Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 25/11/2026 04:33:22

Annales UMCS Informatica AI IX, 1 (2009)
165–177; DOI: 10.2478/v10065-009-0013-0

Annales UMCS
Informatica
Lublin-Polonia
Sectio AI
http://www.annales.umcs.lublin.pl/

Quality of Experience (QoE) Assurance by a Multi–path Balanced Traffic–Splitting Algorithm in MPLS Networks

CS Krishnadas*, Rajarshi Roy †

Dept. of E & ECE, Indian Institute of Technology, Kharagpur, India

Abstract

Multi-Protocol Label Switching (MPLS) technology has proven its worth for delivering new services while at the same time allowing migration from old to new generation networks. Avoidance of congestion is one of the major performance objectives of traffic engineering in MPLS networks. **Load balancing** can prevent the congestion caused due to inefficient allocation of network resources. Another important aspect in network performance is the end user perception of the quality delivered by the network called the **Quality of Experience (QoE)**. The final arbiter of service performance is the end user whose opinion about quality is based on his or her perception. This end user perception of audiovisual quality is quantified by **Mean opinion score (MOS)**. The network parameters that affect the MOS are delay, Jitter and loss.

Though a number of multipath load balancing algorithms have been proposed in [**1**] and [**2**], none have proportioned traffic keeping the QoE constraint in mind. Here, a multipath load balancing algorithm is used to optimally split the incoming traffic based on the effect of average delay and jitter offered by the network so that the QoE measure of MOS is maximized. These initial results indicate that desirable QoE can be achieved by finite and small number of executions of an appropriate iterative load balancing algorithm once the step-size and the

E-mail address: kaydee$_0$07@yahoo.com
†*E-mail address:* royr@ece.iitkgp.ernet.in

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 25/11/2026 04:33:22

weights of the composite cost function representing combined effect of average delay and jitter are judiciously chosen.

1. Introduction

A. Overview

Traffic engineering is concerned with the performance optimization of operational networks. Its main objective is to reduce the congestion and improve resource utilization across the network by managing the traffic distribution inside a network. Multi–path routing is one of the mechanisms for load balancing in MPLS networks [**3**] where the total load from a source to a destination is spatially distributed over several paths. Multi Protocol Label Switching (MPLS) [**4**] can provide the connections with Label Switched Paths (LSPs) that are explicit routes connecting pairs of ingress-egress edge routers in an MPLS network.

The quality of the network as perceived by the end user is an important indicator of how well the system meets its target of minimum quality expected by the users. Quality of Experience (QoE) is basically a subjective measure of end to end performance at the services level, from the point of view of the users. It has been shown in [**5**] and [**6**] that QoE is affected by network parameters like delay, jitter and loss.

The objective of this study is to split the incoming traffic over pre established LSPs so that the QoE metric of MOS delivered by the MPLS network is maximized. In this work, the objective function is characterized based on the average delay and jitter experienced by the network and returns a value that is a measure of the QoE. By optimally splitting the incoming traffic, our algorithm maximizes the MOS for a given network topology.

The algorithm used in this paper exploits the **E-Model** as recommended in **ITU-T G.107** [**7**] which returns a value for a "Rating factor R" which offers an estimate of the user opinion called the Quality of Experience (QoE). The load balancing algorithm attempts to push the value of the MOS to a satisfactory region (Recommended MOS is above 4) by optimally splitting the input traffic at the ingress router along predefined LSPs. The E-model is a well established computational model that uses the transmission parameters to predict the subjective quality. It uses a psycho-acoustic R-scale whose values range from 0 to 100 and can be mapped to MOS rankings and User Satisfaction as shown in Table 1 [**7**]. The E-model fundamentally addresses objective quality assessment for voice traffic. However, [**5**] and [**6**] show that reasonable correlation exists between the subjective quality assessment scores for audiovisual quality provided by the users and the objective quality assessment scores provided by

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 25/11/2026 04:33:22

video quality monitoring softwares like Telchemy VQMon tool that uses the E-model and traces obtained from various videoconferencing tasks as an input for its analysis. Therefore, it is reasonable to use E-model as a tool for estimating QoE for audiovisual signal as well.

Table 1. Relation between R value, MOS and user satisfaction

This work first presents the algorithm as a solution of a single criteria optimization problem, first based on delay and then on jitter. This helps us to study the effect of delay and jitter on the MOS. Then the two parameters are combined by weighted summation to realize a cost function for a multi objective optimization problem that proposes maximization of the MOS for a given network topology. It is shown that by optimally splitting the traffic, the function can be maximized to provide a desirable QoE.

B. Paper organization.

The paper is organized a follows. Section 2 presents the analytical problem formulation and main features and assumptions of the system model and algorithm. The same is followed by the description of the execution procedure of the algorithm. Section 3 presents the results and behavior of the algorithm for different network topologies and establishes the objective of this work. We conclude in Section 4.

2. The load balancing algorithm

In this section we present the basis of the work along with the formulation of the problem, lay down the features and describe the assumptions made and the load balancing algorithm.

A. Problem formulation

The problem is one of optimization where the objective function is based on the **E-Model** as recommended in **ITU-T G.107** [**7**]. Let us describe the details of this E-model as presented there using equations (1)–(X). The objective function returns a value for a "Rating factor R" which gives an estimate of the user opinion called the Quality of Experience (QoE) delivered by the network. The 'R factor' can also be expressed as a Mean Opinion Score (MOS). The output of the E-model in a first step is a transmission rating factor R, which combines all transmission parameters relevant for the considered connection. This rating factor R is composed of various components and the resultant expression is the following:

$$R = Ro - Is - Id - Ie + A, \tag{1}$$

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 25/11/2026 04:33:22

where

Ro – Basic signal–to–noise ratio

Is – Represents Combination of all impairments which occur more or less simultaneously with the voice signal

Id – Represents the impairments caused by delay

Ie – Represents impairments caused by low bit rate codec

A – Advantage factor.

The R factor can be mapped to MOS by the formula:

$$MOS = 1 + 0.035R + R\,(R-60)\,(100-R)\cdot 7\cdot 10^{-6}.$$

The problem considers the effect of network delay /jitter dependent quantity Ta, which is subsequently introduced, on the QoE and optimizes the function so as to push the MOS to a satisfactory value. As this quantity only affects Id, the objective function is characterized assuming default values for all other impairments except Id and that leads to the following expression:

$$R = 94 - Id.$$

The impairment factor due to delay (Id) is expressed as, with

$$Id = 25\left\{\left(1+X^6\right)^{1/6} - 3\left(1+\left[\frac{X}{3}\right]^6\right)^{1/6} + 2\right\} \tag{2}$$

$$X = \frac{\log\left(\frac{Ta}{100}\right)}{\log 2}. \qquad \text{Where, } Ta \text{ is the delay.}$$

Assuming an M/M/1 model, the average delay is expressed by the following formula:

$$Ta = \sum_i \left(\lambda i/\lambda s\right)/\left(\mu - \lambda i\right), \tag{3}$$

where, i is the index number of the i–th LSP operating between the ingress and Egress routers.

By substituting (3) and (2) in (1), the rating factor 'R' can be characterized as a function of the traffic rate (λ). The problem is now a maximization problem as described below.

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 25/11/2026 04:33:22

Maximize,

$$f(\lambda) = 94 - 25\left[\left(1+\left(\left(\log\left(\sum_i \left(\left(\lambda i/\lambda s\right)/\left(\mu-\lambda i\right)\right)/100\right)\right)/\log 2\right)^6\right)^{1/6} - 3\left(1+\left(\left(\left(\log\left(\sum_i \left(\left(\lambda i/\lambda s\right)/\left(\mu-\lambda i\right)\right)/100\right)\right)/\log 2\right)/3\right)^6\right)^{1/6} + 2\right]. \tag{4}$$

Subject to

$$\lambda s = \sum_i \lambda i \qquad \forall i \in \text{the } IE \text{ pair and } \lambda i \geq 0. \tag{5}$$

B. Features and Assumptions

Some of the features and assumptions of the algorithm are stated below:

• The load balancing algorithm is executed at the ingress router, and the feedback of QoE is obtained from the Egress router which runs a module whose output is the 'R' factor or MOS value.

• A fixed topology is assumed. That is, several explicit LSPs have been set up between an ingress and egress node using standard protocols such as CR-LDP or RSVP-TE [**4**] or manually.

• An M/M/1 model is assumed.

• It is assumed that the all the updates are immediately reflected in all the link flows i.e., total feedback delay is zero.

• Optimization decision is based on the average delay and jitter offered by the network.

The functional model is shown in Fig. 1. Please note that a similar architecture is also considered in [**2**]. At the ingress node, the load balancing algorithm optimally splits the incoming traffic. In the monitoring phase, if the QoE measure (MOS or 'R' factor) goes below an acceptable limit, transition is made to the load balancing phase, a new flow vector is obtained and the traffic splitting is performed accordingly.

C. The load balancing Algorithm

The gradient projection algorithm [**8**] is used to solve the constrained optimization problem defined by expressions (4) and (5). The algorithm recursively adjusts the traffic rates towards a direction of the gradient that reduces the flows on the non shortest paths with the corresponding increment of flow being

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 25/11/2026 04:33:22

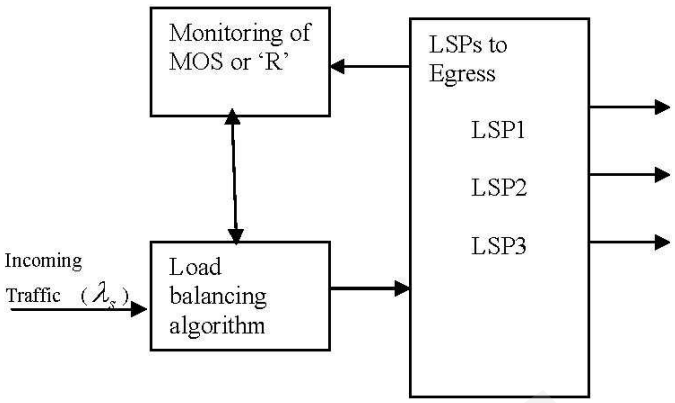

Fig. 1. Functional model

shifted onto the Minimum First Derivative Length (MFDL) path. This optimal routing characterization is based on the Kuhn Tucker theorem [**8**]. Here the first derivative of the function in (4) with respect to a given path is interpreted to be the length of the corresponding path.

$$dp = df(\lambda)/d(\lambda i).$$

Here, dp is the so–called "path length" of path with index i.

The algorithm starts with some initial guess for the flow vector (λ_s) and then proceeds according to an iteration process. The iteration of the gradient projection algorithm attempts to obtain an optimal flow vector (λ_s^*) on every iteration. The iteration takes the form:

$$\lambda_i^{k+1} = \lambda_i^k - \min\left[\lambda_i^k, \alpha^k \nabla f\left(\lambda_i^k\right)\right]$$

where

$$\nabla f\left(\lambda_i^k\right) = dp - d\overline{p}$$

dp =First derivative length of path p which is a non MFDL path
$d\overline{p}$ = First derivative length of path $\overline{p}$ which is an MFDL path
k = No of iterations
α^k = step size

Thus $(dp - d\overline{p})$ determines the amount of flow to be shifted from the non MFDL path to the shortest path. The step size in the algorithm is chosen to be a constant. However, a method of line search can also be adopted to calculate the step size. The algorithm terminates when there is no appreciable change i.e., $\lambda_p^{k+1} - \lambda_p \leq \varepsilon$, for some predefined ε.

The projection algorithm is simple to implement. However, the convergence is slow as the solution approaches the neighborhood of the optimal solution. In our case, the objective is to push the operating point close to the optimal or

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 25/11/2026 04:33:22

into the 'good' region and not necessarily to reach a maximum value. Hence the algorithm is well suited for this application.

3. Numerical results

In this section, we first describe the method adopted to study the implementation of the algorithm. The results of the algorithm for different topologies are then presented followed by the inferences on the behavior of the algorithm.

A. Methodology

For the numerical computation, two network topologies have been used as shown in Fig. 2 and Fig. 3. Similar benchmark topologies were used by other researchers before [**2**]. The first topology consists of a single pair of ingress egress node, with two pre established LSPs. The second topology has two ingress nodes and two egress nodes thus providing a multi-flow scenario. For each of the cases, the algorithm was implemented first with average delay as the decision criteria for splitting the incoming traffic. After that, jitter is used as the decision criteria and finally a weighted summation of delay and jitter is formulated to be used as a cost function for a single objective optimization problem.

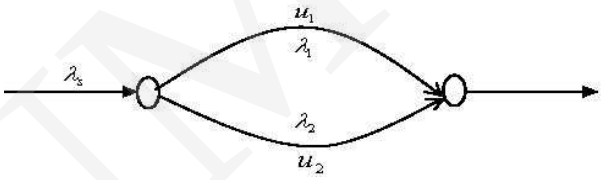

Fig. 2. Network Topology 1

In Topology 1, the average rate of incoming traffic λ_s is set at 100 units. This is split over the two paths and the initial split is assumed to be the following: $\lambda_1 = 59$ units and $\lambda_2 = 41$ units. The capacities of both the links are assumed to be same: $u_1 = u_2 = 60$ units.

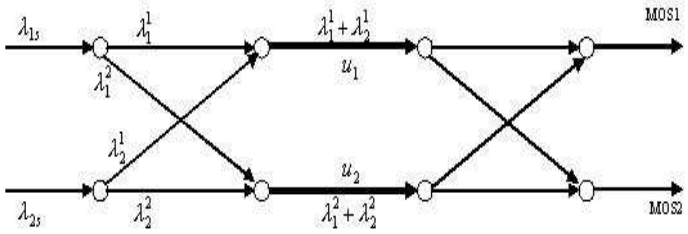

Fig. 3. Network Topology 2

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 25/11/2026 04:33:22

In topology 2, the average rate of incoming traffic from source 1 is assumed to be $\lambda_{1s} = 100$ units. This is split over the two paths where the initial split is assumed to be the following:$\lambda_1^1 = 59$ units and $\lambda_1^2 = 41$ units. For source 2, the average rate of incoming traffic is assumed to be $\lambda_{2s} = 70$ units and this is split over the two paths with the initial split assumed as $\lambda_2^1 = 40$ units and $\lambda_2^2 = 30$ units. The capacities of the bottleneck links are assumed to be $u_1 = 100$ units and $u_2 = 80$ units.

B. Effect of Delay

Here we show the performance of the algorithm when the decision criteria for the traffic split is the average delay offered by the network.

Fig. 4 shows the convergence results for topology 1. The initial value of MOS is 3.0120. The algorithm is executed at the ingress router and by iteratively splitting the traffic the MOS is finally pushed up to 4.2544.

Fig. 5 shows the convergence results for topology 2. The initial value of MOS1=2.9673 and MOS2=2.9977 for sources 1 and 2 respectively. The algorithm is executed at the ingress routers simultaneously for both the sources and the MOS is finally pushed up to MOS1=4.1983 and MOS2=4.2147 for the two sources.

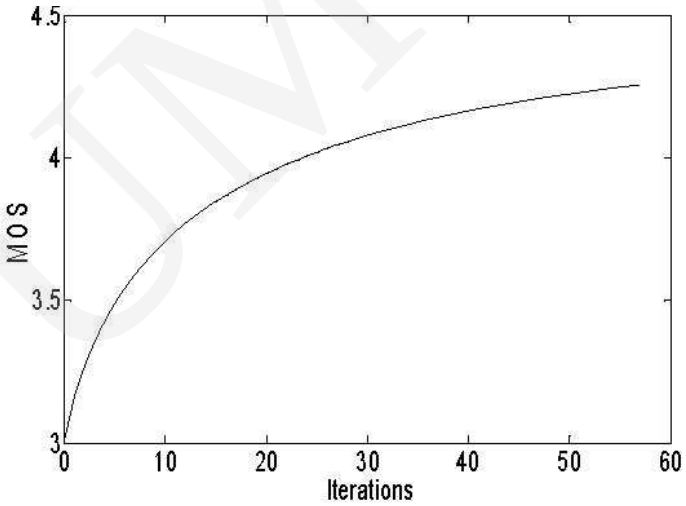

Fig. 4. Convergence of MOS for topology 1

The number of iterations required to reach the final value is large in this case. However, the recommended MOS for most applications is above 4 which can be achieved in a relatively small number of iterations as seen from the graph. The step size is assumed to be constant and is equal to 0.5 in both cases.

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 25/11/2026 04:33:22

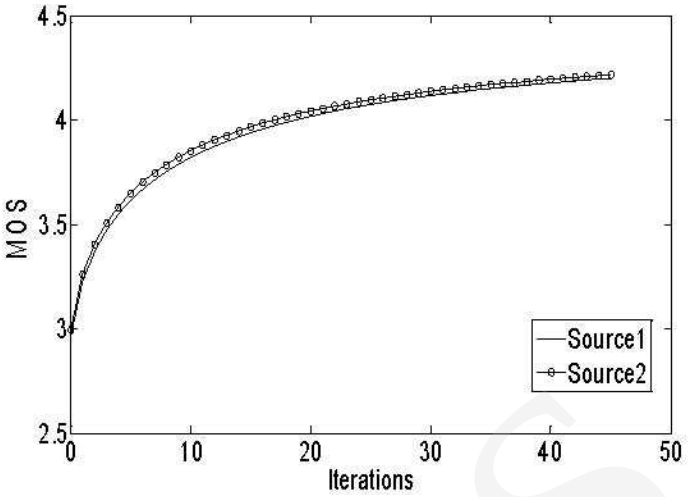

Fig. 5. Convergence of MOS for topology 2

C. Effect of jitter

Here we show the performance of the algorithm when the decision criteria for the traffic split is the jitter offered by the network. Jitter is related to variation of delay. Therefore, in equation (2), the delay is substituted by an expression which represents a quantitative measure of jitter and is computed as,

$$T_j = \sum_i (\lambda i / \lambda s) \,/\, (\mu - \lambda i)^2$$

Fig. 6 shows the convergence results for topology 1. The initial value of MOS for the above configuration is 2.8768. The algorithm is executed at the ingress router and by iteratively splitting the traffic the MOS is finally pushed up to 4.4142.

Fig. 7 shows the convergence results for topology 2. The initial value of MOS1=3.0449 and MOS2=3.0837. The algorithm is executed at the ingress routers simultaneously for both the sources and the MOS is finally pushed up to MOS1=4.4221 and MOS2=4.4236 for the two sources.

The effect of jitter on QoE is much more pronounced than delay as evident from the gradient of the convergence plot. This trend was reported by other researchers before [**5, 6**]. However, the methodology used by them to establish this claim was mostly experimental in nature. Further it was observed that by setting the step size to 0.05, the results of jitter can be made to approach closely the same of delay with a step size of 0.5. This observation indicates that the effect of jitter on QoE is almost 10 times the same of delay. This empirical observation leads us to construct the composite cost function used in subsection

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 25/11/2026 04:33:22

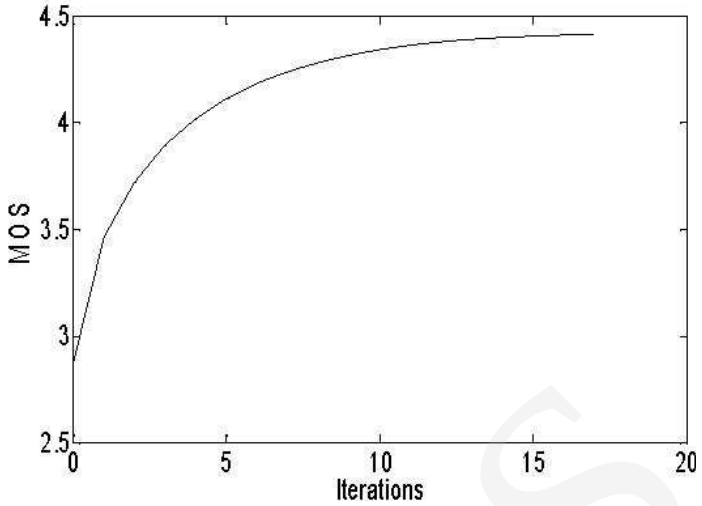

Fig. 6. Convergence of MOS for topology 1

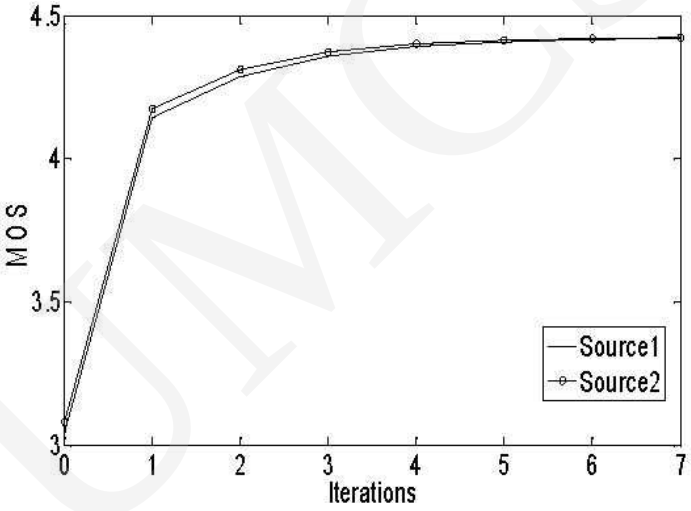

Fig. 7. Convergence of MOS for topology 2

III-D as a weighted summation of delay and jitter by using a weight of 1 for the jitter and 1 to 10 for the delay.

D. Combined effect of delay and jitter

Here we show the performance of the algorithm when the decision criteria for the traffic split is the cost function which represents a combination of delay and jitter. In equation (3), the network parameter is taken as the weighted summation of delay and jitter.

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 25/11/2026 04:33:22

$$T = W_1 (T_a) + W_2 (T_j) .$$

Fig. 8 shows the convergence results for topology 1 with weights for delay chosen as 1, 3, 5, 7 and for jitter fixed at 1. The results when the algorithm is executed are tabulated in Table 2.

Fig. 9 shows the convergence results for topology 2 with weights for delay chosen as 1, 3, 5, 7 and the same for jitter is fixed at 1. The algorithm is executed at the ingress routers simultaneously for both the sources and by iteratively splitting the traffic; the MOS is maximized for the two sources.

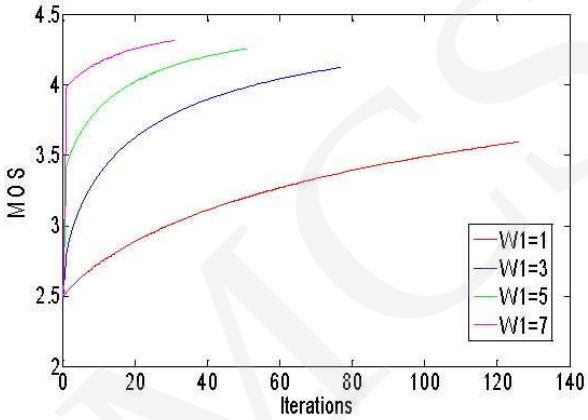

Fig. 8. Convergence of MOS for topology 1

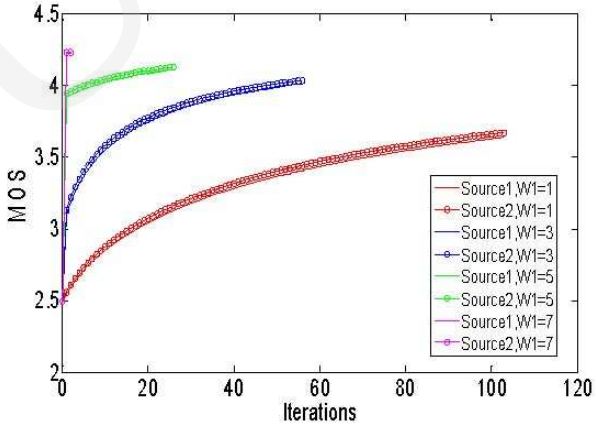

Fig. 9. Convergence of MOS for topology 2

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 25/11/2026 04:33:22

Table 2.

Delay weight (W1)	Initial MOS	Final MOS	Final Split	No. of iterations
1	2.4933	3.5928	57.7765, 42.2235	127
3	2.4933	4.1212	56.9186, 43.0814	78
5	2.4933	4.2548	56.5772, 43.4728	52
7	2.4933	4.3132	56.2768, 43.7232	32

Table 3.

Delay weight (W1)		Initial MOS	Final MOS	Final Split	No. of Iterations
1	Source 1	2.4841	3.6360	58.2527, 41.7473	104
	Source 2	2.4957	3.6698	39.2527, 30.7473	
3	Source 1	2.4841	4.0111	57.8305, 42.1695	57
	Source 2	2.4957	4.0324	38.8305, 31.1695	
5	Source 1	2.4841	4.1097	57.6361, 42.3639	27
	Source 2	2.4957	4.1251	38.6361, 31.3639	
7	Source 1	2.4841	4.2215	57.2068, 42.7932	3
	Source 2	2.4957	4.2260	38.2068, 31.7932	

The above results have been computed with different weights for delay and the weight for jitter set as 1. The step size is set as 0.05. It is observed that as the weights for delay are increased, the convergence rate increases drastically and the required number of iterations reduces significantly. Further, it is also observed that the MOS value can be pushed much higher by increasing the weight for delay. It is seen that the rate of convergence in all cases becomes slower as the optimum value is approached. This is a typical feature of the gradient projection algorithm.

4. Conclusions

In this paper we have focused on optimizing a function that returns a value of the MOS based on the network parameters of delay and jitter. In our numerical results we have shown that our algorithm optimizes the objective function by

Pobrane z czasopisma Annales AI- Informatica **http://ai.annales.umcs.pl**
Data: 25/11/2026 04:33:22

splitting the traffic so as to push the MOS to a desirable value thus providing the end user of the network with a good QoE. For future work the performance of the algorithm will have to be studied by implementing it in a networking environment with audiovisual traces feeding a software emulation of MPLS ingress router and QoE measuring software at the Egress router. We also plan to explore the possible incorporation of propagation and resequencing component of delay and use of decentralized fuzzy goal programming based algorithms with delayed feedback.

References

[1] Lee K., Toguyeni A., Rahmani A., *Comparison of multipath algorithms for load balancing in a MPLS Network*, LNCS 3391, Springer-Verlag Berlin Heidelberg (2005) 463.

[2] Elwalid A., Jin C., Low S. and Widjaja I., *MATE: MPLS Adaptive Traffic Engineering*, INFOCOM'2001 Alaska (2001) 1300.

[3] Allan D., *Guidelines for MPLS Load Balancing*, RFC2026, <draft-allan-mpls-loadbal-01.txt> (2003).

[4] Rosen E. C., Viswanathan A., and Callon R., *Multiprotocol label switching architecture*, RFC 3031, Internet draft <draft-ietf-mpls-arch-05.txt> (1999).

[5] Calyam P., Sridharan M., Mandrawa W., Schopis P., *Performance measurement and analysis of H.323 traffic*, Proceedings of the Passive and Active Measurement Workshop (PAM), Antibes Juan-les-Pins, France (2004) 137.

[6] Calyam P., Lee Chang-Gun, *Characterizing voice and video traffic behavior over the Internet*, In Advances in Computer Science and Engineering: Reports, Imperial College Press, Proceedings of ISCIS 05, Turkey 1 (2005).

[7] ITU-T Recommendation G.107. *The E-model, a computational model for use in transmission planning* (1998).

[8] Bertsekas D. and Gallager R., *Data networks*, Prentice-Hall Inc., 2nd ed.