Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control
Journal homepage: http://kinetik.umm.ac.id
ISSN: 2503-2267
Vol. 10, No. 4, November, Pp. 583-592

Evaluation of traffic distribution performance of ECMP and PCC+CAKE for Multi-ISP load balancing on real networks using MikroTik

Moh Fathurrohim*[1], Achmad Basuki[1]
Universitas Brawijaya, Indonesia[1]

Article Info

Keywords:
Load balancing, ECMP, PCC, CAKE, Mikrotik, Multi-ISP, Passive Monitoring, Active Monitoring

Article history:
Received: June 23, 2025
Accepted: September 22, 2025
Published: November 01, 2025

Cite:
M. Fathurrohim and A. Basuki, "Evaluation of Traffic Distribution Performance of ECMP and PCC+CAKE for Multi-ISP Load Balancing on Real Networks Based using Mikrotik", *KINETIK*, vol. 10, no. 4, Nov. 2025. https://doi.org/10.22219/kinetik.v10i4.2374

*Corresponding author.
Moh Fathurrohim
E-mail address:
fathurrohim@student.ub.ac.id

Abstract

Imbalance in bandwidth utilization among Internet Service Providers (ISPs) is a major challenge in network management within educational institutions, especially when differences in ISP capacity cause overload on one main path. To address this issue, this study proposes the application of load balancing methods using Equal-Cost Multi-Path (ECMP) and Per-Connection Classifier (PCC) optimized with the CAKE queue type. The implementation is carried out using MikroTik devices, which support the flexible configuration of both methods. Testing is conducted on a real network using a combination of passive monitoring approach—through the analysis of actual traffic and ISP utilization—and active monitoring. The evaluation results show that the ECMP method still produces an uneven traffic distribution, with a tendency to concentrate the load on one path. In contrast, PCC+CAKE is able to distribute traffic more evenly according to the ISP bandwidth ratio. In addition, PCC+CAKE shows more stable performance on throughput, RTT, and jitter, and has very low packet loss. Therefore, PCC+CAKE is recommended as a more effective load balancing method to increase the efficiency of ISP utilization and overall network quality in a multi-ISP environment.

1. Introduction

A robust and reliable network infrastructure is essential for modern educational institutions, supporting all aspects of teaching, learning, and administration. Universities, in particular, must manage diverse user demands across their campuses. An effectively managed network ensures seamless connectivity, facilitates digital learning, and enables access to vital online resources. Without a strong network, educational quality and operational efficiency can be significantly compromised, leading to a diminished experience for both students and faculty. Historically, the university's reliance on a dual-ISP setup, with a 100 Mbps allocation for four buildings and 200 Mbps for a single building, led to the primary ISP becoming overloaded while the backup remained underutilized.

In response to these challenges, the network infrastructure at Universitas STRADA Indonesia was evaluated and upgraded by adopting a three-ISP configuration. This configuration was designed not only to increase the total available capacity but also to enhance system resilience. The three service providers utilized are ISP1 (Biznet) with a capacity of 100 Mbps, ISP2 (CBN) with a capacity of 200 Mbps, and ISP3 (IForte) with a capacity of 200 Mbps. It is important to note that all three connections use dedicated bandwidth, ensuring consistent availability in accordance with the allocated capacity. The selection of three ISPs was based on a redundancy strategy involving physical routing paths. ISP1 (Biznet) and ISP2 (CBN) share the same or adjacent physical installation routes, making them vulnerable to simultaneous disruptions, such as fiber cuts caused by falling trees or construction activities. To mitigate this risk, ISP3 (IForte) was installed through a completely independent physical route. Therefore, if physical disruptions affect the primary routes of ISP1 and ISP2, the university's network connectivity can still be maintained through ISP3, thereby ensuring the continuity of critical services.

Several previous studies have been conducted to compare the performance of various load balancing methods, especially PCC, ECMP, and NTH, with varying results depending on the parameters tested. Research by Saharuna et al. (2020) shows that the NTH method produces more stable throughput among clients, with an average distribution of about 25% per client across two ISPs. However, PCC has advantages in lower delay and jitter, as well as a smaller packet loss value, ranging from 7% to 8%, compared to NTH, which reaches 8% to 9%[1]. Research by Tantoni et al. (2022), using four internet lines and the MikroTik RB2011UiAS-2HnD device, shows that PCC produces the highest throughput, ECMP has the smallest packet loss, and NTH records the lowest delay and jitter[2]. Another study by Pakiding et al. (2021) through GNS3 simulation shows that the PCC method excels in delay (12.95 ms), jitter (13.60 ms), and packet loss (0.24%) parameters. However, in terms of CPU load efficiency, the NTH method is lighter with a consumption of only 32%, compared to ECMP (34%) and PCC (61%). In addition, in terms of traffic distribution, PCC and NTH are able to distribute the load evenly, while ECMP tends to concentrate traffic on one main path[3].

This study focuses on comparing the performance of load balancing methods using Per-Connection Classifier (PCC) and Equal-Cost Multi-Path (ECMP), as well as optimizing the PCC method with the use of the CAKE queue type. The aim is to evaluate the effectiveness of each method in a multi-ISP network scenario with different bandwidth capacities. The parameters used include throughput, round-trip time (RTT), and retransmission for TCP traffic, as well as packet loss and jitter for UDP traffic.

Testing is conducted using two approaches: active monitoring and passive monitoring. Active monitoring is performed using tools such as iPerf3 to generate synthetic traffic, while passive monitoring is performed by capturing real traffic using Wireshark to obtain data from actual user traffic. In addition, passive monitoring also includes bandwidth utilization analysis based on the total data usage from each ISP during the testing period, to measure the efficiency of traffic distribution generated by each load balancing method. This research has the following main contributions:

1. Analyzes the performance of the PCC and ECMP methods in distributing traffic on a multi-ISP network in a campus environment.
2. Proposes optimization of the PCC method with the CAKE queue type to increase fairness and reduce network latency.
3. Conducts testing on the real network of Universitas STRADA Indonesia, using active and passive monitoring approaches to compare the performance of the three load balancing methods more comprehensively. Active monitoring is performed using iPerf3 to measure TCP (throughput, RTT, retransmit) and UDP (packet loss, jitter) parameters, while passive monitoring is performed by recording utilization data from each ISP during testing.

2. Research Method

This research was conducted on the real network of Universitas STRADA Indonesia using MikroTik CCR2004 devices and the RouterOS version 7.17 operating system[4]. The research stages consisted of designing the network topology, implementing load balancing methods, and testing network performance by performing passive monitoring and active monitoring[5].

2.1 Traffic Load Balancing

Traffic load balancing is the process of distributing traffic or data traffic that runs in a network[6]. Load balancing is useful to ensure that no single resource, whether computing or bandwidth, becomes overloaded[7]. Essentially, load balancing works by dividing traffic or data traffic to several resources from congested paths to less congested ones[8].

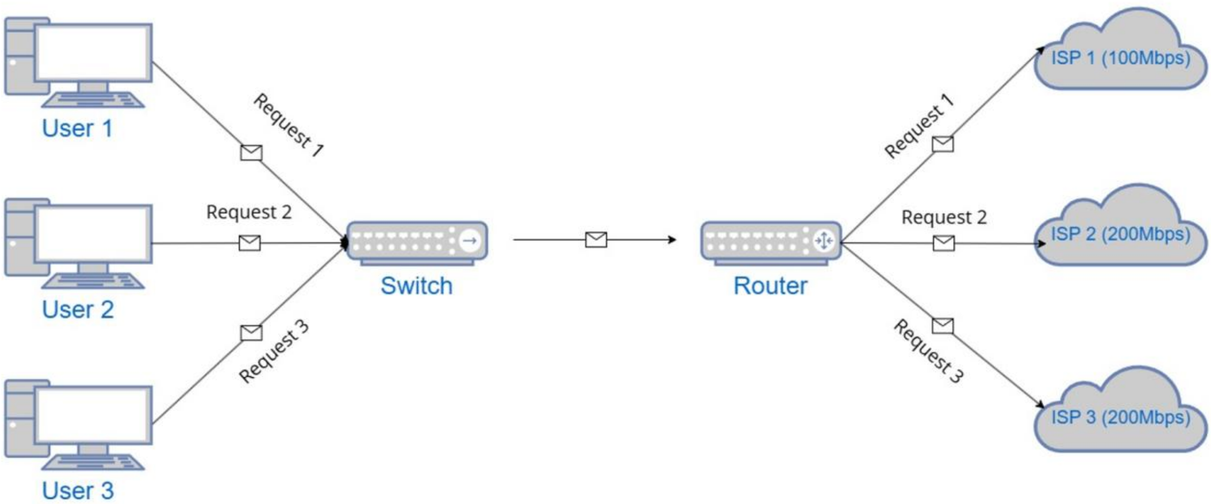


Figure 1. Load Balancing Working Scheme

Figure 1 shows the workflow of the load balancing system in a multi-ISP network. In this illustration, several users (User 1, User 2, and User 3) send data requests that enter through the switch and are forwarded to the router. The router is responsible for distributing these requests to three different ISPs (ISP 1, ISP 2, and ISP 3).

2.1.1 ECMP

ECMP (Equal-Cost Multi-Path) is a load balancing method used in networks to distribute data traffic to several resources, which improves network performance and availability by ensuring that traffic is not concentrated on one path[9]. ECMP works by dividing data to all available paths evenly for each packet sent to improve network performance[10]. Unlike the round-robin per-packet method, which may lead to out-of-order packets and disrupt TCP sessions, modern ECMP employs a per-flow hashing mechanism, ensuring that all packets belonging to the same connection session are consistently transmitted through the same path[11]. This process is achieved by calculating a hash value from the packet header information (5-tuple: source IP, destination IP, protocol, source port, and destination port), which is then mapped to an outgoing path using a modulo operation based on the number of available paths[12]. Configuration is performed on RouterOS version 7.17 and involves interface ether1 for ISP1 gateway, ether2 for ISP2 gateway, and ether3 for ISP3 gateway, as shown in Figure 2. These three entries enable ECMP mode because they use identical parameters and are intended for load balancing outgoing connections based on new connections (per-connection)[13][14]. With this configuration, the router will perform traffic distribution in a round-robin manner across the three paths[15][16]. However, this approach does not consider the bandwidth capacity of each ISP, which can lead to bottlenecks on paths with lower bandwidth in real-world conditions.

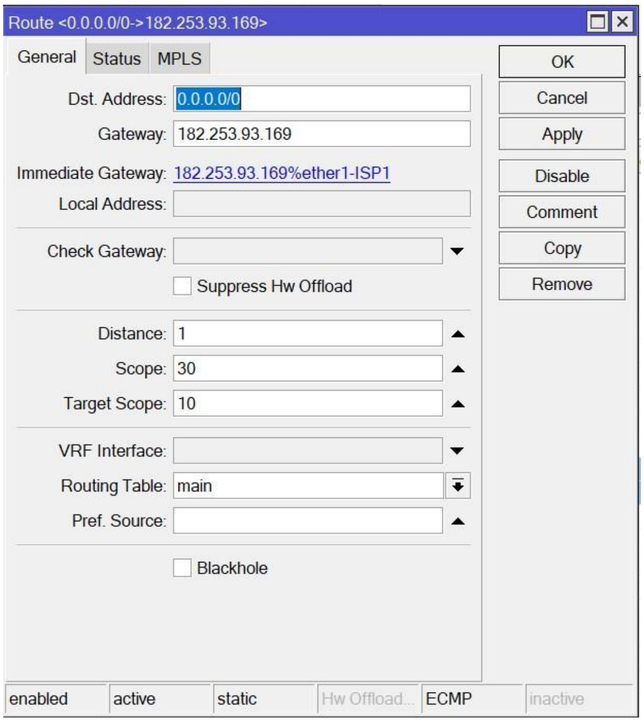
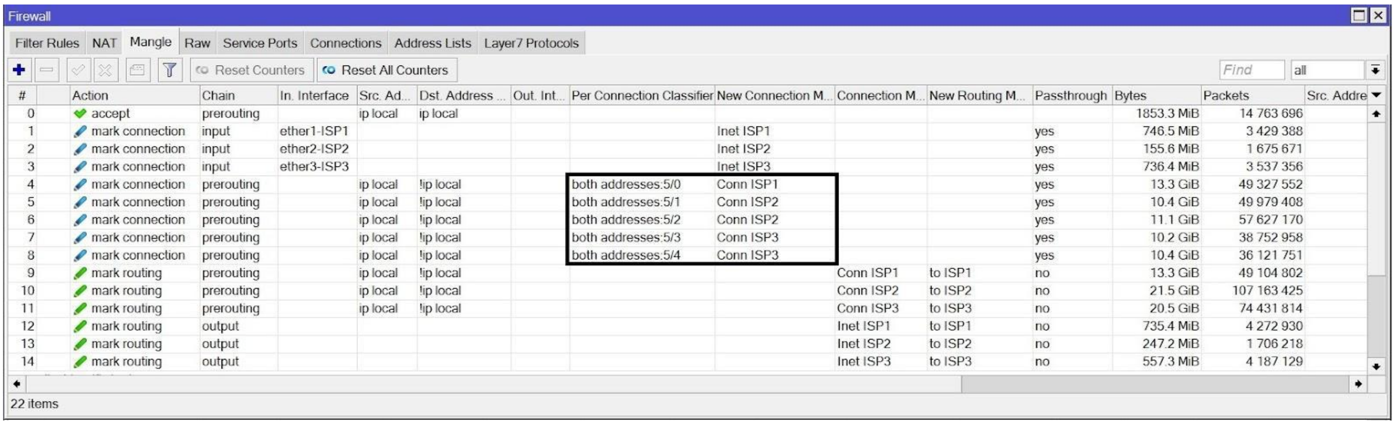


Figure 2. ECMP Configuration

2.1.2 PCC+CAKE

Per-Connection Classifier (PCC) is a load balancing technique that utilizes network connection data to distribute traffic based on connection classification[17]. In a network, PCC is often used to distribute traffic to multiple servers or applications. This can improve network performance and availability by ensuring that the same traffic is sent to the same server or application[18]. PCC operates by performing a hashing process on specific connection attributes, such as source address, destination address, or both, and then utilizing the hash result to categorize the connections into groups (buckets) defined by the PCC parameter, which consists of a denominator and a remainder. The denominator represents the total number of groups into which the connections are divided, while the remainder indicates the specific group assignment, starting from zero. In this study, to achieve a load distribution ratio of 1:2:2 among ISP1 (100 Mbps), ISP2 (200 Mbps), and ISP3 (200 Mbps), a denominator of five was employed. Accordingly, ISP1 was assigned one group (remainder 0), ISP2 was assigned two groups (remainders 1 and 2), and ISP3 was assigned two groups (remainders 3 and 4). Each new incoming connection is hashed, and the result of the hash modulo five determines its group. The connection is then marked (mark-connection), and the marking is subsequently used to assign a routing mark (mark-routing) that directs the traffic through the appropriate ISP gateway. This mechanism enables weighted load balancing with greater precision, ensuring proportional distribution of traffic in accordance with the bandwidth capacity of each ISP. PCC method is implemented using the *mangle* feature on the RouterOS firewall[19][20]. The configuration is based on the PCC parameter with a combination of source and destination addresses (both addresses). Traffic is divided into five parts (five denominators), and each denominator is directed to a different ISP according to the 1:2:2 ratio, namely one part to ISP1, two parts to ISP2, and two parts to ISP3. This ratio is determined based on the bandwidth capacity of each ISP: 100 Mbps, 200 Mbps, and 200 Mbps.

As shown in Figure 3, there are several mark-connection and mark-routing rules used to mark new connections and determine routing paths. These rules ensure that incoming traffic is redirected to the appropriate gateway according to the determined connection classification. This configuration allows for fairer and more efficient traffic distribution according to the path capacity of each ISP.



#	Action	Chain	In. Interface	Src. Ad...	Dst. Address ...	Out. Int...	Per Connection Classifier	New Connection M...	Connection M...	New Routing M...	Passthrough	Bytes	Packets	Src. Addre
0	accept	prerouting		ip local	ip local							1853.3 MiB	14 763 696	
1	mark connection	input	ether1-ISP1					Inet ISP1			yes	746.5 MiB	3 429 388	
2	mark connection	input	ether2-ISP2					Inet ISP2			yes	155.6 MiB	1 675 671	
3	mark connection	input	ether3-ISP3					Inet ISP3			yes	736.4 MiB	3 537 356	
4	mark connection	prerouting		ip local	!ip local		both addresses:5/0	Conn ISP1			yes	13.3 GiB	49 327 552	
5	mark connection	prerouting		ip local	!ip local		both addresses:5/1	Conn ISP2			yes	10.4 GiB	49 979 408	
6	mark connection	prerouting		ip local	!ip local		both addresses:5/2	Conn ISP2			yes	11.1 GiB	57 627 170	
7	mark connection	prerouting		ip local	!ip local		both addresses:5/3	Conn ISP3			yes	10.2 GiB	38 752 958	
8	mark connection	prerouting		ip local	!ip local		both addresses:5/4	Conn ISP3			yes	10.4 GiB	36 121 751	
9	mark routing	prerouting		ip local	!ip local				Conn ISP1	to ISP1	no	13.3 GiB	49 104 802	
10	mark routing	prerouting		ip local	!ip local				Conn ISP2	to ISP2	no	21.5 GiB	107 163 425	
11	mark routing	prerouting		ip local	!ip local				Conn ISP3	to ISP3	no	20.5 GiB	74 431 814	
12	mark routing	output							Inet ISP1	to ISP1	no	735.4 MiB	4 272 930	
13	mark routing	output							Inet ISP2	to ISP2	no	247.2 MiB	1 706 218	
14	mark routing	output							Inet ISP3	to ISP3	no	557.3 MiB	4 187 129	

Figure 3. PCC Configuration

The CAKE (Common Applications Kept Enhanced) queuing discipline is designed to ensure fairness and maintain low latency through three key mechanisms[21]. First, CAKE enforces per-flow fairness by applying hashing and automatic isolation of individual traffic flows, thereby preventing bandwidth-intensive flows, such as large downloads from dominating the queue and degrading the performance of latency-sensitive applications such as VoIP or online meeting, where in triple-isolate mode, flows are separated based on source IP, destination IP, and internal stream identifiers[22]. Second, CAKE integrates the CoDel algorithm as its Active Queue Management (AQM) mechanism, which monitors the sojourn time of packets in the queue and proactively drops packets that exceed the latency threshold. This not only prevents buffer buildup but also signals transport protocols such as TCP to adjust their sending rates, thereby keeping latency consistently low[23]. Third, CAKE includes a built-in traffic shaper that accurately aligns outbound traffic with the actual link capacity, preventing congestion in upstream devices managed by the ISP, which are a common source of bufferbloat[24]. Through these combined mechanisms, CAKE provides equitable bandwidth distribution[25][26], mitigates the dominance of aggressive flows, and maintains a stable and responsive network performance, particularly when integrated with PCC for load balancing[27]. The configuration is performed on RouterOS version 7.17 by adding a new queue type called "cake", as shown in Figure 4.

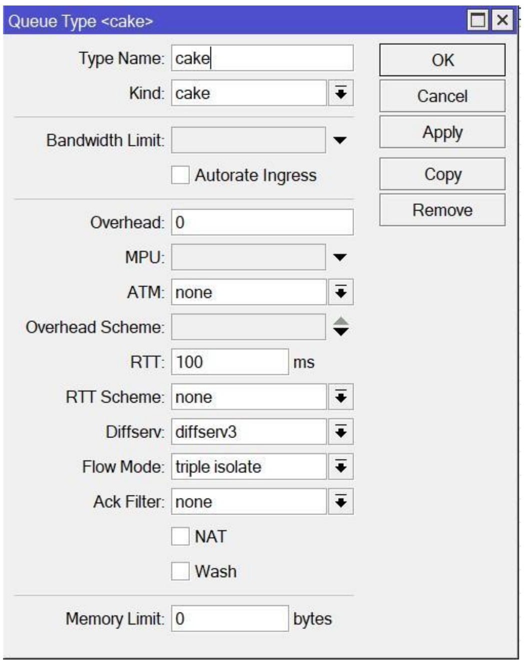


Figure 4. Queue Type CAKE

2.2 Passive Monitoring

Passive monitoring is used to evaluate the effectiveness of traffic distribution between ISPs over a longer period of time by recording the total data usage (upload and download) of each ISP during the five-week test period for both methods, namely ECMP and PCC+CAKE. Data is presented weekly to provide a more detailed picture of the traffic distribution pattern during the observation period. Information is obtained from the interface traffic feature in MikroTik RouterOS, which provides actual volume data that passes through each ISP path. With this data, ISP utilization can be analyzed by comparing the actual bandwidth usage on each path with the available capacity. This evaluation provides a more comprehensive and in-depth picture of the load balancing efficiency of each method, not only from the short-term technical performance (TCP/UDP) side, but also from the long-term traffic distribution side in real operational conditions.

2.3 Active Monitoring

Active monitoring is carried out using the iperf3 application with a connection to a public server with a static IP address 84.17.57.129 located in Hong Kong[28]. The purpose of this test is to determine the effectiveness of each load balancing method (ECMP and PCC+CAKE) in real traffic conditions that occur during working hours. To ensure the results were not influenced by fluctuations in the external ISP backbone or international routing performance, each ISP was asked to optimize the routing path to the destination test server's IP address. Each method is tested for 30 minutes using TCP traffic and 30 minutes using UDP traffic, with tests conducted once per day during working hours from 08:00 to 12:00. This time range was chosen to represent heavy network traffic conditions, making the results more relevant to evaluating the effectiveness of load balancing under actual operational conditions. For TCP testing, measurement parameters include throughput, round-trip time (RTT), and retransmit. While in UDP testing, the parameters measured are packet loss and jitter[29]. All tests are carried out separately for upload and download, with parallel connection settings (multi-stream) to simulate real user traffic[24].

3. Results and Discussion

This chapter presents the results of the implementation and testing of the load balancing methods that have been described in the previous chapter. The discussion begins with the design of the network topology used in this research, followed by the implementation of each method (ECMP and PCC+CAKE), as well as the network performance measurement results, which include TCP, UDP parameters, and bandwidth utilization based on passive and active monitoring. Quantitative analysis is conducted to compare the effectiveness of each method in the real network context of Universitas STRADA Indonesia.

3.1 Passive Monitoring Results

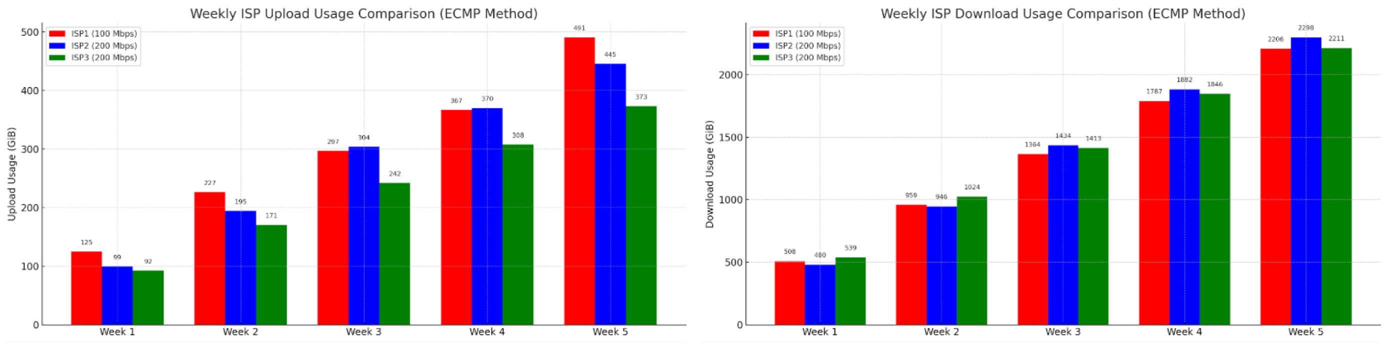
Passive monitoring is carried out with the aim of understanding how actual traffic distribution is divided among each ISP path during natural network usage. Monitoring is performed on the total bandwidth usage (utilization) of each ISP during the five-week testing period. These observations are conducted for two load balancing methods, namely ECMP and PCC. In this case, the PCC method also includes PCC that has been optimized with the CAKE queue type because the CAKE queue implementation is only tested in active measurements using TCP and UDP parameters. Therefore, the analysis of ISP utilization in this section focuses on the effectiveness of load sharing based on real traffic passing through each ISP interface during the weekly testing period.

3.1.1 Utilization ECMP Method

The weekly upload usage graph, depicted in Figure 5(a), shows that the traffic distribution pattern of the ECMP method divides traffic evenly among the three ISP paths, without considering the bandwidth capacity of each path. It can be seen that ISP1 (100 Mbps), ISP2 (200 Mbps), and ISP3 (200 Mbps) have relatively comparable upload volumes from week to week. For example, in the fifth week, the upload volume of ISP1 reached approximately 490.7 GiB, while ISP2 and ISP3 recorded 445.4 GiB and 373.2 GiB, respectively. This shows that ISP1 actually handles more upload traffic than the other two ISPs which have double the bandwidth capacity. Whereas, in terms of ideal capacity ratio, the traffic should be divided in a 1:2:2 proportion, with an allocation of around 20% for ISP1. This condition reemphasizes the weakness of the ECMP method in a multi-ISP scenario with non-homogeneous capacities. Because this method does not take into account the actual capacity of each path, overloading can occur on ISPs with smaller capacity and underutilization on ISPs with larger capacity, which in the long term, can disrupt the stability and efficiency of the network system.

Figure 5(b) shows the comparison of weekly download usage of each ISP during the five-week testing using the ECMP method. Based on the graph, it appears that the download traffic distribution tends to be even among the three ISPs, regardless of the difference in bandwidth capacity. ISP1, which only has a capacity of 100 Mbps, is recorded to handle a traffic volume that is almost comparable to ISP2 and ISP3, which each have a capacity of 200 Mbps. This is a typical characteristic of the ECMP method, which by default divides traffic evenly (round-robin) without considering path capability or capacity.

For example, in the fifth week, the download volume of ISP1 reached around 2205.8 GiB, only slightly different from ISP2 (2297.7 GiB) and ISP3 (2210.9 GiB). Whereas theoretically, if the distribution proportion takes path capacity into account (1:2:2 ratio), the download volume of ISP1 should be in the range of 20% of the total traffic. This imbalance shows that ECMP has the potential to cause over-utilization on ISPs with smaller bandwidth, which in real conditions can cause bottlenecks, delays, or a decrease in service quality, especially during high traffic.



(a) (b)

Figure 5. (a) Upload ECMP (b) Download ECMP

3.1.2 Utilization PCC+CAKE Method

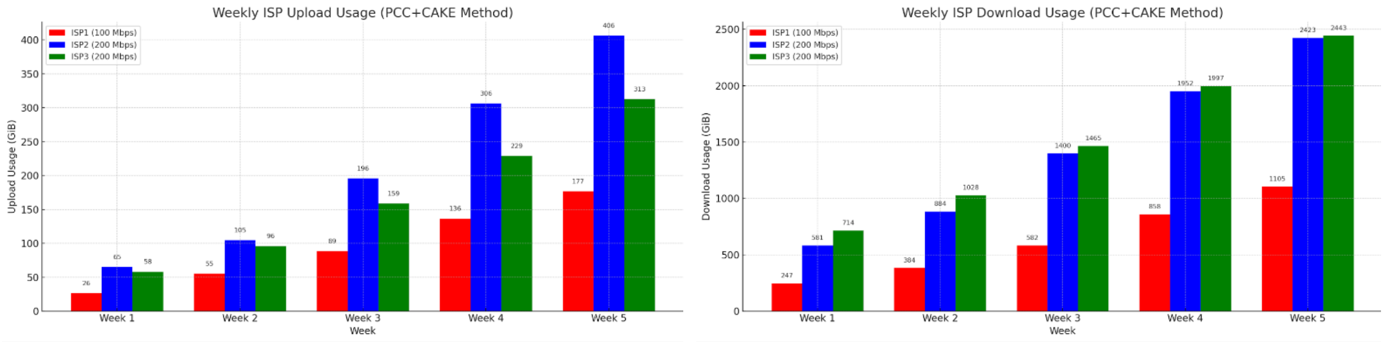
The weekly ISP upload usage graph using the PCC+CAKE method, depicted in Figure 6(a), shows a more proportional traffic distribution pattern compared to the ECMP method. The PCC+CAKE method, which is configured with a 1:2:2 distribution ratio, successfully directs traffic to each ISP more evenly according to the bandwidth capacity. The data distribution across the three ISPs appears consistent and as expected. ISP1, which has a capacity of 100 Mbps, receives a smaller traffic load compared to ISP2 and ISP3, each of which has a capacity of 200 Mbps.

As an illustration, in the fifth week, ISP1 recorded an upload volume of 177 GiB, while ISP2 reached 406 GiB and ISP3 recorded 313 GiB. Although there are slight differences between ISP2 and ISP3, these differences are still within reasonable limits and still show a distribution that is close to ideal according to the 1:2:2 ratio. The consistent distribution pattern from week to week also shows that the PCC method is able to maintain the stability of traffic flow during the testing period. This reflects the advantage of the PCC method in mapping and distributing traffic load based on the characteristics of the available paths.

The weekly ISP download usage graph with the PCC+CAKE method, depicted in Figure 6(b), shows a traffic distribution pattern that is consistent and proportional to the capacity of each path. The PCC+CAKE method, which is configured with a 1:2:2 ratio among ISP1, ISP2, and ISP3, successfully distributes download traffic in an almost ideal manner. From week to week, download usage continues to increase, reflecting stability in traffic management and the ability of this method to handle load growth gradually.

In the fifth week, ISP1 recorded a total download of 1,105 GiB, while ISP2 and ISP3 recorded 2,423 GiB and 2,443 GiB, respectively. These values align with the bandwidth capacity ratio used in the configuration, where ISP1 has 100 Mbps, and ISP2 and ISP3 each have 200 Mbps. The relatively small difference between ISP2 and ISP3 shows a fairly even distribution across paths with equal capacity, while still allocating a smaller portion to paths with lower capacity, namely ISP1.

This pattern continues to repeat from week to week, showing that the PCC method is able to maintain the consistency of traffic distribution even though there are variations in user load. This efficiency also shows that the PCC+CAKE method is not only effective for uploads, but is also reliable in managing download traffic on a multi-ISP network with non-homogeneous bandwidth.



(a) (b)

Figure 6. (a) Upload PCC+CAKE (b) Download PCC+CAKE

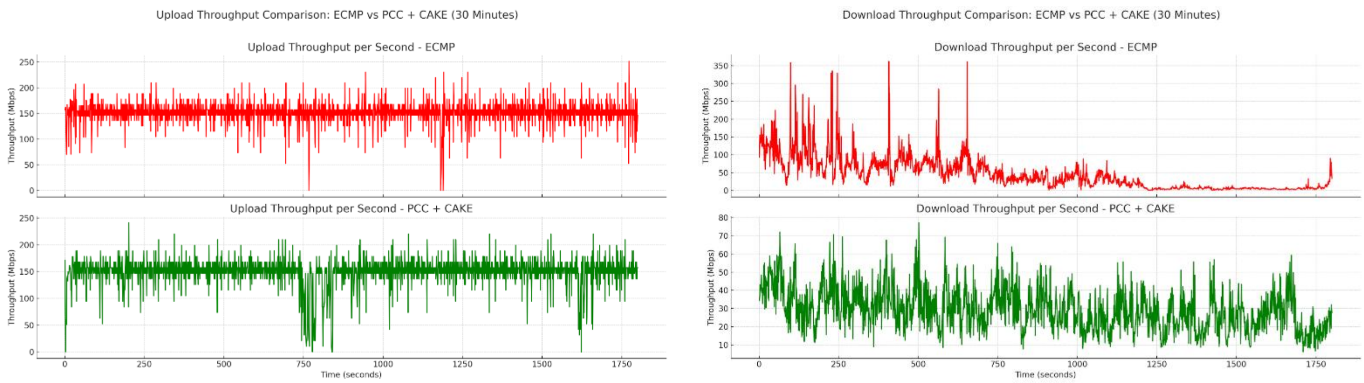
3.2 Active Monitoring Results

Active monitoring using the iPerf3 application is directed at a public iPerf3 server to measure network performance during working hours. The purpose of this test is to determine the effectiveness of each load balancing method (ECMP and PCC+CAKE) in real traffic conditions. Each method is tested for 30 minutes using TCP traffic and 30 minutes using UDP traffic. For TCP testing, measurement parameters include throughput, round-trip time (RTT), and retransmit[30], while in UDP testing, the parameters measured are packet loss and jitter[31]. All tests are carried out separately for upload and download, with parallel connection settings (multi-stream) to simulate real user traffic.

3.2.1 Throughput

The upload throughput evaluation, depicted in Figure 7(a), reveals comparable performance between the two methods. ECMP achieved a mean throughput of 151.12 Mbps (max: 251.81 Mbps), marginally higher than the 148.68 Mbps mean achieved by PCC+CAKE (max: 241.20 Mbps). The minimum recorded throughput of 0.0 Mbps for both methods indicates intermittent transfer stalls.

Conversely, the download throughput results, depicted in Figure 7(b), highlight a significant trade-off between peak performance and stability. ECMP recorded a superior peak throughput of 362.14 Mbps but demonstrated severe instability; its throughput collapsed to 0 Mbps after the 1250-second mark and failed to recover. In contrast, PCC+CAKE delivered a lower but highly consistent throughput (max: 77.14 Mbps, mean: 29.38 Mbps). This finding suggests that while ECMP can achieve higher burst download rates, the predictable stability of PCC+CAKE is advantageous for disruption-sensitive services.



(a) (b)

Figure 7. (a) Upload Throughput (b) Download Throughput

3.2.2 Packet Loss

Figure 8 shows a comparison of packet loss per second between the ECMP and PCC+CAKE methods for 30 minutes. The ECMP graph displays fairly high spikes in packet loss up to 30 packets, while PCC+CAKE is more stable with small fluctuations and no extreme spikes. Statistically, ECMP recorded an average packet loss of 1.55 packets/second with a maximum of 30, while PCC+CAKE only had 0.31 packets/second with a maximum of 6. Although visually PCC+CAKE appears to fluctuate more often, numerical analysis proves that this method is more consistent and effective in minimizing packet loss, making it superior for maintaining the quality of network connections.

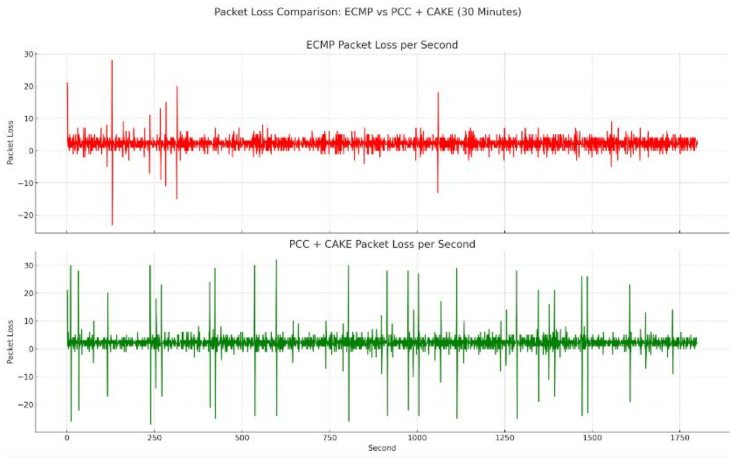
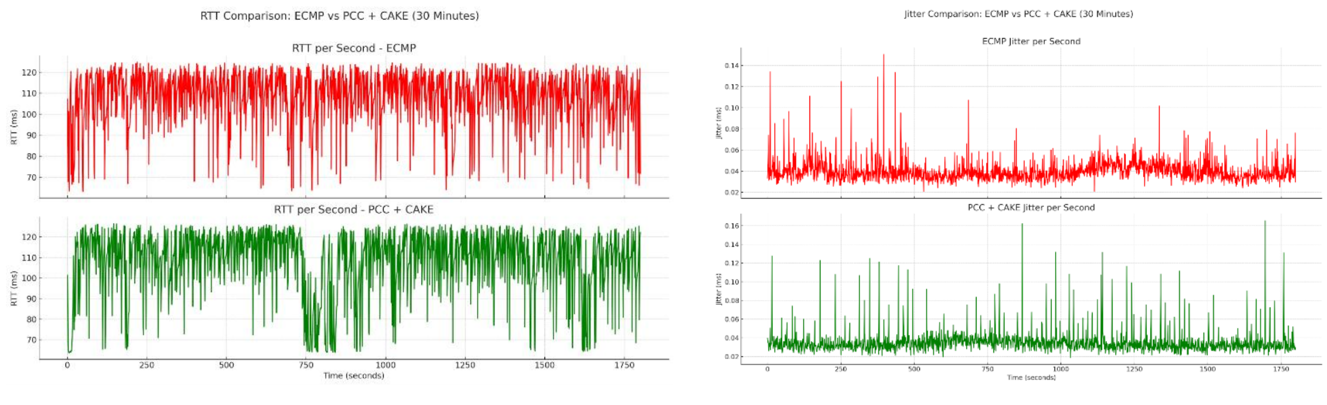


Figure 8. Packet Loss

3.2.3 RTT (Round-Trip Time) and Jitter

The analysis of the research results indicates that the differences in Round-Trip Time (RTT) and Jitter between the ECMP and PCC+CAKE methods are not significant, as illustrated in Figure 9. This outcome is influenced by the quality of the ISP links used, the active testing conditions, and the load balancing mechanisms themselves. Although PCC+CAKE demonstrated slightly better performance, the difference was minimal, with ECMP recording an average RTT of 108.30 ms compared to 108.12 ms for PCC+CAKE, while the average jitter of PCC+CAKE was 0.0352 ms, slightly lower than ECMP's 0.0402 ms. The primary factor contributing to this similarity was the utilization of three dedicated bandwidth ISP connections (Biznet, CBN, and IForte), whose routes to the test server in Hong Kong had already been optimized, thereby ensuring a highly stable baseline network quality and minimizing the influence of the load-balancing method on latency.

Furthermore, the per-flow hashing mechanism in modern ECMP ensures that each connection session remains on the same path, preventing packet reordering that typically increases jitter. The test conditions using iPerf3 focused more on throughput and packet loss, while RTT and jitter are inherently more affected by physical distance and the number of hops to the same destination server used in both methods. On the other hand, CAKE in PCC primarily functions to prevent latency spikes caused by bufferbloat by managing queue lengths and distributing bandwidth fairly, rather than reducing the inherent latency of the communication path. Therefore, although PCC+CAKE proved more effective in stabilizing throughput and ensuring fair traffic distribution, the underlying ISP quality and the nature of latency measurements made the differences in RTT and jitter between the two methods only marginal.



(a) (b)

Figure 9. (a) RTT (Round-Trip Time) (b) Jitter

4. Conclusion

The evaluation of throughput reveals a critical trade-off between peak performance and stability, a factor that significantly influences the quality of experience for end-users. While the ECMP method initially demonstrated a superior peak download throughput of 362.14 Mbps, its performance proved to be extremely unstable, ultimately collapsing to 0 Mbps after the 1250-second mark of the test without recovery. This volatility, characterized by high burst rates followed by a complete failure, renders ECMP unreliable for services that depend on a consistent data stream.

In contrast, the PCC+CAKE method delivered a lower but highly consistent download throughput, with a mean of 29.38 Mbps and a maximum of 77.14 Mbps. Although these figures are lower than ECMP's peak, the key advantage of PCC+CAKE lies in its stability. For real-world applications prevalent in an educational environment, such as video conferencing, VoIP, online learning platforms, and video streaming, a predictable and steady data flow is far more crucial than intermittent bursts of high speed.

The severe instability of ECMP would translate into a poor user experience. For instance, a student participating in an online lecture or a faculty member attending a video conference might experience frequent buffering, frozen video, and dropped calls. These disruptions can severely hamper the educational experience and reduce productivity. The high-peak throughput offered by ECMP has little practical value if the connection is too erratic to sustain these sensitive applications.

On the other hand, the consistent throughput provided by PCC+CAKE ensures a reliable and uninterrupted user experience. This stability is essential for maintaining the quality of real-time communication and streaming services. Therefore, despite its lower peak throughput, the predictable and steady performance of PCC+CAKE makes it a more suitable and advantageous solution for networks where disruption-sensitive services are a priority.

References

[1] Z. Saharuna, R. Nur, and A. Sandi, "Analisis Quality Of Service Jaringan Load Balancing Menggunakan Metode PCC Dan NTH," *CESS (Journal Comput. Eng. Syst. Sci.*, vol. 5, no. 1, p. 131, 2020. https://doi.org/10.24114/cess.v5i1.14629

[2] A. Tantoni, M. T. A. Zaen, and L. Mutawalli, "Komparasi QoS Load Balancing Pada 4 Line Internet dengan Metode PCC, ECMP dan NTH," *J. Media Inform. Budidarma*, vol. 6, no. 1, p. 110, 2022. https://doi.org/10.30865/mib.v6i1.3436

[3] R. Pakiding, C. Iswahyudi, and R. Y. Ariyana, "Simulasi Perbandingan Load Balancing Dengan Metode PCC, ECMP, Dan NTH Menggunakan GNS3," *J. Jarkom*, vol. 9, no. 1, pp. 30–39, 2021.

[4] E. B. P. Manurung and M. Ula, "Comparison Analysis of Bandwidth Speed Using Mikrotik Routerboard 750," *J. Techno Nusa Mandiri*, vol. 20, no. 1, pp. 48–53, 2023. https://doi.org/10.33480/techno.v20i1.4129

[5] F. W. Christanto, S. Susanto, and A. Priyanto, "Load Balancing-Failover Methods using Static Route with Address List, ECMP, PCC, and Nth for Optimizing LAN Network: A Comparison," *Int. J. Commun. Networks Inf. Secur.*, vol. 11, no. 3, pp. 409–416, 2019. https://doi.org/10.17762/ijcnis.v11i3.4340

[6] T. E. Ali, A. H. Morad, and M. A. Abdala, "Load balance in data center SDN networks," *Int. J. Electr. Comput. Eng.*, vol. 8, no. 5, pp. 3084–3091, 2018. https://doi.org/10.11591/ijece.v8i5.pp3084-3091

[7] M. I. Rachmad Anwar and D. Priyawati, "Comparative Analysis of PCC and ECMP Methods in Load Balancing Using GNS3 Simulator," *Sistemasi*, vol. 13, no. 2, p. 458, 2024. https://doi.org/10.32520/stmsi.v13i2.3954

[8] A. Ashipala and J. A. Abolarinwa, "SDN-Enabled Data Offloading and Load Balancing in WLAN and Cellular Networks," *SSRN Electron. J.*, 2023. https://dx.doi.org/10.2139/ssrn.4332902

[9] S. Ali and R. Alubady, "RWRR: Remind Weighted Rounding Robin for Load Balancing in Fog Computing," *ISAS 2023 - 7th Int. Symp. Innov. Approaches Smart Technol. Proc.*, pp. 1–7, 2023, https://doi.org/10.1109/ISAS60782.2023.10391499.

[10] S. Agarwal, R. Kaushik, and S. Dawra, "View of Optimized Load Balancing Using Adaptive Algorithm In Cloud Computing With Round Robin Technique." p. 8, 2024. https://doi.org/10.53555/kuey.v30i2.6847

[11] R. P. Dhanya and V. S. Anitha, "Implementation and Performance Evaluation of Load Balanced Routing in SDN based Fat Tree Data Center," *2023 6th Int. Conf. Inf. Syst. Comput. Networks, ISCON 2023*, pp. 3–8, 2023. https://doi.org/10.1109/ISCON57294.2023.10112200

[12] T. Badageri, B. Hamdaoui, and R. Langar, "Load-Balanced Multipath Routing Through Software-Defined Networking," *20th Int. Wirel. Commun. Mob. Comput. Conf. IWCMC 2024*, pp. 1068–1073, 2024. https://doi.org/10.1109/IWCMC61514.2024.10592316

[13] R. Fahrizal, M. I. Santoso, and M. Z. Arifin, "Implementation Multipath Routing with Equal Cost Multipath (ECMP) and per Connection Classifier (PCC)," *Proceeding - 2020 2nd Int. Conf. Ind. Electr. Electron. ICIEE 2020*, pp. 169–173, 2020. https://doi.org/10.1109/ICIEE49813.2020.9277496

[14] R. Yanwastika Ariyana and V. D. Abdillah, "Network Design Using Load Balancing Equal Cost Multi Path (ECMP) Method At Lexima Consultant Office," *SAGA J. Technol. Inf. Syst.*, vol. 1, no. 2, pp. 31–36, 2023. https://doi.org/10.58905/saga.v1i2.81

[15] R. Dewanto, R. Munadi, and R. M. Negara, "Improved Load Balancing on Software Defined Network-based Equal Cost Multipath Routing in Data Center Network," *J. Infotel*, vol. 10, no. 3, p. 157, 2018. https://doi.org/10.20895/infotel.v10i3.379

[16] P. Lu, C. Jing, and X. Zhu, "GraphSAGE-Based Multi-Path Reliable Routing Algorithm for Wireless Mesh Networks," *Mob. Commun. Handbook, Third Ed.*, pp. 683–694, 2023. https://doi.org/10.3390/pr11041255

[17] W. Wiharti, I. L. Rimra, S. Rifka, I. Hidayatullah, and A. F. Kasmar, "Load Balancing and Fail Over MikroTik Implementation Using Per Connection Classifier (PCC) on Two Internet Providers Interconnection," *Int. J. Adv. Sci. Comput. Eng.*, vol. 5, no. 2, pp. 129–135, 2023. http://doi.org/10.62527/ijasce.5.2.135

[18] M. B. Pratama, A. H. Mirza, Suryayusra, and A. R. Mukti, "Implementation of Load Balancing Using PCC (Per-Connection Classifier) Method with Failover Based on MikroTik Router (Case Study at RSUD Siti Fatimah)," *ARRUS Journal of Engineering and Technology*, vol. 4, no. 2. pp. 151–164, 2024. http://doi.org/10.35877/jetech3112

[19] A. Wahid, S. G. Zain, and J. M. Parenreng, "Implementation of Load Balancing Using the PCC (Per Connection Classifier) Method on Computer Networks to Improve Responsiveness," vol. 04, 2024. https://doi.org/10.31763/iota.v4i4.810

[20] E. R. Amalia, Nurheki, R. Saputra, C. Ramadhana, and E. H. Yossy, "Computer network design and implementation using load balancing technique with per connection classifier (PCC) method based on MikroTik router," *Procedia Comput. Sci.*, vol. 216, pp. 103–111, 2022. https://doi.org/10.1016/j.procs.2022.12.116

[21] A. Jalil, J. Kobayashi, and T. Saitoh, “Optimization Algorithm for Balancing QoS Configuration in Aggregated Robot Processing Architecture,” *Proc. Int. Conf. Artif. Life Robot.*, pp. 928–932, 2023. https://doi.org/10.5954/icarob.2023.gs4-4

[22] M. Seenivasan, R. Ramesh, and R. Dhanabhakkiyam, “Performance Analysis of Queueing Model with Two types of Vacations and Single Service with Feedback,” *2025 5th Int. Conf. Adv. Electr. Comput. Commun. Sustain. Technol. ICAECT 2025*, pp. 1–7, 2025. http://doi.org/10.1109/ICAECT63952.2025.10958997

[23] W. Auriga, Y. Yuhandri, and S. Sumijan, “View of Perbandingan Algoritma Queue Type SFQ, RED,FIFO dan PCQ Pada Jaringan Nirkabel Berbasis Router Mikrotik.pdf.” 2024.

[24] M. Aziz and I. Suharjo, “Common Application Kept Enhanced Pada Mikrotik Routeros 7 Untuk Mengatasi Bufferbloat,” vol. 13, no. 3, 2025. http://dx.doi.org/10.23960/jitet.v13i3.7088

[25] M. Guo, Q. Guan, W. Chen, F. Ji, and Z. Peng, “Delay-Optimal Scheduling of VMs in a Queueing Cloud Computing System with Heterogeneous Workloads,” *IEEE Trans. Serv. Comput.*, vol. 15, no. 1, pp. 110–123, 2022. https://doi.org/10.1109/TSC.2019.2920954

[26] K. T.A., O. O.A., A. O.O., O. O.O., and O. R.O.A., “A Simulation-Based Comparative Study of Controlled Delay (Codel) with Random early Detection (Red) for Network Performance Evaluation,” *Br. J. Comput. Netw. Inf. Technol.*, vol. 6, no. 1, pp. 1–10, 2023. https://doi.org/10.52589/bjcnit-nmhj4s4q

[27] S. Prahara and I. Ali, “Optimalisasi Jaringan Internet Dengan Optimalisasi Load Balancing Menggunakan Parameter QoS (Studi Kasus : SMK Bina Warga Lemahabang),” vol. 7, no. 1, pp. 211–217, 2023.

[28] C. Dumitrache, G. Predusca, G. Gavriloaia, N. Angelescu, D. Circiumarescu, and D. C. Puchianu, “Comparative analysis of routing protocols using GNS3, Wireshark and IPerf3,” *2022 14th Int. Conf. Electron. Comput. Artif. Intell. ECAI 2022*, pp. 1–6, 2022. https://doi.org/10.1109/ECAI54874.2022.9847499

[29] T. Farhan and B. Santoso, “View of Effect of Load Balancing Bonding and Failover on Speed, Latency, Average, and Packet Loss.” Journal of Applied Informatics and Computing (JAIC), 2024.

[30] J. Yao, S. Pang, J. J. P. C. Rodrigues, Z. Lv, and S. Wang, “Performance Evaluation of MPTCP Incast Based on Queuing Network,” *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 2, pp. 695–703, 2022. https://doi.org/10.1109/TGCN.2021.3125860

[31] L. Zhang and L. Xiao, “Performance analysis for a two-stage queuing system with online service,” *IEEE Int. Conf. Ind. Eng. Eng. Manag.*, vol. 2020-Decem, pp. 878–882, 2020. https://doi.org/10.1109/IEEM45057.2020.9309758