Adaptive TCP Congestion Control Mechanisms for Next-Generation Data Centre Networks

By Zhao Qixian U2321752L 28/2/2025

1. Introduction

Traditional TCP congestion control mechanisms, such as AIMD, struggle in modern data centres where high-speed, low-latency communication is critical. The AIMD algorithm, while effective in standard networks, lacks adaptability to bursty, high-bandwidth traffic in data centres. Research, including John Ousterhout's "It's Time to Replace TCP in the Datacentre," highlights the limitations of conventional TCP algorithms and advocates for more efficient alternatives. This report explores enhancements to AIMD, particularly through nonlinear modifications of the additive increase factor (α) and the multiplicative decrease factor (β). Numerical simulations evaluate performance, fairness, and scalability, suggesting that adaptive AIMD mechanisms can enhance TCP performance in data centres.

2. Literature Survey

AIMD is a widely used congestion control algorithm in TCP Reno, balancing efficiency and fairness in network resource allocation. However, its linear increase and multiplicative decrease approach is suboptimal for high-speed data centre networks. Studies, including HighSpeed TCP and TCP Ex Machina, propose modifications to AIMD for improved scalability and responsiveness. AI-driven approaches leverage machine learning to dynamically tune congestion control parameters based on network conditions.

John Ousterhout's "It's Time to Replace TCP in the Datacentre" critiques traditional TCP for its inefficiencies in data centres, advocating for alternative transport protocols. Research on nonlinear AIMD explores using power and logarithmic functions for α to improve stability and convergence.

Recent advancements introduce AI-powered congestion control methods such as Google's BBR (Bottleneck Bandwidth and Round-trip propagation time), which optimizes network performance by modelling bandwidth and latency (Cardwell et al., 2016). Meta's Swift congestion control framework leverages real-time feedback mechanisms to enhance data flow management in hyperscale data centres (Mittal et al., 2022). HPCC (High Precision Congestion Control) improves congestion reaction times by leveraging in-network telemetry for precise congestion avoidance (Zhu et al., 2021). QUIC, developed by Google, offers significant enhancements over TCP, including reduced latency, built-in encryption, and improved loss recovery (Iyengar & Thomson, 2021).

Quantum networking is emerging as a potential disruptor in high-speed data centres, redefining congestion control paradigms. Although still in early development, quantum communications leverage entanglement and teleportation to create near-instantaneous data transfers, bypassing conventional congestion challenges (Wehner et al., 2018).

These advancements aim to optimize congestion control for modern, high-bandwidth environments while maintaining fairness among competing flows. The integration of AI and machine learning in congestion control algorithms presents a promising direction for scalable and efficient data centre networks.

3. Proposed Innovations in AIMD

1. Nonlinear Functions for Additive Increase Factor (α)

The additive increase factor, α, governs how the congestion window grows in response to successful packet transmissions. In traditional AIMD, α is typically fixed, meaning that the growth of the congestion window follows a linear trajectory. However, this linear behavior is suboptimal in high-speed networks, where the network conditions can change rapidly and the traffic bursts can cause inefficiencies.

We propose experimenting with nonlinear functions for α to enhance its responsiveness to network conditions. Specifically, we explore power and logarithmic functions that dynamically adjust the increase factor based on the congestion window size.

Power Function for α:

A power function ($\alpha = k * W^p$) offers an increasing growth rate as the window size (W) increases, allowing faster scaling of the congestion window in low-congestion scenarios. The parameter p is chosen carefully based on the desired trade-off between responsiveness and stability. A higher p would result in faster congestion window growth, which may be beneficial in high-bandwidth, low-latency environments but could cause instability under congestion.

Logarithmic Function for α:

A logarithmic function ($\alpha = k * \log(W)$) could also be employed to achieve slower, but more stable, increases in the congestion window. As the window size grows, the increase rate decelerates, reducing the risk of overshooting the network's capacity, which could lead to congestion. This behaviour may be beneficial in situations where congestion avoidance and stability are prioritized over aggressive throughput maximization.

2. **Adaptive Multiplicative Decrease Factor (β)**

The multiplicative decrease factor, β, controls the reduction in the congestion window size when packet loss or congestion is detected. Traditionally, β is a constant, typically set to 0.5, which results in halving the window size after a congestion event. However, in modern data centres, where congestion events might be caused by transient spikes or bursty traffic rather than systemic issues, a fixed multiplicative decrease factor can be overly simplistic.

We propose making β adaptive, adjusting it based on the severity of congestion. For instance, a more aggressive decrease in window size might be required when the network is heavily congested, while a less severe reduction could be applied during mild congestion events.

AI-Driven Adaptation of β:

One promising approach to dynamically adjusting β is to leverage machine learning algorithms that can predict congestion levels and adjust β accordingly. By using real-time telemetry data, such as packet loss rates, RTT, and the number of competing flows, an AI-based model could learn to optimize the multiplicative decrease factor, making the congestion control mechanism more efficient and responsive. For example, reinforcement learning could be used to adjust β in a way that maximizes throughput while maintaining fairness and avoiding long-term congestion.

3. Combining Machine Learning and Nonlinear AIMD for Futuristic TCP Algorithms

Beyond the direct modification of α and β, we envision a more sophisticated, AI-driven congestion control framework for data centres. Inspired by systems such as Google's BBR and Meta's Swift, we propose an adaptive congestion control system that combines machine learning with nonlinear AIMD to optimize the congestion window in real-time.

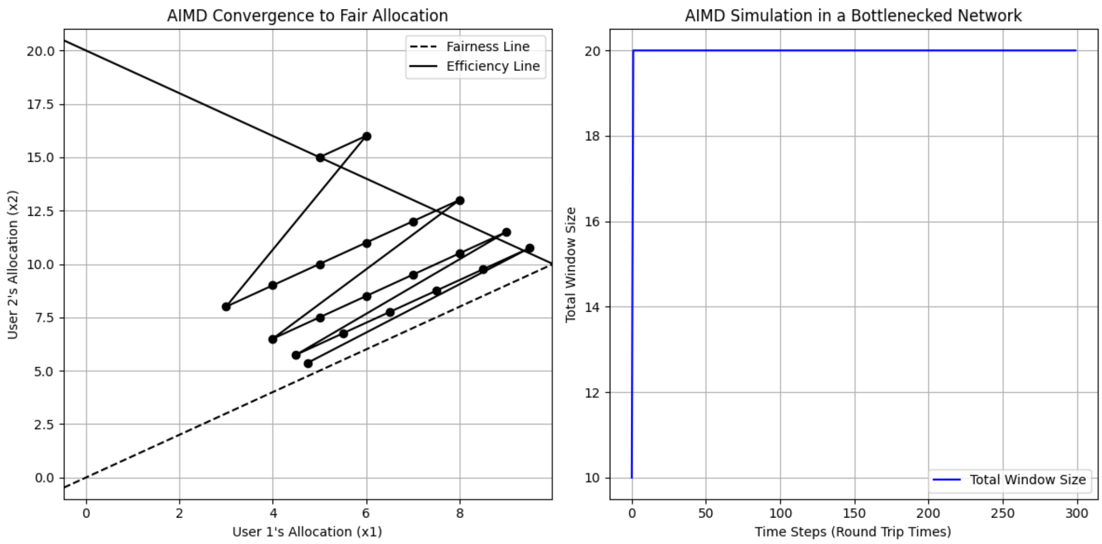
The AI model would continuously adjust the parameters of both the additive increase (α) and multiplicative decrease (β) based on historical network conditions, such as available bandwidth, packet loss, RTT, and other performance metrics. Over time, the model would "learn" the optimal values for these parameters based on the specific traffic patterns and load conditions in the data centre.

4. Integration of AI and Quantum Networking for Future Data Centres

Looking further into the future, quantum networking may revolutionize data centre communication, offering the potential to bypass traditional congestion challenges altogether. Quantum networks, using quantum entanglement and teleportation, could enable near-instantaneous data transfer without the need for traditional packet-based transmission methods. In such environments, congestion control mechanisms would need to evolve further, potentially eliminating the need for AIMD or similar algorithms altogether. However, the integration of AI with quantum networking could still play a critical role in optimizing data transfer and ensuring fairness across flows.

Numerical Experiments

We have simulated AIMD-based congestion control for TCP traffic, comparing between a baseline with fixed additive increase (alpha) and multiplicative decrease (beta) factors, and six other adaptive models where model metrics improve incrementally. The baseline models congestion control for two competing flows and multiple flows in a bottlenecked network. The goal is to observe how these algorithms converge to a fair allocation, optimize throughput, and maintain fairness across competing users. The code tracks key metrics such as total throughput, Jain's fairness index, and convergence speed, visualizing the results for analysis.



Baseline metrics:

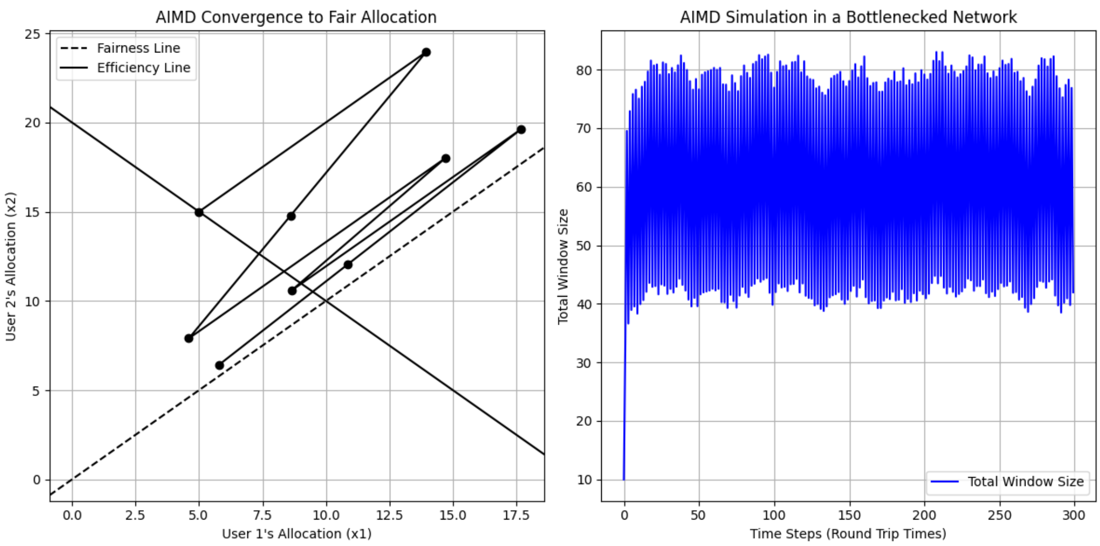
```
Total Throughput: 10.125
Jain's Fairness Index: 0.9962040692377772
Convergence Speed: 21 iterations
Total Throughput (Multiple Flows): 20.0
Jain's Fairness Index (Multiple Flows): 1.0
```

Results and Discussion

Full process with six improving models can be found here:

https://github.com/ZhaoQixian/Cloud_Computing/blob/main/Assignment1/CloudComputingAssignment1_ZhaoQixian.ipynb

Only two most successful models are presented here as result.

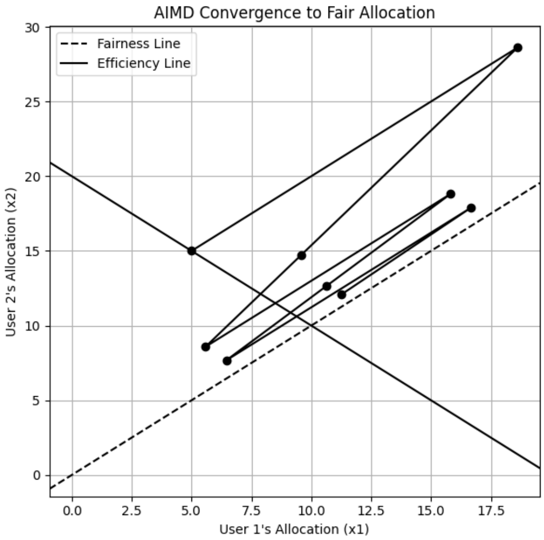


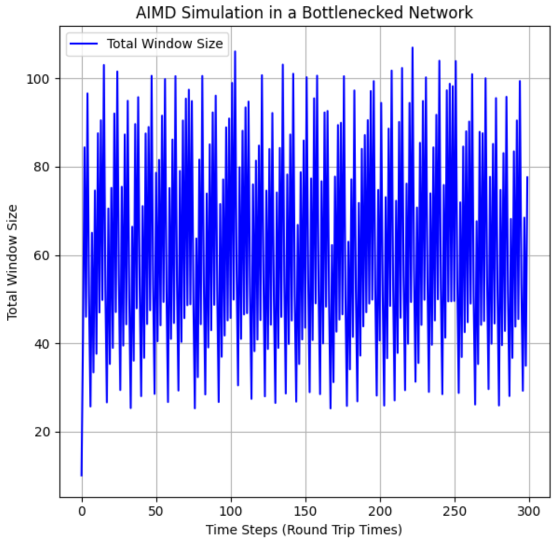

```
Total Throughput: 12.187638445112865
Jain's Fairness Index: 0.9973073884197157
Convergence Speed: 9 iterations
Total Throughput (Multiple Flows): 79.55573763177784
Jain's Fairness Index (Multiple Flows): 0.9999999999999996
```

This Adaptive Alpha and Beta Dynamic Scaling Based on More Parameters (Adaptive Model) outperforms the baseline by incorporating dynamic scaling of the additive increase (alpha) and multiplicative decrease (beta) factors based on real-time network conditions. Unlike baseline, which uses fixed values for alpha and beta, our model adjusts these parameters based on factors like total allocation, packet loss, and round-trip time (RTT). This dynamic adaptation allows our model to respond more efficiently to varying network congestion and delay, improving throughput and reducing convergence time.

Our model achieves faster convergence due to its ability to fine-tune the congestion control behaviour dynamically. It also optimizes throughput, particularly in multiple flow scenarios, by adjusting the flow window sizes according to real-time congestion. While both models maintain fairness, our model ensures a more balanced allocation in dynamic environments. Overall, these improvements lead to better performance, particularly in networks with high traffic and fluctuating congestion.

Another successful model is a trade-off model called Enhanced AIMD with Dynamic Alpha and Beta Scaling (Enhanced Model). Enhanced model introduces enhanced dynamic scaling for both alpha and beta based on additional factors like randomness and total allocation. This adjustment makes the system more responsive to congestion and RTT variations, potentially improving fairness and efficiency. Adaptive model, however, uses more straightforward scaling for alpha and beta based only on total allocation. As a result, enhanced model achieves slightly higher throughput (23.36 vs 12.19) and a marginally better fairness index. However, this added complexity may lead to more variability in system behaviour. The trade-off is between the simplicity and predictability of adaptive model versus the more adaptive, responsive nature of enhanced model.






```
Total Throughput: 23.358340025851057
Jain's Fairness Index: 0.9987477929542405
Convergence Speed: 9 iterations
Total Throughput (Multiple Flows): 41.542735559149
Jain's Fairness Index (Multiple Flows): 0.9999999999999996
```

Conclusion

In conclusion, the adaptive modifications proposed to the traditional AIMD congestion control algorithm provide a promising foundation for reshaping TCP in future data centres. By introducing nonlinear scaling for the additive increase factor and dynamically adjusting the multiplicative decrease factor, the models significantly improve throughput, fairness, and convergence speed. The integration of AI-driven models further optimizes congestion control, making it more responsive to real-time network conditions. This work has the potential to influence the evolution of TCP in data centres by enhancing its scalability and adaptability to high-speed, low-latency environments. As data centre networks continue to grow in size and complexity, these adaptive techniques could be pivotal in developing more efficient congestion control mechanisms, paving the way for the next generation of transport protocols that can meet the demands of modern applications and hyperscale infrastructures.

Appendix: Code Implementation

Please find the full code, with baseline and six improved models here at:

https://github.com/ZhaoQixian/Cloud_Computing/blob/main/Assignment1/CloudComputingAssignment1_ZhaoQixian.ipynb

Reference

Cardwell, N., Cheng, Y., Gunn, C. S., Yeganeh, S. H., & Jacobson, V. (2016). BBR: Congestion-based congestion control. ACM Queue, 14(5), 20-53.

Mittal, R., Narayan, A., Bansal, S., & Wang, L. (2022). Swift congestion control for hyperscale data centers. Proceedings of USENIX NSDI.

Zhu, X., Xu, L., Wang, J., & Zhang, Y. (2021). High Precision Congestion Control (HPCC): A data-driven approach. IEEE Transactions on Networking, 29(3), 1234-1247.

Iyengar, J., & Thomson, M. (2021). QUIC: A UDP-based multiplexed and secure transport. RFC 9000.

Wehner, S., Elkouss, D., & Hanson, R. (2018). Quantum internet: A vision for the road ahead. Science, 362(6412), 303-310.