# Wireshark

Dr.sc.nat. Michael J.M. Wagner*

Revision 282

## Inhaltsverzeichnis



*michael@wagnertech.de

# 1 Einführung

> Wireshark (englisch *wire* „Draht", „Kabel" und *shark* „Hai") ist eine freie Software zur Analyse und grafischen Aufbereitung von Datenprotokollen (Sniffer), die 2006 als Fork des Programms *Ethereal* (...) entstanden ist. Solche Datenprotokolle verwenden Computer auf verschiedensten Kommunikationsmedien wie dem lokalen Netzwerk, Bluetooth oder USB. Das Netzwerk-Analyse-Tool kann Administratoren, Netzwerk-Experten und Sicherheits-Experten bei der Suche nach Netzwerkproblemen, der Ermittlung von Botnet-Verbindungen oder beim Netzwerk-Management behilflich sein.[1]

# 2 Internet

Für die „Erfindung" des Internets werden zwei Ereignisse gehandelt:

- 1968: Aufbau des *ARPANET* durch das US-Amerikanische Verteidigungsministerium
- 1991: Einführung des HTTP-Protokolls durch Roy Fielding, Tim Berners-Lee und andere am CERN[2].

Das *ARPANET* war ein Computer-Netzwerk, ursprünglich im Auftrag der US-Luftwaffe ab 1968 entwickelt. Es ist der Vorläufer des heutigen Internets. Kennzeichen waren die teilvermaschte Netztopologie und die paketvermittelten Netze[3]

Das *hypertext transfer protocol* (HTTP) ist ein Internet-Protokoll zum Abruf von Texten, die ihrerseits Verweise auf andere Texte haben können (*Hypertexte*).

Das Internet ist das „Netz der Netze", also ein Zusammenschluss verschiedener Netze. Man unterscheidet dabei private und öffentliche Netze. Die Netzelemente der öffentlichen Netze haben Weltweit eindeutige Adressen. Als Adressen dienen im Internet Vierertupel von Zahlen zwischen 0 und 254 (IPv4-Adresse). Da sich diese Zahlen schlecht merken lassen, gibt es parallel ein System, das Domainnamen in solche Zahlentupel umsetzt. Dieser Dienst nennt sich *Domain Name Service* (DNS).

Die Netzelemente privater Netze können vom Internet aus nicht adressiert werden, da sie keine eindeutige Adresse haben. Netzelemente desselben Netzes teilen sich die ersten Ziffern ihrer Adresse. Damit das *routing* (die Suche des Datenwegs zum gewünschneten Netzelement) weiß, ob sich dieses Element im eigenen Netz oder in einem anderen Netz befindet, muss bekannt sein, wieviele führende Ziffern das Netz bestimmen. Die Anzahl der führenden Ziffern wird durch die Netzmaske (*netmask*) bestimmt. Daten, die für ein anderes Netz bestimmt sind, werden dem Router zur Weiterleitung übergeben. Diese Informationen kann man sich in den „Verbindungsinformationen" anschaun (s. Abb. 1). Dabei bedeuten:

- IP-Adresse: Adresse des Netzelements
- Subnetz-Maske: Die ersten drei Ziffern des Tupels bestimmen das Netz.

[1] https://de.wikipedia.org/wiki/Wireshark
[2] Wikipedia: HTTP (7.7.2017)
[3] Wikipedia: ARPANET (7.7.2017)

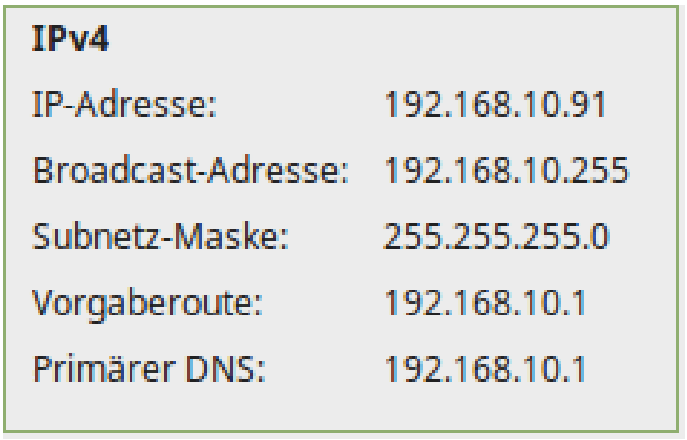


Abbildung 1: Verbindungsinformationen

- Vorgaberoute: Die Adresse des Routers.
- Primärer DNS: DNS-Server, der für die Auflösung von Domainnamen angefragt wird.

Wie in der Abbildung ersichtlich, handelt es sich um IP**v4**-Adressen. Seit Jahrzehnten gibt es einen weiteren Standard: IPv6. IPv6-Adressen sind deutlich länger (128 bit, 16 Byte). Daher ist es möglich, jedem Gerät weltweit eine eindeuige Adresse zu geben. IPv6-Adressen haben etwa folgendes Aussehen: `2001:4ca0:4f04:f096::8d54:95e0`. IPv6 wird in diesem Kurs nicht weiter besprochen.

Abbildung 2 zeigt ein schematisches Netzwerkbild. Es handelt sich hierbei um Netze mit einer sternförmigen Topologie. Netzwerke, die über Ethernet verbunden sind, haben typischerweise diese Topologie. Router bilden die Netzübergänge. Diese gehören zu beiden (mehreren) Netzen.

Zur Unterscheidung öffentlicher und privater Netze haben sich auch die Begriffe *worldwide area net* (WAN) und *local area net* (LAN) eingebürgert. Ist ein Rechner im LAN nicht mit einem Netzwerkkabel, sondern über eine Funkstrecke mit dem Netz verbunden, so spricht man von einem *wireless local area net* (WLAN). Das mit dem LAN verbundene Ende der Funkstrecke nennt sich *access point* (AP). Ist ein AP mit einem Router in ein Gerät integriert, sprich man von einem WLAN-Router.

Im Folgenden werden Netzwerkdieste beschrieben, die typischerweise nicht nach innen für das LAN wirken, sondern nach außen in das Internet. Sie werden daher meist nicht aus dem privaten Netz heraus angeboten, sondern von bereits in einem öffentlichen Netz befindlichen Server.

## 2.1 Webserver

Webserver gibt es in verschiedenen Ausprägungen:

- HTTP-Server
  - beantwortet HTTP-Anfragen mit statischen Dateien aus dem Dateisystem
  - Beispiel: Apache
- Servlet-Engine

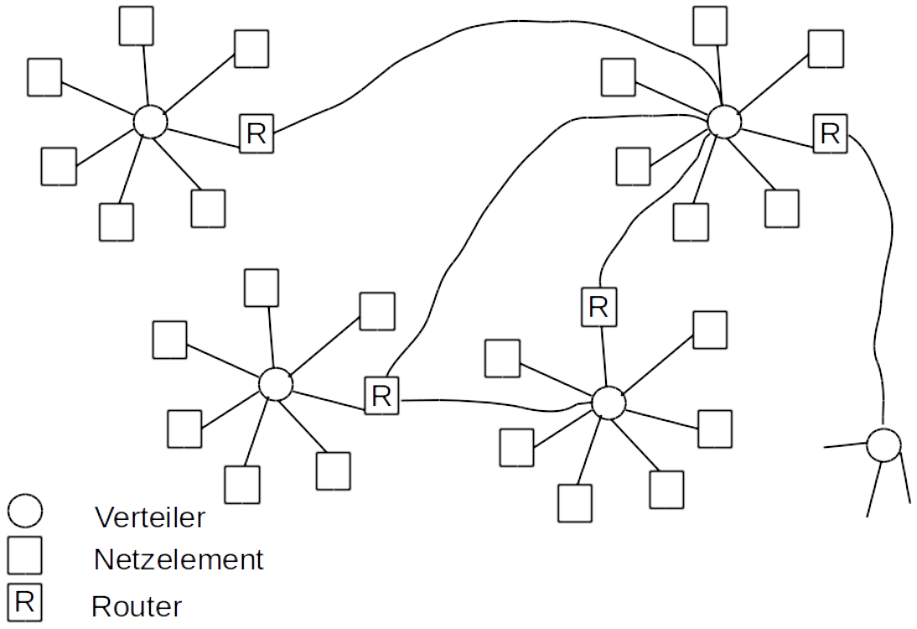


Abbildung 2: Sternförmige, über Router verbundene Computernetze

  - beantwortet HTTP-Anfragen durch Delegation an Programmcode
  - Apache + PHP-Plugin
  - Tomcat (Java Servlet Engine)
- Application Server
  - Servlet-Engine + EAI-Middleware
  - Geronimo (ASF)
  - JBoss (JBoss)
  - WebSphere (IBM)

Während man in den 90er Jahren häufig statische HTML-Seiten antraf, für die ein schlichter HTTP-Server genügte, haben sich heute *content management systems* (CMS) etabliert. Diese ermöglichen die Pflege der Seiteninhalte über eigene Internetseiten, die zur Adminstration dienen. Die CMS benötigen entweder *servlet engines* oder *application server*.

Statische Seiten eignen sich für den Unterricht, um die Funktionsweise von HTTP zu demonstrieren. Sollen im Unterricht nicht nur einzelne alleinstehende Seiten, sondern ein echter *hypertext* entstehen, so müssen die Seiten auf einen Webserver gebracht werden.

## 2.2 Mailserver

Ein Server, der für den Empfang und das Versenden von E-Mails arbeitet, muss stets vom Internet aus erreichbar sein. Grundsätzlich empfiehlt es sich aus Datenschutzgründen nicht-private Mails nicht bei den großen Mail-Anbietern (GMX, t-online, ...) zu betreiben.

Für die Verwendung von E-Mails ist ein Mail-Programm nötig. Dieses kann entweder lokal auf dem Rechner des Anwenders laufen (Outlook, Thunderbird, ...) oder als Web-Applikation auf dem Mailserver selbst. In letzterem Fall wird das Mailprogramm über den Browser (HTTP) bedient. Letzeres Verfahren hat den Nachteil, dass keine Ende-zu-Ende-Verschlüsselung verwendet werden kann.

Beim Austausch von E-Mails sind zwei Protokolle von Bedeutung:

- *Simple Mail Transfer Protocol* (SMTP): Dieses Protokoll wird zum Versenden von E-Mails verwendet, gegebenenfalls in zwei Schritten:
  - Wird ein lokaler Mail-Client verwendet, so schickt dieser die Mail an den eigenen Mailserver.

    Grund: Der fremde Mailserver akzeptiert meist keine Mails, die keinen „ordentlichen" Absender haben (Spam).
  - Der eigene Mailserver schickt die Mail an den fremden Mailserver, der die Mail in das Postfach des Nutzers legt.
- *Internet Message Access Protocol* (IMAP): Über dieses Protokoll holen lokale Mailprogramme die Mails vom Server und können sie auf dem lokalen Rechner speichern.

# 3 TCP/IP Protokollstapel

## 3.1 OSI-Modell

IT-Protokolle lassen sich mit Hilfe des OSI-Modells veranschaulichen. Das OSI-Modell umfasst sieben Schichten:[4]

- 1. Schicht / Bitübertragung: Umwandlung der Bits in ein zum Medium passendes Signal und physikalische Übertragung.
- 2. Schicht / Sicherung: Segmentierung der Pakete in Frames und Hinzufügen von Prüfsummen.
- 3. Schicht / Vermittlung: Routing der Datenpakete zum nächsten Knoten.
- 4. Schicht / Transport: Zuordnung der Datenpakete zu einer Anwendung.
- 5. Schicht / Sitzung: Steuerung der Verbindungen und des Datenaustauschs.
- 6. Schicht / Darstellung: Umwandlung der systemabhängigen Daten in ein unabhängiges Format.
- 7. Schicht / Anwendung: Funktionen für Anwendungen sowie die Dateneingabe und -ausgabe.

[4] https://de.wikipedia.org/wiki/OSI-Modell (12.4.2021)
[5] [TCP/IP] S. 11.

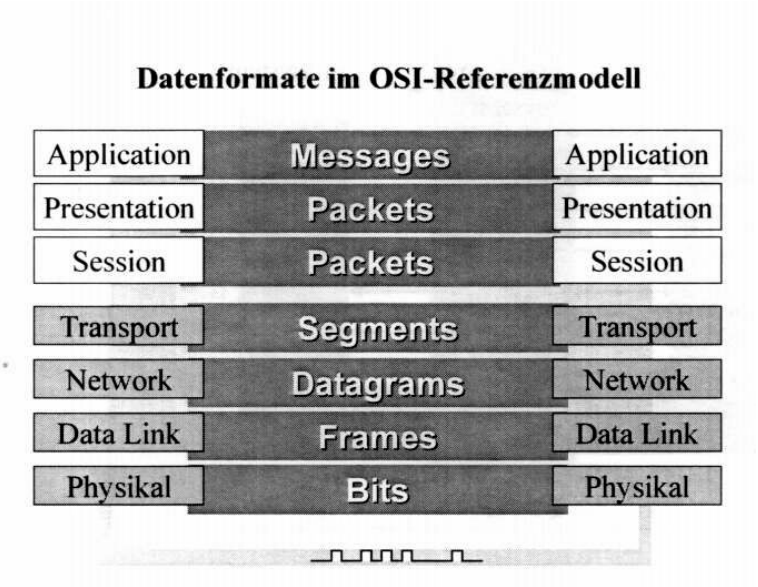


Abbildung 3: OSI-Referenzmodell[5]

## 3.2 TCP/IP-Modell

Die Schichten des OSI-Modells werden wie folgt auf das TCP/IP-Modell abgebildet:[6]

| OSI-Schicht | TCP/IP-Schicht |
|---|---|
| 7. Anwendung<br>6. Darstellung<br>5. Sitzung | Anwendung |
| 4. Transport<br>3. Vermittlung<br>2. Sicherung<br>1. Bitübertragung | Transport (TCP, ...)<br>Internet (IP)<br>Ethernet (MAC)<br>Netzwerkkabel |

Die Darstellung in Wireshark orientiert sich an dieser Schichtung:

- Frame: gesamtes Datenpaket
- Ethernet
- Internet Protocol
- Transmission Control Protocol, ... (hier können verschiedene Protokolle aufsetzen)
- Name der Anwendung (falls vorhanden)

Aufgabe:

- Starten Sie Wireshark.
- Laden Sie die Datei `Programmieren_3.dump`.

[6] https://de.wikipedia.org/wiki/OSI-Modell (12.4.2021)

- Betrachten Sie das erste Paket (TCP/ssh).
- Betrachten Sie Paket 11 (UDP/KNXnet).

# 4 Netzwerkdienste

In diesem Kapitel werden nach und nach die verschiedenen Netzwerkdienste vorgestellt, die zum Funktionieren des Internets beitragen. Zur Illustration der jeweiligen Dienste werden diese dann anhand einer kleinen Übung mit der Lernsoftware Filius[7] behandelt.

Für die Übungen mit Filius ist ein „Schulnetz" vorbereitet:

Aufgabe:

- Starten Sie *filius* und laden Sie das vorbereitete Schulnetz (`Schulnetz.fls`[8])
- Starten Sie die Simulation.
- Verbinden Sie sich mit dem Rechner `172.16.10.0` (einfach draufklicken) und starten Sie darin die Konsole.
- Öffnen Sie das Protokollfenster des Rechners (rechter Mausklick „Datenaustausch anzeigen").

## 4.1 Routing

Das *routing* ist eines der zentralen Netzwerkdienste. Jedes Netz ist durch den führenden Teil der IP-Adresse identifiziert. Jeder Rechner entscheidet anhand der Adresse, ob für ein IP-Paket das Ziel im eigenen Netz zu suchen ist, oder an einen Router übergeben wird. Jeder Rechner/Router hat eine Standardroute (*default route*), an die er alle Pakete weitergibt, deren Netze er nicht kennt. Auf diese Weise besteht die Hoffnung, dass jedes Paket irgendwann sein Ziel erreicht. *IP routing* ist nicht deterministisch!

Wir müssen also zwei Dinge unterscheiden: Den Knoten, an den ein Datenpaket erst mal gegeben wird, u.U. mit der Bitte der Weiterleitung und den Rechner, den das Paket final erreichen soll. Daher werden sog. OSI-Modell[9] verschiedene Protokollschichten unterschieden. Die Schicht, die die Kommunikation von Paketen mittels IP-Adressen beschreibt, ist Schicht 3 (Vermittlung). Die direkte Punkt-zu-Punkt-Kommunikation ist in Ebene 2 (Netzzugang) beschrieben.

In OSI-Ebene 2 werden die Rechner nicht mit ihren IP-Adressen, sondern mit den MAC-Adressen adressiert. Oftmals hat ein Rechner mehrere Netzwerkschnittstellen zur Verfügung, die zu verschiedenen Netzen gehören (z.B. Netzwerkkabel und WLAN). Die Zustellung von Datenpaketen läuft daher nach folgendem Muster ab:

[7] `http://lernsoftware-filius.de` (20.01.2021)
[8] Verfügbar unter `http://wagnertech.de/public/KMBD/`
[9] `https://de.wikipedia.org/wiki/OSI-Modell` (21.01.2021)

- Prüfe die Ziel-IP-Adresse: Liegt sie in einem meiner Netze?
  - Wenn Ja, frage in diesem Netz nach der MAC-Adresse dieses Geräts.
    - Wenn niemand antwortet, beende mit einer Fehlermeldung
  - Sende das Paket an den Rechner mit der ermittelten MAC-Adresse
- Ermittle die Defaultroute.
- Prüfe, zu welchem meiner Netze die Defaultroute gehört.
- Frage in dem Netz nach der MAC-Adresse der Defaultroute.
  - Wenn niemand antwortet, beende mit einer Fehlermeldung
- Sende das Paket an die MAC-Adresse der Defaultroute.
- Die Defaultroute wird das Paket übernehmen. Im Paket steht die IP-Adresse des eigentlichen Ziels. Jetzt beginnt im Router das Spiel von neuem.

Das Ebene-2-Protokoll zur Ermittlung der MAC-Adresse nennt sich *address resolution protocol* (ARP).

Zur Überprüfung der eigenen IP- und MAC-Adressen stehen folgende Befehle zur Verfügung:

```
ipconfig /all      (Windows)
ip a               (Linux)
ipconfig           (filius)
```

Um ein Paket an einen anderen Rechner zu schicken, gibt es den Befehl `ping <ziel>`. Für die Verfolgung der Route stehen folgende Befehle zur Verfügung:

```
tracert <ziel>       (Windows)
traceroute <ziel>    (Linux)
traceroute <ziel>    (filius)
```

Aufgabe:

- Überprüfen Sie die eigene IP- und MAC-Adresse.
- Pingen Sie einen Rechner im eigenen Netz, sowie einen Rechner in einem anderen Netz an. Im eigenen Netz können Sie stets Ihre Defaultroute anpingen.
- Betrachten Sie die Route zu diesen Rechnern.
- Führen Sie die genannten Operationen auch auf dem Client `172.16.10.1`.
- Betrachten Sie den Datenaustausch.

## 4.2 Domain Name Service (DNS)[10]

Das TCP/IP-Protokoll verwendet IP-Adressen um Rechner im Netz zu adressieren. Da diese für den menschlichen Nutzer wenig aussagekräftig sind, zudem sich immer wieder mal ändern, stehen alternativ dazu Rechnernamen zur Verfügung. Die Umsetzung dieser Rechnernamen in IP-Adressen leistet der Domain Name Service (DNS).

DNS ist eine verteilte Datenbank. Es besteht aus zwei wichtigen Elementen: Zum einen den Namenservern, die die DNS-Server-Software ausführen und auf Anfrage Namensinformationen herausgeben, zum anderen aus dem hierarchischen System der Domain-Namen selbst.

Ein vollständiger Domain-Name (*fully qualified domain-name* (FQDN) ist dabei von rechts nach links zu interpretieren. Ganz rechts steht die *top level domain* (TLD). Diese werden der *Internet Assigned Numbers Authority* (IANA), einer Unterorganisation der *Internet Corporation for Assigned Names and Numbers* (ICANN) verwaltet. Die wichtigsten TLD sind die US-amerikanischen `.com`, `.net`, etc. und die Länderkennungen `.de`, `.ch`, ...

Neben der TLD steht die *second level domain* (SDL). Diese können sich Organisationen und Privatpersonen bei den TLD registrieren lassen. Für die .de-Domain ist die DENIC zuständig. SDL und TDL bilden die *Zone*. Innerhalb einer Zone können weitere Hosts definiert sein (`www.example.de`, `ftp.example.de`, ...). Die Verwaltung der Zone erfolgt üblicherweise durch den Inhaber der SLD.

Jeder Rechner, der mit dem Internet kommuniziert, kennt mindestens einen Nameserver. Dieser wird entweder direkt konfiguriert, oder dem Rechner über DHCP (s. Abschn. 4.3) mitgeteilt. Der Host, der eine Namensauskunft wünscht, befragt nun die ihm bekannten Nameserver der Reihe nach.

Umgekehrt kann das DNS-System IP-Adressen in FQDN umsetzen. Zur Überprüfung beider Richtungen stehen auf einem Linux-System folgende Befehle zur Verfügung:

```
nslookup <dns-name>     (Windows)
host <dns-name>         (Linux, filius)
```

Mit diesen Kommandos kann auch der FQDN für eine gegebene IP-Adresse abgefragt werden.

Aufgabe:

- Ermitteln Sie die IP-Adresse von `wagnertech.de`.
- Fragen Sie, welcher Name zur erhaltenen IP gehört.
- Machen Sie die diesselben Abfragen in filius und betrachten Sie den Datenaustausch.

## 4.3 Domain Host Configuration Protocol (DHCP)[11]

Das DHCP-Protokoll versorgt Rechnner innerhalb eines Netzes mit Konfigurationsdaten. Die wichtigsten sind dabei

[10] Suse 10: S. 600ff.
[11] TCP/IP

- die IP-Adresse,
- die IP-Adresse des Default-Gateways,
- die IP-Adresse des Name-Servers.

Aktiviert ein Client eine Schnittstelle (Netzwerkkarte, WLAN-Verbindung), die über das DHCP-Protokoll konfiguriert werden soll, läuft folgender Datenaustausch ab:

- Der Client schickt ein *DHCP Discover* als Broadcast, um im Netz befindliche DHCP-Server ausfindig zu machen.
- Der Server antwortet mit einem *DHCP Offer*. Dem Client werden Konfigurationsdaten angeboten.
- Der Client akzeptiert das Angebot mit einem *DHCP Request*.
- Der Server bestätigt die Konfigurationsdaten mit einem *DHCP ACK*.

Damit ist im Gutfall die Kommunikation beendet. Will ein Client die IP-Adresse wieder frei geben, erfolgt das mit einem *DHCP Release*.

Soll auf einem Client überprüft werden, ob dieser über DHCP mit den richtigen Daten versorgt worden ist stehen dafür folgende Befehle zur Verfügung:

```
ipconfig /all      (Windows)
ip a               (Linux)
ipconfig           (filius)
```

Aufgabe:

- Führen Sie die genannten Befehle auf Ihrem Rechner aus.
- Führen Sie in *filius* `ipconfig` auf der Konsole des Client `172.16.10.0` aus.
- Betrachten Sie hier auch den Nachrichtenfluss, insbesondere den Verkehr nach dem Start des Rechners.

## 4.4 Network Address Translation (NAT)

Beim Routing wird davon ausgegangen, dass eine IP-Adresse weltweit eindeutig ist. Eine Ausnahme bilden dabei die privaten Netzwerke. Kommuniziert ein Rechner eines privaten Netzwerkes mit dem Internet, muss seine (private) IP-Adresse am Gateway auf eine eindeutige Adresse umgesetzt werden. Als Sendeadresse wird die Adresse des Gateways eingesetzt. Auf dem Rückweg muss das Gateway dann wissen, welchem internen Rechner die Antwort zugestellt werden muss. Diesen Vorgang nennt man *Network Address Translation* (NAT) oder *IP Masquerade*. Um nun auf einen Dienst, den ein Rechner im privaten Netz anbietet, auch vom Internet aus zugreifen zu können, muss der Dienst formal am Gateway angesprochen werden. Das Gateway muss dann wissen, an welchen internen Rechner die Anfragen delegiert werden müssen (*port forwarding*).

## 4.5 Übungen

Mit folgenden Aufgaben werden die Auswirkungen des Ausfalls bestimmter Netzwerkdienste illustriert.

### DHCP, DNS, Routing

Aufgabe:

Im grünen Netz werden die Funktionen DHCP und DNS durch den ads-Server zur Verfügung gestellt. Um zu demonstrieren, wie sich ein Serverausfall auswirkt sollen nun folgende Schritte durchgeführt werden:

- Verwenden Sie einen Client im grünen Netz.
- Ermitteln Sie mit `host lmu.de` die IP-Adresse dieses Rechners und notieren Sie sich diese.
- Beenden Sie den DNS-Server und den DHCP-Server auf dem ads-Rechner.
- Ermitteln Sie die Route zum `lmu.de`-Rechner, einmal indem Sie im `traceroute`-Befehl den Rechnernamen, einmal indem Sie die IP-Adresse verwenden.
- Versuchen Sie mit dem Webbrowser die Internetseite von `lmu.de` anzuschaun: Einmal indem Sie `lmu.de` in den Browser eingeben, einmal mit der IP-Adresse.
- Verbinden Sie einen weiteren Linux-Client mit dem grünen Netz und versuchen Sie dasselbe.
- Schalten Sie DHCP- und DNS-Server auf ads wieder ein.

Zur Erklärung: Der erste Client über DHCP die Adressen von DNS-Server und Gateway bekommen. Wenn die Namensauflösung über DNS ausfällt, kann die Route immer noch auf Basis der IP-Adressen ermittelt werden. Der zweite Client bekommt aber über DHCP keine Adressen mitgeteilt und kennt daher das Gateway nicht, über das er ins Internet käme.

Aufgabe:

Eine andere Ausfallmöglichkeit ist der Übergang ins Internet.

- Der Ausfall der Internetverbindung wird dadurch simuliert, dass die IP-Adresse der roten Schnittstelle am ipfire verändert wird. Damit können keine Pakete zurückgeroutet werden.
- Prüfen Sie mit `ipconfig`, ob der Client mit den richtigen DNS- und Gateway-Daten versorgt wurde.
- Prüfen Sie am Client im grünen Netz mit `host ads` die schulinterne Namensauflösung.
- Prüfen Sie mit `host lmu.de` die Auflösung eines Internet-Namens.
- Prüfen Sie mit `ping 85.214.60.111` die Verbindung zum Internet.

### Firewall

Das blaue Netz ist durch restriktive Regeln in der Firewall des ipfire geschützt. Ihr neuer Client im blauen Netz hat zwar die Adressen von DNS-Server und Gateway erhalten, es funktioniert aber fast nichts.

Aufgabe:

- Fügen Sie dem blauen Netz einen weiteren Client hinzu.
- Prüfen Sie mit `ping <IP-Adresse>`, ob DNS-Server und Gateway erreichbar sind.

  Die Firewall ist so eingestellt, dass `ping`-Verkehr grundsätzlich erlaubt ist.
- Prüfen Sie mit `host lmu.de`, ob der Name aufgelöst werden kann.
- Prüfen Sie mit `ping 85.214.60.111`, ob `lmu.de` erreichbar ist.
- Versuchen Sie mit dem Webbrowser die Internetseite von `lmu.de` anzuschaun: Einmal indem Sie `lmu.de` in den Browser eingeben, einmal mit der IP-Adresse.
- Öffnen Sie im ipfire die Firewallregeln und fügen Sie für Ihre IP eine Regel wie für den bestehenden Client hinzu. Führen Sie obige Prüfungen erneut aus.

# 5 Bedienung von Wireshark

## 5.1 Anzeige und Analyse

In diesem Abschnitt werden grundlegende Bedienungselemente von Wireshark vorgestellt.

- Profil: Rechts unten finden Sie die Anzeige des aktuellen Profils. Mit der rechten Maustaste können Sie ein neues Profil erstellen.
- Einstellungen → Appearance → Layout: Fenstereinteilung
- Zeitformat: Oftmals wird eine absolute Angabe der Zeit benötigt. Dies kann unter Ansicht → Format der Zeitanzeige angepasst werden.
- Weitere Spalten: Teilwerte aus Paketen lassen sich über Rechtsklick als weitere Spalten der Anzeige hinzufügen.
- Filter definieren: Filter können durch Rechtsklick auf eine Eigenschaft des Pakets definiert werden.
- Filter speichern: Innerhalb des ausgewählten Profils kann ein eingestellter Filter durch Drücken des „+"-Knopfes (rechts oben) gespeichert werden.
- Filter-Referenz[12]

[12] https://www.wireshark.org/docs/dfref/

Aufgabe:

- Laden Sie die Datei `Programmieren_3.dump`.
- Definieren Sie einen Filter für die Pakete des `knx`-Protokolls für `TUNNELLING_REQUEST` der Quelle `192.168.0.113`.
- Speichern Sie diese Filtereinstellung.
- Legen Sie eine weitere Spalte für den `Service Type Identifier` an.

## 5.2 Mitschnitt des Netzwerkverkehrs

Um auf einer Schnittstelle den Netzwerkverkehr mitschneiden zu können, werden gundsätzlich Administratorerechte benötigt. Da es unschön ist eine komplette Anwendung wie Wireshark mit so vielen Privilegien auszustatten, werden schlanke Komanndozeilenwerkzeuge für diese Aufgabe bevorzugt. Ein weiterer Vorteil von Kommandozeilenwerkzeugen ist, dass diese auf entfernten Rechnern, an denen der Mitschnitt erfolgen soll, leichter verfügbar sind.

Die Windows-Variante liefert diverse Kommandozeilenwerkzeuge[13]. Für den Mitschnitt eignet sich `dumpcap`[14]. Auf Linux-Maschinen steht `tcpdump`[15] zur Verfügung. Beide Werkzeuge akzeptieren Filterausdrücke im `pcap`-Format[16].

Aufgabe:

- Starten Sie einen Webbrowser.
- Schneiden Sie mit einem Werkzeug Ihrer Wahl den Verkehr von und nach Port 80 mit.[17]
- Rufen Sie eine „einfache" Intenetseite (z.B. `http://wagnertech.de/ports.html`) auf. Falls sich der Browser hartnäckig weigert das `http`-Protokoll (nicht `https`!) zu verwenden, kann man sämtliche gespeicherten Daten des Browsers löschen oder das Kommandozeilenwerkzeug `wget` verwenden.
- Analysieren Sie die mitgeschnittenen Daten mit Wireshark.

[13] https://wiki.wireshark.org/Tools
[14] https://www.wireshark.org/docs/man-pages/dumpcap.html
[15] https://www.tcpdump.org/manpages/tcpdump.1.html
[16] https://www.tcpdump.org/manpages/pcap-filter.7.html
[17] `sudo tcpdump -s 65535 -w p80.dump "port 80"`

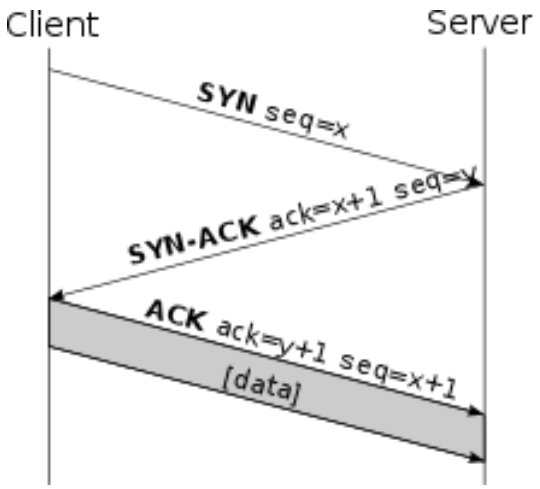


Abbildung 4: TCP-Verbindungsaufbau[18]

# 6 Ausgewählte Protokolle

## 6.1 TCP

Das *transfer control protocol* gewährleistet eine verbindungsorientierte gesichterte Verbindung. Eine TCP-Verbindung muss daher zuvor aufgebaut werden. Der Server muss bereit sein, eine neue Verbindung anzunehmen. Der Client iniziiert den Verbindungsaufbau (s. Abb. 4). Datenübertragungen der Anwendungsschicht werden mit einem `ACK` auf der TCP-Schicht bestätigt.

Aufgabe:

- Betrachten Sie sich beim Mitschnitt der letzten Aufgabe den Verbindungsaufbau.
- Filtern Sie nach Paketen, die rein auf TCP-Ebene ausgetauscht werden, nicht aber zum `HTTP`-Protokoll gehören.

## 6.2 ICMP

Auf dem IP-Protokoll baut auch das *internet controll message protocol* (ICMP) auf. Dieses einfache verbindungslose Protokoll meldet Fehler und steuert Nachrichten im Auftrag von IP.

Die ICMP-Fehlermeldungen werden von den Knoten generiert, die auf ein Übertragungsproblem stoßen, und sie werden an den Knoten geschickt, das problemverursachende Datagramm ausgesendet hat. Sowohl Router als auch Host-Rechner können ICMP-Fehlermeldungen aussenden:[19] ICMP-Pakettypen[20]

[18] https://de.wikipedia.org/wiki/Transmission_Control_Protocol (13.4.2021)
[19] [TCP/IP]: S. 115f.
[20] https://de.wikipedia.org/wiki/Internet_Control_Message_Protocol#Die_ICMP-Pakettypen

Die bekannteste Anwendung ist der Pingrequest: Ein ICMP-Paket (Typ 8: Echoanforderung) wird versendet und mit Typ 0 (Echoantwort) quittiert. Das Pingprogramm misst die Zeit zwischen Anforderung und Antwort.

Aufgabe:

- Schneiden Sie einen Pingrequest mit.
- Analysieren Sie den Request mit Wireshark.

## 6.3 UDP

Das verbindungslose *user datagram protocol* (UDP) wird meist zur Übertragung kleiner Datenmengen verwendet. Die wesentliche Erweiterung zum IP-Protokoll ist das Enthaltensein einer Portnummer.[21] Typische Anwendungen, die UDP verwenden, sind die Namensauflösung (*domain name service*, DNS) und `traceroute`.

`traceroute` versendet ein UDP-Paket beginnend mit einer *time to live* (TTL) von 1. TTL ist Teil des IP-Protokolls. Der TTL-Zähler wird bei jedem Router um eins verringert. Erreicht TTL den Wert 0, wird eine ICMP-Fehlermeldung (Typ 11: Zeitüberschreitung) vom Router an den Sender zurück gesenset. `traceroute` erhöht den Start-TTL-Wert so lange, bis das Ziel erreicht wird. Durch die erhaltenen ICMP-Fehlermeldungen kann die Route rekonstruiert werden.

Aufgabe:

- Schneiden Sie einen `traceroute`-Befehl an einen Server Ihrer Wahl mit.
- Analysieren Sie den Mitschnitt, indem Sie auf UDP und ICMP-Pakete filtern.
- Verwenden Sie TTL als zusätzliche Spalte.

## 6.4 SMTP

Der Transport einer E-Mail findet immer zwischen einem Client (er versendet die E- Mail) und einem Server (er nimmt die E-Mail an) statt. Für das weitere Verständnis eines Mailservers ist es wichtig, sich vor Augen zu führen, dass auch ein Mailserver als Client agieren kann – nämlich dann, wenn er eine E-Mail zu einem anderen Server transportiert.

Wenn ein Client eine E-Mail an einen Server übergibt, dann werden immer die folgenden Schritte absolviert:

- Client verbindet sich mit Server.
- Server und Client stellen sich einander vor.
- Client erzählt dem Server, was er veranlassen will.

[21] [TCP/IP]: S. 57.

- Server prüft, ob der Client (oder Absender) autorisiert ist.
- Server übernimmt E-Mail (oder weist sie ggf. auch ab).
- Server bestimmt den Weg zum Empfänger.
- Server transportiert die E-Mail näher zum Empfänger.

Je Nachdem, welches Protokoll – SMTP oder ESMTP – zur Anwendung kommt, werden dabei weniger (SMTP-Protokoll) oder mehr (ESMTP-Protokoll) Informationen in der Kommunikation zwischen Client und Server ausgetauscht.[22]

Dieser SMTP-Nachrichtenaustausch kann einfach mit einer Telnet-Emulation nachgestellt werden:

```
$telnet server.de 25
HELO sender.de
MAIL FROM:max@sender.de
RCPT TO:monika@server.de
DATA
--- Some Text ---
.
QUIT
```

Aufgabe:

- Schneiden Sie das SMTProtokoll mit.
  Ein „normaler" SMTP-Server wird das Protokoll wegen Spam-Prüfungen abbrechen.
- Analysieren Sie den Mitschnitt mit Wireshark.
  Unterscheiden Sie dabei folgende Protokollphasen:
  - Namensauflösung (UDP/DNS)
  - TCP-Verbindungsaufbau
  - SMTP-Verkehr

## 6.5 Verschlüsselte Protokolle

### Grundlagen

Bei Verschlüsselungsverfahren wird grundsätzlich zwischen symmetrischen und asymmetischen Verfahren unterschieden. Bei den symmetrischen Verfahren müssen beide Seiten denselben Schüssel für Ver- und Entschlüsselung haben. Dieses Verfahren ist nur möglich, wenn die Partner über einen bereits gesicherten Weg die Schlüssel austauschen können. Der Vorteil von symmertrischen Verfahren liegt im geringeren Rechenaufwand.

Bei den asymmetrischen Verschlüsselungsverfahren werden zwei Schlüssel erzeugt. Mit dem einen wird die Nachricht verschlüsselt, mit dem anderen entschlüsselt. Einer der Schlüssel wird dabei für

[22] Hildebrandt (2008): S. 8.

privat erklärt und niemandem mitgeteilt. Der andere Schlüssel wird veröffentlicht. Damit lassen sich zwei unterschieliche Anwendungsfälle konstruieren:

- Authenzität: Eine Nachricht stammt wirklich vom Absender.

  Dazu verschlüsselt der Absender die Nachricht mit seinem privaten Schlüssel. Sie kann von jedem, der den öffentlichen Schlüssel des Absenders hat, entschlüsselt werden. Dieser kann dabei sicher sein, dass die Nachricht wirklich vom Absender stammt, da kein anderer dessen privaten Schlüssel hat.

  Hierbei ist es effektiver, nicht die gesamte Nachricht zu verschlüsseln, sondern nur den Hashwert, d.h. einen Wert, der über einen bekannten Algorithmus aus der Nachricht generiert wird. Damit ist die Nachricht gegen Veränderungen gesichert. Würde ein Angreifer die Nachricht verändern, würde sich ein anderer Hashwert ergeben, der dann nicht mehr mit dem vom Absender verschlüsselten Wert übereinstimmt.

- Vertraulichkeit: Eine Nachricht kann nur Empfänger gelesen werden.

  Der Absender verschlüsselt die Nachricht mit dem öffentlichen Schlüssel des Empfängers. Nur der Besitzer des zugehörigen privaten Schlüssels, also der Empfänger, kann die Nachricht entschlüssln.

Bei den asymmetrischen Verschlüsselungverfahren ist das bekannteste das RSA-Verfahren [23]. Diese nutzt den Satz von Euler-Fermat, nach dem für zwei Primzahlen *p*, *q* und $n = p \cdot q$ gilt:

$$M^{(p-1)(q-1)} \bmod n = 1 \tag{1}$$

Wählt man nun zwei Schlüssel $e$ und $d$, für die gilt:

$$e \cdot d \bmod (p-1)(q-1) = 1 \tag{2}$$

folgt daraus:

$$e \cdot d = k \cdot (p-1)(q-1) + 1, k \in \mathbb{N} \tag{3}$$

Die Verschlüsselung erfolgt nun nach der Vorschrift $N = M^e \bmod n$, die Entschlüsselung mit dem anderen Schlüssel: $M = N^d \bmod n$.

Zu zeigen ist nun:

$$(M^e \bmod n)^d \bmod n = M \tag{4}$$

Mit $(a \bmod n)(b \bmod n) = ab \bmod n$ lässt sich daraus bilden:

$$\begin{aligned} M^{ed} \bmod n &= M^{k\cdot(p-1)(q-1)+1} \bmod n && \text{(s. Gl. 3)} \\ &= M^{k\cdot(p-1)(q-1)} \bmod n \cdot M \\ &= (M^{(p-1)(q-1)} \bmod n)^k \cdot M && \text{(s. Gl. 1)} \\ &= M \end{aligned}$$

Mit gpg steht ein Programm zur Verfügung, mit dem sich Schüssel erzeugen und verwalten, sowie Nachrichten ver- und entschlüsslen lassen:

[23] Rivest, Shamir, Adleman. A Method for Obtaining Digital Signatures and Public Key Cryptosystems, Comm.ACM 21/2 (1978).

`gpg --gen-key`: Erzeugt ein Schlüsselpaar

`gpg --list-key`: Zeigt vorhandene öffentliche Schlüssel an

`gpg --import <file>`: Importiert einen fremden öffnetlichen Schlüssel

`gpg --armor --export <uid> > <file>`: Exportiert den genannten Schlüssel in eine Datei

`gpg --armor -e -r <email@empfaenger> <file>`: Verschlüsselt eine Datei mit dem öffenlichen Schlüssel des Empfängers und legt sie mit der Endung `.asc` ab.

`gpg -d <enc-file> > <decr-file>`: Entschlüsselt die gegebene Datei

Anmerkung: Die Option `--armor` bewirkt, dass nur druckbare Zeichen im Ergebnis sind.

Grundsätzlich funktionierten alle verschlüsselten Protokolle nach demselben Muster:

- Client baut eine TCP-Verbindung auf.
- (Client stellt die ihm zur Verfügung stehenden Verschlüsselungsverfahren vor.)
- (Server stellt die ihm zur Verfügung stehenden Verschlüsselungsverfahren vor.)
- Client wählt ein Verfahren aus.
- (Server schickt seinen öffentlichen Schlüssel)
- Client generiert einen symmetrischen Sitzungsschlüssel, verschlüsselt diesen mit dem öffentlichen Schlüssel des Servers und schickt ihn an den Server.
- Die eigentliche Kommunikation erfolgt nun mit dem symmetrischen Schlüssel.

Bei zertifikatbasierten Protokollen (`https`, `smtps`) merkt sich der Client das Serverzertifikat. Ein Zertifikat enthält neben dem öffentlichen Schlüssel und dem Hinweis auf den verwendeten Algorithmus einen Fingerabdruck, mit dem zusammen mit der Vertrauenskette, über die das Zertifikat ausgestellt wurde, die Echtheit überprüft werden kann. Damit verkürzt sich der dargestellt Ablauf um die eingeklammerten Punkte.

Aufgabe:

- Schneiden Sie eine ssh-Sitzung mit, indem Sie sich mit der Linux-VM verbinden.
- Analysieren Sie den Verkehr mit Wireshark:
  - Schränken Sie die betrachtenen Pakete auf den Austausch mit der VM ein.
  - Schränken Sie weiter auf das `ssh`-Protokoll ein.
  - Vollziehen Sie die oben dargestellten Schritte nach.
- Schneiden Sie eine https-Sitzung mit.
- Werten Sie diese Daten aus.

# 7 Quellen


| | |
|---|---|
| Hildebrandt (2008) | Hildebrandt, Ralf; Koetter, Patrick Ben (2008): Postfix. Einrichtung, Betrieb und Wartung. |
| .[TCP/IP] | ICN TI Enabling. Kurs ICP/IP Grundlagen. Siemens 1999. |