

Congestion Control Approach During Transient Link Cost Updates in IP Networks

Simon Tembo^{1,*}, Ken-Ichi Yukimatsu², Shohei Kamamura³

¹Department of Electrical and Electronic Engineering, University of Zambia, Lusaka, Zambia

²Department of Computer Science and Engineering, Akita University, Akita-shi, Akita, Japan

³Department of Computer Science, Seikei University, Musashino-shi, Tokyo, Japan

Abstract When forwarding paths in an IP network are altered due to link failures or cost updates, preventing transient loops and link congestion is crucial for maintaining optimal network performance. To address the challenge of transient link congestions during simultaneous link cost updates—a significant challenge for ISPs, especially large-scale ones engaged in traffic engineering—we propose the *Priority Link Cost Update Sequence Order* method. This approach prioritizes updates on heavily loaded links, redistributing traffic more efficiently to reduce peak link loads and stabilize performance within capacity limits. The core of our algorithm lies in identifying these heavily loaded links and systematically directing traffic to less congested alternatives, effectively mitigating transient congestion and ensuring stable network operations during the update process. The results highlight its potential to enhance reliability and sustain service quality under dynamic link cost changes.

Keywords Transient Loops, Link Costs, High Load Link, Maximum Load, Link overflow, Congestion Control

1. Introduction

IP networks are foundational to modern communication systems, enabling efficient data transfer between devices across vast and distributed infrastructures. These networks use protocols such as Open Shortest Path First (OSPF) to determine optimal routes for packet delivery based on link costs, which represent the cost of transmitting data across network links. In such systems, dynamically updating link costs during periods of congestion or evolving traffic patterns presents a significant challenge for network operators, particularly in ensuring the avoidance of transient forwarding loops and maintaining high-quality performance metrics such as delay, connectivity, and throughput.

A review of the literature highlights various approaches and strategies for link cost optimization in OSPF-based IP networks. Previous studies have shown that progressively increasing (equation 1) the Interior Gateway Protocol (IGP) link cost of a congested link can effectively remove it from traffic routing while avoiding transient loops [1,3,4,8,9,14]. Research also indicates that systematically increasing the link cost to a sufficiently large value ensures that the link can be safely shut down without disrupting network stability [1,9]. Additionally, minimizing the maximum link load

across a network has been identified as a critical objective in optimizing network performance and resource utilization [1,2,9,22]. Several studies have proposed traffic engineering strategies to reroute traffic dynamically, often leveraging Shortest Path First (SPF) computations to determine alternative paths and distribute traffic more evenly [10,16]. Despite these advancements, challenges remain in developing methods to manage link overflow effectively during transient link cost updates.

$$m_l = m_0 + 1 \quad (1)$$

where m_0 is the initial link cost and m_l is the new cost.

In this paper, we present an innovative approach to control link overflow during transient link cost updates in IP networks. Our method employs a brute-force algorithm to generate permutations of link cost update sequences and identify patterns that maintain loop avoidance and network stability. Using matrix operations, we represent the initial topology (\mathbf{a}_{ij}), the required link cost updates (\mathbf{b}_{ij}), and the target topology (\mathbf{z}_{ij}), which corresponds to the state of 100% optimum routing. By adhering to loop avoidance rules [1,9,11,17], we iteratively update link costs to achieve the desired target topology, as illustrated in Figure 1.

To evaluate the effectiveness of our approach, we use the network's maximum load as a key performance metric. Our goal is to maintain maximum link loads below capacity, thereby preventing link overflow. In cases where maximum load exceeds link capacity, we propose rerouting traffic to alternative paths and updating link costs based on a Priority

* Corresponding author:

simon.tembo@unza.zm (Simon Tembo)

Received: Dec. 4, 2024; Accepted: Dec. 22, 2024; Published: Dec. 28, 2024

Published online at <http://journal.sapub.org/ijnc>

Link Cost Update Sequence Order. This iterative process ensures optimized traffic distribution while adhering to Service Level Agreements (SLAs) established by Internet Service Providers (ISPs) to guarantee essential performance metrics [5,6,7,10].

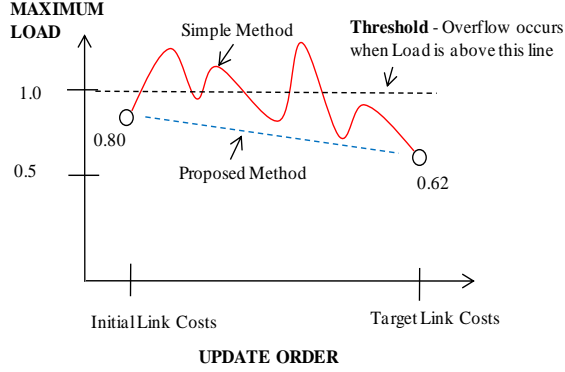


Figure 1. Model of Performance Evaluation

By integrating these techniques with OSPF-based routing strategies, this study contributes to advancing traffic engineering methods in IP networks. Our findings demonstrate how specific network conditions can be strategically influenced to minimize link load, mitigate congestion, and ensure stable and efficient data transfer in dynamic traffic environments. The proposed approach addresses existing gaps in managing transient link cost updates and provides a robust framework for enhancing network reliability and performance.

2. Problem Statement

This section provides an illustrative example to demonstrate how link overflow can occur during the transient process of updating link costs. As shown in Figure 2, employing the route optimization method [2,3], the issue of link overflow on link 3-7 during the transient state can be avoided by carefully determining the sequence of link cost updates.

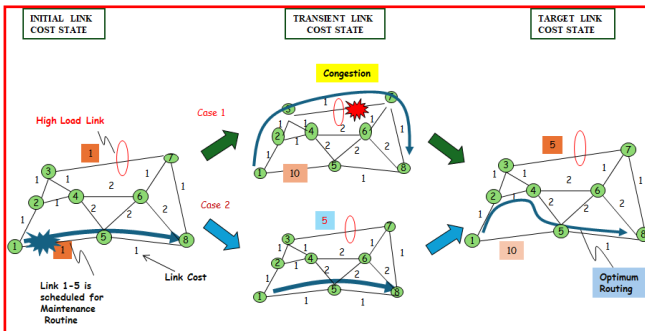


Figure 2. Simple Model -To Illustrates Permutation Problem

In the example, the best solution is first updating link 3-7 and then update link 1-5 to avoid link overflow during the transient state. Our method moves traffic from congested links to other parts of the network to attain load balancing condition.

Table 1 show that the permutation results become very huge as the number of required updatable links increases. For instance, the permutation for COST 239 [4,12,15] with 6 different updatable links is 720.

Table 1. Examples of Permutation Results for Updatable Links in COST239

	TOPOLOGY	NODES	LINKS	UPDATABLE LINKS	PERMUTATIONS
1	COST239	11	50	6	720

In this paper, we present a Link Cost Update Sequence Order (LCUSO) method to update link costs to control congestion until reaching the target link costs. The use of our priority sequence update algorithm minimizes the maximum link utilization during transient link cost updates thus avoids link overflows. Finding a sequence order for link cost update which is transient link overflow free for huge permutations is not a practical computation time. If the link cost updates are not performed carefully, can cause link overflows during the transient period. In this paper, we present an algorithm, LCUSO, to find a sequence order of link cost update which is link overflow free, so that we eliminate the need for a full search. The objective is moving traffic from congested links to other parts of the network, as illustrated in Figure 2, to attain load balancing. With known traffic demands, the load balancing is formulated as an optimization problem.

In this paper, to demonstrates the effectiveness of our congestion control method, the **Initial Topology** displays network conditions of congestion state, i.e. with Link overload problem before we use our proposed, LCUSO congestion control method to reach the link cost update of the **Target Topology** with optimum routing condition.

3. Proposed Link Cost Update Sequence Order Method

This section provides an overview of the application of our algorithm to control link overflow in IP networks. The proposed *Link Cost Update Sequence Order* (LCUSO) method is a targeted approach that effectively addresses the challenge of congestion by minimizing the maximum load across the network through strategic link cost updates.

At the core of the LCUSO method is the precise identification of **High Load Links** and **High Load Nodes**, which are critical in determining areas of congestion within the network. Once identified, a **forwarding tree** is constructed to strategically select specific links connected to **leaf nodes** for cost adjustments. By carefully redistributing traffic away from overloaded links to underutilized ones, the algorithm achieves a balanced load distribution across the network.

The LCUSO method's efficiency lies in its ability to make dynamic and accurate adjustments based on real-time network conditions. This ensures not only the alleviation of link congestion but also the optimization of overall network performance, highlighting the reliability and practicality of the approach in modern IP network management.

3.1. Overview of the LCUSO Method

We provide an overview of our approach. The input data comprises the network topology with predefined initial link costs, specified target link costs, and the traffic matrix. Utilizing the Brute Force method, we generate permutations of link cost updates to identify sequence order patterns. To guarantee compliance with the loop avoidance condition [1,3,4,8,11,13,20], we employ matrix addition rules ($\mathbf{a}_{ij} + \mathbf{b}_{ij} = \mathbf{z}_{ij}$) to execute the link cost updates during matrix operations.

4. Performance Evaluation

4.1. Simulations Conditions

In our evaluation, we use the COST 239 topology model (i.e. an actual European network) [12,15] illustrated in Figure 3 and updatable links (Table 1). In this paper we set the link capacity to be uniform for all links in the network model. We use a gravity model type of traffic demand according to the population distribution.

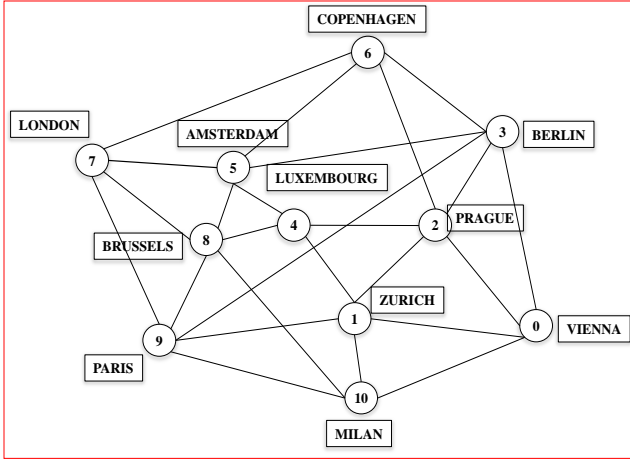


Figure 3. European Network Model - COST 239 [12,15]

5. Results and Discussion

In this paper, we define the optimum routing when we achieve the minimum maximum link utilization during the transient period of link cost updates. Link overflow occurs when the transient maximum load exceeds that of the initial state. We use Brute Force approach to determine the update sequence order for all permutations results.

5.1. Results with Link Overflow

In this paper, we aim to validate the effectiveness of our approach to controlling network congestion. To achieve this, we begin our simulations with a network in an initial state of congestion. To demonstrate the robustness of our method, we examine two distinct network congestion conditions, with optimum routing performances of 12.5% and 0%, as illustrated in Figures 4 and 5, respectively.

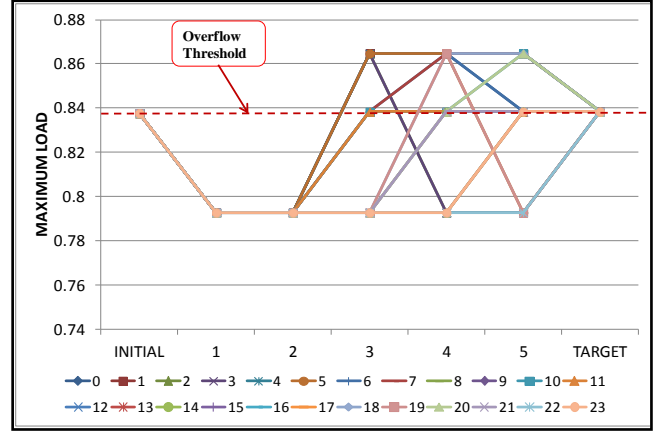


Figure 4. Congestion Condition occurs with 12.5% Optimum Routing

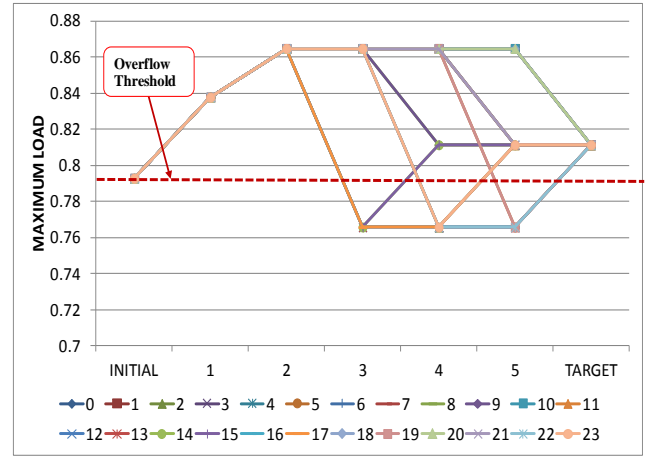


Figure 5. Congestion Condition 0% Optimum Routing

The occurrences of congestion under these conditions, during transient link cost updates, reveal 24 patterns corresponding to the permutations of all possible link cost updates. These scenarios highlight how link overflow arises under different configurations.

Our evaluation focuses on mitigating link overflow in already congested networks (Figures 4 and 5) by performing targeted link cost updates. The study emphasizes whether specific network conditions can be strategically influenced to control or reduce link overflow, thereby enhancing overall network performance.

5.2. Use of Random Selection Method to Select Links for Link Cost Updates

In this paper, we employ the "Random Selection Method to select Links for Link Cost Updates," as a process in which network links are chosen at random for cost updates. This approach does not adhere to any specific pattern or criteria, relying instead on randomness to determine which links are updated. The main objective of this method is to introduce a stochastic or unbiased element into the link update process, simplifying implementation in some cases and providing a baseline for evaluating the performance of the proposed LCUSO method.

In this section, we demonstrate the effectiveness of both congestion control methods under existing network congestion conditions by updating link costs for both the Random Selection Method and the LCUSO Method. We start by highlighting the effectiveness of the Random Selection approach.

Following this, we present the link selection method for updates, designed to efficiently manage link overflow during transient link cost updates, thereby validating the implementation of our proposed LCUSO method.

We conducted simulations using the Random Selection method for both network scenarios, with initial congestion levels as depicted in Figures 4 and 5. Despite using Random Selection to mitigate congestion, the evaluation after the simulation indicated that link overflow occurred during the transient phase, as shown in Figures 6 and 7.

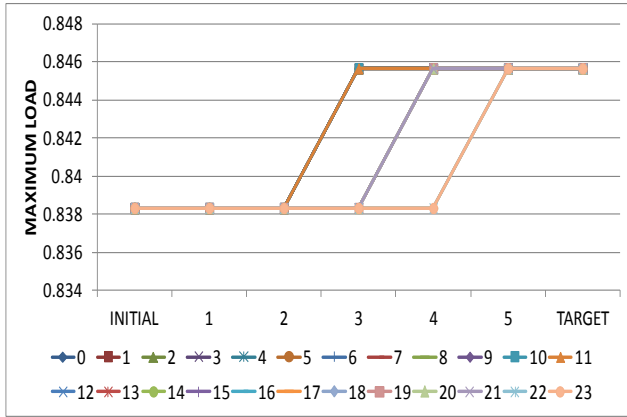


Figure 6. Results for Random Selection Method as a Congestion Control of Congestion in Figures 4

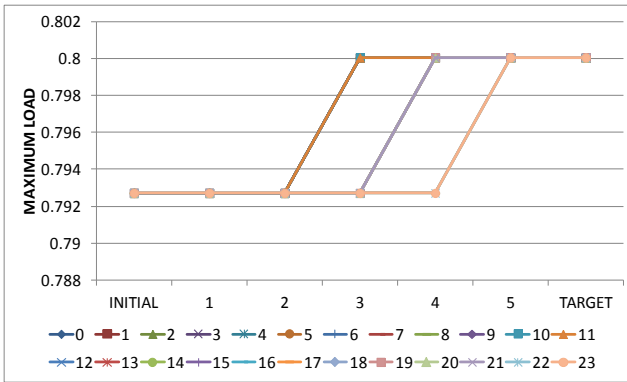


Figure 7. Results for Random Selection Method as a Congestion Control of Congestion in Figure 5

5.3. Use of Link Cost Update Sequence Order (LCUSO) Method to Select Links for Link Cost Updates

5.3.1. Algorithm Overview

In this section, we introduce the algorithm developed to efficiently prevent link overflow during transient link cost updates, highlighting the implementation of our proposed *Link Cost Update Sequence Order (LCUSO)* method. This method effectively identifies links for cost updates, significantly

reducing the maximum load in the network. The approach focuses on detecting the High Load Link and High Load Node, and subsequently building a forwarding tree that prioritizes cost updates for links connected to leaf nodes. This ensures efficient redistribution of traffic from overloaded links to alternate paths within the network. The LCUSO method, therefore, offers a systematic and precise process for selecting links for cost updates. Figure 8 presents the algorithm's flowchart, outlining each step and emphasizing the robustness of our approach.

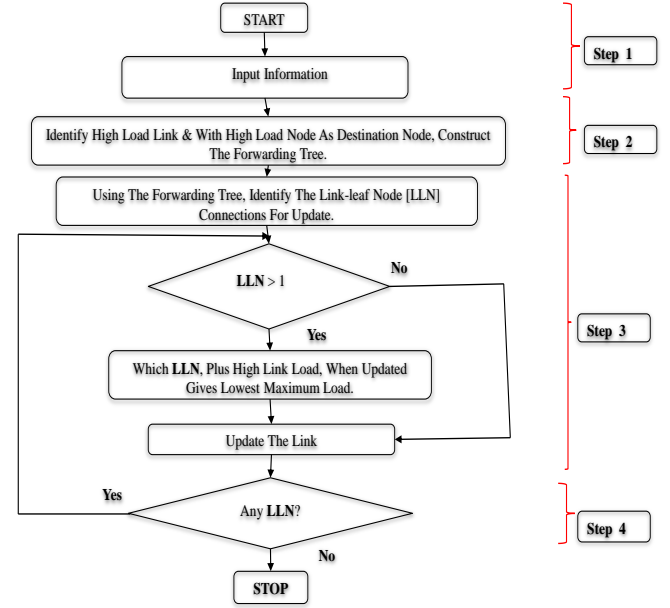


Figure 8. Flowchart of LCUSO Method

5.3.2. Simulation Results and Performance Evaluation of the Proposed LCUSO Congestion Control Method

In this section, we present the simulation results without *link overflow* during the transient link cost updates as shown in figures 9. The simulation results affirm the successful application of our proposed LCUSO method, as demonstrated in Figures 9, where no link overflow occurs during transient link cost updates. By employing the LCUSO method, we effectively identify the links requiring updates and, subsequently, utilize the Brute Force approach to determine the optimal update sequence from all possible permutations. This methodology underscores the robustness and precision of our approach in managing transient link updates efficiently. Figure 9 presents the simulation results obtained using the proposed LCUSO method for a network under the congestion conditions depicted in Figure 4. The results indicate that no link overflow occurs throughout the network, highlighting the method's effectiveness.

The evaluation considers all 24 possible permutation patterns of link cost updates, with each permutation achieving optimal routing, resulting in an optimal routing performance of 16.7%. These results validate the flawless operation of the LCUSO method, consistently delivering perfect routing performance.

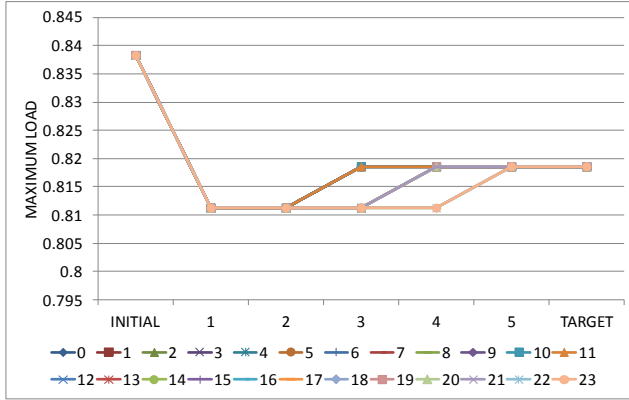


Figure 9. Results for LCUSO Method of Network Condition in Figure 4 with No Link Overflow and achieving 16.7% Optimum Routing

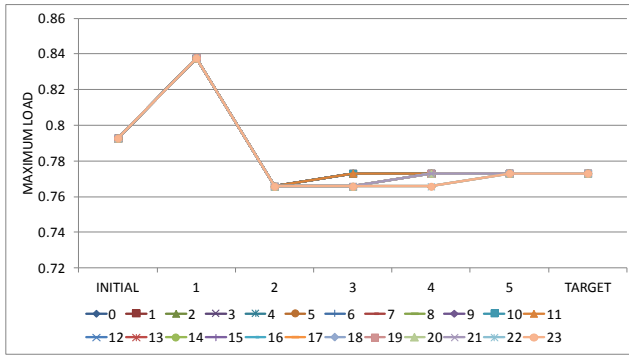


Figure 10. Results for LCUSO Method of Network Condition in Figure 8 with No Link Overflow after the second link cost update

Figure 10 presents the simulation results obtained using the proposed LCUSO method for a network under the congestion condition depicted in Figure 5. While the method does not achieve complete congestion-free outcomes, it provides the closest solution for the given congestion level, showing congestion in the initial link cost update. The results indicate that congestion occurs only during the first link cost update, after which the network remains congestion-free, demonstrating the method's effectiveness. The evaluation considers all 24 permutation patterns of potential link cost updates, with each pattern achieving optimal routing immediately after the first update, ultimately leading to optimal routing performance. These results confirm that the proposed LCUSO method functions effectively, delivering flawless routing performance.

6. Conclusions

We have introduced and validated a **Link Cost Update method** (LCUSO) designed to prevent and manage transient link overflow during link cost updates. This method is particularly useful when multiple links require updates for traffic engineering purposes. To address these challenges efficiently, we developed an algorithm that eliminates the need for exhaustive search by optimizing computation time.

The LCUSO algorithm demonstrated exceptional performance, achieving **16.7% optimal routing** under normal traffic loads

in a typical COST 239 network. It strategically identifies high-load links and prioritizes updates to leaf nodes, effectively controlling congestion and enhancing optimal routing performance. Our results, illustrated in Figures 9 and 10, highlight how network performance improves when updates prioritize leaf nodes and high-load links.

Key findings from our study validate the efficacy of the LCUSO method:

- Without prioritizing high-load links and leaf nodes** during link cost updates, transient link overflow occurs, even when using an optimal (brute force) approach.
- With prioritization of high-load links and leaf nodes**, transient link overflow is effectively mitigated, ensuring stable network performance even under the same conditions, ensuring smooth traffic flow and optimal routing.

Our evaluation conclusively demonstrates the robustness and practicality of the LCUSO method in preventing transient link overflow in the COST 239 network model. This highlights the algorithm's capability to address real-world network challenges.

7. Future Directions

Building on this success, future research will focus on extending the LCUSO method to accommodate a variety of network architectures and adapt to high traffic conditions, ensuring its scalability and applicability across a broader range of traffic engineering scenarios. In this study, we employed the COST 239 model, limiting link cost updates to a maximum of 6 links (as detailed in Table 1) to achieve 100% optimal routing. Attempts to exceed this number—such as updating 7, 8, or 9 links—proved impractical. For example, some simulations suggested that reaching 100% optimal routing would require updating 9 links. However, due to I/O constraints imposed by the operating system, our Brute Force algorithm could handle only 6 factorial (6!) combinations. As a result, our evaluation was limited to scenarios involving a maximum of 6 link updates.

REFERENCES

- [1] P. Francois, M. Shand and O. Bonaventure "Disruption free topology reconfiguration in OSPF networks," in Proceedings of IEEE INFOCOM 2007.
- [2] B. Fortz, M. Thorup, "Internet Traffic Engineering by Optimizing OSPF Weights," in proceedings of INFOCOM 2000.
- [3] Takahashi, R., Tembo, S., Yukimatsu, K. I., Kamamura, S., Miyamura, T., & Shiimoto, K. (2011, June). "Dispersing hotspot traffic in backup topology for IP fast reroute". In *2011 IEEE International Conference on Communications (ICC)* (pp. 1-5). IEEE.

- [4] Tembo, S., Yukimatsu, K. I., Takahashi, R., Kamamura, S., Miyamura, T., & i Shiimoto, K. (2012). "A new backup topology design method for congestion avoidance in IP fast reroute". *Int J Netw Commun*, 2(5), 123-131.
- [5] Nzobokela, Kulonga, Simon Tembo, and Brilliant Habeezu. "Enhancing Network Performance and Quality of Service (QoS) in a Wired Local Area Network (LAN)."
- [6] Nzobokela, Kulonga. *Comparative analysis of different network designs, connections, and performance measurement using optimized network engineering tools (OPNET) modeler*. Diss. The University of Zambia., 2024.
- [7] Ngoma, Lugodo. *Congestion avoidance in internet protocol (IP) networks*. Diss. University of Zambia, 2016.
- [8] Tembo, S., Yukimatsu, K. I., Kamamura, S., Miyamura, T., Shiimoto, K., & Hiramatsu, A. (2012). Usage of Pythagorean Triple Sequence in OSPF. *Communications and Network*, 4(01), 73-82.
- [9] P. Francois and O. Bonaventure, "Avoiding transient loops during IGP Convergence in IP Networks," in *Proc. of IEEE INFOCOM*, March 2005.
- [10] J. T. Moy, "OSPF Version 2," IETF RFC 1247. July 1991.
- [11] L. Shi, J. Fu and X. Fu "Loop-Free Forwarding Table Updates with Minimal Link Overflow," Proceeding of IEEE ICC 2009.
- [12] M. J. O'Mahony, "Results from the COST 239 Project Ultra-high-Capacity Optical Transmission Network," in Proceedings of 22nd European Conference on Optical Communication (EOC'96), pp. 11-14, Sept. 1996.
- [13] V. Jacobson, "Congestion Avoidance and Control." *Computer Communication Review* 18, no. 4 August 1988.
- [14] P. Gross, "Choosing a Common IGP for the Internet (The IESG's Recommendation to the IAB). RFC 1371, October 1992.
- [15] M. J. O'Mahony, "Results from the COST 239 Project. Ultra-high Capacity Optical Transmission Network," in Proceedings of 22nd European Conference on Optical Communication (ECOC'96), pp. 11-14, Sept. 1996.
- [16] D. Thaler et al., "Multipath Issues," IETF, RFC2991, Nov. 2000.
- [17] L. Shi, J. Fu and X. Fu "Loop-Free Forwarding Table Updates with Minimal Link Overflow," Proceeding of IEEE ICC 2009.
- [18] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, and C.Doit, "Characterization of failures in an IP backbone," in IEEE INFOCOM, 2004.
- [19] R. Teixeira and J. Rexford "Managing routing disruptions in internet service provider networks," IEEE Communications Magazine, 2006.
- [20] J. Fu, P. Sjödin and G. Karlsson "Loop-Free Updates of Forwarding Tables", IEEE Transactions on Network and Service Management, Vol. 5, No. 1, March 2008.
- [21] S.S.W. Lee, P.K. Tseng and A. Chen "Link Weight Assignment and Loop-Free Routing Table Update for Link State Routing Protocols in Energy-Aware Internet", Future Generation Computer Systems (2011), doi:10.1016/j.future.2011.05.003.
- [22] M. S. Gharajeh and M. Alizadeh, "OPCA: Optimized Prioritized Congestion Avoidance and Control for Wireless Body Sensor Networks," *International Journal of Sensors, Wireless Communications and Control*, vol. 6, no. 2, pp. 118-128, Aug. 2016.