## Aufgabe 1

### Sliding-Window-Verfahren

Das Sliding-Window-Verfahren ist eine Technik zur soenannten Flusskontrolle in Netzwerken. Es hilft dabei, Daten effizient und ohne Überlastung zu übertragen. Anstelle jede Datenpaket-Übertragung einzeln zu bestätigen, ermöglicht es dem Sender, mehrere Pakete gleichzeitig zu senden, bevor eine Bestätigung vom Empfänger zurückkommt.

Die Fenstergröße bestimmt dabei, wie viele Pakete gleichzeitig gesendet werden dürfen, bevor eine Bestätigung gesendet wird. Sobald dies geschieht, 'schiebt' sich das Fenster weiter und der Sender darf neue Pakete senden.

Das Ziel ist es, die Übertragungsrate dynamisch anzupassen, um die Spiecherkapazität des Mediums optimal auszunutzen.

### TCP Tahoe

TCP Tahoe ist eine frühe Version von TCP. Sie wurde eingeführt, um mit Paketverlusten und Überlastungen im Netzwerk besser umzugehen.

Tahoe verwendet drei Hauptmechanismen:

- Slow Start: Beginnt mit kleiner Fenstergröße (1) ud verdoppelt diese mit jeder Besätigung, bis eine Staugrenze erreicht wird.
- AIMD: (Addditive Increase / Multiplicative Decrease) Lineares Wachstum bei Bestätigung und Exponentielle Reduktion bei Congestion.
- Fast Retransmit: Wenn 3 gleiche ACKs empfangen werden, wird das Paket sofort neu gesendet, ohne auf den Timeout zu warten.

### TCP Reno

TCP Reno ist eine verbesserte Version von TCP Tahoe. Es enthält dieselben Grundmechanismen, führt aber bessere Reaktion auf Paketverluste ein.

Es teilt sich die Funktionen von Tahoe, verändert sie aber leicht. Anstelle die Fenstergröße bei drei gleichen empfangenen ACKs gleich wieder auf 1 zu reduzieren, wird die Fenstergröße nur halbiert. Darüber hinaus wird sich vorsichtiger an das bekannte Fensterlimit herangetastet, um längstmöglich eine schnelle übermittlung zu ermöglichen.

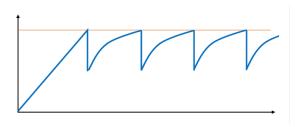


Abbildung 1: Veranschaulichung zu TCP Reno

### TCP Vegas

TCP Vegas ist ein moderner Ansatz zur Staukontrolle, der nicht auf Verluste, sondern auf die Round-Trip-Time (RTT) achtet.

Vegas misst die Round-Trip-Time (RTT) der Pakete und vergleicht die erwartete Rate mit der tatsächlichen Rate. Wenn die RTT ansteigt, reduziert Vegas vorsorglich die Sende-Rate. Damit soll eine Überlastung und damit einhergehende paketverluste präventiv vermieden werden.

### Protokolle und ihre Zuordnung im OSI-Modell

Protokoll	OSI-Schicht	Begründung
UDP	Schicht 4 – Transport	UDP ist ein verbindungsloses Protokoll zur Daten-
		übertragung ohne Bestätigung oder Fehlerkorrek-
		tur.
TCP	Schicht 4 – Transport	TCP ist verbindungsorientiert, stellt Zuverlässig-
		keit, Flusskontrolle und Fehlerkorrektur sicher.
IP	Schicht 3 – Netzwerk	IP regelt die logische Adressierung und das Rou-
		ting von Datenpaketen über Netzwerke.
ARP	Schicht 2 – Sicherung	ARP wandelt IP-Adressen in MAC-Adressen auf
		der lokalen Verbindung um.
RARP	Schicht 2 – Sicherung	RARP bestimmt die IP-Adresse zu einer bekann-
		ten MAC-Adresse (heute veraltet).
ICMP	Schicht 3 – Netzwerk	ICMP dient der Übertragung von Fehler- und Kon-
		trollmeldungen
DNS	Schicht 7 – Anwendung	DNS übersetzt Domainnamen in IP-Adressen und
		wird direkt von Anwendungen genutzt.
HTTP	Schicht 7 – Anwendung	HTTP ist das grundlegende Protokoll zur Über-
		tragung von Webseiten.
HTTPS	Schicht 7 – Anwendung	HTTPS ist HTTP über TLS/SSL zur sicheren Da-
		tenübertragung.
POP	Schicht 7 – Anwendung	POP wird verwendet, um E-Mails vom Server her-
		unterzuladen (typisch: POP3).
IMAP	Schicht 7 – Anwendung	IMAP ermöglicht den Zugriff und die Synchroni-
		sierung von E-Mails auf dem Server.
SMTP	Schicht 7 – Anwendung	SMTP ist das Standardprotokoll zum Versenden
		von E-Mails.
SSH	Schicht 7 – Anwendung	SSH erlaubt sichere Remote-Verbindungen über ei-
		ne verschlüsselte Verbindung.

## Aufgabe 2

- a) nmap -sn 192.168.1.0/24 -> Ergibt bei mir 10 aktive Hosts
- **b)** nmap -O scanme.nmap.org -> nmap.org nutzt wahrscheinlich Linux (Linux 4.19 5.15 (97%)
- c) whois nmap.org -> Creation Date: 1999-01-18T05:00:00Z

- d) sudo nmap -sS -T4 -p- <IP-Adresse>
  - sS: SYN-Scan (schnell und stealthy)
  - T4: Timing-Option für höhere Geschwindigkeit (1 = langsam, 5 = aggressiv)
  - -p-: scannt alle 65535 TCP-Ports
- e) Der SYN-Scan (-sS) sendet ein TCP-SYN-Paket an einen Port. Wenn der Port offen ist, antwortet das Ziel mit SYN-ACK und Nmap schickt darauf RST, um die Verbindung nicht zu vollenden. Wenn der Port geschlossen ist, kommt ein RST zurück.

Der Vorteil ist, dass nie eine vollständige TCP verbindung aufgebaut wird, weshalb der Scan sehr unauffällig ist.

f) Am häufigsten sind die Ports 80 und 443 aufgetreten. Diese sind für http und https zuständig.

# Aufgabe 3

Unter Linux habe ich die DHCP Pakete wie folgt ausgelöst:

sudo dhclient -r (Gibt die aktuelle IP-Adresse frei) sudo dhclient (Fordert eine neue an)

Zu beobachten war, dass die Kommunikation ausschließlich über die Portts 67 und 68 lief und das verwendete Protokoll immer UDP war.

```
└- Dynamic Host Configuration Protocol (Release)
    Message type: Boot Reguest (1)
    Hardware address length: 6
   Transaction ID: 0xcd7c0d4c
    Seconds elapsed: 0
 > Bootp flags: 0x0000 (Unicast)
-Client IP address: 192.168.178.66
    Your (client) IP address: 0.0.0.0
    Next server IP address: 0.0.0.0
   Relay agent IP address: 0.0.0.0
-Client MAC address: ASUSTekCOMPU_6c:88:71 (e8:9c:25:6c:88:71)
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
   Option: (53) DHCP Message Type (Release)
     -Length: 1
     DHCP: Release (7)
   Option: (54) DHCP Server Identifier (192.168.178.1)
     Length: 4
     DHCP Server Identifier: 192.168.178.1
   Option: (255) End
    Ontion End: 255
```

Abbildung 2: Packet Details zur Freigabe

```
    □ Dynamic Host Configuration Protocol (Discover)

     Message type: Boot Request (1)
     Hardware type: Ethernet (0x01)
     Hardware address length: 6
     Hops: 0
     Transaction ID: 0x542ff908
     Seconds elapsed: 0
    Bootp flags: 0x0000 (Unicast)
     -0... = Broadcast flag: Unicast
-000 0000 0000 0000 = Reserved flags: 0x0000
     Client IP address: 0.0.0.0
     Your (client) IP address: 0.0.0.0
    Next server IP address: 0.0.0.0
Relay agent IP address: 0.0.0.0
     Client MAC address: ASUSTekCOMPU_6c:88:71 (e8:9c:25:6c:88:71)
     Server host name not given
    Boot file name not given
Magic cookie: DHCP
    Option: (53) DHCP Message Type (Discover)
    DHCP: Discover (1)
Option: (50) Requested IP Address (192.168.178.66)
       Length: 4
Requested IP Address: 192.168.178.66
    Option: (55) Parameter Request List
       Length: 7
       Parameter Request List Item: (1) Subnet Mask
       Parameter Request List Item: (28) Broadcast Address
       Parameter Request List Item: (2) Time Offset
       -Parameter Request List Item: (3) Router
-Parameter Request List Item: (15) Domain Name
       Parameter Request List Item: (6) Domain Name Server
       Parameter Request List Item: (12) Host Name
    Option: (255) End
```

Abbildung 3: Packet Details zur Suche

```
Dynamic Host Configuration Protocol (Offer)
    Message type: Boot Reply (2)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Transaction ID: 0x542ff908
    Seconds elapsed: 0
    Bootp flags: 0x0000 (Unicast)
    0..... Broadcast flag: Unicast
.000 0000 0000 0000 = Reserved flags: 0x0000
    Client IP address: 0.0.0.0
    Your (client) IP address: 192.168.178.66
    Next server IP address: 192.168.178.1
Relay agent IP address: 0.0.0.0
    Client MAC address: ASUSTekCOMPU_6c:88:71 (e8:9c:25:6c:88:71)
    Client hardware address padding: 000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
   Option: (53) DHCP Message Type (Offer)
     Length: 1
DHCP: Offer (2)
   Option: (54) DHCP Server Identifier (192.168.178.1)
      Length: 4
  DHCP Server Identifier: 192.168.178.1

Option: (51) IP Address Lease Time
      - IP Address Lease Time: 10 days (864000)
   ∕-Option: (58) Renewal Time Value
     – Length: 4
– Renewal Time Value: 5 days (432000)
   Option: (59) Rebinding Time Value
     -Length: 4
      Rebinding Time Value: 8 days, 18 hours (756000)
  option: (1) Subnet Mask (255.255.255.0)
     Length: 4
Subnet Mask: 255.255.255.0
   Option: (3) Router
      Length: 4
      Router: 192.168.178.1
   Option: (6) Domain Name Server
     -Length: 4
  Domain Name Server: 192.168.178.1
Option: (15) Domain Name
     Length: 9
Domain Name: fritz.box
    Option: (28) Broadcast Address (192.168.178.255)
    Length: 4
      Broadcast Address: 192.168.178.255
  /-Option: (255) End
```

Abbildung 4: Packet Details zum 'Angebot'

```
> Dynamic Host Configuration Protocol (Request)
     Message type: Boot Request (1)
Hardware type: Ethernet (0x01)
     Hardware address length: 6
     Hops: 0
     Transaction ID: 0x542ff908
     Seconds elapsed: 0
     Bootp flags: 0x0000 (Unicast)
     -0... = Broadcast flag: Unicast
-0.00 0000 0000 0000 = Reserved flags: 0x0000
     Client IP address: 0.0.0.0
     Your (client) IP address: 0.0.0.0
     Next server IP address: 0.0.0.0
Relay agent IP address: 0.0.0.0
     Server host name not given
Boot file name not given
    -Magic cookie: DHCP
-Option: (53) DHCP Message Type (Request)
    -Length: 1
-DHCP: Request (3)
-Option: (54) DHCP Server Identifier (192.168.178.1)
      Length: 4
   DHCP Server Identifier: 192.168.178.1

Option: (50) Requested IP Address (192.168.178.66)
       Length: 4
       Requested IP Address: 192.168.178.66
    Option: (55) Parameter Request List
       – Length: 7
– Parameter Request List Item: (1) Subnet Mask
       Parameter Request List Item: (28) Broadcast Address
Parameter Request List Item: (2) Time Offset
       Parameter Request List Item: (3) Router
Parameter Request List Item: (15) Domain Name
       Parameter Request List Item: (6) Domain Name Server
Parameter Request List Item: (12) Host Name
      Option End: 255
```

Abbildung 5: Packet Details zur Anfrage

```
→ Dynamic Host Configuration Protocol (ACK)

     Message type: Boot Reply (2)
     Hardware type: Ethernet (0x01)
     Hardware address length: 6
     Hops: 0
Transaction ID: 0x542ff908
    -Seconds elapsed: 0
-Bootp flags: 0x0000 (Unicast)
     -0... = Broadcast flag: Unicast
.000 0000 0000 0000 = Reserved flags: 0x0000
     Client IP address: 0.0.0.0
     Your (client) IP address: 192.168.178.66
     Next server IP address: 192.168.178.1
Relay agent IP address: 0.0.0.0
     Client MAC address: ASUSTekCOMPU_6c:88:71 (e8:9c:25:6c:88:71)
     Boot file name not given
Magic cookie: DHCP
     Option: (53) DHCP Message Type (ACK)
   Length: 1
DHCP: ACK (5)
Option: (54) DHCP Server Identifier (192.168.178.1)
       DHCP Server Identifier: 192.168.178.1
     Option: (51) IP Address Lease Time
      Length: 4
       -IP Address Lease Time: 10 days (864000)
    COption: (58) Renewal Time Value
Length: 4
       Renewal Time Value: 5 days (432000)
   Option: (59) Rebinding Time Value
       Length: 4
-Rebinding Time Value: 8 days, 18 hours (756000)
    Option: (1) Subnet Mask (255.255.255.0)
       – Length: 4
– Subnet Mask: 255.255.255.0
   √-Option: (3) Router
       - Length: 4
- Router: 192.168.178.1
   v-Option: (6) Domain Name Server
       Length: 4
-Domain Name Server: 192.168.178.1
   ∕-Option: (15) Domain Name
      Length: 9
Domain Name: fritz.box
   option: (28) Broadcast Address (192.168.178.255)
       Broadcast Address: 192.168.178.255
   √-Option: (255) End
      Option End: 255
    - Padding [...]: 000000000000000000000
```

Abbildung 6: Packet Details zur Antwort

# Aufgabe 4

#### Pseudocode:

Für alle Router im Netzwerk:

Initialisere Tabelle mit Kosten zu direkten Nachbarn

### Aktualisiere im Intervall:

Gehe alle Wege ab iterativ ab:

Berechne die Summe der Kosten für die neuen Verbindungen Markiere die günstigsten Verbindungen

#### Die Unterschiede zu dem Link-State-Verfahren sind:

- Beim Distanzvektorverfahren tauscht ein Router seine Routinginformationen nur mit seinen direkten Nachbarn aus, während beim Link-State-Verfahren jeder Router die vollständige Netzwerktopologie kennt und sie an alle anderen Router sendet.
- Distanzvektor-Router verwenden einfache Algorithmen wie Bellman-Ford, während Link-State-Router komplexere Verfahren wie den Dijkstra-Algorithmus einsetzen.
- Die Konvergenz (also das Anpassen an Netzveränderungen) ist bei Distanzvektor langsamer, da Informationen schrittweise weitergegeben werden, während Link-State schneller reagiert.
- Link-State-Verfahren sind robuster und genauer, da jeder Router das vollständige Bild des Netzwerks hat, während Distanzvektorverfahren anfälliger für Probleme wie das "Count-to-Infinity"-Problem sind.
- In kleinen Netzwerken sind Distanzvektorverfahren oft ausreichend, in größeren und komplexeren Netzen hingegen sind Link-State-Verfahren besser geeignet.

**a**)

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

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

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

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

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

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

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

b)

Wenn sich die Verbindungskosten von x nach y auf 7 steigern, wird ein Schritt weniger benötigt, da direkt ersichtlich wird, dass jetzt in jedem Fall die direkte Verbindung die schnellste ist,

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

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

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

Von x	Via	Via	Via
	x	y	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	14		1

Von z	Via	Via	Via
	x	у	z
Zu x	7	8	
Zu y	14	1	
Zu z			

**c**)

C würde es zuerst mitbekommen, da es direkt mit D verbunden ist. In der nächsten Iteration würde es B von C erfahren und in wieder der nächsten A von B, da dieser Weg kürzer ist, als direkt von C nach A.