**Literature survey on how TCP algorithms can be designed for futuristic datacenters**

In the current landscape, traditional TCP such as TCP Reno and TCP Cubic was designed for the internet, not for the high-speed, low-latency environments of futuristic data centers needed in the current world.

Modern, futuristic data centers require high bandwidth and low latency. The short-lived flows, common in datacenter workloads, don’t benefit from traditional TCP’s slow-start mechanism. TCP offers inefficient congestion control in highly dynamic environments. Traditional TCP do not cater to emerging needs of futuristic data centers like Stargate, which require scalable, low-latency communication and advanced congestion control to handle modern AI workloads.

Currently, to provide high bandwidth and scalability, Clos network is the network topology used in datacenters. It ensures that multiple paths exist between servers, reducing bottlenecks and enabling low-latency communication. Clos networks require congestion control algorithms that can handle multi-path routing and dynamic traffic patterns.

Traditional TCP uses AIMD (Additive Increase Multiplicative Decrease), which may not be optimal for futuristic data centers which need advanced congestion control. Newer algorithms like DCTCP (Data Center TCP), TIMELY, and HPCC (High Precision Congestion Control) are designed specifically for data centers. These algorithms focus on low latency, high throughput and fairness.

John Ousterhout’s paper, "It’s Time to Replace TCP in the Datacenter", argues that TCP is outdated for futuristic data centers. He proposes alternatives like Homa, a transport protocol designed for datacenters that uses receiver-driven flow control and prioritizes short messages.

Nevertheless, for TCP algorithms to be adapted and designed for futuristic data centers, these are some TCP algorithms that may be suitable for implementation in futuristic data centers, and should be used in order to adapt to provide high-speed and low-latency to support futuristic data centers. For example, the DCTCP (Data Center TCP) algorithm, which uses Explicit Congestion Notification (ECN) to detect congestion early and adjust the sending rate more precisely, will reduce latency and improve throughput in data centers ("Data Center TCP (DCTCP)" by Alizadeh et al., 2010). Another algorithm would be TIMELY, which uses Round-Trip Time (RTT) as a congestion signal instead of packet loss or ECN, would be able to achieve low latency and high throughput, especially for short flows ("TIMELY: RTT-based Congestion Control for the Datacenter" by Mittal et al., 2015). Another example is HPCC (High Precision Congestion Control) algorithm, which leverages in-network telemetry (INT) to get precise congestion feedback from switches, which would provide ultra-low latency and high bandwidth utilization. ("HPCC: High Precision Congestion Control" by Li et al., 2019). Another example would be Homa, a receiver-driven protocol that prioritizes short messages and uses network priorities to reduce latency, as it outperforms TCP in modern data center environments, especially for AI workloads. ("Homa: A Receiver-Driven Low-Latency Transport Protocol Using Network Priorities" by Ousterhout et al., 2018). Other approaches that modern TCPs could use include QUIC, a modern transport protocol designed for low-latency communication, often used in web applications, as well as RDMA (Remote Direct Memory Access), which bypasses the CPU and OS to enable high-speed data transfer, often used in high-performance computing (HPC) and AI datacenters.

TCP algorithms should adhere to these design principles to be able to adapt to futuristic data centers. Firstly, low latency, to reduce queueing delays and RTT. Secondly, high throughput, to maximize bandwidth utilization without causing congestion. Thirdly, fairness, to ensure fair resource allocation among competing flows. Fourthly, scalability, to handle large-scale datacenter networks with thousands of servers. Fifthly, adaptability, to dynamically adjust to changing traffic patterns and workloads. And lastly, integration with AI/ML, as TCP can use machine learning to predict congestion and optimize flow control and TCP parameters in real time.

Newer TCP algorithms could also leverage programmable switches such as P4 to implement custom congestion control algorithms.This would combine the best features of TCP, RDMA, and newer protocols like Homa, to ensure low latency and high throughput that are needed in futuristic data centers.

To summarise, traditional TCP, while effective for the internet, is not ideal for modern data centers. Newer protocols like DCTCP, TIMELY, HPCC, and Homa are better suited for futuristic data centers. Futuristic data centers will require AI-driven, low-latency, and scalable congestion control algorithms to meet the demands of AI workloads.

I’ve used simulations to test and compare different TCP algorithms.A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generatedA graph with different colored rectangles

Description automatically generated

**Numerical examples and experiments with different number of TCP users/flows sharing a single bottleneck**A screenshot of a computer program

Description automatically generatedA screenshot of a computer

Description automatically generatedA graph with purple line

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA graph with numbers and lines

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated