Zusammenfassung - Kryptographie

Marc Meier

6. November 2015

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, 2]

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 9) 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 [6]

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.

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.

2 Protokolle

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

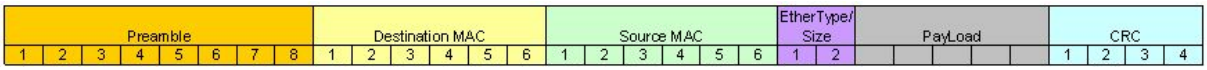
Network Layer : Paket

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) und 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 sendene 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)

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.

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

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

2.4.4 Trunking / Link Aggregation

Ermöglicht die Zusammenfassung mehrerer Ports zur Erhöhung des Datensatzes.

2.5 Asynchronous Transfer Mode

2.6 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)

2.6.1 IPv4

Wurde im RFC 791[10] 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)

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

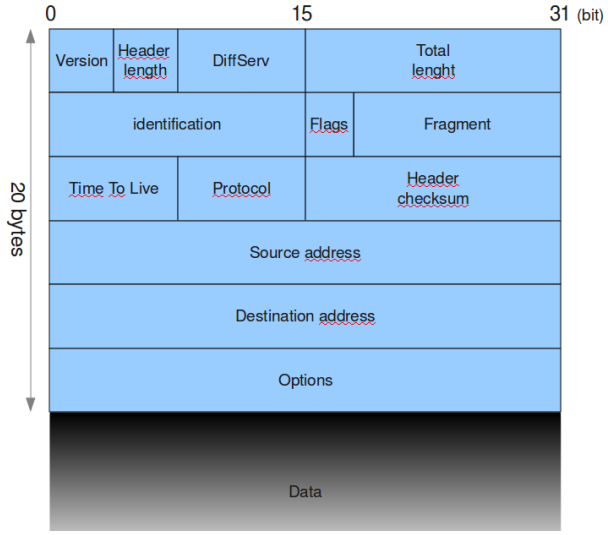
Identification : Falls ein Paket fragmentiert wird, haben alle Fragmente die selbe Identification.

Flags : Reserved[3], 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

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[5] 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)

2.6.2 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 :

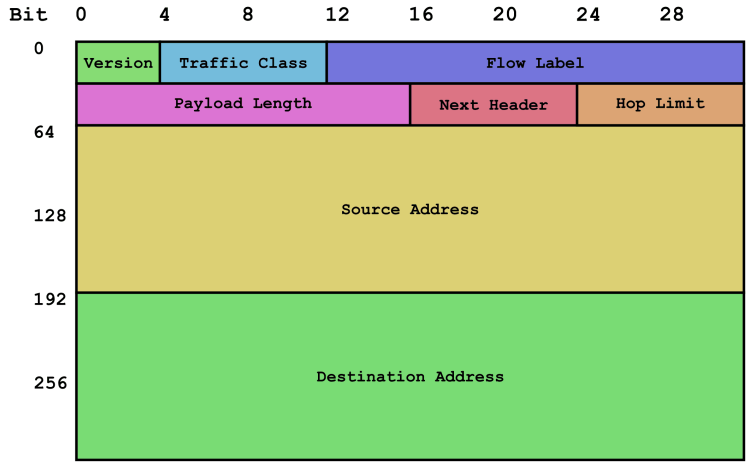
- 0 uncharakterisierter Verkehr
- 1 „Füllmaterial", z.B. Newsgroups
- 2 zeitunkritischer Verkehr, z.B. EMail
- 3 reserviert
- 4 Mengendaten, z.B. FTP, NFS
- 5 reserviert
- 6 Interaktive Anwendungen, z.B. telnet
- 7 Steuerung, z.B. SNMP

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. [12, 11]

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[4] bietet beispielsweise:

[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

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



2.7 User Datagram Protocol

Bei UDP handelt es sich um ein Layer-4-Protokoll.[8] 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.

2.8 Transmission Control Protocol

TCP ist ebenfalls ein Layer-4-Protokoll.[9] Es garantiert eine Ankunft der Pakete in korrekter Reihenfolge. Clienten sehen die Verbindung als bidirektionalen Datenstrom, tatsächlich findet die Kommunikation über Pakete statt.

2.8.1 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 [7] auch bis zu 1GB, dann Linksverschiebung um bis zu 14 Bits, $2^{14} \cdot 64k = 1G$).

2.8.2 Zustände

Zum Aufbau einer Verbindung wird der **3-Way-Handshake** durchgeführt: → SYN — ← SYN + ACK — *rightarrow* ACK. Ein Timeout findet typischerweise nach 75 Sekunden statt.

Zum Abbau der Verbindung genügt das Senden und Quittieren eines FIN.

Literatur

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

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

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

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

[5] 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`.

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

[7] 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`.

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

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

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

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

[12] 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`.