

Method for Congestion Avoidance 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 In IP networks, changes to forwarding paths—whether due to link failures or link cost updates—necessitate orderly updates to routing tables across all routers. Failure to do so can lead to transient routing loops and link congestion. When forwarding paths change due to link failures or cost adjustments, timely and coordinated routing table updates are essential to prevent both transient loops and link overloads. While existing approaches focus on ensuring loop-free transitions, they often overlook the risk of transient link overflow during cost updates. This issue becomes particularly pronounced when Internet Service Providers (ISPs) perform traffic engineering by adjusting the costs of multiple links simultaneously, which constitutes a *complex permutation problem*. Determining an *optimal*, overflow-free update sequence within practical computation time is challenging. This paper introduces a novel *Link Cost Update Method* designed to minimize maximum link load, thereby avoiding transient congestion during cost updates. The proposed approach identifies high load links and strategically sequences the cost updates to redistribute traffic more effectively across the network. Our objective is to alleviate congestion on overloaded links by dynamically shifting traffic loads. The method's effectiveness is demonstrated through quantitative evaluations, focusing on load reduction and stability metrics during sequential updates. We model traffic demands based on service-level agreements (SLAs) to ensure throughput for virtual leased lines. Implementation and validation of the algorithm were conducted using the OPNET simulator, showcasing its practical applicability and performance benefits in real-world network scenarios.

Keywords Link Costs, High Load Link, Maximum Load, Transient State, Congestion Avoidance

1. Introduction

In this paper we present the overview of our method to Avoid Link Overflow during Link cost update. We apply the Brute Force approach to perform the permutations of the link cost update and determine the patterns of the sequence orders. To ensure we observe the condition of loop avoidance [1,9], we use the rules of addition ($\mathbf{a}_{ij} + \mathbf{b}_{ij} = \mathbf{z}_{ij}$) of the matrices operation to perform link cost update. Whereas \mathbf{z}_i represents the **Target Topology** which is in this case is the Target Topology (Figure 1) to achieve complete optimum routing i.e. 100% Optimum Routing. Whereas \mathbf{a}_{ij} is the **Initial Topology** (during Congestion Avoidance Method) and \mathbf{b}_{ij} is the link cost update needed to achieve the optimum routing i.e. 100% Optimum Routing. It has been proved that the IGP link cost of a link can always be increased to a larger link cost by progressively increasing the cost of the link by unit (equation (1)) [1,3,4,8,9] until the target cost is reached without

causing transient forwarding loops, as illustrated in Figure 1.

$$m_1 = m_0 + 1 \quad (1)$$

where m_0 is the initial link cost and m_1 is the new cost.

Past research [1,2,9] has proved that the link can be shut down by increasing its IGP link cost until it becomes large so that it cannot carry traffic anymore. It was shown that when the target cost has been reached, the link can then be safely shutdown [1,9].

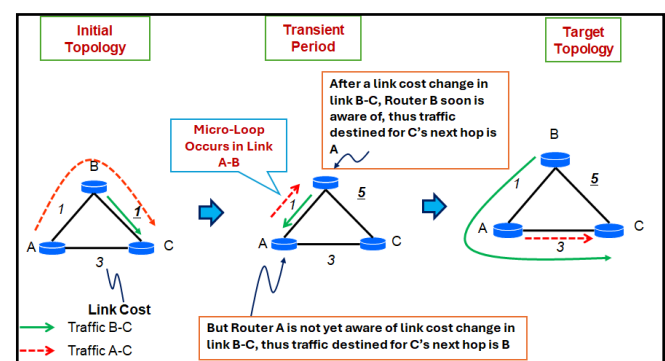


Figure 1. Micro-Loop Problem [1,20,21]

* Corresponding author:

simon.tembo@unza.zm (Simon Tembo)

Received: Dec. 4, 2024; Accepted: Dec. 19, 2024; Published: Dec. 21, 2024

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

The growing reliance on internet backbones has led many Internet Service Providers (ISPs) to establish Service Level Agreements (SLAs) with their customers, offering basic assurances regarding delay, connectivity, and throughput [5,6,7,10]. To meet these commitments and efficiently manage network resources, ISPs increasingly implement traffic engineering policies. One common strategy to achieve traffic engineering objectives—such as balancing traffic loads across a network—is adjusting link costs within Open Shortest Path First (OSPF) based IP networks [10,16].

OSPF, a link-state-based Interior Gateway Protocol (IGP) for IP networks, ensures that all packets are transmitted over the shortest paths, determined by the link costs associated with each network link [10,14,19]. However, dynamically updating link costs from their initial values to optimal target values in response to changing network conditions presents a significant challenge for ISPs [17,19]. Although network operators often configure these optimal link costs in anticipation of specific traffic conditions, fluctuations in network traffic require periodic re-optimization [18]. This process involves updating link costs for multiple networks links and constitutes a complex permutation problem (Figure 2).

This paper aims to analyze the impact of network conditions on link utilization, specifically investigating whether certain conditions can lead to link overflow.

2. Problem Statement

In this section, we use an example to illustrate how possible it is for a link overflow to occur during the transient link cost update process. For example, in figure 2 using route optimization method [2], and depending on the order of link cost updates we can avoid the problem of link overflow in link 3-7 during the transient state. 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.

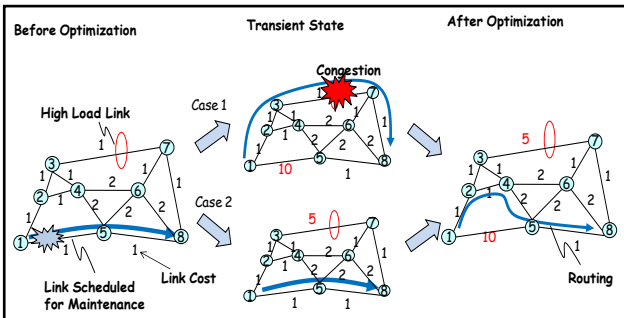


Figure 2. Simple Model –Link Cost Updates to avoid Link Overflow

Table 1. Examples of Permutation Results for given Updatable Links

	TOPOLOGY	NODES	LINKS	UPDATABLE LINKS	PERMUTATIONS
1	SIMPLE MODEL	8	22	2	2
2	COST239	11	50	6	720

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 [12,15] with 6 different updatable links is 720.

In this paper, we present a *Link Cost Update Sequence Order* method to update link costs to avoid link overflows 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.

3. Proposed Priority Link Cost Update Sequence Order Method

In this section, we present an overview of the application of our algorithm to avoid link overflow in IP Networks. The application of our proposed *Link-Leaf Node (LLN)* method effectively avoids link overflow in IP Networks. By correctly identifying links for link cost updates, our algorithm minimizes the maximum load across the network. Central to this approach is the precise identification of High Load Links and High Load Nodes, followed by the construction of a forwarding tree to strategically select links connected to the leaf nodes for cost updates. This ensures the efficient redistribution of traffic from overloaded links to other available links in the network, demonstrating the reliability and practicality of the LLN method in optimizing network performance.

3.1. Overview of the LLN Method

We present an overview of our method. The input data involves the network topology with known initial link costs, known target link costs and the traffic matrix. We apply the Brute Force approach to perform the permutations of the link cost update and determine the patterns of the sequence orders. To ensure we observe the condition of loop avoidance [1,3,4,8], we use the rules of addition ($z_{ij}=a_{ij}+b_{ij}$) of the matrices operation to perform link cost update of the matrices operation to perform link cost update.

4. Performance Evaluation

4.1. Simulations Conditions

In our evaluation, we use the COST 239 topology model [12,15] (the actual European network) illustrated in Figure 3 with 6 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.

In Figure 4, we present Initial and Target topologies of the network configuration of each simulation study. The network models are different in terms of link costs in the Initial topology compared to that of the Target topology.

The link costs for links without labels are configured with link cost of 1. *The link costs in red ink are the identified links*

that need update in the initial topology to that in the target topology. We apply the Brute Force approach to determine the update sequence order for all permutations results.

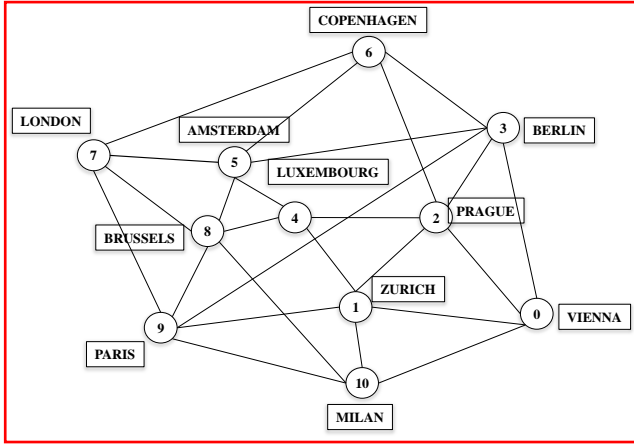


Figure 3. European Network Model - COST 239 [12, 15]

4.2. Use of Random Selection Method to select Links for Link Cost Updates

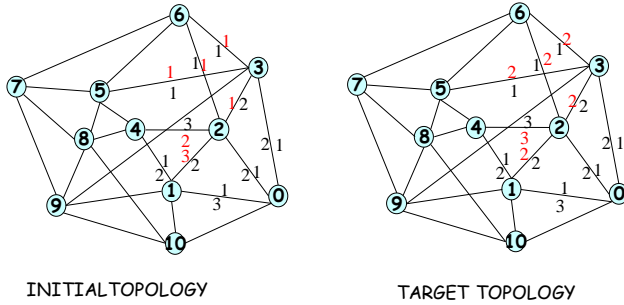


Figure 4. Random Selection of Links for Update in the Initial and Target Topologies

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 (see Figure 4) 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 LLN method.

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.

5.1. Link Overflow due to Random Selection of Links for Update During Transient Link Cost Updates

In this section, we demonstrate that when we use a random

to select links for update, link overflow occurs during the transient link cost updates as shown in figure 5. Figure 5 represent the simulation results for the network conditions in Figure 4. Figure 5 shows the 24 patterns of the permutations results of all the possible link cost updates. In Figure 6, we illustrate the optimum routing achieved in network condition of Figure 4. In this setup, the evaluation is that we are only able to achieve 3 out of 24 patterns as optimum routing. The patterns in Figure 5 that shows the optimum routing feature as illustrated in Figure 6 are 13, 16, and 22. Therefore the optimum routing performance for this network condition is 12.5% (i.e. 3/24).

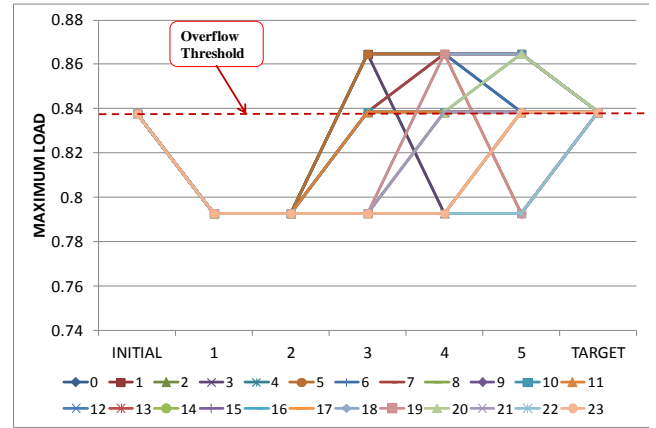


Figure 5. Use of Random Selection method leads to Link Overflow with 12.5% Optimum Routing

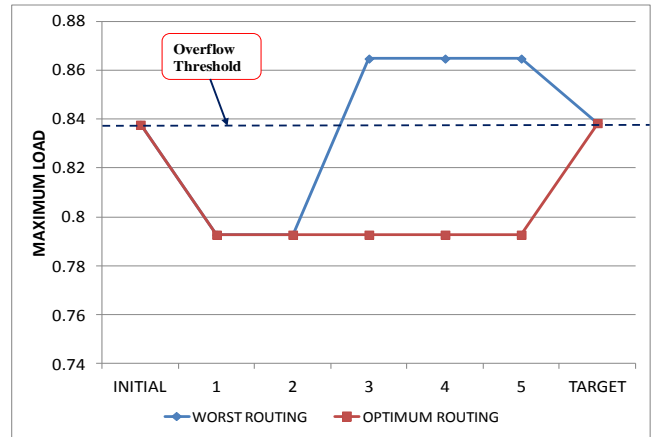


Figure 6. Worst Routing versus Optimum Routing

5.2. Proposed LLN Method to Select Links for Update During Transient - Algorithm for Avoidance of Link Overflow During the Transient Link Cost Updates

5.2.1. Algorithm Overview

In this section, we present the algorithm designed to effectively prevent link overflow during transient link cost updates, affirming the adoption of our proposed Link-Leaf Node (LLN) method. This method reliably identifies links for cost updates, successfully minimizing the maximum load in the network. The approach is centered on pinpointing the High Load Link and High Load Node, followed by constructing a forwarding tree to prioritize links connected to

leaf nodes for cost updates. By doing so, the algorithm ensures efficient traffic redistribution from overloaded links to alternative paths in the network. The LLN method thus provides a systematic and precise mechanism for selecting links for cost updates. The flowchart of the algorithm, depicted in Figure 7, demonstrates the step-by-step process and highlights the robustness of our approach.

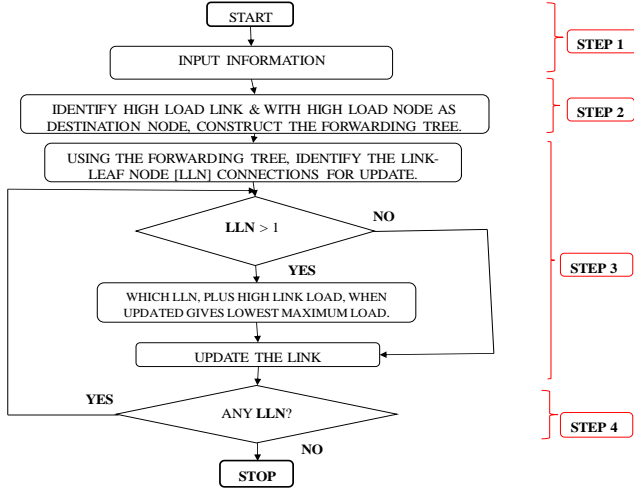


Figure 7. Flowchart of LLN Method

5.2.2. Results without Link Overflow During the Transient Link Cost Updates

In this section, we present the simulation results without *link overflow* during the transient link cost updates as shown in Figure 8. The simulation results affirm the successful application of our proposed LLN method, as demonstrated in Figures 8, where no link overflow occurs during transient link cost updates. By employing the LLN 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.

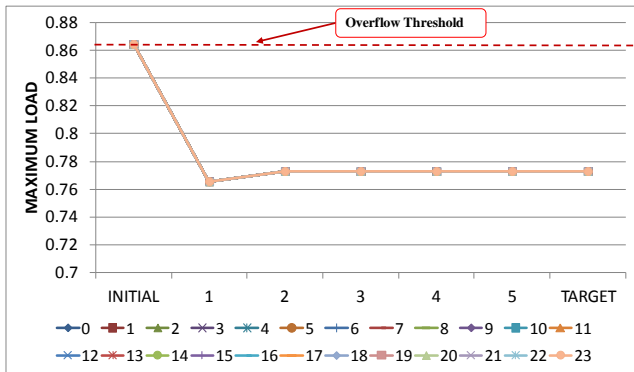


Figure 8. No Link Overflow and achieves 100% Optimum Routing

Figure 8 illustrates the simulation results achieved by applying the proposed LLN method. The results demonstrate that there is no link overflow across the network, demonstrating its effectiveness. The evaluation covers all 24 permutation patterns of possible link cost updates, with each pattern yielding

optimal routing, resulting in a 100% optimum routing performance (i.e., 24 out of 24 patterns). These findings confirm that the proposed LLN method operates flawlessly, delivering perfect routing performance.

6. Conclusions

We have successfully introduced and validated a Link Cost Update method designed to avoid transient link overflow during link cost updates. This method is particularly effective when multiple links require updates for traffic engineering purposes. Our proposed algorithm addresses such practical challenges with efficient computation, eliminating the need for exhaustive search. The LLN algorithm achieved 100% optimal routing in a typical COST 239 network under normal traffic conditions. The variations in network performance, as depicted in Figure 8, depend on the prioritization of Leaf Nodes during updates.

In summary, our findings affirm the efficacy of the LLN method:

- When the network condition for link cost updates excludes High Link Load and prioritized Leaf Nodes, transient link overflow occurs even with the optimal (brute force) approach.
- Conversely, when the network condition includes High Link Load and prioritized Leaf Nodes, transient link overflow is effectively prevented, even with the optimal (brute force) approach.

Our evaluation conclusively demonstrates that the Link Cost Update method successfully prevents transient link overflow in the COST 239 network model. This validation underscores the algorithm's practicality and robustness. As future work, we aim to extend the method's applicability to seamlessly handle diverse network models and accommodate heavy traffic loads.

REFERENCES

- P. Francois, M. Shand and O. Bonaventure "Disruption free topology reconfiguration in OSPF networks," in Proceedings of IEEE INFOCOM 2007.
- B. Fortz, M. Thorup, "Internet Traffic Engineering by Optimizing OSPF Weights," in proceedings of INFOCOM 2000.
- 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.
- Tembo, S., Yukimatsu, K. I., Takahashi, R., Kamamura, S., Miyamura, T., & Shiimoto, K. (2012). "A new backup topology design method for congestion avoidance in IP fast reroute". *Int J Netw Commun*, 2(5), 123-131.
- 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.