.

URI Uniform Resource Identifier.

URL Uniform Resource Locator.

URN Uniform Resource Name.

VCI Virtual Channel Identifer.

VID VLAN Identifier.

VPI Virtual Path Identifier.

WWW World Wide Web.