Rechnernetze und Netzwerkprogrammierung

Aufgabenblatt 2 (Gruppe: Walcher, Dworski, Aigensberger)

Ü 2.1 Internetarchitekturen

Allgemeine Architektur des Internets und Zugangsmöglichkeiten:

Die Architektur des Internets beruht auf einem dezentralen und hierarchischen Modell, das mehrere Schichten umfasst. Sie gliedert sich im Wesentlichen in folgende Hauptkomponenten:

1. **Endgeräte:** Diese umfassen Computer, Smartphones, Server und IoT-Geräte, die als Endpunkte im Netzwerk fungieren. Sie ermöglichen den Benutzern den Zugriff auf Informationen und Dienste im Internet.
2. **Zugangsschicht:** Diese Schicht stellt die Verbindung zwischen Endgeräten und dem Internet her. Zu den gängigen Zugangstechnologien zählen:
    - **Kabelgebundene Verbindungen:** Ethernet, DSL, Kabelmodem.
    - **Drahtlose Verbindungen:** WLAN, Mobilfunknetze (2G, 3G, 4G, 5G).
3. **Netzwerkschicht:** Diese Schicht verwaltet die Weiterleitung von Datenpaketen zwischen verschiedenen Netzwerken. Sie verwendet Protokolle wie IP (Internet Protocol), um die Adressierung und das Routing von Daten zu ermöglichen.
4. **Transport- und Anwendungsschicht:** Die Transportschicht (z.B. TCP, UDP) gewährleistet die zuverlässige oder unzuverlässige Übertragung von Daten zwischen Anwendungen. Die Anwendungsschicht bietet Protokolle für spezifische Dienste, wie HTTP für Webseiten, SMTP für E-Mail und FTP für Dateiübertragungen.
5. **Infrastruktur:** Zu den grundlegenden Infrastrukturen gehören Router, Switches, Gateways und Serverfarmen, die für das Routing, die Datenweiterleitung und das Hosting von Diensten zuständig sind.

Ü 2.2 Leitungsvermittelnde vs. Paketvermittelnde Netzwerke

Unterschiede zwischen leitungsvermittelnden und paketvermittelnden Netzwerken:

- **Leitungsvermittelnde Netzwerke (Circuit-switched Networks):**
    - **Definition:** Bei dieser Technologie wird eine dedizierte Verbindung (Leitung) zwischen den Kommunikationspartnern für die gesamte Dauer der Kommunikation aufrechterhalten.
    - **Beispiel:** Telefonnetz (Festnetz), wo eine physische Leitung zwischen Anrufer und Angerufenen aufgebaut wird.
    - **Vorteile:**
        - Stabile und vorhersehbare Übertragungsrate.
        - Geringe Latenz während der Verbindung.
    - **Nachteile:**
        - Ineffiziente Ressourcennutzung, da die Leitung während der gesamten Kommunikation reserviert ist, auch wenn keine Daten übertragen werden.
        - Hohe Kosten bei der Bereitstellung und Wartung.
- **Paketvermittelnde Netzwerke (Packet-switched Networks):**

- **Definition:** Daten werden in kleine Pakete zerlegt und über ein gemeinsames Netzwerk verschickt. Jedes Paket kann einen unterschiedlichen Pfad zum Ziel nehmen.
- **Beispiel:** Das Internet, wo Daten in IP-Paketen übertragen werden.
- **Vorteile:**
  - Effiziente Nutzung der Netzwerkressourcen, da Bandbreite dynamisch zugewiesen wird.
  - Flexible und robuste Netzwerkinfrastruktur.
- **Nachteile:**
  - Variable Latenz und mögliche Paketverluste.
  - Komplexe Routing-Algorithmen erforderlich.

Ü 2.3 ISO/OSI-Referenzmodell

Das ISO/OSI-Referenzmodell:

Das ISO/OSI-Modell (Open Systems Interconnection Model) ist ein theoretisches Rahmenwerk zur Beschreibung von Netzwerkprotokollen und -kommunikation. Es besteht aus sieben Schichten:

1. **Schicht 1: Bitübertragungsschicht (Physical Layer)**
   - **Funktion:** Übertragung von Rohdaten über ein physisches Medium (z.B. Kabel, Funk).
   - **Schnittstelle:** Hardware-Schnittstellen, wie elektrische Signale, optische Signale.
2. **Schicht 2: Sicherungsschicht (Data Link Layer)**
   - **Funktion:** Fehlererkennung und -korrektur, Rahmenbildung, MAC-Adressenverwaltung.
   - **Schnittstelle:** Logische Verbindungen zu direkten Nachbarn (z.B. Ethernet, PPP).
3. **Schicht 3: Netzwerkschicht (Network Layer)**
   - **Funktion:** Routing von Datenpaketen, logische Adressierung (z.B. IP-Adressen).
   - **Schnittstelle:** Interaktion mit der Transport- und Sicherungsschicht (z.B. IP).
4. **Schicht 4: Transportschicht (Transport Layer)**
   - **Funktion:** Gewährleistung der Datenintegrität, Segmentierung und Wiederherstellung von Daten.
   - **Schnittstelle:** Verbindet Anwendung und Netzwerk (z.B. TCP, UDP).
5. **Schicht 5: Sitzungsschicht (Session Layer)**
   - **Funktion:** Verwaltung von Sitzungen und Verbindungen zwischen Anwendungen.
   - **Schnittstelle:** Bereitstellung von Sitzungsmanagement.
6. **Schicht 6: Darstellungsschicht (Presentation Layer)**
   - **Funktion:** Datenformatierung, Verschlüsselung und Kompression.
   - **Schnittstelle:** Interaktion zwischen der Anwendung und der Sitzungsschicht.
7. **Schicht 7: Anwendungsschicht (Application Layer)**

- **Funktion:** Bereitstellung von Netzwerkdiensten für Anwendungen (z.B. E-Mail, Web).
- **Schnittstelle:** Interaktion mit Endbenutzern und Anwendungen (z.B. HTTP, FTP).

Vertikale und horizontale Schnittstellen:

- **Vertikale Schnittstellen:** Betreffen die Interaktion zwischen benachbarten Schichten, wobei jede Schicht auf die Dienste der darunter liegenden Schicht zugreift.
- **Horizontale Schnittstellen:** Betreffen die Kommunikation zwischen gleichwertigen Schichten auf unterschiedlichen Geräten.

Ü 2.4 ISO/OSI- vs. TCP/IP-Referenzmodell

Das TCP/IP-Referenzmodell:

Das TCP/IP-Modell ist eine praxisorientierte Netzwerkarchitektur, die weniger Schichten als das ISO/OSI-Modell hat. Es besteht aus vier Schichten:

1. **Schicht 1: Netzwerkschicht (Network Interface Layer)**
   - Entspricht der Bitübertragungs- und Sicherungsschicht (OSI).
   - Beispielprotokolle: Ethernet, Wi-Fi, ARP.
2. **Schicht 2: Internetschicht (Internet Layer)**
   - Entspricht der Netzwerkschicht (OSI).
   - Beispielprotokolle: IP (IPv4, IPv6), ICMP.
3. **Schicht 3: Transportschicht (Transport Layer)**
   - Entspricht der Transportschicht (OSI).
   - Beispielprotokolle: TCP, UDP.
4. **Schicht 4: Anwendungsschicht (Application Layer)**
   - Entspricht den Schichten 5-7 (OSI).
   - Beispielprotokolle: HTTP, FTP, SMTP, DNS.

Unterschiede zwischen ISO/OSI und TCP/IP:

- Das TCP/IP-Modell hat eine vereinfachte Struktur mit weniger Schichten.
- TCP/IP wurde ursprünglich für das Internet entwickelt, während ISO/OSI ein theoretisches Modell ist.

Ü 2.5 TCP vs. UDP

Funktionalität der Transportschicht:

Die Transportschicht ist dafür verantwortlich, Daten zwischen Endsystemen zu übertragen und sicherzustellen, dass sie korrekt ankommen. Sie bietet zwei Hauptprotokolle: TCP (Transmission Control Protocol) und UDP (User Datagram Protocol).

Unterschiede zwischen TCP und UDP:

- **TCP (Transmission Control Protocol):**
  - **Verbindung:** Verbindungsorientiert; stellt eine Sitzung zwischen Sender und Empfänger her.
  - **Zuverlässigkeit:** Garantiert die Zustellung von Paketen durch Fehlererkennung, Sequenzierung und Wiederholungsmechanismen.
  - **Anwendungsprotokolle:** HTTP, FTP, SMTP, Telnet.
  - **Nachteile:** Höhere Latenz aufgrund der Verbindungsverwaltung und der Fehlerkorrektur.
- **UDP (User Datagram Protocol):**
  - **Verbindung:** Verbindungslos; es gibt keine vorherige Sitzung.
  - **Zuverlässigkeit:** Keine Garantie für die Zustellung oder Reihenfolge von Paketen.
  - **Anwendungsprotokolle:** DNS, DHCP, VoIP, Streaming-Dienste.
  - **Vorteile:** Geringere Latenz und Overhead, was für zeitkritische Anwendungen vorteilhaft ist.

Ü 2.6 Ein fiktives Applikationsprotokoll

Network Working Group
Request for Comments: XXXX
Category: Draft

Daniel Dworski
AAU
October 2024

Application Layer Protocol For ATMs

Status of This Memo

This document specifies a protocol used for the communication between ATMs and their central server. It is not any kind of standard.

Abstract

In this document the protocol is defined, which is meant for a save communication between the ATM and its central server. It will not include which encryption algorithms or standards are best for the communication, but define what information shall be transferred in order to minimize risk.

Table of Contents

Introduction

The ATM should minimize using valuable data when communication with the central server. In order to do so the protocol is session based. This means the server should be able to save a session key for every ATM which is registered to it.

Protocol Definition

There are three types of messages the ATM can send to the central server:

1. CREATE_SESSION_REQUEST
2. WITHDRAW_REQUEST
3. CLOSE_SESSION_REQUEST

Each one of them has the basic structure:
|Messagetype|Timestamp|ATM_ID|Session_ID|
Additionally, when requesting to create a session the number and pin of the entered card is being transferred. On the other hand, the WITHDRAW_REQUEST contains the amount of money the customer wishes to withdraw.

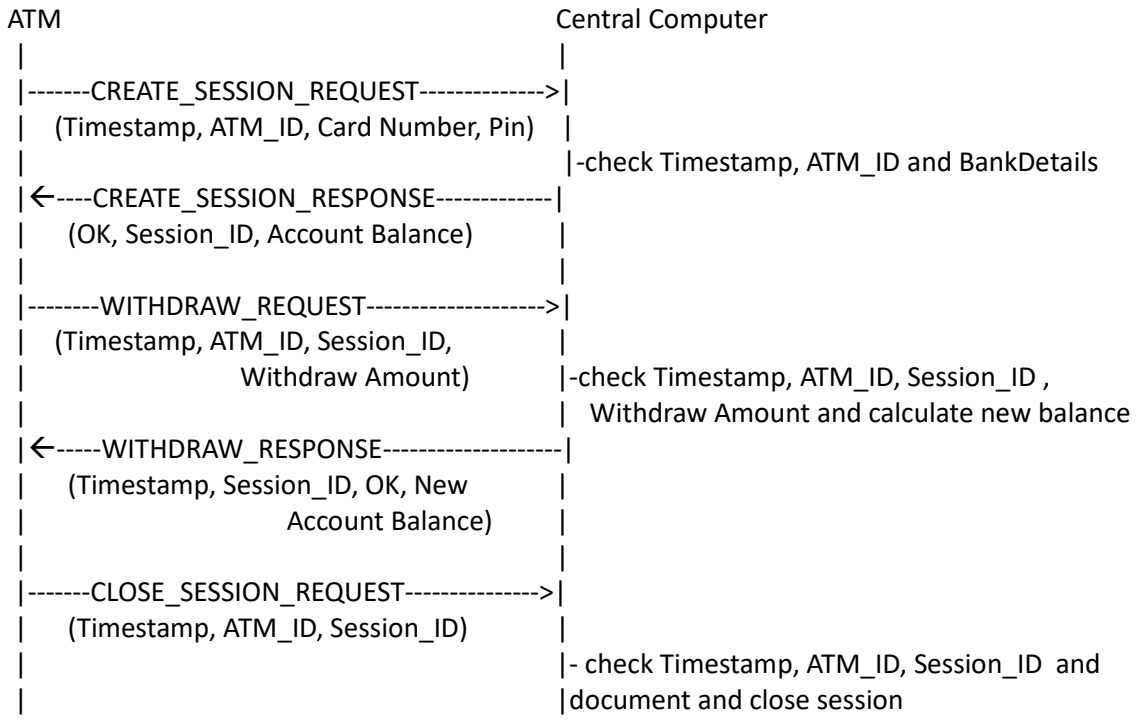
The server can only answer with two kinds of responses:

1. CREATE_SESSION_RESPONSE
2. WITHDRAW_RESPONSE

The basic structure of these responses is:
|Messagetype|Timestamp|Session_ID|Status|
Dependent on the status of the response an additional field containing the error message may be added. When successfully identifying the ATM and creating a session, its ID and the balance of the involved account will be added to the response. The WITHDRAW_RESPONSE only contains additionally the new balance of the account.

Not included is the encryption of the defined data, because this protocol only aims to minimize the critical communication between the ATM and the central server. The save encryption and decryption is the responsibility of the transport layer. Please make sure you apply an up-to-date encryption algorithm on the transport layer.

Visual Procedure Of The Protocol



Security Considerations

Considering there is no encryption algorithm or standard defined, make sure that the connection is End-to-end save. This document assumes the correct measures have already been made in the transport layer.

IANA Considerations

This protocol is not meant to be applied in the real world so no action is required by IANA.

References

RFC 3: Documentation Conventions

RFC Editor, "How to Write an RFC – A Tutorial", IETF-62, Minneapolis, MN, USA, March 2005. Available at: https://www.ietf.org/proceedings/62/slides/editor-0.pdf.

Ü 2.7 IEEE 802.11-Wireless-LAN-Technik

Grundlegend erstmal einen Router und einen Access Point, dass auch alle Winkel im Haus eine ausreichende Netzwerkverbindung haben:

- AVM FRITZ!Box 7590 AX 242.00€
- AVM FRITZ!Powerline 1240E/1000E WLAN Set 127.99€

Anschließend zwei Laptops, die mit WLan 802.11 sich verbinden können:

- MacBook Air 15.3 Inch M38 GB Shared Memory, 256 GB SSD 2x 1,350.25€

Und ein Handy, um mobil zu sein:

- Apple iPhone 15 787.56€

Weitere Austattung beträgt einen Fernsehe, Drucker und zuletzt noch einen Nas:

- LG 55UR78006LK 55 Inch UHD TV 60 Hz Smart TV 427.50€
- Brother DCP-J1050DW 3-in-1 DIN A4 Multifunction Printer 123.02€
- Synology DS223 2-Bay Diskstation NAS 273.47

In Summe kommt man auf 4 682.04€

Ü 2.8 Streaming

Unternehmen, die ihre Streaming Services anbieten haben die angebotenen Videos auf ihren Server auf unterschiedlicher Qualität gespeichert und übertragen dieses durch das Internet zu ihren Kunden. Dabei verändern sie die gesendete Qualität je nachdem, wie gut die Internetverbindung des Kunden ist. Dabei gibt es viele solche Unternehmen:

1. Youtube verwendet den HTML5-Videoplayer, der das HLS Protokoll standardmäßig eingestellt hat. Doch auch RTMP, RTMPS und DASH können verwendet werden um die Videos davon zu streamen. [https://www.dacast.com/blog/video-streaming-protocol/]

2. Netflix verbindet den Kunden mit ihrem nahesten CDN Server. Dieser überträgt anschließend das Video in kleinen Blöcken durch das HLS Protokoll. Dabei beobachtet es das Netzwerk und den User und verändert den Server, die Bitrate oder andere Optimierungen. [https://medium.com/@aadityabchatterjee/from-pixels-to-perfection-a-deep-dive-into-netflixs-engine-algorithm-how-your-next-binge-is-a6774fae06c7]

3. Twitch verwendet das RTMP um einen Stream zu broadcasten. Doch um einen Stream anzuschauen wird ebenfalls das HLS Protokoll verwendet. [https://federicogianno.medium.com/streaming-secrets-a-comprehensive-look-at-twitch-tvs-technical-infrastructure-bbe3d5d81736]

Ü 2.9 Protokolle und ihre Anwendungsschicht

Bedeutung von IETF und RFCs

IETF: Die Abkürzung steht für **Internet Engineering Task Force**. Dies ist eine offene Gemeinschaft von Netzwerkdesignern, Betreibern, Anwendungsentwicklern und anderen, die an der Entwicklung und Förderung von Internet-Standards beteiligt sind.

RFC: Die Abkürzung steht für **Request for Comments**. RFCs sind Dokumente, die Standards, Protokolle, Verfahren und Technologien für das Internet beschreiben. Obwohl sie als "Kommentare" bezeichnet werden, sind viele RFCs als offizielle Internetstandards anerkannt.

HTTP-Protokolle

HTTP/0.9:

- **Einführung:** HTTP/0.9 wurde 1991 eingeführt und war die erste Version des Hypertext Transfer Protocols.
- **Eigenschaften:**
  - Extrem einfach; es unterstützt nur GET-Anfragen ohne Header.
  - Überträgt nur HTML-Dokumente.
  - Keine Unterstützung für Verbindungen oder Statuscodes; der Server sendet die HTML-Datei direkt als Antwort.

HTTP/1.0:

- **Einführung:** HTTP/1.0 wurde 1996 veröffentlicht.
- **Eigenschaften:**
  - Führt Header-Informationen ein, die zusätzliche Informationen über die Anfrage oder die Antwort enthalten.
  - Unterstützt unterschiedliche Methoden wie GET, POST und HEAD.
  - Erlaubt das Setzen von Statuscodes (z.B. 404 Not Found).
  - Verbindung wird nach jedem Request geschlossen.

HTTP/1.1:

- **Einführung:** HTTP/1.1 wurde 1999 veröffentlicht und ist eine bedeutende Weiterentwicklung von HTTP/1.0.
- **Eigenschaften:**
  - Persistent Connections: Verbindungen bleiben offen für mehrere Anfragen, was die Leistung verbessert.
  - Chunked Transfer Encoding: Erlaubt das Senden von Daten in Teilstücken.
  - Introduziert neue Methoden (z.B. OPTIONS, PUT, DELETE) und Header (z.B. Host, Range).
  - Verbesserte Fehlerbehandlung.

HTTP/2.0:

- **Einführung:** HTTP/2.0 wurde 2015 eingeführt.
- **Eigenschaften:**
    - Binary Protocol: Wechselt von Text- zu Binärformat, was die Verarbeitung vereinfacht und effizienter macht.
    - Multiplexing: Erlaubt mehrere Anfragen über eine einzige Verbindung gleichzeitig, wodurch die Ladezeiten erheblich verkürzt werden.
    - Server Push: Der Server kann Ressourcen proaktiv an den Client senden, bevor diese angefordert werden.
    - Header Compression: Reduziert die Größe der Header-Daten für schnellere Übertragungen.

HTTP/3.0:

- **Einführung:** HTTP/3.0 ist die neueste Version, die auf dem QUIC-Protokoll basiert und derzeit in Entwicklung ist (Stand 2024).
- **Eigenschaften:**
    - Verwendet UDP anstelle von TCP, was eine schnellere Übertragung und verbesserte Latenz bietet.
    - Integrierte Verschlüsselung, was die Sicherheit erhöht.
    - Bietet die gleichen Multiplexing- und Server-Push-Funktionen wie HTTP/2, aber mit einer noch besseren Handhabung von Verbindungsabbrüchen und Paketverlusten.

E-Mail-Protokolle

SMTP (Simple Mail Transfer Protocol):

- **Beschreibung:** SMTP ist das Standardprotokoll zum Senden von E-Mails über das Internet. Es wird hauptsächlich für den Versand von Nachrichten von einem E-Mail-Client zu einem E-Mail-Server und zwischen E-Mail-Servern verwendet.
- **Verwendung:** SMTP verwendet Port 25 (oder Port 587 für sichere Verbindungen). Es ist ein textbasiertes Protokoll, das Befehle wie HELO, MAIL FROM, RCPT TO und DATA verwendet.
- **Beispiel:**
    1. Ein Benutzer sendet eine E-Mail von einem E-Mail-Client an einen SMTP-Server.
    2. Der SMTP-Server übernimmt die E-Mail und sendet sie an den Zielserver.
    3. Der Zielserver speichert die E-Mail im Postfach des Empfängers.

POP3 (Post Office Protocol Version 3):

- **Beschreibung:** POP3 ist ein Protokoll, das verwendet wird, um E-Mails von einem Server herunterzuladen. Es ist darauf ausgelegt, E-Mails lokal auf dem Client zu speichern.

- **Verwendung:** Nach dem Herunterladen von E-Mails werden sie normalerweise vom Server gelöscht, es sei denn, der Client ist so konfiguriert, dass eine Kopie auf dem Server bleibt.
- **Port:** Standardmäßig Port 110 (oder Port 995 für sichere Verbindungen).

IMAP (Internet Message Access Protocol):

- **Beschreibung:** IMAP ist ein Protokoll, das es Benutzern ermöglicht, E-Mails auf einem Server zu verwalten, ohne sie lokal herunterladen zu müssen. IMAP synchronisiert den E-Mail-Client mit dem Server, sodass Änderungen an einer Nachricht auf einem Gerät auch auf anderen Geräten sichtbar sind.
- **Verwendung:** IMAP ist ideal für Benutzer, die von mehreren Geräten auf ihre E-Mails zugreifen möchten.
- **Port:** Standardmäßig Port 143 (oder Port 993 für sichere Verbindungen).

DNS-Protokoll

DNS (Domain Name System):

- **Beschreibung:** DNS ist ein hierarchisches System zur Namensauflösung, das Domainnamen in IP-Adressen übersetzt. Es ermöglicht die Verwendung von benutzerfreundlichen Namen anstelle von schwer merkbaren IP-Adressen.
- **Funktionsweise:**
    1. **Domain-Name-Anfrage:** Wenn ein Benutzer eine URL in den Browser eingibt, sendet der Client eine DNS-Anfrage an einen DNS-Resolver.
    2. **Auflösung:** Der Resolver fragt den DNS-Server, um die IP-Adresse zu finden, die mit dem Domainnamen verknüpft ist. Wenn der Server die Antwort nicht hat, fragt er andere Server in der DNS-Hierarchie, beginnend mit den Root-Servern.
    3. **Antwort:** Der Resolver erhält die IP-Adresse und sendet sie zurück an den Client, der dann die Verbindung zum gewünschten Server aufbaut.
- **Protokoll:** DNS-Anfragen und -Antworten werden typischerweise über UDP auf Port 53 gesendet. Es unterstützt auch TCP für bestimmte Operationen, wie z.B. das Übertragen von großen Zoneninformationen.

Ü 2.10 Einführung zu Wireshark

Wireshark (www.wireshark.org (http://www.wireshark.org)) ist ein für viele Plattformen verfügbares Tool zur Aufzeichnung von Netzwerkpaketen bzw. für deren weitere Analyse. Installieren Sie sich eine aktuelle Version dieser Software und machen Sie sich mit der Handhabung vertraut. Laden Sie dazu die Datei dump_protocols.pcap aus dem Moodle-Kurs auf Ihren Rechner und öffnen Sie diese Datei mit Wireshark.

- Erklären Sie kurz die Ausgabe des Programms bzw. die Funktionalität der einzelnen Ansichten.
    - Paketliste: Zeigt alle erfassten Pakete mit Infos wie Zeit, Quell-/Zieladresse und Protokoll.
    - Paketdetails: Zeigt Details des ausgewählten Pakets nach Protokollschichten (z.B. Ethernet, IP).
    - Paket-Bytes: Zeigt die Rohdaten des Pakets in Hex- und ASCII-Form.
- Welche Möglichkeiten zur Filterung bietet Ihnen Wireshark? Geben Sie ein Beispiel für eine Filterbedingung an.
    - Wireshark bietet viele verschiedene Möglichkeiten, die Pakete zu filtern. Einige Beispiele sind:
        - UDP, TCP-Pakete (Protokolle)
        - Filter nach IPv4, IPv6 oder MAC-Adressen
        - und vieles mehr.
- Starten Sie selbst eine Aufzeichnung (Capture) Ihres Netzwerkverkehrs. Erklären Sie die Schritte, die dazu notwendig sind.
    - Zuerst muss eines der Interfaces ausgewählt werden. Hier gibt es z.B. Bluetooth, WiFi, Ethernet oder VPN-Verbindungen.
    - Nach Auswahl eines Interfaces durch Doppelklick wird der Traffic erfasst.
- Wozu dient beim Aufzeichnen der Promiscuous Mode?
    - Der promiscuous Mode dient dazu, auch Pakete zu erfassen, die nicht direkt an den Host gerichtet sind.

Ü 2.11 Protokolle mit Wireshark

Analysieren Sie die Datei `dump_protocols.pcap` mit Wireshark und beantworten Sie folgende Fragestellungen:

- Welche Netzwerkprotokolle werden in der Kommunikation verwendet?
    - ARP (Address Resolution Protocol)
    - ICMP (Internet Control Message Protocol)
    - TCP (Transmission Control Protocol)
    - DHCP (Dynamic Host Configuration Protocol)
    - DNS (Domain Name System)
    - HTTP (Hypertext Transfer Protocol)

- Ordnen Sie jedes der Protokolle einer Schicht im TCP/IP-Referenzmodell zu und stellen Sie die Hierarchie der einzelnen Protokolle graphisch dar.
    - Layer 2 (Data Link): ARP
    - Layer 3 (Network): ICMP
    - Layer 4 (Transport): TCP
    - Layer 7 (Application): DHCP, DNS, HTTP

Ü 2.12 Wireshark – HTTP(S)

Laden Sie die Datei `dump_http.pcap` aus dem Moodle-Kurs herunter und analysieren Sie die Datei mit Wireshark. Die Datei beschreibt den Download einer recht trivialen Webseite durch einen HTTP-Client. Beantworten Sie folgende Fragestellungen:

- Welche Objekte werden vom Client via HTTP angefordert?
    - Der Client forder 3 Objekte mithilfe von GET an `\test\`, `\test\logo.gif` und `\test\TechnikErleben.png`
- Recherchieren Sie die Bedeutung der einzelnen Header-Felder bei den Anfragen bzw. Antworten des Servers.

1. ARP (Address Resolution Protocol)
    - HTYPE: Hardware-Typ (z.B. Ethernet).
    - PTYPE: Protokoll-Typ (z.B. IPv4).
    - HLEN/PLEN: Längen der Hardware-/Protokoll-Adressen.
    - Operation: Anfrage (1) oder Antwort (2).
    - SHA/SPA: MAC-/IP-Adresse des Absenders.
    - THA/TPA: MAC-/IP-Adresse des Ziels.
2. ICMP (Internet Control Message Protocol)
    - Type: Nachrichtentyp (z.B. Echo-Anfrage 8).
    - Code: Untertyp der Nachricht.
    - Checksum: Fehlerprüfung.
    - Rest of Header: Variiert je nach Typ (z.B. Identifier bei Echo).
3. TCP (Transmission Control Protocol)
    - Quell-/Zielport: Ports des Absenders und Empfängers.
    - Sequenznummer: Verfolgt die Reihenfolge der Daten.
    - Bestätigungsnummer: Erwartete nächste Sequenznummer.
    - Flags: Steuerflags (z.B. SYN, ACK).
    - Fenstergröße: Empfangsfenstergröße.
    - Checksum: Fehlerprüfung.
4. DHCP (Dynamic Host Configuration Protocol)
    - Op-Code: Anfrage (1) oder Antwort (2).

    - HTYPE/HLEN: Hardwaretyp/Länge der MAC-Adresse.
    - Hops: Anzahl der Relay-Stationen.
    - XID: Transaktions-ID.
    - Client IP: Client-IP (falls bekannt).
    - Your IP: Zugewiesene IP-Adresse.
    - Options: Zusätzliche Konfigurationen.

5. DNS (Domain Name System)

    - Transaktions-ID: Identifiziert die Anfrage.
    - Flags: Steuerbits (z.B. Anfrage oder Antwort).
    - Fragen/Antworten: Abgefragte Domains und deren Antworten.
    - Zusätzliche Felder: Infos wie Nameserver.

6. HTTP (Hypertext Transfer Protocol)

    - Methode (z.B. GET): Art der Anfrage.
    - URL: Angeforderte Ressource.
    - HTTP-Version: Version des Protokolls.
    - Header-Felder: Weitere Infos (z.B. Host, User-Agent).

- Wie viele TCP-Verbindungen werden insgesamt aufgebaut? Wie unterscheidet sich das von dump_protocols.pcap?
    - Es wurden `68` Packets in total versendet, `6` davon `HTTP`
    - Für eine Connection muss `SYN` für die Init des Handshakes geschickt werden welche dann mit `SYN, ACK` bestätigt wird.
    - Bei Wireshark kann man mit `Statistics>Conversations>TCP` die Zustande gekommenen Connections einsehen welche `3` waren
- Bestimmen Sie, wie viele Bytes in jeder Verbindung ausgetauscht werden und wie lange die einzelnen Verbindungen bestehen.
    - Connection 1: Total Bytes `1kb`, (A>B, B>A) = `476b, 760b`, Relative Duration `0.3013`
    - Connection 2: Total Bytes `6kb`, (A>B, B>A) = `616b, 5b`, Relative Duration `0.3012`
    - Connection 3: Total Bytes `30kb`, (A>B, B>A) = `22b, 22b`, Relative Duration `0.5022`

Ü 2.13 Wireshark – HTTP(S)

Verbinden Sie sich mit `https://reference.dashif.org/dash.js/latest/samples/dash-if-reference-player/index.html` und analysieren Sie grob die Vorgänge beim Streamen eines Videos. Beantworten Sie folgende Fragestellungen:

- Welche Objekte werden vom Client via HTTP angefordert? Hinweis: nur jene beim Videostreaming, andere Objekte (z.B. HTML, Text, Bilder) können vernachlässigt werden.
    - Hauptsächlich werden `.m4v`- und `.m4a`-Dateien angefordert.
        - `.m4v`: `GET /akamai/bbb_30fps/bbb_30fps_3840x2160_12000k_31.m4v`