Übung 6

Mittwoch, 2. Juli 2025 12:43

Aufgabe 1:

- **1. Sliding-Window**

Was ist das?

Das Sliding-Window-Verfahren ist eine Technik zur **Flusskontrolle** und **Effizienzsteigerung** beim Datentransport über Netzwerke – insbesondere bei **TCP-Verbindungen**.

Flusskontrolle braucht man da ein Sender Daten oft schneller senden kann, als ein Empfänger sie verarbeiten oder zwischenspeichern kann. Das würde dazu führen dass:

- der **Empfänger-Puffer überläuft**,
- **Daten verloren gehen**,
- oder **Verbindungen instabil werden**.

Funktionieren tut die Flusskontrolle so, dass:

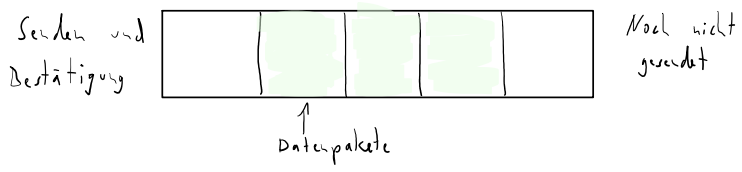
- **Der Empfänger** dem Sender mitteilt, wie viele Bytes er noch verarbeiten kann → das nennt man das **Empfangsfenster (Receive Window)**.
- **Der Sender** darf dann nur so viele Daten schicken, wie in dieses Fenster passen.
- Wenn der Empfänger Daten verarbeitet und Platz schafft, wird das Fenster wieder größer, und der Sender darf mehr schicken.

Wie funktioniert es?

Anstatt nur ein Datenpaket zu senden und auf die Bestätigung (ACK) zu warten, erlaubt das Sliding-Window mehreren Paketen, gleichzeitig "im Flug" zu sein. Der Sender hat ein "Fenster", das angibt, wie viele unbestätigte Pakete gesendet werden dürfen. Dieses Fenster kann sich verschieben (engl. "slide"), wenn Bestätigungen eintreffen.

Wozu braucht man das?

- Um die Bandbreite effizienter auszunutzen
- Um Wartezeiten zwischen Senden und Empfangen zu reduzieren
- Um eine zuverlässige Datenübertragung bei gleichzeitiger hoher Geschwindigkeit zu ermöglichen



Die Farben bedeuten:

- **Weiß links:** Daten wurden **schon gesendet und bestätigt** (ACK erhalten).
- **Grün in der Mitte:** Daten wurden **gesendet, aber noch nicht bestätigt**.
- **Weiß rechts:** Daten wurden **noch nicht gesendet**, dürfen aber bald gesendet werden, wenn das Fenster „weitergeschoben“ wird.

Dieses Fenster bewegt sich (daher „sliding“) **nach rechts**, sobald Bestätigungen für gesendete Pakete eintreffen.

- **2. TCP Tahoe**

Was ist das?

TCP Tahoe ist eine frühe Variante von TCP, die Mechanismen zur **Staukontrolle** (Congestion Control) einführt.

Staukontrolle versucht, die **Datenrate so zu steuern**, dass:

- das Netzwerk **nicht überfüllt** wird,
- **Paketverluste vermieden** werden,
- und **alle Verbindungen fair** nebeneinander funktionieren.

Wie funktioniert es?
Tahoe nutzt drei Hauptmechanismen:

1. **Slow Start**: Zu Beginn wird das Sende-Fenster (gibt an wie viele Daten ein Sender auf einmal ins Netzwerk schicken darf) exponentiell vergrößert, um die verfügbare Bandbreite zu ertasten.
2. **Congestion Avoidance**: Wenn ein Schwellenwert erreicht ist, wird das Wachstum des Fensters verlangsamt (linear statt exponentiell).
3. **Fast Retransmit**: Wenn 3 gleiche ACKs empfangen werden, wird ein Paketverlust vermutet und das Paket wird neu gesendet.

Bei Paketverlust: Das Fenster wird **auf 1** zurückgesetzt und Slow Start beginnt erneut.

Wozu braucht man es?
Um Überlastung im Netzwerk zu vermeiden und den Datentransport anzupassen, falls Pakete verloren gehen.

- **3. TCP Reno**

Was ist das?
TCP Reno ist eine Weiterentwicklung von TCP Tahoe mit verbesserter Reaktion auf Paketverluste.

Wie funktioniert es?
Es nutzt ebenfalls Slow Start, Congestion Avoidance und Fast Retransmit, aber zusätzlich:

- **Fast Recovery**: Nach Fast Retransmit wird das Fenster **nicht ganz zurückgesetzt**, sondern nur reduziert, um nicht komplett von vorne anzufangen.

Wozu braucht man es?
Reno reagiert **schneller und effizienter** auf vereinzelte Paketverluste als Tahoe, dadurch steigt die Leistung bei mittleren Störungen im Netzwerk.

- **4. TCP Vegas**

Was ist das?
TCP Vegas ist ein intelligenterer TCP-Ansatz, der nicht nur auf **Verluste** reagiert, sondern **Paketverzögerungen** erkennt, bevor Verluste auftreten.

Wie funktioniert es?

- Es vergleicht die **erwartete** mit der **tatsächlichen** Übertragungsrate.
- Wenn Pakete länger als erwartet unterwegs sind, wird das als ein **Anzeichen für beginnende Überlastung** gesehen.
- Vegas passt die Sende-Rate **proaktiv** an, bevor es zu Paketverlusten kommt.

Wozu braucht man es?

- Um effizienter und **frühzeitiger auf Staus** zu reagieren
- Um Paketverluste **ganz zu vermeiden**, statt nur darauf zu reagieren
- Für stabilere und fairere Netzwerkauslastung

Protokoll	ISO/OSI Layer	Begründung
IP	3. Layer (Networklayer)	Sorgt für logische Adressierung und Routing von Datenpaketen, bestimmt den Weg von Paketen vom Sender zum Empfänger über mehrere Netzwerke
ICMP	3. Layer (Networklayer)	Tauscht Fehler- und Statusmeldungen zwischen Netzwerkgeräten aus, unterstützt so die Routing-Funktion von IP
ARP	3. Layer (Networklayer)	Übersetzt IP-Adressen in MAC-Adressen für korrekte Zustellung von Paketen im lokalen Netzwerk
RARP	3. Layer (Networklayer)	Übersetzt MAC-Adresse in IP-Adresse
UDP	4. Layer (Transport Layer)	bietet eine einfache, schnelle Ende-zu-Ende-

		Datenübertragung, ohne Fehlerkorrektur, also minimaler Transportdienst
TCP	4. Layer (Transport Layer)	Baut Verbindung auf, nutzt sie, baut sie ab, sorgt für die Ende-zu-Ende-Kommunikation mit Kontrolle
DNS	7. Layer (Application Layer)	Netzwerkdienst, der von Anwendungen benutzt wird um Domainnamen in IP-Adressen aufzulösen
DHCP	7. Layer (Application Layer)	Netzwerkdienst, der wichtige Konfigurationsdaten zuweist (IP-Adresse, Subnetzmaske…)
NAT	3. Layer (Networklayer)	Wandelt private IP-Adressen in öffentliche um

Aufgabe 2:

a) Wie viele Hosts befinden sich in ihrem lokalen Klasse-C-Netz?
- nmap -sn -T4 192.168.0.0/24 (-sn: Ping Scan, überprüft ob Hosts im Netzwerk erreichbar sind, -T4: kürzere Timeouts, schnelleres Senden von Paketen, 192.168.0.0/24: CIDR-Notation für C-Klasse-Netzwerk
- 2 Host sind erreichbar von 256 Hosts
- In Wireshark sind bei Abfrage IGMPv2 Protokolle zu sehen

b) Welches Betriebssystem wird von scanme.nmap.org verwendet?
- nmap -O scanme.nmap.org (-O: Aktiviert Betriebssystem-Erkennung)
- Linux Kernel Version 4.19-5.15

c) An welchem Datum wurde die Webseite nmap.org registriert?
- https://www.whois.com/whois/nmap.org
- Registriert am 18.01.1999

d) Wie kann man möglichst effektiv eine größere Menge an Adressen nach offenen TCP-Ports scannen?
- Bereiche von Subnetzen angeben statt einzeln z. B. nmap 192.168.1.0/24
- -T4 oder -T5 für schnellere Scans
- -sn um nur Hosts zu scannen, die auch online sind

e) Wie funktioniert der SYN-Scan und für was kann man ihn verwenden?
- SYN-Scan ist halboffender TCP-Scan (Verbindung wird nicht vollständig aufgebaut)
- Nutzt das SYN-Paket von TCP-Verbindung, um zu prüfen, ob ein Port offen ist
  -> Scanner schickt an Zielort ein SYN-Paket (Anfrage zum Verbindungsaufbau)
  -> bei SYN/ACK: Port ist offen -> Scanner sendet RST (Reset) damit Verbindung nicht vollständig aufgebaut wird
  -> bei RST: Port ist geschlossen
  -> Keine Antwort oder ICMP unreachable: Port ist gefiltered (Firewall blockiert oder Paket wird verworfen)
- Schnelles, unauffälliges Erkennen offener Ports, Port- und Sicherheitsprüfung, Netzwerkanalyse
- nmap -sS ziel-ip (-sS aktiviert SYN-Scan)

f) Welches sind die offenen Ports, die bei Ihren bisherigen Nmap-Scans am häufigsten auftreten, und wofür werden sie verwendet?
- Src Port: 49350 (Quellport meines Rechners)
- Dst Port: 54656 - 54706 (Port auf dem Zielsystem)

Aufgabe 3:
Ich habe in cmd folgendes eingegeben: ipconfig /release und ipconfig /renew

DHCP Release:

```
Frame 485: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits)
Ethernet II, Src: 56:40:97:50:19:56 (56:40:97:50:19:56), Dst: CompalBroadb_f5:42:c7 (5c:35:3b:
Internet Protocol Version 4, Src: 192.168.0.254, Dst: 192.168.0.1
User Datagram Protocol, Src Port: 68, Dst Port: 67
Dynamic Host Configuration Protocol (Release)
```

Ethernet II: MAC-Adresse meines Clients, MAC-Adresse des DHCP-Servers
IPv4: aktuelle IP-Adresse meines Clients, IP-Adresse des DHCP-Servers
UDP: DHCP läuft auf UDP, immer von Port 68 (Client) zu Port 67 (Server)

DHCP Discover:

```
Frame 799: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits)
Ethernet II, Src: 56:40:97:50:19:56 (56:40:97:50:19:56), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
User Datagram Protocol, Src Port: 68, Dst Port: 67
Dynamic Host Configuration Protocol (Discover)
```

Ethernet II: MAC-Adresse meines Clients, Broadcast-Adresse weil Client den DHCP-Server im Netzwerk sucht
IPv4: Src: 0.0.0.0 (Client hat noch keine IP, Ziel-Broadcast-Adresse
UDP: von Port 68 (Client) zu Port 67 (Server)

DHCP Offer:

```
Frame 821: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits)
Ethernet II, Src: CompalBroadb_f5:42:c7 (5c:35:3b:f5:42:c7), Dst: 56:40:97:50:19:56 (56:40:97:50:19:56)
Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.254
User Datagram Protocol, Src Port: 67, Dst Port: 68
Dynamic Host Configuration Protocol (Offer)
```

Ethernet II: Paket kommt vom DHCP-Server zu MAC-Adresse des Clients
IPv4: Src: von IP-Adresse des DHCP-Servers nach Client-MAC IP
UDP: von Port 67 (Server) zu Port 68 (Client)

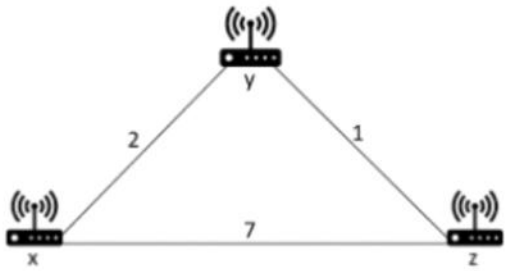
DCHP Request:

```
Frame 826: 360 bytes on wire (2880 bits), 360 bytes captured (2880 bits)
Ethernet II, Src: 56:40:97:50:19:56 (56:40:97:50:19:56), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 0.0.0.0, Dst: 255.255.255.255
User Datagram Protocol, Src Port: 68, Dst Port: 67
Dynamic Host Configuration Protocol (Request)
```

Ethernet II: Paket kommt von MAC-Adresse des Clients zu Broadcast-Adresse
IPv4: Src: 0.0.0.0 (Client hat noch keine IP, Ziel-Broadcast-Adresse
UDP: von Port 68 (Client) zu Port 67 (Server)

Aufgabe 4:

a) Bestimmen Sie die Routingtabellen in jedem Schritt, indem Sie Ihren zuvor entwickelten Algorithmus anwenden. Geben Sie danach den kostengünstigsten Weg von Router „z“ zu Router „x“ an. Sie können die Lösungen einfach direkt in die Tabellen der PDF einfügen und diese in den Pull Request hinzufügen.



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			

Kostengünstigster Weg von Router z zu Router x: Von z via y zu x Kosten = 3

b) Die Kosten zwischen „x“ und „y“ steigen nun von 2 auf 7. Berechnen Sie die Routingtabellen mit Hilfe des Algorithmus. Ändert sich der kostengünstigste Pfad von „z“ nach „x“?

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	14		1

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

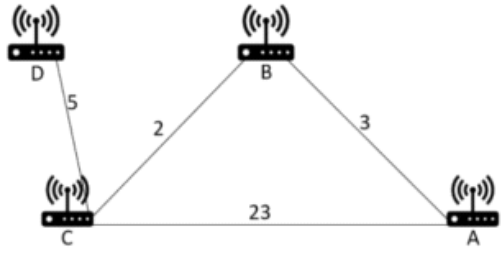
Von x	Via x	Via y	Via z
Zu x			
Zu y		7	8
Zu z		8	7

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

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

Kostengünstigster Weg von Router z zu Router x: Von z via x zu x Kosten = 7

c) Sehen Sie sich den unteren Graphen an. Router „D“ fällt nun auf einmal aus. Beschreiben Sie, ob und wann die anderen Router merken, dass keine Verbindung mehr zu „D“ möglich ist.



Router C merkt es direkt, da er keine ACKs mehr bekommt oder Timeouts bekommt.
Router B und A merken es beide erst nach C, da sie nur indirekt über C mit D kommunizieren.