



# A novel multicast traffic engineering technique in SDN using TLBO algorithm

Reza Mohammadi<sup>1</sup>  · Reza Javidan<sup>1</sup> · Manijeh Keshtgari<sup>1</sup> · Reza Akbari<sup>1</sup>

Published online: 4 December 2017  
© Springer Science+Business Media, LLC, part of Springer Nature 2017

## Abstract

In recent years, multicast communication is widely used by network providers to deliver multimedia contents. Quality of service (QoS) provisioning is one of the most important issues while transmitting multimedia contents using multicast. Traditional IP multicasting techniques suffer from reliability, scalability and have limitations to provide appropriate QoS for multimedia applications based on service level agreement (SLA). Nowadays, the advent of software defined networking (SDN), enables network providers to manage their networks dynamically and guarantee QoS parameters for customers based on SLA. SDN provides capabilities to monitor network resources and allows to dynamically configure desired multicasting policies. In this paper, we proposed a novel multicasting technique to guarantee QoS for multimedia applications over SDN. To deliver multimedia contents in an efficient manner, our proposed method models multicast routing as a delay constraint least cost (DCLC) problem. As DCLC problem is NP-Complete, we proposed an approximation algorithm using teaching–learning-based optimization to solve this problem. We evaluated our proposed method under different topologies. Experimental results confirmed that our proposed method outperforms IP multicast routing protocol, and it achieves a gain of about 25% for peak signal-to-noise ratio.

**Keywords** SDN · Multicasting · DCLC · TLBO · QoS

## 1 Introduction

Recently, multimedia contents such as IPTV, video conferencing, online gaming and video-on-demand are rapidly growing [1–3]. Generally, in computer networks, multimedia contents are delivered to a group or many users rather than a single user. In such cases, a group of users request the same content from a source. Multicasting is a technique, which distributes the same data from a source to a group of destination users simultaneously. Implementing

multicast techniques reduces network resources wastages and increases network performance [4].

Typically, multicasting is classified into two different types: IP multicasting and Application Layer Multicasting (ALM). In IP multicast, each source sends the data toward destinations over IP infrastructure network only once. In this way, intermediate routers along the path between the source and destinations are responsible to replicate the data packets to reach the destinations [5]. In addition to this, IP multicast routing protocols use multicast tree to route the data packets from the source to the destinations [6]. Due to in IP multicasting techniques, the data packets are transmitted over each link only once, network resources are used efficiently. Meanwhile, to enable IP multicasting, network equipments such as routers should support IP multicasting, and also network administrator should configure them. Furthermore, IP multicasting suffers from issues such as reliability, scalability and maintenance signaling [5,7]. ALM is another multicasting technique, which is implemented at the application layer to overcome the limitations of IP multicasting [8]. In ALM, multiple copies of the same content are transmitted via common links, i.e., each copy of content routed separately [7,8].

---

✉ Reza Mohammadi  
r.mohammadi@sutech.ac.ir

Reza Javidan  
Javidan@sutech.ac.ir

Manijeh Keshtgari  
Keshtgari@sutech.ac.ir

Reza Akbari  
akbari@sutech.ac.ir

<sup>1</sup> Department of Computer Engineering and IT, Shiraz University of Technology, Shiraz, Iran

Although ALM in compared to IP multicast has advantages such as adaptability, updatability and immediate deployability, it causes long delay and high bandwidth utilization. Due to the mentioned limitations of ALM and IP multicasting, providing QoS for multimedia contents is one of the most key concerns in multicasting [9].

SDN is a new networking paradigm, which separates control plane from data plane. SDN can facilitate network management and also enable network administrators to program and manage their networks dynamically [10,11]. In SDN, the data plane equipments, i.e., switches, handle network traffic based on the forwarding rules installed by the control plane, i.e., the controller. The logically centralized controller has a global view about network, and can collect network statistics. Due to the controller has a global knowledge about network, it can adaptively install appropriate forwarding rules over the data plane equipments regarding the network status. This mechanism makes SDN as a suitable candidate for managing multicast communications. In fact, the capabilities of SDN can help network providers to guarantee QoS for their customers. As the matter of fact, network providers can program the controller based on their desired policies in order to provide QoS regarding SLA. On the other hand, Most of the popular IP multicast protocols such as MOSPF [12], PIM-SM [13] and CBT [14] use shortest path tree (SPT) to route packets between the source and destinations. These IP multicast protocols suffer from requiring more signaling message to construct and manage the multicast tree. Moreover, they compute shortest path tree only based on the links costs which are static and determined by the network administrator. Such limitations of IP multicast protocols can be eliminated by using SDN. In SDN, the controller can install appropriate forwarding rules over the data plane switches using southbound APIs such as OpenFlow (OF) [15]. OF is an open standard for SDN, and widely used by developers and researchers [16]. As the matter of fact, OF is a protocol which determines how the controller can communicate with the data plane switches. Each OF switch has a set of flow tables and each flow table has a set of forwarding rules. A forwarding rule consists of a set of header fields and actions [17]. A forwarding rule can be installed by the controller reactively or proactively. In reactive mode, once an OF switch receives a packet, it first checks the flow tables to find any forwarding rules matched with the packet. If it finds a matched forwarding rule for the packet, it performs the related action. Otherwise, it forwards the packet to the controller [17]. Upon receiving the packet, the controller installs the appropriate forwarding rule for the packet. All future similar packets will be forwarded based on the installed forwarding rule. Unlike the reactive mode, the controller installs the forwarding rules before receiving any request from switches in proactive mode. It is evident that

the proactive mode is a proper way to develop desired traffic engineering methods.

In this paper, we proposed a novel and efficient traffic engineering approach for multicasting multimedia traffic over SDN. In our proposed technique, the controller gathers network statistics periodically to achieve the accuracy about the ongoing traffic and link conditions in the network. Then, it models multicast routing as a DCLC problem [18]. The objective of using DCLC for multicast routing is to compute the minimum cost multicast tree regarding end-to-end delay between the source and destinations. We assigned the cost of each link regarding to its utilization. Therefore, by computing the DCLC problem, the controller can achieve a multicast tree with low congested edges and satisfied end-to-end delay. In this way, the controller proactively installs forwarding rules over the data plane switches, i.e., OF switches to route multimedia traffic through an efficient multicast tree, and consequently increases QoS for multimedia contents. Due to NP-completeness of DCLC [19], we also proposed a heuristic method based on TLBO algorithm [20] to tackle this problem. We evaluated our proposed method for video conferencing application, and experimental results confirmed that our proposed method outperforms PIM-SM IP multicast routing and ALM in terms of various QoS metrics.

In summary, the contributions of this paper includes:

1. The QoS provisioning in multicast applications over SDN as a DCLC problem is modeled.
2. An optimization method based on TLBO algorithm is proposed as a solver for DCLC multicast routing.
3. The performance of the proposed multicast traffic engineering method is evaluated for a video conference application and compared with other solutions such as ALM and IP multicasting methods using Mininet emulator.

The remainder of this paper is organized as follows. In Sect. 2, we review some related works about multicast traffic engineering over SDN. We describe our proposed method in Sect. 3. Formulating and solving the DCLC problem using TLBO algorithm is described in Sect. 4. In Sect. 5, simulation details and test results are presented. Finally, conclusions are described in Sect. 6.

## 2 Related works

In this section, we will review some literatures about multicasting techniques in SDN. Iyer et al. [21] presented Avalanche as an SDN-based system to enable multicasting in commodity switches for data centers. Avalanche uses a multicast routing called Avalanche Routing Algorithm (AvRA) to minimize the size of routing tree for each multicast

group in data center. AvRA algorithm is suitable for FatTree topologies and efficiently increases the bandwidth utilization. However, avalanche is useful for multicast communication in the special network topologies like tree, and cannot be applied for any type of topology [22]. In [23], a clean slate approach called Multiflow is proposed for multimedia multicasting. The aim of Multiflow is to manage multicast group, and decrease end-to-end delay between the source and destinations. In Multiflow, the edge of each link in the network is assigned based on the distance between two nodes. Multiflow uses Dijkstra algorithm to compute shortest path tree. Due to Multiflow only takes care about delay, it does not guarantee packet loss reduction, and it might be cause high packet loss ratio.

Ananta et al. [22] introduced an extended Dijkstra algorithm to calculate shortest tree for multicast communication in SDN. Their proposed algorithm considers both edge weight and node weight to construct shortest multicast tree. Their experimental results show that the extended algorithm increases throughput and decreases jitter. However, there is no QoS guarantee mechanism in their proposed algorithm. In [24], an SDN-based testbed proposed for service assurance in IPTV networks. The authors implemented Dijkstra algorithm to compute multicast tree between IPTV server and customers. In addition, they proposed a new architecture for OpenFlow switches to support IPTV service. In their proposed approach, the cost of links is determined based on the quality of received video. In [25], a multicast framework called Castflow introduced by Marcondes et al., which manages multicast groups and performs multicast routing in SDN. Similar to Multiflow, the cost of each link in Castflow is defined as the distance between the source and destinations. Castflow computes Minimum Spanning Tree (MST) using Prim algorithm to route network traffic. Nonetheless, Castflow does not have any mechanism to provide QoS for end hosts' applications.

Noghani and Sunay [7] introduced a framework for multimedia multicasting over SDN. Their framework calculates minimum cost multicast tree using MiniMax algorithm. Their experimental results show that MiniMax incurs low packet loss. A fault tolerant IP multicasting technique over SDN is presented in [26]. The authors developed a method which computes two different multicast trees between source and destinations. Once a switch in primary multicast tree fails, the controller changes the forwarding rules of network to the alternative multicast tree. In [27], the authors presented a network-layer single-source multicast framework called Lcast. Lcast is an inter-domain multicast framework which uses Locator/ID Separation Protocol (LISP) overlay router to make the network scalable.

All the above reviewed multicasting techniques over SDN do not guarantee perfectly QoS for multimedia applications. QoS provisioning is required while transmitting multimedia

traffic. Because the end users receiving multimedia contents have constraints in terms of QoS metrics, which need to be satisfied by the network provider while transmitting the traffic to the end users. Therefore, in this paper we propose an efficient technique to enable network providers to deliver their multimedia services more efficiently regarding network conditions.

### 3 The proposed method

As we mentioned in Sect. 1, QoS provisioning for multimedia application is a critical issue for network providers. Generally, packet loss and end-to-end delay are two important QoS parameters in multimedia applications [28,29] which adversely affect the quality of received video at end users. Therefore, it is necessary to reduce these parameters in order to provide QoS. In legacy IP networks, multicast routing protocols compute Shortest Path Tree (SPT) to route the traffic along the tree. In this way, the cost of each link is usually determined by network administrator and multicast routing protocol does not take care about any QoS parameter such as packet loss and end-to-end delay. This is because in legacy networks, there is no efficient mechanism to achieve an accurate global view about network in order to deploy routing policy based on network condition. With the help of SDN, it is possible to gather current network condition and implement desired routing policies at run time. Typically, in computer networks, higher link utilization results link congestion, thus might lead to packet drops. If the congestion of a link exceeds the ingress buffer size of the relevant switch, it causes packet loss. To tackle this problem, one common solution is to route traffic through low congested links.

In this paper, we defined the link cost as the utilization of the link. This policy leads to compute the minimum multicast tree with low congested links. In addition to this, in most multimedia applications end-to-end delay of each video packet should not exceed 200ms [30,31], because the exceeded packets affect the quality of received video. To guarantee this constraint, our proposed method models the QoS multicast routing as a DCLC problem, which we will discuss in Sect. 4. Figure 1 illustrates our proposed traffic engineering architecture for multicast communications in SDN.

According to Fig. 1, our proposed traffic engineering schema is implemented in application layer over control plane, and consists of following modules:

- Statistics Collector Module (SCM): it periodically collects network links statistics i.e. the delay and utilization of links, stores the statistics in two separate matrixes, called `delay_matrix` and `utilization_matrix`. For delay calculation of a link between two specific switches, this module creates a new type packet and inserts current time

as a timestamp field in the header field of the packet. This packet is sent to the first switch (front of the link), and the SCM module enforces the switch to forward the packet toward the tail of the link. Upon reception of the packet, since there is not any matched forwarding rule for the packet in the flow table of the second switch (tail of the link), it is forwarded toward the controller. The SCM module can measure the delay of the link by subtracting the timestamp value of the header field of the packet from current time. This process is performed for all links in the network. In order to collect the utilization of each link, SCM module gathers the statistics of the connected ports of the switch in both sides of a link and divides the current transmitting rate of the port to the capacity of the port. Then, the maximum obtained value for both sides of the link is considered as the utilization of the link. This process is performed for all links in the network.

- **Multicast Tree Finder Module (MTFM):** it is responsible for calculating minimum cost multicast tree by modeling the multicast routing as a DCLC problem. After this, it executes  $TLBO()$  to solve the problem.  $TLBO()$  function uses  $delay\_matrix$  and  $utilization\_matrix$  as input values to assign links' costs and satisfy delay constraint in DCLC. The output of  $TLBO()$  function is the minimum cost multicast tree.
- **Route Inserter Module (RIM):** it is executed after SCM and MTFM in order to inject the output of MTFM as a set of forwarding rule entries into the relevant switches.

The controller periodically executes SCM, MTFM and RIM, respectively. This mechanism facilitates QoS provisioning because the controller computes efficient multicast tree based on current status of the network by considering delay constraint. Moreover, the cost of each link is determined as a function of its utilization, thus it computes minimum congested multicast tree, which reduces packet loss.

### 4 Problem formulation

DCLC is one of the linear programming problems, which aims to minimize the cost of multicast tree while guaranteeing the end-to-end delay constraint between the source and destinations [18]. In DCLC, the communication network is modeled as an undirected connected weighted graph  $G = (N, L)$ , where  $N$  is s set of network nodes and  $L$  is a set of network links. The link  $e = (i, j)$  denotes the link from  $i \in N$  to the node  $j \in N$ .  $C(e)$  and  $d(e)$  are the weighted cost and delay for the link  $e$ , respectively. In this paper, we consider  $C(e)$  as the utilization of link  $e$ . Since the graph is undirected, we set maximum value of the utilization of link  $e$  in both sides as a cost of the link. Let  $s \in N$  be a source node and  $D = \{v_1, v_2, \dots, v_n\}$  is a set of the destination nodes,

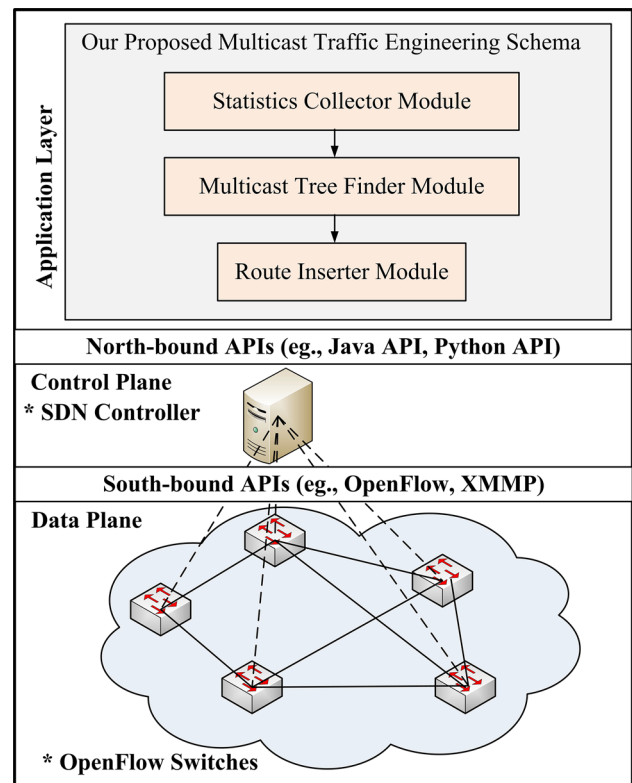


Fig. 1 Architecture of the proposed method

respectively. Here, the index  $n$  is the number of destinations. Let  $T = (s, D)$  be a tree which connects  $s$  to all destinations, and  $P(s, v)$  a feasible path from  $s$  to  $v$ . The minimum cost multicast tree  $MT^*$  can be defined as follows:

$$MT^* = \min \sum_{v \in D} \sum_{e \in P(s,v)} cost(e) \quad \forall v \in D, P(s, v) \in T \quad (1)$$

The end-to-end delay of  $P(s, v)$  is the aggregated delay of each link along the path and is calculated as follows:

$$delay(P(s, v)) = \sum_{e \in P(s,v)} d(e) \quad (2)$$

Finally, DCLC problem is defined as follows:

$$MT^* = \min \sum_{v \in D} \sum_{e \in P(s,v)} cost(e) \quad (3)$$

$$s.t. \quad delay(P(s, v)) \leq \Delta$$

$$\forall v \in D, P(s, v) \in T$$

where  $\Delta$  is the end-to-end delay threshold of the path between  $s$  and  $v$ . Due to NP-completeness of DCLC problem [19], some heuristic and approximation algorithms have been proposed in literatures [32,33]. In this paper, to overcome this limitation, we propose a heuristic DCLC solver using TLBO algorithm.

TLBO [20] is a nature-inspired optimization algorithm, which is based on the effect of influence of a teacher on learners. In fact, TLBO emulates the teaching–learning process of a classroom to solve problems in an efficient manner [34]. In a classroom, a teacher attempts to increase the average result of the class. In addition to this, a learner with less knowledge can learn new knowledge from other learner who is better than him/her. TLBO inspires this mechanism as well, and describes two main modes of the learning process: teacher phase and learner phase. In teacher phase, the learners increase their knowledge by learning from the teacher while in learning phase; the learner interacts with each other in order to increase their knowledge. TLBO is a population-based algorithm in which a set of learners is considered as population, and the best solution in the population is defined as the teacher, i.e., the best solution of the objective function. One of the main advantages of TLBO is that it requires only the population size and the number of iterations as the control parameters. Each iteration in this algorithm both the teacher and learner phases are executed, and if the best solution (teacher) is better than existing solution, it is replaced with new one. Moreover, during each iteration the learners are improved based on teacher and themselves. The best solution is returned when the number of iterations is achieved.

Our proposed traffic engineering method takes the advantage of TLBO algorithm to solve DCLC problem. To this purpose, we consider a learner as a feasible multicast tree from source to destinations. In the initialization step, a set of learners is randomly created. The position of the  $k$ -th learner is represented as follows:

$$X_k = \{x_{k1}, x_{k2}, \dots, x_{kn}\} \tag{4}$$

where  $n$  is the number of destination nodes of learner  $k$  and  $x_{ki}$  is a feasible path from the source node to destination  $i$ . According to TLBO algorithm, the new position of the  $k$ -th learner in teacher phase is calculated as follows:

$$X_{k,new} = X_{k,old} + r[X_{teacher} - (T_F \cdot X_{mean})] \tag{5}$$

where  $X_{k,old}$  is the previous position of learner  $k$ ,  $r$  is a random number in the range  $[0, 1]$ ,  $X_{teacher}$  is the best solution,  $X_{mean}$  is the mean, respectively. Variable  $T_F$  is teaching factor which can be either 0 or 1. In fact, this variable decides the value of mean to be changed and calculated as follows:

$$T_F = \text{round}[1 + \text{rand}(0, 1) * (2 - 1)] \tag{6}$$

As we mentioned before, each learner increase its knowledge in learning phase as well as teacher phase. In learning phase, learner  $k$  randomly selects learner  $j$  and then the new position of learner  $k$  is updated as follows:

If  $f(X_k) < f(X_j)$

$$X_{k,new} = X_{k,old} + r[X_k - X_j]$$

Else

$$X_{k,new} = X_{k,old} + r[X_j - X_k]$$

where  $r$  is a random number in the range  $[0, 1]$ .

Algorithm 1: our proposed TLBO algorithm for DCLC problem

```

Inputs:
Cost: utilization_matrix
Delay: delay_matrix
Δ: delay constraint parameter
D: the set of destination nodes , i.e., D=1,2,...,n
s : source node
P: population size
Begin_TLBO()
Set i=1
X = {x1, x2, ..., xn}
Objective function f(X)=∑v∈D ∑e∈P(s,v) cost(X)
Create initial population randomly Xk k = 1,2,...,P
Calculate objective function f(X) for whole learners
WHILE i <= MaxIteration
Function Teacher_Phase()
Calculate the mean of each design variable → Xmean
Identify minimum cost solution → Xteacher
For k=1 to P
TF = round[1 + rand(0,1) * (2 - 1)]
Xk,new=Xk,old + r [Xteacher - (TF·Xmean)]
Calculate objective function for
new mapped learner f(Xk,new)
If f(Xk,new) < f(Xk,old) and Xk,new satisfies
Δ delay threshold
Xk,old = Xk,new
End If
End For
End Function
Function Learner_Phase()
For k=1 to P
Randomly select another learner Xj , such that j ≠ k
If f(Xk,old) < f(Xj)
Xk,new = Xk,old + r[Xk,old - Xj]
Else
Xk,new = Xk,old + r[Xj - Xk,old]
End If
If f(Xk,new) < f(Xk,old) and Xk,new satisfies
Δ delay threshold
Xk,old = Xk,new
End If
End For
End Function
Set i=i+1
End While
Post-process results
Return best solution
End TLBO
    
```

The pseudo code shown in Algorithm 1 demonstrates our proposed TLBO algorithm for DCLC problem in detail. As shown in Algorithm 1, our proposed algorithm receives *utilization\_matrix* and *delay\_matrix* to compute the cost of solution, and it checks the end-to-end delay constraint. In each iteration, if the new solution is better than current solution and satisfies delay constraint, it is replaced with the current solution. After the termination condition is met, the post-processing function converts the best solution to a vector as a suitable input for RIM module, which is responsible to install paths. In this paper, we set MaxIteration to 100, and the number of learners (population size) to  $0.8 \times \text{number of nodes}$  in the network.

## 5 Performance evaluation

In this section, we conducted a comprehensive simulation in order to evaluate our proposed method. To assess our SDN-based multicast traffic engineering method, we compared it with IP multicast (PIM-SM) and ALM. PIM-SM is widely used in IP multicast and uses a shortest path multicast tree for routing between source and destinations [13]. As we mentioned in Sect. 1, in IP multicast, routes are computed by the routers in a distributed manner. Unlike SDN, the routers just have a local view on the network.

### 5.1 Experimental setup

We used mininet [35] as an emulator for implementing all target scenarios along with OpenDayLight [36] as a network controller. Since PIM-SM is not implemented in Mininet switches, this paper leverages the OpenDaylight controller to compute PIM-SM multicast trees by installing forwarding rules in the switches. The multicast tree of PIM-SM is calculated using SPT algorithm in which the cost of the route between the root of the tree and each destination is minimum. In PIM-SM, the cost of each link depends on the capacity of that link. Indeed, higher capacity of the link leads to the lower cost value. Moreover, in our experiments, ALM sends the data packets in unicast manner through shortest paths. All experiments are executed on a machine with following configuration: CPU Intel Core i7-4700MQ-2.4GHz and 6 GB RAM. To investigate the behavior of the proposed method under different workloads, we simulated two different test-bed scenarios: small and large. The topologies of the scenarios are generated with BRITTE [37] topology generator using well-known Waxman's probability model [38]. The bandwidth of each link in both topologies is set 10Mbps, and the delay for each link is randomly assigned from 20 to 100ms. Furthermore, we generated artificial cross traffic to congest the network and assess the performance of multicas-

ting method under different workloads. Configurations of the scenarios is as follows:

- **Scenario-1** the test-bed network consists of 10 OpenFlow switches. There are 16 links in the network for connecting OpenFlow switches together, and each switch connects to an average 3.2 other switches. There are 2 cross traffic which generate 3 Mbps UDP traffic.
- **Scenario-2** the test-bed network consists of 30 OpenFlow switches. There are 36 links in the network for connecting OpenFlow switches together, and each switch connects to an average 2.13 other switches. There are 4 cross traffic which generate 3 Mbps UDP traffic.

For all simulations, the controller is configured as a way to query the network statistics, execute the proposed method every 20s and announce the minimum cost multicast tree to the relevant switches. We have adjusted different interval values for {5, 10, 15, 20, 25, and 30} s. The best solution obtained in 20s interval. The maximum tolerable end to end delay for video streaming is set to 200ms [30]. This value is the  $\Delta$  threshold value in the DCLC problem which is used for modeling the multicast tree. To emulate the video conferencing application, we configured the source node to stream video traffic with resolution 1280\*720 at 30 fps to the multicast group for 300s. The video content which is streamed by the source node is the Big Buck Bunny animation [39]. The simulation results are analyzed for different multicast group sizes in terms of following QoS metrics:

- PSNR (Peak Signal-To-Noise Ratio): PSNR is a metric which is useful to measure video quality. This metric is used to calculate the error between the original streamed video and the received video.
- Average end to end delay: it is the average time interval from source to destinations for each successfully delivered video packet.
- Jitter: it is defined as an average of the deviation from average latency. The variation in arrival time of packets might be caused by network congestion or route changes.
- Packet loss ratio: it is the total number of video packets which haven't been delivered at destinations to the total number of packets streamed at the source of traffic.

### 5.2 Experimental results

In this section, we discuss and analyze the results for the two scenarios under different size of multicast group. Figures 2 and 3 illustrate average jitter and average end-to end delay for both scenarios, respectively. As it can be seen from Figs. 2 and 3, it can be concluded that our proposed method

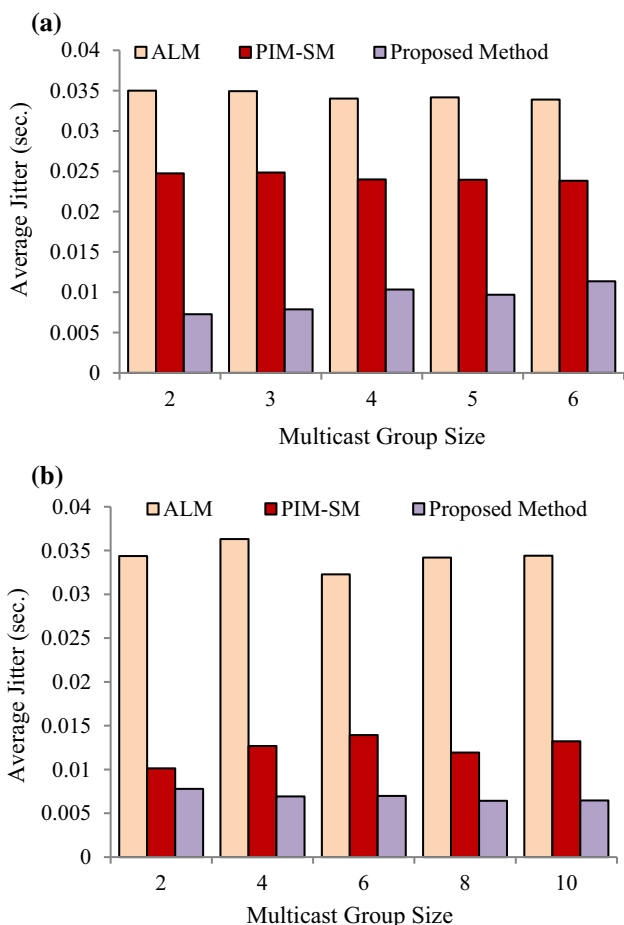


Fig. 2 Average jitter versus multicast group size. a Scenario-1, b scenario-2

reduces jitter and end-to-end delay compared to ALM and PIM-SM. This is because of two reasons: first our proposed method gathers the statistics of link delays periodically. This mechanism helps to choose links with low delay. Second, our proposed models multicast QoS routing as a DCLC problem which guarantees end-to-end delay constraint. On the other hand, PIM-SM only computes multicast tree based on the link costs which are assigned by administrator. In this way, multimedia packets might be routed through the links which have long delay. As a result, the multimedia packets incur long delay. We observe that ALM has the worst performance in terms of jitter and end-to-end delay. Because in ALM the source streams multimedia contents for each client separately, more traffic are injected to the network and causes delay for each stream.

Simulation results for the packet loss ratio in Fig. 4 for different number of clients per group confirm that our proposed method achieves better performance. As we mentioned in previous section, when our proposed method models the routing to DCLC, it assigns link costs with respect to the

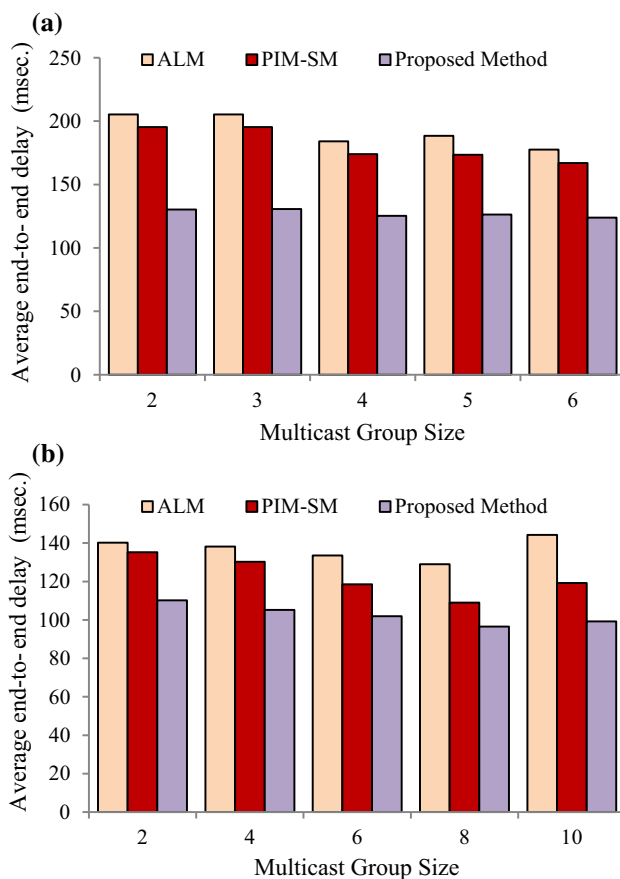
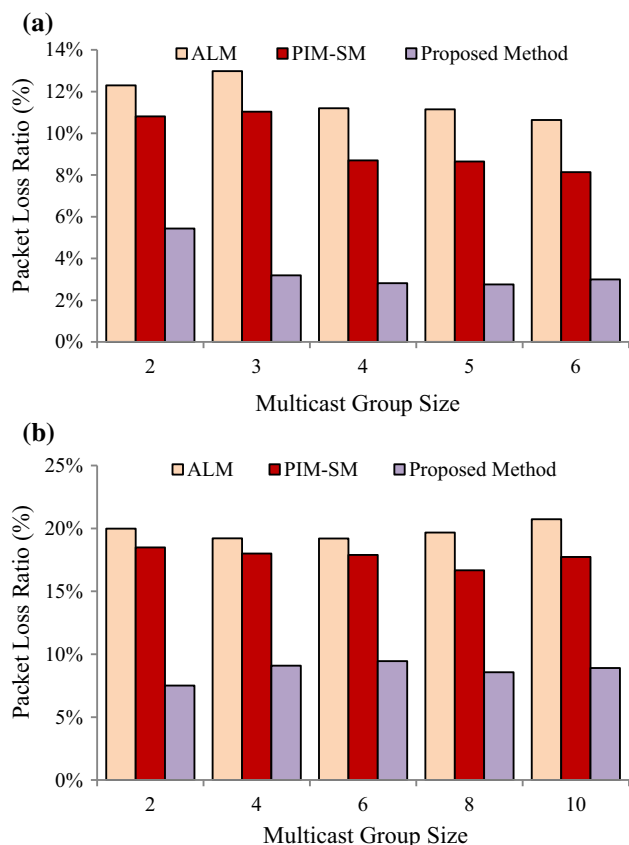


Fig. 3 Average end-to-end delay versus multicast group size. a Scenario-1, b scenario-2

utilization of links. Higher the link utilization link, higher the link cost, therefore when TLBO algorithm solves DCLC problem, it chooses low cost links, i.e., low utilized links to create minimum multicast tree. Consequently, multimedia contents are routed through low utilized link which are not congested. This mechanism not only reduces delay but also decreases the probability of packet loss due to congestion and buffer overflow. In our proposed method, the controller periodically gathers link utilization statistics and computes multicast tree. Therefore, if a link or set of links are congested, in the next time that the controller computes multicast tree, other low congested links are computed to create multicast tree.

Figure 4 also shows that PIM-SM and ALM have higher packet loss ratio. Because these two methods do not take care about link congestion for computing path between source and destination, the packet loss ratio is higher in compared to our proposed method. Moreover, in ALM the source streams multiple copy of the same multimedia content which leads more link utilization and more packet loss ratio.

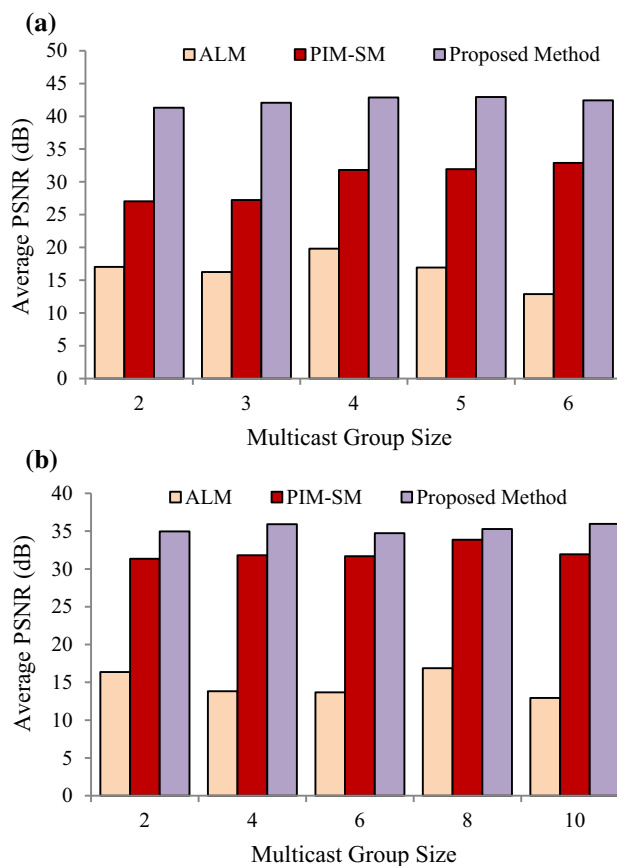
Figure 5 shows the simulation results for the important multimedia QoS metric, i.e., PSNR. With respect to this met-



**Fig. 4** Average packet loss ratio versus multicast group size. **a** Scenario-1, **b** scenario-2

ric, we can judge about the received video at the client side. Higher value for PSNR means that the video at the receiver side has higher quality. In multimedia applications, packet loss is one of the most important parameters which reduces the PSNR. As the matter of fact, due to packet loss in network, some packets of video are not received by the receiver. Therefore, some frames of original streamed video will be corrupted, i.e., the PSNR value reduces.

From Fig. 5, we can conclude that the proposed method increases the PSNR significantly compared to other two methods. Because our method uses low congested and short delay links to compute multicast route and also periodically gathers the network statistics, it achieves better performance and as a result, causes to increase the PSNR. Moreover, We observe from Fig. 5 that our proposed method achieves a gain of about 40 and 10% for PSNR in Scenario-1 and Scenario-2, respectively. In contrast to our method, PIM-SM and ALM are static methods and as we mentioned before, they do not have any mechanism to take care about packet loss and end-to-end delay. Because ALM consumes more resources compared to PIM-SM and also uses more bandwidth, it has the worst performance in terms of PSNR.



**Fig. 5** Average PSNR versus multicast group size. **a** Scenario-1, **b** scenario-2

## 6 Conclusions

In this paper, we proposed an adaptive multicast traffic engineering method for multicast communication over SDN. Traditional IP multicast techniques have limitations to provide QoS for multicast traffic such as multimedia. To overcome the limitations of traditional techniques, our proposed approach takes the advantages of SDN to facilitate network providers in order to provide QoS for multimedia applications. In this paper, first we modeled multicast routing as a DCLC problem and we proposed a heuristic algorithm based on TLBO to solve the problem. Then, we evaluated the proposed method for video conferencing applications under different workloads and scenarios. We investigated the results with respect to important QoS metrics in multimedia applications such as PSNR, packet loss and delay. Experimental results confirmed that our proposed method outperforms other multicasting techniques such as PIM-SM and ALM.



## References

- Szymanski, T. H., & Gilbert, D. (2010). Design of an iptv multicast system for internet backbone networks. *International Journal of Digital Multimedia Broadcasting*, 2010(1), 1–14.
- Schmitt, M., Redi, J., Cesar, P., & Bulterman, D. (2016). 1 mbps is enough: Video quality and individual idiosyncrasies in multiparty hd videoconferencing. In *2016 Eighth international conference on quality of multimedia experience (QoMEX)* (pp. 1–6). IEEE.
- Pavlić, M., & Bratković, D. (2015). Comparison of different ngn aggregation networks scenarios. In *2015 57th international symposium ELMAR (ELMAR)* (pp. 81–84). IEEE.
- Lee, J., & de Veciana, G. (2001). Resource and topology discovery for ip multicast using a fan-out decrement mechanism. In *INFOCOM 2001. Proceedings of the twentieth annual joint conference of the IEEE computer and communications societies, IEEE* (Vol. 3, pp. 1627–1635). IEEE.
- Gu, W., Zhang, X., Gong, B., & Wang, L. (2015). A survey of multicast in software-defined networking. In *2015 5th international conference on information engineering for mechanics and materials (ICIMM)*.
- Mohammadi, R., Javidan, R., & Keshtgari, M. (2017). OpenIPTV: A comprehensive SDN-based IPTV service framework. *Multimedia Systems*. <https://doi.org/10.1007/s00530-017-0553-x>.
- Noghani, K. A., & Sunay, M. O. (2014). Streaming multicast video over software-defined networks. In *2014 IEEE 11th international conference on mobile ad hoc and sensor systems* (pp. 551–556). IEEE.
- Hosseini, M., Ahmed, D. T., Shirmohammadi, S., & Georganas, N. D. (2007). A survey of application-layer multicast protocols. *IEEE Communications Surveys & Tutorials*, 9(3), 58–74.
- Lao, L., Cui, J.-H., Gerla, M., & Chen, S. (2007). A scalable overlay multicast architecture for large-scale applications. *IEEE Transactions on Parallel and Distributed Systems*, 18(4), 449–459.
- Hu, F., Hao, Q., & Bao, K. (2014). A survey on software-defined network and openflow: From concept to implementation. *IEEE Communications Surveys & Tutorials*, 16(4), 2181–2206.
- Kreutz, D., Ramos, F. M., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 103(1), 14–76.
- Moy, J. (1994). MOSPF: Analysis and Experience. RFC 1585. *Internet Engineering Task Force, Fremont*. Available from <https://tools.ietf.org/html/rfc1585>.
- Farinacci, D., Liu, C., Deering, S., Estrin, D., Handley, M., & Jacobson, V., et al. (2006). Protocol independent multicast-sparse mode (pim-sm): Protocol specification. RFC 4601.
- Ballardie, T., Francis, P., & Crowcroft, J. (1993). Core based trees (CBT). *ACM SIGCOMM Computer Communication Review*, 23(4), 85–95.
- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., et al. (2008). Openflow: Enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 38(2), 69–74.
- Lara, A., Kolasani, A., & Ramamurthy, B. (2014). Network innovation using openflow: A survey. *IEEE Communications Surveys & Tutorials*, 16(1), 493–512.
- Specification, O. S. (2011). Version 1.2 (wire protocol 0x03). Open Network Foundation.
- Sriram, R., Manimaran, G., & Murthy, C. S. R. (1998). Preferred link based delay-constrained least-cost routing in wide area networks. *Computer Communications*, 21(18), 1655–1669.
- Xiao, Y., Thulasiraman, K., Fang, X., Yang, D., & Xue, G. (2012). Computing a most probable delay constrained path: Np-hardness and approximation schemes. *IEEE Transactions on Computers*, 61(5), 738–744.
- Rao, R. V., Savsani, V. J., & Vakharia, D. (2011). Teaching-learning-based optimization: A novel method for constrained mechanical design optimization problems. *Computer-Aided Design*, 43(3), 303–315.
- Iyer, A., Kumar, P., & Mann, V. (2014). Avalanche: Data center multicast using software defined networking. In *2014 sixth international conference on communication systems and networks (COMSNETS)* (pp. 1–8). IEEE.
- Ananta, M. T., Jiang, J.-R., & Muslim, M. A. (2014). Multicasting with the extended dijkstra's shortest path algorithm for software defined networking. *International Journal of Applied Engineering Research*, 9(23), 21017–21030.
- Bondan, L., Müller, L. F., & Kist, M. (2013). Multiflow: Multicast clean-slate with anticipated route calculation on openflow programmable networks. *Journal of Applied Computing Research*, 2(2), 68–74.
- Thorpe, C., Olariu, C., Hava, A., & McDonagh, P. (2015). Experience of developing an openflow sdn prototype for managing iptv networks. In *2015 IFIP/IEEE international symposium on integrated network management (IM)* (pp. 966–971). IEEE.
- Marcondes, C. A., Santos, T. P., Godoy, A. P., Viel, C. C., & Teixeira, C. A. (2012). Castflow: Clean-slate multicast approach using in-advance path processing in programmable networks. In *IEEE symposium on computers and communications (ISCC)*. IEEE.
- Kotani, D., Suzuki, K., & Shimonishi, H. (2012). A design and implementation of openflow controller handling ip multicast with fast tree switching. In *2012 IEEE/IPSJ 12th international symposium on applications and the internet (SAINT)* (pp. 60–67). IEEE.
- Coras, F., Domingo-Pascual, J., Maino, F., Farinacci, D., & Cabellos-Aparicio, A. (2014). Lcast: Software-defined inter-domain multicast. *Computer Networks*, 59, 153–170.
- Wang, Z., & Crowcroft, J. (1996). Quality-of-service routing for supporting multimedia applications. *IEEE Journal on Selected areas in communications*, 14(7), 1228–1234.
- Campbell, A., Coulson, G., Garcia, F., Hutchinson, D., & Leopold, H. (1993). Integrated quality of service for multimedia communications. In *INFOCOM' 93. Proceedings. Twelfth annual joint conference of the IEEE computer and communications societies. Networking: Foundation for the future, IEEE* (pp. 732–739). IEEE.
- Egilmaz, H. E., Dane, S. T., Bagci, K. T., & Tekalp, A. M. (2012). Openqos: An openflow controller design for multimedia delivery with end-to-end quality of service over software-defined networks. In *Signal & Information processing association annual summit and conference (APSIPA ASC), 2012 Asia-Pacific* (pp. 1–8). IEEE.
- Pantel, L., & Wolf, L. C. (2002). On the impact of delay on real-time multiplayer games. In *Proceedings of the 12th international workshop on network and operating systems support for digital audio and video* (pp. 23–29). ACM.
- Ling, Z., Wei-xiong, D., & Yu-xi, Z. (2010). Delay-constrained multicast routing algorithm based on average distance heuristic. arXiv preprint [arXiv:1003.3317](https://arxiv.org/abs/1003.3317).
- Xu, Y., & Qu, R. (2012). A hybrid scatter search meta-heuristic for delay constrained multicast routing problems. *Applied Intelligence*, 36(1), 229–241.
- Rao, R. V., & Patel, V. (2013). An improved teaching-learning-based optimization algorithm for solving unconstrained optimization problems. *Scientia Iranica*, 20(3), 710–720.
- <http://mininet.org/download/>. Visited on 03/17/2016.
- <https://www.opendaylight.org/> Visited on 03/17/2016.
- Medina, A., Lakhina, A., Matta, I., & Byers, J. (2001). Brite: An approach to universal topology generation. In *Proceedings of the ninth international symposium on modeling, analysis and simulation of computer and telecommunication systems, 2001.* (pp. 346–353). IEEE.

38. Waxman, B. M. (1988). Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9), 1617–1622.
39. <https://peach.blender.org/>. Visited on 03/17/2016.



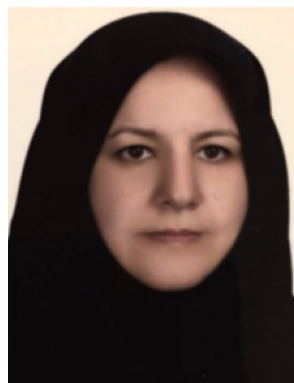
**Reza Mohammadi** born in 1986. He is graduated from MSc degree in Computer Networks from Shiraz University of Technology in 2013. Now he is a Ph.D. student on Computer Networks in Shiraz University of Technology. He has many publications in international conferences and journals regarding Underwater Wireless Sensor Networks (UWSNs). Currently he focused on Software Defined Networks (SDN) as a new trend in computer networks. His major fields of interest are Software

Defined Networks (SDNs), heuristic algorithms, Underwater Wireless Sensor Networks (UWSNs), Ad hoc Networks and Underground Wireless Sensor Networks (UWSNs).



**Reza Javidan** born in 1970. He is graduated from M.Sc. Degree in Computer Engineering (Machine Intelligence and Robotics) from Shiraz University in 1996. He received Ph.D. degree in Computer Engineering (Artificial Intelligence) from Shiraz University in 2007. His major fields of interest are Network security, Underwater Wireless Sensor Networks (UWSNs), Software Defined Networks (SDNs), artificial intelligence, image processing and SONAR systems. He is

now member of faculty and lecturer in Department of Computer Engineering and Information Technology in Shiraz University of Technology.



**Manijeh Keshtgari** received her B.Sc. in Computer engineering from Shiraz University in 1986, her Master in Electrical and Computer Eng. from Colorado State University, USA in 1992 and Ph.D. degree in Computer Eng. from Sharif University of Technology in 2004. Her research interests include wireless Networks, Fiber Optic Networking (SDN) and Named Data Networking (NDN). He is now member of faculty and lecturer in Computer Engineering and

IT Department in Shiraz University of Technology. In addition she is lecturer of Computer Science in Department of Computer Science in University of Georgia, USA.



**Reza Akbari** received his M.Sc. in artificial intelligence from Isfahan University of Technology (2006). He has received his Ph.D. from Shiraz University (2011). He is an Assistant Professor at the department of computer engineering and information technology of the Shiraz University of Technology. His areas of research include evolutionary computation, engineering optimization, and search based software engineering.