Traffic Engineering and Manageability for Multicast Traffic in Hybrid SDN

Cheng Ren[1,2], Sheng Wang[1,*], Jing Ren[1], Xiong Wang[1]
[1]Key Lab of Optical Fiber Sensing and Communication, Education Ministry of China
University of Electronic Science and Technology of China
[e-mail: wsh_keylab@uestc.edu.cn, renjing@uestc.edu.cn, wangxiong@uestc.edu.cn]
[2] School of Electrical Engineering and Information, Southwest Petroleum University
[e-mail: rencheng@swpu.edu.cn]
*Corresponding author

Abstract

Multicast communication can effectively reduce network resources consumption in contrast with unicast. With the advent of SDN, current researches on multicast traffic are mainly conducted in the SDN scenario, thus to mitigate the problems of IP multicast such as the unavoidable difficulty in traffic engineering and high security risk. However, migration to SDN cannot be achieved in one step, hybrid SDN emerges as a transitional networking form for ISP network. In hybrid SDN, for acquiring similar TE and security performance as in SDN multicast, we redirect every multicast traffic to an appropriate SDN node before reaching the destinations of the multicast group, thus to build up a core-based multicast tree substantially which is first introduced in CBT. Based on the core SDN node, it is possible to realize dynamic control over the routing paths to benefit traffic engineering (TE), while multicast traffic manageability can also be obtained, e.g., access control and middlebox-supported network services. On top of that, multiple core-based multicast trees are constructed for each multicast group by fully taking advantage of the routing flexibility of SDN nodes, in order to further enhance the TE performance. The multicast routing and splitting (MRS) algorithm is proposed whereby we jointly and efficiently determine an appropriate core SDN node for each group, as well as optimizing the traffic splitting fractions for the corresponding multiple core-based trees to minimize the maximum link utilization. We conduct simulations with different SDN deployment rate in real network topologies. The results indicate that, when 40% of the SDN switches are deployed in HSDN as well as calculating 2 trees for each group, HSDN multicast adopting MRS algorithm can obtain a comparable TE performance to SDN multicast.

1. Introduction

Multicast communication is designed to deliver traffic from a source to all the destinations of a multicast group. It exploits a multicast tree instead of disjoint unicast paths to avoid inefficient usage of network resources such as the unnecessary traffic duplication [1]. Compared with unicast, multicast communication is able to effectively reduce overall bandwidth consumption in backbone networks by around 50% [2]. And a recent report [3] shows that 69% of Internet traffic is video, which is a perfect usage scenario for multicast and can greatly benefit from multicast delivery. Current IP multicast routing standards, such as PIM-SM, build the shortest-path tree between the source and every single destination in a group. However, it is difficult to construct an efficient tree and implement the traffic engineering, because each router calculates the path in a distributed manner based on the local information about the network. Moreover, since IP multicast makes no access control over the host, any host is qualified to send traffic to a group and able to subscribe to a multicast group and start receiving group traffic. Thus, this imposes a high security risk for the network.

Software defined networking (SDN) is a newly emerging network paradigm. A logically centralized controller makes all control decisions based on global vision on the network to enable greater intelligence especially in network algorithm design. With the rise of SDN, the SDN multicast comes along with much more advantages in contrast with IP multicast [4, 5]. Firstly, the centralized access control can be implemented in SDN, in order to alleviate IP multicast security concerns. Moreover, as opposed to the shortest path tree in IP multicast, flexible multicast tree can be constructed for SDN multicast to enhance TE performance.

However, migration to full SDN cannot be finished in one step, since full deployment of SDN has its own set of challenges. Network operators are reluctant to migrate from traditional network to full SDN basically for two reasons. One is that replacing all legacy network infrastructures with specialized SDN equipment requires huge capital expenditures. The other is that one-step SDN upgrade for the entire ISP network is impractical because the SDN technology is immature and potentially full of software vulnerabilities, not to mention the performance and security risks [6]. Given the statements above, a preferable choice arises out of incremental SDN deployment, that is Hybrid SDN (HSDN) [7]. In HSDN, the controller makes decisions only for SDN nodes, while the rest legacy nodes still follow the OSPF-like shortest path routing policy. To the best of our knowledge, in this paper we first deal with multicast issues in HSDN.

In HSDN, in order to acquire similar TE and security performance as in SDN multicast, for each multicast group, we firstly leverage unicast routing to redirect the multicast packets to an appropriate SDN node which serves as the core node, and afterwards route the packets to the destinations whereas the multicast routing is utilized. Thus, a core-based multicast tree is established which is essentially a routing problem. The concept of core-based multicast tree is first proposed in the Core Based Trees (CBT) protocol [8], whereby a single tree is built for a destination set regardless of the source node, thus to largely alleviate the scalability issue. In this paper, unlike the CBT, we define a multicast group to be one source node together with a set of corresponding destinations, which can be represented by a *(source, destinations)* pair. And in contrast to CBT, one significant difference is that we appoint one SDN node as the core node for each multicast group, namely a *(source, destinations)* pair, rather than one core node for a destination set as in CBT. Then, different

trees can be set up at the granularity of group other than the destination set, which can greatly enhance the TE performance for the network. Admittedly, this may bring accompanying negative influence on the scalability issues compared with CBT. In order to alleviate the adverse effects, we establish an additional constraint that any SDN node can act as the core node for a limited number of groups.

Thus, for multicast in HSDN, right on the core SDN node for each group, the centralized access control can be executed to alleviate security concern for multicast communication. In addition, since SDN node can enable path diversity by routing traffic to arbitrary outgoing links, we take full advantage of this SDN flexibility to create a total number of K trees for each group. On top of that, the usage of OpenFlow's group table select type makes it possible for the core SDN node to split up the unicast traffic originating from the source node of a multicast group onto the overall K trees, each with optimized splitting fraction, so as to further benefit TE performance of the network.

However, randomly picking SDN node serving as the core node for each group may lead to negative consequence such as network congestion. Furthermore, it is considered that the placement of the core SDN node per group should contribute to optimizing the routing paths between the source and destinations. And more than that, the SDN deployment rate plays an crucial role in effecting the core SDN node determination, that is to say, the SDN deployment rate is also a considerable issue which has a critical impact on the TE performance. For instance, in an extreme situation that a small hybrid network has only one SDN node, making all multicast traffic passes through the sole SDN node to gain the global control may make the links adjacent to the SDN node become overloaded, not to mention the resulting excessively long paths as well as the induced delay. And actually, different locations of upgraded SDN nodes may lead to different TE performance. In this paper, for simplicity, the static migration algorithm is leveraged [5] with respect to the SDN migration sequence issue. Last but not least, we also take into consideration the appropriate number of trees calculated for each group, namely the value of K. For a small K, the resulting TE performance may remain unsatisfactory. For a large K, it may impose a heavy workload on the controller, while seldom improving the TE performance of the network.

Then, there are *three challenges* we are confronted with. The *first challenge* is which SDN deployment rate is adequate to make the tradeoff between deployment budget and network performance. The *second challenge* is to select the appropriate SDN node acting as the core node for every multicast group. And the *third challenge* comes with the determination of sufficient number of trees calculated per group to get a desirable TE performance while inducing no overwhelming workload of the controller. The main contributions of this work are:

- We firstly formulate the routing problem as a node-link model to minimize the maximum link utilization, whereby we jointly determine an appropriate SDN node per group as the core node to which the multicast traffic is redirected utilizing unicast routing, as well as constructing multiple multicast core-based trees which are leveraged for the core SDN node to split the multicast traffic onto, thus to realize the manageability and load balance for multicast traffic in HSDN.
- Since it is difficult to find the solution to the node-link model in polynomial time especially for large scale topology, we propose another link-path model as well as developing the multicast routing and splitting (MRS) algorithm to obtain an approximate results efficiently.
- We conduct extensive simulations on real network topologies, and make evaluations of

several algorithms as well. The results explicitly indicate that, when 40% of the SDN nodes are deployed as well as splitting the multicast traffic onto a small number of trees for every multicast group, the proposed MRS algorithm can obtain comparable traffic distribution characteristics to SDN multicast. Not only that, MRS algorithm can also generate multicast trees with a little larger number of edges on average, which has a limited influence on the routing efficiency.

The rest of the paper is organized as follows: Section 2 discusses the related work and motivation. Section 3 presents the problem formulations. Section 4 is the elaboration of MRS algorithm. Section 5 presents several simulation results. We draw the conclusions in Section 6.

2. Related Works And Motivation

2.1 Related Work

The issues of traffic engineering for multicast traffic in full SDN have attracted extensive attention in the literature. In CastFlow [9] the authors propose a multicast clean-state approach on which the calculation of all possible routes from sources to group members is done in advance, aiming to reduce event delays. In [10] multiple backup multicast trees are set up and cached in the network controller for each multicast group, in order to switch trees with little packet loss and without duplicate packets delivery in OpenFlow network. Craig et al. implement traffic load balancing by adjusting the link weights for shortest path tress in SDN [11]. DYNSDM [12] proposes a network-layer multi-tree mechanism to reduce control plane complexity of ISP network, as well as improving TE performance by distributing traffic on multiple trees in SDN. In [13] the authors employ multipath multicast for robustness and load balancing to obtain video quality improvement specifically. Avalanche [4] leverages SDN to enable multicast in data center networks, so as to achieve highly efficient bandwidth utilization. However, none of these works have focused on the traffic engineering for multicast in HSDN.

With respect to constructing the multicast trees, there are mainly two types of architectures, source-based and core-based. In [14] multiple source-based multicast trees are constructed to facilitate multi-party video conferencing application. Huang et al. [15] propose a branch-aware Steiner tree and jointly minimize the numbers of the edges and the branch nodes for a single multicast tree in SDN, in order to address the issues on traffic engineering and flow table scalability. Not only that, they also take the link capacity constraint into consideration and design an approximation algorithm to solve the problem for multiple trees [1] and focus on the recovery cost [16]. Contrary to the source-based multicast tree building technique, CBT [8] firstly propose the core-based tree architecture, namely the shared tree algorithm, in order to alleviate the scalability issues faced by the source-based architecture. The tree includes a single node called core node, and all the traffic is firstly routed to the core node, then to every destination of the group [17, 18]. However there exists no polynomial time algorithm that can find the optimal core node of a multicast spanning tree [19]. In [20] the authors propose a novel core based algorithm which is used to solve a routing problem with several constraints, so as to establish a multicast tree. However, none of the papers deal with the construction of the multicast core-based trees in HSDN.

2.2 Motivation

In this paper, we focus on the multicast in HSDN and aim to acquire similar TE and security performance as the SDN multicast. Thus, we redirect traffic of every multicast

group to a core node using the unicast routing, then to every destination of the group following multicast routing, whereby a core-based multicast tree is constructed substantially. More importantly, we constrain the core node of every multicast group to be a SDN node to benefit TE performance as well as implementing traffic manageability. Based on the single SDN node, it is possible to realize access control, traffic measurement, firewall actions and middlebox-supported network services [6], in addition to gaining the dynamic routing control over the outgoing links of the SDN node. On top of that, in order to further enhance the TE performance, we build up a number of K trees per group from the core SDN node to the destinations. And the multicast traffic flows out of the core SDN node can be split among the K trees using the group table select type [12], since the group table select feature is fully supported by OpenFlow version 1.1 and above. With respect to the scalability issue, we restrain that every SDN node can act as the core node for a limited number of groups to alleviate the problem to some extent.

Specifically, for the desirable TE performance as well as the multicast traffic manageability in HSDN, there are three problems to deal with. Determining the sufficient SDN deployment rate becomes the *first challenge*. The *second challenge* comes with searching an appropriate SDN node acting as the core node for each group [21]. And the *third challenge* is to find a sufficient number of trees calculated per group to balance the computing resources consumption with network performance.

We illustrate the three challenges using **Fig. 1**. There are eleven routers in the topology and two multicast groups g_1 and g_2. Group g_1 consists of source node s_1 and two destinations d_1 and d_2. Group g_2 is comprised of source node s_2 and two destinations d_2 and d_3. The two groups have identical traffic demands at 20 Mbps. We assume that the capacities for link *(m, d_2)* and *(h, d_2)* are 15 Mbps and for all the rest links are 25 Mbps. For simplicity, all the link weights are set to be 1. **Fig. 1(a)** illustrates the shortest path trees for both group g_1 and g_2 in traditional IP network. The red solid lines indicate the multicast tree T_1 for g_1 and green dashed lines indicate the multicast tree T_2 for g_2. It leads to congestion with maximum link utilization of 160%. In order to optimize the TE performance, we upgrade the network to hybrid SDN.

For the first step, we upgrade single node b to be the SDN node. If the multicast traffic is not redirected to the core SDN node, which is expressed with core SDN node waypoint constraint in the following section, the multicast trees for the two groups are not changed. If we put the core SDN node waypoint constraint into effect and calculate one tree for each group, the corresponding trees are shown in **Fig. 1(b)**. Unfortunately, the maximum link utilization stays the same as **Fig. 1(a)**. Hence, it seems that deploying one SDN node in the network is not sufficient to gain the desirable TE performance, which exactly demonstrate that the first challenge of SDN deployment rate is definitely essential.

For the second step, we simultaneously upgrade three nodes *a, b* and c to be SDN nodes, together with the core SDN node waypoint enforcement. Note that it is still to calculate only one tree for each group. Two different core SDN node selection policies are executed separately. The first one is a random selection of the core SDN node for g_1 and g_2. We take node b serving as the core node for both two groups as an example. Apparently, this random selection policy makes no difference with **Fig. 1(b)** and leads to congestion again as shown in
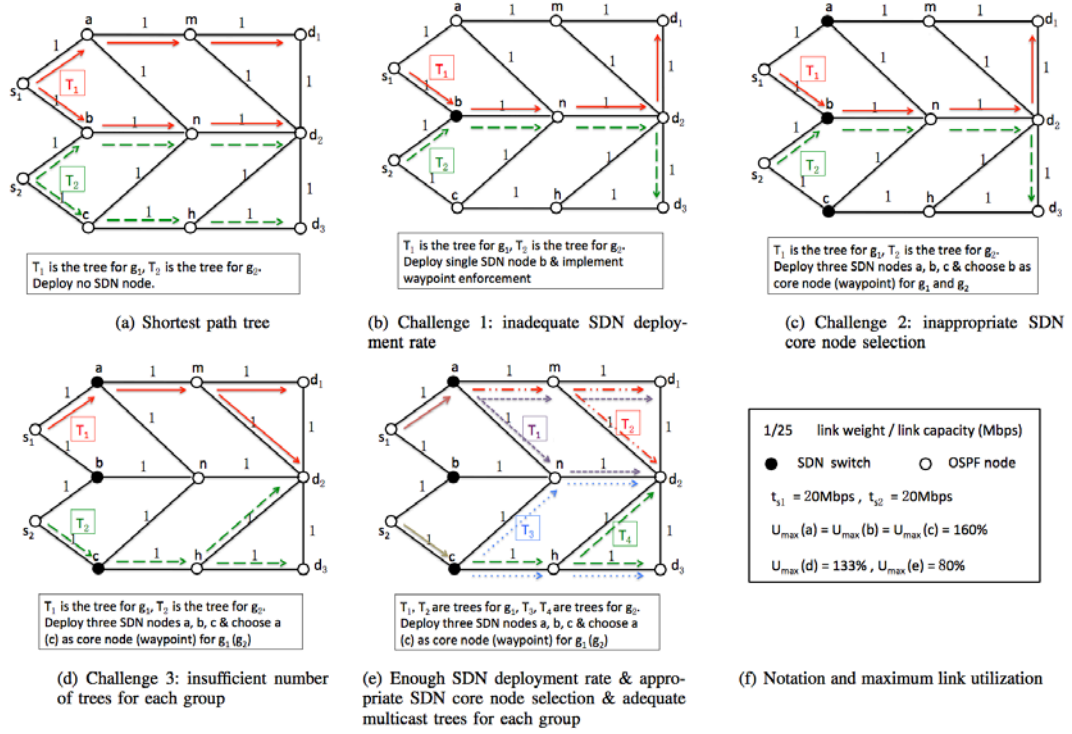
Fig. 1. An example of the three challenges

Fig. 1(c). The other one is an elaborate selection policy. Node *a* and c are chosen to be the core SDN node for g_1 and g_2 respectively shown in **Fig. 1(d)**. The maximum link utilization decreases to 133%. Therefore, an appropriate SDN core node determination for every group, which is the second challenge to deal with, is crucial to improve the TE performance. But still, the network congestion exists. Thus, it is reasonable for us to consider more trees, rather than only one, for every group to further benefit TE performance.

To take it one step further, we increase the number of trees calculated for each group in the third step. As shown in **Fig. 1(e)**, group g_1 (g_2) selects node *a* (*c*) as the core SDN node. And there are two trees calculated for each group. T_1 and T_2 (T_3 and T_4) are for group g_1 (g_2). For simplicity, every tree carries out 50% of the multicast traffic demand in each group. The resulting maximum link utilization finally decreases to 80% which demonstrates calculating sufficient trees for each group dose make sense to get a much better TE performance.

This example indicates that the inflexible routing policy may result in network disaster like congestion in both traditional network and HSDN. With respect to the realization of the unicast routing from the source to the core SDN node, Fibbing [22], Panopticon [23] and Telekinesis [24] can be the candidate methods discussed in our previous work [21], which we call as detouring routing policy in the rest of the paper.

3. Problem Formulation

We set the TE goal to be load balancing the network traffic, and take minimizing the maximum link utilization as the optimization objective. This is a common goal for WAN,

ISP and enterprise networks, as more traffic can be packed and processed with the same network infrastructures [7]. TE goal is met by establishing well-designed multicast trees.

Since the migration period of legacy network to full SDN will span several years, the SDN deployment rate and locations of upgraded SDN nodes are known and of certainty at a certain time during the long period, which can be a realistic situation for the network operators. Thus, given a fixed SDN deployment rate at a time ranging from 20% to 60% and the specific upgraded locations of SDN switches using static migration algorithm [5] in HSDN, we gradually increase the number of trees K calculated for each group (spanning from 2 to 10). Based on this, the determinations of the appropriate core SDN node per group and the optimized multicast traffic splitting fractions on the K trees are substantially routing problems. We manage to obtain the optimal maximum link utilization, whereby the corresponding exact SDN deployment rate and the precise value of K can be determined, and thus to indicates the accurate numerical relationships between the three challenges and the TE performance. In this section, we formulate the problem as a mixed integer programming (MIP) model to minimize the maximum link utilization using node-link model.

Table 1. Node Link Formulation $\mathbf{P_1}$

Parameter	Description	Variable	Description
V	The set of nodes.	$x_{ij}^{t_s}$	A binary variable to indicate whether link e_{ij} is on the path from original node s to core SDN node for traffic demand t_s.
S	The set of source nodes of all multicast groups, $S \subseteq V$.		
s	The source node of a multicast group, $s \in S$.		
D_s	The set of destination nodes in the multicast group with the source node s, $D_s = \{d_i, \forall d_i \in V, \forall s \in S\}$.	$y_{d,ij}^{t_s,k}$	A binary variable to indicate, for the kth tree of a group with traffic demand t_s, whether link e_{ij} is on the path from core SDN node to the destination node $d \in D_s$.
T	The traffic matrix. $T = \{t_s, \forall s \in S\}$.		
t_s	The traffic demand for multicast group $(s, D_s), \forall s \in S$.	$\beta_i^{t_s}$	A binary variable to indicate whether node i is chosen to be the core node of a group with traffic demand t_s.
SDN	The set of SDN nodes. $\forall c \in SDN, SDN \subseteq V$.		
E	The set of links.	$\varepsilon_{ij}^{t_s,k}$	A binary variable to indicate whether link e_{ij} is on the kth tree of a group with traffic demand t_s.
e_{ij}	The link from node i to node j.		
C_{ij}	The capacity of link $e_{ij} \in E$.		
K	The set of k trees calculated for every multicast group, $|K| = K$.	$f^{t_s,k}$	A decimal variable to indicate the fraction of traffic on the kth tree of a group with traffic demand t_s.
k	The kth tree of K for a multicast group, $k \in K$.	$\delta_{ij}^{t_s,k}$	A intermediate decimal variable to represent the value of $\varepsilon_{ij}^{t_s,k} \times f^{t_s,k}$.
u_{ij}	The link utilization of link e_{ij}.		
U	The maximum link utilization in the network.		
$sp_{d,ij}^{t_s}$	A binary parameter to indicate whether link e_{ij} is on the shortest path from legacy node i to destination node d for traffic demand t_s, $\forall d \in D_s$.		
π_i	A binary parameter to indicate whether node i is a SDN node.		

We divide the entire routing paths of a multicast group from the source node to all the corresponding destinations into two parts. One part is the unicast routing path from original node to the core SDN node. The other part is the multicast tree structure from the core SDN node to the destinations in the group. In the first unicast part, for the sake of good TE performance, we do not constrain legacy node to follow the shortest path routing. We leverage the detour routing policies [22-24] to fulfill the unicast path routing in the following model which redirects the multicast traffic from the source node to the calculated core SDN node. On the contrary, for the second multicast tree part, all the legacy nodes follow the shortest path routing for simplicity and interference-free of routing. And when it comes to the SDN node, traffic is split to the outgoing links utilizing SDN flexibility to enrich the tree diversity, since a number of K trees are needed per group to implement load balance of the network. We list the notations used in the node link formulation $\mathbf{P_1}$ in **Table 1**.

As we divide the whole routing paths from original node s to destination set D_s into two parts, that is the unicast part $s \to c$ and multicast tree part $c \to D_s$, where c is the appropriate SDN node chosen to act as the unique core node, namely, the waypoint for every multicast group.

$$P_1: \qquad\qquad \min \quad U$$

$$subject \quad to:$$

$$\sum_{j\in V} x_{ij}^{t_s} - \sum_{j\in V} x_{ji}^{t_s} = \begin{cases} 1-\beta_i^{t_s}, & i=s \\ \\ -\beta_i^{t_s}, & others \end{cases} ,\forall s \in S, \forall t_s \in T \qquad (1)$$

$$\sum_{j\in V} y_{d,ij}^{t_s,k} - \sum_{j\in V} y_{d,ji}^{t_s,k} = \begin{cases} \beta_i^{t_s}-1, & i=d \\ \\ \beta_i^{t_s}, & others \end{cases} ,\forall k \in K, \forall t_s \in T, \forall d \in D_s \qquad (2)$$

$$y_{d,ij}^{t_s,k} \le sp_{d,ij}^{t_s}, \forall k \in K, \forall t_s \in T, \forall d \in D_s, \forall i \notin SDN \qquad (3)$$

$$\sum_{i\in V} \beta_i^{t_s} = 1, \forall t_s \in T, \forall i \in V \qquad (4)$$

$$\beta_i^{t_s} \le \pi_i, \forall t_s \in T, \forall i \in V \qquad (5)$$

$$\varepsilon_{ij}^{t_s,k} \ge x_{ij}^{t_s}, \forall k \in K, \forall t_s \in T, \forall e_{ij} \in E \qquad (6)$$

$$\varepsilon_{ij}^{t_s,k} \ge y_{d,ij}^{t_s,k}, \forall k \in K, \forall t_s \in T, \forall e_{ij} \in E, \forall d \in D_s \qquad (7)$$

$$\sum_{j\in V} \varepsilon_{ji}^{t_s,k} \le 1, \forall k \in K, \forall t_s \in T, \forall i \in V \qquad (8)$$

$$\sum_{k\in K} f^{t_s,k} \le 1, \forall t_s \in T \qquad (9)$$

$$\delta_{ij}^{t_s,k} \ge f^{t_s,k} + \varepsilon_{ij}^{t_s,k} - 1, \forall k \in K, \forall t_s \in T, \forall e_{ij} \in E \qquad (10)$$

$$\sum_{t_s\in T}\sum_{k\in K} \delta_{ij}^{t_s,k} * t_s \le C_{ij} * U_{ij}, \forall e_{ij} \in E \qquad (11)$$

$$U \ge U_{ij}, \forall e_{ij} \in E \qquad (12)$$

$$\sum_{t_s\in T} \beta_i^{t_s} \le M, \forall i \in SDN \qquad (13)$$

For every multicast group (s, D_s), $\forall s \in V$ with the traffic demand t_s, the first two equations are the flow continuity constraints, in order to find the paths from source node s to

every destination $\forall d \in D_S$ which are used to build up a total number of K trees. Note that the (s, D_s) pair can uniquely define one multicast group with a source node s and the corresponding destination set D_s. More specifically, for the first unicast part, equation (1) is used to establish the unicast path from s to c, where c is an appropriate SDN node selected to play the role of core node, namely the waypoint in each multicast group. Exactly when the source node s is chosen to be the core SDN node, namely of $\beta_s^{t_s} = 1$, the first unicast part contains no link except the source node. In this case, the upper branch of equation (1) is equal to 0. For the node that is not the source node, the lower branch of equation (1) demonstrates that, only the designated core SDN node can make the results of equation (1) equals to -1, otherwise, it equals to 0 which constructs the routing continuity constraints. And for the second tree part, equation (2) is leveraged to construct a number of K trees from c to the destination set D_s. In case that the destination node d is selected to be the core SDN node, which means $\beta_d^{t_s} = 1$, the path between s and d only includes the unicast part. Otherwise, for the lower branch of equation (2), it makes the routing continuity constraints for every single multicast tree. Constraint (3) ensures that all the legacy nodes follow the shortest path routing policy in the second multicast tree part for simplicity and free of routing interference, rather than the detouring routing policy followed by the legacy node in the first unicast part mentioned in Section 2.2. Equation (4) guarantees that, for each multicast group (s, D_s), there is only one SDN node selected to function as the core node. Equation (5) constrains that it is necessary for the core node to be a SDN node first. The binary variable $\varepsilon_{ij}^{t_s,k}$ is used to construct the whole routing paths for each multicast group based on the two divided parts obtained from eq. (1) and (2). And constraint (6) and (7) describe the tree construction details. Specifically, for the kth tree of multicast group (s, D_s), all the positive $\varepsilon_{ij}^{t_s,k}$ can be integrate to form the corresponding entire multicast tree. Not only that, equation (8) denotes that there is no more than one incoming link at every node in the whole routing paths of group (s, D_s), $\forall s \in V$, in order to strictly set up tree structure which demands loop-free routing. We use constraint (9) to describe that the summation of splitting fractions in each tree $\forall k \in K$ for group (s, D_s) does not exceed 1. Moreover, $f^{t_s,k} * \varepsilon_{ij}^{t_s,k}$ can indicate the traffic ratio on link (i, j), however, it is not linear because both $f^{t_s,k}$ and $\varepsilon_{ij}^{t_s,k}$ are decision variables. To address this issue, we introduce a new decision variable $\delta_{ij}^{t_s,k}$ to represent the traffic splitting fraction on link (i, j) of the kth tree for group (s, D_s), thus, $\delta_{ij}^{t_s,k}$ equals to $f^{t_s,k} * \varepsilon_{ij}^{t_s,k}$ actually. Moreover, equation (10) is proposed to create a relationship among the three variables. Based on this, constraint (11) further denotes the link capacity constraint. With respect to the node capacity, namely the limited flow table resource on each SDN node, equation (13) is used to ease the node scalability problem, whereby each SDN node i can act as the core node, which makes $\beta_i^{t_s}, \forall t_s \in T$ equals to 1, for no more than a number of M multicast groups. In this paper, we make the constant M equal to half of the total number of multicast traffic demands in the network.

4. Algorithm Design

Since there are many variables in the node-link model proposed in section 3, it is difficult to get the optimal results in polynomial time especially for large scale topology. Thus, we further introduce a link-path model to solve the problem efficiently, while making use of the node-link model as an optimal results in comparison with the link-path model. For a link-path model, we should compute all the available routing paths for every multicast group before solving it. However, it is hard to directly and effectively acquire all the multicast routing paths especially for large scale network topology. In this section, firstly we design the spanning tree formulation (STF) algorithm to compute a most promising subset of all the possible routing paths for every single multicast group. Afterwards, the multicast routing and splitting (MRS) algorithm is proposed to solve the routing problem effectively.

4.1 Link path formulation P₂

Table 2. Link Path Formulation $\mathbf{P_2}$

Parameter	Description	Variable	Description
K	The set of trees for every multicast group, $\forall k \in K$, $|K| = K$.	$x_s^{c,m}$	A binary variable to indicate whether the traffic of a multicast group uses the mth path from source node s to SDN node c.
$f_{s,c,m}^{ij}$	A binary parameter to indicate whether link e_{ij} is on the mth path from source node s to the core SDN node c.	$y_s^{c,k}$	A decimal variable to indicate the fraction of traffic from source node s in a multicast group to the corresponding destination node set D_s carried by the kth tree of the multicast group.
$g_{c,D_s,k}^{ij}$	A binary parameter to indicate whether link e_{ij} is on the kth subtree from SDN node c to destination node set D_s with the source node s in a multicast group.	$\varepsilon_{s,k}^{ij}$	A binary variable to indicate whether link e_{ij} is on the kth tree of the multicast group with the source node s.
$P_{s,c}$	The set of paths from source node s of a multicast group to a SDN node c.		
$p_{s,c,m}$	The mth path in $P_{s,c}$.		

In the link-path model, we also divide the entire routing paths into two parts as in the node-link model for group (s, D_s), $\forall s \in V$. In the first unicast path $s \rightarrow c$, $P_{s,c}$ is used to denote all the candidate paths between source node s and SDN node c. A binary variable $x_s^{c,m}$ is proposed to denote whether the mth path from s to c is selected to carry out the whole traffic t_s of group (s, D_s). In the second tree part $c \rightarrow D_s$, a number of K trees are calculated for each group to implement load balance in the network. We use a decimal variable $y_s^{c,k}$ to indicate the splitting ratio on every tree $\forall k \in K$ of the group (s, D_s). We list the notations used in the link path formulation $\mathbf{P_2}$ in **Table 2**.

$$P_2: \qquad\qquad \min \quad U$$

$$subject \quad to:$$

$$\sum_{c \in SDN} \sum_{p_{s,c,m} \in P_{s,c}} x_s^{c,m} = 1, \forall s \in S \tag{14}$$

$$\sum_{k \in K} y_s^{c,k} = \sum_{p_{s,c,m} \in P_{s,c}} x_s^{c,m}, \forall s \in S, \forall c \in SDN \tag{15}$$

$$l_{ij} = \sum_{t_s \in T} \sum_{c \in SDN} \left(\sum_{p_{s,c,m} \in P_{s,c}} x_s^{c,m} * f_{s,c,m}^{ij} + \sum_{k \in K} y_s^{ij} * g_{c,D_s,k}^{ij} \right), \forall e_{ij} \in E \tag{16}$$

$$U \geq l_{ij} / C_{ij}, \forall e_{ij} \in E \tag{17}$$

$$\sum_{s \in S} \sum_{p_{s,c,m} \in P_{s,c}} x_s^{c,m} \le M, \forall c \in SDN \tag{18}$$

$$\varepsilon_{s,k}^{ij} \ge \sum_{c \in SDN} \sum_{p_{s,c,m} \in P_{s,c}} x_s^{c,m} * f_{s,c,m}^{ij}, \forall s \in S, \forall k \in K, \forall e_{ij} \in E \tag{19}$$

$$\varepsilon_{s,k}^{ij} \ge \sum_{c \in SDN} y_s^{c,k} * g_{s,Ds,k}^{ij}, \forall s \in S, \forall k \in K, \forall e_{ij} \in E$$
$$\tag{20}$$

$$\sum_{i \in V} \varepsilon_{s,k}^{ij} \le 1, \forall s \in S, \forall k \in K, \forall j \in V \tag{21}$$

We set the optimization objective function as minimizing the maximum link utilization with details shown in the link-path model **P₂.** For the first unicast part, equation (14) ensures that only a single path between source node s and core SDN node c is selected to route the multicast traffic t_s of group *(s, Dₛ),* which makes the unicast part determination for every multicast group. Note that $\sum_{p_{s,c,m} \in P_{s,c}} x_s^{c,m}$ is also binary which indicates whether SDN node c is chosen to be the core node for the group *(s, Dₛ).* And, if that is the case, which means $\sum_{p_{s,c,m} \in P_{s,c}} x_s^{c,m}$ equals to 1, the multicast traffic t_s will be split once reaching the core SDN node c and then routed on every single tree $k, \forall k \in K$ with optimized splitting fraction $y_s^{c,k}$ of t_s for the kth tree, while $\sum_{k \in K} y_s^{c,k}$ is required to be equal to 1.

That leads to the relationship between the two decision variables, $y_s^{c,k}$ and $x_s^{c,m}$, which are detailed in equation (15). Equation (16) and (17) build the link capacity constraint. Moreover, with respect to the SDN node scalability issue, we use constraint (18) to guarantee that any SDN node can act as the core node for a total number of M groups at most, namely imposing restrictions on the number of table entries needed per SDN node, in order to ease the scalability problem. In addition, a binary variable $\varepsilon_{s,k}^{ij}$ is defined to indicate whether link *(i, j)* is on the kth tree of the group *(s, Dₛ).* Thus, we come up with three constraints (19), (20), and (21), whereby the multicast tree structure is strictly established. Specifically, equation (19) and (20) are proposed to establish the tree structure based on the two decision variables $y_s^{c,k}$ and $x_s^{c,m}$. And equation (21) demonstrates that there is no more than one incoming links for each node, so as to build a strictly tree structure with no loop.

4.2 The Spanning Tree Formulation Algorithm

Previously, we divide the whole multicast routing paths per group into two parts. In the first unicast part, we make $P_{s,c}$, $\forall s \in S, \forall c \in SDN$ denote all the available paths between s and c. In the second tree part starting from c to every single destination $\forall d \in D_s$, a total number of K trees are expected for each group to partially route the traffic t_s to benefit TE performance.

Since **P₂** is a link path model, we should compute all the available trees before solving it. However, it is difficult to directly and efficiently acquire all the available trees for large scale network topology. Thus, we firstly utilize the k-shortest path algorithm, Yen's

algorithm explicitly, to search a most promising subset of P_{sc}. It is worth noting that, in the first unicast part, the detour routing policy mentioned in Section 2 is performed for the legacy routers to fulfill the optimal routing construction on behalf of the TE performance. Then, in the second tree part, we ensure every legacy router to follow the shortest path routing for simplicity and interference-free of routing, while fully utilizing the SDN routing flexibility to enrich the tree diversity for each group. That is to say, when it comes to a legacy node, the shortest path routing policy is executed, namely only one outgoing link can be used. When it comes to a SDN node, it is allowable to leverage multiple outgoing links to reach every single destination. On top of that, a number of K trees are constructed for every group. Note that, the multicast traffic t_s is not split on the unicast path $s \rightarrow c$, namely a single path from source node s to the core SDN node c is selected to route t_s. When t_s flows out of the core SDN node c, it is split among K trees to realize load balance.

　　　　We propose a heuristic algorithm, the spanning tree formulation (STF) algorithm, in order to build the spanning trees from core SDN node c to D_s for the multicast group $(s,\ D_s)$ as shown in Algorithm 1. Given a graph $G(V,\ E)$, at the initial step from line 1-7, for each group $(s,\ D_s)$, $\forall s \in V$ and every SDN node $c \in SDN$, a path set $P_{c,D_s,d_i}, \forall c \in SDN, \forall s \in V, \forall d_i \in D_s$ are precomputed using the K-shortest path algorithm, which connect the core SDN node c to every single destination $d_i \in D_s$. Not only that, each path set P_{c,D_s,d_i} should satisfy the constraints the same as the restrictions in the node-link model P_1, that is only the shortest path egress port to d_i on the legacy node is allowed to use to reach the destination d_i. On the contrary, all the egress ports on the SDN node which have access to the destination d_i are allowed, whereby the path diversity can be fully utilized to create enough paths. On top of that, at the second step from line 8-10, for every group $(s, D_s), \forall s \in V$ and every SDN node $c \in SDN$ which may possibly act as the core node candidate, we randomly select one path p_{c,D_s,d_i} from each path set P_{c,D_s,d_i}, thus to accumulate

---

Algorithm 1: The Spanning Tree Formulation (STF) Algorithm

　Input: $G(V,E)$, $S = \{s, \forall s \in V\}$, $D_s = \{d_i, \forall d_i \in D_s, \forall s \in V\}$, $SDN = \{c, \forall c \in SDN\}$, K, $k = 0$.

　Output: Spanning trees

　1: **for** $\forall s \in S$ **do**
　2: 　**for** $\forall c \in SDN$ **do**
　3: 　　**for** $\forall d_i \in D_s$ **do**
　4: 　　　Run K-shortest path to calculate P_{c,D_s,d_i};
　5: 　　**end for**
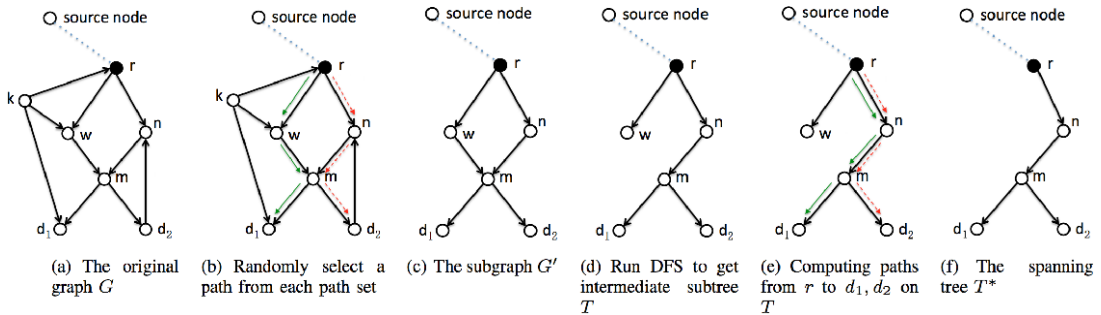　6: 　**while** $k \leq K$ **do**
　7: 　　　$k = k + 1$;

8:   **for** $\forall d_i \in D_s$ **do**

9:   Randomly select a path $p_{c,D_s,d_i} \in P_{c,D_s,d_i}$;

10:   **end for**

11:   Combine the paths together to form a subgraph $G_{c,D_s,k} \in G$;

12:   Run DFS on $G_{c,D_s,k}$ from node c to obtain a spanning tree $T_{c,D_s,k}$;

13:   **for** $\forall d_i \in D_s$ **do**

14:   Compute the path $q_{c,D_s,k}$ from d_i on tree $T_{c,D_s,k}$

15:   **end for**

16:   Combine all the path $q_{c,D_s,k}, \forall d_i \in D_s$ to get the final spanning tree $T^*_{c,D_s,k}$

17: **end while**

18: **end for**

19: **end for**

20: Return $T^* = \{T^*_{c,D_s,k}, \forall s \in S, \forall c \in SDN, k = 1..., K\}$

---

a total number of $|D_s|$ paths and then combine them together to form a subgraph $G_{c,D_s} \subseteq G$. At the third step from line 11-14, on each subgraph $G_{c,D_s}, \forall c \in SDN, \forall s \in V$ we run Depth-First-Search (DFS) algorithm from the SDN node c to obtain an intermediate tree T_{c,D_s}. Then, based on the tree $T_{c,D_s}, \forall c \in SDN, \forall s \in V$, the corresponding on-tree paths q_{c,D_s,d_i} from c to every single destination $d_i \in D_s$ are computed. Finally from line 15 to the end, we combine all the resulting paths $q_{c,D_s,d_i}, \forall c \in SDN, \forall s \in V$ to build a spanning tree $T^*_{c,D_s}, \forall c \in SDN, \forall s \in V$.

(a) The original graph G (b) Randomly select a path from each path set (c) The subgraph G' (d) Run DFS to get intermediate subtree T (e) Computing paths from r to d_1, d_2 on T (f) The spanning tree T^*

Fig. 2. The multicast tree construction

Example. We illustrate the proposed STF algorithm using **Fig. 2**. In **Fig. 2(a)**, given a multicast group with the source node and two destinations d_1 and d_2, after redirecting the multicast traffic of the group from the source node to the sole SDN node r which serves as

the core node of the group, we manage to build an optimal tree from the core SDN node r to d_1 and d_2 based on the graph G which consists of 4 legacy nodes apart from the source node and the two destinations. Firstly, we compute two path sets from r to d_1 and d_2 respectively. Then, we randomly choose one path from each path set which are $r \rightarrow w \rightarrow m \rightarrow d_1$ and $r \rightarrow n \rightarrow m \rightarrow d_2$ denoted by green solid line and red dashed line separately in **Fig. 2(b)**. Thus, a subgraph G' is obtain consisting of the two paths shown in **Fig. 2(c)**. On top of G', we run DFS algorithm starting from node r to gain an intermediate tree T described in **Fig. 2(d)**. After that, based on T we continue to compute the paths from core SDN node r to the two destinations d_1 and d_2 indicated by the red dashed line and green solid line in **Fig. 2(e)**. Afterwards, by employing combinations of the two resulting paths, the final spanning tree T^* can be achieved presented in **Fig. 2(f)**.

4.3 Multicast Routing and Splitting (MRS) Algorithm

We propose a multicast routing and splitting (MRS) algorithm to calculate the maximum link utilization as shown in Algorithm 2. From line 1-3, we leverage k-shortest path algorithm to obtain the subset of $P_{s,c}$ and parameter $f_{s,c,m}^{ij}$. From line 4-6, the Algorithm 1 is leveraged to calculate the tree set $T^* = \{T_{c,D_s,k}^*, \forall t_s \in T, \forall c \in SDN, k=1...,K\}$ and parameter $g_{c,D_s,k}^{ij}$. Then, we solve the link-path formulation P_2 of the routing problem in line 7 and finally gain the minimum maximum link utilization U.

Time Complexity. We first find the k-shortest paths between any two nodes in $G(V, E)$ with Yen's algorithm in $O(K|V|(|E|+|V|log|V|))$ time as the pre-processing procedure. Since the time cost of DFS algorithm is $O(|V|+|E|)$, then we finish the total tree constructions for $G(V, E)$, *where* a number of K trees are calculated for each multicast group, in $O(K|V|(3|V|+|E|))=O(K|V|(|V|+|E|))$. Therefore, the overall time complexity of the MRS algorithm is $O(K(|V|(|E|+|V|log|V|)+|V|(|V|+|E|)))$.

Algorithm 2: Multicast Routing and Splitting (MRS) Algorithm

Input: $G(V,E)$, $T = \{t_s, \forall s \in V\}$, $SDN = \{c, \forall c \in SDN\}$.
Output: Minimum maximum link utilization U

1: **for** $\forall t_s \in T$ **do**
2:    **for** $\forall c \in SDN$ **do**
3:      Run K-shortest path to obtain the subset of $P_{s,c}$ and parameter $f_{s,c,m}^{ij}$;
4:      Run Algorithm 1 to calculate $T^* = \{T_{c,D_s,k}^*, \forall t_s \in T, \forall c \in SDN, k=1...,K\}$

and parameter $g_{c,D_s,k}^{ij}$;
5:    **end for**
6: **end for**
7: Solve link-path formulation P_2;
8: Return U

---

5. Simulation Results

In this section, we conduct extensive simulation experiments to evaluate the proposed MRS algorithm, as well as some other comparable algorithms in real network topologies.

Table 3. Topologies In Experiment

Topology	Number of Nodes	Number of Links
Biznetworks	29	33
Ebone	28	66
Exodus	22	51

5.1 Simulation Setup

The information for the three real network topologies that we refer to are shown in **Table 3** provided by Rocketfuel [25] and the Internet topology zoo. For each topology, different traffic matrices and link capacities are random generated to acquire more general results. Each simulation result is averaged by 100 samples. The source and destinations are randomly selected from each network. All the simulation experiments are implemented in an Intel E5-2620 sever with six CPUs and 16GB memory.

We compare the MRS algorithm with the following algorithms: 1) the shortest path tree (SPT) algorithm in IP network, 2) the Node-Link algorithm employing Mix Integer Linear Programming (MILP) formulation proposed in section III**,** which finds the optimal solution of the routing problem in HSDN, 3) the Prim algorithm to obtain a minimum spanning tree per multicast group in full SDN (PrimSDN), 4) the BestSDN algorithm which creates K trees for every single group using the MILP formulation in full SDN, rather than in HSDN as the Node-Link algorithm. And it serves as a lower bound to compare the results of other comparison algorithms mentioned above. We change the SDN deployment rate, the number of destinations $|D_s|$ as well as the number of trees K calculated per group to obtain different simulation results. The performance metrics include 1) the maximum link utilization, 2) the traffic concentration which equals to the ratio between the peak link utilization and the average link utilization in the network [11, 26], 3) The average number of edges per multicast tree normalized by SPT.

5.2 SDN deployment rate

Since the deployment rate has a significant impact on the TE performance, we firstly focus on the determination of an adequate SDN deployment rate in order to make an optimal tradeoff between deployment budget and network performance. Greedy algorithm is used to increasingly upgrade node that forwards more volumes of multicast traffic [5]. For each given traffic matrix, we change the SDN deployment rate ranging from 20% to 60% increasing by 10%, while changing the group size $|D_s|$ from 3 to 8 increasing by 1 and the number of trees K from 2 to 10 increasing by 2, in order to obtain the corresponding average maximum link utilization (AMU) and the traffic concentration.

For the space limitation, we just present the average maximum link utilizations in the case of three parameter settings denoted by a parameter pair (a, *b),* which involves the group size a and the number of trees *b* per group. For example, as shown in **Fig. 3(a)**, Ebone(4, 4) denotes for each group there are 4 destinations and 4 calculated trees in Ebone. From **Fig. 3**, we can find that when the SDN deployment rate increases to 40%, the average

maximum link utilization of MRS algorithm becomes lower than the PrimSDN algorithm based on the three parameter settings (4, 4), (5, 6) and (8, 10) for the three topologies respectively. Moreover, as shown in **Fig. 4**, the traffic concentrations for the three topologies also decease rapidly when the SDN deployment rate reaches 40%. The rest parameter settings are not listed here because of space cause, which bring to the same conclusion as **Fig. 3** and **Fig.4**.

5.3 Sufficient number of trees per group

Obviously, a small K may have slight effect on the TE performance, while a large K may induce a heavy workload on the SDN controller. Thus, we intend to find a sufficient number of trees for each group, in order to make a tradeoff between the network performance and the computational cost.

We use topology Ebone as an example to illustrate how the different value of K influences the network performance. In **Fig. 5**, we find that when SDN deployment rate stays at 30%, multicast groups which involve a larger number of destinations may requires a much bigger value of K to get a equally average maximum link utilization compared with PrimSDN algorithm. By contrast, when the SDN deployment rate increases to 40%, the increase of destinations per group has little effect on the number of trees K needed for the MRS algorithm to gain a comparable TE performance to PrimSDN. More specifically, with the deployment rate at 30%, each group with 5 destinations requires 4 trees as shown in **Fig. 5 (a)**, while groups with 8 destinations require more than 10 trees presented in **Fig. 5 (b)**. When the SDN deployment rate is raised to 40%, approximately 2 trees are need for each multicast group to gain an almost equal TE performance as PrimSDN demonstrated in
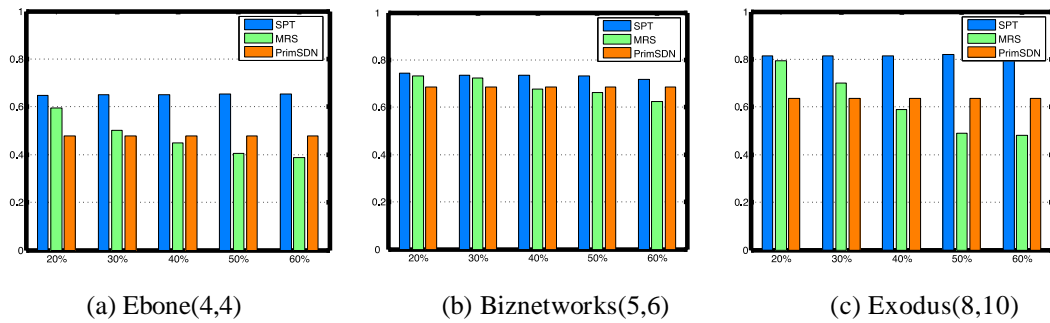

| (a) Ebone(4,4) | (b) Biznetworks(5,6) | (c) Exodus(8,10) |

Fig. 3. The average maximum link utilization vs. SDN deployment rate

| (a) Ebone(4,4) | (b) Biznetworks(5,6) | (c) Exodus(8,10) |
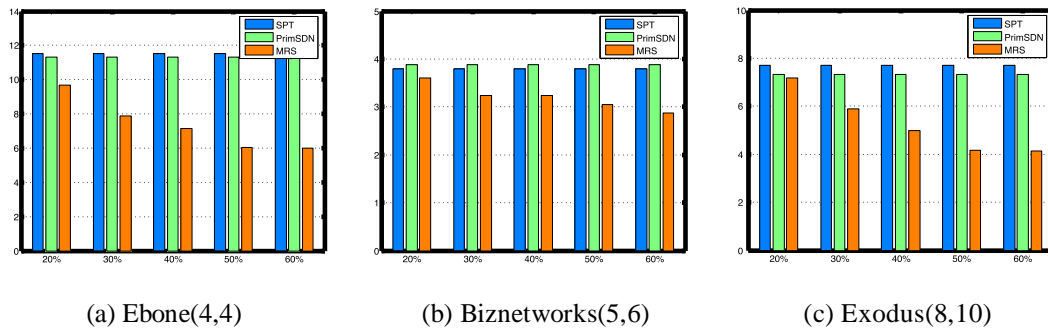
Fig. 4. The traffic concentration vs. SDN deployment rate

(a) 30% & Group size: 5   (b) 30% & Group size: 8

(c) 40% & Group size: 5   (d) 30% & Group size: 8
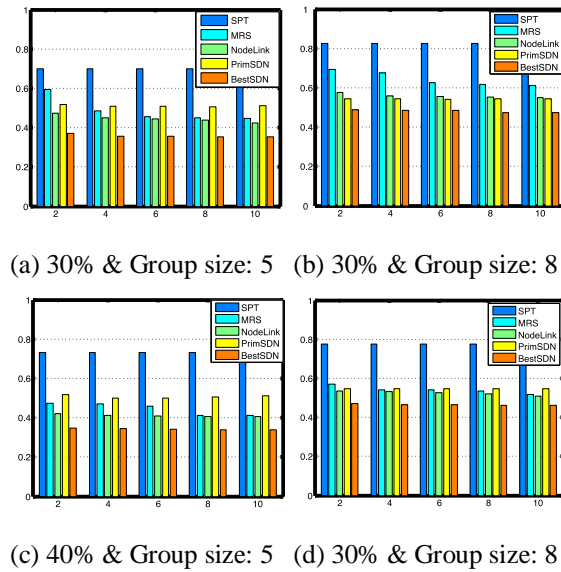
Fig. 5. The average maximum link utilization vs. SDN deployment rate & group size on Ebone

(a) Ebone   (b) Biznetworks   (c) Exodus
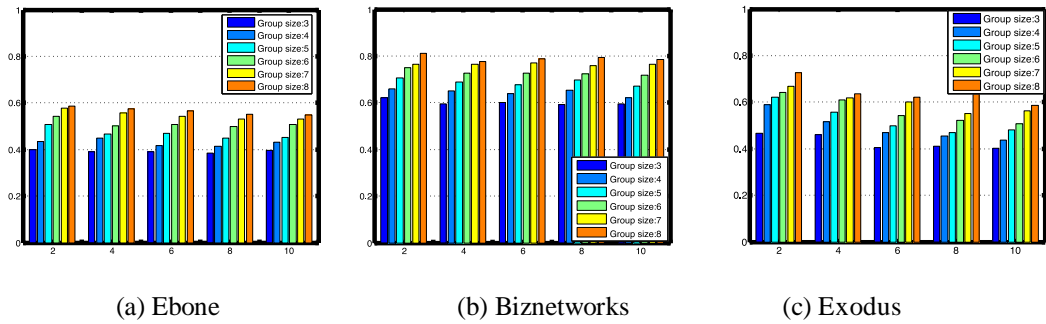
Fig. 6. The average maximum link utilization vs. The number of trees per group using MRS algorithm

Fig. 5(c) and **(d)**. That is to say, a relatively lower SDN deployment rate requires a much larger number of trees per group to obtain a desirable TE performance, especially for the multicast groups with big size. Also, because the space is limited, we do not present the simulation results of the other two topologies, which come to the identical conclusion as Ebone.

Thus, in order to make the tradeoff between the network performance, the deployment budget and the computing workload for the controller to obtain the calculated trees, we fix the SDN deployment rate to 40%. After that, we change the number of destinations, namely the group size, from 3 to 8 and the number of trees per group ranging from 2 to 10, aiming to find an adequate number of trees needed for the MRS algorithm to balance the computing cost with TE performance. On top of that, we average 100 samples to get the average maximum link utilization for the three topologies separately. From **Fig. 6**, we can infer that there is a slight difference in the average maximum link utilization when we gradually increase the value of K from 2 to 10 per group. Then we can come to the conclusion that, when K is specified to be 2 as well as setting the SDN deployment rate at

40%, the HSDN multicast using MRS algorithm can obtain comparable TE performance to SDN multicast which leverages PrimSDN algorithm.

5.4 Simulation On Routing Efficiency

Table 4. Averaged Normalized Hops Per Multicast Tree

Topology	Group Size	Algorithm	Normalized Hops
Biznetworks	3	MRS	1.101
Biznetworks	3	PrimSDN	0.913
Biznetworks	5	MRS	1.202
Biznetworks	5	PrimSDN	0.905
Biznetworks	8	MRS	1.133
Biznetworks	8	PrimSDN	0.897
Ebone	3	MRS	1.278
Ebone	3	PrimSDN	0.778
Ebone	5	MRS	1.202
Ebone	5	PrimSDN	0.788
Ebone	8	MRS	1.133
Ebone	8	PrimSDN	0.711
Exodus	3	MRS	1.262
Exodus	3	PrimSDN	0.764
Exodus	5	MRS	1.190
Exodus	5	PrimSDN	0.785
Exodus	8	MRS	1.104
Exodus	8	PrimSDN	0.819

Improving the link utilization is not the only objective in TE, since paths may get longer when traffic is balanced, as well as implementing the core SDN node waypoint enforcement. The previous metrics indicate that MRS algorithm is effective at load balancing the network traffic. And it is also very important to quantify the routing efficiency, namely the deviation from the hop count in the shortest path tree. We calculate the average number of edges per multicast tree based on MRS, PrimSDN and SPT algorithms in the three topologies individually. The results are normalized by the hop count in SPT algorithm as shown in **Table 4**. Obviously, MRS generates a little larger amount of hops per tree as the price for lower maximum link utilization, which are 1.1-1.3 times larger than SPT and 1.2-1.7 times larger than PrimSDN. Thus, we come to the conclusion that MRS algorithm has a limited range of influence on the routing efficiency.

From the perspective of time complexity, as the time complexity of the proposed algorithm is in polynomial time scale, the proposed MRS algorithm can be deployed in large network topology, as well as the WAN environment. Of course, it comes at the cost of controller overhead, and may not respond on-the-fly to network changes. On the other hand, when deploying SDN in real networks, large networks like WAN are always partitioned into several smaller domains for numerous reasons such as scalability, incremental deployment, security and so on. And each domain runs one controller. This can alleviate the scalability issue greatly, since in every SDN domain a limited number of nodes are centrally controlled by one controller.

6. Conclusion

In HSDN, for implementing multicast traffic engineering as well as enabling the manageability for the multicast traffic, e.g., access control, traffic measurement, firewall

actions and middlebox-supported network services [6], all packets of each multicast group are forwarded to an appropriate SDN node acting as the core node before reaching the destinations. Thus, on one hand, the security risk can be alleviated by centralized access control on top of the core SDN node. On the other hand, the dynamic control over the routing paths can also be achieved to benefit TE performance. In addition, a number of K trees starting from the core SDN node to the destinations are computed, in order to further enhance the TE performance. In this paper, a node-link model is firstly proposed. However, it is difficult to obtain the optimal results in polynomial time for large scale networks. Thus, we propose the MRS algorithm based on a link-path model to effectively solve the routing problem. Compared with other state-of-art works, the simulation results are inspiring with significant improvement in traffic distribution characteristics, while generating an acceptably larger number of edges per tree on average. We also show that when 40% of the SDN switches are deployed as well as calculating 2 trees per group, the HSDN multicast can achieve comparable network performance to that of SDN multicast. In the future work, we will carry out experiments on real SDN devices and consider other hybrid network scenario from the perspective of network performance and scalability issues.

References

[1] L. H. Huang, H. C. Hsu, S. H. Shen, D. N. Yang, and W. T. Chen, "Multicast traffic engineering for software-defined networks," in *Proc. of Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on.* IEEE, pp. 1-9, 2016. Article (CrossRef Link)

[2] R. Mali, X. Zhang, and C. Qiao, "Benefits of multicasting in all-optical networks," in *Proc. of Photonics East (ISAM, WDC, IEMB). International Society for Optics and Photonics*, pp. 209-220, 1998. Article (CrossRef Link)

[3] I. Cisco, "Cisco visual networking index: Forecast and methodology, 2011-2016," *CISCO White paper,* pp. 2011-2016, 2012. Article (CrossRef Link)

[4] A. Iyer, P. Kumar, and V. Mann, "Avalanche: Data center multicast using software defined networking," in *Proc. of Communication Systems and Networks (COMSNETS), 2014 Sixth International Conference on.* IEEE, pp. 1-8, 2014. Article (CrossRef Link)

[5] Y. Guo, Z. Wang, X. Yin, X. Shi, **J.** Wu, and H. Zhang, "Incremental deployment for traffic engineering in hybrid sdn network," in *Proc. of Computing and Communications Conference (IPCCC), 2015 IEEE 34th International Performance.* IEEE, pp. 1-8, 2015. Article (CrossRef Link)

[6] K. Poularakis, G. Iosifidis, G. Smaragdalds, and L. Tassiulas, "One step at a time: Optimizing sdn upgrades in isp networks," in *Proc. of Proceedings of IEEE INFOCOM,* 2017. Article (CrossRef Link)

[7] D. K. Hong, Y. Ma, S. Banerjee, and Z. M. Mao, "Incremental deployment of sdn in hybrid enterprise and isp networks," in *Proc. of Proceedings of the Symposium on SDN Research.* ACM, p. 1, 2016. Article (CrossRef Link)

[8] T. Ballardie, P. Francis, and J. Crowcroft, "Core based trees (cbt)," in *Proc. of ACM SIGCOMM Computer Communication Review,* vol. 23, no. 4. ACM, pp. 85-95, 1993. Article (CrossRef Link)

[9] C. A. Marcondes, T. P. Santos, A. P. Godoy, C. C. Viel, and C. A. Teixeira, "Castflow: Clean-slate multicast approach using in-advance path processing in programmable networks," in *Proc. of Computers and Communications (ISCC), 2012 IEEE Symposium on.* IEEE, pp. 000 094-000101, 2012. Article (CrossRef Link)

[10] D. Kotani, K. Suzuki, and H. Shimonishi, "A design and implementation of openflow controller handling ip multicast with fast tree switching," in *Proc. of Applications and the Internet*

(SAINT), 2012 IEEE/IPSJ 12th International Symposium on. IEEE, 2012, pp. 60-67.
Article (CrossRef Link)

[11] A. Craig, B. Nandy, I. Lambadaris, and P. Ashwood-Smith, "Load balancing for multicast traffic in sdn using real-time link cost modification," in *Proc. of Communications (ICC), 2015 IEEE International Conference on.* IEEE, pp. 5789-5795, 2015. Article (CrossRef Link)

[12] J. Ruckert, J. Blendin, R. Hark, and D. Hausheer, "Dynsdm: Dynamic and flexible software-defined multicast for isp environments," in *Proc. of Network and Service Management (CNSM), 2015 1 1 th International Conference on.* IEEE, pp. 117-125, 2015.
Article (CrossRef Link)

[13] M. W. Lee, Y. S. Li, X. Huang, Y. R. Chen, T. F. Hou, and C. H. Hsu, "Robust multipath multicast routing algorithms for videos in software-defined networks," in *Proc. of Quality of Service (IWQoS), 2014 IEEE 22nd International Symposium of* IEEE, pp. 218-227, 2014.
Article (CrossRef Link)

[14] M. Zhao, B. Jia, M. Wu, H. Yu, and Y. Xu, "Software defined network-enabled multicast for multi-party video conferencing systems," in *Proc. of Communications (ICC), 2014 IEEE International Conference on.* IEEE, pp. 1729-1735, 2014. Article (CrossRef Link)

[15] L. H. Huang, H. J. Hung, C. C. Lin, and D. N. Yang, "Scalable and bandwidth-efficient multicast for software-defined networks," in *Proc. of Global Communications Conference (GLOBECOM), 2014 IEEE.* IEEE, pp. 1890-1896, 2014. Article (CrossRef Link)

[16] S. H. Shen, L. H. Huang, D. N. Yang, and W. T. Chen, "Reliable multicast routing for software-defined networks," in *Proc. of Computer Communications (INFOCOM), 2015 IEEE Conference on.* IEEE, pp. 181-189, 2015. Article (CrossRef Link)

[17] R. Mukherjee and J. W. Atwood, "Rendezvous point relocation in protocol independent multicast-sparse mode," *Telecommunication Systems,* vol. 24, no. 2, pp. 207-220, 2003.
Article (CrossRef Link)

[18] M. Castro, P. Druschel, A. M. Kermarrec, and A. I. Rowstron, "Scribe: A large-scale and decentralized application-level multicast infrastructure," *IEEE Journal on Selected Areas in communications,* vol. 20, no. 8, pp. 1489-1499, 2002. Article (CrossRef Link)

[19] W. Ralph and Z. Martina, "Multicast communications: Protocol and applications," 2000.
Article (CrossRef Link)

[20] K. Stachowiak and P. Zwierzykowski, "Rendezvous point based approach to the multi-constrained multicast routing problem," *AEU-International Journal of Electronics and Communications,* vol. 68, no. 6, pp. 561-564, 2014. Article (CrossRef Link)

[21] C. Ren, S. Wang, J. Ren, X. Wang, T. Song, and D. Zhang, "Enhancing traffic engineering performance and flow manageability in hybrid sdn," in *Proc. of Global Communications Conference (GLOBECOM), 2016 IEEE.* IEEE, pp. 1-7, 2016. Article (CrossRef Link)

[22] S. Vissicchio, 0. Tilmans, L. Vanbever, and J. Rexford, "Central control over distributed routing," in *Proc. of Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication.* ACM, pp. 43-56, 2015. Article (CrossRef Link)

[23] D. Levin, M. Canini, S. Schmid, F. Schaffert, A. Feldmann *et al.,* "Panopticon: Reaping the benefits of incremental sdn deployment in enterprise networks," in *Proc. of USENIX ATC,* 2014.
Article (CrossRef Link)

[24] C. Jin, C. Lumezanu, Q. Xu, Z. L. Zhang, and G. Jiang, "Telekinesis: controlling legacy switch routing with openflow in hybrid networks," in *Proceedings of the 1st ACM SIGCOMM Symposium on Software Defined Networking Research.* ACM, p. 20, 2015.
Article (CrossRef Link)

[25] N. Spring, R. Mahajan, and D. Wetherall, "Measuring isp topologies with rocketfuel," *ACM SIGCOMM Computer Communication Review,* vol. 32, no. 4, pp. 133-145, 2002.
Article (CrossRef Link)

[26] H. Kaur and J. Singh, "Performance comparison of olsr, grp and tora using opnet," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 2, no. 10, pp. 260-267, 2012. Article (CrossRef Link)
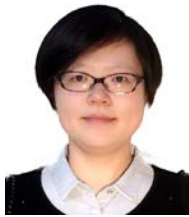
Cheng Ren received her master degree in circuit and system from University of Electronic Science and Technology of China (UESTC), Chengdu, China in 2006. Now, she is a Ph.D. candidate in Communication Engineering in UESTC. Her research interests include software-defined networking and network resource allocation.

Sheng Wang received the B.S., M.S., and Ph.D. degrees in Electrical Engineering from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 1992, 1995, and 1999, respectively. He is now a Professor and a Research Group Leader in UESTC. His research interests include network design, optical switching, and next generation networks.

Jing Ren received the B.S., and Ph.D. degree in Communication Engineering from University of Electronic Science and Technology of China (UESTC), Chengdu, China in 2007 and 2015. Currently, she is a postdoctoral researcher in UESTC. Her research interests include network architecture and protocol design, information-centric networking and software-defined networking.

Xiong Wang received the Ph.D. degree in communication and information engineering from University of Electronic Science and Technology of China (UESTC), Chengdu, China in 2008. He is currently an Associate Professor with the Key Laboratory of Broadband Optical Fiber Transmission and Communication Networks, School of Communication and Information Engineering, University of Electronic Science and Technology of China, Chengdu, China. His research interests include network architecture and protocol design, information-centric networking and software-defined networking.