



Zwischensysteme

Christian Schülke

Elias Martin

Lena Höhn

Laura Pech

Auszubildende Fachinformatiker für Anwendungsentwicklung und Systemintegration

Projektarbeit

2019 / 2020

IF 18 - 4

S5

Betreut von Herr Räntzsch

BSZ für Elektrotechnik

Strehleener Pl. 2, 01219 Dresden

30.06.2020

Erklärung der Projektgruppe

Wir erklären hiermit, dass diese Dokumentation von uns verfasst wurde – abgesehen von der der Unterstützung der Betreuer – und dass nur die angegebenen Quellen und Hilfsmittel verwendet wurden.

Datum/ Unterschrift:




Elias Martin



Lena Höhn



Christian Schülke



Laura Pech

Inhalt

1	Vorbereitung	3
1.1	Begriffserklärung Zwischensystem und Endsystem	3
1.2	Nutzen der Protokollschichten des Osi-Modells	3
1.3	Einordnung von Switch und Layer-3-Switch in Abbildung	7
1.4	Zweck und Header-Aufbau von ICMP, Nutzung des Protokolls in der Konsole unterschiedlicher Betriebssysteme	7
1.5	Peer-2-Peer Netzwerk entsprechend DFÜ-Modell	8
1.6	Skizze des Protokollstapels	9
1.7	Headerstruktur für HTTP, TCP, IPv4, IPv6 und Ethernet	9
1.8	Analyse von Anweisungen für das Protokoll IPv6	11
2	Durchführung	14
2.1	Versuchsaufbau	14
2.1.1	Installieren und Konfigurieren der Software auf den WS	14
2.1.2	Netzwerkfunktionalität beider Workstations	16
2.1.3	Netzwerkfunktionalität beider Workstations	17
2.1.4	Installieren von Wireshark	19
2.2	Aufgaben	20
2.2.1	Aufzeichnung und Analyse der ICMP requests und replys	20
2.2.1.1	Fabliches Markieren von Bestandteilen der Paketinhalten	20
2.2.1.2	Ermittlung der WS1-IP und des dazugehörigen Hexadezimalcode im IP-Header	20
2.2.1.3	Bestimmung der MAC und des NIC	21
2.3	Aufzeichnung der Protokollübertragung von hallo.htm zur WS2	21
2.3.0.1	Header und Payload jeder Protokollschicht und Zuordnung zu OSI Schichten	21
2.3.0.2	Verhältnis Payload zur Paketgröße nach DoD-Modell	26
2.3.0.3	Anteil der Nutzdaten zum Frame für den Request und den Response	30
2.3.1	Umstellung von IPv4 auf IPv6	30
2.3.1.1	Ausführung des Befehls ipv6 if	30
2.3.1.2	Testen der Verbindung mit ping, Ermittlung des korrektem Befehls	31
2.3.1.3	Testen der Aufzeichnung der Kommunikationsbeziehung mit Wireshark und die Unterschiede zu 2.2.1.1	31
2.3.1.4	Testen einer Ordnerfreigabe zur WS2	32
3	Abbildungsverzeichnis	36
4	Tabellenverzeichnis	40
5	Quellen	41
6	Glossar	42

1 Vorbereitung

1.1 Begriffserklärung Zwischensystem und Endsystem

Ein **Endsystem** oder Endgerät ist ein Computer oder ein anderes Peripheriegerät, an welchem keine weiteren Geräte angeschlossen sind.

Beispiele für Endsysteme sind:

- Drucker
- Geldautomat
- Surfstation
- Lautsprecher
- Kamera

Außerdem muss es **Zwischensysteme** geben, die gesendete Datenpakete an die richtige Adresse weiterleiten.

Solche Zwischensysteme sind Switches, Bridges und Router.

1.2 Nutzen der Protokollschichten des Osi-Modells

Das OSI Modell ist ein Modell, welches die Ebenen die ein Netzwerk ausmachen beschreibt.

Bitübertragungsschicht

(Physical Layer)

- elektrische / physische Übertragung der Daten

Sicherungsschicht

(Data Link Layer)

- alle Vorkehrungen, die dafür sorgen, dass aus der physikalischen Übertragung ein verlässlicher Datenfluss wird

Vermittlungsschicht

(Network Layer)

- Komponenten und Protokolle, die an der Verbindung zwischen Rechnern beteiligt sind
- das sogenannte Routing - Weiterleiten von Daten in andere logische und oder physikalisch inkompatible Netzwerke
-

Transportschicht
(Transport Layer)

- verbindungsorientierte Protokolle wie TCP und verbindungslose Protokolle wie UDP
- ein wichtiger Aspekt dieser Schicht ist Multiplexing - Anbindung der Datenpakete an konkrete Prozesse auf den kommunizierenden Rechnern
- Segmentierung des Datenstroms und Datenstauvermeidung

Kommunikationssteuerungsschicht
(Session Layer)

- sichert Kommunikation zwischen kooperierenden Anwendungen oder Prozessen auf verschiedenen Rechnern
- organisiert und synchronisiert Datenaustausch

Darstellungsschicht
(Presentation Layer)

- Konvertierung und Übertragung von Datenformaten, Datensätzen, Zeichensätzen, grafische Anweisungen und Dateidienste
- systemabhängige Darstellung von Daten
- Datenkompression, Verschlüsselung
- stellt sicher, dass Daten die von der Anwendungsschicht des einen Systems gesendet werden von der Anwendungsschicht eines anderen Systems gelesen werden können

Anwendungsschicht
(Application Layer)

- unmittelbare Kommunikation zwischen Benutzeroberflächen der Anwendungsprogramme
- Das Anwendungsprogramm selbst zählt nicht dazu

Tabelle 1: Osi Modell

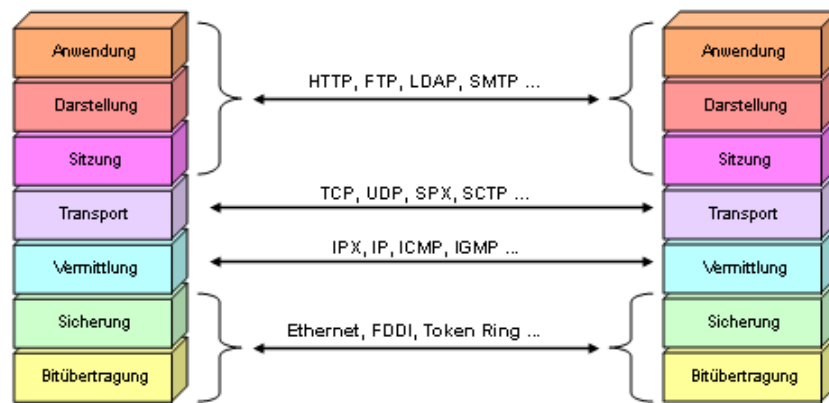


Abbildung 1: Osi Modell

Repeater

Ein Repeater verstärkt ganz simpel die elektronischen Signale und nutzt deswegen nur die 1. Schicht des OSI-Modells. Der in der Abbildung unterhalb veranschaulichte HUB (Multi-Port-Repeater) sendet ein von PC0 gesendetes Paket an alle anderen PC's weiter, da er keine Möglichkeit hat, an ein bestimmtes Gerät zu senden, da er keine MAC-Adressen speichert. Er teilt keine Broadcastdomäne.

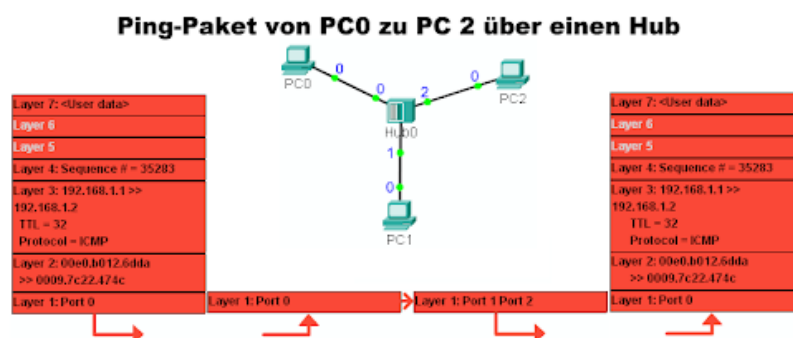


Abbildung 2: Ping über einen Hub

Bridge

Eine Bridge verbindet mehrere über Kabel verbundene Netzwerke / PCs miteinander, sodass sie ein einzelnes Netz repräsentieren. In Bezug auf das OSI Modell werden über die erste Schicht Signale versandt und über die zweite Schicht werden die Signale einem Zielort via einer Link-Layer-Address zugeordnet. In der folgenden Abbildung sieht man einen Switch (Multi-Port-Bridge), welcher die von PC0 gesendeten Pakete anhand der MAC-Adresse, aus dem entsprechenden Port, an PC2 weiterleitet.

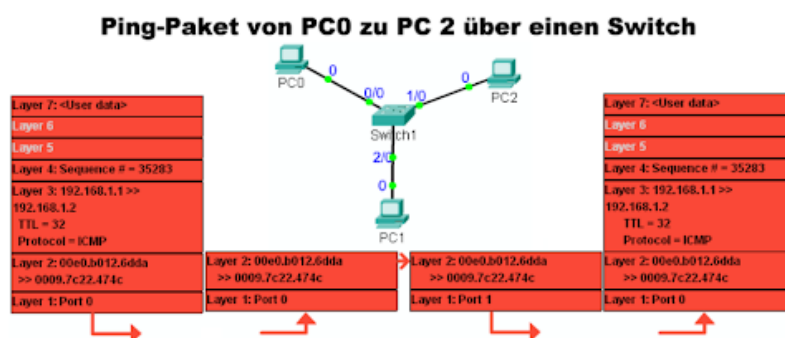


Abbildung 3: Ping über einen Switch

Router

Ein Router verbindet mehrere Netzwerke miteinander. Anhand von Routing-Tabellen (statisch oder dynamisch) leitet er die Datenpakete in die entsprechende Netze oder über ein Default-Gateway weiter. Die Routing-Entscheidungen geschehen aufgrund von IP-Adressen (OSI-Layer 3) und ggf. weiteren Parametern z.B. anhand der Pfadkosten beim OSPF-Protokoll. PC0 sendet das Datenpaket an Router1 mit seiner IP-Adresse als Quelladresse und der Zieladresse (PC2). Der Router kennt das Zielnetz, da es direkt angeschlossen ist und sendet es an PC2 weiter.

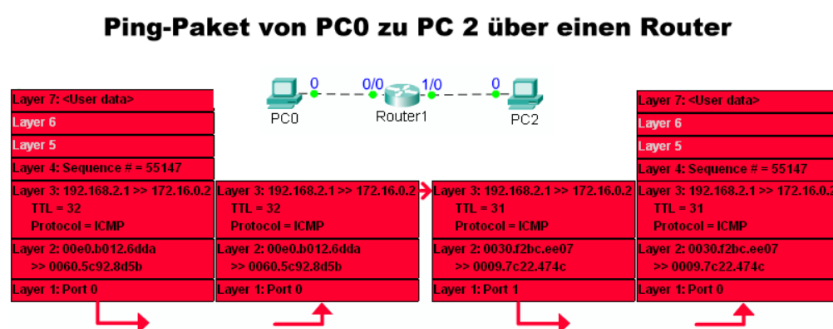


Abbildung 4: Ping über einen Router

Gateway

Ein Gateway ist ein computer-ähnliches Gerät, welches eine Kommunikation zwischen zwei oder mehr unterschiedlichen Systemen herstellt. Ein Gateway kann Software oder Hardware sein. Um mit verschiedenen Arten von Netzwerken zu kommunizieren, muss der Netzwerk Gateway auf mehreren Schichten des OSI Modells operieren, unter Umständen auch auf allen 7, da er zum Beispiel Sessions erstellen und verwalten muss, Daten entschlüsseln und unterschiedliche physikalische und logische Adressen übersetzen muss. Wegen der vielen Aufgaben die ein Gateway gleichzeitig erledigen muss ist er um einiges langsamer, als die anderen oben genannten Geräte. Der Gateway, der in der Aufgabenstellung dargestellt wird greift auf alle 7 Layer zu und muss deswegen eine Software auf einem Computer sein, auf die ein Benutzer via einer Oberfläche zugreifen kann.

1.3 Einordnung von Switch und Layer-3-Switch in Abbildung

Switch

Ein Layer-2-Switch arbeitet mit Data-Link Layer-Adressen (MAC). Er benutzt also nur die Bitübertragungsschicht und die Sicherungsschicht. Er sendet die Daten weiter an einen fest angegebenen Punkt anhand der MAC-Adressen.

Er ist als Äquivalent zur Bridge zu sehen. Denn er ist im Prinzip eine Multi-Port-Bridge und im Gegensatz zum HUB, erzeugt er keinen unnötigen Traffic, da nur an ein bestimmtes Ziel und nicht an alle gesendet wird (durch MAC-Adressen-Tabelle).

Layer-3-Switch

Ein Layer-3-Switch ist ein Switch, welcher um die Routing-Funktionen erweitert wurde und sonst alle Funktionen von einem Layer-2-Switch beibehält. Deshalb ist er mit dem Router aus Abbildung 1 gleichzusetzen. Durch die Erweiterung auf Layer 3 unterstützen diese Switches auch Inter-VLAN-Routings. Erweiterte Funktionen, wie NAT, IPSec oder Firewall-Filtering werden allerdings nicht unterstützt.

1.4 Zweck und Header-Aufbau von ICMP, Nutzung des Protokolls in der Konsole unterschiedlicher Betriebssysteme

Das ICMP (Internet Control Message Protocol) tauscht Informations- und Fehlermeldungen über IPv4 in Rechnernetzen aus. Das Äquivalent in IPv6 heißt ICMPv6.

Für jeden Rechner und Router ist es Standard, dass sie ICMP verstehen.

ICMP Pakete dienen dazu Diagnose Informationen zurück an die Quelle zu senden, wenn der Router Pakete verwirft. Beispielsweise, wenn das Ziel nicht erreichbar ist oder die TTL abgelaufen ist.

So wird zum Beispiel mit dem Befehl "ping" ein Test Datenpaket über das ICMP Protokoll gesendet.

- Zweck, Headeraufbau
- Nutzung in der Konsole unterschiedlicher Betriebssysteme

Der Befehl ping ist unter den meisten Betriebssystemen wie Windows, Linux (und anderen Unixartigen), Unix oder macOS, aber auch als Analysetool auf Geräten wie Routern nutzbar. Die Ausführung des Befehls und somit die Aussendung der ICMP Pakete unterscheiden sich jedoch je nach Betriebssystem, so sendet Windows eine begrenzte Anzahl an Paketen während Linux eine unbegrenzte Anzahl sendet und nur manuell abgebrochen werden kann.

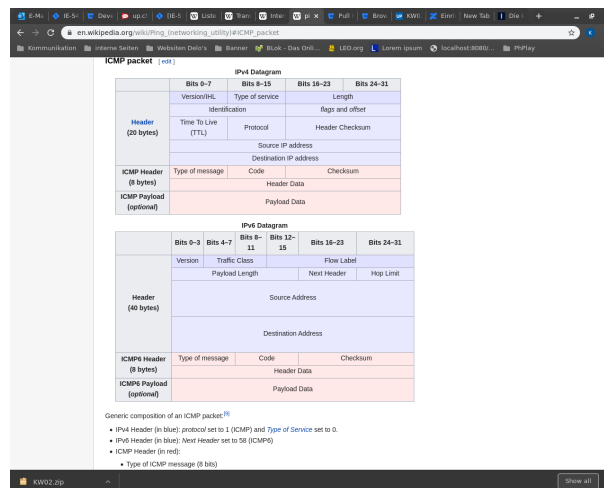


Abbildung 5: IMCP Aufbau

1.5 Peer-2-Peer Netzwerk entsprechend DFÜ-Modell

DDE bezeichnet die Datenendeinrichtung und stellt den Sender oder den Empfänger dar. Sie kontrolliert und steuert die Datenfernübertragung.

DÜE bezeichnet die Datenübertragungseinrichtung. Sie ist die Verbindung zwischen den DDE's und dem Netzwerk und wandelt die Daten in eine geeignete Form für die Übertragung um.

Zwischen DDE und DÜE befindet sich eine serielle Schnittstelle (V.24, RS232), ein USB Kabel oder eine Funkverbindung wie Bluetooth.

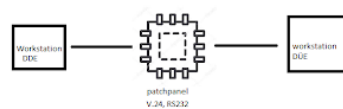


Abbildung 6: DFÜ Modell

1.6 Skizze des Protokollstapels

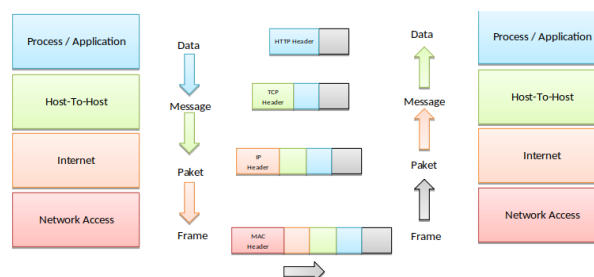


Abbildung 7: Skizze eines Protokollstapels eines HTTP Requests

1.7 Headerstruktur für HTTP, TCP, IPv4, IPv6 und Ethernet

Http Response Header

Übergeben zusätzliche Informationen über die Antwort, die nicht in die Status Line passen. Sie Enthalten Informationen über den Server und weitere Zugänge zu der Quelle. Diese sind durch die Request-URI gekennzeichnet.

response-header	=	Accept-Ranges	; Section 14.5
		Age	; Section 14.6
		ETag	; Section 14.19
		Location	; Section 14.30
		Proxy-Authenticate	; Section 14.33
		Retry-After	; Section 14.37
		Server	; Section 14.38
		Vary	; Section 14.44
		WWW-Authenticate	; Section 14.47

Abbildung 8: HTTP Response Header

HTTP Request Header

Geben zusätzliche Informationen über den Request mit, wie über den Client selbst, zum Server.

request-header	=	Accept	; Section 14.1
		Accept-Charset	; Section 14.2
		Accept-Encoding	; Section 14.3
		Accept-Language	; Section 14.4
		Authorization	; Section 14.8
		Expect	; Section 14.20
		From	; Section 14.22
		Host	; Section 14.23
		If-Match	; Section 14.24
		If-Modified-Since	; Section 14.25
		If-None-Match	; Section 14.26
		If-Range	; Section 14.27
		If-Unmodified-Since	; Section 14.28
		Max-Forwards	; Section 14.31
		Proxy-Authorization	; Section 14.34
		Range	; Section 14.35
		Referer	; Section 14.36
		TE	; Section 14.39
		User-Agent	; Section 14.43

Abbildung 9: HTTP Request Header

TCP Header

Dieser Header enthält die zur Kommunikation erforderlichen Daten und Dateiformat-beschreibende Informationen. Normalerweise sind TCP Header 20 Bytes lang, können aber mit den kaum

genutzen Optionen erweitert werden.

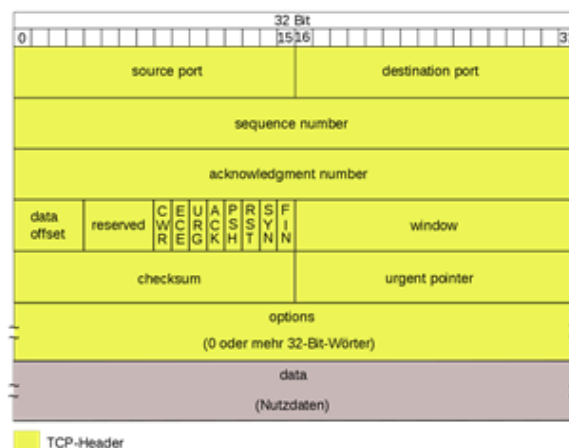


Abbildung 10: TCP Header

IPv4 Header

Die Länge von IPv4 Header ist normalerweise 20 Bytes, kann aber mit bestimmten Optionen auf bis 60 Bytes erweitert werden.

0-3	4-7	8-13	14-15	16-18	19-23	24-27	28-31
Version	IHL	DSCP	ECN	Gesamtlänge			
Identifikation				Flags	Fragment Offset		
TTL		Protokoll		Header-Prüfsumme			
Quell-IP-Adresse							
Ziel-IP-Adresse							
evtl. Optionen ...							

Abbildung 11: IPv4 Header

IPv6 Header

Hat eine feste Länge von 40 Bytes. Optionale selten genutzte Daten können in Extension Headern zwischen Header und Payload gesendet werden.

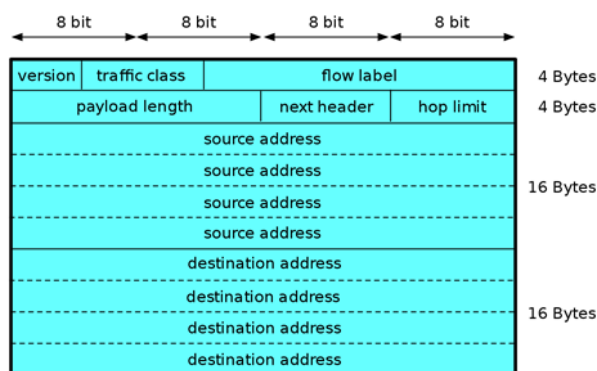


Abbildung 12: IPv6 Header

Ethernet

Enthält die Zieladresse (6 Bytes), Quell-MAC-Adresse (6 Bytes), das EtherType field (2 Bytes) und optional einen IEEE 802.1Q tag oder einen IEEE 802.1ad tag (4 Bytes).

1.8 Analyse von Anweisungen für das Protokoll IPv6

<p>netsh interface ipv6 show interfaces (früher ipv6 if)</p>	<p>Pingt das angegebene Interface an und gibt dabei die link-layer-adress, die ipv6 Adressen die zu dem Interface gehören und das aktuelle MTU und die maximale Anzahl der MTU's die das Interface unterstützen kann. Interface 1 ist ein Pseudo-Interface.</p> <p>zeigt an:</p> <p>Index - Bereichskennung</p> <p>Met - gibt die Pfadkosten an, je niedriger, desto besser, kann wenn es mehrere Routen gibt dazu verwendet werden zu entscheiden, welche Route verwendet wird</p> <p>MTU - maximale Anzahl an MTU's die das Interface unterstützt</p> <p>state - status des Interfaces, ist Interface enabled oder disabled</p> <p>name - Name des Interfaces</p>
<p>ping -6 ::1</p>	<p>Pingt den localhost an. Das heißt es wird ein ICMP-Paket mit einem TTL Wert von 128 gesendet und kommt vom localhost wieder zurück.</p>
<p>ping -6 Adresse%Bereichskennung</p>	<p>Pingt die Adresse über das in der Bereichskennung angegebene Interface an. Zum Beispiel wenn man die Adresse fe80::1%SCHNITTSTELLE einen ping heraus sendet, wird ein ICMP Paket an die Link-Local-Adresse fe80::1 über die Schnittstelle "SCHNITTSTELLE" an.</p>
<p>netsh interface ipv6 show route</p>	<p>Pingt jeden Hop bis zum Host an und verfolgt dabei die Route.</p> <p>Dabei werden ICMP-Pakete mit immer höher werden dem TTL-Wert ausgesandt, die dann nacheinander von den beteiligten Routern bearbeitet werden. Der höchste TTL-Wert entspricht dann dem des Hosts.</p> <p>Die Ausgabe zeigt dann die Hops bis zum Ziel an.</p> <p>Angezeigt werden:</p> <ul style="list-style-type: none"> - Der wie viele Hop wurde bewältigt - die Zeit die gebraucht wurde um den Hop zu bewältigen - die IP des Hops und die Benennung

ipv6 [-p] rc [IfIndex [Adress]]	Zeigt den ping zum "route cache" bzw den Ziel caches, von welchen es mehrere geben kann, je nachdem, wie viele Interfaces auf dem Weg passiert werden. Es werden von jedem Route Cache Eintrag das nächste Interface und die Nachbar Adresse angezeigt. Desweiteren wird der Pfad MTU zur Erreichung des Ziels durch das Interface und ob es ein Interface spezifisches route cache Eintrag ist angezeigt.
---------------------------------	--

Tabelle 2: Terminal Befehle

2 Durchführung

2.1 Versuchsaufbau

2.1.1 Installieren und Konfigurieren der Software auf den WS

Planung

- Zwei Win 8 Rechner starten
- Rechner per P2P verbinden
- Auf einer Maschine XAMPP installieren mit Auswahl nur Apache, auf der zweiten Maschine Wireshark installieren
- Auf beiden PC's Firewall deaktivieren
- DHCP auf statische IP's umstellen: 10.0.0.2 & 10.0.0.3
- Im xampp-Verzeichnis unter htdocs die Seite hallo.htm einfügen
- Im browser die Seite über "localhost/hallo.htm" und "ip:\hallo.htm" aufrufen

DoD Schichtmodel

Network Access Schicht

(Die Schicht, auf der die Geräte physisch verbunden sind)

- Bei uns über internen virtuellen Switch gelöst
- MAC Adresse wird ausgelesen

Internet Schicht (Die Schicht, die Netzwerkweite Verbindungen unabhängig von Übertragungsmedium ermöglicht)

- IP Adresse wird ausgelesen

Host to Host Schicht (Die Schicht, die den Transport von Daten und eine Peer to Peer Verbindung ermöglicht)

- Haben wir über Arbeitsgruppe ermöglicht

Process (Die Schicht, die die tatsächlichen Nutzdaten entsprechend den jeweiligem Protokoll ablegt)

- Ist bei uns via HTTP

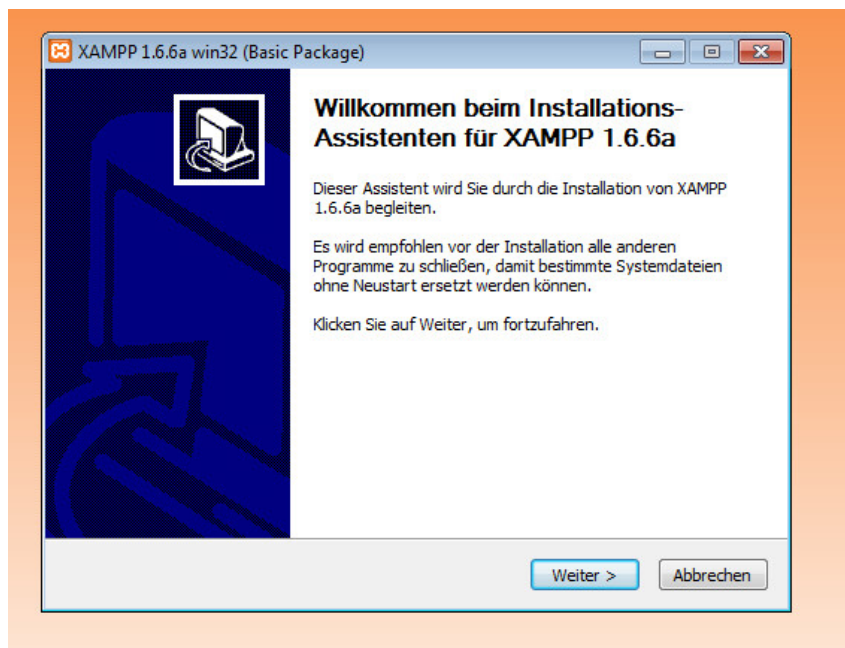


Abbildung 13: Installation mit XAMPP Wizard

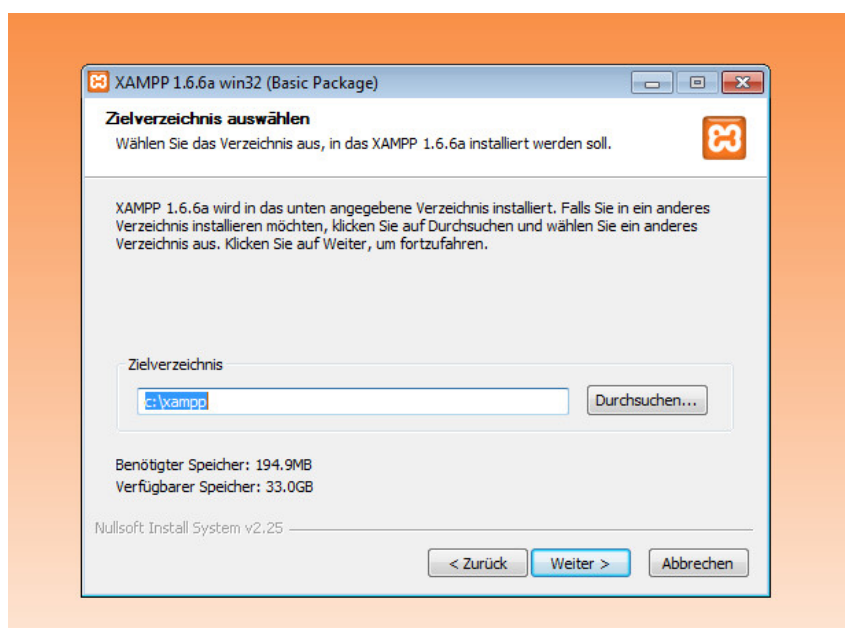


Abbildung 14: Installationsverzeichnis

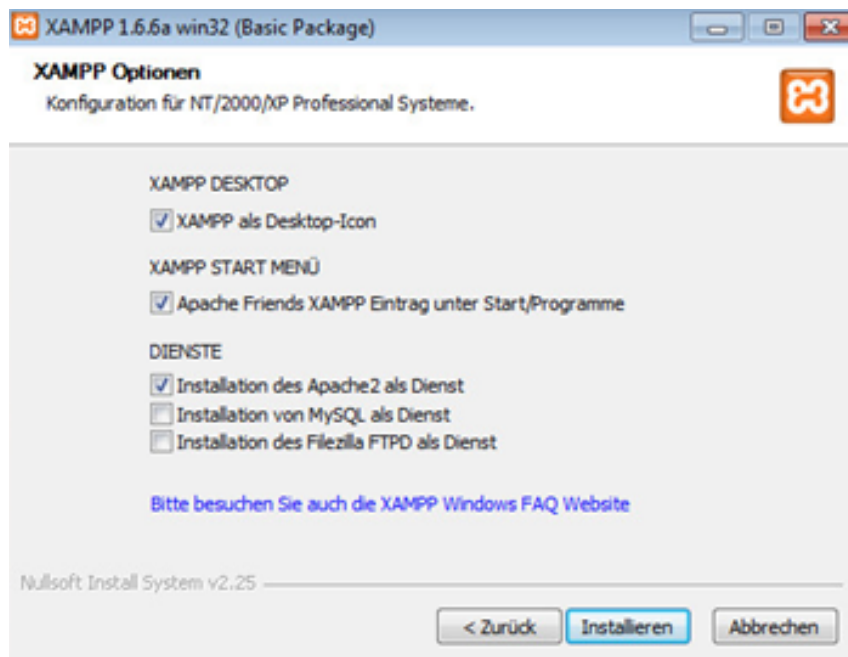


Abbildung 15: Konfiguration (Apache)

2.1.2 Netzwerkfunktionalität beider Workstations

Planung

- IPs der WS ermitteln
- gegenseitiges pingen der WS

```
C:\Users\Administrator>ipconfig

Windows-IP-Konfiguration

Ethernet-Adapter LAN-Verbindung 2:

    Verbindungsspezifisches DNS-Suffix:
    IPv6-Adresse. . . . . : 2003:c2:771d:ad01:7ccb:eb5e:5501:2097
    Temporäre IPv6-Adresse. . . . . : 2003:c2:771d:ad01:6101:e146:fef:edc
    Verbindungslokale IPv6-Adresse . : fe80::7ccb:eb5e:5501:2097%13
    IPv4-Adresse (Auto. Konfiguration): 169.254.32.151
    Subnetzmaske . . . . . : 255.255.0.0
    Standardgateway . . . . . : fe80::1%13

Ethernet-Adapter LAN-Verbindung:

    Verbindungsspezifisches DNS-Suffix:
    Verbindungslokale IPv6-Adresse . : fe80::f56b:a48a:e779:65bc%11
    IPv4-Adresse . . . . . : 10.0.0.3
    Subnetzmaske . . . . . : 255.0.0.0
    Standardgateway . . . . . :

Tunneladapter isatap.{2D7BCEE0-8C60-40D9-99B7-F7E771A6A1D0}:

    Medienstatus. . . . . : Medium getrennt
    Verbindungsspezifisches DNS-Suffix:

Tunneladapter isatap.{D08562DC-5D49-4A94-AD7D-5CC7D09F4629}:

    Medienstatus. . . . . : Medium getrennt
    Verbindungsspezifisches DNS-Suffix:

C:\Users\Administrator>ping 10.0.0.2

Ping wird ausgeführt für 10.0.0.2 mit 32 Bytes Daten:
Antwort von 10.0.0.2: Bytes=32 Zeit<1ms TTL=128
Antwort von 10.0.0.2: Bytes=32 Zeit<1ms TTL=128
Antwort von 10.0.0.2: Bytes=32 Zeit<1ms TTL=128
Antwort von 10.0.0.2: Bytes=32 Zeit<1ms TTL=128

Ping-Statistik für 10.0.0.2:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
    <0% Verlust>,
    Ca. Zeitangaben in Millisek.:
    Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms
```

Abbildung 16: Ping von Workstation 2 zu Workstation 1

```

C:\Users\Administrator>ipconfig

Windows-IP-Konfiguration

Ethernet-Adapter LAN-Verbindung:

    Verbindungsspezifisches DNS-Suffix:
    Verbindungslokale IPv6-Adresse . . : fe80::51c:acf8:a2e2:a8f5%11
    IPv4-Adresse . . . . . : 10.0.0.2
    Subnetzmaske . . . . . : 255.0.0.0
    Standardgateway . . . . . :

Tunneladapter isatap.{2D7BCEE0-8C60-40D9-99B7-F7E771A6A1D0}:

    Medienstatus. . . . . : Medium getrennt
    Verbindungsspezifisches DNS-Suffix:

C:\Users\Administrator>ping 10.0.0.3

Ping wird ausgeführt für 10.0.0.3 mit 32 Bytes Daten:
Antwort von 10.0.0.3: Bytes=32 Zeit<1ms TTL=128
Antwort von 10.0.0.3: Bytes=32 Zeit<1ms TTL=128
Antwort von 10.0.0.3: Bytes=32 Zeit<1ms TTL=128
Antwort von 10.0.0.3: Bytes=32 Zeit<1ms TTL=128

Ping-Statistik für 10.0.0.3:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
    (0% Verlust),
    Ca. Zeitangaben in Millisek.:
    Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms

```

Abbildung 17: Ping von Workstation 1 zu Workstation 2

2.1.3 Netzwerkfunktionalität beider Workstations

Planung

- XAMPP installieren
- Apache starten
- testen ob Dienst läuft - in browser ip des anderen servers eingeben und prüfen, ob web-seite aufgerufen wird

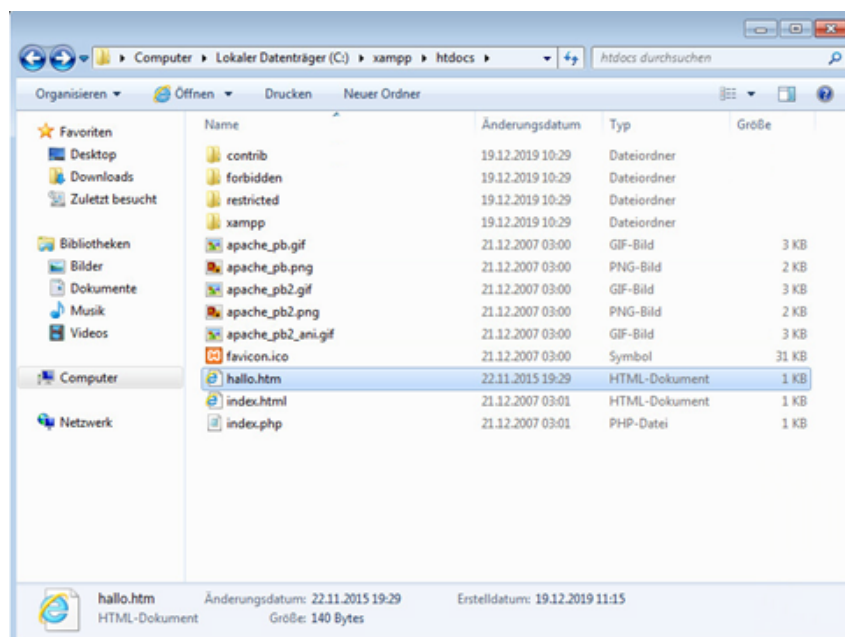


Abbildung 18: hallo.htm nach C:\xampp\htdocs kopieren

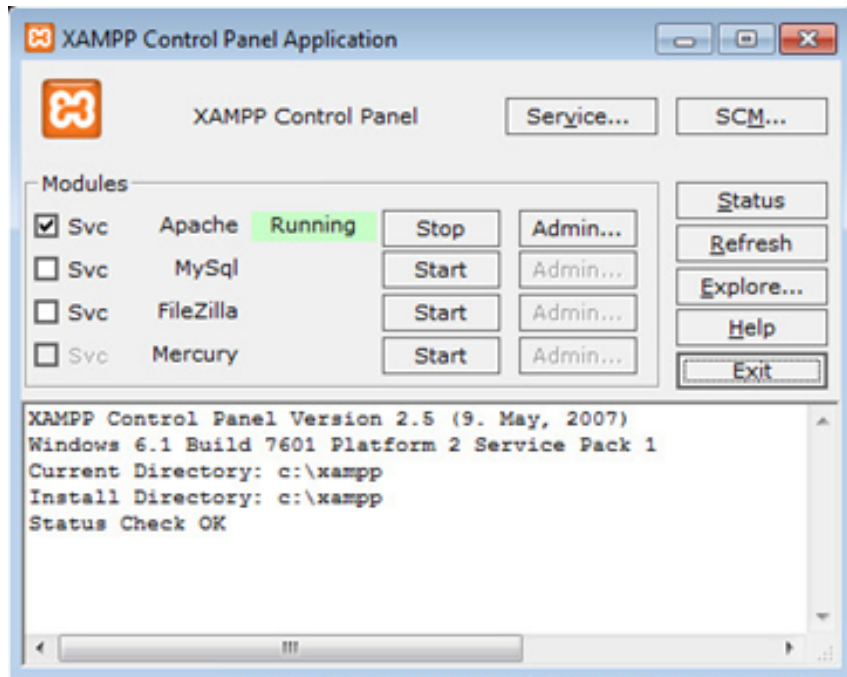


Abbildung 19: Apache in XAMPP Starten

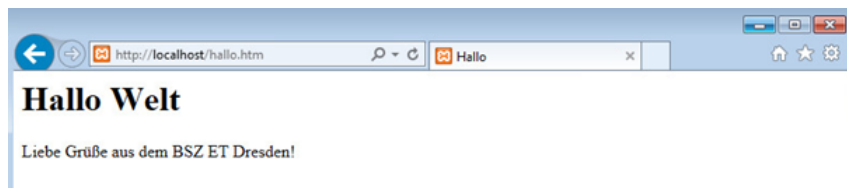


Abbildung 20: lokaler Aufruf von hallo.htm

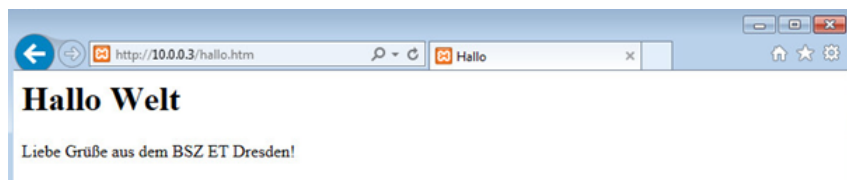


Abbildung 21: Aufruf von hallo.htm per IP

2.1.4 Installieren von Wireshark

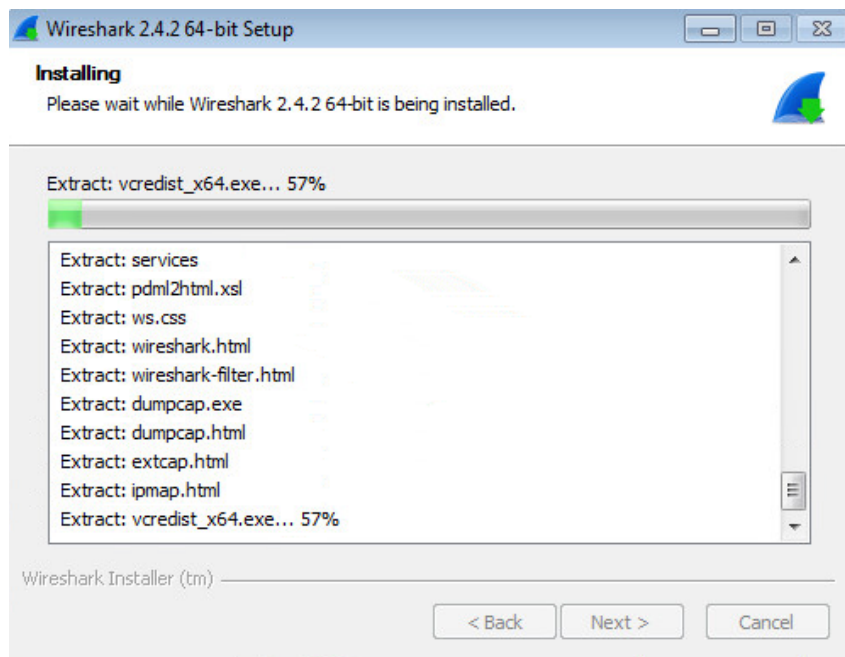


Abbildung 22: Installation von Wireshark

2.2 Aufgaben

2.2.1 Aufzeichnung und Analyse der ICMP requests und replys

2.2.1.1 Falsches Markieren von Bestandteilen der Paketinhalten

Request

0000	00 21 9b 7d 50 ae 00 21 9b 66 ba 20 08 00 45 00	..!}P...f. ..E.
0010	00 3c 4a 9b 00 00 80 01 dc 21 0a 00 00 03 0a 00	.<J.....!.....
0020	00 02 08 00 4d 2f 00 01 00 2c 61 62 63 64 65 66	...M/...,abcdef
0030	67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76	ghijklmnopqrstuv
0040	77 61 62 63 64 65 66 67 68 69	wabcdefghi

Legende:

Ethernet II

IPv4

Payload

Abbildung 23: Wireshark Mitschnitt von ping request

Reply

0000	00 21 9b 66 ba 20 00 21 9b 7d 50 ae 08 00 45 00	..f. ..!}P...E.
0010	00 3c 7f 4a 00 00 80 01 a7 72 0a 00 00 02 0a 00	.<J.....r.....
0020	00 03 00 00 55 2f 00 01 00 2c 61 62 63 64 65 66	...U/...,abcdef
0030	67 68 69 6a 6b 6c 6d 6e 6f 70 71 72 73 74 75 76	ghijklmnopqrstuv
0040	77 61 62 63 64 65 66 67 68 69	wabcdefghi

Legende:

Ethernet II

IPv4

Payload

Abbildung 24: Wireshark Mitschnitt von ping reply

2.2.1.2 Ermittlung der WS1-IP und des dazugehörigen Hexadezimalcode im IP-Header

IP Header

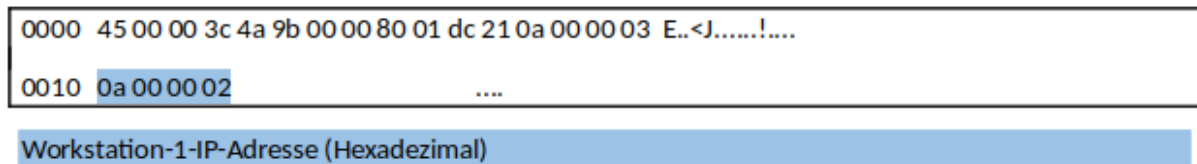


Abbildung 25: IP Header

2.2.1.3 Bestimmung der MAC und des NIC

NIC WS 1: 00-21-9B-7D-50-AE

NIC WS 2: 00-21-9B-66-BA-20

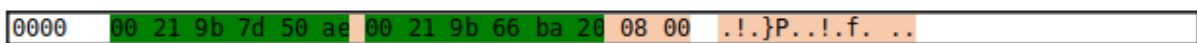
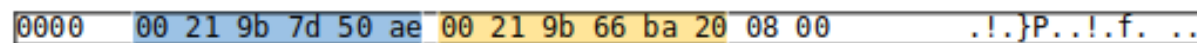


Abbildung 26: Wireshark Mitschnitt von ping request

2.3 Aufzeichnung der Protokollübertragung von hallo.htm zur WS2

2.3.0.1 Header und Payload jeder Protokollschicht und Zuordnung zu OSI Schichten

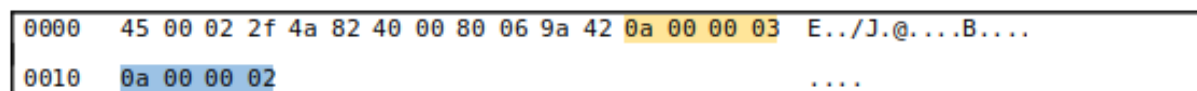
HTTP-Request



MAC-Adressen: Destination: 00:21:9b:7d:50:ae, Source: 00:21:0b:66:ba:20

Größe: 14 Byte

Abbildung 27: Ethernet II (Schicht 2)



IPv4-Adressen: Source: 10.0.0.3, Destination: 10.0.0.2

Größe: 20 Byte

Abbildung 28: IPv4 (Schicht 3)

0000	04 1d 00 50 93 5b 58 44 93 63 6d c9 50 18 01 00	...P.[XD.cm.P...
0010	aa 8e 00 00

Port: Source: 1053, Destination: 80

Größe: 20 Byte

Abbildung 29: TCP (Schicht 4)

```

0000 47 45 54 20 2f 68 61 6c 6c 6f 2e 68 74 6d 20 48 GET /hallo.htm H
0010 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 31 TTP/1.1..Host: 1
0020 30 2e 30 2e 30 2e 32 0d 0a 43 6f 6e 6e 65 63 74 0.0.0.2..Connect
0030 69 6f 6e 3a 20 6b 65 65 70 2d 61 6c 69 76 65 0d ion: keep-alive.
0040 0a 43 61 63 68 65 2d 43 6f 6e 74 72 6f 6c 3a 20 .Cache-Control:
0050 6d 61 78 2d 61 67 65 3d 30 0d 0a 55 70 67 72 61 max-age=0..Upgra
0060 64 65 2d 49 6e 73 65 63 75 72 65 2d 52 65 71 75 de-Insecure-Requ
0070 65 73 74 73 3a 20 31 0d 0a 55 73 65 72 2d 41 67 ests: 1..User-Ag
0080 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 ent: Mozilla/5.0
0090 20 28 57 69 6e 64 6f 77 73 20 4e 54 20 36 2e 33 (Windows NT 6.3
00a0 3b 20 57 69 6e 36 34 3b 20 78 36 34 29 20 41 70 ; Win64; x64) Ap
00b0 70 6c 65 57 65 62 4b 69 74 2f 35 33 37 2e 33 36 pleWebKit/537.36
00c0 20 28 4b 48 54 4d 4c 2c 20 6c 69 6b 65 20 47 65 (KHTML, like Ge
00d0 63 6b 6f 29 20 43 68 72 6f 6d 65 2f 36 31 2e 30 cko) Chrome/61.0
00e0 2e 33 31 36 33 2e 31 30 30 20 53 61 66 61 72 69 .3163.100 Safari
00f0 2f 35 33 37 2e 33 36 0d 0a 41 63 63 65 70 74 3a /537.36..Accept:
0100 20 74 65 78 74 2f 68 74 6d 6c 2c 61 70 70 6c 69 text/html,appli
0110 63 61 74 69 6f 6e 2f 78 68 74 6d 6c 2b 78 6d 6c cation/xhtml+xml
0120 2c 61 70 70 6c 69 63 61 74 69 6f 6e 2f 78 6d 6c ,application/xml
0130 3b 71 3d 30 2e 39 2c 69 6d 61 67 65 2f 77 65 62 ;q=0.9,image/web
0140 70 2c 69 6d 61 67 65 2f 61 70 6e 67 2c 2a 2f 2a p,image/apng,/*
0150 3b 71 3d 30 2e 38 0d 0a 41 63 63 65 70 74 2d 45 ;q=0.8..Accept-E
0160 6e 63 6f 64 69 6e 67 3a 20 67 7a 69 70 2c 20 64 ncoding: gzip, d
0170 65 66 6c 61 74 65 0d 0a 41 63 63 65 70 74 2d 4c eflate..Accept-L
0180 61 6e 67 75 61 67 65 3a 20 64 65 2d 44 45 2c 64 anguage: de-DE,d
0190 65 3b 71 3d 30 2e 38 2c 65 6e 2d 55 53 3b 71 3d e;q=0.8,en-US;q=
01a0 30 2e 36 2c 65 6e 3b 71 3d 30 2e 34 0d 0a 49 66 0.6,en;q=0.4..If
01b0 2d 4e 6f 6e 65 2d 4d 61 74 63 68 3a 20 57 2f 22 -None-Match: W/"
01c0 38 35 2d 35 61 37 39 66 61 36 66 30 66 30 38 30 85-5a79fa6f0f080
01d0 22 0d 0a 49 66 2d 4d 6f 64 69 66 69 65 64 2d 53 "..If-Modified-S
01e0 69 6e 63 65 3a 20 54 68 75 2c 20 30 32 20 4e 6f ince: Thu, 02 No
01f0 76 20 32 30 31 37 20 30 38 3a 35 30 3a 32 37 20 v 2017 08:50:27
0200 47 4d 54 0d 0a 0d 0a GMT....

```

Größe: 519 Byte

Abbildung 30: HTTP

HTTP-REsponse

0000	00 21 9b 66 ba 20 00 21 9b 7d 50 ae 08 00	..!.f..!.})P...
------	---	-----------------

MAC-Adressen: Source: 00:21:9b:7d:50:ae, Destination: 00:21:0b:66:ba:20

Größe: 14 Byte

Abbildung 31: Ethernet II (Schicht 2)

0000	45 00 01 e4 7f 20 40 00 80 06 65 ef 0a 00 00 02	E....@...e....
0010	0a 00 00 03

IPv4-Adressen: Destination: 10.0.0.2, Source: 10.0.0.3

Größe: 20 Byte

Abbildung 32: IPv4 (Schicht 3)

0000	00 50 04 1d 93 63 6d c9 93 5b 5a 4b 50 18 01 00	.P...cm..[ZKP...
0010	fe 3e 00 00	.>..

Port: Destination: 80, Source: 1053

Größe: 20 Byte

Abbildung 33: TCP (Schicht 4)

```

0000 48 54 54 50 2f 31 2e 31 20 32 30 30 20 4f 4b 0d HTTP/1.1 200 OK.
0010 0a 44 61 74 65 3a 20 54 68 75 2c 20 30 32 20 4e .Date: Thu, 02 N
0020 6f 76 20 32 30 31 37 20 30 38 3a 35 30 3a 33 37 ov 2017 08:50:37
0030 20 47 4d 54 0d 0a 53 65 72 76 65 72 3a 20 41 70 GMT..Server: Ap
0040 61 63 68 65 2f 32 2e 34 2e 31 37 20 28 57 69 6e ache/2.4.17 (Win
0050 33 32 29 20 4f 70 65 6e 53 53 4c 2f 31 2e 30 2e 32) OpenSSL/1.0.
0060 32 64 20 50 48 50 2f 35 2e 36 2e 31 35 0d 0a 4c 2d PHP/5.6.15..L
0070 61 73 74 2d 4d 6f 64 69 66 69 65 64 3a 20 54 68 ast-Modified: Th
0080 75 2c 20 30 32 20 4e 6f 76 20 32 30 31 37 20 30 u, 02 Nov 2017 0
0090 38 3a 35 30 3a 33 37 20 47 4d 54 0d 0a 45 54 61 8:50:37 GMT..ETa
00a0 67 3a 20 57 2f 22 38 35 2d 35 61 37 39 66 61 36 g: W/"85-5a79fa6
00b0 66 30 66 30 38 30 22 0d 0a 41 63 63 65 70 74 2d f0f080"..Accept-
00c0 52 61 6e 67 65 73 3a 20 62 79 74 65 73 0d 0a 43 Ranges: bytes..C
00d0 6f 6e 74 65 6e 74 2d 4c 65 6e 67 74 68 3a 20 31 ontent-Length: 1
00e0 33 33 0d 0a 4b 65 65 70 2d 41 6c 69 76 65 3a 20 33..Keep-Alive:
00f0 74 69 6d 65 6f 75 74 3d 35 2c 20 6d 61 78 3d 31 timeout=5, max=1
0100 30 30 0d 0a 43 6f 6e 6e 65 63 74 69 6f 6e 3a 20 00..Connection:
0110 4b 65 65 70 2d 41 6c 69 76 65 0d 0a 43 6f 6e 74 Keep-Alive..Cont
0120 65 6e 74 2d 54 79 70 65 3a 20 74 65 78 74 2f 68 ent-Type: text/h
0130 74 6d 6c 0d 0a 0d 0a tml....

```

Größe: 311 Byte

Abbildung 34: HTTP

```

0000 3c 68 74 6d 6c 3e 0a 3c 68 65 61 64 3e 0a 3c 74 <html>.<head>.<t
0010 69 74 6c 65 3e 48 61 6c 6c 6f 3c 2f 74 69 74 6c itle>Hallo</titl
0020 65 3e 0a 3c 2f 68 65 61 64 3e 0a 3c 62 6f 64 79 e>.</head>.<body
0030 3e 0a 3c 68 31 3e 48 61 6c 6c 6f 20 57 65 6c 74 >.<h1>Hallo Welt
0040 3c 2f 68 31 3e 0a 4c 69 65 62 65 20 47 72 26 75 </h1>.<h2>Liebe Gr&u
0050 75 6d 6c 3b 26 73 7a 6c 69 67 3b 65 20 61 75 73 uml;&szlig;e aus
0060 20 64 65 6d 20 42 53 5a 20 45 54 20 44 72 65 73 dem BSZ ET Dres
0070 64 65 6e 21 0a 3c 2f 62 6f 64 79 3e 0a 3c 2f 68 den!.</body>.</h
0080 74 6d 6c 3e 0a tml>.

```

Größe: 133 Byte

Abbildung 35: Payload

2.3.0.2 Verhältnis Payload zur Paketgröße nach DoD-Modell

Request

Schicht	Größe in Byte	Protokoll	Verhältnis Payload zu Paketgröße
Application	519	Data (HTTP)	90,58 %
Host-to-Host	20	Header (TCP)	-
Host-To-Host Gesamt	539	Data (HTTP + Payload + TCP)	94,07 %
Internet	20	Header (IPv4)	-
Internet Gesamt	559	Data (HTTP + Payload + TCP + IPv4)	97,56 %
Network-Access	14	Header (Ethernet)	-
Paket Gesamt	573	Data (gesamt)	100 %

Tabelle 3: Payload des Requests

Response

Schicht Größe in Byte	Protokoll	Verhältnis Payload zu Paketgröße	
Application	444	Data (HTTP + Payload)	89,14 %
Host-to-Host	20	Header (TCP)	-
Host-To-Host Gesamt	464	Data (HTTP + Payload + TCP)	93,17 %
Internet	20	Header (IPv4)	-
Internet Gesamt	484	Data (HTTP + Payload + TCP + IPv4)	97,19 %
Network-Access	14	Header (Ethernet)	-
Paket Gesamt	498	Data (gesamt)	100 %

Tabelle 4: Payload der Response

2.3.0.3 Anteil der Nutzdaten zum Frame für den Request und den Response

Request

Frame: 573 Bytes

HTML: 0 Bytes

Anteil: 0 %

Response

Frame: 498 Bytes

HTML: 133 Bytes

Anteil: 26,7 %

2.3.1 Umstellung von IPv4 auf IPv6

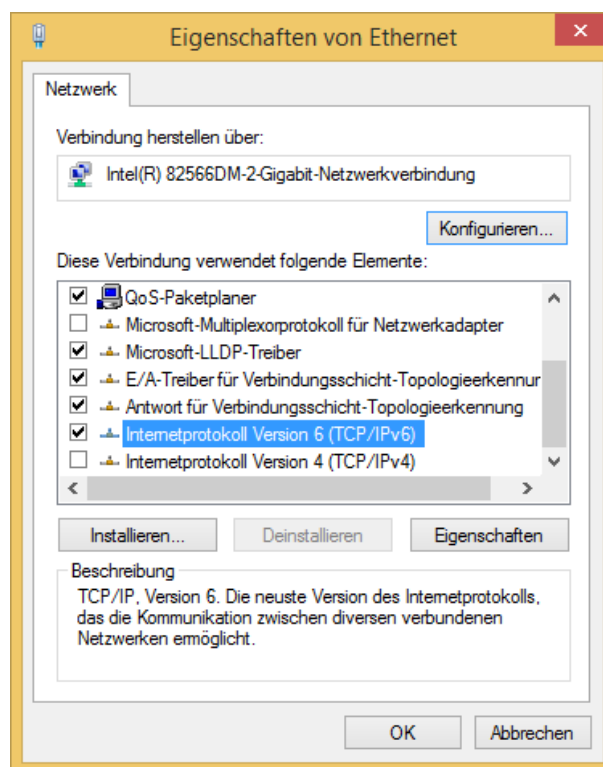


Abbildung 36: Umstellung von IPv4 auf IPv6

2.3.1.1 Ausführung des Befehls `ipv6 if`

```
C:\Users\Admin>netsh
netsh>int ipv6 show int
```

Idx	Met	MTU	State	Name
1	50	4294967295	connected	Loopback Pseudo-Interface 1
3	10	1500	connected	Ethernet

```
netsh>
```

Abbildung 37: Ausführung des Befehls `ipv6 if`

2.3.1.2 Testen der Verbindung mit ping, Ermittlung des korrektem Befehls

```
C:\Users\Admin>ping fe80::67:3fd:f529:369

Ping wird ausgeführt für fe80::67:3fd:f529:369 mit 32 Bytes Daten:
Antwort von fe80::67:3fd:f529:369: Zeit<1ms
Antwort von fe80::67:3fd:f529:369: Zeit<1ms
Antwort von fe80::67:3fd:f529:369: Zeit<1ms
Antwort von fe80::67:3fd:f529:369: Zeit<1ms

Ping-Statistik für fe80::67:3fd:f529:369:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0
    (0% Verlust),
    Ca. Zeitangaben in Millisek.:
        Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms
```

Abbildung 38: Ausführen eines Pings auf eine IPv6

2.3.1.3 Testen der Aufzeichnung der Kommunikationsbeziehung mit Wireshark und die Unterschiede zu 2.2.1.1

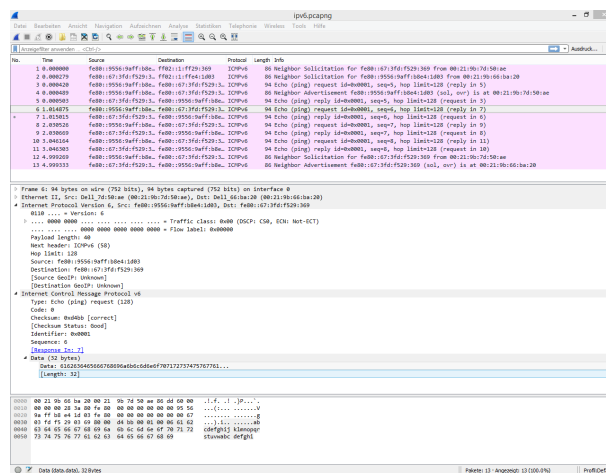


Abbildung 39: Mitschnitt via Wireshark

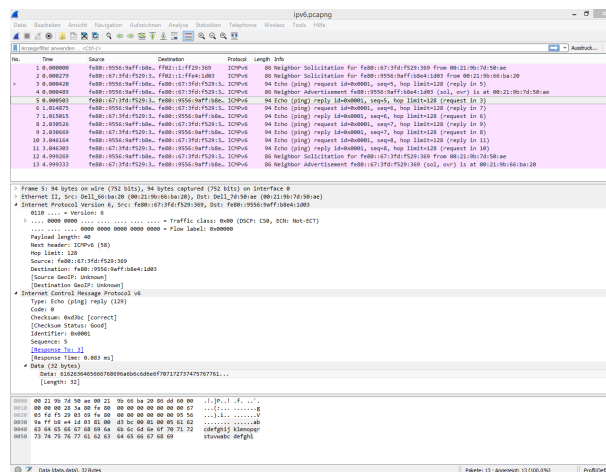


Abbildung 40: Mitschnitt via Wireshark

Unterschiede zu 2.2.1.1

- Bei IPv6 ist IP-Header ist größer, da dieser die längeren IPv6 Adressen enthält
- Checksum im Payload hat sich verändert

- Bei IPv4 gibt es zwei Identifier (BE & LE), bei IPv6 nur einen
- Bei IPv4 gibt es zwei Sequence numbers (BE & LE), bei IPv6 nur eine sequence (number)
- Bei IPv4 gibt es eine "Time to live", bei IPv6 heißt diese nun "Hop limit"

2.3.1.4 Testen einer Ordnerfreigabe zur WS2

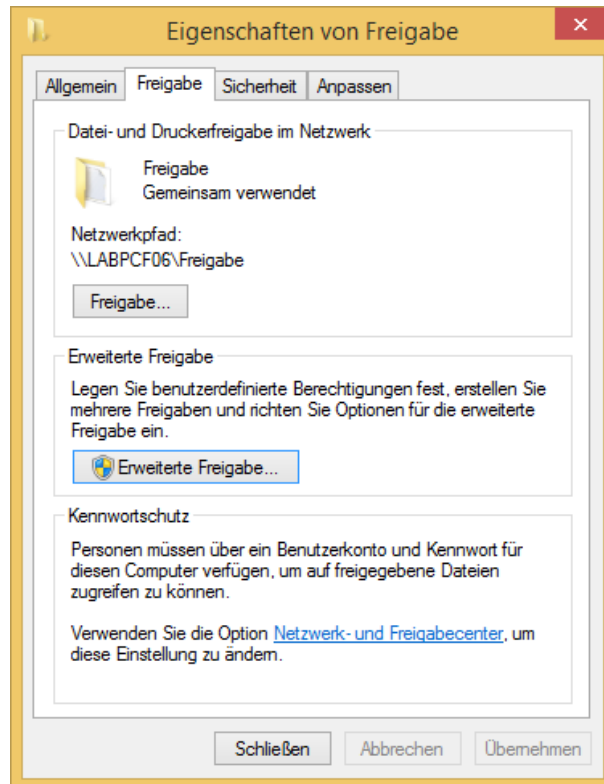


Abbildung 41: Eigenschaften der Freigabe

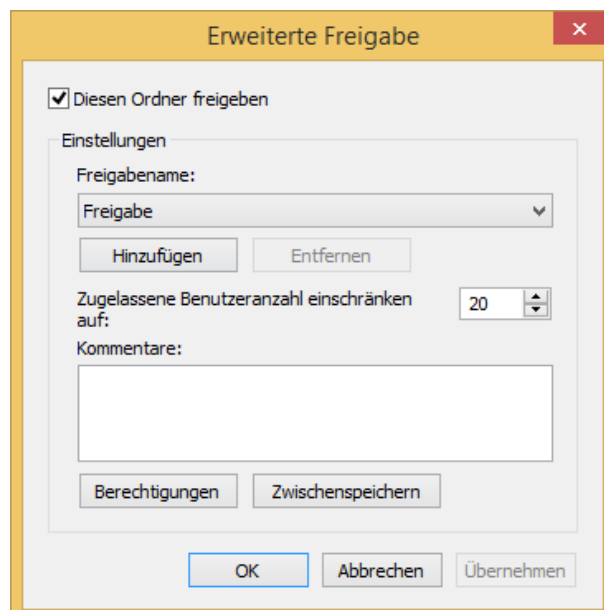


Abbildung 42: Erweiterte Freigabe

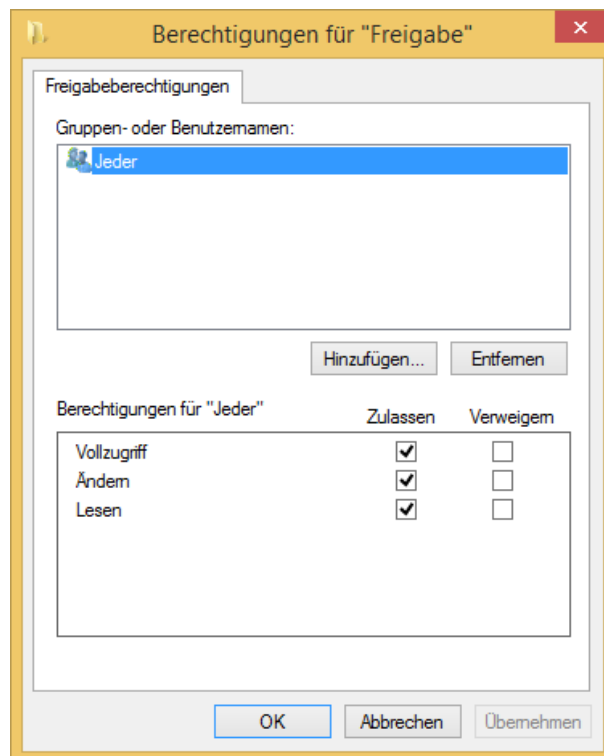


Abbildung 43: Berechtigungen für die Freigabe

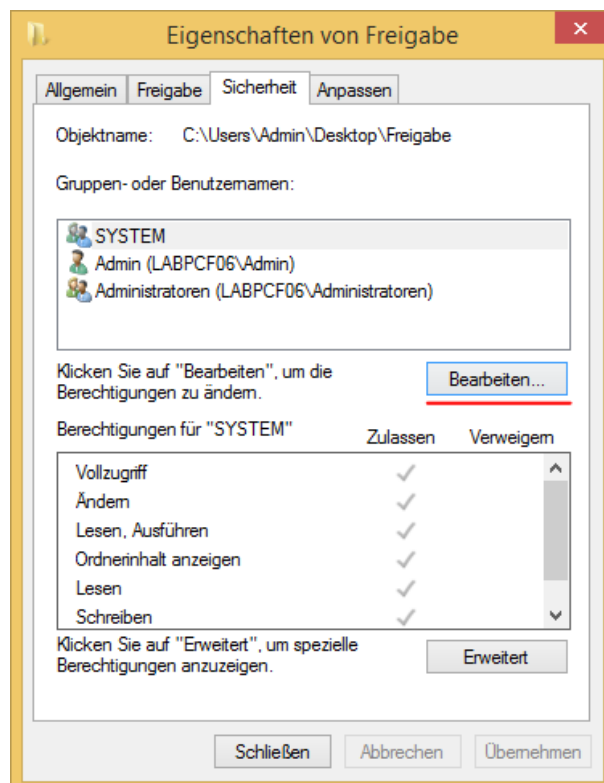


Abbildung 44: Bearbeiten der Sicherheit der Freigabe

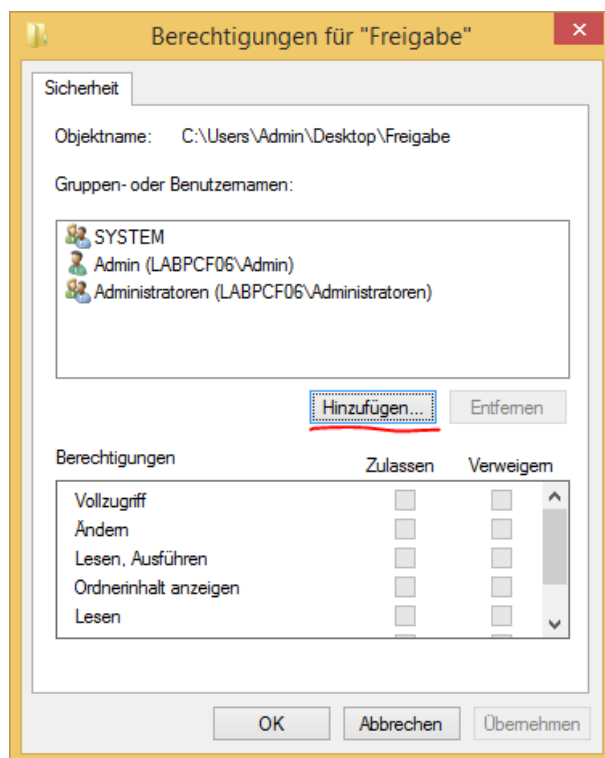


Abbildung 45: Berechtigungen hinzufügen

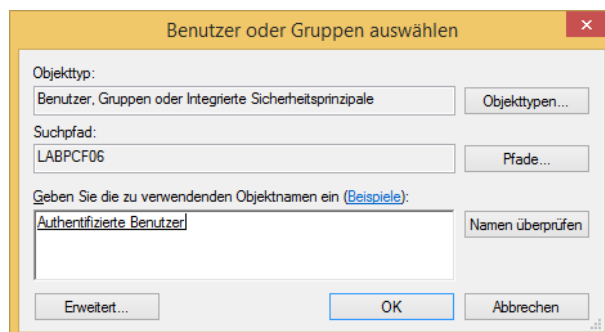


Abbildung 46: Einstellen der Bruntzergruppen

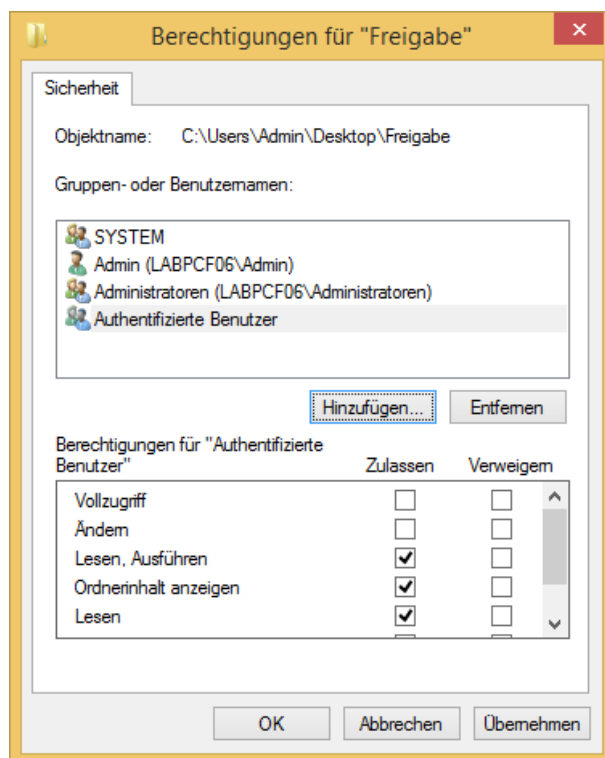


Abbildung 47: Berechtigungen für Freigabe



Abbildung 48: Anzeige des Freigegebenen Dokuments auf WS1

3 Abbildungsverzeichnis

1	Osi Modell	
	https://www.der-wirtschaftsingenieur.de/bilder/it/OSI-Modell3.PNG	
	3.1.2020 14:30	5
2	Ping über einen Hub	
	https://www.tinohempel.de/info/info/netze/osi.htm	
	8.1.2020 13:00	5
3	Ping über einen Switch	
	https://www.tinohempel.de/info/info/netze/osi.htm	
	8.1.2020 13:00	6
4	Ping über einen Router	
	https://www.tinohempel.de/info/info/netze/osi.htm	
	8.1.2020 13:00	6
5	IMCP Aufbau	
	https://en.wikipedia.org/wiki/Ping_networking_utility#ICMP_packet	
	2.1.2020 15:00	8
6	DFÜ Modell	
	eigene Zusammenstellung	
	8
7	Skizze eines Protokollstapels eines HTTP Requests	
	eigene Zusammenstellung	
	9
8	HTTP Response Header	
	http://www.coder-welten.de/glossar/request-und-response-18.html	
	07.01.2020 15:00	9
9	HTTP Request Header	
	http://www.coder-welten.de/glossar/request-und-response-18.html	
	07.01.2020 15:00	9
10	TCP Header	
	https://upload.wikimedia.org/wikipedia/commons/thumb/f/fd/TCP_Header.svg/1024px-TCP_Header.svg.png	
	2.1.2020 14:00	10
11	IPv4 Header	
	https://de.wikipedia.org/wiki/IPv4#Header-Format	
	2.1.2020 14:30	10
12	IPv6 Header	
	https://upload.wikimedia.org/wikipedia/commons/thumb/c/cd/IPv6_Header.svg/1280px-IPv6_Header.svg.png	
	2.1.2020 14:45	10

13	Installation mit XAMPP Wizard	
	Screenshot	
	15
14	Installationsverzeichnis	
	Screenshot	
	15
15	Konfiguration (Apache)	
	Screenshot	
	16
16	Ping von Workstation 2 zu Workstation 1	
	Screenshot	
	16
17	Ping von Workstation 1 zu Workstation 2	
	Screenshot	
	17
18	hallo.htm nach C:\xampp\htdocs kopieren	
	Screenshot	
	17
19	Apache in XAMPP Starten	
	Screenshot	
	18
20	lokaler Aufruf von hallo.htm	
	Screenshot	
	18
21	Aufruf von hallo.htm per IP	
	Screenshot	
	18
22	Installation von Wireshark	
	Screenshot	
	19
23	Wireshark Mitschnitt von ping request	
	eigene Zusammenstellung	
	20
24	Wireshark Mitschnitt von ping reply	
	eigene Zusammenstellung	
	20
25	IP Header	
	eigene Zusammenstellung	
	21
26	Wireshark Mitschnitt von ping request	
	eigene Zusammenstellung	
	21

27	Ethernet II (Schicht 2)	
	eigene Zusammenstellung	
	21
28	IPv4 (Schicht 3)	
	eigene Zusammenstellung	
	21
29	TCP (Schicht 4)	
	eigene Zusammenstellung	
	22
30	HTTP	
	eigene Zusammenstellung	
	23
31	Ethernet II (Schicht 2)	
	eigene Zusammenstellung	
	24
32	IPv4 (Schicht 3)	
	eigene Zusammenstellung	
	24
33	TCP (Schicht 4)	
	eigene Zusammenstellung	
	24
34	HTTP	
	eigene Zusammenstellung	
	25
35	Payload	
	eigene Zusammenstellung	
	25
36	Umstellung von IPv4 auf IPv6	
	Screenshot	
	30
37	Ausführung des Befehls ipv6 if	
	Screenshot	
	30
38	Ausführen eines Pings auf eine IPv6	
	Screenshot	
	31
39	Mitschnitt via Wireshark	
	Screenshot	
	31
40	Mitschnitt via Wireshark	
	Screenshot	
	31

41	Eigenschaften der Freigabe	
	Screenshot	
	32
42	Erweiterte Freigabe	
	Screenshot	
	32
43	Berechtigungen für die Freigabe	
	Screenshot	
	33
44	Bearbeiten der Sicherheit der Freigabe	
	Screenshot	
	33
45	Berechtigungen hinzufügen	
	Screenshot	
	34
46	Einstellen der Bruntzergruppen	
	Screenshot	
	34
47	Berechtigungen für Freigabe	
	Screenshot	
	35
48	Anzeige des Freigegebenen Dokuments auf WS1	
	Screenshot	
	35

4 Tabellenverzeichnis

1 Osi Modell	4
2 Terminal Befehle	13
3 Payload des Requests	27
4 Payload der Response	29

5 Quellen

- <https://tools.ietf.org/html/rfc2616#section-6.2> 2.1.2020 13:00
- <https://de.wikipedia.org/wiki/IPv6#Header-Format> 2.1.2020 14:45
- <https://www.lancom-systems.de/docs/LCOS/referenzhandbuch/topics/aa1066622.html> 3.1.2020 11:45
- [https://docs.microsoft.com/en-us/previous-versions/windows/embedded/aa450452\(v%3Dmsdn.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/embedded/aa450452(v%3Dmsdn.10)) 3.1.2020 13:00
- [https://docs.microsoft.com/de-de/previous-versions/windows/embedded/aa450443\(v=msdn.10\)](https://docs.microsoft.com/de-de/previous-versions/windows/embedded/aa450443(v=msdn.10)) 3.1.2020 13:00
- <https://de.wikipedia.org/wiki/Loopback> 3.1.2020 13:15
- https://de.wikipedia.org/wiki/Time_to_Live 3.1.2020 13:30
- <https://de.wikipedia.org/wiki/OSI-Modell> 3.1.2020 14:30
- <https://www.datenschutzbeauftragter-info.de/osi-modell-so-kommunizieren-rechner/> 6.1.2020 11:00
- <https://www.black-box.de/de-de/page/26203/Information/Technische-Ressourcen/black-box-erklaert/lan/Layer-2,3-und-4-Switching> 6.1.2020 11:00
- [https://en.wikipedia.org/wiki/Bridging_\(networking\)](https://en.wikipedia.org/wiki/Bridging_(networking)) 6.1.2020 14:30
- <https://www.cpcstech.com/routers-bridges-information.htm> 8.1.2020 14:00
- <https://www.sciencedirect.com/topics/computer-science/gateway-router> 8.1.2020 15:00
- [https://de.m.wikipedia.org/wiki/Metrik_\(Netzwerk\)](https://de.m.wikipedia.org/wiki/Metrik_(Netzwerk)) 12.06.2020 10:00

6 Glossar

ICMP	Internet Control Message Protocol - ein Protokoll mit dem überprüft werden kann, ob ein Host im Netzwerk aktiv ist
TTL	time to live, gibt beim ICMP die verbleibende maximale Lebenszeit im Netzwerk in Sekunden an
MTU	Maximum Transmission Unit, maximale Größe unfragmentierter Datenpakete
loopback	Schleifenschaltung mit Nachrichten- oder Informationskanal in dem Sender und Empfänger identisch sind
ping	ein Konsolenbefehl, welcher unter fast allen Betriebssystemen funktioniert und ICMP Protokolls Pakete zu interfaces sendet und zurückbekommt, ob diese aktiv sind oder nicht
interface	zu deutsch "Schnittstelle", hier in dieser Dokumentation wird zumeist mit interface eine virtuelle oder physische Schnittstelle zwischen Netzen gemeint
header	<p>Bei Rechnernetzwerken besitzt jedes von einem Rechner versandte Datenpaket einen Header, der Daten über den Absender, Empfänger, Typ und Lebensdauer des Datenpakets enthält.</p> <p>Beim Hypertext Transfer Protocol (HTTP) werden über den Header HTTP-Cookies und Informationen wie Dateigröße übertragen</p>
Payload	Nutzdaten, die keine Steuer- oder Protokollinformationen enthalten. Nutzdaten sind unter anderem Sprache, Text, Zeichen, Bilder und Töne.
OSI Schichtmodell	Modell, welches die Ebenen die ein Netzwerk ausmachen beschreibt
DoD Schichtmodell	Modell welches Datenübertragungen darstellt
link-layer adress	feste Adressen wie die MAC Adresse