

Netzwerkanalyse mit Wireshark: Was passiert im Netzwerk?

Luis Herzog

April 2023



Abbildung 1: Wireshark Logo [21]

Inhaltsverzeichnis

1	Daten	3
1.1	Metadaten	3
2	Das Programm: Wireshark	4
2.1	Geschichte	4
2.2	Funktionsumfang	4
2.3	Die Benutzeroberfläche	5
2.4	Anwendungsbereiche	5
3	Netzwerk	6
3.1	Aufbau	6
3.2	Protokolle	7
3.3	Open System Interconnection Modell	7
3.3.1	Anwendungsschicht	8
3.3.2	Präsentationsschicht	9
3.3.3	Sitzungsschicht	9
3.3.4	Transportschicht	10
3.3.5	Netzwerkschicht	12
3.3.6	Sicherungsschicht	12
3.3.7	Bitübertragungsschicht	13
4	Analyse	13
4.1	Beispiel	13
4.1.1	Der Webserver	13
4.1.2	Das ‘loopback device‘	15
4.1.3	Ergebnis	15
4.2	Browser	15
4.3	Aufrufen einer Website	15
4.4	Streamen von Videos	15
5	Fazit	15
6	Anlagen	22

Die Digitalisierung ist in aller Munde. Durch die Digitalisierung wächst die Anzahl von Netzwerken, nicht nur in Deutschland, sondern weltweit. Netzwerke werden überall benutzt. Die meisten Menschen besitzen Heimnetzwerke. Aber es gibt noch viel mehr Netzwerke, zum Beispiel in Flughäfen, Cafés, Bibliotheken und noch so viel mehr. Diese Vielfalt von Netzwerken lässt einen Gedanken aufleuchten: „Wie funktionieren Netzwerke?“. Durch die Digitalisierung haben sich Netzwerke stark verändert und vor allem vergrößert. Dadurch sind Netzwerke oft sehr komplex. Die Wireshark Software erlaubt es, einen tieferen Blick in das Netzwerk zu werfen und Details herauszufinden. In dieser Arbeit wird die Arbeitsweise eines Netzwerks dargelegt und die Wireshark Software erklärt.

Bevor man über die Wireshark Software sprechen kann, müssen die fundamentalen Fakten geklärt werden:

1 Daten

Statista beschreibt Daten als „Messwerte, die im Rahmen von Befragungen, Beobachtungen oder Experimenten erhoben werden“[30]. Der Duden gibt mehrere Definitionen. Die dritte Definition gibt Daten als „elektronisch gespeicherte Zeichen, Angaben, Informationen“[11] an. Da es viele verschiedene Arten von Daten gibt, gibt es viele verschiedene Definitionen. Im Netzwerk beschreiben Daten den Inhalt Paketen. Zu diesem Inhalt gehören Informationen zum Paket und die zu vermittelnde Nachricht, welche alle durch elektronisch gespeicherte Zeichen dargestellt werden.

1.1 Metadaten

Hierzu gehören ebenfalls die Metadaten. Diese Daten sind die schon oben angegebenen Information über weitere Daten. Mit diesen Daten lässt sich der Inhalt besser Ordnen, Katalogisieren und Analysieren. Metadaten sind strukturierte Daten und können bei physischen Gegenständen, sowie bei digitalen Dateien Einsatz finden. Ein Beispiel für Metadaten wäre der Autor bei einem Buch, oder der Komponist bei einem Musikstück.[24] In der Netzwerktechnik sind diese Metadaten besonders wichtig. Diese Metadaten sind im Adresskopf der Datenpakete gespeichert und enthalten beispielsweise Informationen über die Quelladresse und Zieladresse des Pakets, sowie über die Lebensdauer des Pakets.[26] Mehr hierzu wird später, unter ‘**Exkurs: UDP/TCP**’ erklärt.

2 Das Programm: Wireshark

Als “Schweizer Taschenmesser“^[31] der Netzwerktechnik wird es von heise.de bezeichnet. Wireshark wird so gut wie überall in der IT-Branche benutzt. Sei es beim Finden von Netzwerkproblemen oder beim Lernen, wie ein Netzwerk funktioniert. Somit bietet das Programm eine Menge Möglichkeiten. Wireshark ist ein “network packet analyzer“^[8]. Es erlaubt dem Benutzer den Netzwerkverkehr aufzunehmen und zu analysieren. Mit Wireshark kann man sozusagen in ein Netzkabel ‘reinschauen‘.

2.1 Geschichte

Wireshark wurde 1997 von Gerald Combs als ‘Ethereal‘ ins Leben gerufen. Das Programm wurde im Juli 1998 als Version 0.2.0 unter einer GPL Lizenz^[14] veröffentlicht.^[7] Somit konnten viele Menschen am Code des Programms mithelfen, um es zu verbessern. Im Jahr 2006 wurde das Projekt in ‘Wireshark‘ umbenannt. Anschließend wurde im Jahr 2008 die Version 1.0 als erste komplette Version veröffentlicht. Sieben Jahre später wurde 2015 die Version 2.0 mit einem komplett neuen Benutzeroberfläche vorgestellt. Letztendlich wurde im Jahr 2023 die Wireshark Foundation¹ gegründet, welche jetzt das Programm stützt. ^[5]

2.2 Funktionsumfang

Wireshark ist eine Software mit sehr vielen Funktionen und bietet fast alles rund um das Thema Netzwerkanalyse. Mit wireshark kann man den Netzwerkverkehr aufnehmen, mit hohem Detail darstellen und einfach exportieren oder speichern. Außerdem kann man exportierte Netzdumps von Wireshark anderen Programmen importieren. Da Wireshark sowohl auf UNIX Plattformen, als auch auf Windows verfügbar ist, geht dies sogar Plattformübergreifend. Ebenso kann man die Datenpakete anhand verschiedener Kriterien filtern und durchsuchen. Die gerade eben genannten Funktionen kratzen aber nur an der Oberfläche. Wireshark bietet noch viel mehr Funktionen.^[9]

¹<https://wiresharkfoundation.org/>

2.3 Die Benutzeroberfläche

Das voreingestellte Layout ist im unteren Screenshot dargestellt. In diesem Fall wird gerade die Netzwerkschnittstelle ‘wg-mullvad’ aufgezeichnet. Im oberen Teil ist die Symbolleiste zu finden. Dort findet man Knöpfe zum starten und stoppen der Aufnahme. Recht daneben befinden sich die Knöpfe zum Laden und Speichern von Netzwerkdumps. Weiter rechts folgen Knöpfe zum Sortieren der Paketliste. Darunter befindet sich die Liste mit den aufgenommenen Paketen. Dort kann direkt die Zeit, Quelle und Ziel des Pakets abgelesen werden. Rechts daneben kann das benutzte Netzwerkprotokoll und die Länge abgelesen werden. Ganz rechts steht weitere Information zum Paket. Unter dieser Liste befinden sich zwei Fenster. Rechts steht der rohe Inhalt des Pakets. Links hingegen steht der von Wireshark automatisch ausgelesene und geordnete Inhalt des Pakets. Hier kann jegliche Information zum Datenpaket gefunden werden. Unter diesem Fenster steht der Name der Quelle, deren Netzwerkverkehr aufgezeichnet wird. Unten ist ein Beispielscreenshot zu sehen.

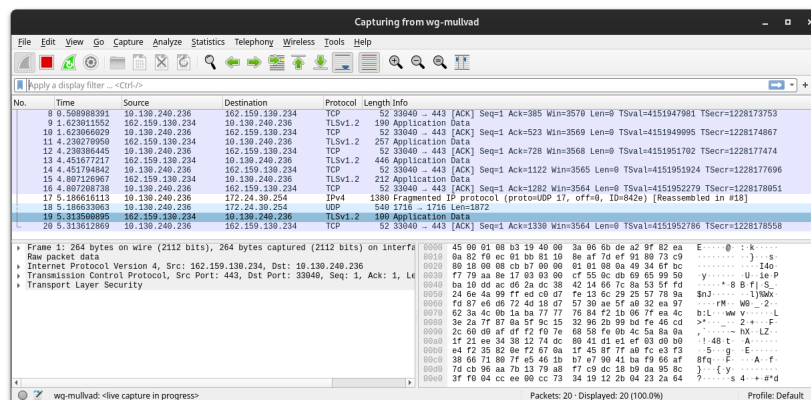


Abbildung 2: Beispielscreenshot aus der Wireshark Software [19]

2.4 Anwendungsbereiche

Wie schon im oberen Teil dargestellt, bietet Wireshark sehr viele Funktionen. Somit kann Wireshark in sehr vielen Situationen Anwendung finden. Es kann beispielsweise von Netzwerk Administratoren zum Finden, Analysieren und Lösen von Netzwerkproblemen benutzt werden. Das Programm kann außerdem von Sicherheitsanalysten eines Netzwerks benutzt werden, um Sicherheitsprobleme

in Netzwerken zu finden. Wiederum kann es auch von Anwendungsentwicklern benutzt werden, um die Umsetzung von Netzwerkprotokollen auszutesten. Zuletzt kann es auch benutzt werden um mehr über das Netzwerk und den Netzwerkverkehr zu lernen und diesen zu analysieren. Es gibt natürlich auch viele weitere Möglichkeiten Wireshark zu benutzen. [10]

3 Netzwerk

In der Informationstechnologie ist ein Netzwerk ein Zusammenschluss mehrerer Computer [...], die untereinander kommunizieren, also Daten untereinander austauschen können.[22] Soweit geht die Definition.

3.1 Aufbau

Es gibt allerdings viele verschiedene Arten von Netzwerken, welche unterschiedlich aufgebaut sind. Diese unterschiedlichen Netzwerke werden in Topologien unterteilt. Durch diese Topologien können selbst sehr komplexe Netzwerke gut veranschaulicht werden. Es gibt fünf Haupt-

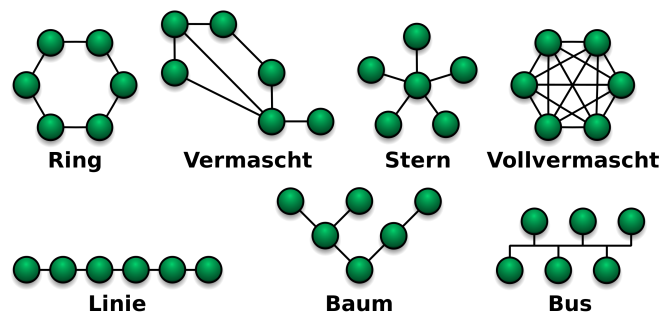


Abbildung 3: Die Netzwerktopologien [13]

topologien. Bei der Ringtopologie sind alle Knoten in einem Ring miteinander verbunden. Die Linien-Topologie ist ähnlich, mit dem Unterschied, dass die Knoten an den Enden nicht mehr direkt miteinander verbunden sind. Bei der Sterntopologie sind alle Knoten mit einem Hauptknoten verbunden. Die Baumtopologie ist eine erweiterte Form der Sterntopologie, bei der mehrere Knoten hierarchisch miteinander verbunden sind. Zuletzt gibt es noch die Bustopologie, bei der alle Knoten an einem Übertragungsmedium hängen. Netzwerke können ebenfalls vermascht sein, was bedeutet, dass die Knoten mehr Verbindungen aufweisen, als theoretisch nötig wären.[23] Der Beispielaufbau unten zeigt ein Netzwerk basierend auf der Sterntopologie.

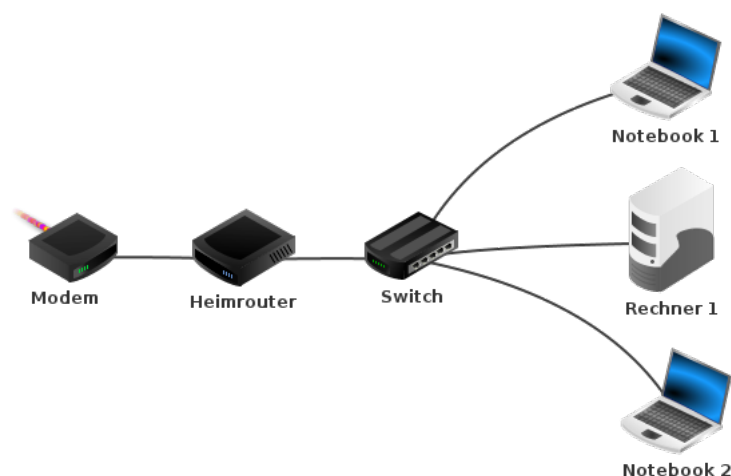


Abbildung 4: Aufbau eines Beispielnetzwerk [15]

3.2 Protokolle

Damit alle Komponenten im Netzwerk richtig miteinander kommunizieren können, wurden standardisierte Protokolle entwickelt, damit alle Komponenten ‘die gleiche Sprache sprechen’. Es gibt verschiedene Protokolle für verschiedene Zwecke. Beispielsweise gibt es HTTP für den Transfer von Webseiten oder FTP zum Senden von Dateien. Da es sehr viele Protokolle gibt und diese recht komplex aufgebaut sind, gibt es das OSI-Modell, um den Netzwerkverkehr mit Protokollen vereinfacht darzustellen. Im nächsten Abschnitt wird dieses Modell genauer beleuchtet.

3.3 Open System Interconnection Modell

Das Open System Interconnect² Modell, auch OSI-Modell genannt, beschreibt die Voraussetzungen, die für eine Kommunikation innerhalb eines Netzwerks nötig sind. Dieses wurde 1983 von durch die ‘Internationale Organisation für Normung’, kurz ‘ISO’ standardisiert. Dies ist notwendig,

²dt: Offenes System für Kommunikationsverbindungen

OSI-Modell		
Host Layers	7	Application
	6	Presentation
	5	Session
	4	Transport
Media Layers	3	Network
	2	Data Link
	1	Physical

damit sich alle Komponenten im Netzwerk, auch wenn diese von verschiedenen Herstellern produziert wurden, reibungslos miteinander funktionieren.

Wenn ein Paket beispielsweise von einem Computer im Netzwerk losgeschickt wird, muss es mehrere Stationen durchlaufen. Das Paket verlässt den Rechner über die Netzwerkkarte und wird durch ein Übertragungsmedium über weitere Netzwerkkomponenten, wie Hubs oder Router bis zur Netzwerkkarte des Zielrechners geleitet. Dort wird dieses dann interpretiert, um korrekt dargestellt zu werden. All diese Schritte werden durch ein Protokoll festgehalten und durch das OSI-Modell spezifiziert, damit jede Station auf diesem Weg weiß, wohin das Paket möchte, woher es kommt und welche Eigenschaften es hat. So wird ein Standard geschaffen, mit dem alle Computersysteme miteinander kommunizieren können.

Da diese Datenkommunikation relativ komplex ist, wurde das Modell in sieben Schichten eingeteilt. Die oberen vier Schichten gehören zu den Anwendungsorientierten Schichten³. Die unteren drei Schichten werden Transport Schichten⁴ genannt. Jede Schicht behandelt eine Anforderung, die für eine funktionierende Kommunikation erfüllt werden muss. Ein zu übertragendes Paket durchläuft vor der Versendung die Schichten 7 - 2, wobei dem Paket bei jeder Schicht Protokoll-Informationen hinzugefügt werden, die dann im Protokoll des Datenpaketes auffindbar sind. Die erste und letzte Schicht wandelt das Paket in technisch übertragbare Daten um und schickt dieses über das Übertragungsmedium weg. Das Übertragungsmedium kann hierbei ein Kabel sein, oder aus einer Antenne bestehen. Auf der Empfängerseite wird dieser Prozess rückwärts durchgeführt. Hierbei wird die jeweilige Protokoll-Information nach der Interpretierung durch die jeweilige Schicht entfernt, bis zum Inhalt des Paketes.

Im folgenden werden die einzelnen Schichten einzeln beleuchtet, um einen besseren Einblick zu gewähren.

3.3.1 Anwendungsschicht

Die Anwendungsschicht⁵ stellt die Daten dar, mit welchen der Nutzer interagiert. Softwareanwendungen, wie Web-Browser und E-Mail clients stützen sich auf die

³engl: Host layer

⁴dt: Media layer

⁵engl: application layer

siebte Schicht, um dem Nutzer aussagekräftige Daten zu präsentieren.

Hierzu gehören Protokolle, wie HTTP⁶, welches benutzt wird, um Websites welche in HTML⁷ geschrieben sind zu präsentieren, oder SMTP⁸, welches benutzt wird um E-Mails zu präsentieren.

3.3.2 Präsentationsschicht

Die Präsentationsschicht⁹ ist in erster Linie dafür verantwortlich, die Daten so aufzubereiten, dass diese in der Anwendungsschicht verwendet werden können. Ein wichtiger Teil dabei ist die Verschlüsselung. Die Präsentationsschicht muss, wenn die Geräte durch eine verschlüsselte Verbindung kommunizieren, auf Senderseite eine Verschlüsselung hinzufügen und diese auf der Empfängerseite korrekt dekodieren.

Ebenso ist die Präsentationsschicht für die Komprimierung der Daten verantwortlich. Dadurch kann die Geschwindigkeit und Effizienz der Kommunikation erhöht und die benötigte Bandbreite minimiert werden.

3.3.3 Sitzungsschicht

Anschließend folgt die Sitzungsschicht¹⁰. Diese ist für das Öffnen und Schließen der Kommunikation der beiden Geräte zuständig. Hier wird die Kommunikation in Sitzungen eingeteilt. Eine Sitzung reicht von der Öffnung bis zur Schließung der Verbindung. Somit wird sicher gestellt, dass die Sitzung lange genug geöffnet bleibt, um alle Daten zu übertragen. Wenn alle Daten erfolgreich übertragen wurden, leitet die Sitzungsschicht die umgehende Schließung der Sitzung ein, um Ressourcen zu sparen.

Eine weitere sehr wichtige Aufgabe der Sitzungsschicht ist die Sicherung der Datenverbindung durch synchronisierte Checkpoints. Wenn beispielsweise bei der Übertragung einer 450 Megabyte großen Datei bei 234 Megabyte die Verbindung unterbrochen wird, kann nach einer Neuverbindung die Übertragung bei 230 Megabyte wieder aufgenommen werden, da es einen Checkpoint der Datei bei 230 Megabyte gibt.

⁶Hyper Text Transfer Protocol

⁷Hyper Text Markup Language

⁸Simple Mail Transfer Protocol

⁹engl: presentation layer

¹⁰engl: session layer

3.3.4 Transportschicht

In der Transportschicht¹¹ werden die Datenpakete vor dem Versenden in Segmente zerlegt. Im Empfangsgerät werden die diese Segmente durch die Transportschicht wieder korrekt zusammengesetzt, sodass diese von der Sitzungsschicht benutzt werden können.

Die Transportebene ist ebenfalls für die Fluss- und Fehlersteuerung zuständig. Hierbei wird die Übertragungsgeschwindigkeit so festgelegt, dass ein ggf. langsamer Empfänger nicht durch die ggf. schnelle Geschwindigkeit des Senders überfordert wird. Beim Empfänger wird durch die Fehlersteuerung ein vollständiger Empfang aller Daten sichergestellt. Wenn die empfangenen Daten nicht vollständig sind, werden diese durch dieses System erneut angefordert, um die Vollständigkeit der Daten zu garantieren.

Hierzu gehören die Protokolle TCP¹² und UDP¹³

Exkurs: UDP/TCP Die zwei Protokolle UDP und TCP sind die meist verwendeten Protokolle, weswegen die Protokolle in den oberen Schichten darauf aufbauen.

TCP ist ein Verbindungsorientiertes Protokoll. Der Client und der Server müssen erst eine Verbindung herstellen, damit Daten in beide Richtungen übertragen werden können. TCP verfügt außerdem über ein erhebliches System für die Fehlerkontrolle. Dadurch ist TCP ein sehr zuverlässiges, aber eher langsames Protokoll. Es wird benutzt, um Dateien, Websites oder Bilder zu übertragen.[2] In der unten dargestellten Tabelle sind die Eigenschaften übersichtlich aufgefächert.

Eigenschaften	TCP
Verbindungsstatus	Erfordert eine bestehende Verbindung zur Datenübertragung
Datensequenzierung	Erfordert Sequenzierung
Garantierte Übertragung	Übertragung wird garantiert
Neuübertragung von verlorenen Paketen	Neuübertragung von verlorenen Paketen ist möglich
Fehlerüberprüfung	Umfangreiche Fehlerüberprüfung und Bestätigen der Daten
Geschwindigkeit	Langsamer als UDP
bestmögliche Verwendung	Streaming bei Videokonferenzen, VoIP oder Livestreams

Tabelle 2: TCP [1]

¹¹engl: transport layer

¹²Transmission Control Protocol

¹³User Datagram Protocol

Der Verbindungsaufbau läuft wie folgt ab: Der Client sendet eine Anfrage zum Verbindungsaufbau¹⁴ zum Server. Wenn dies möglich ist, sendet der Server eine Bestätigung¹⁵ zurück und baut den Kanal auf. Jetzt können der Client und Server miteinander kommunizieren und die Daten können übertragen werden. Wenn ein Datenpaket fehlerhaftes übertragen wird, wird es nochmal geschickt. In der rechten Abbildung ist dieser Datenverkehr in einem Datenflussdiagramm dargestellt.

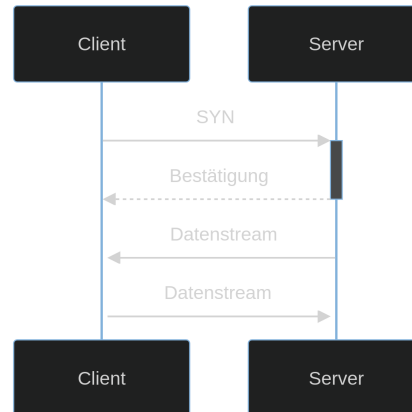


Abbildung 5: Datenflussdiagramm TCP[16]

UDP ist hingegen ein simpleres und verbindungsloses Protokoll. Das System für Fehlerkontrolle ist eher minimal, weswegen die korrekte Übertragung nicht garantiert ist. Die Daten werden kontinuierlich zum Empfänger geschickt, auch wenn dieser diese nicht empfängt. Dies macht UDP nicht ideal für das Verschieken von Dateien, aber um so besser zum streamen. Da das Protokoll wesentlich schneller ist, kann es mit der benötigten Bandbreite arbeiten kann. Eine Fehlerkontrolle ist bei Streams aufgrund der Datenmenge sowieso nicht möglich. Es ist somit Ideal für das Umsetzen von Video-calls, VoIP¹⁶ oder das Livestreamen auf Videoportalen oder im Fernsehen.[3] Die genauen Eigenschaften sind in der unteren Tabelle dargestellt.

Eigenschaften	UDP
Verbindungsstatus	Verbindungslos, keine weiteren Anforderungen
Datensequenzierung	Keine Sequenzierung
Garantierte Übertragung	Keine garantierte Übertragung
Neuübertragung von verlorenen Paketen	Keine Neuübertragung von verlorenen Paketen
Fehlerüberprüfung	Minimale Fehlerberprüfung durch Checksums
Geschwindigkeit	Schneller als TCP
bestmögliche Verwendung	Schicken von Dateien, Email, Bilder, Websites, etc.

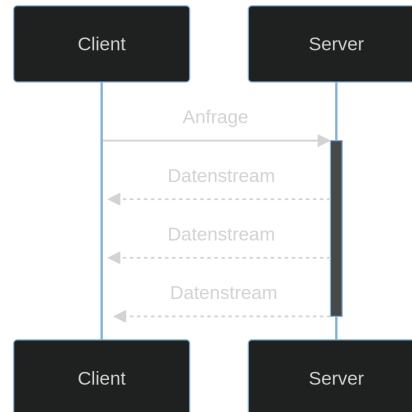
Tabelle 3: UDP [1]

¹⁴SYN-Paket

¹⁵ACK-Paket

¹⁶Voice over IP

Der Verbindungsaufbau läuft wie folgt ab: Der Client schickt dem Server eine Anfrage. Daraufhin antwortet der Server direkt mit dem Datenstream, auch wenn der Client den Datenstream nicht mehr aufnimmt. Der Stream bleibt solange offen, bis der Server den Transfer eigenständig schließt.



3.3.5 Netzwerkschicht

Darauf folgt die Netzwerkschicht¹⁷. Diese gewährt das Kommunizieren zwischen Geräten in verschiedenen, miteinander verbundenen Netzwerken. Beim Versenden, werden die Segmente der Transportschicht erneut in kleinere

Abbildung 6: Datenflussdiagramm UDP[17]

Datenpakete aufgeteilt und mit weiteren Informationen versehen. Diese Informationen sind für die, auf dem Weg gelegenen, Knoten gedacht, um diesen das Ziel des Pakets aufzuweisen. Es wird der beste physikalisch mögliche Weg ausgesucht, um die Daten sicher an ihr Ziel zu bringen. Dieser Prozess wird Routing genannt.[4] Wenn sich beide Geräte im selben Netzwerk befinden, wird diese Ebene übersprungen.

Zu den Protokollen für diese Schicht gehören das IP¹⁸, das ICMP¹⁹, das IGMP²⁰ und die IPsec²¹ Suite. Ein Beispielgerät, welches in dieser Schicht agiert, wäre ein Router.

3.3.6 Sicherungsschicht

Die Sicherungsschicht²² stellt die vorletzte Schicht dar. Diese ist der Netzwerkschicht sehr ähnlich. Der wesentliche Unterschied ist, dass die Sicherungsschicht für die Kommunikation von zwei Geräten innerhalb eines Netzwerks zuständig ist. Die

¹⁷engl: network layer

¹⁸Internet Protocol

¹⁹Internet Control Message Protocol

²⁰Internet Group Message Protocol

²¹Internet Protocol Security

²²engl: Data Link Layer

Sicherungsschicht ist ebenfalls für die Fluss- und Fehlerkontrolle in der netzinternen Kommunikation zuständig.

Beispielgeräte für diese Ebene wären Bridges und Switches.

3.3.7 Bitübertragungsschicht

Die unterste Schicht wird durch die Bitübertragungsschicht²³ dargestellt. Hier sind die Daten als Bitstrom vorhanden, eine Zeichenkette bestehend aus einen und nullen. Hier muss sich auf mehrere Konventionen geeinigt werden. Zu Beginn müssen die Gegebenheiten des Übertragungsmediums festgelegt werden. Dies betrifft die Wahl des Materials und die Funktion der einzelnen Leitungen. Ein Kabel kann beispielsweise aus Kupfer oder Glasfaser bestehen und zwei innere Leitungen haben: eine Datenleitung und eine Steuerleitung. Bei der Übertragung über Funk wird beispielsweise durch Luft übertragen. Es muss ebenfalls die Übertragungsrichtung und Geschwindigkeit festgelegt werden. Ein Kabel kann in eine Richtung²⁴, abwechselnd in beide Richtungen²⁵ oder in beide Richtungen gleichzeitig²⁶ übertragen.

4 Analyse

In der Analyse soll viel benutzte Software getestet werden. Ich habe mich für Web-Browser entschieden, da diese am meisten benutzt werden.[25] Zu Beginn soll die Software getestet werden. Hierfür habe ich einen simplen Test erstellt.

4.1 Beispiel

In diesem Kapitel soll eine Beispielkommunikation vereinfacht dargestellt werden. Es wird ein lokaler Webserver erstellt und anschließend kontaktiert. Dieser Transfer wird dann durch Wireshark aufgezeichnet.

4.1.1 Der Webserver

Code:

```
from flask import Flask #importieren von den benoetigten Bibliotheken
```

²³engl: physical layer

²⁴simplex

²⁵halb-duplex

²⁶duplex

```
import os
from datetime import datetime

hostName = "localhost" #setzen des Hostnamen
serverPort = 8080 #setzen des Ports (Erklaerung unter dem Code)
app = Flask(__name__) #der Flask Server wird instanziiert
@app.route('/') #wenn zum index geroutet wird...
def index():
    return '<p>willkommen in der http webserver demo</p>' #...soll diese
        response zurueckgeschickt werden
@app.route('/time') #wenn zu /time geroutet wird...
def time():
    now = datetime.now()
    current_time = now.strftime("%H:%M:%S") #erstellen der response
    return f'Jetzige Zeit: {current_time}' #...soll diese response
        zurueckgeschickt werden

if __name__ == '__main__':
    app.run(host=hostName, port=serverPort) #starten des Servers
```

Mit der ‘flask‘ Bibliothek können Webserver in Python leicht umgesetzt werden. Sie erlaubt es Webserver zu erstellen, die auf requests hören und mit dem angefragten Inhalt antworten.[12]

Weitere Erklärung zum Code: “Ports sind ein Merkmal der Protokolle TCP und UDP“[27]. Diese werden benutzt, um den Datenverkehr zu ordnen. Da auf einem Server oft mehrere Programme aktiv sind, die gleichzeitig über das Netzwerk kommunizieren müssen diese eingeteilt werden, sodass die Pakete nicht durcheinander kommen. Jedes Programm legt seinen Port fest, bzw. bekommt vom Betriebssystem einen Port zugewiesen. Dieser kann zwischen 0 und 65535 liegen.[29]

Anschließend wird Wireshark geöffnet und das richtige Interface, welches aufgenommen werden soll, ausgewählt. In ‘diesem Fall ist es das ‘Loopback:io‘ interface. (siehe: Anlagen, Das Wireshark interface, auswählen des Loopback Geräts)

4.1.2 Das ‘loopback device‘

Das Loopback interace ist eine “Pseudo-Netzwerkschnittstelle zum gefahrenlosen Testen“[28]. Der Nertzwerktraffic wird in diesem Fall, da der Server auf ‘localhost‘live ist. Das loopback interface agiert als virtuelle Netzwerkkarte und ist nicht wirklich als Hardware vorhanden. Da ich diesen Test auf einem Linux Betriebssystem durchführe ist diese Funktion durch Wireshark unterstützt.[6]

4.1.3 Ergebnis

Nun muss der Server kontaktiert werden. Dies kann man mit einem Web-Browser machen. Mit Firefox habe ich dann ‘http://localhost:8080‘ kontaktiert. In Wire-shark sind direkt mehrere neue Einträge zu erkennen. **(siehe: Anlagen, der Datentransfer wurde aufgezeichnet)**

Im nächsten Schritt wird das erste request, mit dem ‘HTTP‘ Protocoll angesehen. **(siehe: Anlagen, Analyse des ‘GET‘ requests)**. Unten rechts ist der Inhalt des Pakets zu erkennen. Darin sind einige Daten zum Browser enthalten und zum Inhalt, welcher angefragt wird. Diese Daten werden später in der Analyse weiter durchleuchtet.

Der Server antwortet mit diesem angeforderten Inhalt **(siehe: Anlagen, Analyse der response)**. Der Inhalt ist, der oben im Code festgelegte Satz, “Willkommen in der http Webserver demo“

4.2 Browser

4.3 Aufrufen einer Website

4.4 Streamen von Videos

5 Fazit

Dieses Thema hat mir persönlich sehr viel gezeigt. Mir war vorher gar nicht klar, wie viel wirklich im Netzwerk vor sich geht. Sobald man die Wireshark Software öffnet, wird man direkt mit mehreren Requests überschwemmt. Ich finde es trotz dessen bemerkenswert, wie der Datenverkehr durch die ganzen Schritte, optimiert und effizienter gestaltet wird. Jedes einzelne Konzept ist sehr durchdacht und für einen bestimmten Zweck optimiert. Ich finde es trotz dessen die

Frage ‘Was passiert im Netzwerk?’ zu beantworten. Wie schon oben festgestellt gibt es viele verschiedene Arten von Netzwerken und deren Verwendungszweck. Jedes Netzwerk bietet somit unterschiedliche Voraussetzungen, die zu verschiedenen Ergebnissen führen. Die Frage, was wirklich übertragen wird spielt in diesem Sinne ebenfalls eine große Rolle. Bevor ich zu diesem Thema recherchiert habe, war ich den Metadaten negativ gegenüber gestanden. Ich habe diese immer als Tracker, zum Sammeln von personenbezogenen Daten angesehen. Jetzt weiß ich mehr. Mir ist klar geworden, dass der Datenverkehr ohne diese Metadaten nicht funktioniert, da diese zur Steuerung des Datenverkehrs nötig sind. Genauso, wie Spongebob nicht ohne Wasser funktioniert.



Abbildung 7: Spongebob funktioniert nicht ohne Wasser[20]

Abbildungsverzeichnis

1	Wireshark Logo [21]	1
2	Beispielscreenshot aus der Wireshark Software [19]	5
3	Die Netzwerktopologien [13]	6
4	Aufbau eines Beispielnetzwerk [15]	7
5	Datenflussdiagramm TCP[16]	11
6	Datenflussdiagramm UDP[17]	12
7	Spongebob funktioniert nicht ohne Wasser[20]	16
8	Das Wireshark interface, auswählen des Loopback Geräts [19] . .	23
9	Das Wireshark interface, warten auf Datentransfer [19]	24
10	Das Wireshark interface, der Datentransfer wurde aufgezeichnet [19]	24
11	Das Wireshark interface, Analyse des ‘GET‘ requests [19]	25
12	Das Wireshark interface, Analyse der response [19]	25

Tabellenverzeichnis

1	OSI-Modell [18]	7
2	TCP [1]	10
3	UDP [1]	11

Literatur

- [1] Pauline Ashenden. TPC vs. UDP: What's the Difference? Zuletzt besucht: 4. November 2023, Tabelle inspiriert, Quelle: <https://www.livesize.com/blog/tcp-vs-udp/>.
- [2] Pauline Ashenden. What ist TCP? Zuletzt besucht: 3. November 2023 Quelle: <https://www.livesize.com/blog/tcp-vs-udp/>.
- [3] Pauline Ashenden. What ist UDP? Zuletzt besucht: 3. November 2023 Quelle: <https://www.livesize.com/blog/tcp-vs-udp/>.
- [4] Cloudflare.de. Was ist Routing? — IP-Routing. Zuletzt besucht: 3. November 2023, Quelle: <https://www.cloudflare.com/de-de/learning/network-layer/what-is-routing/>.
- [5] The Wireshark Contributors. A Brief History of Wireshark. Zuletzt besucht: 2. November 2023, Quelle: https://www.wireshark.org/docs/wsug_html_chunked/ChIntroHistory.html.
- [6] The Wireshark Contributors. Loopback. Zuletzt besucht: 1. November 2023, Quelle: <https://wiki.wireshark.org/CaptureSetup/Loopback#supported-platforms>.
- [7] The Wireshark Contributors. Open Source Software. Zuletzt besucht: 2. November 2023, Quelle: https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html#open-source-software.
- [8] The Wireshark Contributors. What is Wireshark? Zuletzt besucht: 2. November 2023, Quelle: https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html#ChIntroWhatIs.
- [9] The Wireshark Contributors. Wireshark - Features. Zuletzt besucht: 15. Juni 2023, Quelle: https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html#ChIntroFeatures.
- [10] The Wireshark Contributors. Wireshark - Intended purposes. Zuletzt besucht: 16. Mai 2023, Quelle: https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html#ChIntroPurposes.

- [11] Duden.de. Daten. Zuletzt besucht: 3. November 2023, Quelle: <https://www.duden.de/rechtschreibung/Daten>.
- [12] The flask Team. Flask 3.0.0. Zuletzt besucht: 2. November 2023, Quelle: <https://pypi.org/project/Flask/>.
- [13] Predatorix Foobaz, Parzi. Netzwerktopologien. Zuletzt besucht: 3. November 2023, Quelle: <https://commons.wikimedia.org/wiki/File:NetzwerkTopologien.svg>.
- [14] The gnu Foundation. Gnu General Public License. Zuletzt besucht: 2. November 2023, Quelle: <https://www.gnu.org/licenses/old-licenses/gpl-2.0.html>.
- [15] Luis Herzog. Abbildung eines Beispielnetzwerks. Erstellt mit Filius.
- [16] Luis Herzog. Datenflussdiagramm TCP. Zuletzt besucht: 3. November 2023, Nachempfunden von: Abbildung 1 von <https://www.geeksforgeeks.org/differences-between-tcp-and-udp/> Erstellt mit revealjs.
- [17] Luis Herzog. Datenflussdiagramm UDP. Zuletzt besucht: 3. November 2023, Nachempfunden von: Abbildung 2 von <https://www.geeksforgeeks.org/differences-between-tcp-and-udp/> Erstellt mit revealjs.
- [18] Luis Herzog. Osi-Modell. Zuletzt besucht: 3. November 2023, Tabelle inspiriert, Quelle: https://en.wikipedia.org/wiki/OSI_protocols.
- [19] Luis Herzog. Screenshots aus der Wireshark Software. Selbst erstellt <https://luisherzog.de>.
- [20] Stephen Hillenburg. ‘I don’t need it’. Zuletzt besucht: 4. November 2023, Coverbild von <https://tuna.voicemod.net/sound/d745ae33-f574-44fa-90fb-6627489f90c4>.
- [21] Ktdryer. Wireshark Logo. Zuletzt besucht: 3. November 2023, Quelle: https://de.wikipedia.org/wiki/Wireshark#/media/Datei:Wireshark_icon.svg.
- [22] kurthelectronic.de. Was ist ein Netzwerk? Zuletzt besucht: 3. November 2023, Quelle: <https://www.kurthelectronic.de/fachwissen/wie-funktioniert-ein-it-netzwerk/>.

- [23] kurthelectronic.de. Welche Topologien können zum Verbinden genutzt werden? Zuletzt besucht: 3. November 2023, Quelle: <https://www.kurthelectronic.de/fachwissen/wie-funktioniert-ein-it-netzwerk/>.
- [24] Dipl. Ing. (FH) Stefan Lubert. Was sind Metadaten? Zuletzt besucht: 3. November 2023, Quelle: <https://www.security-insider.de/was-sind-metadaten-a-358dc75ad6978ece7b58d89b045f5ec6/>.
- [25] Joerg Geiger Michael Humpal. Ziemlich sicher auf Ihrem System: Die beliebtesten Programme der Welt. Zuletzt besucht: 3. November 2023, Quelle: https://www.chip.de/news/Ziemlich-sicher-auch-auf-Ihrem-System-Die-beliebtesten-Programme-der-Welt_111539116.html.
- [26] Netzwerke.com. TCP/IP – Transmission Control Protocol / Internet Protocol. Zuletzt besucht: 3. November 2023, Quelle: <https://www.netzwerke.com/TCP-IP.htm>.
- [27] Reinhard Schiedermeier. Fußnote [19]: Ports. Zuletzt besucht: 2. November 2023, Quelle: <https://sol.cs.hm.edu/4129/html/356-ports.xhtml>.
- [28] Reinhard Schiedermeier. Loopback-Device. Zuletzt besucht: 1. November 2023, Quelle: <https://sol.cs.hm.edu/4129/html/353-loopbackdevice.xhtml>.
- [29] Reinhard Schiedermeier. Ports. Zuletzt besucht: 2. November 2023, Quelle: <https://sol.cs.hm.edu/4129/html/356-ports.xhtml>.
- [30] Statista.de. Definition Daten. Zuletzt besucht: 3. November 2023, Quelle: <https://de.statista.com/statistik/lexikon/definition/42/daten/>.
- [31] Johannes Weber. Mit Wireshark die Last von NTP - Servern messen. Zuletzt besucht: 31. Oktober 2023, Quelle: <https://www.heise.de/hintergrund/Mit-Wireshark-die-Last-von-NTP-Servern-messen-4416283.html>.

Ich erkläre hiermit, dass ich die Seminararbeit ohne fremde Hilfe angefertigt und nur die im Literaturverzeichnis angeführten Quellen und Hilfsmittel benutzt habe

6 Anlagen

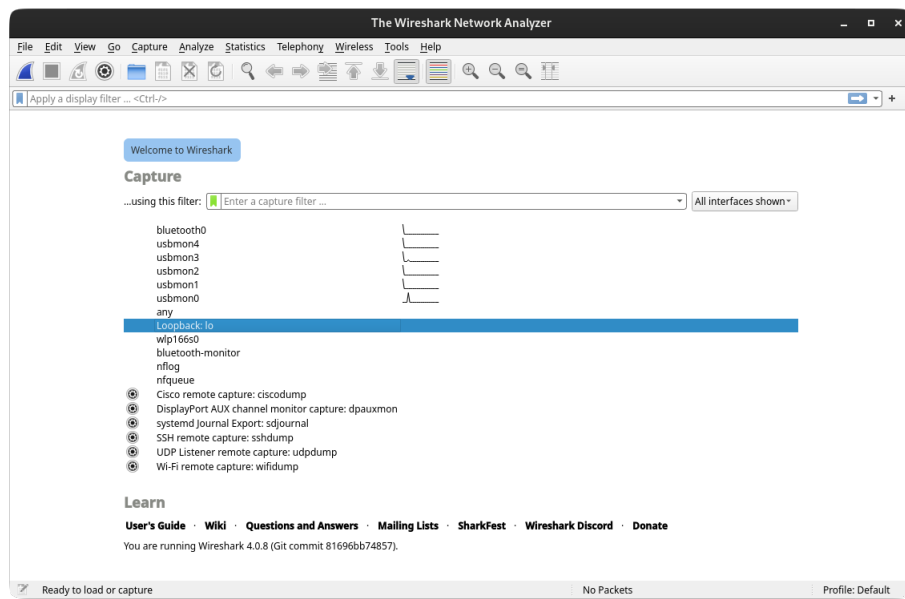


Abbildung 8: Das Wireshark interface, auswählen des Loopback Geräts [19]

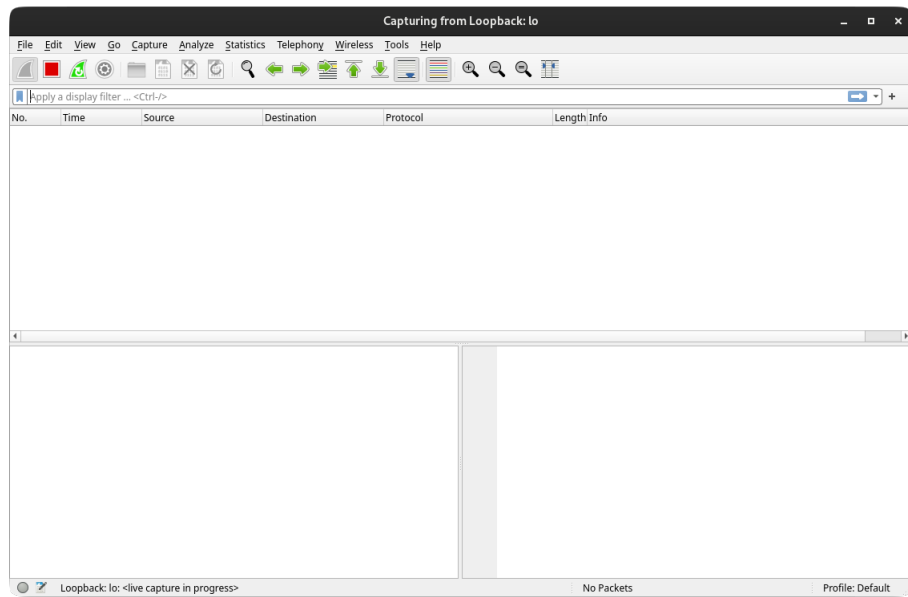


Abbildung 9: Das Wireshark interface, warten auf Datentransfer [19]

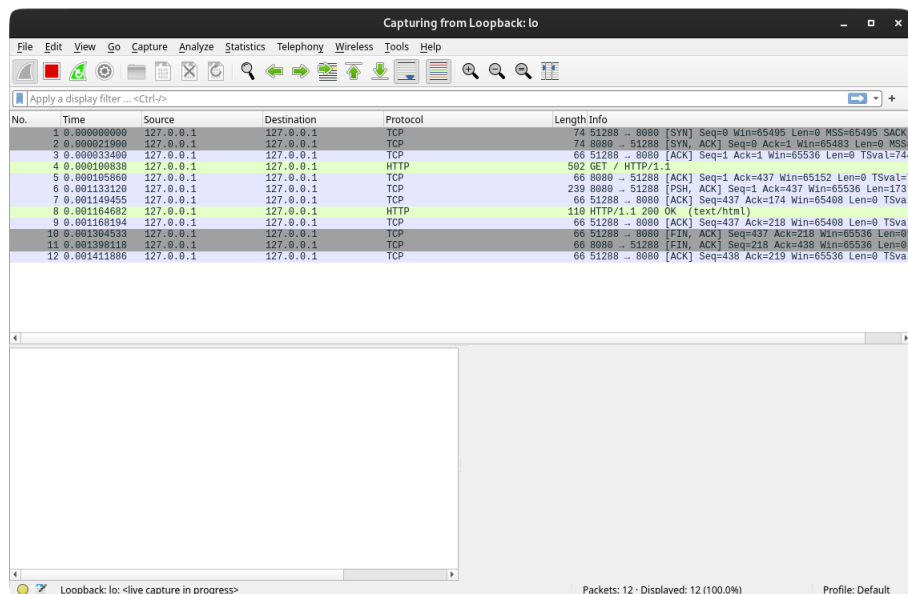


Abbildung 10: Das Wireshark interface, der Datentransfer wurde aufgezeichnet [19]

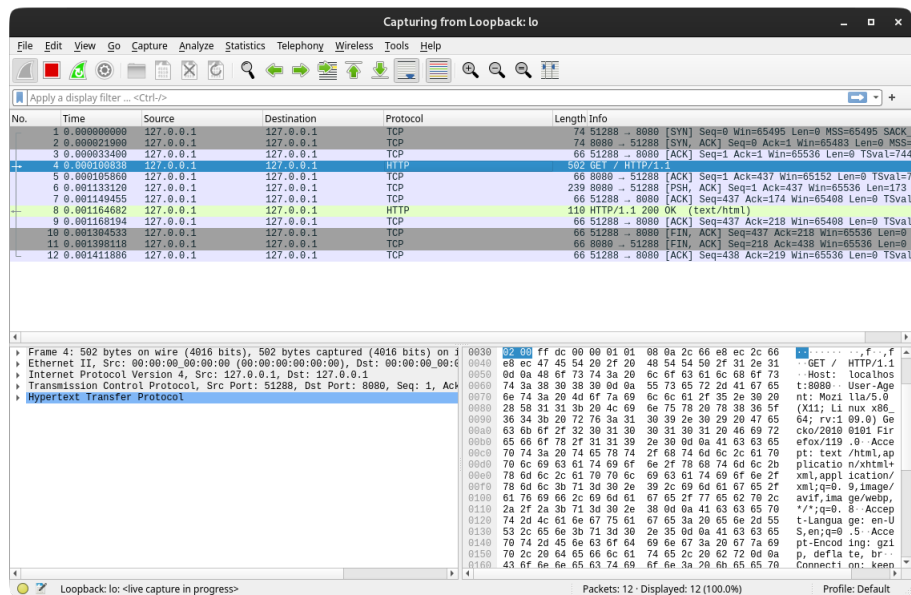


Abbildung 11: Das Wireshark interface, Analyse des ‘GET’ requests [19]

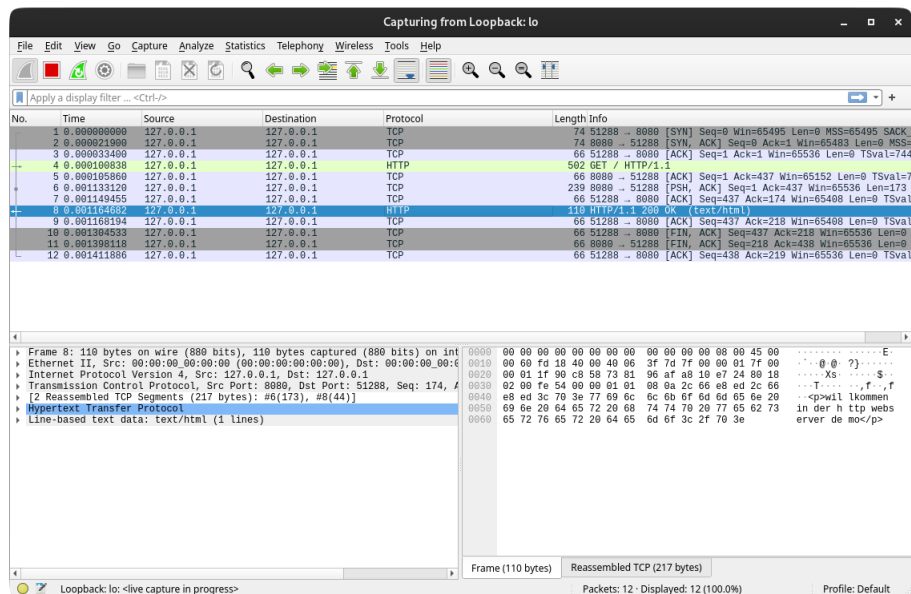


Abbildung 12: Das Wireshark interface, Analyse der response [19]