A Novel MPLS Traffic Load Algorithm for Efficient Traffic Management

Khadidja Sakhri,* Ahmed Korichi, Azzaoui Nadjet

Department of Computer Science and Technology of Information and Communication
University of Kasdi Merbah, UKMO
Ouargla, Algeria

*Email: sakhri.khadidja@univ-ouargla.dz

ABSTRACT

The ever-increasing network user use places enormous strain on the existing network infrastructure, causing congestion and lowering network performance. Multi-Protocol Label Switching (MPLS) is becoming the standard transport technology for core networks due to its potential to create procedures and improve network availability, reliability, and efficiency. Indeed, it was employed in traffic engineering to govern network flow to optimize resource use, network performance and reduce congestion. Load balancing is one of the approaches used to decrease congestion. As of now, the network does not offer quick traffic path adjustment based on flow priority. This study proposes a new technique to alleviate congestion in MPLS networks. The goal is to minimize the packet loss while exploiting underutilized links considering the priority of the traffic flows. Our approach's performance was tested using the Omnet++ simulator.

Keywords: *MPLS Network, MPLS Traffic Engineering, Congestion, Load Balancing; Omnet++ simulator*

1. INTRODUCTION

Nowadays, wireless technology has improved significantly, enabling consumers to access a variety of services on a variety of scales. The number of mobile phone users has risen dramatically in recent years. Mobile internet usage has expanded significantly as a result of the proliferation of applications and services [1]. Email, web surfing, audio/video streaming, internet phone, and real-time multimedia services all require varying levels of service quality. The networks must provide an adequate level of service for each application, resulting in a suitable level of service. In light of this, Mobile IP (MIP) has been established to benefit mobile users, allowing for access to the services from anywhere. Thus, allowing both macro and micro-mobility. However, it does not ensure the Quality of Service (QoS) of the services delivered to users [2]. Packets were previously routed via less-than-optimal channels in terms of time and bandwidth. As a result, it is not a viable method for efficiently providing services. To address this issue, route optimization techniques have been developed. These techniques assist in selecting the most efficient path between source and destination in terms of delay, bandwidth, and jitter. However, traditional IP tunnelling of packets is a lengthy operation that requires each intermediate router to match the destination IP address. As a result, a new technology called Multi-Protocol Label Switching (MPLS) was invented, eliminating the need for traditional IP address matching and providing a speedier routing mechanism. As a result, applications receive an acceptable level of service [3].

MPLS has established itself as the de facto standard transport technology for packet-based core networks. MPLS technology enables the development of various techniques for increasing network availability, reliability, and efficiency. Congestion reduction is a critical performance aim for traffic and resource management [4]. Indeed, MPLS is also used in traffic engineering to manage traffic flows over a network, optimizing resource use and performance while decreasing congestion. Congestion often manifests as one of two types: when the network's resources are insufficient or insufficient to handle the load being offered. Conversely, traffic streams are not adequately routed to available resources, causing some network resources to be over-utilized while others remain under-utilized [5]. Congestion concept can be

alleviated by spending money to expand the network's capacity. In this vein, load balancing solutions can help alleviate congestion issues caused by unequal network traffic distribution. As a result, the most pressing problem is balancing network traffic while maintaining optimal resource allocation and enhancing network speed and internet quality of service **[6].** While the exact path through the transit network is irrelevant to the sender or receiver of the traffic, network managers frequently wish to route traffic more efficiently between specific source and destination address combinations. MPLS shifts packets from router to router via the network by appending a brief label containing precise routing instructions to each package, rather than forwarding them based on next-hop lookups. The label-switched paths that arise are referred to as label-switched routes (LSPs). LSPs manage traffic flow throughout the network and facilitate traffic forwarding. LSPs can be generated manually or via signaling protocols. Within an MPLS context, signaling protocols are utilized to establish LSPs for transit traffic **[7].**

MPLS traffic engineering incorporates the following components:
- ✓ MPLS LSPs for packet forwarding IGP extensions for disseminating network topology and connection attributes
- ✓ Constrained Shortest Path First (CSPF) for path computation and selection • RSVP extensions for establishing forwarding states and reserving resources along the path
- ✓ Traffic engineering's primary goals are to minimize high-loss situations: A negligible number of packets were lost.
- ✓ Maintain a balance between network performance and Quality of Service (QoS) and network operation and maintenance costs.

Numerous traffic engineering solutions are available, including IP-over-ATM, constraint-based routing, and others. MPLS-Traffic Engineering solves the constraints of these systems by precisely combining the flexibility of layer 3 with the capabilities of layer 2 for traffic management and is widely recognized as the fundamental methodology for future generation IP-based networks. Its emergence demonstrates a strong technological commitment to traffic engineering **[8].**
The strategies outlined above have made significant strides toward optimizing flow. However, more outstanding adaptive network adjustment capabilities are critical because the network is increasingly susceptible to congestion. Many works address this subject, we highlight the main ones:

The Topology-Based Static Load Balancing Algorithm TSLB presented in **[9]** improves the classic shortest path routing algorithm in that the path is picked based on its bandwidth demand. Otherwise, it will be removed from the collection of possible values. The authors in **[10]** proposed the Resource-Based Static Load Balancing Algorithm RSLB. It precomputes the collection of paths the same way as TSLB does. Still, it evaluates the approaches to select the one with the least bandwidth required to accommodate the incoming arrival traffic. However, if only a few low-rate flows are sent over an extended period, the Internet's high-capacity links will remain inactive for an extended period. These static methods have drawbacks: they are both static, and in some cases, they can result in inefficient resource consumption.

The Dynamic Load Balancing Algorithm (DLB) developed in **[9]** can consider both the network's topology and the bandwidth required by new traffic. According to the simulation results reported in **[9]**, DLB is ineffective in the presence of a significant number of parallel routes connecting the Ingress and Egress nodes and frequent traffic rerouting. Indeed, the techniques described above are only applicable for unicast traffic with no priority.

In **[10]**, the authors developed a Load Balancing Technique with Deviation Path (LBDP) that changes a selected flow to a deviation path when congestion is imminent. The deviation path is determined using the shortest path algorithm. Another issue with the LBDP is that it may place a disproportionate amount of traffic on some nodes compared to others. Additionally, it selects the qualified deviation path from a limited set of possible ways. The likelihood of choosing an unqualified path is relatively high. For example, suppose two deviation pathways exist. In that case, the shortest of which has a used bandwidth that is exceptionally close to the threshold specified, the other of which has a used bandwidth that is less than the threshold. In this vein, LBDP will choose the shortest path regardless of whether the used bandwidth is greater than the threshold. Another congested situation will occur in this circumstance, despite our best efforts to avert it. In the worst-case scenario, when there is a high traffic volume, the network will be subjected to a rising frequency of congestion, resulting in increased packet loss **[11].**

To address this issue more precisely, we propose specifying a constraint that considers the consumed bandwidth and the requested bandwidth if these two factors fall below the defined threshold. Indeed, our work aims to present a new scheme for reducing congestion in an MPLS network using a load balancing mechanism. The key concept is to efficiently reroute LSPs from the network's most congested links to

balance the overall link load and better use network resources by exploiting the underutilized ones.

Traffic engineering (TE) enables efficient and dependable network operations while maximizing resource utilization and traffic throughput. Traffic engineering allows traffic flow to be redirected away from the shortest path chosen by the internal gateway protocol (IGP) and onto a possibly less congested physical path across a network. Apart from source routing, the network must perform the following functions to assist with traffic engineering:

- Calculate a path at the source, taking all constraints into accounts, such as bandwidth and administrative requirements.
- Once the path is computed, distribute network topology and link characteristics information throughout the network;
- Reserve network resources and adjust link attributes.

The remaining sections of the paper are organized as follows. **Section 1** highlights a high-level overview of Traffic Engineering in MPLS networks. **Section 2** then highlights proposed algorithm. **Section 3** discusses the findings, followed by **Section 4's** conclusions.

2. PROPOSED ALGORITHM

This section introduces a novel Load Balancing Algorithm for MPLS NETworks, dubbed LBMNET, which prioritizes traffic flows.
K is an MPLS network with a source node (host1) and a destination node (host2). When host1 sends a packet to host2, the package is routed across the network via the LSRs routers (LSR1, LSR2,.... LSR7).
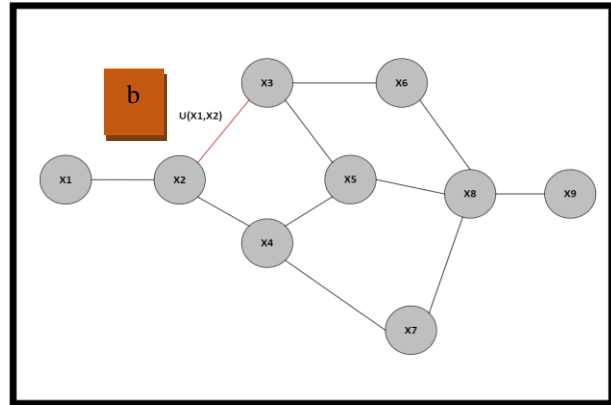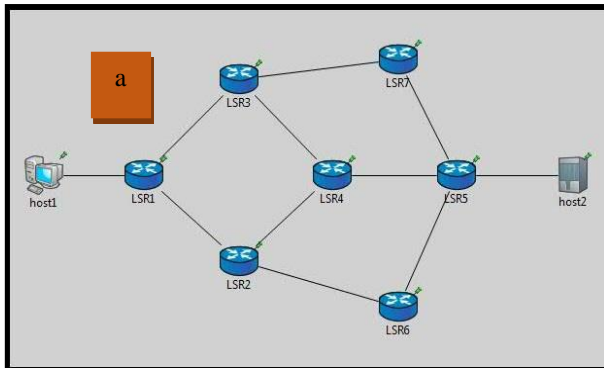

Figure 1: (a) K: MPLS-TE Network, **(b)** Network Topology.

Assume that G(X, U) is a multigraph representing a segment of the network K, with X=x_1, x_2, x_3,...x_n representing the set of network nodes (LSR routers) and U=u_1,u_2,u_3...u_n meaning the set of arcs representing network branches represented by network pathways (**Eq.1**). The term u(i,j) refers to the connection between nodes i and j. We define Succ[x, i] as the set of I successors of node x and an integer i. We utilize an additional symbol ω which means "indefinite" and does not belong to X. This enables better administration of the list's end. Succ [x, i] equals y, y$\in$X. Succ[x,i]=ω, $\omega\notin$X, x has only i-1 successors. The following is a list of the successors' representations:

$$
\begin{aligned}
&X_1: X_2\ \omega \\
&X_2: X_3\ X_4\ \omega \\
&X_3: X_6\ X_5\ \omega \\
&X_4: X_5\ X_7\ \omega \\
&X_5:\ X_8\ \omega \\
&X_6: X_8\ \omega \\
&X_7: X_8\ \omega \\
&X_8: X_9\ \omega \\
&X_9:\ \omega
\end{aligned}
\tag{1}
$$

To define the deviation node of a link u, we shall refer to the preceding section's list of successors. Assume that u_1 is a path connecting x_1 to xi, with u_1=x_1,x_2,....x_k,....xi, and that u_2 is another path connecting x_1 to xi, with u_2=x_1,x_2,.....x_k,....xi. If x_1,x_2,....x_k = x_1,x_2,....x_k and $x_{k+1}\in$Succ[x_k,i], and $x_{k+1}\neq x_{k+1}$, then the node x_k is the deviation node, and u_2 is a deviation path of u_1 against the node x_k

Following that, we create a data structure called ER-LSP to represent a resource ER-LSP's reservation. ER-LSP = {<Traffic-ID>, <LSPID>, <UB>}

The term "Traffic-ID" refers to the traffic that the ER-LSP carries. LSPID is the identifier for this ER-LSP path, and UB is the ER-LSP's used bandwidth rate. Assuming ɣ is a threshold, we create a data structure called CL for Congested LSPs. CL is composed of all LSP paths whose UB is greater than or equal ɣ (UB>= ɣ). CL= {ER-LSP1, ER-LSP2, ER-LSP3 …}

Selected-LSP is the ER-LSP that has the highest UB in CL, making it the most congested path.

We create a new data structure called ST (Selected Traffic) that contains all Traffic flows associated with the Selected-LSP. ST= {Traffic-ID1, Traffic-ID2 …}

Assume that Selected-LSP is a path connecting nodes ni and nj; Selected-LSP=ni, n_1, n_2,... n_k... nj. Additionally, there is another path from node ni to node nj known as P, which is defined as ni, n_1, n_2,...nk...nj. If ni, n_1, n_2,...nk equals ni, n_1, n_2,...nk. And nk+1=nk+1, the deviation node is nk+1, and P is the deviation path of Selected-Path against nk+1. Following that, we create a Negotiated Deviation path collection NDC, which contains all of Selected-LSP's deviation paths. the adopted notations are presented in **Table- 1**.

Table- 1. Adopted notations.

Notation	Description
NDC	P1, P2, P3,
RB	A traffic flow's Requested Bandwidth.
FC	Capacity Available for Use (Free Capacity)
UB	Used Bandwidth.
Y	Determined Threshold.
CL	set of paths congested.
Selected-LSP	MOST! Congested path from CL
ST	Set of Traffic flows associated with the Selected-LSP. ST= {Traffic-ID1, Traffic-ID2 …}
Pending Traffic	Traffic that will be rerouted on the deviation path
NDC	Set of deviation paths

The specific steps of our Algorithm are as follows: When an ingress node receives a new traffic flow,

- Check the UB rates of the adjacent links periodically: if exist an ER-LSP with (FC ≤ ɣ), then establish CL containing ER-LSP, else no congestion is about to happen.

- Select the most congested path P where FC of P is the smallest in CL, and assign P to Selected-LSP.

- Establish ST, and select traffic flow from ST whose priority is lower with minimum RB, and assign traffic flow's Traffic-ID to Pending-Traffic.

- Find the deviation node then the deviation paths existed (Establish NDC)

- Find successors then the deviation node, after that select deviation paths starting from this node and going to the same destination.

- Select P whose FC is the largest in NDC, if (RB<FC> ɣ), then assign Traffic-ID of P to Pending-Traffic, and switch Pending-Traffic, else Delete P from NDC, Capacity is insufficient to satisfy the bandwidth request of the selected traffic flow, it fails to transport the traffic, and algorithm ends.

- If FC of Selected-LSP ≤ ɣ, then select traffic flow from ST whose priority is lower with minimum RB, and assign traffic flow's Traffic-ID to Pending-Traffic, else delete Pending-Traffic from ST and selected-LSP from CL, Algorithm finishes successfully.

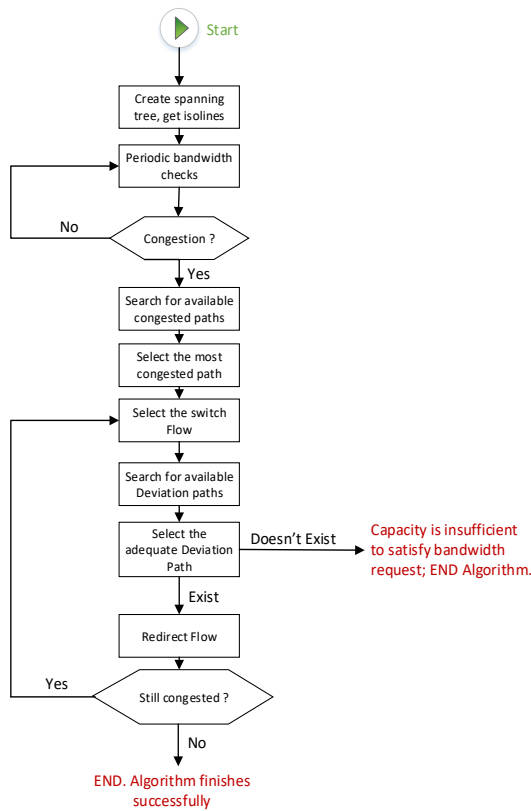Following is the explained LBMNET algorithm flow chart:

Figure 2. LBMNET Model flow chart.

Figure. 3. Data transmission paths after congestion in LSR2.

3. LBMNET APPROACH

LBMNET is capable of considering both the network topology and the traffic bandwidth requirement concurrently and the traffic flow priority. When the load is light, traffic flows can be mapped on the short and high-capacity routes. However, when the load is heavy, traffic flows with lower capacity and minimum bandwidth are selected and rerouted to another suitable route (a deviated route). Thereby limiting congestion on the high-capacity route and saving it for critical, high-priority traffic flows. The LBMNET algorithm can significantly improve resource efficiency, network throughput, and performance by delivering a lower packet loss rate (Figure 3), as a result, ensuring the quality of the critical services.

4. FINDING RESULTS

Figure 3 illustrates the congestion that occurred in LSR2 after our approach was applied. A new path was generated as a result of our algorithm's implementation. There are now two (2) options:
The initial LSP is as follows: (LSR1 – LSR2 – LSR4 – LSR3 – LSR7 – LSR5), and the new deviation path is as follows: (LSR1 – LSR2 – LSR6 – LSR5).
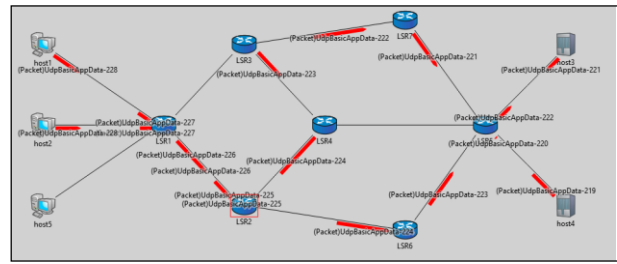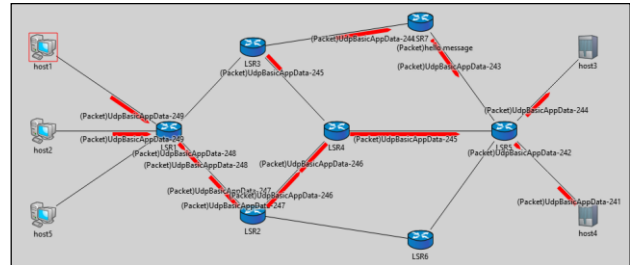

Figure. 4. Data transmission paths after congestion in LSR4.

While Figure 4 revealed that LSR4 was congested, our approach generated a new path. There are now two (2) options:
The original path was as follows: (LSR1 – LSR2 – LSR4 – LSR3 – LSR7 – LSR5), while the new deviation path was as follows: (LSR1 – LSR2 – LSR4 – LSR5).
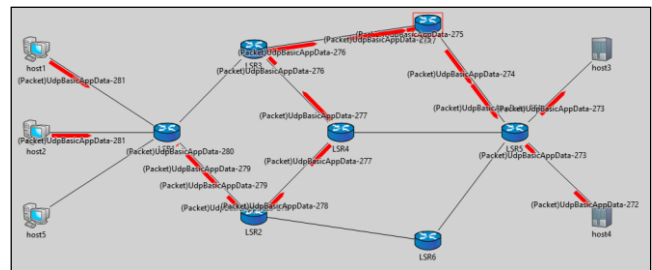

Figure. 5. Data transmission paths after the end of congestion.

Once the transient congestion subsides (Figure 5), all LSPs resume their usual paths.

Figure 6 shows a decrease in the bandwidth value at the level of the ppp1 interface connected to LSR4 (which is the congested path) beginning at 2.2s (congestion). In contrast, the bandwidth value increases at the ppp2 interface connected to LSR6 (a deviating path). The same is true for the Bandwidth Graph at LSR4 (Figure. 7).
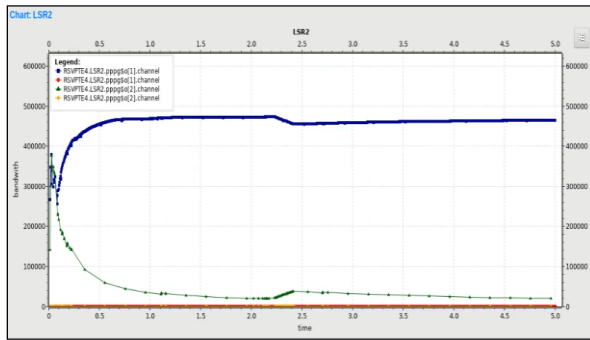
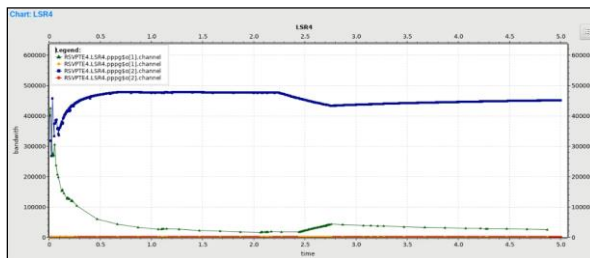**Figure. 6.**   Graph of the bandwidth change at LSR2.

Figure. 7.   Graph of the bandwidth at LSR4.

Figures 8 and 9 illustrate a decrease in the end-to-end delay value at the level of hosts 3 and 4, beginning at the moment 2.2s (congestion), and continuing until the congestion subsides at the time 2.8s.
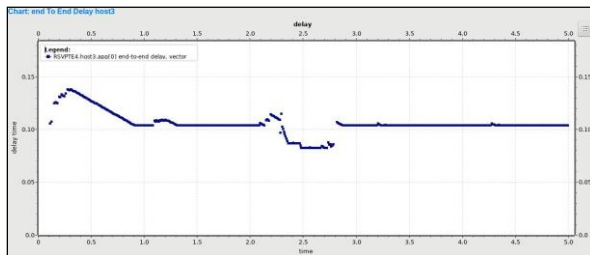

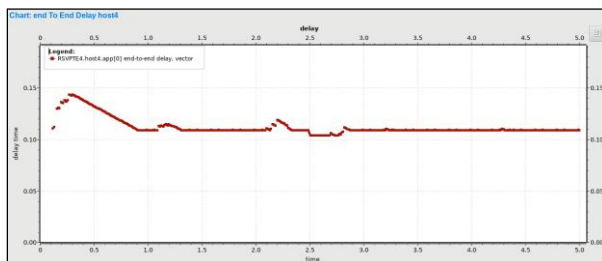**Figure. 8.**   End to end Delay at Host3.

Figure. 9.   End to end Delay at Host4.

5.   CONCLUSION

MPLS networks face a significant challenge in meeting functional requirements as traffic, subscribers, and services grow. With the growth of latency- and packet-loss-sensitive traffic, the need for network performance control and traffic classification has emerged. A discussion of routing algorithms in general was presented and discussed. Indeed, MPLS Traffic Engineering and the four primary load balancing algorithms used in MPLS TE was presented. Following that, our newly proposed load-balancing algorithm, LBMNET was introduced. Finally, the performance of LBMNET and discovered that it could improve both network throughput and resource utilization was examined.   thereby alleviating congestion. The obtained results validated the suggested algorithm's reliability and efficiency.

REFERENCES

[1] J. Jiang, ***"Intelligent City Traffic Scheduling Optimization Based on Internet of Things Communication***", Wireless Communications and Mobile Computing, vol. 2021, pp. 1-10, 2021. Available: 10.1155/2021/7823982.

[2] P. Appandairaj and K. Kannan, ***"Software-Defined Multilayered Admission Control for Quality-of-Service Assurance in Mobile Ad-hoc Networks"***, Wireless Communications and Mobile Computing, vol. 2020, pp. 1-23, 2020. Available: 10.1155/2020/2989751.

[3] I. Aijaz and S. Idrees, "***Performance Evaluation of Multi-protocol Label Switching-Traffic Engineering Schemes"***, ICST Transactions on Mobile Communications and Applications, vol. 6, no. 19, p. 166550, 2021. Available: 10.4108/eai.8-10-2020.166550.

[4] B. Chaurasia, W. Manjoro and M. Dhakar, ***"Traffic Congestion Identification and Reduction"***, Wireless Personal Communications, vol. 114, no. 2, pp. 1267-1286, 2020. Available: 10.1007/s11277-020-07420-0.

[5] I. Nurhaida and I. Ichsan*, **"CONGESTION CONTROL PADA JARINGAN KOMPUTER BERBASIS MULTI PROTOCOL LABEL SWITCHING (MPLS)"***, Simetris: Jurnal Teknik Mesin, Elektro dan Ilmu Komputer, vol. 11, no. 1, pp. 77-88, 2020. Available: 10.24176/simet. v11i1.3671.

[6] N. Rismawati and M. Mulya, ***"Analisis Pemilihan Metode Quality Of Service Dengan Traffic Policing Dan Traffic Shaping Sebagai Pembanding Bandwidth Pada Cisco Router Internet Service Provider"***, Jurnal ULTIMA InfoSys, vol. 9, no. 1, pp. 37-44, 2018. Available: 10.31937/si.v9i1.848.

[7] A. Poulhès, C. Pivano and F. Leurent, "***Hybrid Modeling of Passenger and Vehicle Traffic***

along a Transit Line: a sub-model ready for inclusion in a model of traffic assignment to a capacitated transit network", Transportation Research Procedia, vol. 27, pp. 164-171, 2017. Available: 10.1016/j.trpro.2017.12.079.

[8] M. Shyni, *"Detection Monitoring Of Secure Packet Transfer Over Network Traffic",* International Journal Of Engineering And Computer Science, 2016. Available: 10.18535/ijecs/v4i10.25.

[9] Keping Long, Zhongshan Zhang and Shiduan Cheng, "*Load balancing algorithms in MPLS traffic engineering,*" In 2001 IEEE Workshop on High Performance Switching and Routing, 2001, pp.175-179.

[10] F.Li and J.Chen , *"MPLS Traffic Engineering Load Balance Algorithm Using Deviation Path,".* 2012. pp. 601– 604.

[11] K. Supriya, M. Chand and K. Sekhar, "*Congestion Aware Multi-Path Routing Protocol with Load Balancing",* International Journal of Advances in Applied Sciences, vol. 6, no. 1, p. 6, 2017. Available: 10.11591/ijaas.v6.i1.pp6-11.

[12] A. S. Jauhari and A. I. Kistijantoro, *"INET Framework modifications in OMNeT++ simulator for MPLS traffic engineering,"* 2014 International Conference of Advanced Informatics: Concept, Theory and Application (ICAICTA), 2014, pp. 87-92, Available: 10.1109/ICAICTA.2014.7005920.

[13] Oleksandr Lemeshko, Oleksandra Yeremenko, *"Linear Optimization Model of MPLE Traffic Engineering Fast Reroute for Link, Node, and Bandwidth Protection"* In IEEE TCSET, pp.1009-1013, 2018.

[14] R. White and E. Banks. Computer Networking Problems and Solutions: *An innovative approach to building resilient, modern networks* 1st Edition. 1 edition. Addison-Wesley Professional, 2018.

[15] A.S. Monge and K. G. Szarkowicz. *MPLS in the SDN Era: Interoperable Scenarios to Make Networks Scale to New Services*. O'Reilly Media, 2016.

[16] L. Davoli, L. Veltri, P.L. Ventre, G. Siracusano, S. Salsano, *Traffic engineering with segment routing: SDN-based architectural design and open source implementation,* in : Proceedings of the IEEE Fourth European Workshop on Software Defined Networks, 2015, pp. 111-112.

[17] M.C. Lee, J.P. Sheu, *An Efficient Routing Algorithm Based on Segment Routing in Software-Defined Networking*, Computer Networks. 106 (2016)

[18] Arafah, M., Alhindi, H., Mathkour, H., Faisal, M., & Lytras, M. D. (2018). *Smart systems approach for development of explicit congestion marking and traffic engineering model for Diffserv/MPLS networks*. Journal of Ambient Intelligence and Humanized Computing.

[19] Xu Yingzhuo and Geng Qing Yang 2019, *Research on network load balancing method based on simulated annealing algorithm and genetic algorithm*, J. Phys.: Conf. Ser. 1237 022137