A QoE-driven Multicast Strategy with Segment Routing - A Novel Multimedia Traffic Engineering Paradigm

Shujie Yang, Changqiao Xu, *Senior Member, IEEE*, Lujie Zhong, Jiahao Shen and Gabriel-Miro Muntean, *Senior Member, IEEE*

***Abstract*—Quality of Experience (QoE) reflects end users' overall experience and feeling with network services, but needs support in terms of end-to-end Quality of Service (QoS). Segment routing (SR) as a new routing paradigm can provide good end-to-end QoS guarantee, making traditional multimedia traffic routing more efficient and scalable. In this paper, we address two problems related to the new SR mechanism: enabling fine-grained end-to-end QoS routing under a complex network environment and constructing the multicast routing tree with branch node load balancing. To solve these problems, an Inaccurate information-based QoE-driven Routing algorithm (IQdR) and a Branch-aware Multicast Tree (BaMT) algorithm were proposed. Simulation test results that have compared the performance of our proposed solution against that of other algorithms show that the previous works were outperformed. Additionally, the results also show that our multicast architecture improves the scalability of the network in terms of the number of flows.**

***Index Terms*—Traffic Engineering; Segment Routing; Multicast Tree; QoE-driven Routing.**

I. Introduction

Lately, multimedia consumption is gradually shifting from traditional TV to streaming video, including Smart TVS and mobile devices. In addition, Quality of Experience (QoE) is critical for evaluation of customer satisfaction, retention and network services [1]. For the foreseeable future, users' expectations in terms of quality, choice and convenience will continue to increase [2]-[4]. In order to meet these needs, traffic engineering (TE) is a possible paradigm to enhance substantially the experience of multimedia consumers by providing end-to-end Quality of Service (QoS) guarantees [5]-[7]. The TE concept was originally introduced in late 1990s. Its principle is to fully utilize the label switching system to control different traffic flows in different paths under a Multi-Protocol Label Switching (MPLS) environment. Critical services can employ TE and take reliable paths to guarantee quality. Also the network can dynamically adjust any path when network congestion occurs with TE strategies. Therefore, TE is a powerful strategy for implementation of end-to-end QoS or differentiated services. These mainly include QoS unicast routing and QoS multicast routing, which help meet QoE demand in multimedia networks. However, TE has not been widely used in domestic telecommunications and internet field because of the complexity of its control plane, especially when considering QoS multicast routing [8]. In traditional networks, QoS multicast routing suffers from numerous limitations because it needs to exchange a large amount of computation and caching information to maintain the multicast tree. Fortunately, by employing some innovative multicast-based approaches [9], [10], and especially the emerging Software Defined Networking (SDN) paradigm [11], [12], these limitations can be overcome.

In SDN, the control plane and the data plane are separated. The control plane processes information and controls data traffic in the network, whereas the data plane carries it. QoS routing in SDN no longer requires maintaining network state information among routers. Instead, the SDN control plane performs computations and traffic control. At the same time, the SDN control plane can also employ fine-grained table lookup, which is more beneficial to the QoS routing in comparison with the coarse-grained destination-based forwarding in traditional legacy networks. However, fine-grained forwarding requires larger sized tables which means more of the expensive content addressable memory is needed. In addition, implementation of QoS routing needs complex signaling protocols. These all greatly limit the scalability of SDN-based strategies, increasing the difficulty of implementing QoS routing.

In order to address these limitations, employing segment routing (SR) based on SDN as an alternative architecture is highly promising. The SR architecture [13], [14] was standardized by the Internet Engineering Task Force (IETF) SPRING working group [15] to provide effective TE and has been driven by Cisco and supported by many leading telecom companies. The main characteristic of SR is that it leverages the source routing paradigm, where the path description is carried in the packets header. Per-flow states will be maintained only by the ingress nodes, offloading the load of

This work was partially supported by the National Natural Science Foundation of China (NSFC) under Grants Nos. 61872253 and 61871048, by the Science Foundation Ireland (SFI) Research Centres Programme under Grant Number 12/RC/2289 (Insight Centre for Data Analytics) and 16/SP/3804 (ENABLE), by the 111 Project (B18008), by the Open Foundation of State Key Laboratory of Networking and Switching Technology (Beijing University of Posts and Telecommunications) under Grant SKLNST-2018-1-05 and by the China Postdoctoral Science Foundation (2019M650590).

S. Yang, C. Xu and J. Shen are with State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, P. R. China. (e-mail: {zongshi, cqxu}@bupt.edu.cn; shenjiahaoplus@gmail.com).

L. Zhong is with Information Engineering College, Capital Normal University, Beijing, 100048; and the State Key Laboratory of Networking and Swiching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, P. R. China. E-mail: zhonglj@cnu.edu.cn.

G.-M. Muntean is with Performance Eng. Lab, School of Electronic Engineering, Dublin City University, Ireland. E-mail: gabriel.muntean@dcu.ie.

Corresponding author: Lujie Zhong

maintaining forwarding information on other transmit nodes. Consequently, a SR architecture avoids storing millions of label information in each network device along the path, which reduces the amount of forwarding rules in Ternary Content Addressable Memory (TCAM) [16]. Furthermore, signaling protocols such as LDP and RSVP-TE in the traditional MPLS paradigm are not required by SR anymore, making the network more scalable and flexible.

Existing research involving TE with a SR mechanism mainly focused on QoS unicast routing, while QoS multicast routing in this area attracted little attention. Some research performed on QoS multicast routing based on SR was limited to single target optimization i.e. bandwidth optimization or delay optimization [17]. This cannot reflect the characteristics of a complex multimedia network, and does not achieve true QoE-driven routing. Research focused on multi-object multi-constrict (MOMC) problem in multicast routing based on this new architecture are still in its infancy. In addition, how to perform multicast routing in the SR structure is also a tricky problem because SR is based on source address routing, which does not support content replication at branch nodes.

Therefore, this paper focuses on two major problems related to the SR mechanism. The first problem refers to performing fine-grained QoS routing in a complex network environment. To solve it efficiently, a MOMC-based optimization model is proposed in the complex network environment, which also has multiple limitations. These limitations are mostly QoS related i.e. bandwidth, delay, packet loss rate, throughput, etc. Furthermore, considering that in general the information obtained from a complex network has reduced accuracy, statistical theory was used to solve MOMC problems, which have been proven to be NP-hard. To the best of authors' knowledge, this is the first research focusing on solving a MOMC problem in this field.

The second issue is constructing the multicast routing tree based on SR. This paper presents an innovative solution which employs a segment identifier at the branch node to realize the multicast routing with SR. In addition, a further study about load balancing at branch node was also carried on. The main contributions of this paper are summarized as follows:

- A new SR mechanism-based multicast routing model is proposed, with the goal to achieve higher scalability and efficiency. In this model, segment identifier and load balancing processing are used at branch nodes to realize multicast routing with SR.
- The MOMC routing problem is addressed for the first time in the context of a SR architecture. The Inaccurate information-based QoE-driven Routing algorithm (IQdR) is proposed, which focuses on the multi-objective probability as optimization target. IQdR solves the MOMC problem and performs QoS routing in complex network conditions based on inaccurate information exchange.
- Construction of the branch-aware Multicast Tree (BaMT) is performed, in which, a K-candidate multicast tree algorithm is proposed to construct the multicast tree with load balancing.
- The performance of the proposed solution based on the new architecture is analysed. The performance is assessed in comparison with other algorithms such as Steiner Tree (ST), Shortest Path Tree (SPT), Widest Shortest Path Tree (WSPT) and so on. As the simulation results are better than those of other algorithms, the superiority of our solution based on the new SR architecture is demonstrated.

II. Related Work

A. Segment Routing (SR)

SR as a new network paradigm was proposed in 2013 and has been standardized by IETF [18, 14]. Relevant research work about SR is mainly focused on three aspects: implementation of SR [19],[20], segment list (SL) encoding [21], [22] and TE optimization algorithms [23], [24].

Research related to implementation of SR aimed at the transition from current pure IP network to a SR network. Similar approaches, i.e., deployment of Software Defined Networking (SDN) to legacy networks, have been tried in the last years, and were noted as difficult and sometimes impossible. A possible solution is through soft transition i.e. install SR on IP routers. In [25], authors proposed an architecture to combine the benefits of SR with those of a SDN control plane. An incremental deployment of SR into an ISP network has been proposed in [11]. The work in [26] implemented SR in carrier grade Ethernet networks and performed a detailed simulation study. As SR-research is just emerging, there are many problems that need to be solved to completely transmit to current IP networks, such as SL encoding and TE optimization algorithms.

SL encoding is used as basis to implement SR routing. Its purpose is to design the corresponding coding algorithm to realize the minimum expression of path segment, so as to avoid the limitation of the Maximum SL Depth (MSD) and reduce the length of packet header. Different researches have been proposed to limit the SL length [27]-[30]. Among of them, [27] adopted a greedy algorithm to compute the minimum depth of SL, whereas [30] have determined SL by utilizing an auxiliary graph model.

The object of SL encoding is path computation, which is selected via the TE optimal algorithm. In turn, SR also provides a promising way to implement TE in multimedia networks. In the next subsection, several TE approaches are introduced.

B. Traffic Engineering (TE)

TE research focused mainly on control of multimedia traffic so that network resources can be efficiently utilized. In a traditional IP network, often the shortest path algorithm is used to implement TE. However, it could lead to congestion on bottleneck links because if an algorithm such as OSPF is used to compute the path, quality of service (QoS) is not taken into account. MPLS as a new TE paradigm was also proposed, which can effectively achieve the goal of TE. Regrettably, due to the complexity of MPLS signaling protocol and instruction labeling, it was not widely adopted. Instead, the SR mechanism has become a promising paradigm in TE [31],[32] mainly through simplifying the signaling protocol stack, while retaining some of the advantages of MPLS. Diverse algorithms for TE based on a SR strategy have been proposed. Either online or offline, these algorithms improve network performance in terms of bandwidth utilization or reducing end-to-end transmission delay.

In [33], authors proposed DEFO, an algorithm able to translate high-level goals expressed in a language close to a natural one into compliant network configurations. In [32], offline and online algorithms able to minimize the maximum link utilization in an SR network are proposed. In [31] the SR paradigm is exploited to increase the scalability of TE solutions based on SDN. The use of SR allows definition of per-flow instructions at the border of the network, reducing both numbers of instructions and operations that transit node have to store and perform. A similar approach is proposed in [34], where SR is employed to provide load balancing in a SDN network.

Unicast and multicast are two common approaches used in TE. The authors proposed an efficient unicast routing algorithm based on SR technology in [34]. Since many previous works focused mainly on unicast TE with SR, this paper studies multicast TE with SR. This approach is more complex, but has potential to save larger amounts of resources than unicast-based solutions.

C. Multicast Routing

Multicast is an efficient mechanism to disseminate popular multimedia content to multiple users. Here, several traditional algorithms are introduced for multimedia TE. The Shortest Path Tree (SPT) [35] algorithm is one of the simplest algorithms, which is constructed by finding the shortest paths from a source node to a set of destination nodes. The Widest-Shortest Path Tree (WSPT) is an extension of SPT, which constructs the multicast tree by considering the available bandwidth in the path. The object of WSPT is to implement the communication with the largest amount of residual bandwidth. The Steiner Tree (ST) algorithm [36] is another common multicast algorithm, which involves a combinatorial optimization problem. In fact, the SPT algorithm is a special case of the ST algorithm. Similar with WSPT, the Widest-Steiner Tree (WST) algorithm is an extension of ST, which also considers the path with the most available bandwidth. Additionally, the literature [8] proposed a promising centralized approach based on SDN, bringing efficiency and flexibility to the network. It also analyzed both distributed and centralized approaches based on the multicast content popularity. If one could reduce the number of nodes maintaining forwarding rules, the scalability will improve.

Therefore, a new SR mechanism-based multicast routing model is proposed in this paper. In this model a QoE-driven multicast routing algorithm is combined with the SR strategy in a SDN network context. Our research outputs show that both higher scalability and improved efficiency of the multicast routing than those of other research solutions are achieved.

III. System Overview

A. SR Background

The Segment Routing (SR) is a promising TE paradigm which provides end-to-end communications through breaking up the traditional routing. SR can observably improve the network utilization and control the routing path flexibly by encoding route information into a list of segments, i.e., the Segment List (SL). The key feature of SR is that it adopts the source routing paradigm, which implies the routing path followed by a packet is determined and written to the packet header by the first switch of SR networks (called Ingress SR switch). For the source routing paradigm, Software Define Network (SDN) technology provides excellent solution with the SDN-controller capable of observing the whole network information easily. Therefore, the SDN-based SR architecture has been the most popular implementation of SR paradigm. In Fig. 1, we illustrated the SDN-based SR architecture, which was mainly composed of two parts, SDN controller and switches supporting SR protocol (referred to as SR switch). The SDN-controller is the brain of the whole architecture, responsible for collecting information of the whole network, calculating the optimal path, encoding the path into SL and sending the SL to the Ingress SR node and so on. While other SR switches routed the packet according to SL only instead of maintaining complex flow tables.

Each SR switch is univocally identified by means of a Segment IDentifier (SID) similar with an IP address or an MPLS label. There are two basic type of SIDs, Node-SID and Adjacency (Adj) SID. A Node-SID has global validity in the network domain, denoting identification of a network node. Adj-SID represents a local interface of a node and has local validity, i.e., only node that has emitted this identifier knows what the SID referred to. Node-SID and Adj-SID can all be directly assigned through the IGP extension routing protocol or the SDN-controller. In addition, when the SL size exceeds the capacity limitation in a SR switch, a SL cannot carry the entire Segment. Therefore, the whole SL needs to be divided into multiple SLs, which will be stuck together through a special identifier. This special identifier is called SSID and the node where the SSID is located is called *stick node*. The SDN-controller assigns SSID to *stick nodes* through pushing SSID to the bottom of upstream SL and connecting SSID to adjacent downstream SL. Unlike Adj-SID or Node-SID, stick segment do not identify routing path. When the packet is forwarded to the *stick node* according to the upstream SL of routing path, the SSID will be replaced with a new SL in the light of the association relationship between the SSID and the downstream SL, guiding the forwarding of packets in the downstream of routing path continuously.

Moreover, there are three types of segment operations to implement SR paradigm: PUSH operation, SWAP operation and POP operation. We skip the implementation details here and define three operations as follows. PUSH operation means when the packet enters the SR network, the Ingress SR node will insert a SL stack between the two-layer header and the IP header. Swap operation means when the packet is forwarded on the *stick node*, the SID on the top of the stack is the SSID.

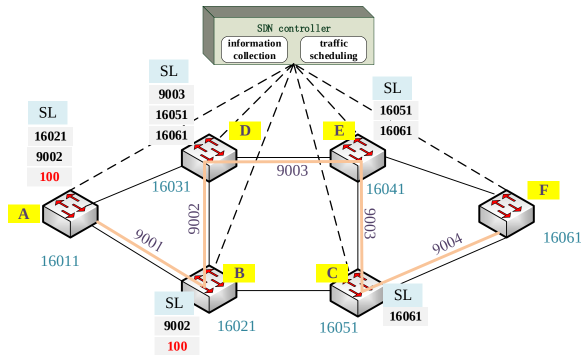


Fig. 1. Traffic engineering with SR

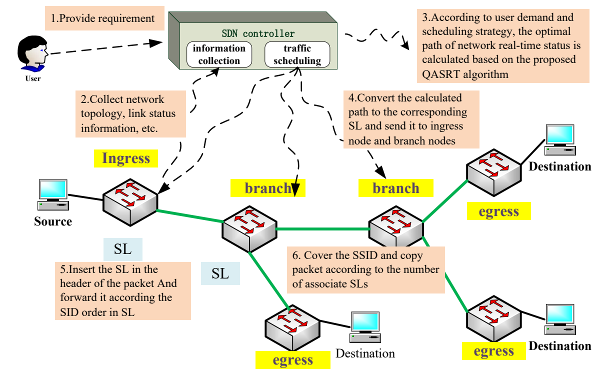


Fig. 2. Multicast routing strategy with SR machanism

According to the association relationship between the SSID and the SL stack, a new SL stack will be used to replace the SSID. POP operation means when the message is forwarded in the SR-TE tunnel, the SID at the top of the SL will be stripped off after the forwarding interface is searched according to the SID at the top of the SL.

To understand how the SR paradigm works better, we reported an example to explain shown in figure 1. (1) The controller collects topology information of the whole network, link status information, and assigns SIDs (or generate notifications to the controller on the device) for each node in the network; for instance the SID of node A is $\{16000\}$. (2) Assuming host A wants to access host F, it will be found many paths by default, such as ABCF,ADEF,ABCEF, etc. If there is no need to schedule the traffic, the flow can be forwarded just following the default shortest path. (3) If a particular path is required, such as a path with bandwidth greater than $8G$, and delay less than $30ms$, then the SDN-controller calculates the path meeting the requirements firstly, for instant, $A \rightarrow B \rightarrow D \rightarrow E \rightarrow C \rightarrow F$. This path corresponds to two SLs, $\{16021, 9002, 100\}$ and $\{9003, 16051, 16061\}$, among which, $\{16021\}$,$\{16051\}$ and $\{16061\}$ are the node-SIDs, $\{9002\}$ and $\{9003\}$ are the Adj-SIDs, and $\{100\}$ is the SSID, associating with the SL $\{9003, 16051, 16061\}$. (4) The SDN-controller assigns the SL $\{16021, 9002, 100\}$ (PUSH operation) to the ingress SR node A only as well as notifications node D the associate relationship between SSID $\{100\}$ and SL $\{9003, 16051, 16061\}$. (5) Ingress node A will forward packet to node B according to the top node-SID of SL, i.e., $\{16021\}$, and then pop up the top SID $\{16021\}$ (POP operation). Similarly, node B will forward the packet with SL $\{9002, 100\}$ to node D and pop up Adj-SID $\{9002\}$. At node D, it will swap SSID $\{100\}$ to SL $\{9003, 16051, 16061\}$ (SWAP operation) and continue to forward the packet according to this SL. When the packet arrives egress node F, it no longer carries any SIDs and it will be continued to be forwarded according the routing table.

B. QoE-driven Multicast Routing Paradigm with SR

We mainly focused on the problem that how to realize QoS multicast routing in SR paradigm. The successful solution to this problem is of benefit to show the advantage of the new multimedia TE paradigm well and further promote this development and application of SR scheme. In this paper, our research achieved two main goals, solving the optimal problem of QoS with multi-objective constrained and constructing multicast tree under the SR paradigm.

For the first problem, we can obtain more information than the distributed network by using the centralized and controllable advantage of the controller. However, the network status information is changing constantly in the real scenario, which implied the information obtained often inaccurate. It is still a difficult problem that how to use these inaccurate information to accurately perceive various of network information as well as propose an optimal routing algorithm that satisfy the need of multiple QoS. To solve these problems, a multi-objective and multi-constraint optimization problem model based on statistics and Bayes theory is proposed. The detailed content of this part will be shown in section IV.

For the second problem, we focused on the implementation of SR strategy at the branch nodes of multicast routing. From our previous introduction, we can see that SR is a source-routing paradigm, that is, the controller assigned the SL of a path only to the ingress node, while other nodes in the network just forward the packet directly according to the SIDs of SL, which implies that the multi-forwarding of the content cannot be completed at the branch node. Therefore, we proposed a multicast routing architecture based on the SSIDs, using the characteristics of SSIDs to complete the branching at branch nodes. At the same time, we also take into account of the heavy load at the branch node. In this paper, a branch-aware multicast tree construction method is proposed to realize the load balancing at the branch nodes, which will be given in detail in chapter V. Since the research on SR is still in its infancy, in order to better understand the following work, we first give an overview of our work .

The whole architecture is controlled by the controller, and all switch supports SR strategy. Compared with traditional MPLS-TE paradigm, the novel architecture mainly has following advantages:

1) Identifier allocation and distribution. Traditional MPLS network needs LDP and other protocols to synchronize and distribute identifier to each node. While SR paradigm no longer needs LDP protocol, the SIDs are all distributed uniformly by the controller, which greatly simplifies of device operation.

2) Identifier table build. In traditional MPLS, the building

of identifier list needs to distribute identifier based on the LDP protocol, which will lead to a large scale of forwarding table. Whereas, SR paradigm can complete the building of identifier table only with IGP protocol, which has high scalability and small scale, approximately the number of entries for N (number of node-SIDs, which generally is the number of entire network nodes) $+A$ (number of Adj-SIDs, which generally is the number of equipment interfaces).

3) Path identification and establishment. In the MPLS network, a LSP is adopted to identify a path, which is manually established or uses LDP and RSVP-TE protocols step by step, which is very complex and difficult to maintain. In SR protocol, the path is made up of an ordered list of segment, and is encapsulated in a packet header. Therefore, the SR path no longer relies on complex signaling protocol (LDP or RSVP - TE), which build the path with hopping by hopping. All intermediate nodes in the network just forward the packet directly according to the SL, which makes the network be with more scalability.

In Fig. 2, an example is given to show how the SR paradigm works on multicast routing. The whole scheme is divided into six steps: (1) The user puts forward the requirement for network QoE, such as wanting to achieve stable transmission, or to obtain higher network speed, and so on. And these requirements will be told to the controller. (2) The controller collects status information of the whole network, such as bandwidth, delay, network jitter, etc. as well as the network topology. (3) According to user's requirement and scheduling strategy, the controller calculates the candidate k optimal path based on our proposed IQdR algorithm at first, which can implement QoE-driven routing with multiple constraints. Then it will further construct the multicast tree according our branch-aware multicast tree algorithm. (4) After that, the controller will convert the calculated path to the corresponding SL and assigned it to both ingress node and branch nodes. (5) At ingress node, the SL will be inserted in the header of the packet to be forwarded to realize SR scheme. (6) In the branch node, it will adopt a one-to-many approach, making SSID associated with multiple SLs at the same time. The branch node will copy this packets according to the number of associated SLs and insert the SL to each packet header. Then the multicast routing with SR architecture can be implemented well.

IV. Inaccurate Information-based QoE-driven Routing Algorithm (IQdR)

QoE support requires guarantee of end-to-end Quality of Service (QoS). This section introduces IQdR, the proposed routing algorithm which provides end-to-end QoS based on inaccurate information exchange.

A. Problem Formulation

Before presenting the details of the IQdR algorithm, we define the notations and the network model used in this paper. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote the network topology, where $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$ is the set of network nodes and $\mathcal{E} = \{(i,j)|v_i, v_j \in V\}$ is the set of links. Each link (i,j) between nodes v_i and v_j is associated with the following QoS metrics: delay $D_{i,j}$, delay jitter $J_{i,j}$, bandwidth $B_(i,j)$, packet loss rate $L_(i,j)$ and cost function $C_(i,j)$. A multicast tree $\mathcal{T}(s,\mathcal{D}), \mathcal{D} \subseteq \mathcal{V} - \{s\}$ is a sub-graph of $\mathcal{G}$ rooted at source s and reaching all destination set $\mathcal{D}$. Let $\mathcal{P}_{\mathcal{T}}(s,m)$ denote a path on $\mathcal{T}$ from source s to a destination $v_m \in \mathcal{D}$. The QoS multicast problem can be given as follows with the QoS requests set $Q = \{D_q, J_q, B_q, L_q\}$:

$$\min C\left(\mathcal{T}\left(s,\mathcal{D}\right)\right) = \sum_{(i,j)\in\mathcal{P}_{\mathcal{T}}(s,m)} C_{i,j} \tag{1}$$

$$s.t. \max_{m\in\mathcal{D}} \left\{ \sum_{(i,j)\in\mathcal{P}_{\mathcal{T}}(s,m)} D_{i,j} \right\} \le D_q \tag{1a}$$

$$\min_{m\in\mathcal{D}} \left\{ B_{i,j}, (i,j) \in \mathcal{P}_{\mathcal{T}}\left(s,m\right) \right\} \ge B_q \tag{1b}$$

$$\max_{m\in\mathcal{D}} \left(\sum_{(i,j)\in\mathcal{P}_{\mathcal{T}}(s,m)} J_{i,j} \right) \le J_q \tag{1c}$$

$$\max_{m\in\mathcal{D}} \left\{ 1 - \prod_{(i,j)\in\mathcal{P}_{\mathcal{T}}(s,m)} \left(1 - L_{i,j}\right) \right\} \le L_q \tag{1d}$$

The primary issue facing QoS routing is how to determine the routing criteria. Service QoS can be described by a variety of metrics including bandwidth, delay, jitter, and cost, etc. These QoS metrics are used to characterize the various requirements of the service and are also a description of the corresponding network state. QoS metrics can be divided into three categories based on the impact on the QoS characteristics of the entire path: addable metrics, multiplicative metrics, and concave metrics. These are defined as follows:

1) If the QoS characteristic of the path is the sum of the corresponding QoS characteristics of each link on the path, the metric is said to be additive, like the delay shown in formula $(1a)$.

2) If the QoS characteristic of the path is the product of the corresponding QoS characteristics of each link on the path, the metric is said to be multiplicable, such as the packet loss rate in formula $(1d)$.

3) If the QoS characteristic of the path is the minimum value of the corresponding QoS characteristic of each link on the path, the metric is said to be concave, such as bandwidth in formula $(1b)$.

The path that satisfies the QoS requirements of the traffic flow is not necessarily the shortest path in the traditional sense. Different QoS metrics and the number of QoS metrics can have a large impact on the complexity of the routing algorithm. For the concave metric, the network topology can be pruned before the algorithm is designed, and the link that does not satisfy the constraint is pruned. While the multiplicable metrics can be converted to additive metrics (the multiplication can be viewed as a simplification of addition). Thus, the challenges faced by QoS routing algorithms are primarily due to the additive metrics.

Assume each link l is appended with a k-dimensional vector $(w_1(l), \cdots, w_k(l))$, $w_i(l) \geq 0, i = 1, \cdots, k$, which is an additive QoS metric with form

$$w_i(p_{s,m}) \triangleq \sum_{l \in p_m} w_i(l), \quad i = 1, 2, \cdots, k$$

where, $p_{s,m} \in \mathcal{P}_{\mathcal{T}}(s, m)$. Then, p can be denoted as $p\left(w_1(p_m), w_2(p_m), \cdots, w_k(p_m)\right)$. The QoS multicast routing problem constructed by the formula (1) can be expressed in the following forms:

$$\min C\left(\mathcal{T}\left(s, \mathcal{D}\right)\right) = \sum_{(i,j) \in p_{s,m}} C_{i,j} \tag{2}$$

$$s.t. \max_{m \in \mathcal{D}} w_i(p_{s,m}) \leq Q_i, i = 1, 2, \cdots k$$

Formula from eq. (2) is a multi-objective optimization problem. In general, there is no global optimal solution, rather than a set of solutions, which are called Pareto optimal solutions. Give the definition that the mapping F maps the Pareto optimal solution to the Pareto layer in the QoS k-dimension metric space W^k with the form:

$$\begin{aligned} &F\left(P_{s,m}\left(w_1(p_{s,m}), w_2(p_{s,m}), \cdots, w_k(p_{s,m})\right)\right) \\ &= \left(w_1(p_{s,m}), w_2(p_{s,m}), \cdots, w_k(p_{s,m})\right) \end{aligned} \tag{3}$$

where, the metric space W^k has the form $W^k = W_1 * W_2 * \cdot * W_k$, $w_i(p_{s,m}) \in W_i, i = \{1, 2, \cdot, k\}$.

From eq. (3), mapping F maps network path P to a point in the k-dimensional QoS metric space $(w_1(p_{s,m}), w_2(p_{s,m}), \cdots, w_k(p_{s,m}))$. Therefore, we can study the relationship between the routing request and the Pareto layer in the space W^k to figure out whether the routing request is feasible, that is to say, discussing the feasibility of routing requests in space W^k. The QoS metric space is divided into three parts based on the Pareto layer: feasible area, infeasible area and NP complete area. When the routing request falls within the feasible area, there is definitely a Pareto best solution to cover the request (i.e. there is a path that satisfies the request). When the routing request falls within the infeasible area, there is definitely no path that can satisfy the request. When the routing request falls within the NP full area, it cannot be determined whether there is a path that can satisfy the request.

B. IQdR Algorithm Description

At present, most end-to-end QoS routing algorithms are proposed under the assumption that routing state information is accurate. These algorithms assume that the entire network state is accurately maintained at each router through the link state routing protocol, which in fact is often inaccurate. There are several reasons for the inaccuracy of the network status information, including the status update protocol itself, network aggregation, propagation delay of network status update information, and approximate calculation. Therefore, it is important to study the QoS routing algorithm based on inaccurate network state information. We focused on the description of the routing algorithm based on inaccurate network state information.

Considering the QoS routing problem based on inaccurate network state information, the probability method is used to construct the mathematical model. Therefore, the original QoS routing problem is transformed into problems with the probability as the optimization objective that each measure value satisfies the constraint. Assuming the optimal path is p^*, we have eq. (4).

$$\phi_i\left(p^*\right) \geq \phi_i\left(p_{s,m}\right), \forall p_{s,m} \in \mathcal{P}_{\mathcal{T}}(s, m) \tag{4}$$

where,

$$\phi_i\left(p_{s,m}\right) \triangleq \Pr(w_i(p_{s,m}) \leq Q_i),$$

and $\Pr$ denotes the probability.

Note that the optimization goal for eq. (4) is associated with the probability that each metric satisfies the constraint. Therefore, the problem is essentially a discrete two-objective optimization problem, which is NP-hard. The natural idea to solve this problem is to calculate $\phi_i\left(p_{s,m}\right)$ first in the network graph expansion process, and then use the known linear or nonlinear search technique to find feasible paths based on $\phi_i\left(p_{s,m}\right)$.

Then, assume $w_i(l_e)$ are non-negative random variables and have mean and variance $\sigma_i^2(l_e)$, $\sigma_i^2(l_e)$, respectively. Since these statistics are relatively stable, they can be broadcast at longer intervals. We adopt a hypothetical class-based network state update mechanism, that is, the QoS metrics on the link l are divided into equal-sized intervals. When the current value of the QoS metric exceeds a certain class boundary, the state update mechanism is triggered, and the broadcast values of the QoS metrics w_i are represented by $w_i^b(l_e)$. We do not make specific requirements for the form of the probability distribution function of $w_i(l_e)$; it is only assumed that the distribution function is continuously differentiable. According to the central limit theorem, as the number of hops of the path $p_{s,m}$ increases, $w_i(p_{s,m}$ tends to a normal distribution with mean of

$$\mu_i(p_{s,m}) = \sum_{l_e \in p_{s,m}} \mu_i(l_e) \tag{5}$$

and variance of

$$\sigma_i^2(p_{s,m}) = \sum_{l_e \in p_{s,m}} \sigma_i^2(l_e) \tag{6}$$

The random variable obeys a normal distribution of the mean μ and the variance σ^2 falls within the interval $(\mu - 3\sigma, \mu + 3\sigma)$ with a probability of more than 99%. Therefore, when $\mu_i(p_{s,m})$ and $\sigma_i^2(p_{s,m})$ are obtained, we take $\mu_i - 3\sigma_i$ as the lower bound of $w_i(p_{s,m}$ and $\mu_i + 3\sigma_i$ as the upper bound of $w_i(p_{s,m}$. Since these boundaries are closely related to the variance, they are called *variance bounds*. In this way, the approximate distribution range of $w_i(p_{s,m}$ in the QoS metric space can be obtained according to the variance bound. Let $w_i^b(l_e)$ represent the last broadcast value of $w_i(l_e)$, and $w_i^b(p_{s,m})$ be the last broadcast value of $w_i(p_{s,m})$. We have,

$$\begin{aligned} w_i(p_{s,m}) &= \sum_{l_e \in p_{s,m}} w_i(l_e) \\ w_i^b(p_{s,m}) &= \sum_{l_e \in p_{s,m}} w_i^b(l_e) \end{aligned} \tag{7}$$

For link l_e, the Link Inaccuracy Degree (LID) $Ina(l_e)$ relative to the QoS metric w_i is defined as

$$Ina_i(l_e) \triangleq \frac{\left|w_i(l_e) - w_i^b(l_e)\right|}{w_i(l_e)} \tag{8}$$

Corresponding to path $p_{s,m}$, the Path Inaccuracy Degree (PID) $Ina(p_{s,m})$ relative to the QoS metric w_i is defined as

$$Ina_i(p_{s,m}) \triangleq \frac{\left|w_i(p_{s,m}) - w_i^b(p_{s,m})\right|}{w_i(p_{s,m})} \tag{9}$$

Then we have

$$Ina_i(p_{s,m}) \leq \max Ina_i(l_e) \tag{10}$$

The inequality from eq. (10) gives the bound of $Ina_i(p_{s,m})$, and next we focus on the bound of $w_i(p_{s,m})$.

Let $\max Ina_i(p_{s,m}) = \max\left(Ina_i(l_e)\right), l_e \in p_{s,m}$, so the inequality is further derived as follows:

If $\max Ina_i(p_{s,m}) < 1$, we have:

$$\frac{w_i^b(p_{s,m})}{1 + \max Ina_i(p_{s,m})} \leq w_i(p_{s,m}) \leq \frac{w_i^i(p_{s,m})}{1 - \max Ina_i(p_{s,m})} \tag{11}$$

If $\max Ina_i(p_{s,m}) \geq 1$, we have:

$$\frac{w_i^b(p_{s,m})}{1 + \max Ina_i(p_{s,m})} \leq w_i(p_{s,m}) \tag{12}$$

Inequalities from eq. (11) and eq. (12) give the bounds of $Ina_i(p_{s,m})$ in different cases. It is not difficult to see that in order to determine the specific scope of these boundaries, we must first determine $\max Ina_i(p_{s,m})$.

We rewrite eq. (8) as follows:

$$Ina_i(l_e) = \left|1 - \frac{w_i^b(l_e)}{w_i(l_e)}\right| \tag{13}$$

It can be seen that when $w_i^b(l_e) > w_i(l_e)$ and $w_i(l_e)$ has a minimum value, $Ina_i(l_e)$ reaches an extreme value, denoted by Λ; when $w_i^b(l_e) \leq w_i(l_e)$ and $w_i(l_e)$ has a maximum value, $Ina_i(l_e)$ also reaches an extreme value, denoted by Ω. Since $w_i(l_e)$ cannot be obtained, we conservatively take the larger of Λ and Ω as the maximum link inaccuracy of link $w_i(l_e)$, i.e.

$$\max Ina_i(p_{s,m}) = \max\{\Lambda, \Omega\} \tag{14}$$

For any $w_i^b(l_e)$, it can be determined which class interval $w_i^b(l_e)$ is in. Assuming that $w_i^b(l_e)$ is in interval $[\Delta_\Lambda^e, \Delta_\Omega^e]$, the network status information will be updated when $w_i(l_e)$ exceeds the interval. Therefore, the upper and lower bounds of $w_i(l_e)$ are essentially Δ_Ω^e and Δ_Λ^e. Thus, Ω and Λ can be determined as follows:

$$\Omega = \max\left\{\frac{\Delta_\Omega^e - w_t^b(l_e)}{\Delta_\Omega^e}\right\}, \quad e = 1, 2, \cdots, \eta \tag{15}$$

$$\Lambda = \max\left\{\frac{w_t^b(l_e) - \Delta_\Lambda^e}{\Delta_\Lambda^e}\right\}, \quad e = 1, 2, \cdots, \eta \tag{16}$$

Substituting eq. (15) into eq. (16), we have

$$\sum_{l_e \in p_{s,m}} \Delta_\Lambda^e(l_e) \leq w_i(p_{s,m}) \leq \sum_{l_e \in p} \Delta_\Omega^e(l_e) \tag{17}$$

Therefore, by integration eq. (11), eq. (12), and eq. (17), the final link boundary of $w_i(p_{s,m})$ is obtained. Let $w_{min}(p_{s,m})$ denote the lowest bound of $w_i(p_{s,m})$ and $w_{max}(p_{s,m})$ denote the upper bound of $w_i(p_{s,m})$. After determining the upper and lower bounds of $w_i(p_{s,m})$, the next step is to use these bounds to determine the feasible probability of path $p_{s,m}$.

First, we give the following definitions:

Definition 1: For two random vectors $\gamma(\alpha_1, \alpha_2, \cdot, \alpha_k)$ and $\lambda = (\beta_1, \beta_2, \cdot, \beta_k)$, if all the elements in γ are contained in λ, then define vector γ cover vector λ, denoting $\gamma \prec \lambda$.

Definition 2: Use $\psi\left(\gamma \prec \lambda\right)$ to represent the probability that the random vector $\gamma(\alpha_1, \alpha_2, \cdot, \alpha_k)$ covers $\lambda = (\beta_1, \beta_2, \cdot, \beta_k)$ denoting

$$\psi\left(\gamma \prec \lambda\right) \triangleq \min\left\{\Pr(\alpha_i \leq \beta_i)\right\}, i = 1, 2, \cdots, k \tag{18}$$

Then, for edge constraint vector $w(w_1(p_{s,m}), w_2(p_{s,m}), \cdot, w_k(p_{s,m}))$ and routing requirement vector $Q(Q_1, Q_2, \cdot, Q_k)$, the probability of feasible solution can be conservatively calculated as in eq. (19).

$$\psi\left(w \prec Q\right) \triangleq \prod_{i=1}^{k} \Pr(w_i \leq Q_i), i = 1, 2, \cdots, k \tag{19}$$

Next the feasible probability of the path phase $p_{s,m}$ for the routing request Q can be determined as in eq. (20).

$$\begin{aligned}\psi\left(w \prec Q\right) &\triangleq \prod_{i=1}^{k} \Pr(w_i \leq Q_i)\\ &= \phi_i(p_{s,m}) \cdot \phi_Q(p_{s,m}), i = 1, 2, \cdots, k\end{aligned} \tag{20}$$

Eq. (18) and eq. (19) give the feasible probability calculation methods, which are both dependent on the calculation of $\phi_i(p_{s,m})$. For path $p_{s,m}$, since the network status information is inaccurate, even when $w_i^b(p_{s,m})$ is less than Q_i, it is not certain that the current value of $w_i(p_{s,m})$ must be less than Q_i. Because $w_i(p_{s,m})$ is a random variable that follows a normal distribution, and $w_i^b(p_{s,m})$ is actually just a sample of $w_i(p_{s,m})$. Based on the distribution of $w_i(p_{s,m})$ and the boundaries imposed on it, we determine $\phi_i(p_{s,m})$ as follows.

$$\phi_i(p_{s,m}) = \begin{cases} 0, & Q_i < w_{min}(p_{s,m}) \\ 1, & Q_i \geq w_{max}(p_{s,m}) \\ F(Q_i), & w_{min}(p_{s,m}) \leq Q_i < w_{max}(p_{s,m}) \end{cases} \tag{21}$$

where $F(x)$ is approximated as in eq. (22).

$$F(x) = \frac{\int_{w_{\min}(p_{s,m})}^{x} \exp\left\{-\frac{(t-\mu_i(p_{s,m}))^2}{2\sigma_i^2(p_{s,m})}\right\} dt}{\int_{w_{\min}(p_{s,m})}^{w_{\max}(p_{s,m})} \exp\left\{-\frac{(t-\mu_i(p_{s,m}))^2}{2\sigma_i^2(p_{s,m})}\right\} dt} \tag{22}$$

Next, we can utilize a known linear or nonlinear search algorithm to search for the path that satisfies the feasible probability threshold greater than the feasible probability of routing request, and get the candidate paths. Overall, the QoS algorithm for inaccurate network state information based on probabilistic methods mainly achieves the following two objectives:

A) Generate a candidate path;

B) Determine the feasible probability of the candidate path.

Repeat these two processes until one of the following conditions is met:

1) The feasible probability of a candidate path is 1;
2) The candidate path cannot found again;
3) The search cost exceeds a certain threshold.

Since finding all Pareto optimal paths is a NP-hard problem, we will generate the first K optimal paths here and then construct the multicast tree based on the first K optimal paths in the next section.

V. Building the Branch-aware Multicast Tree

This section introduces the Branch-aware Multicast Tree algorithm (BaMT) used to build the multicast tree. We mainly focus on two problems: implementation of multicast routing with SR strategy at the branch nodes and selection of branch nodes to realize load balancing in the whole network. Our proposed algorithm will solve these two problems in three steps: pruning process, establishment of the critical node graph and remaining branch path selection.

Pruning Process: The process of pruning is performed mainly based on the $K * |D|$ shortest paths selected in last section. There are K optimal candidate paths to be selected for each destination. As long as K is reasonably selected, we believe that the paths not included in the $K * |D|$ candidate paths are not appropriate in terms of QoS. Prune these paths and we can get a candidate path topology graph to generate the candidate multicast tree. In Fig. 3, we show an example to illustrate this process. It consists of a source node s and four destination nodes set $D = d_1, d_2, d_3, d_4$ denoted by solid dot. Assume that the source node needs to send data to four destination nodes by multicast. First, we calculate four (assuming $K = 4$) candidate paths to each destination node according to the proposed IQdR algorithm; this has been marked with color lines in Fig. $3(a)$. At the same time, in Fig. $3(b)$, we have also showed separately four candidate paths to each destination for a clearer presentation. Then we prune the link which is not included in the candidate paths and generate a new candidate path topology shown in Fig. $3(c)$.

Establishment of Critical Node Graph: Before presenting the detail process of this step, the weighted graph of each candidate link and node in Fig. 1(b) is generated. Usually, weights of links are used in routing when considering paths selection and constructing a multicast tree. In fact, weights of nodes are also critical factors in constructing a multicast tree, especially considering branch node selection. In our algorithm, node weight is taken into account to reflect the load of nodes. This step is based on the premise of K-optimal candidate paths, that is, all the paths to be selected are the K candidate paths to each destination node.

Let ς_i denote the weight value of node v_i. We have eq. (23).

$$\varsigma_i = \frac{\sum\limits_{m \in D} \pi_{m,i}^v}{\sum\limits_{m \in D} \pi_m^v} \tag{23}$$

where, π_m is the total number of candidate paths from source node s to destination node m and $\pi_{m,i}^v$ is a boolean variable whose value is 1 if there is at least one candidate path with destination node m passing through node v_i and otherwise is 0. This destination is more beneficial to select optimal paths for balancing different destinations. The destination of $\pi_{m,i}^v$ indicates that for the same destination node m, if there are multiple shortest paths through v_i, it is only calculated once. Then $\pi_{m,i}^v$ is given by eq. (24).

Algorithm I Establishment of the Critical Node Graph

1. Based on K candidate paths for each destination node generated by IQdR algorithm, generate the candidate graph $BT = (S, D)$ according to the pruning process.
2. For each node $v \in \mathcal{V}$
3. Compute the node weight $\varsigma_i = \frac{\sum_{m \in D} \pi_{m,i}^v}{\sum_{m \in D} \pi_m^v}$
4. For each node$e \in \mathcal{E}$
5. Compute the link weight $\varepsilon_i = \frac{\sum_{m \in D} \pi_{m,i}^e}{\sum_{m \in D} \pi_m}$
6. Established the critical node graph $WT = (WV, WE)$
7. Return WT

$$\pi_{m,i} = \begin{cases} 1, & \sum\limits_{k=1}^{K} P_m^k v_i \geq 1 \\ 0, & others \end{cases} \tag{24}$$

in which, $P_m^k v_i$ represents whether the k-th candidate path with destination m passes through node v_i.

Eq. (23) reflects the importance of a node in the candidate topology graph. Nodes with a larger value of ς_i are usually critical nodes related to all destination nodes and more likely to be selected to construct the multicast tree.

Similarly, let ε_i denote the weight value of edge e_i. We have eq. (25).

$$\varepsilon_i = \frac{\sum\limits_{m \in D} \pi_{m,i}^e}{\sum\limits_{m \in D} \pi_m}. \tag{25}$$

where, $\pi_{m,i}^e$ is also a boolean variables denoting the number of candidate paths to destination node m that pass through link e_i. We have eq. (26).

$$\pi_{m,i}^v = \begin{cases} 1, & \sum\limits_{k=1}^{K} P_m^k e_i \geq 1 \\ 0, & others \end{cases} \tag{26}$$

where $P_m^k e_i$ represents whether the k-th candidate path to destination m passes through link e_i.

Next we obtain the weight topology graph of candidate paths. Based on this, the critical node graph WT can be established in Algorithm I. First, select the node with the largest weight value as the critical node in addition to the source node and the destination node. Second, calculate the link weight and node weight of each path from the source node to the critical node and select the path with the largest weight as the candidate path of multicast Tree. Then, based on the critical node, find out next critical node along the destination node directions. Repeat the above processes until the critical node connected directly to a destination node. It

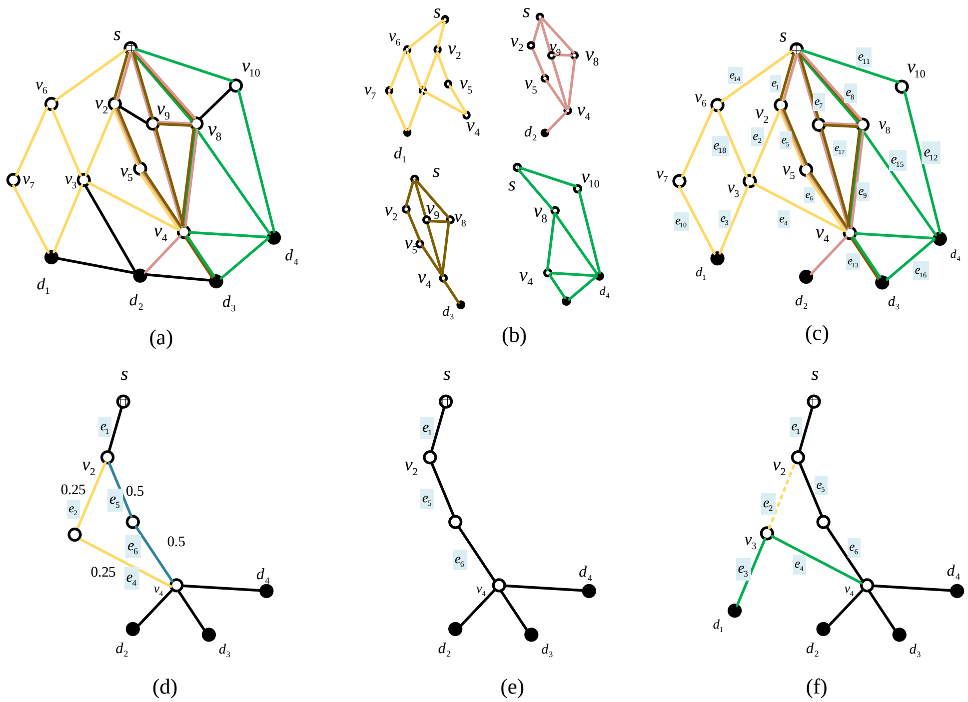


Fig. 3. Traffic engineering with SR mechanism

Algorithm II Building the Branch-aware Multicast Tree

1. **for** destination $d \in D$ and $d \notin WV$ **do**
2. **for** $x = 1$ to $|WV|$ **do**
3. Compute the cost of each path $WP(x, d)$
   $RC(x, d) = W_1 \varsigma_x + W_2 \varepsilon_{x,d}$
4. Find $WP(x, d)$ with the minimum value of $RC(x, d)$
5. Add the sub-path $WP(x, d)$ to WT
6. **end for**
7. **end for**
8. Return WT

is worth nothing that if there are multiple critical nodes with equal weights, the one closer to the source node is selected.

In order to further clarify the proposed algorithm, Fig. 3(c) and Fig. 3(d) illustrate an example. In Fig. 3(c), we calculate the weight of nodes using eq. (23) and eq. (25), and select the nodes with the largest weight as first critical nodes (i.e., node v_2). We continue to select the next largest weight node v_4 and add it to the critical node set. Since v_4 has been connected directly to the destination node, we can get the critical node set $\{v_2, v_4\}$. Next, we continue to select the link e with the largest weight from source point s to v_2 and v_2 to v_4. In Fig 3(c), there is only one path from s to v_1, namely edge e_1, and two path from v_2 to v_4, i.e., $v_2 - e_2 - e_4 - v4$ and $v_2 - e_3 - e_6 - v_4$ as shown in Fig 3(d). By calculating the weight sum of the two paths using eq. (25), we establish the critical node graph as shown in 3(e).

Remaining Branch Path Selection The goal of this step is to select the optimal path for the destination node that is not included in the multicast tree built in Algorithm II. Assuming that there are already K candidate paths to the destination node, we will compare each path and select the least cost one. Besides, in SR routing, the branch nodes are responsible for specific SR rules which are introduced in III-B. Compared with other nodes, the choice of branch nodes directly affects the scalability of the SR routing strategies. Therefore, we try to reduce the consumption caused by extra branch nodes. Let $RC(x, d)$ denote the cost of selecting the path of destination d through intersection node x in multicast Tree T. $RC(x, d)$ can be computed as the sum of the cost of candidate links $\varepsilon_{x,d}$ plus the cost of the branch node. We have eq. (27).

$$RC(x, d) = W_1 \varsigma_x + W_2 \varepsilon_{x,d} \tag{27}$$

where, W_1 and W_2 are the weights of branch node costs and link costs. Here, considering the branch nodes consume more resources in SR routing, we can set a large W_2 in order to account for its importance in the process of constructing the multicast tree. Assume a tree WT has been formed through the first two steps shown in Fig. 3(e), as indicated by a thick black line. Now we continue to join the new sub-path to the destination node d_1. According to the shortest path selected in the first step, there are two optional paths to be selected. The first one is the path $v_2 - e_2 - v_3 - e_3 - d_1$. According to the definition of $\varepsilon_{x,d}$, ς_x, we have $\varepsilon_{v2,d1} = \varepsilon_2 + \varsigma_3 + \varepsilon_3 = 0.25 + 0.25 + 0.25 = 0.75$, while $\varsigma_2 = 0.5$. The value of RC is obtained from eq. (27), $RC_1 = 0.5W_1 + 0.75W_2$. Similarly, when the second path is selected, i.e., $v_4 - e_4 - v_3 - e_3 - d_1$, the cost of sub-path and branch node is $\varepsilon_{v_4,d_1} = \varepsilon_2 + \varsigma_3 + \varepsilon_3 = 0.25 + 0.25 + 0.25 = 0.75$, $\varsigma_3 = 0.75$. Then we have $RC_2 = 0.75W_1 + 0.75W_2$. Here, we set $W_1 > W_2$ to reflect the importance of branch nodes in SR routing. By comparing the cost of this two candidate sub-paths, we choose the second path to be built into the multicast tree which is shown by the green solid line in Fig. 3(f).

VI. Performance Evaluation

In this section, we have compared the performance of the proposed algorithm with other traditional algorithms SPT, WSPT, ST and WST in terms of several aspects, including number of branch nodes, average rejection rate, network throughput, link utilization and algorithm generation time. We simulate the SR environment by employing the Mininet simulator and the open source code on Github [39].

In order to better reflect the demand for high QoE under a large number of multimedia content requests, we have compared the algorithms considering different multicast group size , number of multicast groups, and network size. The comparison of these three cases reflects well the performance of an algorithm in a multicast network. In these cases, the relevant parameters are as follows when considering the group sizes: network size is 100, the number of multicast groups is 10 and packet size is between 100MB-500MB. The number of multicast groups reflects the number of multicast group requests in a network. When a different number of multicast groups is considered, the relevant parameters are: network size is 100, the number of multicast groups is 30 and packet size is between 100MB-500MB. Similarly, for a different network size, the parameters are: multicast network size is 50, number of groups is 50, and packet size is 100MB-500MB.

In the following simulations, we have generated basic multimedia traffic and let critical nodes and links with larger weights have higher opportunity to consume more network resources in order to create a more realistic multimedia network environment. We set $W1 > W2 > 1$ to reflect the importance of branch node. From the weight functions defined in eq. (23) and eq. (25), it can be noted that the larger value of K, the fewer candidate paths. In this case, the weight value of all edges and points are larger, making it difficult to pick out critical nodes and links. On the contrary, if K value is too large, the weight of edges and points will become closer, and too many key nodes and edges will be selected, which will increase the complexity of the algorithm. Therefore, a moderate K value is crucial. We will select K values according to the different topology sizes.

The number of branch nodes is an important metric of a multicast algorithm, which also attributes to the TCAM deficiency and scalability problems in SR. The number of branch nodes influences the cost to maintain SR rules for routing in our novel multicast architecture as well as the scalability of a multicast network. Therefore, in Fig. 4, we compare the number of branch nodes in each algorithm under different conditions. It can be seen that the number of branch nodes increases with either multicast group size, network size or request number increase. This shows that the branch node number is the most intuitive response of a multicast routing algorithm, which also reflects the adaptability of each multicast routing algorithm. Fig. 4 shows that our proposed algorithm has a fewer number of branch nodes, which means our algorithm is more efficient in terms of reducing network cost and increases network scalability. Fig 4(b) shows that the relationship between branch node number and number of multicast group requests is not strictly linear. This is

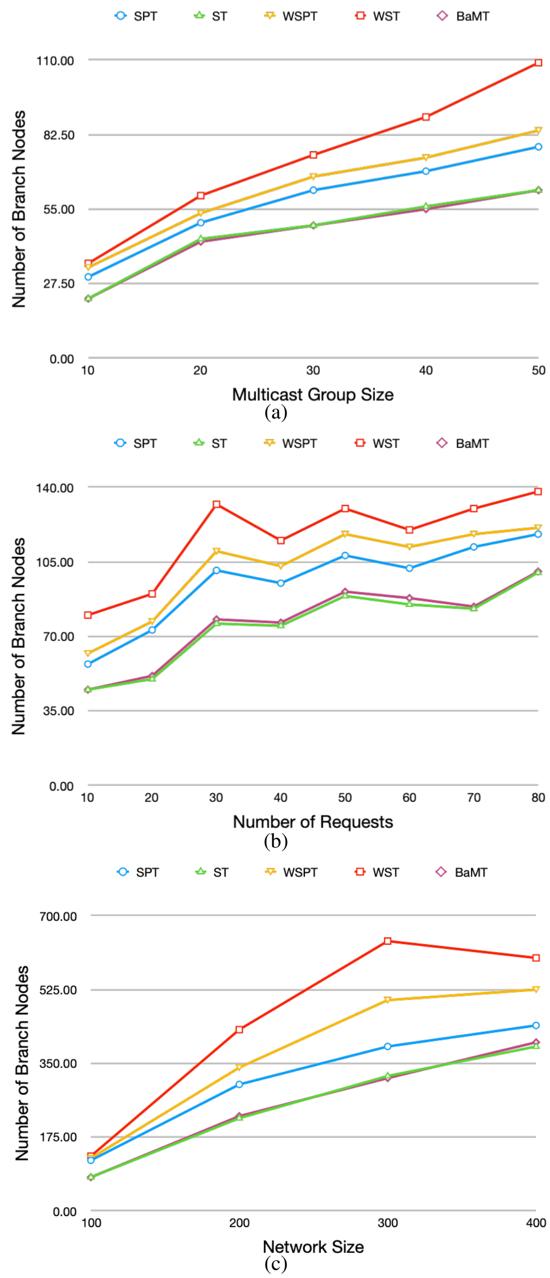


Fig. 4. Algorithm Time with (a) Group number $= 10$, Network size $= 100$; (b) Group size $= 30$, Network size $= 100$; (c) Group size $= 30$,Group number $= 50$.

because the branch node number depends not only on the number of multicast group requests but also on the network topology. With the increase in the number of multicast groups, more critical nodes with high degree will be included in several multicast groups simultaneously, which leads to a temporary decrease in the branch node number Our algorithm does not necessarily have a minimum number branch nodes, because we need to consider the QoE performance at the same time. By comparing the relevant QoE metrics across different algorithms, the advantages of our proposed algorithm are demonstrated.

Fig. 5 shows the average rejection rate, which is also an indication of whether a network can meet user QoE.

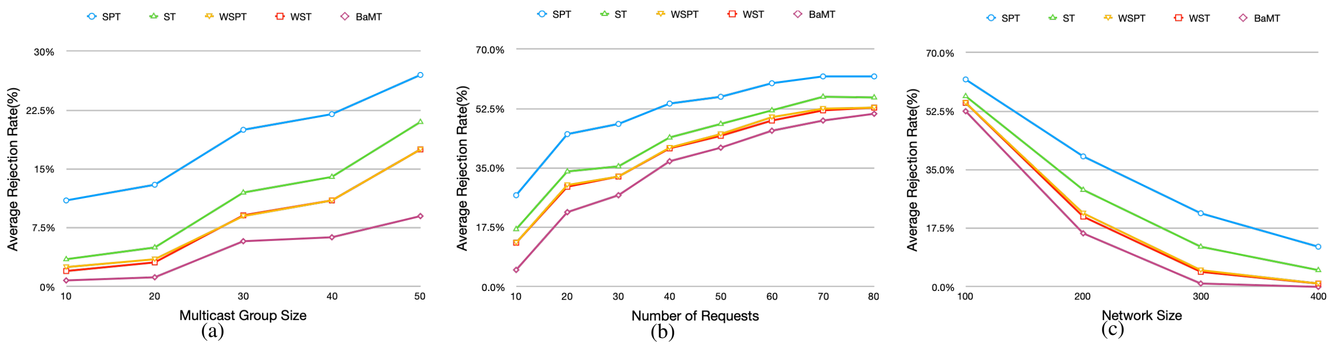


Fig. 5. Average Rejection Rate with (a) Group number $=10$, Network size $=100$; (b) Group size $=30$, Network size $=100$; (c) Group size $=30$,Group number $=50$.

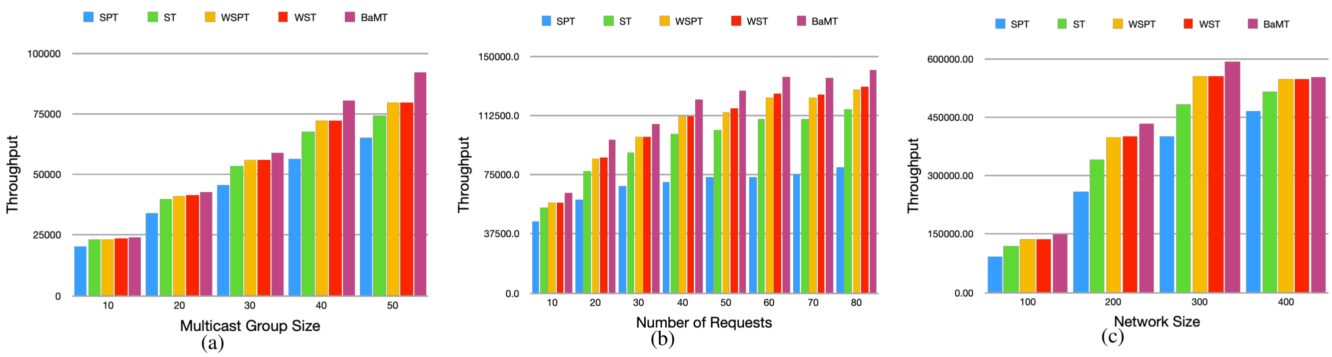


Fig. 6. Average Network Throughput with (a) Group number $=10$, Network size $=100$; (b) Group size $=30$, Network size $=100$; (c) Group size $=30$,Group number $=50$.

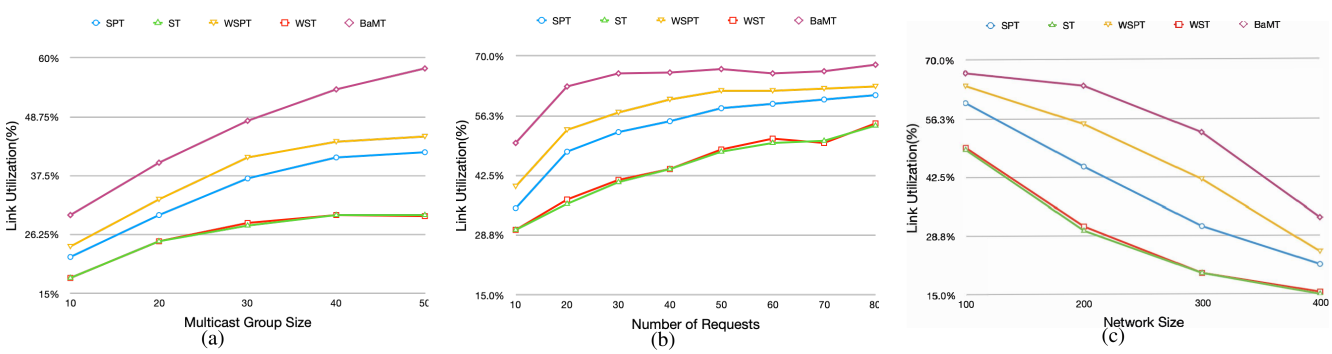


Fig. 7. Link utilization with (a) Group number $=10$, Network size $=100$; (b) Group size $=30$, Network size $=100$; (c) Group size $=30$,Group number $=50$.

The average rejection rate shows the ratio between rejected service requests and total traffic requests in the multimedia network. A multicast algorithm with high rejection rate has low robustness, leading to a bad experience for the multimedia user. Fig. 5(a) shows the average rejection rate under different multicast group sizes. It can be seen that the larger the multicast group size is, the higher the average rejection rate, because a multicast group with large size can increase the possibility of including a bottleneck link. Fig. 5(b) shows the various patterns of user requests. It can be seen the request rejection rate will increase with the multimedia traffic getting larger, which is easy to understand. Fig. 5(c) shows the results in terms of the average rejection rate when the network size varies. The larger the network size is, the lower the average rejection rate. In addition, it can be seen that the rejection rate of our algorithm decreases rapidly with the increase of the network size. This also partly reflects that our proposed algorithm benefits from the scalability of SR mechanism. As can be seen from Fig. 5, the average network rejection rate of our algorithm is lower than that of other algorithms in different cases. Note that the average rejection rates of WSPT and WST are lower than that of SPT and ST algorithms in all cases because WSPT and WST algorithms take the link bandwidth availability into account. Following the simulation results analysis, recommendations can be given with regard to multimedia applications. Improved multimedia service quality can be obtained by increasing the network scale, avoiding the traffic peak, or controlling the multicast group sizes.

Fig. 6 shows comparison results for the different multicast algorithms considering the network throughput with multicast

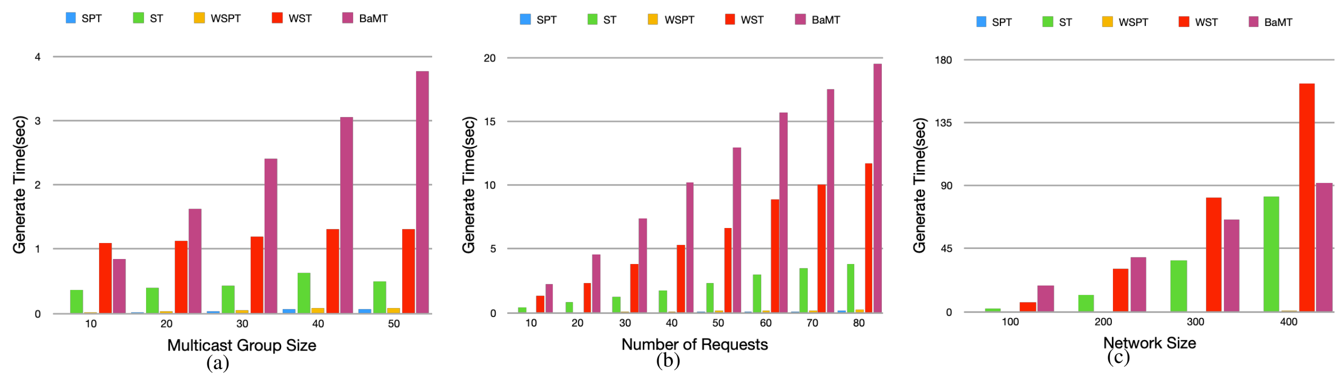


Fig. 8. Algorithm Time with (a) Group number $= 10$, Network size $= 100$; (b) Group size $= 30$, Network size $= 100$; (c) Group size $= 30$,Group number $= 50$.

group size, number of requests, and network size, respectively. The network throughput is the amount of satisfied bandwidth that is routed successfully as a function of the total amount of requested bandwidth that arrives at the network. In general, the average network throughput increases when the requested bandwidth increases. We can observe that the trends of the average rejection rate and the average network throughput are similar. In Fig. 6, SPT and ST algorithms have the lowest average throughput because the SPT and ST algorithms establish the shortest path between nodes only instead of taking the bandwidth availability of the network in account. Although a lower number of branch nodes are obtained, the network resources are not fully utilized. In Fig. 6(c), it can be seen that the network throughput of WSPT, WST and our algorithm no longer increases (even starts to reduce), as we have fixed the multicast group size, and the number of requests in the simulated case. The network has reached full capacity at the $size = 300$, and the network throughput continues to increase only changing the multicast group size or increasing the number of requests. In conclusion, our algorithm has a good performance in terms of both network throughput.

Fig. 7 illustrates link utilization, defined as the consumption of bandwidth by each link over the total default capacity of the network. In general, a higher rejection rate results in lower link utilization, because most of the requests are not routed. From the three figures of Fig. 7, we can see ST and WST algorithms have the lowest link utilization. This is because they are not optimal algorithms, and therefore cannot achieve full link utilization. Instead, our algorithm has the best performance in all cases because it considers the QoE of the whole network. Looking at the performance of the average rejection rate, throughput and link utilization rate, it can be concluded that our algorithm meets very well the requirements of QoE for multimedia applications when employing the SR mechanism. In figure 7, the lowest link utilization rates are for ST and WST algorithms, because they are not optimal algorithms, so it cannot achieve full utilization of the link.

Finally, Fig. 8 reports the average generation time of the five algorithms with different parameters for every request. Note that our algorithm generation time is longer than those of other algorithms because of extra inaccurate information calculation. But this makes sense when considering the improvement of the whole network's QoE performance shown in previous simulation results.

VII. CONCLUSION

This paper focuses on supporting QoE in multimedia networks based on a SR mechanism. Two problems were identified. The first one is how to implement fine-grained routing in a complex network environment. To solve it efficiently, a MOMC-based optimization model was put forward as there are multiple limitations on QoE-driven routing in such a complicated network environment, including bandwidth, delay, packet loss rate, throughput, etc.. Furthermore, considering the information obtained tends to be inaccurate, statistical theory is used to solve MOMC problems, proven to be NP-hard. The second one is how to construct a multicast routing tree based on the proposed new structure. In our solutions, the secondary source address routing was adopted at the branch node to realize the SR routing mechanism. Obviously, it will increase the load of the branch node greatly. Therefore, a further study about load balancing problem was made. The successful solution to this problem is of benefit to show the advantage of the new multimedia TE paradigm well and further promote this development and application of SR technology. Simulations-based testing showed how the proposed solution outperforms existing alternative solutions in terms of diverse QoS performance parameters.

REFERENCES

[1] A. Moldovan, I. Ghergulescu and C. H. Muntean, "VQAMap: A Novel Mechanism for Mapping Objective Video Quality Metrics to Subjective MOS Scale," in IEEE Transactions on Broadcasting, vol. 62, no. 3, pp. 610-627, Sept. 2016

[2] A. Martin et al., "Network Resource Allocation System for QoE-Aware Delivery of Media Services in 5G Networks," IEEE Transactions on Broadcasting, vol. 64, no. 2, pp. 561-574, June 2018.

[3] C. Xu, S. Jia, L. Zhong, H. Zhang and G.-M. Muntean, "Ant-Inspired Mini-Community-Based Solution for Video-On-Demand Services in Wireless Mobile Networks," IEEE Transactions on Broadcasting, vol. 60, no. 2, pp. 322-335, June 2014.

[4] C. Xu, S. Jia, M. Wang, L. Zhong, H. Zhang and G.-M. Muntean, "Performance-Aware Mobile Community-Based VoD Streaming Over Vehicular Ad Hoc Networks," IEEE Transactions on Vehicular Technology, vol. 64, no. 3, pp. 1201-1217, March 2015.

[5] L. Yu, T. Tillo and J. Xiao, "QoE-Driven Dynamic Adaptive Video Streaming Strategy With Future Information," IEEE Transactions on Broadcasting, vol. 63, no. 3, pp. 523-534, Sept. 2017.

[6] A. Hava, Y. Ghamri-Doudane, G.-M. Muntean and J. Murphy, ”Increasing User Perceived Quality by Selective Load Balancing of Video Traffic in Wireless Networks,” IEEE Transactions on Broadcasting, vol. 61, no. 2, pp. 238-250, June 2015.

[7] C. Xu, S. Jia, L. Zhong and G.-M. Muntean, ”Socially aware mobile peer-to-peer communications for community multimedia streaming services,” IEEE Communications Magazine, vol. 53, no. 10, pp. 150-156, Oct. 2015.

[8] J. Bukhari and W. Yoon, ”Multicasting in Next-Generation Software-Defined Heterogeneous Wireless Networks,” IEEE Transactions on Broadcasting, vol. 64, no. 4, pp. 915-921, Dec. 2018.

[9] L. Militano, M. Condoluci, G. Araniti, A. Molinaro, A. Iera and G.-M. Muntean, ”Single Frequency-Based Device-to-Device-Enhanced Video Delivery for Evolved Multimedia Broadcast and Multicast Services,” IEEE Trans. on Broadcasting, vol. 61, no. 2, pp. 263-278, June 2015.

[10] G. Araniti, P. Scopelliti, G.-M. Muntean and A. Iera, ”A Hybrid Unicast-Multicast Network Selection for Video Deliveries in Dense Heterogeneous Network Environments,” in IEEE Transactions on Broadcasting, vol. 65, no. 1, pp. 83-93, March 2019.

[11] N. McKeown, Software-defined networking, INFOCOM Keynote Talk, vol. 17, no. 2, pp. 30-32, 2009.

[12] D. Kreutz, F. M. V. Ramos, P. E. Verłssimo, C. E. Rothenberg, S. Azodolmolky, S. Uhlig, “Software-Defined Networking: A Comprehensive Survey,” Proceedings of IEEE, vol. 103, no. 1, pp. 14-76, Jan. 2015.

[13] C. Filsfils et al., “Segment Routing Architecture.” [Online]. Available: http://tools.ietf.org/html/draft-ietf-spring-segment-routing, Jan. 2018.

[14] Z. N. Abdullah, I. Ahmad and I. Hussain, “Segment Routing in Software Defined Networks: A Survey,” IEEE Communications Surveys & Tutorials, vol. 21, no. 1, pp. 464-486, Firstquarter 2019.

[15] Segment Routing Architecture. Available: https://tools.ietf.org/html/draftietf-spring-segment-routing-08

[16] Y. Kanizo, D. Hay, I. Keslassy, “Palette: distributing tables in software-defined networks,” in Proceedings of IEEE INFOCOM, 2013.

[17] J. Sheu and Y. Chen, “A scalable and bandwidth-efficient multicast algorithm based on segment routing in software-defined networking,” IEEE International Conf. on Communications (ICC), Paris, 2017, pp. 1-6.

[18] C. Filsfils et al., “Segment routing architecture,” draft-filsfils-spring-segment-routing-11, Feb 2017.

[19] Y. Desmouceaux, P. Pfister, J. Tollet, M. Townsley and T. Clausen, “6LB: Scalable and Application-Aware Load Balancing with Segment Routing,” IEEE/ACM Transactions on Networking, vol. 26, no. 2, pp. 819-834, Apr. 2018.

[20] H. Luo, H. Yu and L. Li, “Comments on ”Segment Shared Protection in Mesh Communication Networks With Bandwidth Guaranteed Tunnels,” in IEEE/ACM Transactions on Networking, vol. 15, no. 6, pp. 1616-1616, Dec. 2007.

[21] H. Roomi and S. Khorsandi, ”Semi-Oblivious Segment Routing with Bounded Traffic Fluctuations,” Electrical Engineering (ICEE), Iranian Conference on, Mashhad, 2018, pp. 1670-1675.

[22] S. Mukherjee and S. Roy, “Nearly-2-SAT Solutions for Segmented-Channel Routing,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 35, no. 1, pp. 128-140, Jan. 2016.

[23] T. Schuller, N. Aschenbruck, M. Chimani, M. Horneffer and S. Schnitter, “Traffic Engineering Using Segment Routing and Considering Requirements of a Carrier IP Network,” IEEE/ACM Transactions on Networking, vol. 26, no. 4, pp. 1851-1864, Aug. 2018.

[24] M. C. Lee and J. P. Sheu, “An efficient routing algorithm based on segment routing in software-defined networking,” Computer Network, vol. 103, pp. 44-55, Apr. 2016.

[25] D. Cai, A. Wielosz and S. Wei, “Evolve carrier ethernet architecture with SDN and segment routing,” Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014, Sydney, NSW, 2014, pp. 1-6.

[26] A. Cianfrani, M. Listanti and M. Polverini, ”Translating Traffic Engineering outcome into Segment Routing paths: The Encoding problem,” 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), San Francisco, CA, 2016, pp. 245-250.

[27] A. Giorgetti, P. Castoldi, F. Cugini, J. Nijhof, F. Lazzeri and G. Bruno, ”Path Encoding in Segment Routing,” 2015 IEEE Global Communications Conference (GLOBECOM), San Diego, CA, 2015, pp. 1-6.

[28] D. Lebrun, ”A Linux kernel implementation of Segment Routing with IPv6,” 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), San Francisco, CA, 2016, pp. 1019-1020.

[29] H. Xu et al., “Incremental deployment and throughput maximization routing for a hybrid SDN,” IEEE/ACM Trans. Netw., vol. 25, no. 3, pp. 1861-1875, Jun. 2017.

[30] F. Lazzeri, G. Bruno, J. Nijhof, A. Giorgetti and P. Castoldi, “Efficient label encoding in segment-routing enabled optical networks,” 2015 International Conference on Optical Network Design and Modeling (ONDM), Pisa, 2015, pp. 34-38.

[31] L. Davoli, L. Veltri, P. L. Ventre, G. Siracusano and S. Salsano, “Traffic Engineering with Segment Routing: SDN-Based Architectural Design and Open Source Implementation,” 2015 Fourth European Workshop on Software Defined Networks, Bilbao, 2015, pp. 111-112.

[32] R. Bhatia, F. Hao, M. Kodialam and T. V. Lakshman, ”Optimized network traffic engineering using segment routing,” IEEE Conference on Computer Communications (INFOCOM), Kowloon, 2015, pp. 657-665.

[33] R. Hartert et al., “A declarative and expressive approach to control forwarding paths in carrier-grade networks,” SIGCOMM Comput. Commun. Rev., vol. 45, no. 5, pp. 15-28, Aug. 2015.

[34] M. C. Lee and J. P. Sheu, “An efficient routing algorithm based on segment routing in software-defined networking,” Computer Networks, vol. 103, no. 5, pp. 44-55, July 2016.

[35] P. Narvaez, S. Kai-Yeung, and T. Hong-Yi, “New dynamic algorithms for shortest path tree computation,” IEEE/ACM Transactions on Networking, vol. 8, no. 6, pp. 734-746, December 2000.

[36] E. Aharoni and R. Cohen, “Restricted dynamic Steiner trees for scalable multicast in datagram networks,” IEEE/ACM Transactions on Networking, vol. 6, no. 3, pp. 286-297, June 1998.

[37] L. H. Huang, H. J. Hung, C. C. Lin, and D. N. Yang, “Scalable and bandwidth-efficient multicast for software-defined networks,” in IEEE Global Communications Conference (GLOBECOM), Austin, TX, Dec. 2014, , pp. 1890-1896.

[38] P. L. Ventre, M. M. Tajiki, S. Salsano, C. Filsfils, “SDN Architecture and Southbound APIs for IPv6 Segment Routing Enabled Wide Area Networks,” IEEE Transactions on Network and Service Management, vol. 15, no. 4, pp. 1378-1392, 2018.

[39] https://github.com/netgroup/SDN-TE-SR

[40] S. Knight et al. “The Internet Topology Zoo,” IEEE Journal on Selected Areas in Communications, Vol. 29, No. 9, October 2011

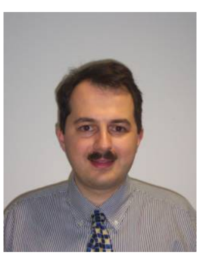
[41] B. M. Waxman, “Routing of multipoint connections,” IEEE Journal on Selected Areas in Communications, vol. 6, no. 9, pp. 1617-1622, Dec. 1988.

Shujie Yang received the Ph.D. degree in Institute of Network Technology from Beijing University of Posts and Telecommunications, Beijing, China, in 2017. He is currently a Postdoctoral with State Key Laboratory of Networking and Switching Technology. His major research interests are in the areas of wireless communications, and wireless networking.

Changqiao Xu (SM15) received the Ph.D. degree from the Institute of Software, Chinese Academy of Sciences, Beijing, China, in January 2009. He is currently a Professor with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China. He has authored/coauthored more than 100 technical papers in prestigious international journals and conferences. His research interests include wireless networking and multimedia communications. He was a Co-Chair of the IEEE MMTC Interest Group Green Multimedia Communications. He is a Senior Member of IEEE.

Lujie Zhong received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2013. She is currently an Associate Professor with the Information Engineering College, Capital Normal University, Beijing, China. She is also with the State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China. Her research interests include communication networks,computer system and architecture,and mobile Internet technology.

Jiahao Shen received the B.E. degree in computer science and technology from China University of Geosciences, Beijing in 2019. Now he is studying for his master degree in Beijing University of Posts and telecommuincations.



Gabriel-Miro Muntean (SM17) is an Associate Professor with the School of Electronic Engineering, Dublin City University (DCU), Ireland, and the Co-Director of the DCU Performance Engineering Laboratory. He has published over 350 papers in top-level international journals and conferences, authored 3 books and 18 book chapters, and edited 6 additional books. He has supervised to completion 20 Ph.D. students and has mentored ten postdoctoral researchers. His research interests include quality, performance, and energy saving issues related to multimedia and multiple sensorial media delivery, technology enhanced learning, and other data communications over heterogeneous networks. He is an Associate Editor of the IEEE TRANSACTIONS ON BROADCASTING, the Multimedia Communications Area Editor of the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS, and chair and reviewer for important international journals, conferences, and funding agencies. He is a Senior Member of IEEE and IEEE Broadcast Technology Society. He is the Project Coordinator for the EU-funded project NEWTON (http://newtonproject.eu).