***Übungsblatt 1 Lösungen***

**✅ 1.1 Was ist ein Sniffer und wozu wird er eingesetzt?**

Ein **Sniffer** ist ein Programm oder Gerät, das **Netzwerkpakete mitschneidet und analysiert**.  
Es wird eingesetzt für:

* Netzwerkdiagnose und Fehleranalyse
* Sicherheitsüberwachung (z. B. verdächtiger Datenverkehr)
* Performance-Analyse
* Protokoll-Inspektion

Beispiel: **Wireshark**

### ✅ ****1.2 Was ist ein Mirror Port (Port Mirroring)?****

Ein **Mirror Port** ist ein spezieller Port auf einem Switch, an den **Kopien des Datenverkehrs anderer Ports** weitergeleitet werden.  
Ziel: Der gesamte Traffic eines Ports (oder mehrerer) kann dort von einem Sniffer überwacht werden, **ohne den normalen Verkehr zu beeinflussen**.

### ✅ ****1.3 Datenrate-Berechnung****

#### **Ein Port mit 100 Mbit/s duplex**:

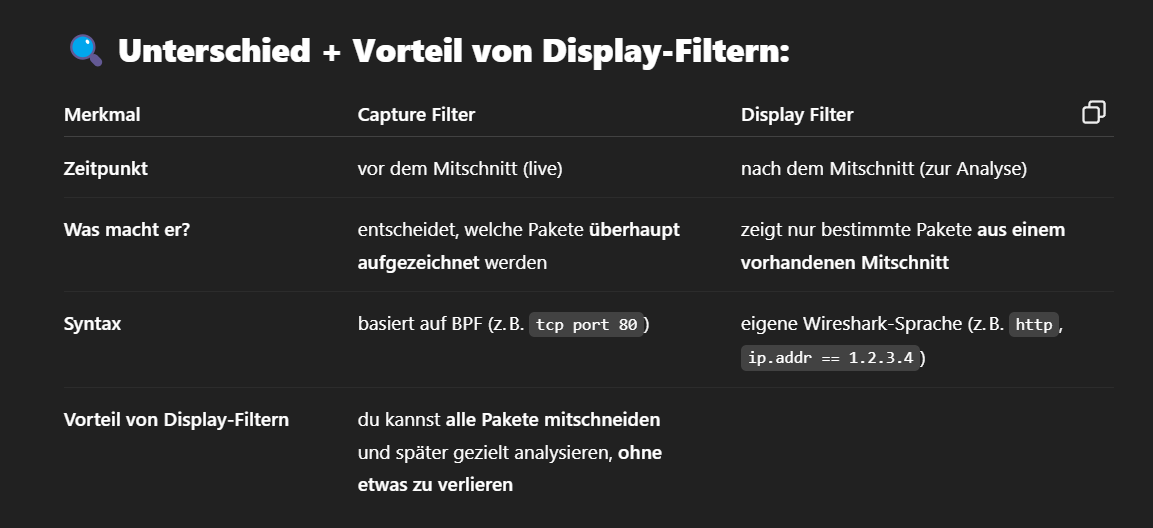
* **100 Mbit/s empfangen + 100 Mbit/s senden = 200 Mbit/s Gesamtdatenrate**
* Der Mirror-Port muss **mindestens 200 Mbit/s** verarbeiten können.

#### **Fünf Ports mit je 100 Mbit/s duplex**:

* Jeder Port = 200 Mbit/s → 5 × 200 Mbit/s = **1000 Mbit/s**
* **Mindestens 1 Gbit/s** (Gigabit-Port!) nötig für den Mirror-Port

### *2.1 Starten Sie Wireshark und sehen Sie sich die Capture Options an. Was ist ein Interface?* 📌 In Wireshark bedeutet Interface:

**„Wo soll Wireshark den Verkehr mitschneiden?“**  
Wenn du z. B. über WLAN im Internet surfst, musst du das **WLAN-Interface** auswählen.



***2.7 Überlegen Sie, welche (legalen und illegalen) Konsequenzen durch diese Transparenz entstehen. Welche technischen Herausforderungen stellen sich beim Sniffen? Diskutieren Sie mögliche Abwehrmaßnahmen.***

## 🔎 **1. Konsequenzen der Transparenz durch Sniffing**

### ✅ ****Legale Folgen / sinnvolle Nutzung:****

* **Netzwerkdiagnose und Fehlersuche**
* **Performance-Analyse** (z. B. wo ist der Flaschenhals?)
* **Sicherheitsüberwachung** (z. B. IDS-Systeme)
* **Analyse von Protokollen** in Ausbildung oder Entwicklung

### ❌ ****Illegale / kritische Nutzung:****

* **Abhören von Passwörtern** oder Sessions in unverschlüsseltem Verkehr (z. B. HTTP, FTP)
* **Auslesen von personenbezogenen Daten**
* **Man-in-the-Middle-Angriffe**
* **Industriespionage** oder Datenschutzverletzung (DSGVO!)

## 🧱 **2. Technische Herausforderungen beim Sniffing**

* **Switched Networks**: Switches senden Pakete nur an den Zielport → Sniffer sieht nichts ohne **Port Mirroring** oder **ARP-Spoofing**
* **Verschlüsselter Traffic (TLS/HTTPS)**: Inhalte sind **nicht lesbar**, außer mit Zugriff auf private Schlüssel oder durch MITM
* **Datenmenge**: hoher Traffic = riesige Datenmengen → schwer zu filtern/analysefähig zu halten
* **Timing**: manche Protokolle sind kurzlebig oder paketabhängig → schwer rekonstruierbar



**✅ 3.1 Wie ist das Internet aufgebaut? Welche Netzkomponente koppelt individuelle Teilnetze und wie werden Systeme adressiert?**

* Das Internet besteht aus vielen **autonomen Teilnetzen** (LANs, ISP-Netze, Uni-Netze, etc.)
* **Router** verbinden diese Netze miteinander
* Systeme werden über **IP-Adressen** adressiert (z. B. 192.0.2.1)
* Jede IP-Adresse ist **weltweit eindeutig** (bei IPv4 oder IPv6)

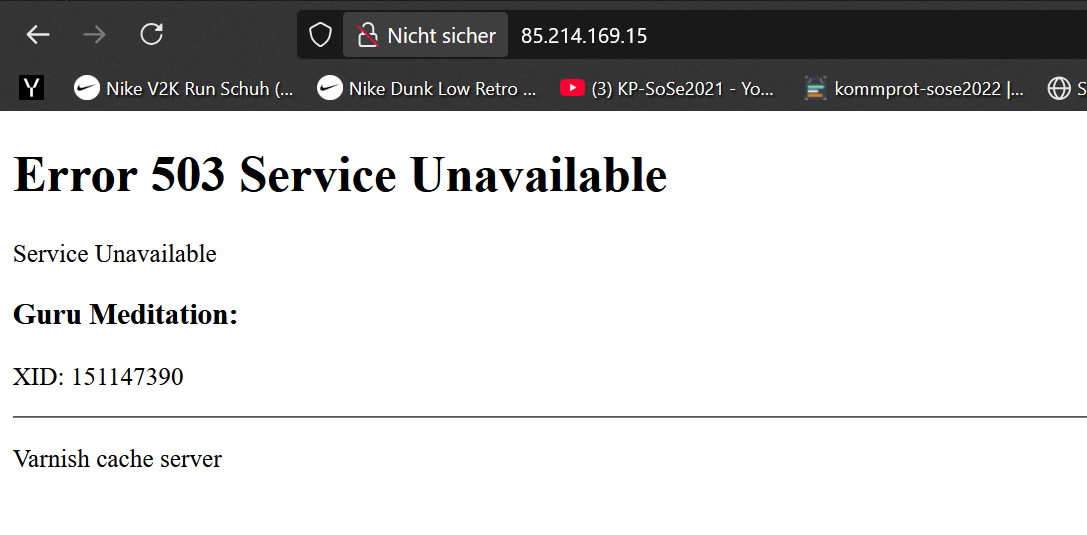
**✅ 3.2 Warum werden im Internet z. B. hs-fulda.de statt IP-Adressen verwendet?**

* IP-Adressen sind **schwer merkbar**, Domains sind **menschlich lesbar**
* Der **DNS (Domain Name System)** übersetzt Namen wie hs-fulda.de in IPs
* Vorteil: IP-Adresse kann sich ändern, der Name bleibt gleich

**3.4 In manchen Fällen gibt der DNS-Server mehrere IP-Adressen zurück – z. B.** IPv4 + IPv6**, oder** mehrere Server zur Lastverteilung**.  
Im Fall von www.hs-fulda.de ist aktuell** nur eine IPv4-Adresse eingetragen**, deshalb wird auch nur diese angezeigt**

**3.5**

Der Webserver antwortet auf die IP-Adresse, aber zeigt einen **Fehler 503**, da der Server ohne den richtigen **Host-Header** (z. B. www.hs-fulda.de) nicht weiß, welche Webseite geladen werden soll.  
Dieses Verhalten ist typisch bei Servern mit **virtuellem Hosting** und vorgeschaltetem **Cache (Varnish)**.



3.6.

Beim Aufruf der IP-Adresse von www.heise.de (193.99.144.85) zeigt der Server eine **funktionierende Antwortseite** an.  
Diese verweist auf https://www.heise.de/ping und zeigt, dass der Server **auch ohne Domainname** antworten kann — ein bewusst konfigurierter HTTP-Endpunkt zur Verfügbarkeitsprüfung.

4.1 Wie ist der DNS-Namensraum aufgebaut? Wie werden DNS-Informationen verwaltet?

### 🔧 ****Aufbau des DNS-Namensraums****

Der **DNS-Namensraum** ist hierarchisch und baumartig aufgebaut:

.

/|\

Root-Zone

/ | \

com org de ...

/ \

google hs-fulda

\ \

www www

 **Wurzel (Root)**: ganz oben steht die Root-Zone, dargestellt durch einen Punkt (.). Sie ist die Ausgangsbasis aller Domains.

 **Top-Level-Domains (TLDs)**: darunter befinden sich Domains wie .com, .org, .de, usw.

 **Second-Level-Domains**: z. B. google.com, hs-fulda.de

 **Subdomains**: z. B. www.google.com, webmail.hs-fulda.de

Jede Ebene ist durch Punkte getrennt und von **rechts nach links** zu lesen.

### 📦 ****Verwaltung von DNS-Informationen****

DNS ist **dezentral** organisiert, aber klar aufgeteilt:

#### 1. **Root-Nameserver**

* Verwalten die Adressen der autoritativen Nameserver für alle TLDs.
* Es gibt 13 Root-Server-Gruppen weltweit (A–M), aber viele Kopien durch Anycast.

#### 2. **TLD-Nameserver**

* Verwalten Domains unter einer bestimmten TLD (z. B. .de, .com).
* Beispiel: Denic verwaltet .de.

#### 3. **Authoritative Nameserver**

* Zuständig für konkrete Domains (z. B. hs-fulda.de).
* Speichern z. B. A-Records, MX-Records etc.

#### 4. **Lokaler Resolver (oft beim ISP oder im Router)**

* Wird vom Client kontaktiert.
* Fragt rekursiv andere Server ab und speichert Ergebnisse im Cache.

### 🔁 Beispiel-Ablauf: Auflösung von www.hs-fulda.de

1. Client fragt lokalen DNS-Resolver (z. B. im Router)
2. Resolver fragt Root-Server → bekommt .de-TLD-Server
3. Resolver fragt .de-Server → bekommt hs-fulda.de-Nameserver
4. Resolver fragt diesen → bekommt IP für www.hs-fulda.de
5. Antwort wird an Client zurückgegeben

### ****Strategien zur Auflösung von Hostnamen (Name Resolution Strategies)****

Es gibt **drei Hauptstrategien**, um einen Hostnamen einer IP-Adresse zuzuordnen:

## 1. **Iterative Auflösung (nicht rekursiv)**

Der **Client fragt Schritt für Schritt** nach – er erhält bei jeder Anfrage eine „Weiterleitung“.

### Ablauf:

1. Client fragt lokalen DNS-Resolver: „Was ist IP von www.hs-fulda.de?“
2. Resolver fragt Root-Server → Antwort: „Frag .de-Nameserver“
3. Resolver fragt .de-Server → Antwort: „Frag hs-fulda.de-Server“
4. Resolver fragt hs-fulda.de-Server → Antwort: „Das ist die IP von www.hs-fulda.de“
5. Client erhält IP

🡺 **Der Resolver macht alles selbst.**

## 2. **Rekursive Auflösung**

Der **Client bittet den Resolver**, die vollständige Auflösung durchzuführen.

### Ablauf:

1. Client fragt lokalen DNS-Resolver → „Gib mir IP von www.hs-fulda.de“
2. Resolver übernimmt alle Anfragen intern:
   * Root → TLD → Authoritative Server
3. Resolver gibt dem Client die endgültige IP zurück.

🡺 **Bequemer für den Client**, aber **mehr Last auf dem Resolver**.

## 3. **Lokale Auflösung (hosts-Datei / Cache)**

### Ablauf:

1. Client schaut zuerst in:
   * lokaler DNS-Cache
   * /etc/hosts (Linux/macOS) oder C:\Windows\System32\drivers\etc\hosts (Windows)
2. Falls nichts gefunden → DNS-Abfrage

🡺 Wird oft für Tests oder lokale Overrides genutzt.

5.

### 📡 ****Aufgabe des Anwendungsprotokolls DHCP (Dynamic Host Configuration Protocol)****

**DHCP** ist ein **Client-Server-Protokoll**, das in IP-Netzen verwendet wird, um Geräten **automatisch Konfigurationsdaten** zuzuweisen.

### ✅ ****Kernaufgaben von DHCP:****

1. **Zuweisung einer IP-Adresse**
   * Temporär (Lease) oder fest (reserviert)
2. **Vergabe weiterer Netzwerkinformationen:**
   * Subnetzmaske
   * Standard-Gateway
   * DNS-Server
   * (optional: Domain-Name, NTP-Server usw.)

### 🔁 ****DHCP-Ablauf (DORA-Prinzip):****

1. **D**iscover  
   Client sendet einen **Broadcast**, um einen DHCP-Server zu finden
2. **O**ffer  
   Server antwortet mit einem IP-Angebot
3. **R**equest  
   Client sendet eine Anfrage: „Ich möchte dieses Angebot annehmen“
4. **A**cknowledge  
   Server bestätigt und übergibt Konfigurationsdaten

### 🎯 ****Vorteile von DHCP:****

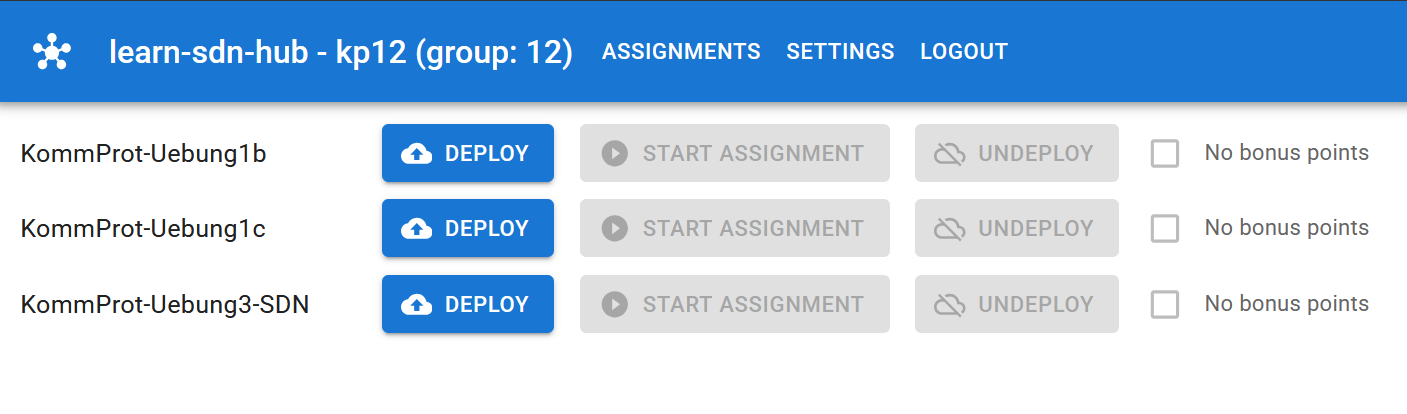
* Keine manuelle IP-Konfiguration nötig
* Zentrale Verwaltung
* Vermeidung von IP-Konflikten
* Effizient bei mobilen Geräten oder großen Netzwerken

Ohne DHCP müssten IP-Adressen **manuell** vergeben werden – das wäre **fehleranfällig und aufwendig**, besonders in großen Netzen.

**Aufgabe 7: Verbindungslose und verbindungsorientierte Kommunikationen**

Um die mit [x] gekennzeichneten Aufgaben bearbeiten zu können, müssen Sie als Gruppe unseren learn-sdn-hub benutzten. Bitte wählen Sie sich dafür in Moodle in eine Gruppe ein, die aus max. 4 Studierenden bestehen darf. Der Ablauf ist wie folgt:

1. Öffnen Sie in Ihrem Browser folgende Webseite: https://prona.informatik.hs-fulda.de/ (von zu Hause müssen Sie hierfür zunächst ins VPN der Hochschule)
2. Verwenden Sie als Username den Namen Ihrer Gruppe und als Passwort „kommprot“. Sie können das Passwort anschließend unter Einstellungen ändern.



Starten Sie für den dritten Teil unserer Übung (1c) „KommProt-Uebung1c“.

7.1 Was sind die Unterschiede zwischen einer verbindungslosen und verbindungs-orientierten Kommunikation? Welche Funktionen werden in beiden Fällen erbracht?

|  |  |  |
| --- | --- | --- |
| Merkmal | Verbindungslos (z. B. UDP) | Verbindungsorientiert (z. B. TCP) |
| Verbindungsaufbau | Kein Aufbau erforderlich | Expliziter Aufbau mittels 3-Wege-Handshake |
| Zuverlässigkeit | Keine Garantie für Zustellung, Reihenfolge oder Fehlerfreiheit | Zuverlässige Übertragung, mit Fehlerkorrektur und Bestätigung |
| Reihenfolge der Daten | Kann verloren gehen oder durcheinander geraten | Reihenfolge wird garantiert |
| Flusskontrolle | Nicht vorhanden | Ja, durch Sliding Window |
| Staukontrolle | Nicht vorhanden | Ja, mit Congestion Control (Slow Start, AIMD, etc.) |
| Overhead | Gering | Höherer Overhead durch Kontrollmechanismen |
| Anwendungsbeispiele | DNS, DHCP, VoIP, Streaming | HTTP, FTP, SMTP, E-Mail, Dateiübertragungen |

**Funktionen in beiden Fällen (gemeinsam)**:

* **Multiplexing/Demultiplexing**: Zuordnung von Datenströmen zu Anwendungsprozessen über Ports.
* **Segmentierung/Reassemblierung**: Aufteilen großer Datenmengen in kleinere Einheiten (Segmente/Datagramme), Zusammensetzen beim Empfänger.
* **Socket-Adressen**: Kombination aus IP-Adresse + Portnummer zur eindeutigen Identifikation der Verbindungspartner.

### ****Transportprotokolle für beide Kommunikationsformen****

| **Kommunikationsform** | **Transportprotokoll** | **Beschreibung** |
| --- | --- | --- |
| **Verbindungsorientiert** | **TCP** | Bietet zuverlässigen, geordneten, fehlerfreien und bidirektionalen Datentransfer |
| **Verbindungslos** | **UDP** | Minimaler Overhead, keine Garantie für Reihenfolge, Zustellung oder Integrität |

### ****Anwendungsprotokolle nach Kommunikationsform****

#### **Verbindungsorientierte Anwendungsprotokolle (über TCP)**

| **Anwendungsprotokoll** | **Beschreibung** |
| --- | --- |
| **HTTP/HTTPS** | Web-Kommunikation |
| **FTP** | Dateiübertragung |
| **SMTP/IMAP/POP3** | E-Mail Versand und Abruf |
| **SSH** | Sichere Fernzugriffe |

#### **Verbindungslose Anwendungsprotokolle (über UDP)**

| **Anwendungsprotokoll** | **Beschreibung** |
| --- | --- |
| **DNS** | Namensauflösung (meist UDP, selten TCP) |
| **DHCP** | Vergabe von IP-Adressen |
| **RTP/SIP** | Sprach-/Videoübertragung (VoIP, Streaming) |
| **SNMP** | Netzwerküberwachung |

### ****Gründe für die Wahl einer Kommunikationsform****

#### ✅ **Verbindungsorientiert (TCP)**

* **Zuverlässigkeit**: Wichtig bei Dateiübertragungen, E-Mail, Webinhalten.
* **Korrekte Reihenfolge**: Muss bei vielen Applikationen gewährleistet sein.
* **Fehlerkontrolle**: Automatische Wiederholung bei Verlust oder Fehler.

#### ✅ **Verbindungslos (UDP)**

* **Geringe Latenz**: UDP ist schneller, da kein Verbindungsaufbau stattfindet.
* **Echtzeitfähigkeit**: Verluste werden in Kauf genommen, z. B. bei VoIP.
* **Geringer Overhead**: Optimal bei einfachen Abfragen wie DNS oder bei Broadcasts wie DHCP.