

Aufgabe 1

Sliding-Window-Verfahren

Das Sliding-Window-Verfahren ist eine Technik zur sogenannten 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 Speicherkapazität 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) und verdoppelt diese mit jeder Bestätigung, bis eine Staugrenze erreicht wird.
- **AIMD:** (Additive 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.

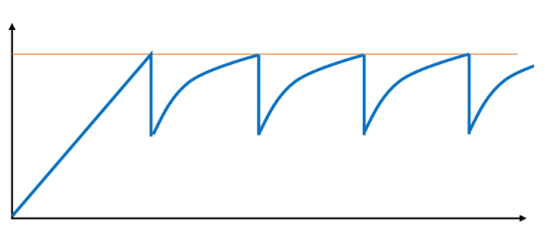


Abbildung 1: Veranschaulichung zu TCP Reno

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 Datenübertragung ohne Bestätigung oder Fehlerkorrektur.
TCP	Schicht 4 – Transport	TCP ist verbindungsorientiert, stellt Zuverlässigkeit, Flusskontrolle und Fehlerkorrektur sicher.
IP	Schicht 3 – Netzwerk	IP regelt die logische Adressierung und das Routing 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 bekannten MAC-Adresse (heute veraltet).
ICMP	Schicht 3 – Netzwerk	ICMP dient der Übertragung von Fehler- und Kontrollmeldungen
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 Übertragung von Webseiten.
HTTPS	Schicht 7 – Anwendung	HTTPS ist HTTP über TLS/SSL zur sicheren Datenübertragung.
POP	Schicht 7 – Anwendung	POP wird verwendet, um E-Mails vom Server herunterzuladen (typisch: POP3).
IMAP	Schicht 7 – Anwendung	IMAP ermöglicht den Zugriff und die Synchronisierung 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 eine 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

[illegible]

Abbildung 3: Packet Details zur Suche

[illegible]

Abbildung 4: Packet Details zum 'Angebot'

[illegible]

Abbildung 5: Packet Details zur Anfrage

[illegible]

Abbildung 6: Packet Details zur Antwort

Aufgabe 4

Pseudocode:

Für alle Router im Netzwerk:

 Initialisiere 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 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			

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 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			

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.