Rechnernetze Übung 6

Aufgabe 1: Konzepte

Das **Sliding-Window-Protokoll** wird dazu verwendet, um die Speicherfähigkeit des Mediums optimal zu nutzen und nicht auf die Quittierung bereits gesendeter TCP-Pakete zu warten. Beim Sliding Window Protokoll wird eine Anzahl an Paketen festgelegt, welche ohne das Abwarten von Quittierungen nacheinander geschickt werden. Wenn ein Paket quittiert wird, slided das Window mit n-Paketen weiter und es wird das nächste Paket verschickt.

TCP Tahoe, Reno und Vegas sind entwickelt worden, um festzulegen wie viele n-Pakete beim Sliding Window Protokoll versendet werden.

Bei **TCP Tahoe** startet man mit einem kleinen Window, z.B. 1, solange alle Packets ankommen wird diese Zahl linear größer. Falls es zu einem Fehler bei der Quittierung kommt wird die Window Größe auf den Startwert zurückgesetzt, dies nennt man einen Slow Start.

Bei **TCP Reno** beginnt genauso, wie Tahoe jedoch wird bei Fehlern die Größe des Windows nur halbiert, wodurch weiterhin viele Pakete gesendet werden können.

Bei **TCP Vegas** durch den Vergleich von zu erwarteter und tatsächlicher Übertragungsrate verglichen und so die Größe des Fensters berechnet.

PINK: Tahoe
ROT: Reno
ORANGE: Vegas

Protokolle:

- 1. Physical Layer:
 - V.24
 - RS-232
- 2. Link Layer:
 - FDMA (Frequency)
 - TDMA (Time)
 - CDMA (Code)
 - CSMA (Listen before Talk)
 - CSMA/CD (Listen while Talk)
 - HDLC
 - Ethernet
 - IEEE 802.X
- 3. Network Layer:
 - IP
 - ICMP

- ARP
- RARP
- Routingprotokolle
- 4. Transport Layer:
 - TCP (Verbindung)
 - UDP (Verbindungslos)
- 5. Session Layer:
- 6. Presentation Layer:
 - XDR
 - ASN.1
 - SOAP
 - JSON
- 7. Application Layer:
 - FTP
 - SSH
 - NFS
 - NIS
 - SMTP
 - POP.3
 - HTTP

Aufgabe 2: DHCP

Capture-Filter: "udp port 67 or udp port 68"

Die DHCP Packets habe ich durch das freigeben und erneuern meiner IP-Adresse im Terminal durch ipconfig /release und ipconfig /renew erzeugt.

```
> Frame 1: 350 bytes on wire (2800 bits), 350 bytes captured (2800 bits) on interface \Device\NPF_{658E5E82-C7B7-4F68-AB3E-7688793EEA11}, id 0
v Ethernet II, Src: Intel_5a:35:52 (70:9c:d1:5a:35:52), Dst: AVMAudiovisu_eb:dc:4d (d4:24:dd:eb:dc:4d)
v Destination: AVMAudiovisu_eb:dc:4d (d4:24:dd:eb:dc:4d)
      Address: AVMAudiovisu_eb:dc:4d (d4:24:dd:eb:dc:4d)
      .....0 ...... = LG bit: Globally unique address (factory default)
.....0 ..... = IG bit: Individual address (unicast)
  v Source: Intel_5a:35:52 (70:9c:d1:5a:35:52)
      Address: Intel_5a:35:52 (70:9c:d1:5a:35:52)
      Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.178.20, Dst: 192.168.178.1
    0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
   Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 336
Identification: 0xbdec (48620)
   000. .... = Flags: 0x0
    ...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 128
    Protocol: UDP (17)
    Header Checksum: 0x0000 [validation disabled] [Header checksum status: Unverified]
    Source Address: 192.168.178.20
```

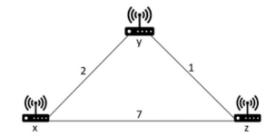
Die Destination ist der für mich zuständige DHCP Server

Der Destination Port ist 67 und der Source Port 68, diese sind für bootp zuständig.

Aufgabe 3: Nmap

- A) Starting Nmap 7.95 (https://nmap.org) at 2024-07-17 10:50 Mitteleuropäische Sommerzeit
- B) Nmap done: 256 IP addresses (0 hosts up) scanned in 208.74 seconds Aggressive OS guesses: Linux 4.19 5.15 (97%), Linux 4.15 (93%), IPFire 2.27 (Linux 5.15 6.1) (92%), Linux 5.4 (91%), Linux 5.0 5.14 (90%), MikroTik RouterOS 7.2 7.5 (Linux 5.6.3) (90%), Linux 3.11 4.9 (90%), Linux 3.2 3.8 (90%), Linux 5.18 (89%), Android TV OS 11 (Linux 4.19) (88%) No exact OS matches for host (test conditions non-ideal).
- C) Creation Date: 1999-01-18T05:00:00.0Z
- D) -T4 -A -v
- E) Der SYN-Scan sendet ein SYN-Paket an den Zielport und analysiert die Antwort (SYN/ACK, RST oder keine Antwort), um den Status des Ports (offen, geschlossen oder gefiltert) zu bestimmen, ohne den TCP-Drei-Wege-Handshake abzuschließen.Der SYN-Scan ist ein effektives und relativ unauffälliges Mittel, um Netzwerke auf offene Ports zu überprüfen
- F) Häufig offene Ports 443 HTTPS, 80 HTTP, 53 DNS, 21 FTP

Aufgabe 4: Routing



Von x	Vi	a	Via	Via
	x		у	Z
Zu x	-	-		
Zu y			2	
Zu z				7

Von y	Via	Via	Via
	X	у	Z
Zu x	2		
Zu y		+	
Zu z			1

Von z	Via	Via	Via
	x	у	Z
Zu x	7		
Zu y		1	
Zu z			+

Von x	Via	Via	Via
	x	y	Z
Zu x			
Zu y		2	8
Zu z	1	3	7

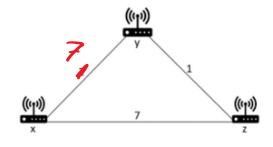
Von y	Via x	Via y	Via z
Zu x	2		8
Zuy		+	
Zu z	9		1

Von z	Via	Via	Via
	x	y	Z
Zu x	4	(്	
Zu y	9	1	
Zu z			-[

Von x	Via	Via	Via
	x	у	Z
Zu x			
Zu y		2	8
Zu z	1	3	7

Von y	Via	Via	Via
	x	y	Z
Zu x	2		8
Zu y			
Zu z	3		//

Von z	Via	Via	Via
	x	у	Z
Zu x	7	3	
Zu y	9	1	
Zu z			\exists



Von x	Via	Via	Via
	x	у	Z
Zu x			
Zu y	\sqcap	1	
,		 	
Zu z			7
			†

Von y	Via	Via	Via
	X	у	Z
Zu x	7		
Zu y		+	
Zu z			1

Von z	Via	Via	Via
	X	у	Z
Zu x	7		
Zu y		1	
Zu z			7

Von x	Vi x	a	Via y	Via z
Zu x	١			
Zu y			7	8
Zu z			8	7

Von y	Via x	Via y	Via z
Zu x	7		8
Zu y		\vdash	
Zu z	8	1	1

Von z	Via	Via	Via
	x	y	Z
Zu x	7	8	
Zu y	8	1	
Zu z			
			7

Von x	Via	Via	Via
	x	y	Z
Zu x			
Zu y		7	8
Zu z		8	7

Von y	Via x	Via v	Via z
Zu x	7		8
Zu y	_		
Zu z	8		/

Von z	Via	Via	Via
	x	у	Z
Zu x	7	8	1
Zu y	8	1	
Zu z	_		1

C)

Wenn Router C seine Verbindungen überprüft.