

# 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, slidet 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.

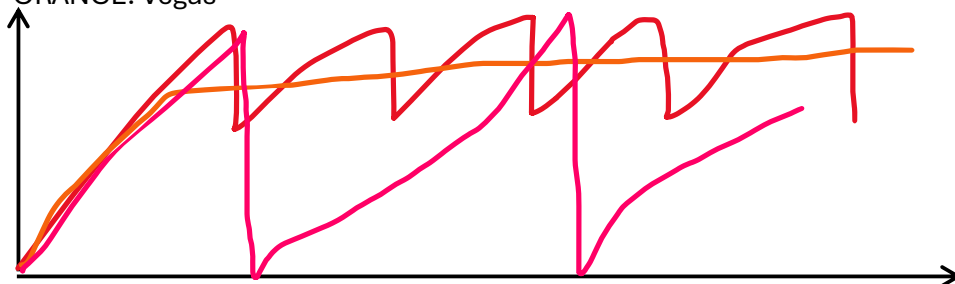
Bei **TCP Reno** beginnt genauso, wie Tahoe jedoch wird bei Fehlern die Größe des Windows nur halbiert, wodurch weiterhin viele Pakete gesendet werden können.

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.

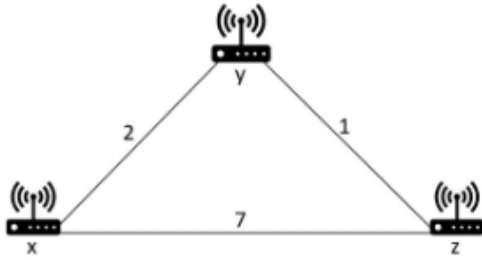
```
> Frame 1: 350 bytes on wire (2800 bits), 350 bytes captured (2800 bits) on interface \Device\NPF_{658E5E82-C7B7-4F68-AB3E-7688793EEA11}, id 0
▼ Ethernet II, Src: Intel_5a:35:52 (70:9c:d1:5a:35:52), Dst: AVMAudiovisu_eb:dc:4d (d4:24:dd:eb:dc:4d)
  ▼ Destination: AVMAudiovisu_eb:dc:4d (d4:24:dd:eb:dc:4d)
    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)
  ▼ Source: Intel_5a:35:52 (70:9c:d1:5a:35:52)
    Address: Intel_5a:35:52 (70:9c:d1:5a:35:52)
    .... 0. .... = LG bit: Globally unique address (factory default)
    .... 0. .... = IG bit: Individual address (unicast)
  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

```
> Frame 1: 350 bytes on wire (2800 bits), 350 bytes captured (2800 bits) on interface \Device\NPF_{658E5E82-C7B7-4F68-AB3E-7688793EEA11}, id 0
> Ethernet II, Src: Intel_5a:35:52 (70:9c:d1:5a:35:52), Dst: AVMAudiovisu_eb:dc:4d (d4:24:dd:eb:dc:4d)
> Internet Protocol Version 4, Src: 192.168.178.20, Dst: 192.168.178.1
▼ User Datagram Protocol, Src Port: 68, Dst Port: 67
  Source Port: 68
  Destination Port: 67
  Length: 316
  Checksum: 0xe6b4 [unverified]
  [Checksum Status: Unverified]
  [Stream index: 0]
  > [Timestamps]
  UDP payload (308 bytes)
> Dynamic Host Configuration Protocol (Request)
```

[illegible]

## Aufgabe 4: Routing



Von x	Via x	Via y	Via z
Zu x			
Zu y		2	
Zu z			7

Von y	Via x	Via y	Via z
Zu x	2		
Zu y			
Zu z			1

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

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

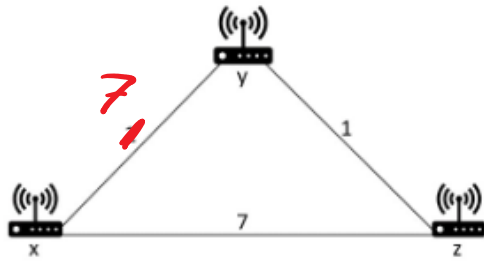
Von y	Via x	Via y	Via z
Zu x	2		8
Zu y			
Zu z	9		1

Von z	Via x	Via y	Via z
Zu x	7	3	
Zu y	9	1	
Zu z			

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

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

Von z	Via x	Via y	Via z
Zu x	7	3	
Zu y	9	1	
Zu z			



Von x	Via x	Via y	Via z
Zu x			
Zu y		7	
Zu z			7

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

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

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

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

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

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

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

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

C)

Wenn Router C seine Verbindungen überprüft.