College of Computing and Data Science

NANYANG
TECHNOLOGICAL
UNIVERSITY

SC4052 CLOUD COMPUTING

Assignment 1 Report

Choh Lit Han Owen
(U2221094K)

Table of contents

Literature Survey

Data centres host a wide range of applications that place different requirements on the network. As with most networks, the network of data centres needs an algorithm to manage this network traffic. However, traditional TCP algorithms seem to struggle with the bursty and high bandwidth nature of the data centre traffic [1], [2]. There are numerous research papers on the effectiveness of TCP in data centres and proposed enhancements to better suit data centre environments. This literature survey reviews three significant contributions in this space.

2.1 Literature survey on “It’s time to replace TCP in the datacenter” [1]

Ousterhout (2022) argues that it is time to replace TCP entirely in data centers, as incremental improvements may no longer be sufficient. His paper critiques TCP's fundamental limitations such as high tail latency caused by TCP’s reliance on packet loss as a congestion signal, inefficiency in handling bursty traffic and many-to-one communication patterns and lack of fine-grained control over congestion feedback.

Ousterhout suggests that alternative transport protocols, such as HOMA, should replace TCP. This perspective challenges the norm of using TCP in data centers and advocates for a clean-slate redesign of transport protocols especially as more applications move into data centres and increase the demand on the network which TCP is unable to manage.

2.2 Literature survey on “TCP ex Machina: Computer-Generated Congestion Control” [2]

Winstein and Balakrishnan (2013) proposed that by using a computer program, we can achieve better network performance than typical algorithms like TCP. By using Remy, data centres would be able to generate optimal congestion control rules based on the defined network and traffic models. Therefore, data centres can automate the design of network protocols to have better performance than current protocols such as TCP and adapt to their specific network conditions.

For futuristic data centres, where traffic patterns are evolving past the limits of current protocols, an algorithm that can optimize based on real-time network metrics is valuable. The advantage that this brings may allow data centres to specialise and be more efficient for specific use cases as they would not be as limited by current network protocols.

2.3 Literature survey on “Data center TCP (DCTCP)” [3]

Alizadeh et al. (2010) introduced Data Center TCP (DCTCP) to optimize congestion control specifically for data centers. On top of implementing a new control scheme, DCTCP uses Explicit Congestion Notification (ECN) unlike TCP which relies on packet loss. This allows DCTCP to detect congestion early to adjust the sender’s transmission rate. Their study found that using DCTCP results in lowered buffer occupancy while maintaining high throughput on top of addressing incast and bursty traffic issues in data centres. Therefore, demonstrating that we can achieve performance gains by modifying congestion control at the transport layer.

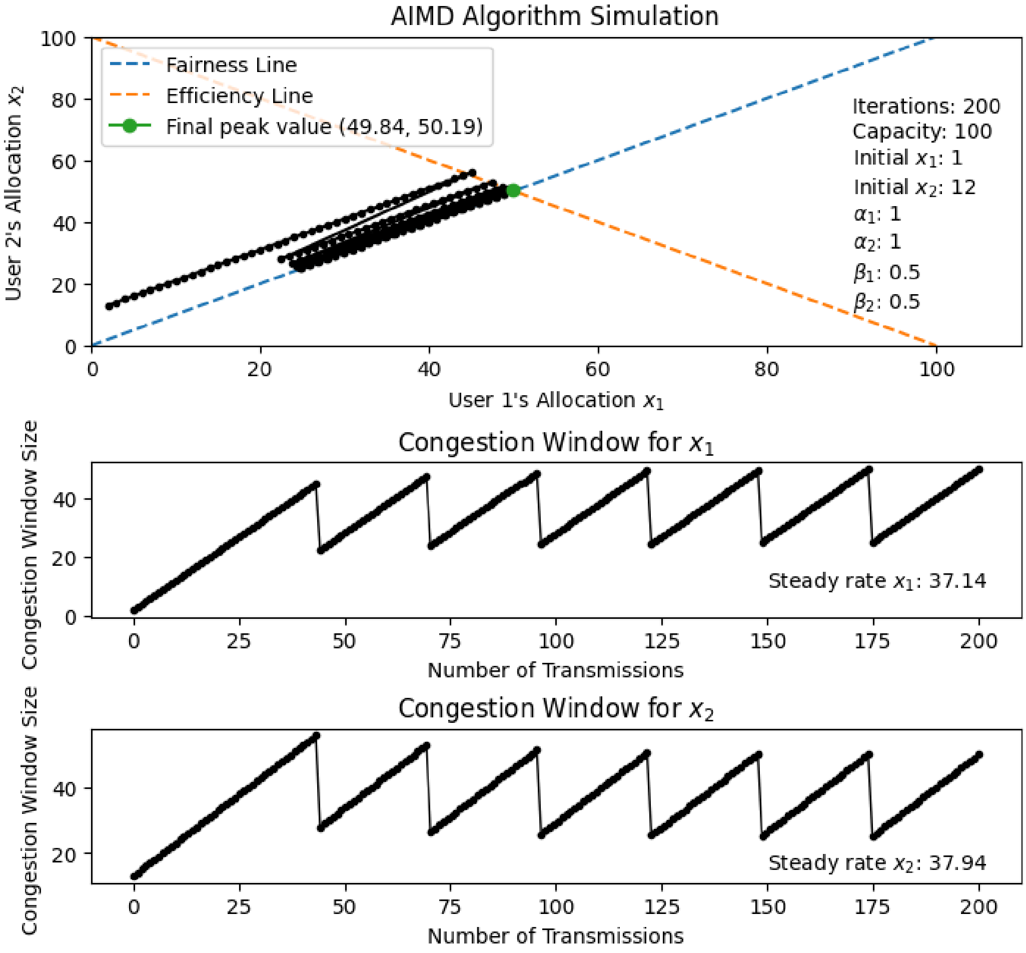
Conclusion of literature survey

The evolution of TCP requires a shift from traditional loss based congestion control to more intelligent and proactive mechanisms. The papers demonstrated that TCP may need to be replaced in favor of computer generated algorithms or changes in the transport layer to make use of the advances in technology such as ECN which are not fully exploited in standard TCP algorithms. Therefore, future congestion protocols should explore integrating new features in network hardware, using machine learning or AI to optimize network traffic and hybrid transport architectures to meet the increasing performance demands of data centres.

Experiments on algorithms for TCP

1. Additive-Increase and Multiplicative-Decrease (AIMD) algorithm

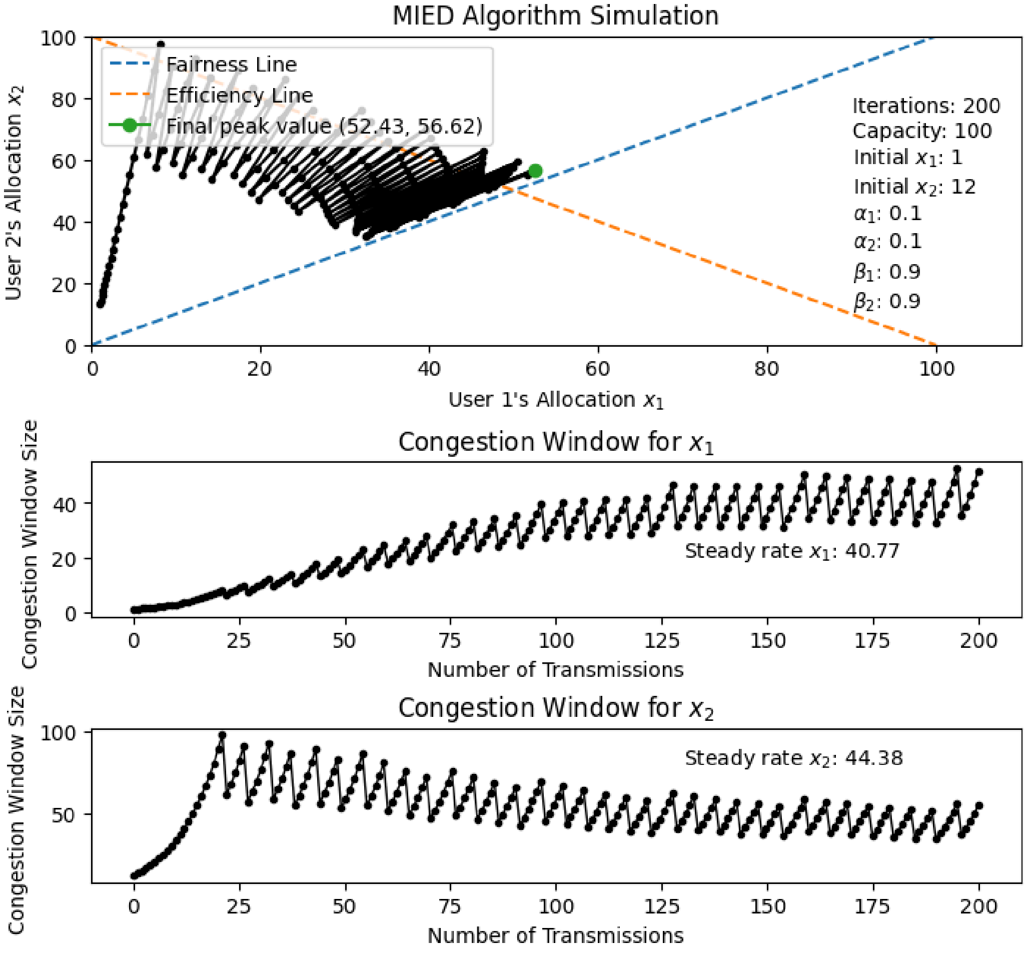
AIMD is an algorithm used in TCP by senders to control the rate at which data is sent over the network. The senders will increase the rate at which they send data (cwnd) by a constant α until congestion is detected. They will then reduce their sending rate by a factor of β. With these parameters, the network traffic will eventually lead to the fairness and efficiency line where all senders get equal share of the network and the throughput is maximised.



From the figure which simulates two senders on the same network, the AIMD algorithm is observed to converge at the fairness and efficiency line. However, the network is not fully utilised as the congestion window for both users is consistently decreasing when it reaches the capacity of the network. Thus, this results in the average rate of transmission when it converges to be 37.14 and 37.94 which is low compared to the optimal window size of 50 each. This is due to the small increase (α=1) compared to the decrease (β = 0.5) which results in more time required to reach network capacity after the window size is decreased.

2. Multiplicative-Increase and Exponential-Decrease (MIED) algorithm

This algorithm aims to address the limitations in AIMD by increasing cwnd multiplicatively $(cwnd = cwnd + \alpha \times cwnd)$ and decrease cwnd exponentially $(cwnd = cwnd^{\beta})$.

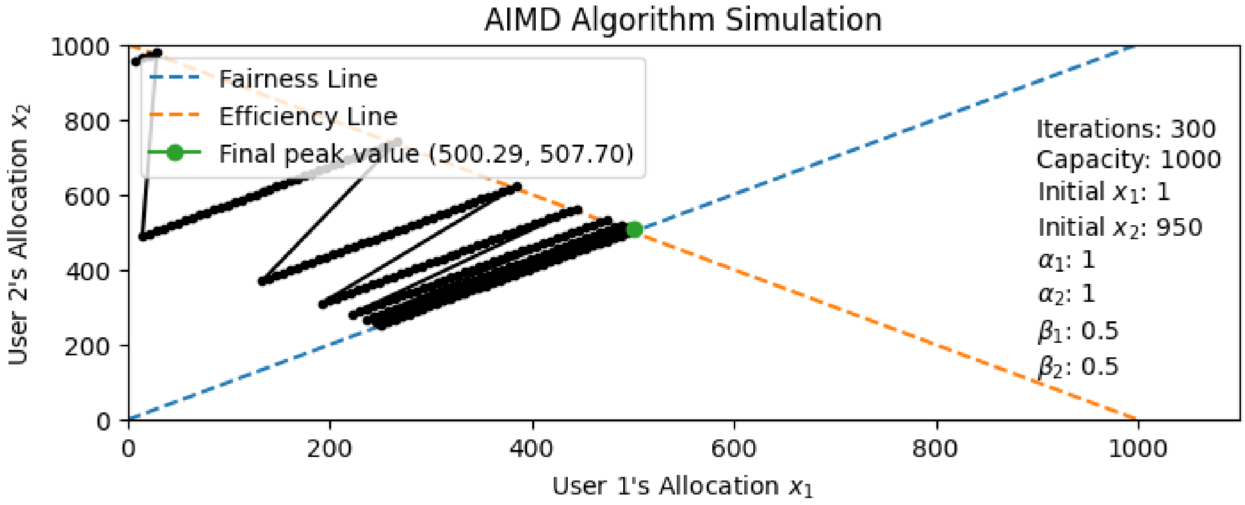


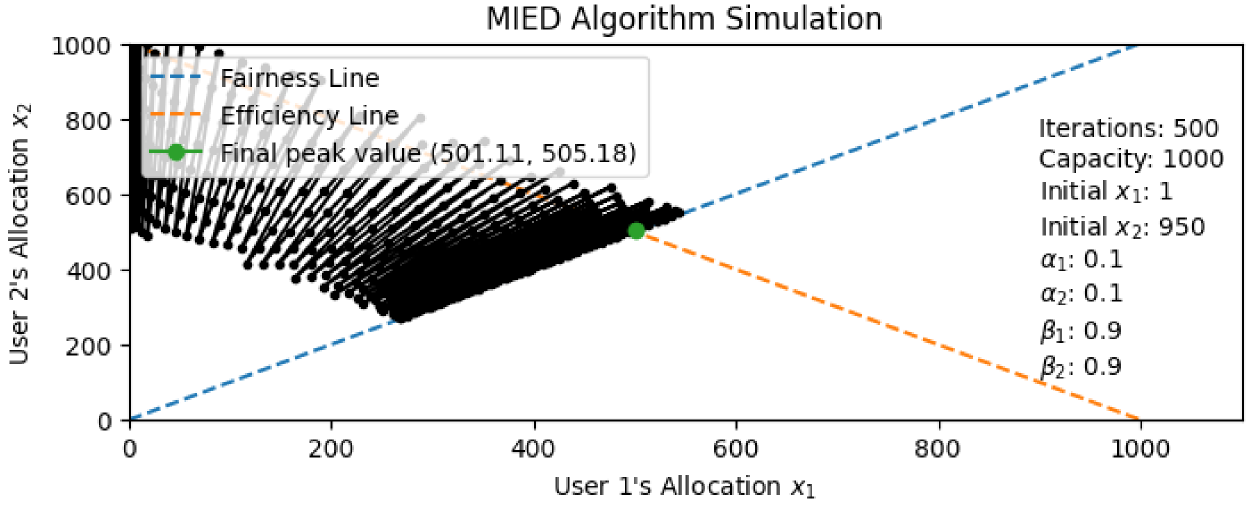
With the same network capacity, number of iterations and initial cwnd as AIMD, the figure above shows that MIED is converging towards the fairness and efficiency line. Additionally, the results show that the network capacity is better utilised as the steady rate of the users is much higher at 40.77 and 44.38 compared to AIMD. However, the figure above suggests that MIED takes a longer time to converge to the fairness line compared to AIMD which gets close to the fairness line at around 100 iterations. Thus, it seems that while MIED may result in a higher efficiency of network utilization, MIED may take longer to result in a fair distribution of network traffic.

Discussion of experiment results

While both AIMD and MIED converges to the fairness and efficiency line, MIED may be preferred in high-speed networking data centres as it reaches the efficiency line much earlier compared to AIMD (22 iterations vs 44 iterations). While the variables in AIMD can be optimised, this optimal value may be difficult to obtain in data centres where there can be many concurrent users and different workloads that can change quickly. Thus the multiplicative increase in MIED would be able to more easily take advantage of the capacity of the network.

However, this may not paint the full picture of the environment in data centres such as the scenario where new users join a network that is near capacity as shown below.


AIMD Algorithm Simulation


MIED Algorithm Simulation

In this scenario, we can make similar observations as the previous experiments. However, the number of iterations taken to be within 5% of the fair value of 500 is more pronounced, (189 for AIMD vs 335 for MIED). This means that MIED may not be ideal in a data centre that has users frequently joining and leaving the network, although there may still be a use case if the greedy behaviour of MIED is desired in order for current users to quickly complete their transmission of messages which frees up the network bandwidth for other users.

Leveraging AI for Adaptive TCP Algorithms

One way to address the trade-offs between AIMD and MIED in dynamic data centre environments can be through AI-driven congestion control as explored in TCP ex Machina [2]. While large language models (LLMs) such as ChatGPT are designed for language processing tasks [4] and are not suited for tuning network parameters, machine learning models such as reinforcement learning agents can be used for adaptive TCP optimizations.

However, LLMs can still support the creation of adaptive TCP algorithms. For example, LLMs could be fine tuned to be used in conjunction with typical machine learning models. This approach could enable network engineers to describe the traffic in the data centre using natural language which an LLM could translate into mathematical models that machine learning models can help to optimise. Alternatively, LLMs may be able to help network engineers by providing insights into the data centre's network and the various workloads which could help network engineers decide the best course of action.

In conclusion, while LLMs may be unable to directly tune network parameters, they can serve as an invaluable supporting role for other tools better suited for this purpose.

Convergence of AIMD

Using Perron-Frobenius Theory and the following positive system [5], we can show that AIMD converges to a numerical solution when $\alpha = 1$ and $\beta = 0.5$ for 3 users.

$$w(k + 1) = Aw(k)$$

Where,

$$A = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} + \frac{1}{1+1+1} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \begin{bmatrix} 1-0.5 & 1-0.5 & 1-0.5 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{2}{3} & \frac{1}{6} & \frac{1}{6} \\ \frac{1}{6} & \frac{2}{3} & \frac{1}{6} \\ \frac{1}{6} & \frac{1}{6} & \frac{2}{3} \end{bmatrix}$$

$$w(0) = \begin{bmatrix} 1 \\ 5 \\ 12 \end{bmatrix}$$

After reaching 13 iterations, we observe that w(13) converges to $\begin{bmatrix} 6 \\ 6 \\ 6 \end{bmatrix}$ for these 3 users despite starting at different window sizes as specified in $w(0)$.

Additionally, the Perron-Frobenius eigenvector of $A = [0.333\ 0.333\ 0.333]$ also shows that AIMD converges to a fair solution.

References

[1] J. Ousterhout, “It’s time to replace TCP in the datacenter,” arXiv (Cornell University), Jan. 2022, doi: 10.48550/arxiv.2210.00714.

[2] K. WINSTEIN and H. BALAKRISHNAN, “TCP ex Machina: Computer-Generated Congestion Control,” in *Computer communication review*, New York, NY: Association for Computing Machinery, 2013, pp. 123–134. doi: 10.1145/2534169.2486020.

[3] M. Alizadeh *et al.*, “Data center TCP (DCTCP),” in *Proceedings of the ACM SIGCOMM 2010 conference*, New York, NY, USA: ACM, 2010, pp. 63–74. doi: 10.1145/1851182.1851192.

[4] University of Central Arkansas “Chat GPT: What is it?” https://uca.edu/cetal/chat-gpt/ (accessed Feb. 24, 2025)

[5] T. Chee Wei, Lecture 2 "Datacenter Networking” SC4052 Cloud Computing, College of Computing and Data Science, NTU, Singapore, Jan, 24, 2025

Appendix

Source Code:
https://github.com/Owen-Choh/SC4052-Cloud-Computing-Assignment-1/blob/main/Assignment_1_experiments.ipynb