

Netzwerkstaukontrolle und die Arbeitsweise verschiedener Algorithmen in TCP

von

Michał Roziel

Matrikelnummer : 5012845

Ein wissenschaftlicher Bericht im Rahmen der Vorlesung
„Wissenschaftliches Arbeiten“
an der htw saar im Studiengang Informatik

Saarbrücken, den 10. Juli 2025

Abstract

Dieser Bericht zielt darauf ab, aktuell benutzte Algorithmen der Netzwerkstaukontrolle in Computernetzwerken zu vergleichen. In dem heutigen Stand von Rechnernetzen werden verschiedene Protokolle zur Netzwerkstaukontrolle verwendet, Ich werde mich im Rahmen dieses Berichts allerdings auf das Transmission Control Protocol beschränken, da dies das am meisten verbreitete ist.

Die Algorithmen der Staukontrolle, welche in diesen Vergleich einfließen sind : TCP BBR, TCP NewReno, TCP Cubic, und TCP Vegas.

Mittels des Open-Source Netzwerksimulators *NS-3* wird ein virtuelles Netzwerk zwischen zwei Endpunkten aufgestellt, über welches ein Künstlicher Datenverkehr erzeugt wird. Der genannte Datenverkehr wird anschließend aufgezeichnet und dient somit als Basis für den Vergleich der Staukontrolle durch die verschiedenen Algorithmen. Die Auswertung, welche aus den Aufzeichnungen stammt, ermöglicht dem Leser ein tieferes Verständnis über Staukontrolle in Computernetzwerken zu erhalten.

Selbstständigkeitserklärung

Ich versichere, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Insbesondere habe ich alle KI-basierten Werkzeuge angegeben, die ich bei der Erstellung, Übersetzung oder Überarbeitung des Textes verwendet habe.

Ich erkläre hiermit weiterhin, dass die vorgelegte Arbeit zuvor weder von mir noch von einer anderen Person an dieser oder einer anderen Hochschule eingereicht wurde.

Darüber hinaus ist mir bekannt, dass die Unrichtigkeit dieser Erklärung eine Benotung der Arbeit mit der Note „nicht ausreichend“ zur Folge hat und einen Ausschluss von der Erbringung weiterer Prüfungsleistungen zur Folge haben kann.

Saarbrücken, den 10. Juli 2025

Unterschrift Michał Roziel

Inhaltsverzeichnis

1. Einleitung	1
1.1. Grundlagen und Begriffe	1
1.1.1. Transmission Control Protocol	1
1.2. Algorithmen der Staukontrolle	2
1.2.1. Typen von Algorithmen der Staukontrolle	2
1.2.2. TCP Reno	3
1.2.3. TCP BBR	3
1.2.4. TCP Vegas	3
1.2.5. Cubic	3
1.3. Definitionen	3
1.3.1. Queuing Delay	3
2. Versuchsaufbau	4
2.1. Analyse der Ergebnisse	4
3. Ergebnisse und Diskussion	5
4. Fazit und Schlussfolgerungen	5
4.1. Offene Fragen	5
4.2. Diskussion	5
Literaturverzeichnis	6
Anhang	7
A. Datenmaterial	7
B. Web-Standards	7

1. Einleitung

Network congestion control (CC) ist ein essentieller Bestandteil der meisten modernen Computernetzwerke. Wenn eine Netzwerkschnittstelle zu einem Zeitpunkt versucht eine zu große Menge an Datenpaketen aufzunehmen, kommt es zu Stau von Datenverkehr und zu einem potentiellen Verlust von Datenpaketen. Aufgrund diesem Vorkommen werden Algorithmen innerhalb von Netzwerkprotokollen verwendet, diese erkennen den Anstau von Datenverkehr im Netzwerk, und helfen den Fluss von Datenpaketen zu steuern. Neben dem effizienten Durchfluss von Informationen ist zeitgleich auch die Fairness, was die Verteilung von Ressourcen einer Netzwerkschnittstelle an ihre Hosts angeht, wichtig. Auch dies wird von CC-Algorithmen gewährleistet.

Mit stets weiterentwickelten Rechnernetzen, und einem jährlich zunehmenden Datenverkehr, wie auch in der Deutschen Internetschnittstelle *DE-CIX* [DC25] in Frankfurt gewinnen diese Algorithmen an Bedeutung.

1.1. Grundlagen und Begriffe

1.1.1. Transmission Control Protocol

Das Transmission Control Protocol (TCP) ist ein weit verbreitetes Netzwerkprotokoll. TCP lässt sich in zu der Schicht 4 (Transportschicht) in dem OSI-Modell einordnen, hierbei liegt es zwischen der Vermittlungs- und Kommunikationsschicht. TCP wurde 1981 unter RFC 793 erstmals standardisiert. [Pos81]

Die Aufgabe des TCP Protokolls ist es, zwischen zwei Hosts eine Verbindung aufzubauen, welche anschließend dazu genutzt wird, Nachrichten in zu verschicken und zu empfangen.

“transport-layer protocol [...] from an application’s perspective, it is as if the hosts running the processes were directly connected;” [JF22, 241].

Dies bedeutet, dass TCP ebenfalls ein gewisses Abstraktionsniveau des Nachrichtenaustausches abnimmt.

Ein klares Unterscheidungsmerkmal des TCP von dem ebenfalls bekannten User Datagram Protocol (UDP) ist, dass TCP verbindungsorientiert arbeitet, während UDP als verbindungslos gilt. Bei TCP wird zu Beginn ein *Three-Way-Handshake* wie folgt durchgeführt :

Der Sender schickt zunächst eine Nachricht mit einem Verbindungswunsch an den Empfänger und setzt das Flag *Synchronize*, **SYN**.

Der Empfänger antwortet mit einer Bestätigung die erste Nachricht erhalten zu haben, und schickt ebenfalls einen Verbindungswunsch. Es werden die Flags **SYN-ACK** (Synchronise-Acknowledge) gesetzt.

Als letztes schickt der Sender eine Bestätigung, dass die Nachricht des Empfängers angekommen ist, dies geschieht wieder mit dem Flag **ACK**. Nun ist die Verbindung bereit, Daten in beide Richtungen zu übertragen. Nach jedem gesendeten Datenpaket folgt eine **ACK** Bestätigung.

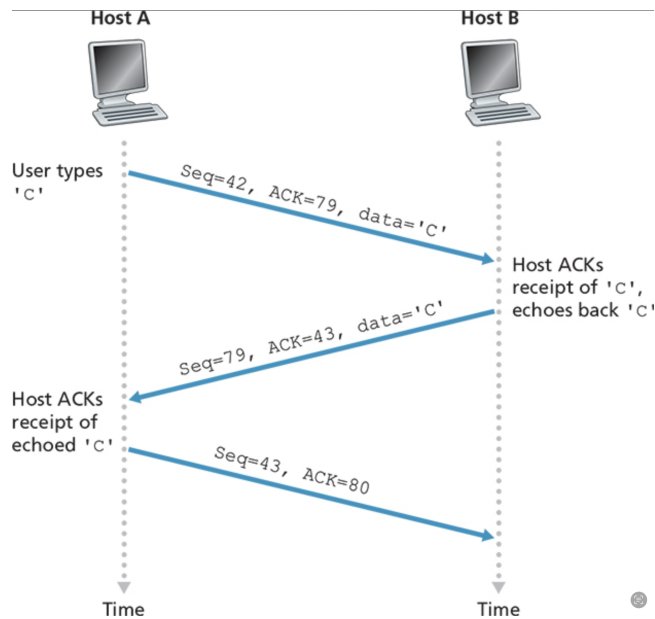


Abbildung 1: *Three-Way-Handshake*.
[JF22, 296]

1.2. Algorithmen der Staukontrolle

Algorithmen der Staukontrolle kommen zum Einsatz, wenn in einem Netzwerk ein potentieller Datenstau erkannt wird. Grundsätzlich starten die Algorithmen in dem Arbeitsmodus *slow start*. Falls Datenstau erkannt wird, treten diese Algorithmen in den Arbeitsmodus *congestion avoidance*.

Hiermit wird mit verschiedenen Vorgehensweisen gegen die Netzwerküberlastung gesteuert.

Es existieren unterschiedliche Typen von CC-Algorithmen, die jeweils ihre eigenen Vor- und Nachteile haben. In diesem Bericht wird jeweils ein Algorithmus pro Typ behandelt.

1.2.1. Typen von Algorithmen der Staukontrolle

- Loss-Based Algorithmen
- Delay-Based Algorithmen
- Model-Based Algorithmen

1.2.2. TCP Reno

1.2.3. TCP BBR

1.2.4. TCP Vegas

1.2.5. Cubic

1.3. Definitionen

1.3.1. Queuing Delay

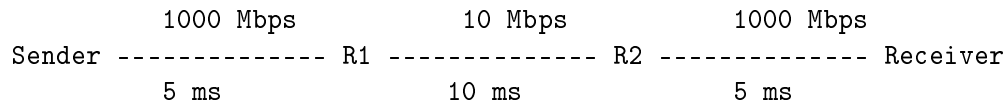
Queuing Delay beschreibt das Warten einer *Queue* auf das Versenden durch das gegebene Netzwerk.

2. Versuchsaufbau

Mit einem Test innerhalb des Command Line Simulators *NS-3.45* lässt sich ein beliebiger CC-Algorithmus ausführen.

Netzwerktopologie

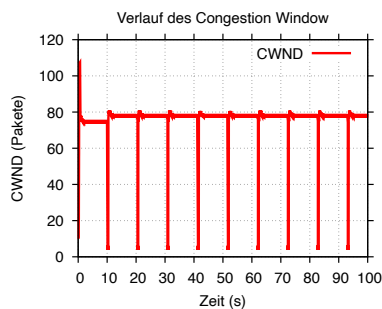
Zunächst definieren wir die Netzwerktopologie auf welcher die CC-Algorithmen laufen werden.



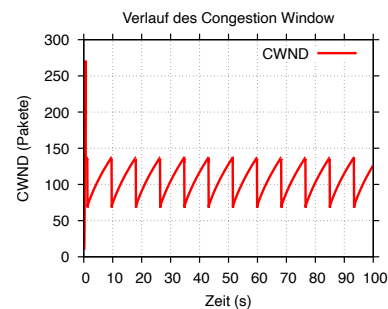
So ist es möglich, einen Netzwerkstau zu simulieren, dieser bleibt für jeden Algorithmus gleich.

Mittels der Navigation in das Directory *ns-3* per Kommandozeile und dem Befehl

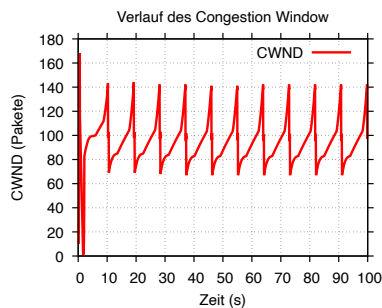
```
./ns3 run examples/tcp/tcp-bbr-example.cc
```



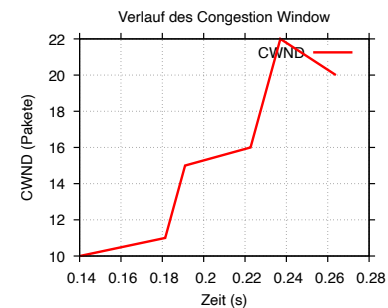
(a) BBR



(b) NewReno



(c) Cubic



(d) Vegas

Abbildung 2: Vergleich der Congestion Windows von BBR, NewReno, Cubic und Vegas.

2.1. Analyse der Ergebnisse

Wie an den *Congestion Window* Graphen erkennbar, besitzt jeder der vorgestellten Algorithmen einen

3. Ergebnisse und Diskussion

4. Fazit und Schlussfolgerungen

4.1. Offene Fragen

4.2. Diskussion

Literaturverzeichnis

- [DC25] DE-CIX. Traffic statistics frankfurt, 2025. Letzter Zugriff: 10. Juli 2025.
- [JF22] Keith W.Ross James F.Kurose. *Computer Networking: A Top-Down Approach*. Pearson, London, 8th edition, 2022.
- [Pos81] Jon Postel. transmission control protocol, 1981.

Anhang

A. Datenmaterial

B. Web-Standards

0