

Document downloaded from:

<http://hdl.handle.net/10251/140823>

This paper must be cited as:

Rego Mañez, A.; Sendra, S.; Jimenez, JM.; Lloret, J. (09-2). Dynamic Metric OSPF-Based Routing Protocol for Software Defined Networks. Cluster Computing. 22(3):705-720.
<https://doi.org/10.1007/s10586-018-2875-7>



The final publication is available at

<https://doi.org/10.1007/s10586-018-2875-7>

Copyright Springer-Verlag

Additional Information

Dynamic Metric OSPF-Based Routing Protocol for Software Defined Networks

Albert Rego¹, Sandra Sendra^{1,2}, Jose M. Jimenez¹, Jaime Lloret¹

¹ Instituto de Investigación para la Gestión Integrada de zonas Costeras, Universitat Politècnica de València.

Carrer del Paranimf, 1, 46730 Grau de Gandia, València, Spain.

² Dept. of Signal Theory, Telematics and Communications Department (TSTC), Universidad de Granada

C/ Periodista Daniel Saucedo Aranda, s/n. 18071 Granada, Spain.

alremae@doctor.upv.es, ssendra@ugr.es, jojher@dcom.upv.es, jlloret@dcom.upv.es

Abstract. Routing protocols are needed in networking to find the optimal path to reach the destination. However, networks are changing both their use finality and their technology. Paradigms like Software Defined Networks (SDNs) introduce the possibility and the necessity to improve the routing protocols. In this paper, a modification of the Open Shortest Path First (OSPF) routing protocol is proposed in order to allow the protocol to change the metric calculation dynamically according to the network requirements. Experiments, which compare our proposal against the OSPF protocol, are performed in five different scenarios. In these scenarios, the performance of the multimedia traffic has been increased 33% in terms of bandwidth utilization, 80% of loss rate reduction and delay reduction on VoIP communications.

Keywords: Routing; OSPF; SDN; dynamic metric; multimedia;

1 Introduction

Nowadays, networks are used for different applications. These applications require network resources of different nature. For example, multimedia applications do not require the same resources as massive downloads or applications in IoT environments.

OSPF [1] is one of the network protocols, of hierarchical interior routing, more used in the current networks. It uses the status of their links to calculate the optimal route to the destination, and employs a metric called cost, whose value is determined by the division of a reference bandwidth between the actual bandwidth of the interface.

~~Current n~~ Networks need a new architecture to solve the problems experienced ~~that~~ are related to their increase in complexity, incoherent policies, impossibility of escalation and dependence on manufacturers. Software Defined Networking (SDN) [2] has a network architecture that provides, among other capabilities, the dynamism, manageability and adaptability necessary to solve the problems mentioned above.

Moreover, SDN can provide support for the necessary network services. It can help in obtaining statistics of the flows that pass through the network nodes.

According to the work presented by Jimenez et al. [3], SDN is presented as a very interesting option to manage the sending of multimedia traffic flows.

In the scope of computer networks, through the use of Artificial Intelligence (AI), based on empirical studies we can use different techniques that allow the creation of adaptive mechanisms to improve their functioning.

Our work is based on the modification of the OSPF metric adapted to SDN. The modification provides the dynamic choice of the metric depending on the type of traffic sent by the network. Moreover, we use AI. We employ a traffic classifier developed in [4] that allows detecting different types of traffic. This work has been designed to focus on multimedia traffic.

Due to the impossibility of OSPF protocol adaptation in certain network circumstances, as its metric is based on bandwidth and is not capable of discriminating flows, all traffic is redirected by the link ~~that~~ ~~which~~ has the highest bandwidth. When using SDN, the cost associated with each interface can be the bandwidth not used by the existing flows. This is useful when the bandwidth is the metric to be used by the network traffic. SDN capabilities bring us the possibility of adding load balancing techniques. It allows measuring, when executing jump to jump decisions, the traffic load and checks the problems in order to take decisions for having different traffic balancing.

This article is an extension of the work presented in [5], where the possibilities of applying OSPF in SDN were tested. In this paper, OSPF is modified in order to add new functionalities, change the metric calculation taking advantage of SDN capabilities.

The rest of this document is structured as follows. Section 2 presents some of the most relevant works related to our study. The proposal is described in Section 3, where the architecture, protocol and messages are discussed. The experiments performed and their results are shown in Section 4. Finally, the conclusion and future work are presented in Section 5.

2 Related Work

In this section, the current state of the art is exposed. Some of the recent works performed by different researches are explained and summarized.

Taking into account the traffic that is generated in the current networks, we are faced with the need to improve and adjust the classic routing protocols. In this way, data transmissions must be optimized especially transmissions that require special treatment to be able to meet specific requirements due to their nature. This section presents some of the most interesting proposals where authors have performed different experimental tests with network protocols and have tried to improve them by modifying the original operation.

The performance of Interior Gateway Protocols has been evaluated in several works. In general, all authors recommend the use of the OSPF protocol as the Interior Gateway Protocol in the networks. Authors such as Sendra et al. in [6] have studied

the most used Interior Gateway Protocols. In their conclusions, they say that the OSPF protocol must be chosen by network administrators when there is restriction in the bandwidth of the network they manage. Also, other authors such as Rakheja et al. in [7] have conducted studies comparing the performance of the RIP, OSPF, IGRP and EIGRP protocols. In their conclusions, they assert that the OSPF protocol is the one with the best overall performance.

During the last years, there have been multiple works in which their authors show improvements for the routing protocols based on artificial intelligence (AI).

In their work [8], Sandra et al. propose the implementation of an intelligent routing protocol in SDN. According to their proposal, the intelligent protocol uses a metric based on previous learning, which allows the choice of the best routes for the transmission of data in the network.

Other authors, such as Barbancho et al. in [9], present the SIR (Sensor Intelligence Routing) algorithm. This new routing algorithm is aimed at achieving a better QoS. It applies to an artificial neural network based on Kohonen's self-organized feature map. Moreover, Barbancho et al. in [10] compare the performance of their SIR routing algorithm with directed diffusion and Energy-Aware Routing. Finally, they affirm that the inclusion of AI techniques in Wireless Sensor Networks (WSN) improves the network performance.

Many authors have studied the use of Swarm Intelligence (SI) to solve the problem of adaptive routing in telecommunication networks. Some of these authors use SI applying intelligence models based on biological swarm (ant colony optimization, particle swarm optimization, swarm robotics, and other swarm intelligence algorithms) for solving problems in the real world of sensor networks.

Arabshahi et al. in [11] study solutions to the problem of routing in wireless networks using SI as a key. They ~~are looking seek~~ to maintain the desired QoS, checking bottlenecks and looking for an adaptive network.

Gunes et al. in [12] present a new protocol called Ant-Colony-Based Routing Algorithm (ARA) that is highly adaptive, efficient and scalable. Their protocol is based on SI, focused on the ant colony based meta-heuristic. The main goal in the design of the protocol was to reduce the overhead caused by the routing protocol.

As actual networks are increasingly large, dynamic and heterogeneous, Ducatelle et al. in [13] indicate that, for their control and management, they require algorithms and novel protocols that are completely distributed. ~~Moreover, they should be at the same time and that at the same time are~~ adaptable, robust and scalable. These protocols also allow the network to behave as an autonomous and self-organizing system. In their work, they review several SI applications, considering routing algorithms in wired and wireless networks, while establishing principles to apply SI to the design of routing algorithms. They also indicate that research fields such as gossip/epidemics algorithms are closely related to SI.

Rajagopalan et al. in [14] present a routing protocol called Ad hoc Networking with Swarm Intelligence (ANSI). Their protocol, through self-organized SI mechanisms, makes better sending decisions than traditional MANET protocols, as they gather more information. In their study, they perform simulations comparing the ANSI and AODV (Ad hoc On Demand Distance Vector) protocols [15]. They verified that ANSI provides better performance results and fewer route errors.

Authors such as Zungeru et al. in [16] show a comparative study of classic routing protocols regarding the use of SI in WSN. They also present the results of a simulation of different protocols in MATLAB to serve as a reference in the future for other researchers.

One of the preferred techniques to carry out routing in WSN, maintaining its maximum useful life, is Clustering. Karaboga et al. in [17] present a Clustering mechanism based on the artificial bee colony algorithm, to prolong the life of the network. They compare the proposed algorithm with LEACH-based protocols. According to the results obtained, Clustering, based on artificial bee colony algorithm, can be applied to WSN routing protocols.

There are also authors who propose the use of SDN to obtain greater efficiency in routing, not only using traditional protocols such as OSPF, IS-IS [18] or BGP [19], but also some hybrid solutions.

The authors Caria et al. in [20] present a hybrid operation mode SDN/OSPF. In their proposal, they use SDN nodes to divide the OSPF domain into subdomains. Within each subdomain, the routing remains stable. The SDN nodes are located at the borders of the subdomains, and are responsible for adjusting the routing updates. In this way routes between subdomains can be optimized. According to their simulation results, it is not necessary to get to deploy SDN on all nodes of the network. In a similar way, Rothenberg et al. in [21] propose [a controller-centric hybrid networking model and present the design of the Route-Flow Control Platform \(RFCP\) along the prototype implementation of an AS-wide abstract BGP routing service, a hybrid network model focused on the controller. Furthermore, while they presenting the RouteFlow Control Platform \(RFCP\) design and, according to the implementation of a prototype for a BGP routing service.](#)

Also, there are authors who propose the use of SDN in ad hoc vehicular networks (VANET). Ming et al. in [22] indicate that in order to efficiently send information in VANET, a protocol that has a short delivery delay time and where low routing overhead is required. They propose a routing protocol based on SDN in which a central controller collects all the information of the rest of the controllers and calculates the optimal routes, based on all the information collected in other points of the network.

Finally, some modifications of the OSPF routing protocol have been proposed in order to improve the network performance. In [23], Ye et al. study the optimization of OWPS Weights with queueing models and they choose packet loss as the metric to use. They improve the packet loss rate in 30-60% depending on the topology.

Other improvement of OSPF is proposed by O'Halloran and Chambers in [24], where they use the network load in order to provide an adaptation of OSPF interface metrics. They develop an algorithm and control mechanism to modify the OSPF interface cost on Mitrotik routers based on the amount of traffic. However, they find problems in knowing the destination of the traffic and the quantity of flows that leave an interface.

Nevertheless, SDN routing solutions have been focused on proposing an architecture where some traditional routing techniques can be applied or trying to change the metric to another specific factor, without taking into account the actual use of the network. There are no previous works in which the way of calculating the paths takes advantage of SDN capabilities and AI techniques. We use both, SDN and AI

techniques to propose a dynamical metric equation attending to the kind of traffic in the network. Through our system, we achieve greater routing adaptability in the network, optimizing multimedia transmissions.

3 Proposal

In this section, the proposal is detailed. First, the architecture is explained. Then, the proposed routing protocol is described. Later, the messages are shown. Finally, the algorithm and messages exchanging process are discussed.

3.1 Architecture

The architecture used is SDN-based. The routers are SDN nodes, i.e. OpenFlow-enabled devices. Besides, there is a central controller, whose aim is to manage the entire network, obtaining the current state of the network and making orders to the nodes. To this usual SDN architecture, an AI module is added. It is in constant communication process with the controller and it is used in order to get the cost of the different possible paths in routing decisions. This architecture is shown in Fig.1. On the left side, the difference between the controller and switches is shown, and their interaction using OpenFlow is displayed. On the right side, the network architecture shows the connection between the different actors in the SDN architecture.

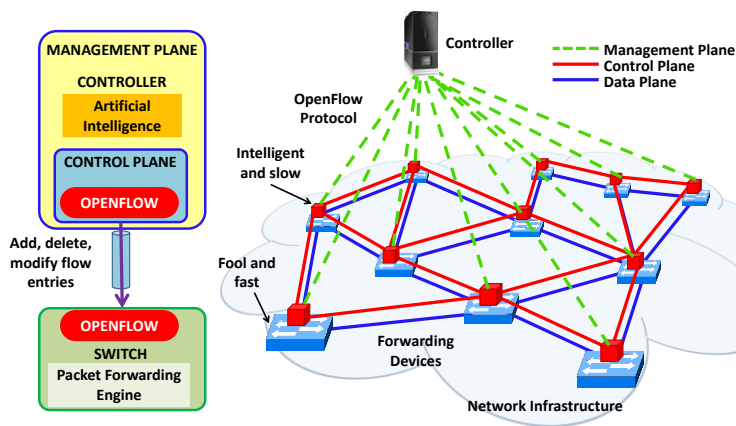
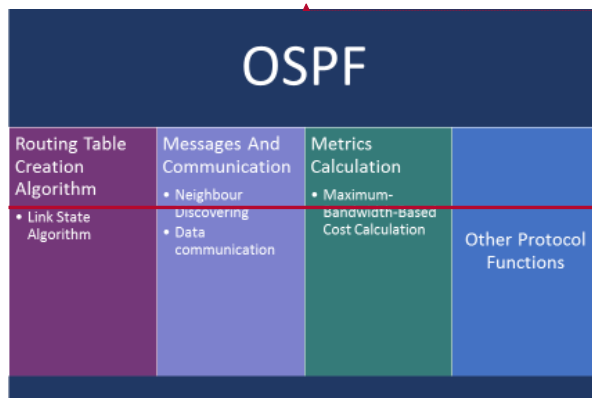


Fig. 1. Elements and network of the proposed architecture.

3.2 Routing Proposal

Routing algorithms can be depicted in several blocks attending to the different functions. Those blocks are shown in Fig. 2. The first module, painted in **Orangepurple**, is the one that builds the routing tables. OSPF is based on link state algorithm. The second one is the group of messages and communication process that provides the possibility to the protocol to work. The messages are exchanged between the routers. The **yellow-green** one contains the metric calculation. Depending on the protocol, the metric calculation can slightly vary. However, there is always a determined formula to calculate the cost of the links. Finally, the protocol has other determined process and functions to bring some new working ways. For instance, OSPF can manage administrative areas.



Con formato: Inglés (Estados Unidos)

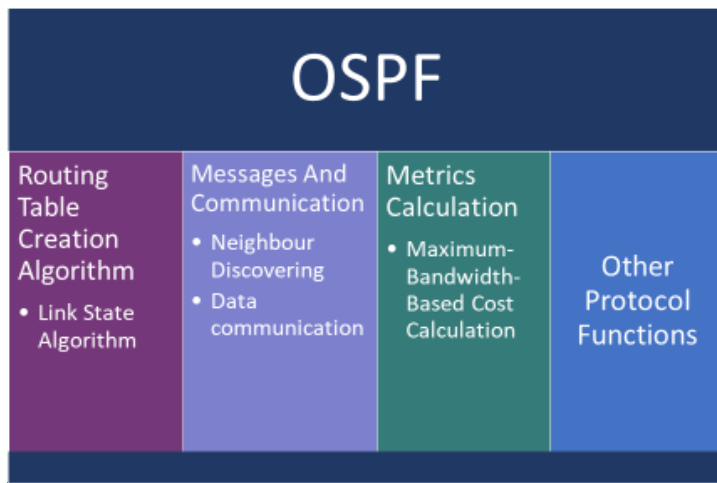


Fig. 2. Routing protocol structure.

~~However, there is always a determined formula to calculate the cost of the link. Finally, the protocol has other determined process and functions to bring some new working ways. For instance, OSPF can manage administrative areas.~~

Our proposal modifies two of those four modules, changing and expanding their function. The new structure is detailed in Fig.3. The figure is quite similar to the one explaining the structure of OSPF. The same four modules (Routing Table, Messages and Communication, Metrics Calculation and Other Protocol Functions) are defined. However, as can be observed, there are some of these modules that have been modified.

The routing table creation algorithm has been split into two different modules that offer the same function. The link state algorithm has not been modified, but a new module called "Link State Dynamic Checking" has been added to the route table creation. This module has been introduced thanks to the possibilities given by the SDN. The nodes are able to gather the statistics of the flows sent through the different paths and send them to the controller. So it allows the SDN routing protocols to make use of actualized link state data. It increases the possibility of adapting the routing protocols to the current state of the network and the changes produced in the network.

Con formato: Fuente: 8 pto

Con formato: Sangría: Primera línea: 0 cm



Fig. 3. New routing protocol structure

This possibility is quite important to choose the best path for all different kinds of traffic, but especially for multimedia traffic, due to the possibility to not only avoid links down or links state changes. In addition, it can be used to get the evolution of the links utilization and be able to apply decisions to ensure minimum QoS (and QoE) levels. These levels show the quality of the multimedia communications.

The other important change introduced by the proposal is the modification of the metric calculation. OSPF uses the formula shown in (1) to calculate the cost of each link.

$$\text{Cost} = \frac{10^8}{\text{bandwidth}(bps)} \quad (1)$$

OSPF uses the bandwidth of the links to calculate the cost. However, when the protocol is especially designed for multimedia traffic, there are some other variables that must be taken into account. Those variables are the QoS factors, which are closely related with the quality that the user perceives. In (2), the EIGRP metric formula is shown [21].

$$M = \left[K1 * BW + \frac{K2 * BW}{256 - \text{load}} + (K3 * \text{delay}) \right] * \frac{K5}{\text{reliability} + K4} \quad (2)$$

It uses several QoS factors like bandwidth, calculated as in (3), delay and loss rate. The coefficients K1-K5 are used to take into account the factors or not.

$$BW = \frac{10^7}{\text{bandwidth}(bps)} \quad (3)$$

Usually, the EIGRP metric is calculated as in (4), taking 1 as the value of the coefficients K1 and K3 and 0 for the others. Therefore, K1-K5 coefficients act as weights for each QoS factors. EIGRP uses only bandwidth and delay as QoS factors.

$$M = BW + \text{delay} \quad (4)$$

Our proposal also takes into account different QoS factors, calculating the cost as in (5).

$$\text{Cost} = \frac{K1 * 10^8}{Bw} + K2 * \text{delay} + K3 * \text{loss} \quad (5)$$

This equation has been designed to take into account the main QoS factors, due to the main role played by the multimedia traffic in the proposal.

The equation is composed by two main elements: a set of QoS factors (bandwidth in bps, delay in seconds and loss rate in percentage) and another one of constant values (K1, K2 and K3). The calculation of these two factors are quite different and is indicated in Fig.3, separated in two different modules.

On the one hand, we have the QoS factors. In the OSPF equation, the bandwidth is a constant, it is the maximum quantity of data that can be sent per second through a specific link when the route table is established. This value does not change unless a topology change happens. In the proposal, there are two changes from the OSPF metric calculation. First, the bandwidth, just as the other QoS factors, it is time-dependent. Its values change along the time. They are based on the statistics provided by the nodes and the controller. These values are dynamically changing, an important fact to maintain QoS and QoE levels in multimedia communications. The other difference is the meaning of the factors. Bandwidth, for being calculated in a dynamical way, is based on the current bandwidth, not the capability of the link. This allows applying techniques like load balancing to avoid a QoE falling, making a better management of the network resources. This is possible because of the use of SDN and the statistics gathering of the nodes in every flow that goes through each link

On the other hand, regarding ~~to~~ the coefficients, they are used to increase the weight of some factors depending on the main traffic in the network. Observing the formula, it seems that ~~the b~~ bandwidth can be the most important QoS factor because it affects in the greatest way to the cost. But it depends on the coefficients values. For example, if the network is being used for video streaming under demand, the importance of the available bandwidth is really greater than the delay. So, K1 should be greater than K2. Otherwise, if the main traffic in the network is multimedia streaming, the weight of the different factors should be similar. Finally, the network could be used for VoIP, where the delay is a crucial factor. In that case, K2 should be greater than K1. The fact of varying the method used to calculate the path cost can improve the performance of the network. This variation is produced by the AI module, in the controller. The AI module will set different values to the coefficients in order to get the best performance in terms of QoS. The SDN controller will manage the different kind of traffic associating them to some categories. This will allow the controller to label the flows and be aware of the main traffic in the network. There are two ways of working with these categories. On the one hand, the category related to the greatest quantity of flows can determine the values of the metric factors. On the other hand, for each flow the system can use the metric values related to the category it has been labeled to.

In conclusion, SDN brings us the possibility to modify the routing algorithm to get dynamically the ~~cost of each path~~ ~~paths~~, being able to change the metric formula depending on the state of the network. To achieve this, it is important to be able to use some structure like the flow table that associates the different flows in the network with the kind of traffic that is composed by. Moreover, it is necessary to be able to have in memory the factors of the different categories for avoiding resending messages when the category used in the network changes.

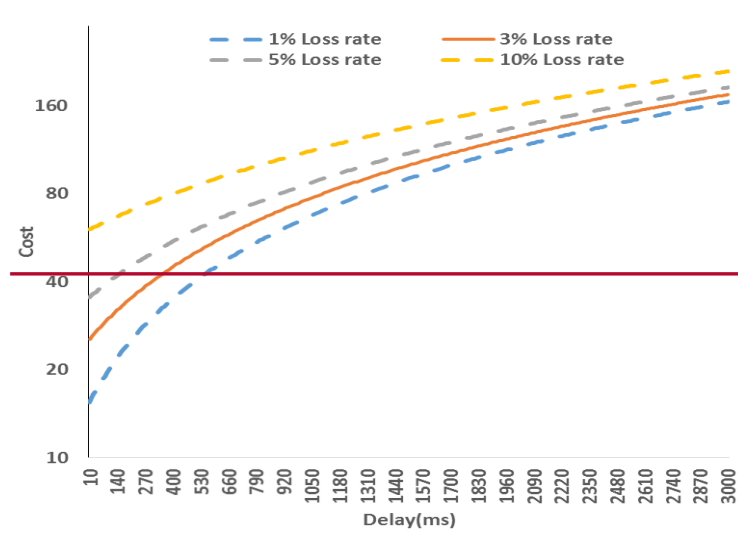
3.3 Metric analysis

Once the formula of the metric proposed has been discussed, ~~in this subsection~~, an analysis of this metric is described ~~in this subsection~~. First, the aim of the system is that the coefficients are determined by the AI module, during a learning process. However, the ~~design of the equation~~ ~~designing parts is a~~ ~~from result from~~ a specifically chosen balance ~~between its factors~~ that is described in this subsection.

In the first place, the available bandwidth must be placed as a denominator and the delay and loss rate as nominators. The use of the Dijkstra algorithm and the cost as a metric to be measured, implies that the greater the cost, the worse the path is. So, the bandwidth must reduce the cost of the path. Regarding the delay and the loss, the decision is that they affect to the cost in a similar way to the bandwidth in the initial point. Then, the AI module will change the coefficient values and will adapt the metrics to the different categories. Thus, an analytical method has been performed to determine an acceptable set of coefficient values. The values chosen for the coefficients are $K1=1$, $K2=0.5$ and $K3=5$. Fig. 4, 5 and 6 show the possible values of the cost for different bandwidth, loss rate and delay.

Fig. 4 shows the values of the metric when the links of the topology have a bandwidth of 10 Mbps. The different loss rate tested forms ~~s~~ different lines in the

graph. From the blue line, with 1% of loss rate, to the yellow one, with 10%, the cost obtained with the formula is displayed ~~for~~ from different delay values. The delay values increase from 10ms to 3000ms. With the minimum delay, the costs obtained are 15.5, 25.5, 35.5 and 60.5 for 1%, 3%, 5% and, 10% loss rate respectively. With 1s of delay, the cost values are 65, 75, 85, and 110. Finally, with 3s of delay the values are 165, 175, 185 and 210.



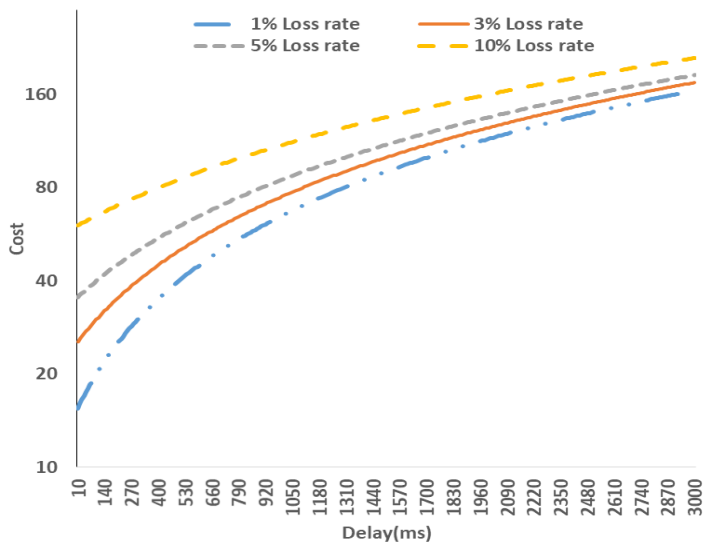


Fig. 4. Cost per delay and loss rate with a bandwidth of 10Mbps.

Fig. 5 shows the values obtained for a topology with 100 Mbps of available bandwidth. The increments of delay and loss rate are the same than in the previous graph. With the minimum delay, we have 15.5 of cost when there is 1% of loss rate. With 3% of loss rate, the cost is 16.5, 10 points more than with 1% of loss rate. Finally, with 5% and 10% of loss rate the cost values are 26.5 and 51.5 respectively. The increment of the bandwidth from 10 Mbps to 100 Mbps is balanced with a 2% increment of loss rate. With 1s of delay, cost values are 56, 66, 76 and 101. Finally, with 3s of delay, cost values are 156, 166, 176 and 201. And with 210ms, the cost with 1% of loss rate increases from 6.5 to 16.5. So, the increment of 200ms of delay is equivalent to the increment of 2% of loss rate.

Con formato: Centrado

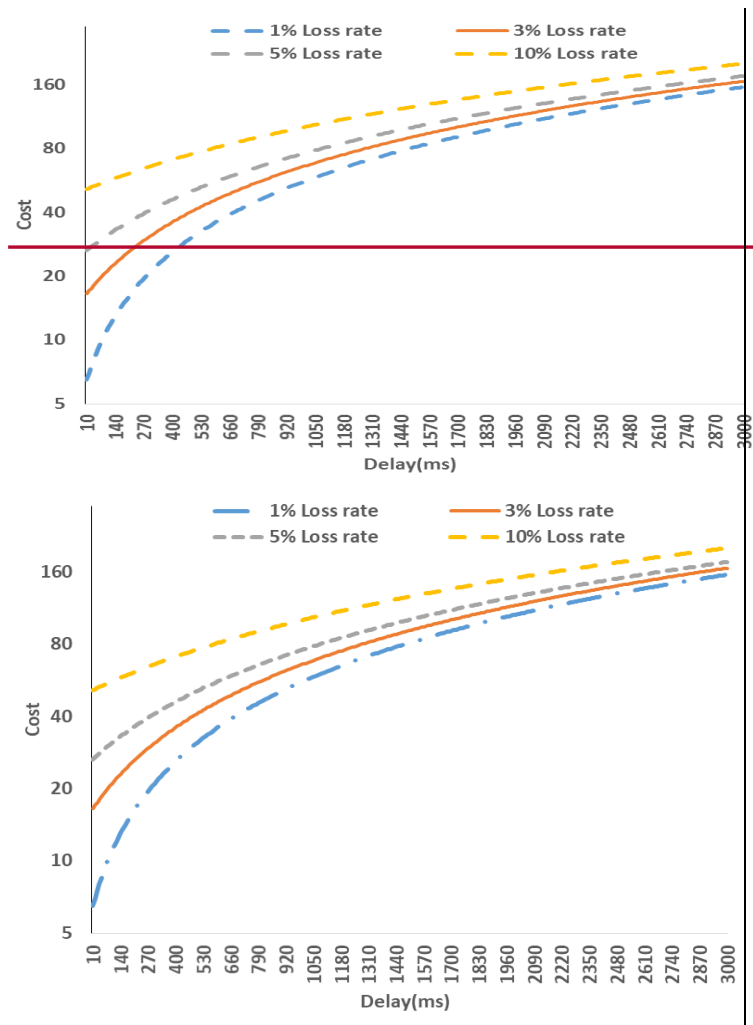


Fig. 5. Cost per delay and loss rate with a bandwidth of 100Mbps.

Finally, Fig. 6 displays the cost with 1000 Mbps. The minimum cost is reduced to 5.6 when there is 1% of loss rate. This is a reduction of only 0.9 points. This is caused when there are different paths to choose with high bandwidth. The differences between these bandwidths is not relevant, and the other QoS factor reduction must be

Con formato: Centrado

taken into account. These QoS factors, delay and loss rate must be the differential factor in these cases. The other cost values with minimum delay are 15.6, 25.6 and 50.6 with 3%, 5% and 10% respectively. With the minimum delay, the costs are 15.5, 25.5, 35.5 and 60.5 for 1%, 3%, 5% and 10% loss rate respectively. With 1s of delay, the cost values are 55.1, 65.1, 75.1 and 100.1. Finally, with 3s of delay the cost values are 155.1, 165.1, 175.1 and 200.1.

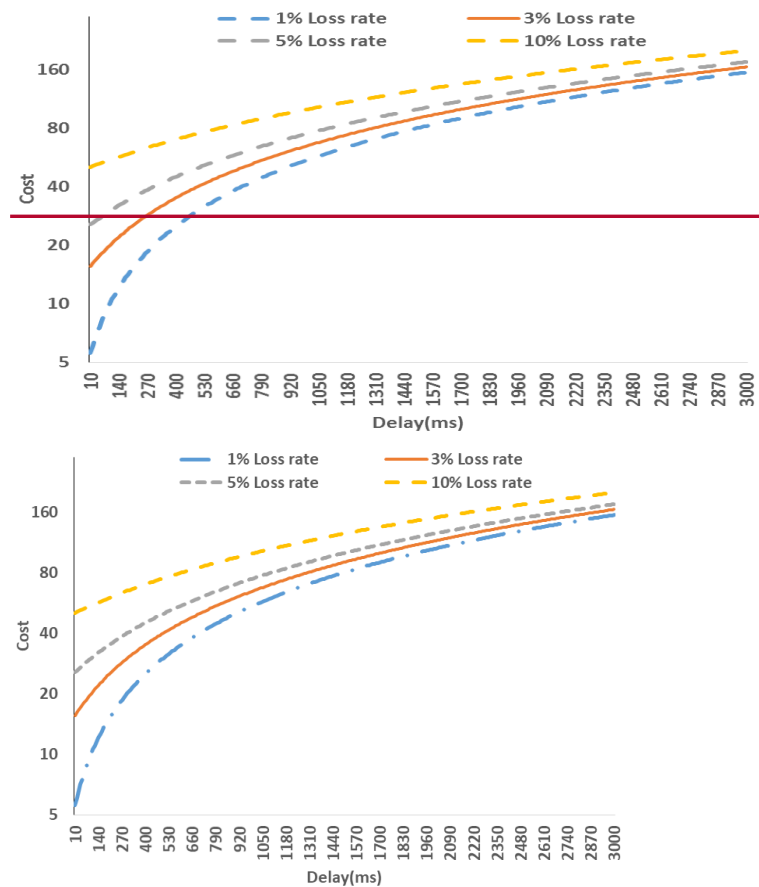


Fig. 6. Cost per delay and loss rate with a bandwidth of 1000Mb/s.

These graphs demonstrate that, initially, the coefficient factors are balanced. Therefore, the network can work properly and the AI module will adjust the

coefficients depending on the use of the network and changing between the different categories. That is possible thanks to the messages described in the next subsection.

3.4 Messages

In Fig. 7, the structure of the messages exchanged between the SDN controller and the nodes is described.

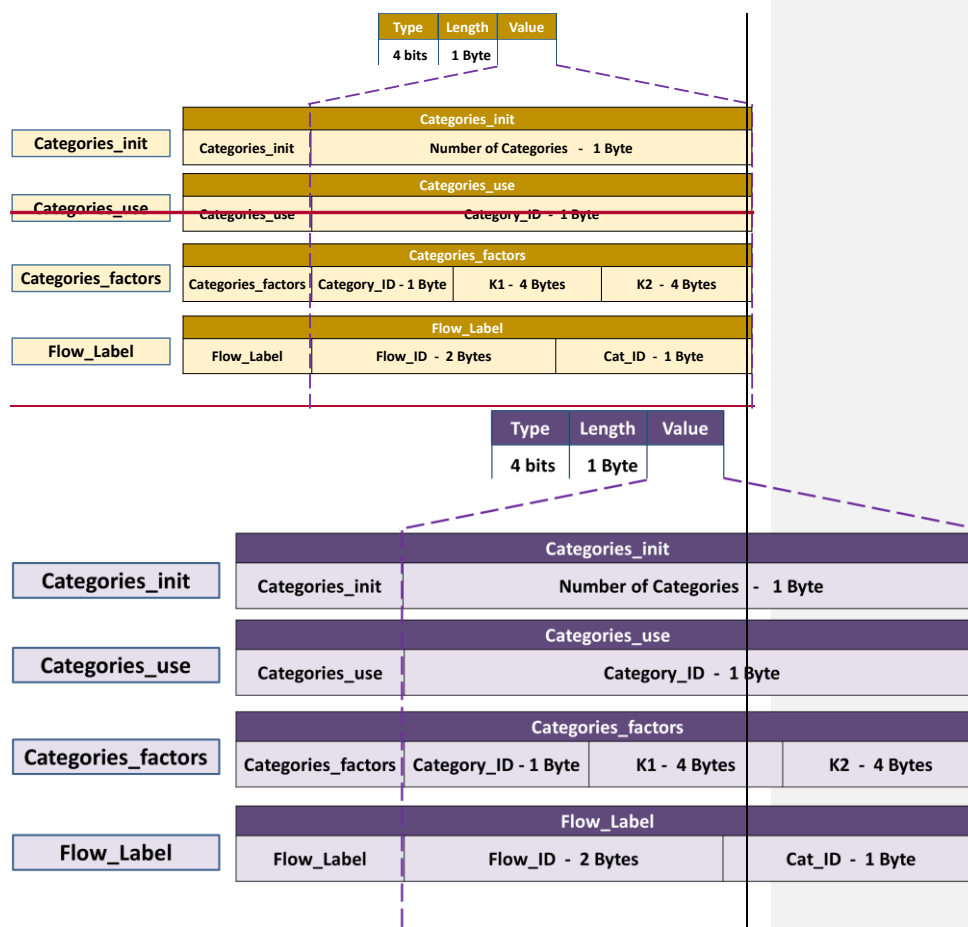


Fig. 7. Messages structure

The messages are detailed below:

- Categories_init: initialize the algorithm, indicating the number of traffic categories that is going to be used.
- Categories_use: notifies the category that is going to be used as a reference to calculate the cost of the path.
- Categories_factors: the factors used in a specific category. It is mainly used to actualize the values of the factors in a category. The category and the values of the three factors are indicated.
- Flow_label: used to assign a specific category to a flow. Useful to treat some flow in a specific way, which is optional and allows each kind of traffic flow be treated with a different metric, not with the category associated to the majority of the traffic. Due to this optional ability, the first byte of the Flow_ID field is zero if the flow will not be treated with its own category metric, which is the normal behavior. Otherwise, the first byte is set to ones to indicate that the flow must use the metric of its category, even the category used is another one.

Con formato: Párrafo de lista, Con viñetas + Nivel: 1 + Alineación: 1,04 cm + Sangría: 1,67 cm

3.5 Algorithm and Process

In this subsection, we describe the algorithm and the messages flow in order to manage the factors and the categories.

Algorithm 1 describes the category management process used by the controller.

First, the algorithm initializes the structures needed to manage and identify the main kind of traffic in the network. Then, a Categories_init message is sent to communicate the number of categories that is going to be used. For each category, a Categories_factors message is sent. This message includes the initial factors values that will be used when that category is being treated. Moreover, if the initial category is not the first one in being sent, a Categories_use message is sent from the controller. Later, the AI module is prepared. It will start to change all the factors of the initial category to find the most accurate balance between them. This change is done every specific period of time and attending to the performance statistics gathered by the controller using the OpenFlow messages.

The process is similar to the one presented in [6]. It is important that the controller informs the AI module which category is being used in the network. Thereby, the factor values calculation is a process associated which each category. In addition, the statistics are also passed to the AI module. The AI module returns the values that will be used to calculate the metrics for the routing protocol when the current category is used. Those values are indicated by a Categories_factors message.

Algorithm 1. Categories management

Given: Categories, Cat_initial

Initialize_Categories()

Send Cat_Init()

For each Category **in** Categories **do**

```

    Send Categories_factors()
End For
If Cat_initial != DEFAULT do
    Send Categories_use()
End If
AI_Initialize(Cat_initial)

Cat_Prev = Cat_initial
ForEach new iteration
    If Cat_Current != Cat_Prev do
        AI_Change_Category(Cat_Current)
    End If
    AI_Get_Factors()
    Send Categories_factors()
    Cat_Prev = Cat_Current
End ForEach

```

The message exchange process is detailed in the following lines and shown in Fig.8. First, the messages, which are sent when the algorithm is initialized, are described. A Categories_init message is always sent in the beginning. Then, for each category, the categories factors must be set by sending a Category_factors message. Those initial values have not been calculated in a learning process yet. They are only initial standard values of the factors. The last step in the process is optional. Usually, the first category is the category used when the traffic of the network is mixed and the factors have similar values. In order to set another category from the beginning, a Categories_use can be sent.

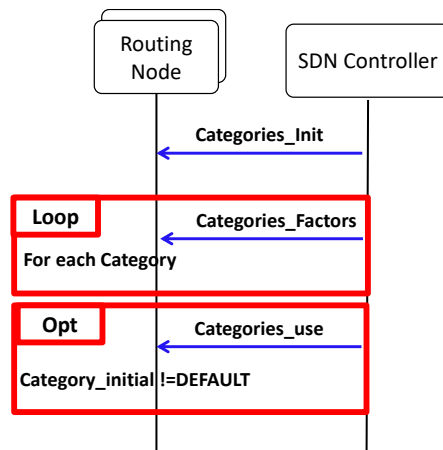


Fig. 8. Initial message exchange

The processes related to new flow and factors actualization scenarios are simpler than the initial process. When a new packet that is not classified is sent to a routing node, this node sends it to the controller by using the standard OpenFlow Packet_in message. Then, the controller analyzes the packet and classifies it into a specific category. It returns the packet and notifies that it has to be treated according to the routing protocol with the standard Packet_out message. The controller sends a Flow_label message to inform the node that the flow has to be classified as a specific category flow. This is useful if the optional way of working using different metrics simultaneously is desired. At that moment, the node can calculate the metric and send the packet from the adequate path. The controller can also send a Categories_use message if the majority of the traffic is now the one associated with a different category. This will depend on the current state of the network and the traffic that is flowing in that moment through the network.

Finally, the controller must communicate the values updated from the AI module to the nodes in order to allow them to work with the most updated version of the metric. For that, when there is an update of the values, the controller sends a Categories_factors message to the nodes. In that message, as it is explained in the previous section, the values used for one specific category are indicated. The controller will send one of those messages for each update. The controller could send a Flow_label message to change the behavior of some flow. This can be done to collect statistics and let the AI module learn.

Both processes described before are shown in Fig. 9a and 9b respectively. The messages exchanged between the controller and the routing nodes are shown. Moreover, the optional messages are also marked as "Opt".

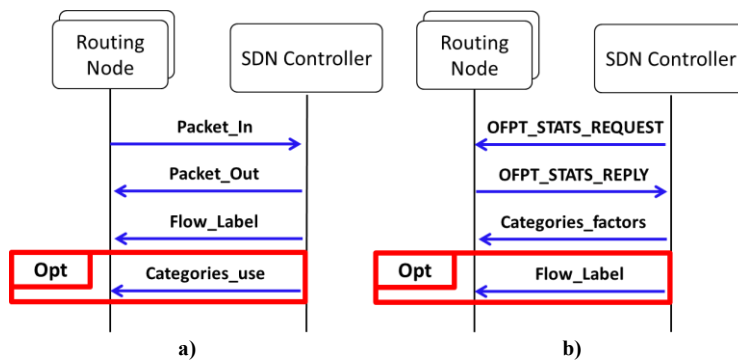
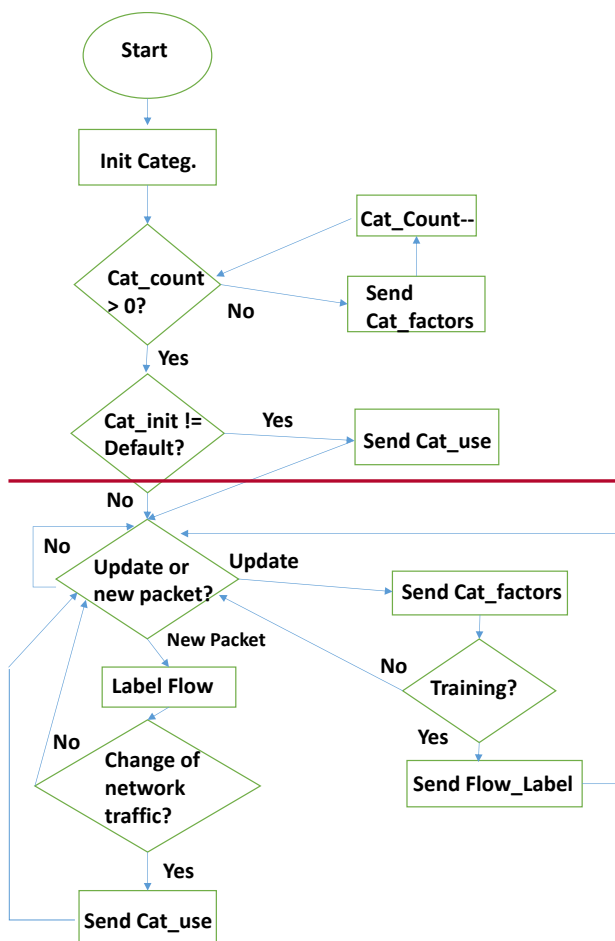


Fig. 9. Message exchange when a new packet arrives (a) and when the AI module updates the different factors used in the metric (b).

Furthermore, Fig. 10 shows a diagram as a graphical resume. In this diagram, the different processes and decisions are displayed. Each process described before can be identified in the diagram. The start of the algorithm is identified with the initialization

of the categories, and the loop to inform the nodes about the new categories and the optional use of Categories_use message. Then, the main loop is reached. The factor update process and the new packet process that are shown in Fig.9 can be identified as different branches of the main loop. The update provokes a new Categories_factors message to be sent and, optionally, a Flow_label message in order to continue learning. The new packet process consists of a Flow_label message and, if the main traffic of the network changes due to this new flow, a Categories_use packet is also sent to the routing nodes.



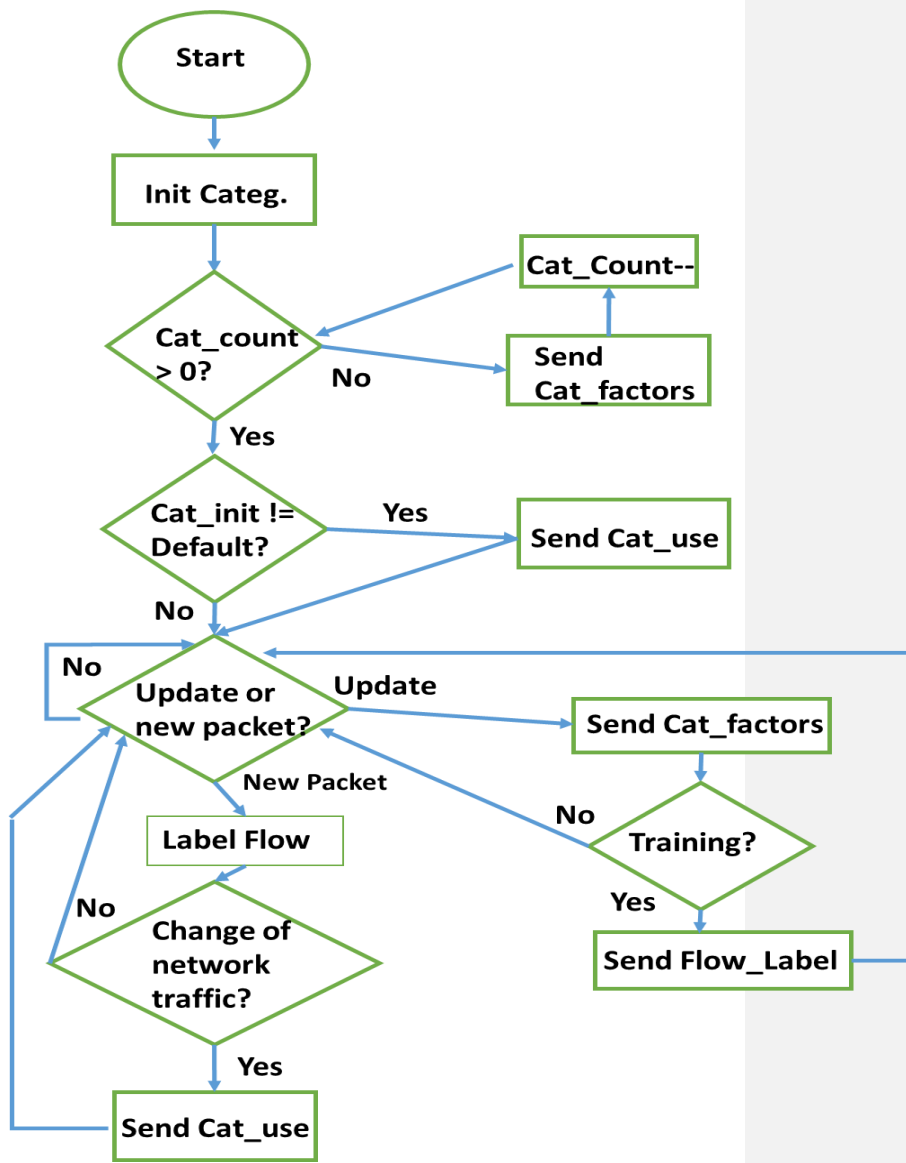


Fig. 10. Algorithm diagram mMessage exchange

4. Methodology and Results

In this section, the experiments done are described. First, the topology and the scenarios tested are shown. Then, the results are displayed.

4.1 Topology, Simulation Environment and Scenarios

The topology used in the experiments is shown in Fig. 11. Different scenarios provoke changes in the characteristics of the links, but all the scenarios can be implemented in the same topology. The links will change their characteristics according to the scenario, but the structure of the available paths will always be like the one depicted in Fig. 11.

In our topology, we have 6 switches, forming different subnetworks. Each subnetwork can be composed by several computers. However, in the simulation, each network is replaced by one single PC. The simulation has been done through Mininet. Mininet emulates both, the PCs and the switches, as Linux hosts. All the elements in the network are labeled and will be referenced in this text as they are in Fig. 11.

Regarding ~~to~~ the scenarios, five different situations ~~has~~^{ve} been chosen to test the proposal. In the first one, the network is used for multimedia streaming with enough resources to handle the transmissions. Then, the second scenario consists in handling several multimedia streaming flows that can consume too much network resources.

In the third scenario, the traffic in the network is not only composed by multimedia streaming flows, but also by VoIP traffic, being its main kind of traffic. The link with the greater bandwidth has also a great delay that can reduce the QoS of the VoIP traffic.

Scenarios 4 and 5 are designed to take into account the loss rate. In scenario 4, a TCP traffic of a file-downloading flow is sent in a network where the multimedia streaming flow is the main traffic. The link with ~~most-the greatest~~ bandwidth has enough loss rate to reduce the QoE of the multimedia traffic. The fifth scenario differs to the fourth in using the optional feature, along with the Flow-label message to manage in different ways the TCP traffic from the UDP multimedia streaming flows.

Table 1 summarizes the features of the different scenarios.

Table 1. Scenarios used for testing.

	<u>Scenario 1</u>	<u>Scenario 2</u>	<u>Scenario 3</u>	<u>Scenario 4</u>	<u>Scenario 5</u>
<u>Flow sent</u>	<u>Multimedia</u>	<u>Multimedia</u>	<u>Multimedia,</u> <u>VoIP</u>	<u>TCP,</u> <u>multimedia</u>	<u>TCP,</u> <u>multimedia</u>
<u>Network resources</u>	<u>Enough</u>	<u>Not enough</u>	<u>Enough</u> <u>bandwidth</u> <u>but high</u>	<u>Enough</u> <u>bandwidth</u> <u>but with</u>	<u>Enough</u> <u>bandwidth</u> <u>but with</u>

Tabla con formato

			<u>delay</u>	<u>losses</u>	<u>losses</u>
<u>Flow-label optional feature</u>	<u>No</u>	<u>No</u>	<u>No</u>	<u>No</u>	<u>Yes</u>

In each scenario, the performance of the multimedia flow is measured in terms of QoS by using Wireshark at the destination side. In both cases, OSPF and the proposal, the performances are tested and compared.

The results of each scenario are discussed in the next subsection.

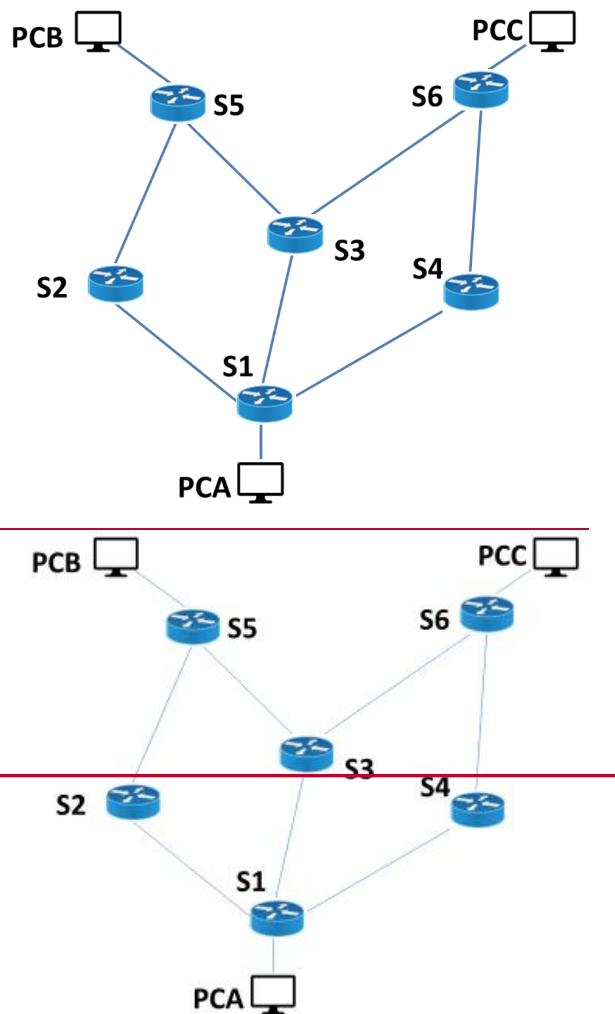


Fig. 11. Topology used in the experiments.

4.2 Results

In this subsection, the results obtained from each scenario are shown. Fig. 12 shows the bandwidth in bits per second obtained in Scenario 1, where the multimedia flows do not exceed the maximum bandwidth. Both flows consume similar bandwidth, with maximums of 3.419 Mbps and 3.266 Mbps and minimums of 6.2 kbps and 6.4 kbps for the OSPF test and for the proposal respectively. The average of the OSPF transmission is 1.070 Mbps and the average of the proposal is 1.015 Mbps.

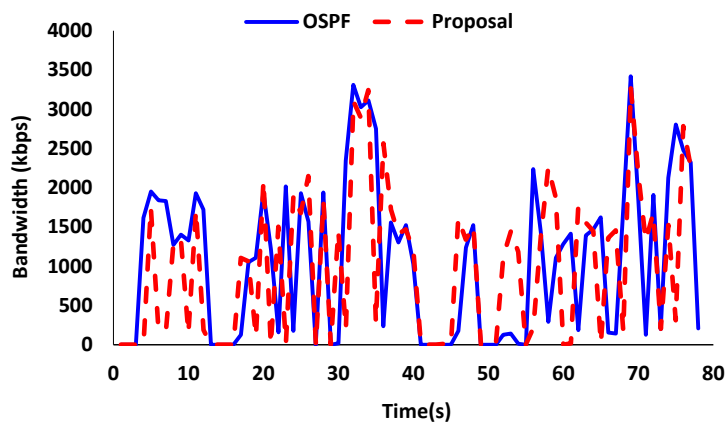


Fig. 12. Bandwidth in bits per second registered in scenario 1.

In Fig 13, the delay of each packet in ms with OSPF and the one produced with the proposal are shown. Both delays are quite similar, having maximums of 19.74ms and 19.85ms respectively and minimums of 0.04ms and 0.03ms. The average is also similar, 6.521ms in the OSPF transmission and 6.541ms by using the proposal.

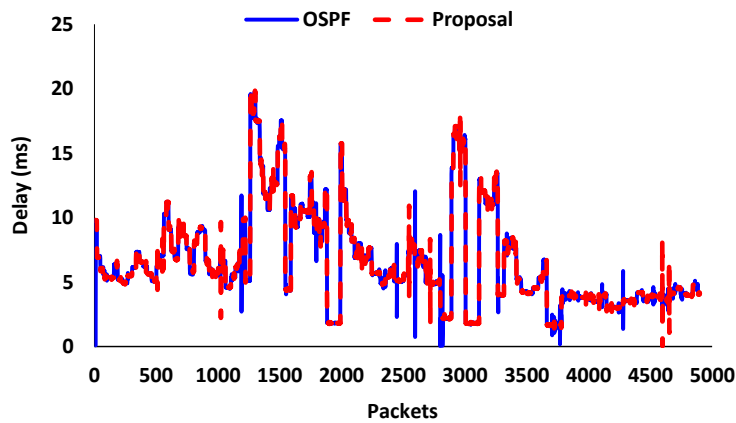


Fig.13. Delay in ms registered in scenario 1.

In terms of jitter, Fig. 14 shows that they are very similar too. The jitter of OSPF reaches 1.2-ms and the jitter of the proposal reaches the 0.86-ms. Their minimums are 0.01ms for both transmissions. The average jitter is 0.065ms for the OSPF transmission and 0.66ms for the proposal.

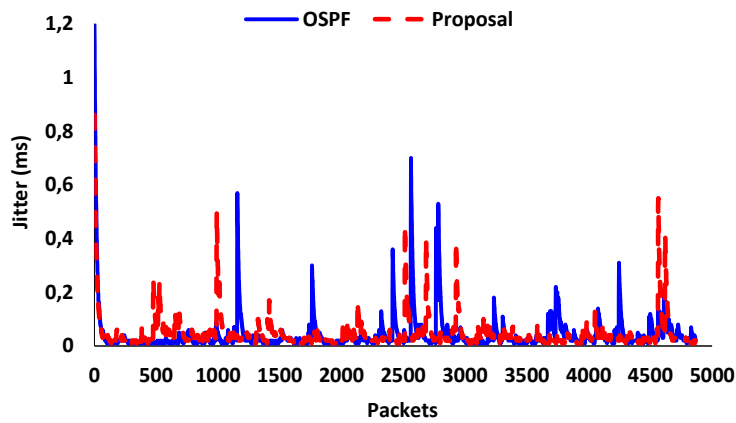


Fig. 14. Jitter in ms for each packet in scenario 1.

Regarding Scenario 2, where the multimedia flows exceed the capabilities of the link, Fig. 15 displays the delay and Fig. 16 the jitter. We have observed that the delay

is similar. The paths, where the traffic is sent through, present similar latencies. In both transmissions, the maximum delay is 19.75ms. However, the minimum delay obtained is different. The OSPF transmission presents a minimum delay of 1.8ms and the proposal obtains 0.02ms of minimum delay. The average delay with OSPF is greater than the one obtained in the transmission using the proposal. OSPF transmission shows a 6.984ms of average delay, while the proposal achieves 6.5ms.

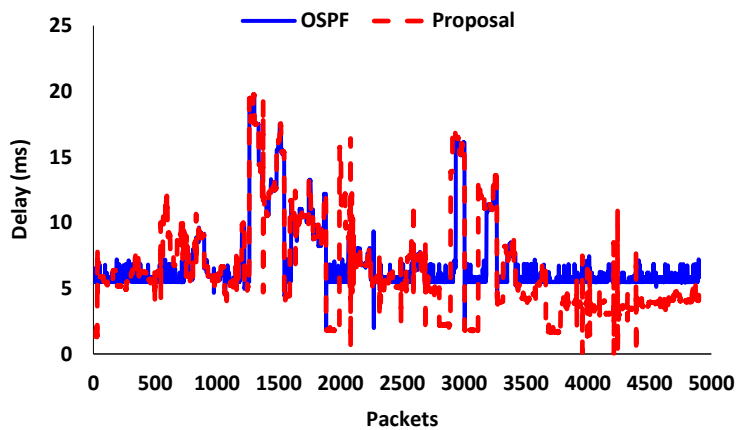


Fig. 15. Delay in ms obtained in scenario 2 for each packet.

As it is shown in Fig. 16, the jitter produced during the transmission is higher in the OSPF case, presenting fluctuations and increasing up to 9.48ms of maximum, while the proposal presents a maximum of 1.63ms, reducing the maximum jitter in 83%. The minimum jitter is 0.01ms in both transmissions, but the greatest difference is presented in the average jitter. The average jitter of the OSPF transmission is 1.396ms while the average of the proposal is 0.082ms. There is a jitter reduction of 93%.

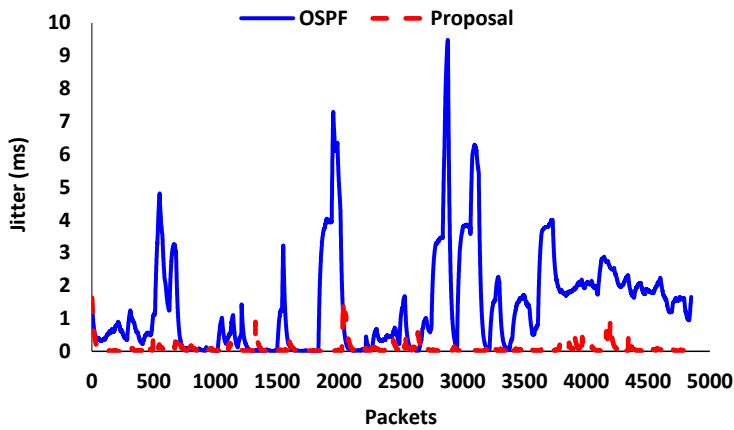


Fig. 16. Jitter in ms for each packet in scenario 2.

Finally, Fig. 17 shows the bandwidth consumed in each case. With the path selected by OSPF, the bandwidth is not able to be greater than 1.961 Mbps, which is the available bandwidth during the transmission. However, by selecting the path according to the available bandwidth, the proposal reaches 3.244 Mbps, avoiding QoE problems and increasing the throughput in 40%. The minimum throughput in the OSPF transmission is 6.4 kbps and with the proposal is 6 kbps. Moreover, the average bandwidth is 1.181 Mbps for OSPF and 1.2 Mbps for the proposal.

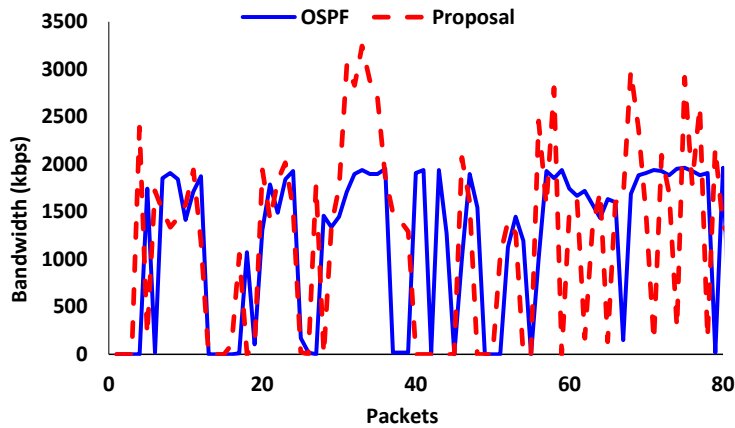


Fig. 17. Bandwidth in bits per second in scenario 2.

In Scenario 3, there is a difference in terms of delay between the path of the highest bandwidth and the one chosen by the proposal for VoIP transmission. Fig. 18 presents its results in terms of delay. OSPF transmission average delay is 558.33ms, while the proposal transmission suffers an average delay of 58.29ms. The maximum delay of the OSPF transmission is 612.55ms and the maximum for our proposal is 108.52ms. The minimum delays are 500ms for OSPF and 0.09ms for the proposal.

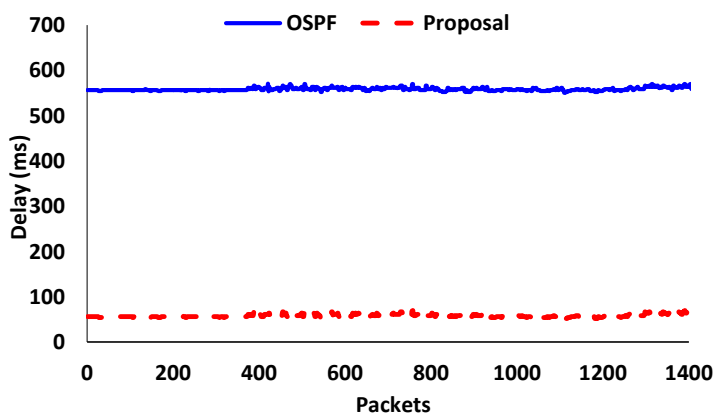


Fig. 18. Delay in ms obtained in scenario 3 for each packet.

The jitter results are presented in Fig. 19. They do not differ too much. Both transmission have a minimum jitter of 0.01ms. The average jitter is also similar with 0.0412ms for the OSPF transmission and 0.0427ms for the proposal. Maximum values are 1.13ms for the OSPF and 0.55ms for the proposal.

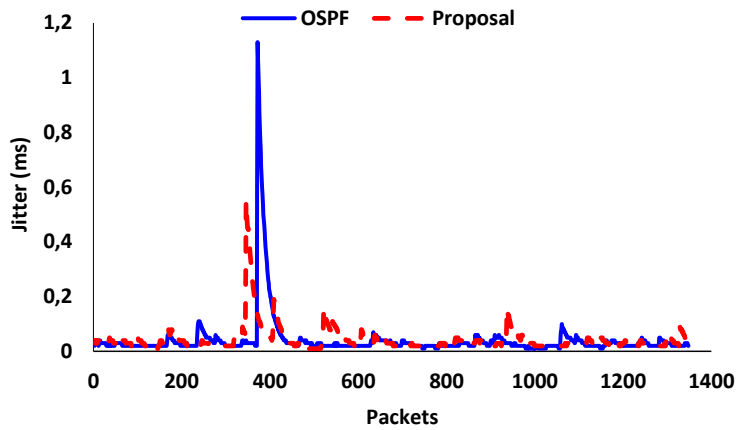


Fig. 19. Jitter in ms for each packet in scenario 3.

In Scenario 4, the path with the greatest bandwidth also presents a high loss rate. Fig. 20 shows the delay obtained from both transmissions. OSPF transmission presents a greater average delay of 8.05ms, while the proposal presents a reduction, obtaining 7.08ms of average delay. The maximum delay is also increased to 59.58ms with OSPF, having a maximum of 52.5ms with the proposal. Nevertheless, the minimum delay is 1.25ms in the OSPF case and 1.83ms with the proposal routing solution.

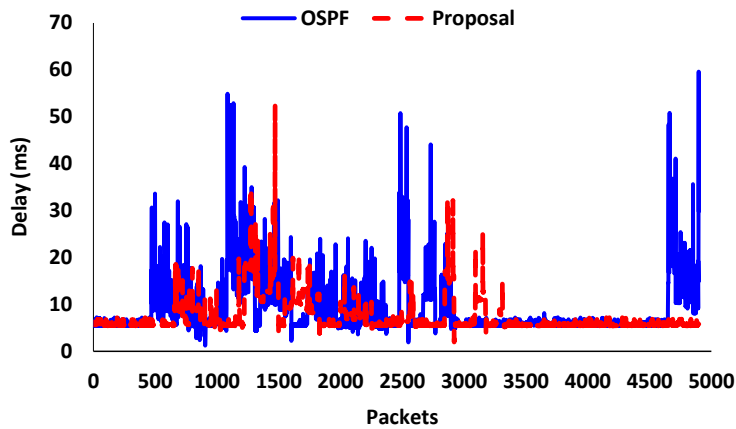


Fig. 20. Delay in ms obtained in scenario 4 for each packet.

Fig. 21 shows the jitter. There is a minimum of 0.01ms in both cases. The differences are presented in maximum and average jitter. OSPF presents an average of 1.51ms, while the proposal achieves 1.4ms. The maximum jitter presented in the transmission using OSPF as a routing protocol is 10.92ms, more than 1ms greater than the maximum with the proposal, which presents 9.67ms.

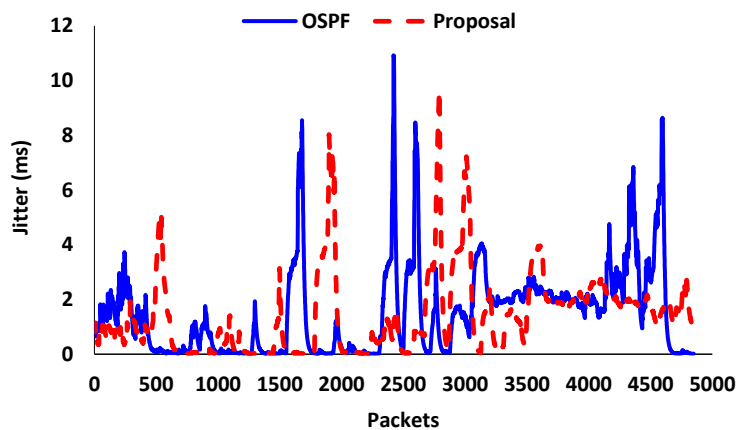


Fig. 21. Jitter in ms for each packet in scenario 4.

Finally, Fig. 22 displays the loss rate of each transmission. The characteristics of the links and the routing decision of OSPF through the maximum bandwidth path causes an increment of loss rate in OSPF transmission. 15% of the multimedia packets are lost during the transmission. With the proposal, the packets are sent through a path with less loss rate, achieving 3% of loss rate.

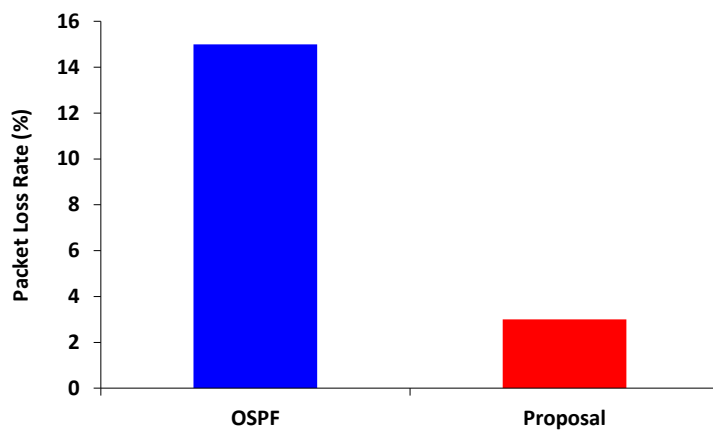


Fig. 22. Loss rate presented with OSPF and the proposal in scenario 4.

Scenario 5 shows the capability of making different decisions depending on which kind of traffic is being sent. As it is explained in Section 3, this functionality is not the central point of the proposal, but it can be applied. This fifth scenario tests the load balancing done when the metric is changed in every kind of flow. Fig. 23 shows that this difference in the metric calculation allows the streaming to exceed the 2 Mbps, having a maximum of 3.27 Mbps of throughput. Without taking individual decisions, the maximum throughput is 1.95 Mbps. Minimum throughput is also increased. Without this characteristic, the minimum throughput is 0.44 kbps and when it is used, it raises to 6.58 kbps. Average bandwidth also increases from 1.17 Mbps to 1.22 Mbps.

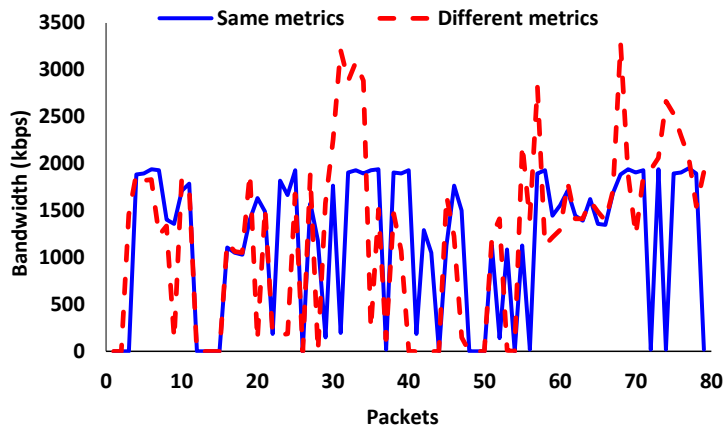


Fig. 23. Bandwidth consumed for each transmission, by using the same metrics for each flow and by using different metric for the different flows.

In addition, Fig. 24 shows a jitter reduction due to the possibility of sending at the maximum bitrate, without limitations. The average jitter is reduced from 1.4ms to 0.1ms, more than 90% of reduction. The maximum jitter is also reduced from 9.67ms to 1.31ms, presenting a reduction of 86%. The minimum jitter is 0.01 in the both scenarios.

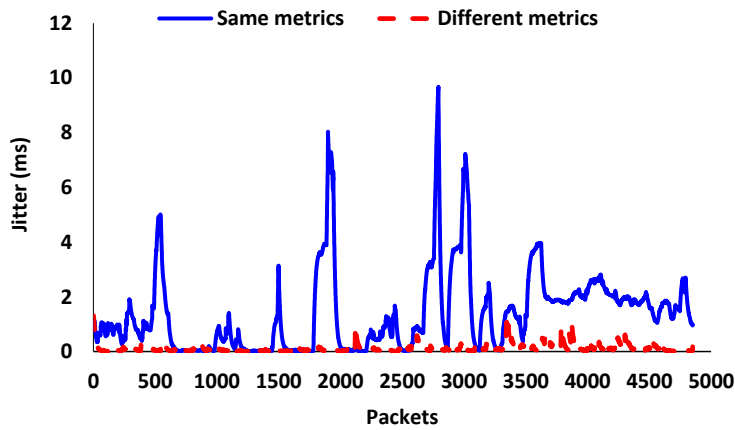


Fig. 24. Jitter obtained in scenario 5.

Table 2 summarizes the results obtained in the tests. For each scenario, the different values of jitter, bandwidth, delay and loss rate are detailed. The first value is for OSPF and the second one for our proposal in all the scenarios except in the last one. In that scenario, the first value is the one obtained without using the optional feature and the second one is the gathered when that feature is activated.

Table 2. Results of each scenario using OSPF and our proposal.

	Scenario 1	Scenario 2	Scenario 3	Scenario 4	Scenario 5
<u>Minimum Jitter</u>	0.01/0.01ms	0.01/0.01ms	0.01/0.01ms	0.01/0.01ms	0.01/0.01ms
<u>Average Jitter</u>	0.065/0.066ms	1.396/0.082ms	0.0412/0.0427ms	1.51/1.4ms	1.4/0.10ms
<u>Maximum Jitter</u>	1.2/0.86ms	9.48/1.63ms	1.13/0.55ms	10.92/9.67ms	9.67/1.31ms
<u>Minimum bandwidth</u>	6.2/6.4kbps	6.4/6kbps	=	=	0.44/6.58kbps
<u>Average bandwidth</u>	1.070/1.015Mbps	1.181/1.2Mbps	=	=	1.17/1.22Mbps
<u>Maximum bandwidth</u>	3.419/3.266Mbps	1.961/3.244Mbps	=	=	1.95/3.27Mbps
<u>Minimum delay</u>	0.04/0.03ms	1.8/0.02ms	500/0.09ms	1.25/1.83ms	=
<u>Average delay</u>	6.521/6.541ms	6.984/6.5ms	558.33/58.92ms	8.05/7.08ms	=
<u>Maximum delay</u>	19.74/19.85ms	19.75/19.75ms	612.55/108.52ms	59.58/52.5ms	=
<u>Loss rate</u>	=	=	=	15/3(%)	=

Tabla con formato

Con formato: Fuente: Negrita

Con formato: Fuente: Negrita

Con formato: Fuente: Negrita

Con formato: Fuente: Negrita

Con formato: Fuente: Negrita

Con formato: Fuente: Negrita

Tabla con formato

Con formato: Fuente: Negrita

Con formato: Fuente: Negrita

Tabla con formato

Con formato: Fuente: Negrita

Con formato: Fuente: Negrita

Con formato: Fuente: Negrita

4.3 Comparison with other OSPF variations

In this section, the proposal is compared with other OSPF variations, concretely the ones presented in Section 2. This comparison is displayed in Table 4.3. On the one hand, in [23], Ye et al. proposed a metric based on some queueing models focused on packet loss. They improved the packet loss rate in 30-60% depending on the topology. Their proposal reduces the loss rate and tries to get the best value of the metric factor within a limited time frame, but they cannot take into account several factors and adapt their weights depending on the use of the network. We can do the same, but achieving a completely adaptive routing solution.

On the other hand, O'Halloran and Chambers use the network load in order to provide an adaptation of OSPF interface metrics in [24]. They try to dynamically modify the OSPF interface cost of the routers. However, technical problems impede

them to achieve the expected performance. With the use of SDN, we can solve their problems, being aware of the number of flows and the destination of each flow.

Table 4.3. Comparison of OSPF modifications.

	Performance Improvement	Adaptive metric	AI metric definition	Technology used	Multiple factor metric
<i>Ye [23]</i>	No Data	External Updates	No	External program with technical problems	No
<i>Chambers [24]</i>	30-60% loss rate	Yes	No	Theoretical queuing model	No
<i>Our Proposal</i>	80% loss rate Up to 90% Jitter Reduction Increment of 40% of Throughput	Yes	Yes	SDN	Yes

5. Conclusion and Future Work

Routing decisions has been traditionally taken depending on factors related to capabilities of the paths or the distance between source and destination. However, these solutions did not take into account the actual use of the network. Each kind of network in each interval of time can need some different resources. Moreover, the importance of these resources also changes through the use of the network. SDN adds adaptability to the network and can be used to propose new kinds of applications and routing solutions. In this paper, we have proposed a dynamical metric solution in order to provide the network the ability to choose the path depending on the main use of the network.

Results show that, when the use of the network does not need more resources in terms of available bandwidth, OSPF and the proposal achieves similar results (Fig. 12 and 13 and 14). However, being able to adapt the metric factors values depending on the use of the network can make use of higher bandwidth when several multimedia streaming flows are present in the network. In the experiments, OSPF chooses a path when the transmission can only use up to 2 Mbps. Nevertheless, our proposal uses load balancing that allows the multimedia streaming to use all the required bandwidth, with an increment of 40% of the throughput. In addition, there is also a jitter reduction of 93%. The third scenario demonstrates that changing the importance of every factor according to the main kind of traffic can improve the quality of the transmission. The proposal changes its metric to prioritize delay reduction. Fig. 18 shows a reduction of more than 400ms. Finally, loss rate can be also taken into account. Moreover, and in

TCP and multimedia streaming flows, using a different metric for each kind of traffic, has achieved ~~in Scenario 5~~ a 90% reduction of the average Jitter ~~in Scenario 5~~. ~~In addition, and it also allows~~ the possibility of using more than 3Mb/s of bandwidth in the multimedia streaming, using a path with an 80% less of loss rate.

The comparison with the other related works show that our proposal is the only one that introduces AI and SDN to modify OSPF protocol. With these modifications, we achieve real time improvements and adaptive routing, which is currently needed to provide realtime services [256][276].

As future work, there are several lines to work in. We will work on dynamical metric routing applications development. For example, building an application on the controller where the user can select manually the kind of traffic in the network and adjust the metric. Moreover, a further investigation will be done in the AI module. We will try to select the best AI technique to learn the optimal metric for each case. In addition, ~~we will apply~~ the proposal to different kinds of networks and applications. ~~Therefore, we will~~ develop a most efficient system and AI module. Characteristics of IoT networks or WSN can provide further info to improve the proposal.

Acknowledgments

This work has been partially supported by the “Ministerio de Educación, Cultura y Deporte”, through the “Ayudas para contratos predoctorales de Formación del Profesorado Universitario FPU (Convocatoria 2015)”. Grant number FPU15/06837, by the “Ministerio de Economía y Competitividad”, through the “Convocatoria 2014. Proyectos I+D - Programa Estatal de Investigación Científica y Técnica de Excelencia” in the “Subprograma Estatal de Generación de Conocimiento”, project TIN2014-57991-C3-1-P, through the “Convocatoria 2016 - Proyectos I+D+I - Programa Estatal De Investigación, Desarrollo e Innovación Orientada a los retos de la sociedad” (Project TEC2016-76795-C6-4-R) and through the “Convocatoria 2017 - Proyectos I+D+I - Programa Estatal de Investigación, Desarrollo e Innovación, convocatoria excelencia” (Project TIN2017-84802-C2-1-P).

Con formato: Inglés (Estados Unidos)

References

- [1] Coltun R., Ferguson D., Moy J., OSPF for IPv6, RFC 5340, DOI: 10.17487/RFC5340, July 2008, <https://rfc-editor.org/rfc/rfc5340.txt>
- [2] Software-Defined Networking (SDN) Definition. Available at: <https://www.opennetworking.org/sdn-definition/> (last accessed on December 15, 2017)
- [3] Jimenez J.M., Romero O., Rego A., Dilendra A., Lloret J., Study of multimedia delivery over software defined networks, Network Protocols and Algorithms, Vol. 7, Issue 4, 2015, pp 37-62, <https://doi.org/10.5296/npa.v7i4.8794>
- [4] Egea S., Rego A., Carro B., Sanchez-Esguevillas A., Lloret J., Intelligent IoT Traffic Classification Using Novel Search Strategy for Fast Based-Correlation Feature Selection in Industrial Environments, IEEE Internet of Things Journal. 2017, <https://doi.org/10.1109/JIOT.2017.2787959>.

- [5] Rego A., Sendra S., Jimenez J. M., Lloret J., OSPF routing protocol performance in Software Defined Networks, Fourth International Conference on Software Defined Systems (SDS 2017), 8-11 May 2017, Valencia, Spain, <https://doi.org/10.1109/SDS.2017.7939153>
- [6] Sendra S., Fernández P. A., Quilez M. A., Lloret J., Study and Performance of Interior Gateway IP routing Protocols, Network Protocols and Algorithms, Vol 2, No 4 (2010), <https://doi.org/10.5296/npa.v2i4.547>
- [7] Rakheja P., Kaour P., Gupta A., Sharma A., Performance Analysis of RIP, OSPF, IGRP and EIGRP Routing Protocols in a Network, International Journal of Computer Applications, vol. 48, issue 18, pp. 6-11, <https://doi.org/10.5120/7446-0401>
- [8] Sendra S., Rego A., Lloret J., Jimenez J. M., Romero O., Including artificial intelligence in a routing protocol using Software Defined Networks, IEEE International Conference on Communications Workshops (ICC Workshops 2017), 21-25 May 2017, Paris, France. 2017, <https://doi.org/10.1109/ICCW.2017.7962735>
- [9] Barbancho J., León C., Molina J., Barbancho A., SIR: A New Wireless Sensor Network Routing Protocol Based on Artificial Intelligence, In Advanced Web and Network Technologies, and Applications. APWeb 2006. Lecture Notes in Computer Science (LNCS), Vol 3842, pp 271-275, https://doi.org/10.1007/11610496_35
- [10] Barbancho J., León C., Molina F. J., Barbancho A., Using Artificial Intelligence in Wireless Sensor Routing Protocols. In Knowledge-Based Intelligent Information and Engineering Systems. (KES 2006). Lecture Notes in Computer Science, Vol. 4251. Springer, pp 475-482, https://doi.org/10.1007/11892960_58
- [11] Arabshahi P., Gary A., Kassabalidis I., Das A., Narayanan S., Sharkawi M. El, Marks I., R. J. (2001). Adaptive Routing in Wireless Communication Networks using Swarm Intelligence. AIAA 19th Annual Satellite Communications System Conference, Toulouse, France, April 17, 2001.
- [12] Gunes M., Sorges U., Bouazizi I., [ARA-the ant-colony based routing algorithm for MANETs](#), International Conference on Parallel Processing Workshops, Vancouver, BC, Canada, 21-21 Aug. 2002, <https://doi.org/10.1109/ICPPW.2002.1039715>
- [13] Ducatelle F., Di Caro G. A., Gambardella L. M., Principles and applications of swarm intelligence for adaptive routing in telecommunications networks, Swarm Intelligence, September 2010, Vol. 4, Issue 3, pp 173-198, <https://doi.org/10.1007/s11721-010-0040-x>
- [14] Rajagopalan S., Shen C., ANSI: A swarm intelligence-based unicast routing protocol for hybrid ad hoc networks, Journal of Systems Architecture, Vol. 52, Issues 8-9, August-September 2006, pp 485-504, <https://doi.org/10.1016/j.sysarc.2006.02.006>
- [15] ~~Perkins C., Belding Royer E., Das S., RFC 3561 Ad hoc On-Demand Distance Vector (AODV) Routing, RFC 3561, DOI: 10.17487/RFC3561, July 2003, Available online: <https://www.rfc-editor.org/info/rfc3561> (accessed on 08 05 2018).~~
- [16] Zungeru A. M., Ang L., Seng K. P., Classical and swarm intelligence based routing protocols for wireless sensor networks: A survey and comparison, Journal of Network and Computer Applications, Vol. 35, Issue 5, September 2012, Pages 1508-1536, <https://doi.org/10.1016/j.jnca.2012.03.004>
- [17] Karaboga D., Okdem S., Ozturk C., Cluster based wireless sensor network routing using artificial bee colony algorithm, Wireless Networks, October 2012, Volume 18, Issue 7, pp 847-860, <https://doi.org/10.1007/s11276-012-0438-z>
- [18] Ginsberg L., Litkowski S., Previdi S., IS-IS Route Preference for Extended IP and IPv6 Reachability, RFC 7775, DOI: 10.17487/RFC7775, February 2016, <https://www.rfc-editor.org/rfc/rfc7775.txt>
- [19] Rekhter Y., Li T., Hares S., A Border Gateway Protocol 4 (BGP-4), RFC 4271, DOI: 10.17487/RFC4271, January 2006, <https://rfc-editor.org/rfc/rfc4271.txt>
- [20] Caria M., Das T., Jukan A., Divide and conquer: Partitioning OSPF networks with SDN, IFIP/IEEE International Symposium on Integrated Network Management (IM 2015), 11-15 May, Ottawa (ON), Canada, 2015, <https://doi.org/10.1109/INM.2015.7140324>
- [21] Rothenberg C. E., Nascimento M. R., Salvador M. R., Corrêa C. N. A., Cunha de Lucena S., Raszuk R., Revisiting routing control platforms with the eyes and muscles of software-defined

Con formato: Inglés (Estados Unidos)

- networking, HotSDN '12 Proceedings of the first workshop on Hot topics in software defined networks, August 13 - 17 (2012), Helsinki (Finland), pp 13-18, <https://doi.org/10.1145/2342441.2342445>
- [22] Zhu M., Cao J., Pang D., He Z., Xu M., SDN-Based Routing for Efficient Message Propagation in VANET, In: Wireless Algorithms, Systems, and Applications (WASA 2015), Lecture Notes in Computer Science, Vol. 9204, pp 788-797, https://doi.org/10.1007/978-3-319-21837-3_77
- [2423] Ye, T., ~~ao & Tahilramani Kaur, Hema~~Hema T. K., & Kalyanaraman, S., ~~hivkumar & Vastola, K. S., anneth & Yadav, S. proi.~~ Minimizing packet loss by optimizing OSPF weights using online simulation. ~~Dynamic Optimization of OSPF Weights using Online Simulation. Modeling, Analysis and Simulation of Computer Telecommunications Systems, 2003. MASCOTS 2003. 11th IEEE/ACM International Symposium on In Proceedings of International Workshop on Quality of Service (IWQoS), Orlando, FL, USA, 27 October 2003, 2002~~ <https://doi.org/10.1109/MASCOT.2003.1240645>.
- [2524] Cian O'Halloran. Dynamic Adaptation of OSPF Interface Metrics based on Network Load. 26th Irish Signals and Systems Conference (ISSC), Ireland. Jun 2015. <https://doi.org/10.1109/ISSC.2015.7163767>
- [265] Mehmet ŞİMSEK, Nurettin Doğan, Muhammet Ali Akcayol. A New Packet Scheduling Algorithm for Real-Time Multimedia Streaming. Network Protocols and Algorithms, Vol 9, No 1-2 (2017). Pp. 28-47. <https://doi.org/10.5296/npa.v9i1-2.12410>
- [276] Ramon Sanchez-Iborra, Maria-Dolores Cano, Joan Garcia-Haro. Revisiting VoIP QoE assessment methods: are they suitable for VoLTE?. Network Protocols and Algorithms, Vol 8, No 2 (2016). Pp. 39-57. <https://doi.org/10.5296/npa.v8i2.9123>

Con formato: Español (España)

Con formato: Español (España)

Con formato: Español (España)

Con formato: Español (España)