Advanced Fuzzy Class-Based Routing in MPLS-TE Networks

MICHAL PIŠTEK, MARTIN MEDVECKÝ
Institute of Telecommunications
Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava
Ilkovičova 3, 812 19 Bratislava
SLOVAKIA
michal.pistek@stuba.sk, martin.medvecky@stuba.sk

Abstract: - The paper focuses on optimizing MPLS-TE networks using Constraint-Based Routing algorithm with implemented fuzzy logic. CBR includes routing algorithms that do not make routing decisions solely on the shortest path towards the destination but they can also base their decisions on various Quality of Service parameters or administrative needs. MPLS-TE networks can benefit from implementing such algorithms by utilizing less loaded paths or limiting the QoS parameters of the traffic to desired values. Implementing multiple constraints into algorithm can have issues with combining these constraints to make the decisions. Fuzzy logic represents an ideal tool for normalizing these constraints in [0,1] interval where the closer the constraint gets to 1, the better the path should be. The paper introduces a new CBR algorithm called Advanced Fuzzy Class Based Algorithm (AFCBA) which, besides setting the proper membership functions and cost calculations, deals with the tie break situations and what actions to take if no path satisfies the requirements of the traffic flows.

Key-Words: - Constraint Based Routing, Fuzzy Logic, MPLS, QoS, AFCBA, WSP, SWP

1 Introduction

In recent years, the exponential growth of traffic demand in communication networks has forced the telecommunication operators to seek the ways to maximize the performance of their networks without expanding the capacity of the links where it is possible. But implementing such mechanisms should always account for guaranteeing necessary performance values represented by Quality of Service (QoS) parameters. Multiprotocol label switching (MPLS) and Constraint-based routing (CBR) provide great symbiotic mechanisms which can optimize network performance without degrading the desired QoS parameters such as delay, jitter or loss rate. [1]

MPLS with implemented traffic engineering mechanisms (MPLS-TE) is a key network technology in nowadays core networks. MPLS-TE provides connection-oriented approach in IP networks. It creates end-to-end Label-switched paths (LSPs) where it can guarantee bandwidth and with the traffic engineering it can truly optimize the network's resources. It enables to use explicit routes which might not be ideal according to the routing algorithms but they enable to use network's resources more efficiently. MPLS-TE can also rely on the routing algorithms to build the LSPs. The third possibility is to use CBR algorithm to make the routing decision.

Constraint-based routing (CBR) represents a class of routing algorithms that base path selection decisions on a set of requirements or constraints, in addition to the destination. These constraints may be imposed by administrative policies, or by QoS requirements. Constraints imposed by policies are referred to as policy constraints, and the associated routing is referred to as policy routing (or policy-based routing). Constraints imposed by QoS requirements, such as bandwidth, delay, or loss, are referred to as QoS constraints, and the associated routing is referred to as QoS routing. [2]

The proposed CBR algorithm is a continuation of our previous work where we implemented fuzzy logic in our multi-constraint approach. The aim of the algorithm is to optimize the network's resources where it accounts for various requirements of various traffic classes. This way the highest priority traffic is most likely to use the shortest possible paths and the lowest priority traffic is routed through the underutilized paths in the network.

The metrics in AFCBA are represented by their membership functions which, as defined by fuzzy logic, can range from 0 to 1. This way we can treat them as equal and make the decision based on all the metrics at once. We use additional class-based metric weights to differentiate how much impact the particular constraint has on the particular traffic priority to make sure that the high priority classes are treated as best as possible whereas the lower priority classes use higher weights for the metrics which improve the overall network's utilization.

The rest of the paper is structured as follows. The next chapter provides a brief survey of CBR algorithms. Then in chapter 3 we present our algorithm with the proposed improvements. The chapter 4 consists of the simulation model and the simulation results. The chapter 5 then provides a conclusion of the article.

2 Related Work

Multiple CBR algorithms were proposed [3][4]. The basic CBR algorithms are Widest Shortest Path (WSP) and Shortest Widest Path (SWP) [5]. SWP optimizes on the maximum available bottleneck bandwidth first and, if there are multiple such paths, selects among them the one with the least hop count.

On the other hand WSP selects the path with maximum available bandwidth capacity of the bottleneck link among the paths having the least hop count. In [6] WSP was implemented with the metric of normalized hops. This metric normalizes the cost of the link based on the hop count. When the link advertizes its cost, the cost is relative to the costs of the other links. Another variation of WSP, TDWSP (Time-dependent WSP) tries to guess the future traffic demand and calculates the optimal paths based on these predictions. [7]

Minimum Interference Routing Algorithm (MIRA) tries to route the new connection so that it interferes as little as possible with the possible future demands by calculating the critical links. The algorithm tries to choose the paths which do not contain critical links which are the links that can negatively affect one or more ingress-egress pairs. [8]

Dynamic Online Routing Algorithm (DORA) is similar to MIRA but watches the number of the paths and not flows transmitting through the link. DORA defines n as the number of paths going through the particular link. The higher n is, the higher the probability of congestion is. Based on n value, DORA calculates the weights of the links, based on which it calculates the most optimal path. [9]

In [10], the new dynamic QoS routing algorithm was presented which is based on available bandwidth and delay. The algorithm has the following steps. Discard the path that has insufficient available bandwidth. Discard the path that does not meet the delay requirement. Discard the path that does not have available wavelength for every link of the path. Discard the path which is critical for the future traffic demands. Discard the path which has less available bandwidth.

Profile Based Routing (PBR) algorithm is based on the pre-known classes of the flows, named as profiles. The approach is splitting the domain resources into the profiles between ingress-egress pairs known a priori, which constitutes the first phase of the algorithm. In the second phase, the flows are admitted one at a time based on their resource requests and the remaining capacity in the traffic class they map into. [11]

Fuzzy Routing Algorithm (FRA) applies fuzzy logic to combine the desired goals into one condition. The goals are maximizing maxflow, i.e., the capacity of the bottleneck link on the path, maximizing the residual bandwidth on the links other than the bottleneck link, and minimizing path length for its number of hops. It defines membership functions for these goals and combines them in the defined rule which finds the best compromised path. [12]

3 Advanced Fuzzy Class Based Algorithm

The algorithm takes the same constraints and membership functions as it was proposed in [13] but we add more depth to the algorithm to account for situations where multiple paths with the same cost are calculated or when no sufficient path exists when the traffic flows arrives.

Advanced Fuzzy Class Based Algorithm (AFCBA) uses three metrics as its constraints and by implementing fuzzification on these constraints it can treat them as equal when the routing decision is made. The metrics that AFCBA uses are hop count, available bandwidth of bottleneck link and path utilization.

The hop count is represented by a_p membership function.

$$a_p = 1 - \frac{h_{max}+1}{h} \tag{1}$$

Here h represents the actual hop count of the particular path p. This function achieves the highest value (= 1) when only one hop is between the ingress LER and the destination (meaning that ingress and egress LERs are directly connected without any LSRs between them). It decreases linearly up to the $h_{max} + 1$ value where h_{max} represents maximum possible hop count.

The available bandwidth of bottleneck link is represented by b_p membership function.

$$b_p = b\frac{1}{b_{max}} \tag{2}$$

Value b represents actual available bandwidth of the particular path's bottleneck link. This function linearly rises up to the b_{max} value which represents maximum possible available bandwidth of the links in the network. The algorithm always uses the link with the lowest available bandwidth of the particular path.

The path utilization is represented by c_p membership function.

$$c_p = \begin{cases} 0, if\ c'_p < 0 \\ \frac{c'_p}{L}, otherwise \end{cases} \tag{3}$$

$$c'_p = \sum_{l_\mu \in p} l_\mu \tag{4}$$

$$l_\mu = \begin{cases} 1, if\ \mu_l < \mu_{min} \\ 1 - \frac{\mu_l - \mu_{min}}{\mu_{avg} - \mu_{min}}, if\ \mu_{min} < \mu_l < \mu_{avg} \\ -1 + \frac{\mu_l - \mu_{max}}{\mu_{avg} - \mu_{max}}, if\ \mu_{avg} < \mu_l < \mu_{max} \\ 0, if\ \mu_l > \mu_{max} \end{cases} \tag{5}$$

$$\mu_{avg} = \frac{\sum_{l=1}^{L} \mu_l}{L} \tag{6}$$

First the algorithm calculates the average utilization μ_{avg} of all the links of the network and then assigns a value l_μ to every link's utilization μ_l based on the following function. Positive l_μ value means that the link has lower than average utilization and the negative value means that its utilization is above average. For every path, the c'_p value is computed which achieves the highest value when all the links of the path have the utilization bellow the predefined minimum utilization value (μ_{min}). The final c_p value is normalized in [0, 1] interval. The aim of c_p value is to route the traffic through the underutilized paths in the network.

For every constraint, we define the following metric weights: weight $w_{1,c}$ for hop count, $w_{2,c}$ for available bandwidth of bottleneck link and $w_{3,c}$ for path utilization to take into consideration various needs of the traffic class c. The weight $w_{1,c}$ should be set high for those traffic classes which require low delay values (e.g. voice). For such classes the weight $w_{2,c}$ should be set low because it enables these classes to use short paths with lower available bandwidth of bottleneck link and on the other hand tries to limit the delay tolerant classes to use such paths. The weight $w_{3,c}$ should be set high for those traffic classes which can tolerate higher delay values (e.g. web traffic) and so there is higher chance that the longer and underutilized path will be chosen for these classes so the shorter ones can be later used by delay intolerant classes and the utilization in network is better distributed.

For better visualization we provide the membership functions a_p and b_p and the function l_μ in Fig. 1, 2 and 3. As mentioned earlier, note that the higher the function's value is, the better the path seems.

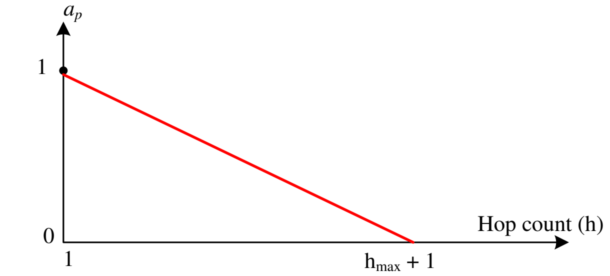

Fig. 1: Membership function a_p

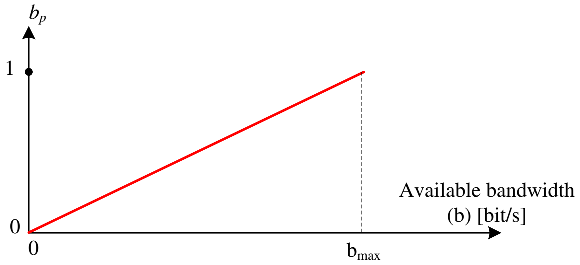

Fig. 2: Membership function b_p

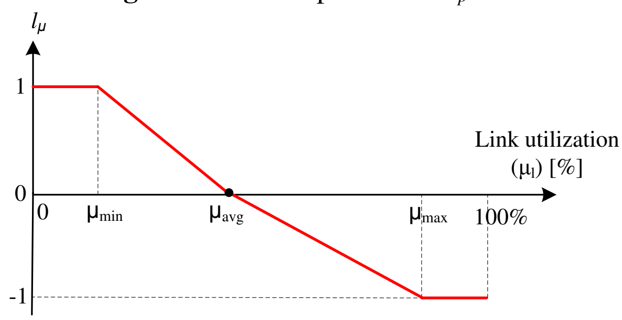

Fig. 3: Function l_μ representing normalized μ_l.

3.1 Proposed Improvements

We base the routing decision on the computed cost $cost_p$ which combines the three constraints and thus provides the best compromise of these constraints. Note that the algorithm calculates this cost for every possible path in the network. The best path for the incoming flow is then represented by the highest

$cost_p$ value. And consequentially the path (LSP) with this cost is chosen and explicitly established using CR-LDP protocol and the particular flow is mapped into this LSP.

$$cost_p = \beta \max\left(w_{1,c}a_p, w_{2,c}b_p, w_{3,c}c_p\right) + (1-\beta)(w_{1,c}a_p + w_{2,c}b_p + w_{3,c}c_p) \quad (7)$$

$$\beta = \begin{cases} 1, if\ \mu_{avg} < \mu_1 \\ 1 - \frac{\mu_{avg}-\mu_1}{\mu_2-\mu_1}, if\ \mu_1 < \mu_{avg} < \mu_2 \\ 0, if\ \mu_{avg} > \mu_2 \end{cases} \quad (8)$$

Here, β parameter represents a number which is used as a representation of OR/AND logical operator in fuzzy logic OWA (Ordered Weighted Averaging) operation where the higher β is, the more it resembles OR. We define β as a function of the average link utilization (as calculated in (6)) so that AFCBA uses the left side of (7) equation when the network's utilization is low and the right side when the utilization rises (See Fig. 4).

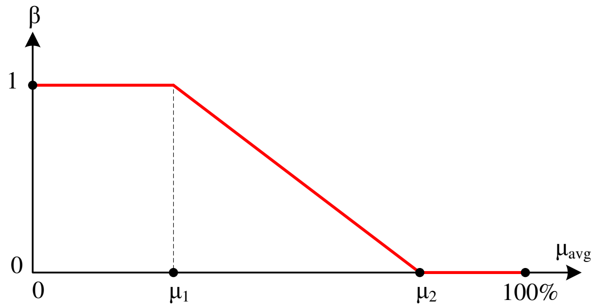

Fig. 4: β as a function of μ_{avg}

This way when the network is underutilized the traffic classes are routed through the path with the best constraint out of the three (a_p, b_p and c_p) and because predefined metric weights influence the traffic classes' priority of the constraints, the path with the most suitable constraint of the particular traffic class is chosen. The more utilized the network becomes, the more compromised path, which accounts for all the constraints, is chosen.

Because using only the left side of (7) can result in more paths with the same results, β is set to 0 if such case occurs. If there are still multiple paths with the same result, the path is chosen based on the underlying shortest path first (SPF) algorithm.

When the network is heavily loaded there can be situations where no path can fulfil the traffic flow's requirements. AFCBA does not function as an admission control mechanism so it still has to choose a path to route the traffic. Our aim in such situations is to choose the path which does not influence the highest priority traffic class so that it can be transmitted without loss and added delay and only lower priority classes will be influenced. So AFCBA discards all the paths which have at least one link that is currently used by highest priority class. If multiple such paths exist, AFCBA chooses the one with the most available bottleneck bandwidth so that the lower priority classes are influenced as little as possible.

See Fig. 5 for the detailed flowchart of the functionality of AFCBA.

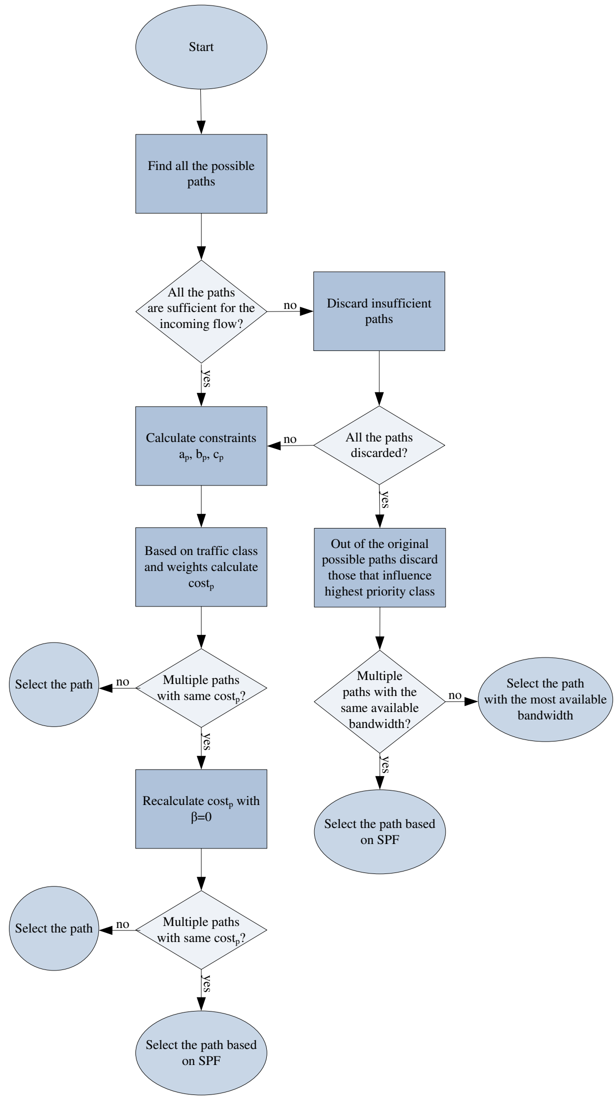

Fig. 5: AFCBA's flowchart

4 Simulations

The simulations were performed using Network Simulator 2 (NS-2). NS-2 is an open source discrete events simulator that is developed in C++ and OTcl. Its reliability for simulating MPLS was questioned in [14] where the real measurements of dealy values differed from the simulation results. Although we compare the algorithms also based on the delay values, these values are dependent mostly on the hop count of the particular path so the NS-2's inaccuracy in this regard does not pose a problem for our simulations.

The network topology (Fig. 6) consists of two Label Edge Routers (LER) and seven Label Switching Routers (LSR) forming fourteen possible paths between the traffic source and the traffic receivers which we list in Table 1 for future referencing. These paths differ from each other in various link's bandwidth (from 2 Mbit/s to 4 Mbit/s), which is indicated by the numbers on the links, and in the hop count (from 2 to 5 hops).

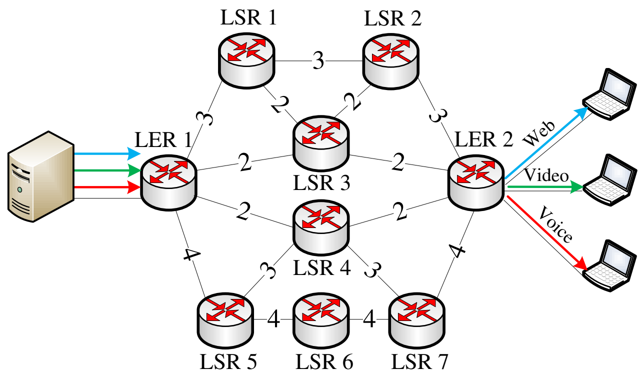

Fig. 6: Simulation model

TABLE 1 – Possible Paths

Path between LER1 and LER2	Path number
LSR3	1
LSR4	2
LSR1 – LSR2	3
LSR1 – LSR3	4
LSR3 – LSR2	5
LSR4 – LSR7	6
LSR5 – LSR4	7
LSR1 – LSR2 – LSR3	8
LSR1 – LSR3 – LSR2	9
LSR3 – LSR1 – LSR2	10
LSR5 – LSR4 – LSR7	11
LSR5 – LSR6 – LSR7	12
LSR4 – LSR5 – LSR6 – LSR7	13
LSR5 – LSR6 – LSR7 – LSR4	14

Three different traffic classes were used in our simulations representing voice, video and web traffic, voice traffic having the highest priority and web traffic having the lowest one. We use constant bit rate generator for voice and video traffic and exponential generator for web traffic. Each traffic class is represented by three flows. See Table 2 for arrival flows of the particular flows.

TABLE 2 – Flows' Arrival Times

Traffic flow	Start [s]
Web1	1
Video1	2
Voice1	3
Web2	4
Video2	5
Voice2	6
Web3	7
Video3	8
Voice3	9

Note that the order is web flow then video flow and voice flow as the last. We use this order to make it the most difficult for highest priority voice flows to find a suitable path. But by using proper constraint weights, AFCBA should be able to find one.

Note that for web traffic we use the highest weight for path utilization constraint so it could be routed through the underutilized paths. For voice traffic we use the highest weight for hop count constraint so it could be routed with the minimal delay. Video traffic has the same weight values for all the constraints. The used constraint weights are as following:

- $w_{1,voice} = 0.5$
- $w_{2,voice} = 0.167$
- $w_{3,voice} = 0.333$
- $w_{1,video} = 0.333$
- $w_{2,video} = 0.333$
- $w_{3,video} = 0.333$
- $w_{1,web} = 0.167$
- $w_{2,web} = 0.333$
- $w_{3,web} = 0.5$

The last tuning parameters represent utilization intervals where μ_{min} and μ_{max} are used for l_μ represented in (5) and μ_1 and μ_2 are used for β represented in (8).

- $\mu_{min} = 10\ \%$
- $\mu_{max} = 90\ \%$
- $\mu_1 = 15\ \%$
- $\mu_2 = 40\ \%$

4.1 Simulation 1

In this scenario we performed simulations where for every arriving traffic flow, AFCBA is able to find a sufficient path. The bit rates for the arriving flows are as follows:

- Web flows: 1000 kbit/s
- Video flows: 1500 kbit/s
- Voice flows: 500 kbit/s

The path selection is based on (7) equation where β is dependent on average link utilization (μ_{avg}) computed in (6) so at the very beginning of the simulation when the first flow Web1 arrives, $\beta = 1$. Because all the bottleneck links have 0% utilization b_p constraint has highest value (= 1) for every path which causes that all the paths choose b_p as the best metric but with the same value.

When AFCBA detects multiple best paths it recomputes $cost_p$ value with $\beta = 0$ and all the constraints are used resulting in Path 12 selection. See Table 3 for $cost_p$ values of paths from Web1's point of view. Note that the original $cost_p$ is equal to 0.5 because $w_{2,web} = 0.5$.

TABLE 3 – The Values Of $cost_p$ For Web1 Flow

Path #	Original $cost_p$	Recomputed $cost_p$
Path 1	0.5	0.8
Path 2	0.5	0.8
Path 3	0.5	0.85
Path 4	0.5	0.77
Path 5	0.5	0.77
Path 6	0.5	0.77
Path 7	0.5	0.77
Path 8	0.5	0.73
Path 9	0.5	0.82
Path 10	0.5	0.73
Path 11	0.5	0.82
Path 12	0.5	**0.9**
Path 13	0.5	0.7
Path 14	0.5	0.7

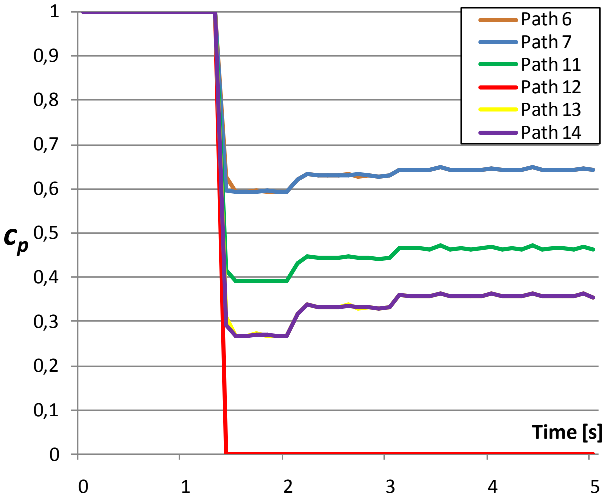

Fig. 7: Values of c_p membership function for path affected by Web1's arrival

After the selection of Path 12, its $cost_p$ value is lowered because path utilization rises which lowers the c_p value. As you can see in Fig. 7, selecting Path 12 causes lower value for another 5 paths (7, 11, 12, 13 and 14). Fig. 7 also shows that c_p value decreases most for the paths which have the most links in common with the chosen Path 12. As you can see, the given paths were not chosen by any traffic flow until the 5[th] second of simulation.

Similar situation happens with Video1 and Voice1 flows' arrivals. See Table 4 for the path selection of these flows.

There are 8 paths with the same best $cost_p$ value for Video1 flow so recomputation is needed which leaves the Path 3 as the only best path. See Fig. 8 for the c_p values of the affected paths.

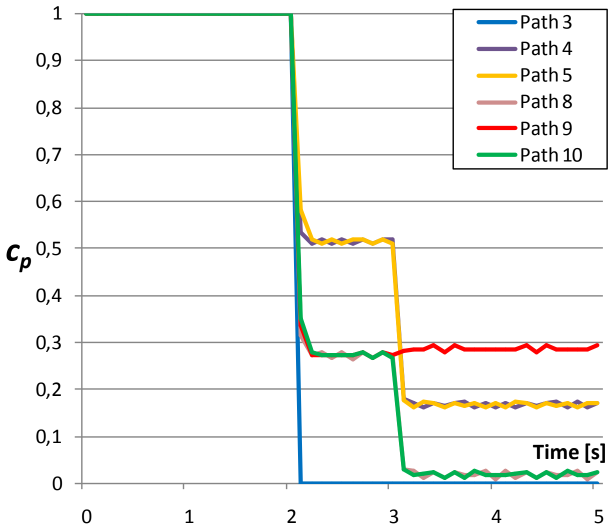

Fig. 8: Values of c_p membership function for path affected by Video1's arrival

TABLE 4 – The Values Of $cost_p$ For Video1 And Voice1 Flows

Path #	Original $cost_p$ Video1	Recomp. $cost_p$ Video1	Original $cost_p$ Voice1	Recomp. $cost_p$ Voice1
1	0.333	0.766	0.371	**0.817**
2	0.333	0.766	0.371	**0.817**
3	0.333	**0.783**	0.085	0
4	0.333	0.699	0.2	0
5	0.333	0.699	0.2	0
6	0.197	0	0.24	0
7	0.199	0	0.24	0
8	0.333	0.633	0.111	0
9	0.333	0.716	0.109	0
10	0.333	0.633	0.111	0
11	0.13	0	0.173	0
12	0.249	0	0.141	0
13	0.089	0	0.125	0
14	0.089	0	0.125	0

When Voice1 flow arrives there are only two paths with same $cost_p$ values but in this case the recomputation does not help because both paths are identical from the point of view of all the constraints. Here the Path 1 is selected as the result of the SPF algorithm. Note that AFCBA sets 0 to paths with less than the best original $cost_p$ so that they do not interfere with the original best paths.

With more flows arriving, the average link utilization is rising and consequentially β is declining. This means that right side of (7) has higher and higher weight so the best compromised path is selected instead of a path where one constraint achieves very good value.

See Fig. 9 for the actual average link utilization during the first 11 seconds of the simulation and Table 5 for β values when the particular flow arrives.

TABLE 5 – The Average Link Utilization And The Corresponding β Value For The Arriving Traffic Flows

Traffic flow	Average link utilization [%]	β value
Web1	0	1
Video1	6.74	1
Voice1	16.55	0.922
Web2	20.04	0.8
Video2	19.92	0.803
Voice2	32.45	0.302
Web3	39.99	0
Video3	26.65	0.534
Voice3	43.6	0

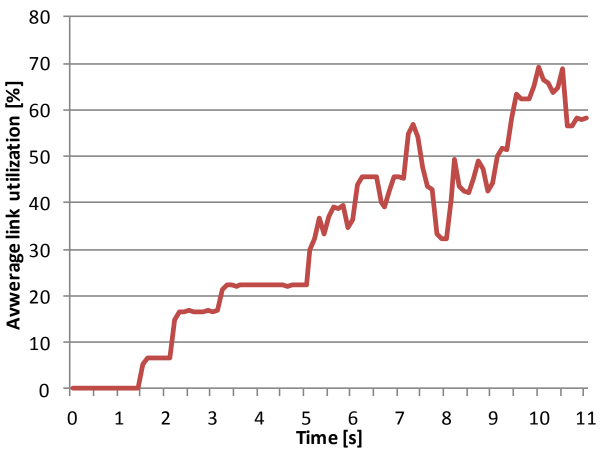

Fig. 9: Average link utilization

For the rest of the flows there was no need for $cost_p$ recomputation so the values in Table 6 represent the original $cost_p$ values. The values marked with "*" represent paths which achieved better $cost_p$ value but are not selected because they cannot fulfill the arriving flow's requirements.

TABLE 6 – The Values Of $cost_p$ For The Rest Of The Flows

Path #	$cost_p$ Web2	$cost_p$ Video2	$cost_p$ Voice2	$cost_p$ Web3	$cost_p$ Video3	$cost_p$ Voice3
1	0.152	0.177	**0.429**	0.216	0.208	0.442
2	**0.56**	0.418*	0.321	0.219	0.535*	**0.817**
3	0.147	0.167	0.273	0.222	0.22	0.3
4	0.126	0.117	0.368	0.286	0.159	0.342
5	0.13	0.121	0.366	0.28	0.162	0.341
6	0.374	0.287*	0.277	0.241	0.372*	0.623
7	0.374	0.287*	0.293	0.25	0.372*	0.628
8	0.047	0.054	0.246	0.199	0.146	0.242
9	0.181	0.146	0.303	0.388	**0.226**	0.2
10	0.059	0.064	0.259	0.278	0.229*	0.219
11	0.295	0.232	0.349	**0.456**	0.314*	0.521
12	0.262	**0.278**	0.22	0.193	0.267*	0.362
13	0.218	0.167	0.111	0.116	0.208	0.351
14	0.218	0.167	0.111	0.119	0.208	0.354

We compared AFCBA with WSP and SWP algorithms in terms of delay and jitter values. SWP firstly chooses based on the most available bottleneck bandwidth and if there are more such paths, SWP chooses the one with the least hops. WSP is similar but has the two steps changed so it firstly chooses based on the hops and then based on the available bottleneck bandwidth. See Fig. 10 and 11 for the results.

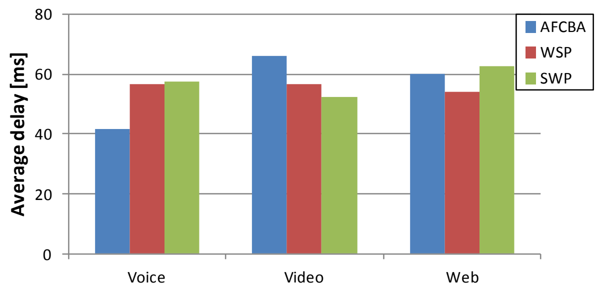

Fig. 10: Average delay comparison

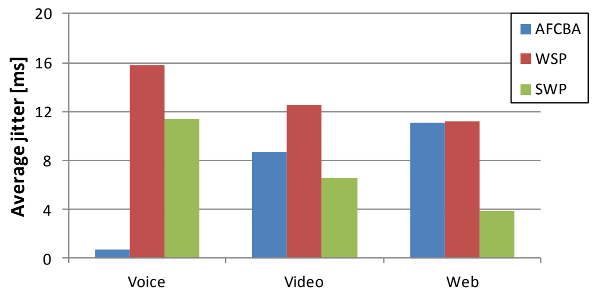

Fig. 11: Average jitter comparison

As you can see, AFCBA achieved the best results for the highest priority voice traffic although its flows arrived as the last in each cycle. This is due to the fact that web traffic and also video traffic had lower weight of the hop count constraint so it could use the shorter paths whereas the lower the priority of the traffic the bigger the weight of the path utilization constraint was so the web traffic prioritized underutilized paths.

4.1 Simulation 2

This scenario represents a situation when AFCBA is unable to select a suitable path for an incoming flow triggering the path selection which avoids the links that participate in transferring the high priority voice traffic. To achieve this situation we raise the bit rate of voice flows to 850 kbit/s so the bit rates of the traffic classes are as follows:

- Web flows: 1000 kbit/s
- Video flows: 1500 kbit/s
- Voice flows: 850 kbit/s

TABLE 7 – The Average Link Utilization And The Corresponding B Value For Arriving Traffic Flows In Scenario 2

Traffic flow	Average link utilization [%]	β value
Web1	0	1
Video1	6.74	1
Voice1	16.55	0.922
Web2	22.36	0.706
Video2	22.25	0.71
Voice2	36.43	0.21
Web3	45.71	0
Video3	32.26	0.31
Voice3	44.33	0

The values of β depicted in Table 7 are fairly similar to Scenario I with slight changes after Voice1 flow arrives.

The path selection is the same up to Video2 flow's path. See Table 8 for the path selection of the remaining flows.

When Video3 flow arrives there is no path in the network that could fulfil its requirement of 1500 kbit/s. AFCBA then looks for those path that do not interfere with high priority (voice) traffic (marked with green) and discards those that do interfere (marked with red). Then AFCBA does not compute $cost_p$ but instead selects the path with the most available bottleneck bandwidth so it has the least impact on the existing traffic going through the selected path and the paths which have some of the links in the selected path. In this case there are three viable paths (Paths 2, 3 and 9). Because Path 2 has the most available bottleneck bandwidth (1000 kbit/s compared to 500 kbit/s of Paths 3 and 9) it is chosen for the Video3 flow. Note that if more paths with the same available bottleneck bandwidth existed, SPF would be used to break the tie.

TABLE 8 – The Path Selection Based On $cost_p$ And Available Bottleneck Bandwidth (ABB) Values For The Last Four Flows

Path #	$cost_p$ Voice2	$cost_p$ Web3	ABB [kbit/s] Video3	$cost_p$ Voice3
1	0.364	0.271	1150	**0.45**
2	0.357	0.219	**1000**	0.419
3	0.301	0.222	500	0.361
4	0.342	0.364	500	0.446
5	0.341	0.356	500	0.438
6	0.315	0.183	650	0.319
7	0.331	0.186	650	0.319
8	0.225	0.273	500	0.31
9	0.332	**0.414**	500	0.401
10	0.252	0.338	500	0.339
11	**0.384**	0.245	650	0.304
12	0.238	0.118	650	0.268
13	0.12	0.116	650	0.119
14	0.12	0.119	650	0.119

As in Scenario I, we compare AFCBA with WSP and SWP. Because there were insufficient resources in the network for every algorithm, we compare them also based on the loss rate which also occurs in the scenario. See Fig. 12 and 13 for the results.

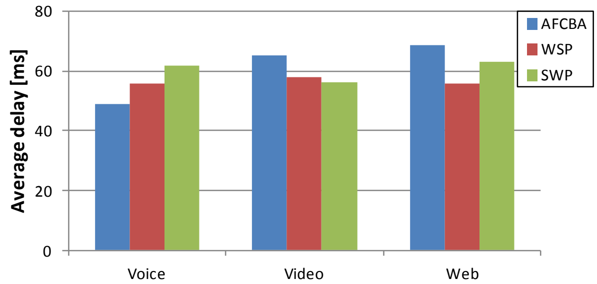

Fig. 12: Average delay comparison

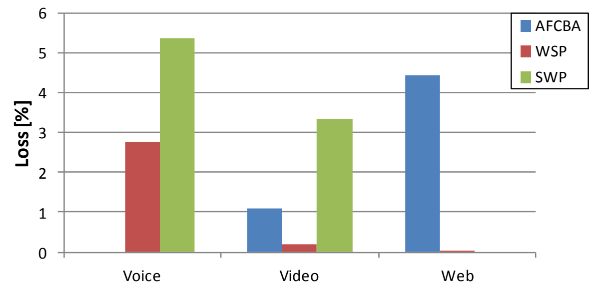

Fig. 13: Loss comparison

Because AFCBA avoids the paths which carry the voice traffic when the congestion occurs, the voice traffic achieved 0% loss rate and the loss increased with decreasing priority. WSP and SWP achieved the opposite results because they do not make such class-based decisions and because the voice traffic flows always arrived as the last in each cycle resulting in less optimal paths compared to web and video traffic.

5 Conclusion

We proposed a new CBR algorithm called Advanced Fuzzy Class Based Algorithm (AFCBA). AFCBA defines three membership functions for the three constraints and then combines them to compute the one cost value to determine the best path. This computation is class-based thanks to the constraint weights which signify the importance of each constraint to the particular traffic class. In this paper, we focused on the situations where multiple paths achieve the same properties and the situations where no feasible path could be found by the algorithm.

The first simulation stressed the situations where multiple paths achieved the same results and this forced AFCBA to recomputation resulting in the only one best path when any tie break happens. We compared AFCBA with WSP and SWP algorithms where our algorithm performed best in terms of delay and jitter values achieved by higher priority flows.

As shown in the second simulation when congestion occurs, AFCBA protects high priority flows by avoiding the paths which would have impact on this traffic, resulting in 0 % loss rate of voice traffic compared to 2.76 % and 5.37 % for WSP and SWP, respectively.

The presented simulation results show that the network can benefit from implementing AFCBA where it optimizes the network utilization with the lower priority flows and prioritizes the higher priority flows when it is possible.

6 Acknowledgement

This work is a part of research activities conducted at Slovak University of Technology Bratislava, Faculty of Electrical Engineering and Information Technology, Institute of Telecommunications, within the scope of the project „Optimalization of resource utilization in MPLS networks“ as part of the Grant programme to support young researchers of STU.

References:

[1] E. Chromy, M. Jadron, M. Kavacky and S. Klucik, “Admission Control in IMS Networks,” in *Journal AEEE - Information and Communication Technologies and Services*, Volume 11, Number 5, 2013, Special Issue, pp. 373-379, E-ISSN 1804-3119, ISSN 1336-1376.

[2] O. Younis and S. Fahmy, “Constraint-Based Routing in the Internet: Basic Principles and Recent Research,” in *IEEE Communications Surveys and Tutorials*, pp. 2-13. ISSN: 1553-877X, 2003.

[3] A. Karaman, “Constraint-Based Routing in Traffic Engineering,” in *Proceedings of the IEEE International Symposium on Computer Networks (ISCN '06)*, pp. 1–6, Istanbul, Turkey, 2006.

[4] T.C. Hung, N.H. Thanh, N.D. Thang, H.W. Jung, T.Kim, S.H. Kim and W.J.Yang, “Advanced Routing Algorithms and Load Balancing on MPLS,” in *Proceedings of the 9th International Conference on Advanced Communication Technology (ICACT '07)*, vol. 3, pp. 1886–1891, Gangwon-Do, Republic of Korea, February 2007.

[5] Q. Ma and P. Steenkiste, “On Path Selection for Traffic with Bandwidth Guarantees,” in *Proceedings of IEEE International Conference on Network Protocols*, pp. 191-202, October 1997.

[6] A.F. Khalifeh and A.H. El-Mousa, “ QoS Routing of VoIP using a Modified Widest-Shortest Routing Algorithm,” In *IEEE/ACS International Conference on Computer Systems and Applications*,” 2007 (AICCSA '07), May 2007. pp. 166-123. ISBN: 1-4244-1030-4

[7] Y. Yang and C. Lung, “The role of traffic forecasting in QoS routing - a case study of time-dependent routing,” In *IEEE International Conference on Communications (ICC 2005)*, vol. 1, May 2005. pp. 224- 228. ISBN: 0-7803-8938-7.

[8] M.S. Kodialam and T.V. Lakshman, “Minimum Interference Routing with Applications to MPLS Traffic Engineering,” In *INFOCOM (2)*, 2000, pp. 884–893.

[9] T.C. Hung, N.H. Thanh, N.D. Thang, H.W. Jung, T. Kim, S.H. Kim and W.J. Yang, “Advanced Routing Algorithms and Load Balancing on MPLS,” In: *Proc. 9th ICACT*, February 2007, vol. 3. pp. 1886 –1891. ISSN: 1738-9445.

[10] S. Kulkarni, R. Sharma, P. Sharan and R.B.R. Gowd, “New Dynamic QoS Routing Algorithm for Optical Networks Based on MPLS With Delay And Bandwidth as Constraints,” In *2012 International Conference on Optical Engineering (ICOE),* Belgaum, July 2012. pp. 1-6. ISBN: 978-1-4673-2462-5.

[11] A. Karaman, “Constraint-Based Routing in Traffic Engineering,” In *ISCN: IEEE*, 2006, pp. 49-54.
[12] J.A. Khan and H.M. Alnuweiri, “A Fuzzy Constraint-Based Routing Algorithm for Traffic Engineering,” *IEEE/Globecom*, Dallas, USA, 2004.
[13] M. Pištek and M. Medvecký, “Class-Based Constraint-Based Routing with Implemented Fuzzy Logic in MPLS-TE Networks,” *Journal of Computer Networks and Communications*, vol. 2014, Article ID 237810, 7 pages, 2014. doi:10.1155/2014/237810.
[14] H. Coll, J. Lloret and F.J. Sanchez, “Does ns2 really simulate MPLS networks?,” *The Fourth International Conference on Autonomic and Autonomous Systems (ICAS 2008)*, Gosier, Guadeloupe. March 16-21, 2008.