NANYANG TECHNOLOGICAL UNIVERSITY

School of Computer Science and Engineering (SCSE)

SC4052: Cloud Computing

Academic Year: 2023-2024, Semester 2

Assignment 1

Teoh Xi Sheng (U2120456L)

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Contents

1. Additive Increase and Multiplicative Decrease (AIMD) for TCP

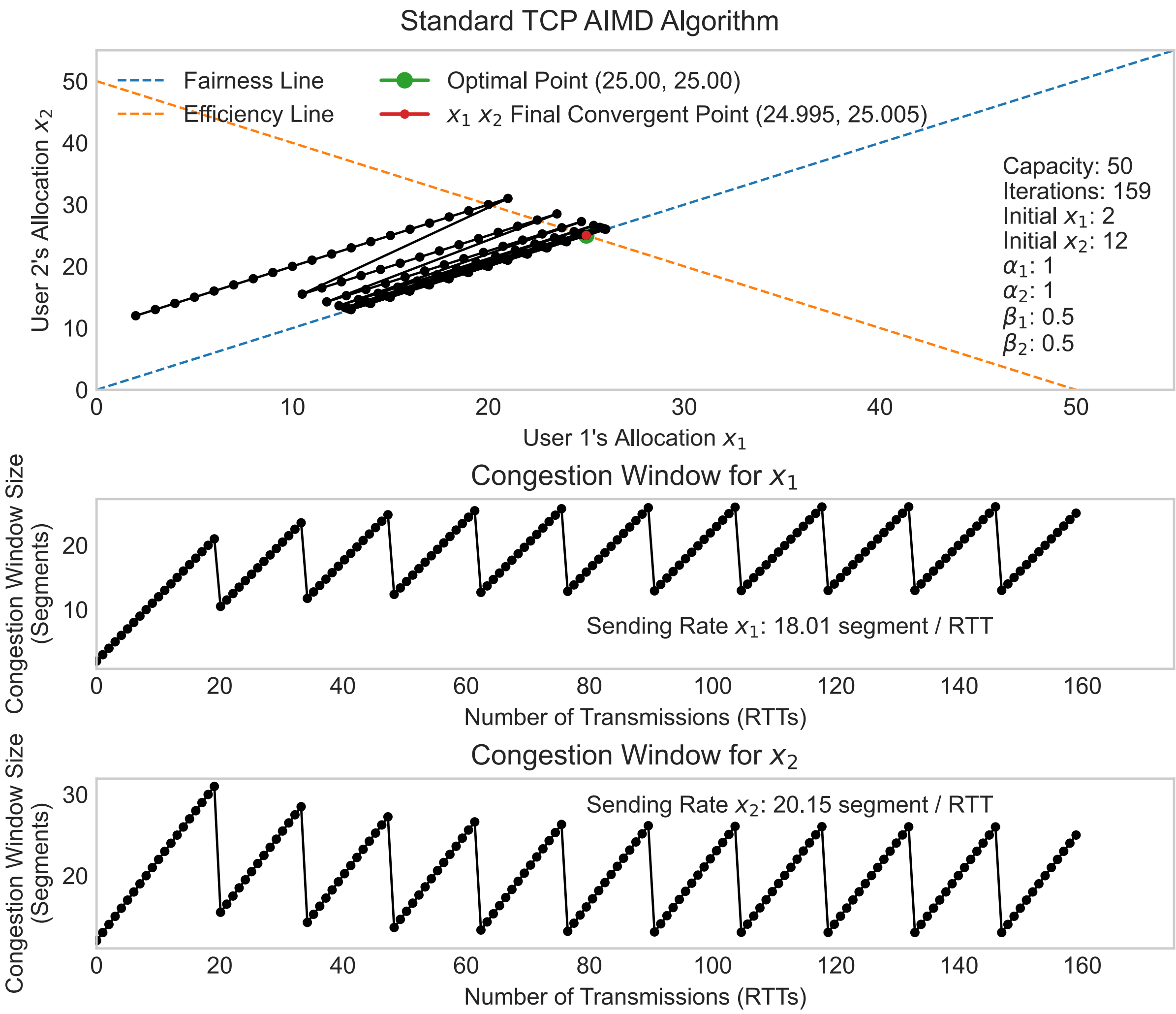
AIMD is a congestion control mechanism used in TCP to control the rate at which data is transmitted over a network from multiple senders. The senders will gradually increase (Additive Increase) their sending rate by (α) until it detects a congestion of the network, (cwnd). The senders will then reduce (Multiplicative Decrease) their sending rate by $(1-\beta)$.

With these parameters, regardless of the size of the congestion window each sender starts with, the AIMD dynamics will eventually lead to a convergence of:

- Fairness, where all senders get equal shares of the network; and
- Efficiency, where all senders are maximising the bandwidth of the same network link.

This can be shown in the figure below with two senders on the same network with the following formula:

$$\mathrm{cwnd}[n+1] = \begin{cases} \mathrm{cwnd}[n+1] = \mathrm{cwnd}[\mathrm{n}] + \alpha, & \text{Addictive Increase where } \alpha = 1 \\ (1-\beta)\cdot \mathrm{cwnd}[\mathrm{n}], & \text{Multiplicative Decrease where } \beta = 0.5 \end{cases}$$



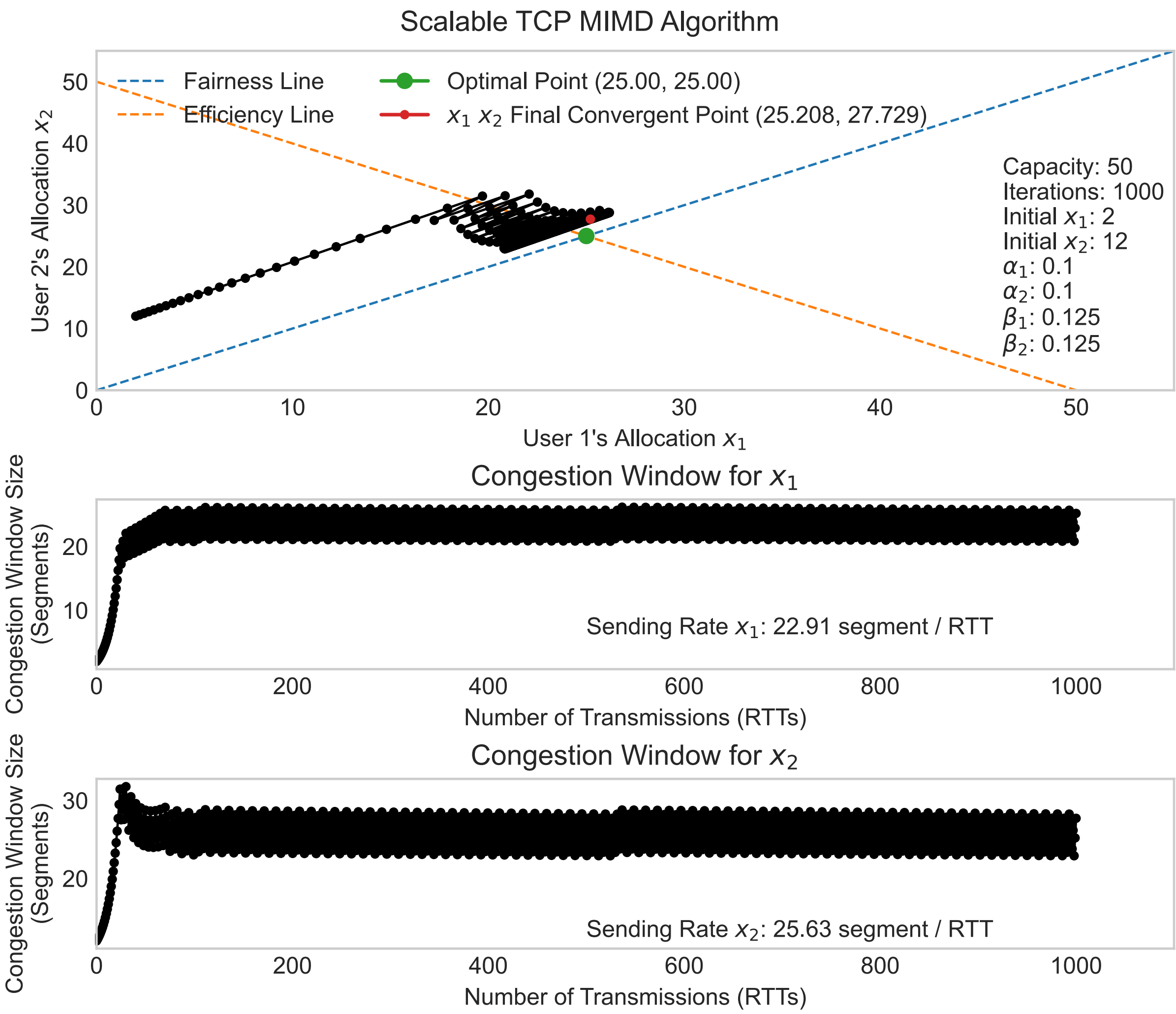
We can observe that the AIMD algorithm eventually converges at a fair and efficient line after 158 iterations. However, the network is not constantly fully utilised; the sum of the average congestion window size per RTT for both senders is only 38.16, which is far from the total network bandwidth of 50 segments.

This is because the AI factor is small ($\alpha = 1$), and the MD factor is large ($\beta = 0.5$), such that it takes longer to reach full network capacity and dropped too aggressively.

2. Multiplicative Increase and Multiplicative Decrease (MIMD) for TCP

Next, we will explore the MIMD variant of TCP, also known as Scalable TCP (SCTP) with the following formula:

$$\mathrm{cwnd}[n+1] = \begin{cases} \mathrm{cwnd}[\mathrm{n}] + (\alpha \cdot \mathrm{cwnd}[\mathrm{n}]), & \text{Multiplicative Increase where } \alpha = 0.1 \\ (1-\beta) \cdot \mathrm{cwnd}[\mathrm{n}], & \text{Multiplicative Decrease where } \beta = 0.125 \end{cases}$$

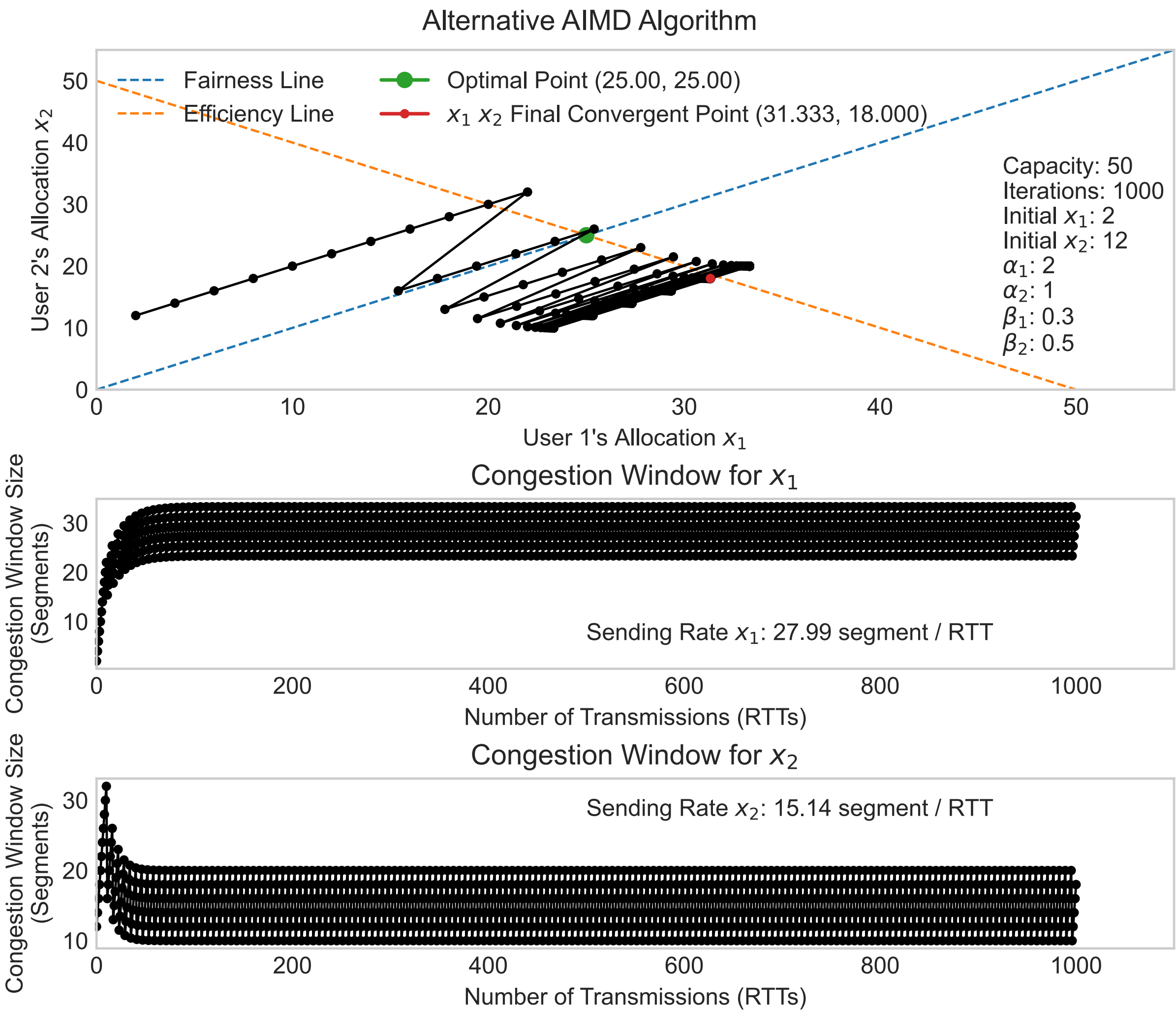


With the same network capacity and initial cwnd as AIMD, we can observe that MIMD does not converge precisely at the optimal point. Still, it significantly increased overall network utilisation with a sum of 48.54 segments/ RTT compared to AIMD. While the network is more utilised, it does not ensure fairness and stability as the congestion window increases exponentially, increasing the frequency of congestion [1].

3. Is the AIMD tuning novel and scalable for high-speed networking in data centre?

While the traditional AIMD with $(\alpha = 1,\ \beta = 0.5)$ converges to efficiency and fairness line, alternative values of α and β can be adjusted to meet the requirements of a network. For instance, we can prioritize different group of user's congestion window based on their subscribed network plan.

To demonstrate this, we can use a different α and β values for other users while maintaining the capacity and initial window sizes.



In this scenario, we can observe that it does not precisely converge, and it prioritise user x_1 that has a higher addictive increase factor but lower decrease factor. The sending rate for each user also varies based on the updated α and β values.

4. Can ChatGPT replace TCP Ex Machina for Data Centres?

TCP Ex Machina is a system designed to run at endpoints to optimise the TCP using analytical and heuristic methods to tailor it to the different usages. It focuses on improving the performance and efficiency of TCP using specific requirements that users require, such as low latency or high throughput [2]. While this approach is similar to reinforcement learning, it differs from what ChatGPT is trying to achieve.

ChatGPT is primarily a language model that is designed for natural language processing tasks, which can be unsuitable for managing data centre networks. However, ChatGPT can be used with TCP Ex Machina to aid engineers in tasks such as network monitoring, or provide insights into the network performance using natural language.

In conclusion, ChatGPT can complement with TCP Ex Machina, but it cannot serve as a direct replacement due to the different functionalities.

5. Perron-Frobenius Theory Approach to AIMD

We can use the following positive system to demonstrate if the AIMD converges with multiple flows.

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

Where:

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

$$w(0) = \begin{bmatrix} 2 \\ 5 \\ 12 \end{bmatrix}$$

After running 50 iterations, we observe that $w(10)$ eventually converges at $\begin{bmatrix} 6.33 \\ 6.33 \\ 6.33 \end{bmatrix}$ with 3 different flows with different starting window size.

The Perron-Frobenius eigenvectors of $A = \begin{bmatrix} 0.33 & 0.33 & 0.33 \end{bmatrix}$ also shows that AIMD ensure fairness.

References:

[1] E. Altman, K. E. Avrachenkov, and B. J. Prabhu, “Fairness in mimd congestion control algorithms,” Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies., 2004. doi:10.1109/infcom.2005.1498360

[2] K. Winstein and H. Balakrishnan, “TCP ex machina,” ACM SIGCOMM Computer Communication Review, vol. 43, no. 4, pp. 123–134, Aug. 2013. doi:10.1145/2534169.2486020

Appendix:

Source Code: https://github.com/xeroxis-xs/NTU_SC4052_Cloud_Computing