A New Routing Algorithm in MPLS Networks using Fuzzy AHP method with guaranteed bandwidth and Delay end-to-end

Meysam Liravi¹, Alireza Qasemi²

¹Department of Computer Engineering, Boushehr Branch, Islamic Azad University, Boushehr, Iran, mysam.liravi@gmail.com
²Department of Statistics and Information Technology, Dadgostari, Boushehr, Iran, alireza@ymail.com

Abstract

Multi-protocol Label Switching (MPLS) is presented as a new approach that combines the advantages of using routing in the third layer of the network and the second layer of the data link. But the most important part of this technology is the implementation of traffic engineering. The most important requirement for traffic engineering is to available network resources (such as bandwidth) across the network to efficiently select the LSP paths. Our main goal in this research is to load balance in the infrastructure of MPLS networks using the Fuzzy AHP method and the Dijkstra algorithm while guaranteeing bandwidth and end-to-end delay. In the proposed algorithm for weighing links, we have used fuzzy AHP techniques, which makes it impossible to use links that are in critical condition when routing. And to improve performance, we have implemented a new approach using high priority bits in network packets to transfer them to destinations. Finally, we simulate the proposed algorithm in MATLAB software and compared it with the MIRA and MDMF and BGLC and GA&PSO algorithms. The simulation results show that the proposed MPLS_FAHP algorithm is better than other algorithms in terms of the number of requests accepted (call blocking ratio) and Mean Length and Maximum Flow.

Keywords: MPLS, Traffic Engineering, Fuzzy-AHP, Quality of service, Routing
1- Introduction

New real-time Internet applications like Voice over IP (VoIP) require setting up constrained paths through the network. IP routing takes into account only the destination of packets and the only way to change routing is to modify metrics used by routing protocols. Hence the existing IP routing protocols cannot provide QoS routing [1][2]. The Internet Engineering Task Force (IETF) has proposed three different service models and mechanisms to support the requested QoS. These models are Integrated Services (Intserv) [3], Resource Reservation Protocol (RSVP)[4], Differentiated Services (Diffserv) and Multi-Protocol Label Switching (MPLS)[5].

MPLS (Multi-Protocol Label Switching), as standardized by the Internet Engineering Task Force (IETF) is a layer 3 packet-switching technology that transmits traffic effectively and supports QoS on the Internet. It is expected that MPLS improves the performance of routing in the network layer [6].

MPLS is used in Internet Service Provider (ISP) networks and as a backbone to Internet Protocol (IP) to provide guaranteed efficient bandwidth and Quality of Service (QoS) provisioning in the network [7]. MPLS supports multiple Layer 2 protocols such as ATM, Frame Relay and Ethernet. Because of the variety of the underlay network structures, MPLS can establish end-to-end IP connections with different QoS characteristics associated with the multiple transport media, its objective to give the router a big power of communication. In MPLS networks, parameters such as send and receive rates, Time to Live (TTL), and packet loss rates can often be specific to network traffic. In other words, a high traffic path for each of the above parameters has inappropriate values. In a high-traffic path, the rate of receiving is lower than the rate of sending data, and this is due to high requests that go to the route. However, if the path has a packet loss, these parameters will not have the desired values, which means that the route has high traffic.

In this paper, we present an optimal new algorithm for traffic engineering in MPLS networks using parameters bandwidth, end-to-end delay, and availability to guarantee QoS. And also to improve performance, we have implemented a new approach using high-priority bits in network packets to transfer them to destinations. We used the AHP method to calculate the weight of the links.

We have paper is organized as follows. In Section 2 the MPLS network infrastructure is studied. In Section 3 the related works are studied. In Section 4, we develop foundations of the proposed algorithm. Our proposed routing algorithm is described in Section 5. In Section 6, we present simulation results and comparison algorithms new with the other algorithms and finally in Section 7, we conclude the study.

2- Multi Protocol Label Switching (MPLS)

MPLS (Multi-Protocol Label Switching), as standardized by the Internet Engineering Task Force (IETF) is a layer 3 packet switching technology that transmits traffic effectively and supports QoS on the Internet. It is expected that MPLS improves the performance of routing in the network layer [8][9]. MPLS is used in Internet Service Provider (ISP) networks and as a backbone to Internet Protocol (IP) to give guaranteed efficient bandwidth and Quality of Service (QoS) provisioning in the network [10]. MPLS supports multiple Layer 2 protocols such as ATM, Frame Relay and Ethernet.

Traditional IP Forwarding routing protocols are used to forward packets in the network in Layer 3 of OSI Model where routing lookups are performed on every hop. Each router in the topology makes an independent decision where to send the packets based on destination IP address. MPLS is a scalable, connection-oriented, independent packet forwarding technique. The packets are forwarded by assigned labels [11]. Labels may correspond to Layer 3 destination IP address or other parameters such as Quality of Service, Source address etc. This protocols works between data link layer(Layer 2).
Fig. 2 shows the different fields of a MPLS Shim header [12]. An MPLS header is 4 bytes long, and the numbers at the top in the figure show the bit positions.

The different fields of an MPLS header are:

- **Label**: the 20-bit value of the field indicates a label based on which the multi-protocol label switching occurs. When a labeled packet arrives at a label switch router (LSR), based on the label value, the next hop of the packet is decided, and it is forwarded accordingly. A label value is relevant only for a given hop. Before forwarding a packet, the LSR will either replace the label value or pop the top-MPLS header out of the label stack.

- **EXP**: the 3-bit EXP field was originally reserved for experimental use, but it is used to indicate the QoS desired for the labeled packet.

- **S**: the 1-bit stack filed value indicates whether the header is the last entry within the label stack. A value of 1 indicates that the header is the last entry (i.e., bottom most entry) of the label stack, and a value of 0 indicates that there are more entries after this header.

- **TTL**: the value of the 8-bit field indicates the time to live the frame.

MPLS is an advanced forwarding scheme. It extends routing with respect to packet forwarding and path controlling. An MPLS capable router, termed Label Switching Router (LSR), examines the label and possibly the experimental field in forwarding the packet. At the ingress LSRs of an MPLS-capable domain IP packets are classified and routed based on a combination of the information carried in the IP header of the packets and the local routing information maintained by the LSRs. An MPLS header is then inserted for each packet. Within an MPLS-capable domain, an LSR will use the label as the index to look up the forwarding table of the LSR. The packet is processed as specified by the forwarding table entry. The incoming label is replaced by the outgoing label and the packet is switched to the next LSR. This label-switching process is very similar to ATM's VCI/VPI processing. Before a packet leaves a MPLS domain, its MPLS header is removed. This whole process is showed in Fig. 3. The paths between the ingress LSRs and the egress LSRs are called Label Switched Paths (LSPs)[13]. MPLS uses some signaling protocol like RSVP [14], RSVP-TE [15], LDP [16] or CR-LDP [17] to set up LSPs.
MPLS has two planes [18]:
1. Control Plane: Control Plane is used for the label distribution and routing information exchange among adjacent devices.
2. Data Plane: Data Plane is used for propagating packets on the basis of label or destination IP address using LFIB controlled by the control plane.

3- RELATED WORK
We divide service quality routing algorithms into MPLS networks into two broad categories of bandwidth-based algorithms and algorithms with delays and bandwidth constraints.

In [19] a multi-objective routing algorithm for Multiprotocol Label Switching networks with multiple service types and traffic splitting is presented. The routing problem is formulated as a multi-objective mixed-integer program, where the considered objectives are the minimization of the bandwidth routing cost and the minimization of the load cost in the network links with a constraint on the maximal splitting of traffic trunks. Two different exact methods are developed for solving the formulated problem, one based on the classical constraint method and another based on a modified constraint method.

In [20,21] algorithms called CSPF (Constrained Shortest Path First) are presented. In these algorithms, the shortest path is selected based on a routing criterion, and the only accessible and measurable criterion is the link bandwidth remaining. First, the algorithm eliminates links that have less residual bandwidth than the requested value and then calculates the weight of each link using the cost function, then the shortest path algorithm such as Dijkstra or Bellman-Ford, the shortest possible path between the source and the destination it finds. Finally, the status of links in the route is updated and prepared for future requests.

In [22], the Widest Shortest Path (WSP) algorithm is expressed. The WSP algorithm chooses a path that has the lowest hop and from the paths that have bandwidth constraints and the number of hop they are the same, the path with the highest bandwidth is selected. Problems WSP algorithm may select a path that could become a congestion point (no request rejection aspect is considered).

In [23,24] the Shortest Widest Path (SWP) algorithm is presented. This algorithm chooses the path that has the most available bandwidth and among the routes with the highest bandwidth, the route with the lowest number of the hop is selected. This algorithm emphasizes the load balancing in the network. This algorithm uses two stages of the Dijkstra algorithm. In the first step, it finds the widest possible path. We assume widest path bandwidth B. Then all the links that are of the less bandwidth is eliminated. Then, for the second time, he uses the Dijkstra algorithm and chooses the shortest path from among the widest paths. Problems SWP algorithm may select a path that could become a congestion point (no request rejection aspect is considered).

In [25], the Shortest Distance Path (SDP) algorithm is expressed. This algorithm chooses the path that has the shortest location distance. The distance of a link is defined as the reverse of the bandwidth of that link, and the distance of a path is the sum of the distances of all links of the path member. This algorithm is a combination of WSP and SWP algorithms. The distance is defined as when the network load is high, it goes to the shortest path, and when the network load is normal, it goes to the widest path. The problem with this algorithm is not using the metric delay.

An algorithm is presented in [26], which is called the Dynamic-alternative Path (DAP) algorithm. In this algorithm, we use the algorithm of the widest short path, but the number of hop is n+1, so that n is the lowest hop that is not included in the Pruning network. An unprintable network is meant to be a network that, before removing network links, does not have enough resources to meet the quality of the service. If you do not find the suitable path with the least hop, the widest path that is chosen is just one more hop and otherwise, the request is rejected.
In [27], a new traffic model has been proposed in the MPLS network that traffic queue balancing for different classes of service and a provider using another provider network. The model result shows that carrying another operator’s traffic may increase delays in an undesirable manner, forcing the carrier to increase the serving rate of LSRs until Utilization is below 60%. This model specifies the mechanisms that should be applied in congestion conditions based on the value of queue length and delay time.

In [28], a bandwidth-guaranteed routing algorithm was called the Minimum Interference Routing Algorithm (MIRA). The MIRA algorithm receives information about the LSP location, such as the ingress and egress routers for LSP requests. In the MIRA algorithm, requests are sent as \((s, d, b)\), where \(s\) and \(d\) are respectively the sources and destination routers, and \(b\) bandwidth is requested. The MIRA algorithm first calculates the maximum flow of the source and destination network for the weight of the network links. After that, it removes links whose bandwidth is less than the bandwidth required. It then finds the shortest path between the source and the destination using the Dijkstra algorithm, routing the sent LSP request, and finally updating the remaining bandwidth of the links. These steps are repeated for all requests. The problem with this algorithm, Cannot detect critical links in topologies with clusters of nodes and also computationally expensive.

In [29], a new routing algorithm proposed in MPLS networks with bandwidth and the end-to-end delay was called the MDMF (Minimum Delay Maximum Flow). This algorithm obtains the path between source and destination using the MIRA [28] algorithm. And then calculates the maximum delay path using the theory of Latency Rate servers [30]. Now, if the delay is calculated from the delay request more, the bandwidth of one of the path links that have the most remaining bandwidth will be incremented, and then the delay will be calculated. This process is repeated until the desired amount of delay to achieve. Otherwise, remove the link on the path with the least bandwidth and re-find the shortest path next and calculate the delay of this path. This process continues until the desired path is found or the request is rejected. The problem with the MDMF algorithm has a high computational time, due to its high complexity to calculate the maximum end-to-end delay in addition to bandwidth.

In [31], proposed an algorithm called bandwidth to constrain the routing algorithm (BCRA). BCRA compromises between network load balancing, reducing path length and minimizing path cost. In this algorithm, a critical link is defined as a link whose load is running above some threshold. In BCRA the threshold is defined as the mean link load throughout the network. A critical path is one that has critical links. Hence a path is more critical if it contains more critical component links. The problem with this algorithm is that it has not taken into account the information of ingress-egress pairs and network topology.

In [32], a hybrid algorithm for dynamic multi-path routing has been proposed using the advantages of two basic PBR [33] and WDP algorithms using the MPLS technique called (Hybrid Multi-Commodity Based Widest Disjoint Path Algorithm) HMBWDP. The algorithm works in two phases: Preprocessing phase and the Path selection phase. In this preprocessing phase traffic profiles \((CID, Pi, Si, di, Bi)\) are generated, corresponding to a real-time network \(G=(V, E)\), where \(CID\) is traffic class identity, \(Si\) the finite set of sources, \(di\) is the finite set of destinations and \(Bi\) is minimum aggregate bandwidth requirement for this traffic profile class \(CID\) between set of sources \(Si\) and set of destinations \(di\). \(Pi\) is the priority assigned to this profile class based on the previous knowledge about the network. Each traffic class is treated as a separate commodity. In Path Selection Phase, Once the Profile Classes are generated in the preprocessing phase. The traffic demand flows are now handled by grouping them as per Profile Class, between source-destination pairs through LSPs. In the path selection phase, Widest Disjoint Paths w.r.t. bottleneck links are generated for the corresponding Profile class flow \(s\) generated and categorized in the preprocessing phase.

In [34], proposed an algorithm called the Load Balancing algorithm using a pheromone deposit for deviation path selection (LBPDP). The general idea of the LBPDP algorithm is that we start by creating the pheromone table in each node using the ant agents who will put a pheromone in each link between two nodes, they will scan large number of network nodes, and collect information about the network while moving and delivers it to the network nodes. Each node will have its table that contains pheromone values for the adjacent links. Secondly, when congestion is about to happen, we create a spanning tree that will be quite useful in the deviation node selection. Thirdly, we choose and select the flow to redirect. Next phase, we select the optimal path by using the pheromone table. And finally, we switch the selected flow on the selected deviation path. The problem with the LBPDP algorithm has a high computational time.
15th Iran Media Technology Exhibition and Conference

In [35], the BGLC algorithm is presented to whose purpose guarantee bandwidth using a load balancing at the network level with the least complexity of time and computation with the minimum possible status information. In this algorithm, after obtaining the total paths between ingress-egress pairs, the level of criticality of links is obtained through the ratio possible demand per link to the total possible demand path through the entire links. The main disadvantages of this algorithm are that it considers a delay of links to be fixed and does not consider maximum flow.

In [36] a hybrid of the fuzzy-CAC algorithm to manage and reduce traffic in MPLS networks is proposed. The CAC algorithm is a common method for generating multicast routing. After the routing is done, the channel allocation is done. In this algorithm, node-level information is first obtained. To scan the entire network, the first Breadth-First Search (BFS) is used. After creating a multicast tree, channel assignment information to link with the node level using the CAC algorithm is performed. The disadvantages of this algorithm are that there is a possibility of traffic between nodes at a level. Because all nodes of the same channel level are assigned the same.

In [37], the genetic and PSO algorithm is used to improve the quality of routing in MPLS networks for traffic management. The PSO algorithm is used for an optimal search in data space to reach the spanning tree. In this method, the genetic algorithm is used to obtain the best position of the particle. The genetic algorithm uses the process of mutation and composition in the process of updating the particle velocity and obtains the optimum value of the particle position. The disadvantages of this algorithm are that if the number of channels is greater than the number of tree levels, some channels are not used at all. And the channel allocation is done in a level-to-level which may be two nodes that are neighbors can choose a common channel and send them together with traffic.

4- Methodology

In this paper, we intend to reduce congestion while meeting quality of service criteria (bandwidth, end-to-end delay and availability) and more capacity to make for future requests. Our goal is to maximize requests while satisfying the quality of service (bandwidth, delay end-to-end and availability).

In this paper, we present a QoS-based routing algorithm for traffic management in MPLS networks using bandwidth and end-to-end delay as constraints. The network is modeled as an undirected graph $G(N, L_S, C, PD, P)$, where $N$ is a set of nodes (routers) and $L_S$ is a set of link $s$ between the nodes in $N$ , $C_{ij}$ is set of bandwidths of the link $s$. In the other words $C_{ij} \in C$ where $(i, j) \in L_S$. PD is set of propagation delays of link $(i, j) \in L_S$ and $P$ is the set of potential ingress–egress pairs. Furthermore let $(s, d)$ be a generic element of $P$.

4-1- Calculate link and path weight

The Analytic Hierarchy Process (AHP) is a theory of relative measurement with absolute scales of both tangible and intangible criteria based on the judgment of knowledgeable and expert people. How to measure intangibles is the main concern of the mathematics of the AHP. In the end we must fit our entire world experience into our system of priorities if we are going to understand it. The AHP reduces a multidimensional problem into a one dimensional one. Decisions are determined by a single number for the best outcome or by a vector of priorities that gives an ordering of the different possible outcomes. We can also combine our judgments or our final choices obtained from a group when we wish to cooperate to agree on a single outcome [38,39,40,41]. The AHP method consists of two hierarchical categories that range from bottom to top, including: 1.Comparing the pair of Alternatives to each of the criteria. 2. Comparing the pair of Alternatives to the original goal. In this method, hierarchical categories are used with crisp numbers. Despite the general acceptance, the AHP method is criticized for not being able to enter uncertain numbers and ambiguity of users in deciding on crisp numbers.

4-2- Calculate End to End Delay

End-to-end delay include time spent in buffers (queues), transmission time and propagation time through crossed links. The delay of packets on a network link is evaluated using $M/M/1$queuing theory. Then, for a packet of $L$ bytes traversing a link $i$ having a capacity of $C_i$ and a used bandwidth of $F_i$, the delay $D_i$ is given by:

$$D_i = \frac{L}{C_i - F_i} + d$$  \hspace{1cm} (1)
Where \( d \) is the release delay in the link \( i \) [47].

According to the class of network traffic, each link metric, namely, bandwidth, transmission delay and proposed availability can be used to select weight calculation. Therefore, each metric has a weight based on its importance in the decision maker. The decision maker employed is Fuzzy Analytical Hierarchy Process (FAHP)[38,40,44,45,46] that is a structured technique used to find the best solution based on the weight of metric according to their importance[48].

![FAHP method](image)

Figure 5. FAHP method.

Finding the best weight for each metric with FAHP method will be done in the following steps. The first step is constructing evaluation matrix. This is a matrix which shows the weight of each metric in comparison with other metrics.

<table>
<thead>
<tr>
<th>Table 1- FAHP routing metric weights</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
</tr>
<tr>
<td>Availability</td>
</tr>
<tr>
<td>Bandwidth</td>
</tr>
<tr>
<td>Delay</td>
</tr>
</tbody>
</table>
Where $m_1, m_2, m_3$ are the fuzzy scale considering the relative importance base on table 2.

Table 2- Linguistic scales for importance

<table>
<thead>
<tr>
<th>Linguistic scales for importance</th>
<th>$M$</th>
<th>$M^{-1}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Just equal</td>
<td>$(1,1,1)$</td>
<td>$(1,1,1)$</td>
</tr>
<tr>
<td>2 Equally Important</td>
<td>$(\frac{1}{2}, \frac{3}{2}, \frac{3}{2})$</td>
<td>$(\frac{2}{3}, 1, 2)$</td>
</tr>
<tr>
<td>3 Weakly more Important</td>
<td>$(1, \frac{3}{2}, 2)$</td>
<td>$(\frac{2}{3}, \frac{2}{3}, 1)$</td>
</tr>
<tr>
<td>4 Strongly more Important</td>
<td>$(\frac{3}{2}, 2, \frac{5}{2})$</td>
<td>$(\frac{2}{3}, \frac{1}{2}, \frac{2}{3})$</td>
</tr>
<tr>
<td>5 Very strongly More Important</td>
<td>$(\frac{5}{2}, \frac{3}{2}, \frac{5}{2})$</td>
<td>$(\frac{1}{2}, \frac{2}{1}, \frac{2}{3})$</td>
</tr>
<tr>
<td>6 Absolutely more important</td>
<td>$(\frac{5}{2}, \frac{3}{2}, \frac{5}{2})$</td>
<td>$(\frac{2}{7}, \frac{1}{3}, \frac{2}{7})$</td>
</tr>
</tbody>
</table>

Multiplying Matrix calculated by multiplying each row of the evaluation matrix and $K^{th}$ root element of this matrix produces the $K^{th}$ Root Matrix. By normalizing the FAHP matrix, Eigenvector Matrix which shows the weights of factors can be obtained as Table 3.

Table 3- Normalized eigenvector matrix

<table>
<thead>
<tr>
<th>Multiplying</th>
<th>$3^{rd}$ Root</th>
<th>Eigenvector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>$m_1m_2$</td>
<td>$\sqrt[3]{m_1m_2}$</td>
</tr>
<tr>
<td>Bandwidth</td>
<td>$m_3/m_1$</td>
<td>$\sqrt[3]{m_3/m_1}$</td>
</tr>
<tr>
<td>Delay</td>
<td>$1/m_2m_3$</td>
<td>$\sqrt[3]{1/m_2m_3}$</td>
</tr>
<tr>
<td>Sum</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Having calculated the Eigenvector Matrix, one should now compute three other matrixes called 
\( E_a, E_b, E_d \), which are considered as Availability Eigenvector Matrix, Bandwidth Eigenvector 
Matrix and Delay Eigenvector Matrix, respectively. Elements of these matrixes are:

\[ a_{ij} = \frac{a_i}{a_j}, \quad b_{ij} = \frac{b_i}{b_j}, \quad d_{ij} = \frac{d_i}{d_j} \]  

(2)

Respectively, \( a_k, b_k \) and \( d_k \) denote availability, bandwidth and delay of link \( k \), respectively. 
Finally FAHP-Value of links in the network can be obtained as equation (3).

\[ [C_1C_2C_3] = [E_aE_pE_d] \quad \text{Eigenvector Matrix} \]  

(3)

To study availability of a path, AHP-Values of links are used as members of that path. In 
this case, availability of a path is defined as the average availability of its links, Bandwidth of 
a path is extracted from minimum bandwidth of its links while delay of a path is the sum of 
delay calculated for links of a path. With this assumption, FAHP-Value of links for a path can 
be calculated. Then, Evaluation Matrix will be constructed for paths using three Eigenvector 
Matrixes of \( E_a, E_b \) and \( E_d \). After that FAHP-Value for \( m \) paths of the network is obtained as 
equation (4).

\[ [c_1'c_2'c_3'] = [E_aE_pE_d] \quad \text{Eigenvector Matrix} \]  

(4)

The weight of selected paths for load balancing and traffic adjustment is obtained from 
normalized weight \( W_k \) by equation (5).

\[ W_k = \frac{P_k}{\sum_{i=1}^{n} P_i}, \quad \text{s.t.} \quad \sum_{i=1}^{n} W_i = 1 \]  

(5)

Where, \( W_k \) is the normalize weight of path \( k \), \( P_k \) represents the weight of path \( k \). In 
equation (5), \( P_k \) is calculated from equation (6).

\[ P_k = \left( \frac{\text{FAHP Cost(Path)}_k}{\sum_{i=1}^{n} \text{FAHP Cost(Path)}_i} \right) \]  

(6)

In equation (6) bandwidth and transmission delay of links specify the initialized value for FAHP-
Value of each link.

5- Proposed routing algorithm

In this section, we present the proposed algorithm. In our proposed algorithm, requests are sent in 
\( (S,D, Bw_{Req}, Delay_{Req}, Priority) \) that \( S \) is the source and destination \( D \). And \( Bw_{Req} \) is the minimum 
bandwidth required and \( Delay_{Req} \) is the maximum delay end to end and \( Priority \) is the priority 
request.

First we find all the possible paths between different nodes. Then we calculate the weights of the 
links by Eq. (6). Then, the list of LSP requests is first requested in order of priority, and then sorted 
in ascending order of requested bandwidth. Now, we classification the max flow min cut algorithm[49] into all the network links as a set of min-cut links \( L_{mc} \) and a set of non min-cut 
\( L_{mc(non)} = L - L_{mc} \). Then we find a path for each request. In our proposed algorithm, among the 
links of the \( L_{mc} \) type, the link with the broadest bandwidth and the least delay is selected, then by the 
Dijkstra algorithm [50] it finds the shortest path and if the links are of the type \( L_{mc(non)} \), we will find 
the shortest path directly by the Dijkstra algorithm.

If the route is not found, the request will be rejected. Now, if the delay calculated by Eq. (1) is 
less than the maximum delay required and the remaining bandwidth is more than the requested 
bandwidth, then a route is established for the current request and we will update the remaining 
amount of bandwidth, otherwise we will save the current request as a rejected request. We will do all 
these steps for all requests. Finally, all rejected requests are routed again. The FAHP Routing Pseudo 
code is presented by Algorithm 1.
Algorithm 1 : MPLS Routing by Fuzzy AHP

1. Input: A network graph $G(N, L_q, C, PD, P)$, a LSP Request $(S, D, Bw_{Req}, Delay_{Req}, Priority)$
2. Output: Route with requested bandwidth and end to end delay or reject the request if no such a route exists.
3. Find all Possible Paths and Links
4. Delete all links that have residual bandwidth less requested bandwidth
5. for all $Link(i, j) \in L_q$
6. Compute FAHP_Value of link according to Eq. (6)
7. Sort the requests list based on the Priority requested and requested bandwidth
8. Classification of $links(L_{mc})$ and $links(L_{mc(non)})$
9. if $L_{mc}$ then
10. Choose the link with max bandwidth and min delay
11. Use Dijkstra Algorithm to the find Path
12. end if
13. if not path found then
14. return reject the request
15. else
16. name the found Path
17. end if
18. Delay(Path$_i$)=end to end delay of Path according to Eq. (1)
19. if ($(Delay(Path_i) \leq Delay_{Req})$ and $(Bw_{Req} < Residual Capacity))$
20. Route the request and update the residual link bandwidth
21. else
22. Save request to Reroute
23. end if
24. end for

Finally, all rejected requests are routed again. We first remove all links whose bandwidth is less than the requested bandwidth. And then we calculate the weights of the links by Eq. (6). Then we divide the network links into groups $L_{mc}$ and $L_{mc(non)}$ by the max flow min cut algorithm. To select a link between several links, the $L_{mc}$ type selects the link with the max bandwidth and the lowest delay, and then assigns a path to each request using the Dijkstra algorithm. Similarly, by Eq. (1), we obtain the delay of the end-to-end paths, but if the delay is less than the maximum delay required and the remaining bandwidth is more requested than the bandwidth, then a path for the rejected request will be established and we update the amount of remaining bandwidth.

Otherwise, the amount of bandwidth will be one of the path links that has the largest amount of bandwidth remaining, due to a delay reduction will increase by one unit. As a result, the latency is calculated again by Eq. (1) and we continue this process until the delay calculated is equal to the delayed request. Or otherwise the current request will be rejected and we will process these steps for all rejected requests. The FAHP Rerouting Pseudo code is presented by Algorithm 2.

Algorithm 2 : MPLS Rerouting Save LSP Requests by Fuzzy AHP

1. Input: A LSP Request $(S, D, Bw_{Req}, Delay_{Req}, Priority)$
2. Delete all links that have residual bandwidth less requested bandwidth
3. for all $Link(i, j) \in L_q$
4. Compute FAHP_Value of link according to Eq. (6)
5. end for
6. Sort the requests rejected list based on the Priority requested and requested bandwidth
7. Classification of $links(L_{mc})$ and $links(L_{mc(non)})$
8. if $L_{mc}$ then
9. Choose the link with max bandwidth and min delay
10. Use Dijkstra Algorithm to the find Path
11. end if
12. Use Dijkstra Algorithm to the find Shortest Path$_i$
13. if not path found then  
14. return reject the request  
15. end if  
16. while true do  
17. Delay(Path_i) = end to end delay of Path according to Eq. (1)  
18. if ((Delay(Path_i) ≤ Delay_{Req}) and (BW_{Req} < Residual Capacity))  
19. Route the request and update the residual link bandwidth  
20. else  
21. Find a link(i, j) ∈ Path_i with maximum residual bandwidth  
22. if link not found then  
23. return reject the request  
24. end if  
25. else  
26. Increment allocated bandwidth of link(i, j) by 1 unit  
27. end if  
28. end while  

6- Simulation  
We used two network topologies to do simulations. The first network topology that we used to simulate is the MIRA topology [51, 52] shown in Fig. 3. This is the topology consists of 15 nodes and 26 links. In this topology (S1, D1), (S2, D2), (S3, D3), (S4, D4) and (S5, D5) are the ingress–egress routers pairs. There are two different kinds of links in the network: The thin links which have a capacity of 1200 units and the thick links have a capacity of 4800 units (we all have the capacity to scale 100) and all the links are bidirectional.

![Figure 6. MIRA topology [51].](image)

The second network topology that we use to simulate is the ANSNET [53] topology. This is the topology consists of 32 nodes. In this topology (S1, D1), (S2, D2), (S3, D3), (S4, D4) and (S5, D5) are the ingress–egress routers pairs. The capacity of each link is 1200 units (we all have the capacity to scale 100) and all the links are bidirectional.

![Figure 7. ANSNET topology [53].](image)

Ingress–egress router pairs for LSP setup requests are chosen randomly from the above potential nodes. We assume that all LSPs are long lived, i.e., once an LSP is routed, it will not be terminated.
The total number of demands is 3000 for MIRA topology and ANSNET topology and granted paths are not removed from the network until the end of simulation [52].

7- Performance evaluation

In this subsection, the simulation results of our proposed MPLS_FAHP algorithm are presented and compared with other related works (MIRA; MDMF; BGLC; GA&PSO).

To investigate effect of different load conditions on the performance of the algorithms, we consider some scenarios. Also, to have realistic conditions, we consider traffic units as follows [52][54]:

- For one directional single voice call, the required bandwidth is 90.4 kbps (1 unit).
- For bi-directions single voice call, with symmetric flow, the required bandwidth is 180.8 kbps (2 units).
- The required bandwidth for a single video call is 676.8 kbps (7.5 units).
- Hence, for a bidirectional video conferencing session the required bandwidth is 857.6 kbps (9.5 units).

The experimented scenario for performance analysis of the algorithm is as follows:

- Scenario 1: requested bandwidth is chosen from \{1,2,3,4\} and end to end delay form \{95,96,97,98,99,100\} and the priority of requests from \{1,2,3,4\} using a uniform distribution.
- Scenario 2: requested bandwidth is chosen from \{1,2,3,4\} and end to end delay form \{60,61,62,63,64,65\} and the priority of requests from \{1,2,3,4\} using a uniform distribution.
- Scenario 3: requested bandwidth is chosen from \{1,2,7.5,9.5\} and end to end delay form \{95,96,97,98,99,100\} and the priority of requests from \{1,2,3,4\} using a uniform distribution.
- Scenario 4: requested bandwidth is chosen from \{1,2,7.5,9.5\} and end to end delay form \{60,61,62,63,64,65\} and the priority of requests from \{1,2,3,4\} using a uniform distribution.

7-1- Call blocking ratio

We use some parameters as the performance metric to compare the different algorithms. First metric is call blocking ratio that have the following definition [52, 55, 56]:

\[
Call \ blocking \ ratio = \frac{\text{Number of rejected requests}}{\text{Total Number of requests}}
\] (7)

Call blocking ratio shows the number of the requests that will be rejected because of the unavailability of desired capacity or congestion to total number of the requests. The call blocking ratio is under effect of the desired network resource utilization. If the call blocking ratio becomes more, the more requests are rejected. While the network may not be full yet and there are probably capacities in the network, inappropriate resource utilization and congestion of request lead to request rejection. So, call blocking ratio is a good performance criterion.

Simulation results are shown in Figs. 8 (MIRA topology) and 9 (ANSNET topology) in terms of call blocking ratio. In terms of the number of rejected requests, the MPLS_FAHP algorithm the lowest number and MIRA algorithm has the highest number of rejected requests. As you can see, with the increasing number of requests, the number of rejected requests in all algorithms also increases. Because by increasing the number of accommodated requests, capacity of the network decreases and chance of rejecting a new request increases.
7-2- Mean Length

We define the second metric as the mean length and can be found as follows [52, 55, 56]:

\[
Mean\ length = \frac{\sum_{i=1}^{NLSP} length(LSP_i)}{NLSP}
\]  

(8)
In Eq. (8), \( N_{LSP} \) shows the number of the set up paths and \( \text{length}(LSP) \) shows the number of the path links.

Mean length of the assigned paths to requests is a proper criterion to evaluate the performance of a routing algorithm. Because paths with more length use more links as resources of the network.

Fig. 10 and Fig. 11 show the mean length for the MPLS_FAHP algorithm compared to other algorithms. The algorithm MPLS_FAHP has performed better than other algorithms and MIRA algorithm has had the worst performance.

**Figure 10. Mean length in MIRA.**

**Figure 11. Mean length in ANSNET.**

### 7.3- Maximum Flow

The third metric is average max flow. The average max flow is defined as the available max flow between all the pairs of set \( P \) and can be found as follows [52, 55, 56]:

\[
\text{max flow}_{\text{avg}} = \frac{\sum_{(s,d) \in P} \text{max flow}(s,d)}{|P|}
\]  

(9)
15th Iran Media Technology Exhibition and Conference

Where $P$ is the set of ingress–egress pairs of the network. Note that the maximum flow (max flow) value between a ingress–egress pair, is an upper bound on the total amount of bandwidth that can be routed between the given ingress–egress pair. This metric represents the available capacity that is saved for the future demands.

Figures 12 (MIRA topology) and 13 (ANSNET topology) show the effect of network load on mean maximum flow of the network by the algorithms. The MIRA algorithm uses less critical links due to proper load distribution across the network and therefore keeps it up to maximum flow other than the other algorithms.

![Figure 12. Maxflow in MIRA.](image)

![Figure 13. Maxflow in ANSNET.](image)

8- T Conclusion

In this paper, we present MPLS Fuzzy AHP routing (MPLS FAHP) algorithm in mpls networks using the Fuzzy AHP method and Dijkstra algorithm, while guaranteeing the quality of service bandwidth and delay end to end. In the proposed algorithm for weighing links, we have used fuzzy
AHP techniques, which makes it impossible to use links that are in critical condition when routing. And in order to improve performance, we have implemented a new approach using high priority bits in network packets to transfer them to destinations. We proposed algorithm in MATLAB simulation. Also, we compared the performance of MPLS_FAHP with some other related algorithms (i.e. MIRA, MDMF, PSO&GA, BGLC). The results show that our proposed algorithm accepts more requests, thereby reducing the call blocking of requests, the result was that reduces network congestion. The results also showed that the path mean length between the source and destination in our proposed algorithm was performance better than other algorithms.

9- References

15th Iran Media Technology Exhibition and Conference


15th Iran Media Technology Exhibition and Conference


