IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received May 12, 2020, accepted June 14, 2020, date of publication June 19, 2020, date of current version June 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3003697

Reliability-Aware Multi-Source Multicast Hybrid Routing in Softwarized Networks

LONG QU[1], (Member, IEEE), AND CHADI ASSI[2], (Fellow, IEEE)
[1]Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China
[2]CIISE Department, Concordia University, Montreal, QC H3G 1M8, Canada

Corresponding author: Long Qu (qulongonline@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61801254, in part by the Zhejiang Natural Science Foundation under Grant LY20F010004, in part by the Scientific Research Foundation of Ningbo University under Grant XYL18006, and in part by the K. C. Wong Magna Fund from Ningbo University.

ABSTRACT Undoubtedly, these days our telecommunication networks are witnessing not only a major spike is data volumes, but also a shift in the mode of communications. Employees, news anchor and students are conducting their daily business and learning activities through online platforms as they shelter homes during this pandemic and this is expected to continue for some time. An overwhelming shift to one-to-many and many-to-many communications is observed and end users expect from their providers efficient, secure and reliable services. Operators of digital platforms are challenged to respond quickly to the rising demand, by enhancing deployability and manageability of their service. Virtualization is a key enabler for enhanced deployability and manageability where virtual functions can be automatically deployed on demand. Another challenge that providers deal with is the individualized requirements by services offered to users which may vary between high reliabilities, low latency, robust security and any combination thereof. This paper considers the problem of provisioning multi-source multicast services where each service consists of a set of in-network virtual functions that must be chained in a particular order to meet the quality of service demanded by end users. We deal with a reliable service where reliability is attained by provisioning backup functions for the service. We first calculate the requirements of VNF backups which account for fewer computing resource consumption. Next, we formulate the multi-source multicast hybrid routing as a Mixed Integer Linear Programming (MILP) and find a solution with optimal VNF placement and traffic routing. We also proposed a K-shortest path-based greedy algorithm to reduce the complexity for solving MILP. Numerical analysis and simulations are conducted to validate the proposed algorithms. Our results show multi-source multicast has a better routing selection compared to single-source multicast due to the more options of multicast sources for providing a reliable network service.

INDEX TERMS Network function virtualization, reliability, delay, multi-source multicast, resource optimization.

I. INTRODUCTION

A. PRELIMINARIES

In recent years, with the development of smart phone and high speed 4G/5G communication networks, the cost of video production and distribution has reduced sharply. Network live video streaming platforms have become more and more ubiquitous and popular for data/content distribution. Further, networks these days are witnessing a surge in data traffic, particularly streaming and video conferencing, which is attributed to the recent and ongoing pandemic that has forced people (students, teachers, employees, etc.) to turn into digital platforms to perform their businesses and do their online learning. As a result, we witness more and more a rising trend of services which relies on one-to-may (multicast) and many-to-many (multi-source multi-destination) communications and such services have stringent and individualized requirements [1], [2]. Such requirements can be quite heterogeneous with some end users requiring more secure and private sessions whereas others requiring more efficient and reliable ones. Operators of digital platforms are hence challenged to respond quickly to rising demands, by enhancing deployability and manageability of their services.

Softwarization, and in particular Network Function Virtualization (NFV), is a key technology for future 5G communication [3] which is expected to bring more automation

The associate editor coordinating the review of this manuscript and approving it for publication was Zehua Guo.

to networks, and facilitate deployability and manageability of network services. In addition network functions, which now are virtualized (i.e., VNFs), can be used by operators to enhance their service by, for example, adding on demand softwarized security functions among others. A myriad of functions can be combined together to create heterogeneous network services that operators can, in a flexible way. Operators can chain a set of virtual network functions [4]–[7] to deliver customized services to end users based on individual needs, e.g., security and privacy, video rendering and compression/decompression, etc..

In this paper, we address an emerging challenge for networks, content providers as well as operators of video tele-conferencing platforms, which is multicasting and service chaining with softwarized functions. We study the problem of provisioning multicast services, placement of virtualized functions, and chaining traffic through these functions to deliver end users with their demanded services. We consider provisioning of services to end users demanding high reliability. Often, in softwarized networks network functions are instantiated on virtual machines (VMs) which run on off the shelf servers. Hardware or software failures cause a failure to the provisioned functions which may disrupt the chain of the ongoing multicast session. For example, a failure of security or privacy-preserving functions for an online learning session involving kids may not be tolerated and can cause financial losses to the service provider. Accordingly, operators to strengthen the posture of their service and deliver reliable one will provision redundant network functions, such that when a primary function fails, its backup resumes service immediately. This work considers then the reliability enhancement for multicast service chaining with redundant virtualized network functions.

B. RELATED WORK AND NOVEL CONTRIBUTIONS

Multicast in NFV-enabled network has become a hot research topic and attracted interests from both academia and industry alike. In [8], the authors studied the NFV multicast resource optimization problem. In their model, the VNFs are placed within a single server. In [9], the optimal service function tree embedding in the shared multicast tree is studied. The proposed model supports the VNF placements of one multicast tree at dispersed network nodes. The authors of [10] studied the NFV multicast problem and proposed a multi-stage solution by separating the multicast traffic forwarding and function delivery. In [11], the authors studied throughput maximization of NFV-enabled multicasting. An approximation algorithm of the cost minimization problem is proposed. Furthermore, the online throughput maximization problem is studied. In [12], the authors studied the problem of NFV multicast with multipath routing. Precisely, one multicast service is allowed to be delivered by multiple multicast trees. For each destination, it is served by the multipath routing of multiple multicast trees. In [13], the authors focus on the problem of delay-aware NFV-enabled multicasting. An approximation algorithm for multicasting problem without delay requirements is proposed. By considering the total processing and transfer delay, an heuristic algorithm is proposed. Although the existing NFV multicast algorithms are able to minimize the resource consumption of multicast routing, they do not consider the reliability and end-to-end delay of network services. As discussed earlier, the reliability and delay requirements of end users are stringent for a reliable network service [14]–[16] (e.g., live streaming, disaster warning, synchronized broadcast, etc.). In this work, we propose a reliability aware NFV multicast resource optimization model with end-to-end delay constraints. In order to increase the acceptance ratio of multicast destination nodes, a multi-source multicast resource optimization model is formulated as a Mixed Integer Linear Programming(MILP). We further propose a K-shortest path-based greedy algorithm to reduce the complexity of MILP.

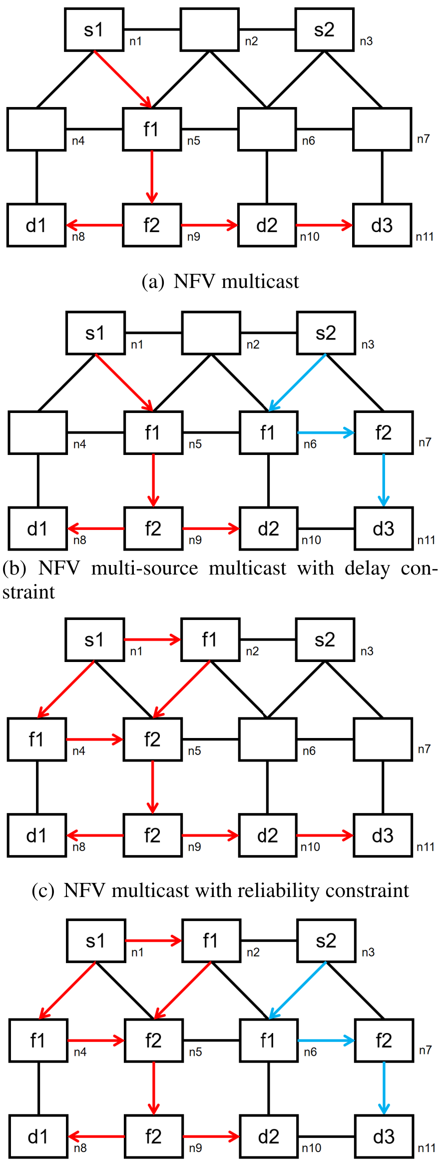
C. PAPER ORGANIZATION

The remaining of this paper is organized as follows. Section II presents the illustrative example. In Section III, we present the mathematical framework enclosing a detailed problem formulation, reliability guarantee of multicast services and greedy algorithm. Numerical results are presented in Section IV. And conclusion is shown in Section V.

II. PROBLEM STATEMENT

A. ILLUSTRATIVE EXAMPLE

An illustrative example is shown in Figure 1. Assume the VNF processing order of multicast service is $f_1 \rightarrow f_2$. There are 3 destinations $d1$, $d2$ and $d3$ in 11-node network ($n1$ - $n11$). Assume the bandwidth consumption for each destination is 2 units. We further assume each link has available bandwidth of 10 units and constant delay of 10 units. The reliability of node is 0.9. And each node has enough computing resource for processing VNFs. In the first case (Figure 1(a)), without reliability and delay constraints, $s1$ is selected as the multicast source. f_1 and f_2 are placed at $n5$ and $n9$ respectively. The link bandwidth consumption is 10 units. The end-to-end delay of d_1, d_2 and d_3 are 30 units, 30 units and 40 units. Next, if the end-to-end delay of $d1$, $d2$ and $d3$ are set as 30 units, $s2$ must be used for $d3$. The available route and VNF placements are shown in Figure 1(b). The red route has been used for $d1$ and $d2$. $d3$ is served by blue route. In this case, although the end-to-end delay is 30 units, it consumes 14 units of link bandwidth. In the third case, we further consider the reliability requirement of NFV multicast (e.g., the multicast destinations require a reliability of 0.98). For each type of VNFs, one backup must be added. There are two $f1$ and two $f2$ which are placed at $n2$, $n4$ $n5$ and $n9$ respectively. The route of Figure 1(c) consumes 16 units of link bandwidth. The reliability of multicast destinations is calculated as $(1-(1-0.9)*(1-0.9))^2 = 0.9801 > 0.98$. Note that the end-to-end delay of $d3$ is 50 units. If we consider the 30 units of delay constraint for $d3$, multi-source multicast must be used. The available routes are shown in Figure 1(d). Thus, the challenges of reliability-aware multi-source multicast



(a) NFV multicast

(b) NFV multi-source multicast with delay constraint

(c) NFV multicast with reliability constraint

(d) NFV multi-source multicast with reliability and delay constraint

FIGURE 1. Illustrative example of Hybrid Routing for NFV multicast.

in NFV-enabled network consists are four folds: 1) the necessary VNF backups; 2) the optimal VNF placements; 3) the optimal multicast sources selection; 4) the optimal routing from multicast source to multicast destinations.

B. NETWORK MODEL

We study the reliability aware multi-source multicast resource optimization model in NFV-enabled network. The reliability of NFV multicast can be guaranteed by providing VNF backups. We model the substrate network as a directed graph which consists of a set of N Physical Nodes (PNs) and a set of L Physical Links (PLs) interconnecting the PNs. The set of VNFs $F = \{f_1, f_2, ..\}$ is defined to describe different types of VNFs (e.g., intrusion detection, firewall, hardware accelerator, video compression/decompression, etc.). We assume PN k ($k \in [1, |N|]_z$) is equipped with a Physical Machine (PM) and a switch which could be used to process the VNFs/backup VNFs and provides the PLs. And each PM k ($k \in [1, |N|]_z$) has a computing capacity $C_k \geq 0$ for processing different types of VNFs. The reliability of PN k is denoted as $k.reliability$. The computing resource consumption is defined as c_{f_j} for the VNF implementation of f_j. Each PL m ($m \in [1, |L|]_z$) has a available bandwidth $B_m \geq 0$ and a transmission delay δ_m for multicast network services. In our work, the reliability of PL is not considered, since it could be transformed as a part of PN reliability. The set of multicast network services is denoted as R. Each multicast service r ($r \in [1, |R|]_z$) is modelled as a 6-tuple $(S_r, D_r, F_r, b_r, \Theta_r, Re_r)$, where the S_r and D_r are the candidate source node set and the destination node set. F_r denotes the ordered VNFs that the multicast network service r must be processed. b_r and Θ_r denote the link bandwidth consumption and end-to-end delay requirement of multicast network service r. And Re_r denotes the reliability requirement of multicast r for all destinations in D_r.

III. MATHEMATICAL MODELLING

A. PROBLEM FORMULATION

In our reliability aware multi-source multicast resource optimization model, we consider the optimal backup node selection and hybrid routing to minimize the total link bandwidth consumption and guarantee the reliability and end-to-end delay for each destination node. We first introduce the VNF backups in NFV multicast. For each destination in multicast r, there is a set of VNFs F_r must be processed in order. The ordered VNFs in multicast r is described as f_{rj}, where j denotes the index of j^{th} VNF in F_r. We further define the number of VNF/backups f_{rj} for source $s (s \in S_r)$ and destination $d (d \in D_r)$ in multicast r as NUM^{rj}_{sd}. With the consideration of VNF backups, the routing from source to destination in one multicast includes serial, parallel and hybrid paths. In order to formulate the hybrid paths with VNF backups, we next decompose the end-to-end path from source to destination into multiple virtual paths. The number of virtual paths from s to d in r is calculated as $NUMP^r_{sd} = \prod_{j \in [1, |F_r|]_z} NUM^{rj}_{sd}$. We define two binary variable x^{rsd}_{ijk} and y^{rsd}_{ijm} to indicate the VNF placements and routing selection for each pair of source-destination nodes ($s, d, s \in [1, |S|]_z$, $d \in [1, |D|]_z$) in multicast network service r. They are:

$$x^{rsd}_{ijk} = \begin{cases} 1 & \text{if the } j^{th} \text{ VNF instance of } r \text{ is hosted on } PN_k \\ & \text{in virtual path } i \text{ for a source-destination node} \\ & \text{pair } (s, d). \\ 0 & \text{otherwise.} \end{cases}$$

$$y^{rsd}_{ijm} = \begin{cases} 1 & \text{if the } j^{th} \text{ VNF instance of } r \text{ is using link } m \\ & \text{in virtual path } i \text{ to receive traffic from uplink} \\ & \text{VNF for a source-destination node pair} (s, d). \\ 0 & \text{otherwise.} \end{cases}$$

Note that PNs and PLs are able to be reused in multiple virtual paths with the same source and the same destination in one multicast. Thus, two binary variable p_{jk}^{rsd} and q_{jm}^{rsd} are defined to indicate the real VNF placements and routing selection for a source-destination pair (s, d). They are:

$$p_{jk}^{rsd} = \begin{cases} 1 & \text{if the } j^{th} \text{ VNF instance of } r \text{ is hosted on } PN_k \\ & \text{for source-destination pair } (s, d). \\ 0 & \text{otherwise.} \end{cases}$$

$$q_{jm}^{rsd} = \begin{cases} 1 & \text{if the } j^{th} \text{ VNF instance of } r \text{ is using link } m \\ & \text{to receive traffic from uplink VNF for} \\ & \text{source-destination pair } (s, d). \\ 0 & \text{otherwise.} \end{cases}$$

For multiple destinations in one multicast, PNs and PLs could be reused in multicast paths. We define three binary variable u_{jk}^{rs}, v_{jm}^{rs} and w_m^{rs} to indicate the VNF placement and routing selection for multicast r. They are:

$$u_{jk}^{rs} = \begin{cases} 1 & \text{if the } j^{th} \text{ VNF instance of } r \text{ is hosted on } PN_k \\ & \text{for source } s. \\ 0 & \text{otherwise.} \end{cases}$$

$$v_{jm}^{rs} = \begin{cases} 1 & \text{if the } j^{th} \text{ VNF instance of } r \text{ is using link } m \\ & \text{to receive traffic from uplink VNF for} \\ & \text{source } s. \\ 0 & \text{otherwise.} \end{cases}$$

$$w_m^{rs} = \begin{cases} 1 & \text{if } r \text{ is using link } m \text{ to receive traffic} \\ & \text{for source } s. \\ 0 & \text{otherwise.} \end{cases}$$

In our work, once the number of backups for each type of VNF is determined, the objective of resource optimization for hybrid routing is to minimize bandwidth consumption. Thus, the objective is mathematically formulated as:

$$Obj : Min\{\sum_r \sum_s \sum_m w_m^{rs} b_r\} \tag{1}$$

Next, we define a binary variable z_{sd}^r to denote the source-destination pair (s, d) of one multicast service r, such that:

$$z_{sd}^r = \begin{cases} 1 & \text{if there is a source-destination pair } (s, d) \\ & \text{for a multicast service } r. \\ 0 & \text{otherwise.} \end{cases}$$

If the destination d of multicast service r is served, there is at most one source node s is activated for d. It is formulated as:

$$\sum_{s \in [1, |S_r|]_z} z_{sd}^r = 1 (\forall d \in [1, |D_r|]_z) \tag{2}$$

According to the definitions of virtual path variables p_{jk}^{rsd}, q_{jm}^{rsd}, for each source-destination pair (s, d) in multicast service r, the formulation of VNF placement and routing selection has the following constraints:

$$x_{ijk}^{rsd} \le p_{jk}^{rsd} (\forall r, s, d, i, j, k) \tag{3}$$

$$\sum_i x_{ijk}^{rsd} \ge p_{jk}^{rsd} (\forall r, s, d, j, k) \tag{4}$$

$$y_{ijm}^{rsd} \le q_{jm}^{rsd} (\forall r, s, d, i, j, m) \tag{5}$$

$$\sum_i y_{ijm}^{rsd} \ge q_{jm}^{rsd} (\forall r, s, d, j, m) \tag{6}$$

The definitions of multicast path variables u_{jk}^{rs}, v_{jm}^{rs} and w_m^{rs} must follow the VNF placements and routing selections from virtual path variables p_{jk}^{rsd}, q_{jm}^{rsd} for each source s in multicast service r. The formulation of VNF placement and routing selection of multicast path has the following constraints:

$$p_{jk}^{rsd} \le u_{jk}^{rs} (\forall r, s, d, j, k) \tag{7}$$

$$\sum_d p_{jk}^{rsd} \ge u_{jk}^{rs} (\forall r, s, j, k) \tag{8}$$

$$q_{jm}^{rsd} \le v_{jm}^{rs} (\forall r, s, d, j, m) \tag{9}$$

$$\sum_d q_{jm}^{rsd} \ge v_{jm}^{rs} (\forall r, s, j, m) \tag{10}$$

$$v_{jm}^{rs} \le w_m^{rs} (\forall r, s, j, m) \tag{11}$$

$$\sum_j v_{jm}^{rs} \ge w_m^{rs} (\forall r, s, m) \tag{12}$$

Based on the network flow balance constraint, the incoming traffic must equal the outgoing traffic at each PN in each virtual path i from source to destination. According to the model of Layered Graph [16], it is mathematically formulated as:

$$\sum_{m.tail=k} y_{ijm}^{rsd} - \sum_{m'.head=k} y_{ijm}^{rsd} = (x_{ijk}^{rsd} - x_{i(j-1)k}^{rsd}) z_{sd}^r \tag{13}$$

Note that the above flow balance constraint handles the order of VNF processing within one virtual path. In order to make sure all VNFs/backups from different virtual paths have the correct VNF processing order, we formulate the following VNF order constraint. It forces no connection between current (j) and previous ($j-1$) VNF/backup ($q_{jm}^{rsd} = 0$) if the next VNF/backup ($j+1$) is placed at the head of current link ($p_{(j+1)m.head}^{rsd} = 1$).

$$q_{jm}^{rsd} \le 1 - p_{(j+1)m.head}^{rsd} \tag{14}$$

To meet the reliability requirement, there are at least NUM_{sd}^{rj} j^{th} VNF/backups for the available source-destination pair ($z_{sd}^r = 1$) of multicast r. It is formulated as:

$$\sum_k p_{jk}^{rsd} = NUM_{sd}^{rj} z_{sd}^r \tag{15}$$

$$\sum_j p_{jk}^{rsd} \le 1 \tag{16}$$

$$\sum_k x_{ijk}^{rsd} \le 1 \tag{17}$$

In order to separate the reliability calculation of different types of VNF/backups, all VNF/backups (e.g., j^{th} VNF/backups) from all virtual paths must have connections between j^{th} and $(j+1)^{th}$ VNF/backups. We define a binary h_{ijk}^{rsd} to indicate the virtual path selection for j^{th} VNF of source-destination pair (s, d) in multicast r. Here, k denotes the VNF placement of $(j+1)^{th}$ VNF. We further introduce a new binary $H^{rsd}_{ii'jk}$ to describe the connections between two virtual paths i and i'. The constraints are formulated as follows:

$$\sum_{i}\sum_{m.tail=k} y^{rsd}_{ijm} h^{rsd}_{i(j-1)k} \ge NUM^{r(j-1)}_{sd} p^{rsd}_{jk} \tag{18}$$

$$\sum_{i} h^{rsd}_{i(j-1)k} \ge NUM^{r(j-1)}_{sd} p^{rsd}_{jk} \tag{19}$$

$$h^{rsd}_{ijk} h^{rsd}_{i'jk} \le H^{rsd}_{ii'jk}(i \neq i') \tag{20}$$

$$x^{rsd}_{ijk} x^{rsd}_{i'jk} \le 1 - H^{rsd}_{ii'jk'}(i \neq i') \tag{21}$$

The bandwidth and computing resources constraints are formulated as:

$$\sum_{r}\sum_{j}\sum_{s} u^{rs}_{jk} c_{f_{rj}} \le C_k \tag{22}$$

$$\sum_{r}\sum_{s} w^{rs}_{m} b_r \le B_m \tag{23}$$

We consider the end-to-end delay constraint for each destination d in multicast service r. Let a variable n^{rsd}_{i} denotes the end-to-end delay of virtual path i from source node s to destination d in r.

$$n^{rsd}_{i} \ge \sum_{m}\sum_{j} y^{rsd}_{ijm} \delta_m \tag{24}$$

$$n^{rsd}_{i} \le \Theta_r \tag{25}$$

B. RELIABILITY GUARANTEE

In NFV-enabled multicast model, we provide each destination with enough VNF backups to guarantee the reliability of the service. However, more computing resources will be consumed with the increase of the number of VNF backups. Thus, it is reasonable to guarantee the reliability for each destination with minimal number of VNF backups. Recall that each destination requires a chain or a sequence of VNF processing (e.g., $f1 \rightarrow f2$ in Fig.1). And any failure in each type of VNF leads to the collapse of the entire chain. An efficient way to improve the chain reliability is by adding a backup for the VNF with the weakest reliability [15]. Next, we need to determine which type of VNF will be the weakest. Since the VNF placements are unknown, the worst case of reliability calculation must be analysed. For each service chain, the worst case of reliability is the placements of VNF with least number of backups in weakest nodes. For example, there are 5 nodes with reliabilities $(0.9-0.94)$: $n1(0.9)$, $n2(0.91)$, $n3(0.92)$, $n4(0.93)$, $n5(0.94)$. And the service chain is assumed as $f1 \rightarrow f2$. We further assume the numbers of $f1$ and $f2$ are 3 and 2, respectively. The worst case of VNF placement should be $n1(f2)$, $n2(f2)$, $n3(f1)$, $n4(f1)$, $n5(f1)$. The reliability of the worst case could be calculated as $(1-((1-0.9)\times(1-0.91)))\times(1-((1-0.92)\times(1-0.93)\times(1-0.94))) = 0.99$. If the reliability requirement is 0.991, the best way is to provide $f1$ with an additional backup. The above procedure is shown in Algorithm 1. First, we need to generate a new list of nodes

---

Algorithm 1 Redundant VNF/Backup Calculation With Reliability Guarantee

1 **Initialization:**
2 Substrate Network (N and L) and Multicast Services (R);
3 $NUM_{rj} := 1(\forall r, j)$;
4 $VNF\_reliability_{rj} := 1(\forall r, j)$;
5 $SFC\_reliability_r := 1(\forall r)$;
6 Generate new List of Nodes LN in ascending order of reliability value;
7 **while** *true* **do**
8 $isFinished := 1$;
9 **for** $r = 1 : |R|$ **do**
10 Generate new List of VNFs LV_r in ascending order of the value of NUM;
11 $count = 1$;
12 **for** $j = 1 : |LV_r|$ **do**
13 **for** $n = 1 : NUM_{rj}$ **do**
14 $VNF\_reliability_{rj} := VNF\_reliability_{rj} \times (1 - LN(count).reliability)$;
15 count++;
16 **end**
17 $VNF\_reliability_{rj} := 1 - VNF\_reliability_{rj}$; $SFC\_reliability_r := SFC\_reliability_r \times VNF\_reliability_{rj}$;
18 **end**
19 **if** $SFC\_reliability_r < Re_r$ **then**
20 Select the VNF j with the smallest value of NUM;
21 $NUM_{rj}++$;
22 $isFinished := 0$;
23 **end**
24 **end**
25 **if** $isFinished == 1$ **then**
26 break;
27 **end**
28 **end**

---

in ascending order of reliability values in Line 6. Next, for each multicast session r, different types of VNFs must be reordered based on the numbers of VNF backups in line 10. In line 12-18, the worst case of reliability is calculated. If the current reliability value does not meet the reliability requirement, the algorithm select the VNF with the least number and provide it with an additional backup in line 19-23. Based on the VNF backup number NUM from Algorithm 1, the optimal VNF placement and hybrid routing in NFV-enabled multicast can be solved by the MILP formulation: Objective: (1); Constraints: (2) - (25).

C. GREEDY ALGORITHM

In this subsection, we introduce a K-shortest path-based greedy algorithm to reduce the complexity of the proposed MILP solution. According to the MILP formulation in Section III-A, the complexity mainly comes from the connections between neighbour VNFs to guarantee the chain reliability (e.g., (18)-(21)). Thus, to reduce the complexity, the original optimization problem is decomposed into three sub-problems. They are: a) Multicast source selection; b) Hybrid routing with delay constraints and c) VNF placements for backups. For solving the first sub-problem, we proposed Algorithm 3 to provide multicast destinations with minimal number of multicast sources. In the second sub-problem, K-shortest-paths algorithm is used to find available paths with reliability and delay constraints for hybrid routing (line 11 - 17 in Algorithm 2 and Algorithm 5). Finally, the third sub-problem of VNF placements is solved in line 24 in Algorithm 2.

In Algorithm 2, *dest_path* and *source_path* are defined to indicate candidate source nodes for destination d (e.g., the end-to-end delay $\leq$ the delay requirement of multicast session r) and destination nodes which can be served by each source s, respectively. Note that, not all source node in $dest\_path(d)$ is necessary for multicast destination d. The minimal number of multicast sources is determined by Algorithm 3. In Algorithm 3, for each destination, we first introduce two sets *all_source_set* and *all_dest_set* to store all available sources and all destinations (line 7 - 8). Next, the source s with max count in *all_source_set* is selected and remove all destinations which s could be served from *all_dest_set* (line 14 - 16). The above selection procedure is stopped when all destinations have been served (line 11 -13). Once the connections between multicast sources and destinations are determined, the candidate paths [17] must be provided for each destination (line 14 in Algorithm 2). Recall that, there are VNF/backups which need to be placed at these candidate paths. Some of paths may not have enough nodes for hosting VNF/backups. Thus, we next find available paths in *k_paths* according to Algorithm 4. The basic idea of Algorithm 4 is checking and extending the path to make sure it has enough nodes for hosting VNF/backups. According to the section of reliability guarantee, the available path must be able to provide $\sum_j NUM_{rj}$ available nodes for VNF/backups placements (line 7 - 15 in Algorithm 4). If the current path is not available for VNF placement due to the lack of available nodes, the algorithm add new parallel path into the current path (line 9 - 11 in Algorithm 4). The purpose of new added node/path which is close to source is to provide enough computing resources for VNF placements.

For each destinations, it only needs one available path from multicast source in order to reduce the link bandwidth resource consumption. And nodes and links in these available paths could be reuse if they provide services for the same multicast source. Therefore, we need to select one available path for each multicast destination with maximal usage of nodes and links. The selection is based on the new defined weight of links and Algorithm 5. The weight of link indicates the count of link shown in all available paths. In order to find the central path (e.g., most nodes and links could be used for multiple multicast destinations) from all available paths, in Algorithm 5, we select the path with largest weight from all available paths.

Algorithm 2 K-Shortest Path-Based Greedy Algorithm for NFV Multicast Routing

1 **Input:**
2   Substrate Network (N and L) and Multicast Services (R);
3     $NUM_{rj}(\forall r, j)$ from **Algorithm 1**;
4 **Output:**
5   NFV Multicast Routing and VNF placements in R;
6 **for** $r = 1 : |R|$ **do**
7   Update the network topology based on the available link bandwidth and the bandwidth requirement of multicast network service r;
8     $dest\_path(d)$ = find candidate source nodes (end-to-end delay $\leq \Theta_r$) for each destination d of multicast network service r;
9     $source\_path(s)$ = find destination nodes which can be served by each multicast source node s according to **Algorithm 3**;
10   %% **VNF placement and multicast routing** %%
11   **for** $s = 1 : |source\_path|$ **do**
12   Update the network status and add an empty link set of *all_links*;
13   **for** $d = 1 : |source\_path(s).dest|$ **do**
14             $k\_paths(d)$ = find K-shortest paths from $source\_path(s)$ to $source\_path(s).dest(d)$;
15             $available\_vnf\_nodes(d)$ = find available nodes in *k_paths* for VNF placements of F_r with an ascending order of distance from $source\_path(s)$;
16             $available\_paths(d)$ = find available paths in *k_paths* with enough number of available nodes according to **Algorithm 4**; $all\_links.add($ find all links from $available\_paths(d))$;
17   **end**
18   %% **Calculate the weights of links in available paths** %%
19   **for** $m = 1 : |all\_links|$ **do**
20             $all\_links(m).weight$ = the count of link m in all available paths;
21   **end**
22   **for** $d = 1 : |source\_path(s).dest|$ **do**
23             $central\_path(d)$ = find central path of d based on the weight of *all_links* according to **Algorithm 5**;
24   do VNF placement in $central\_path(d)$ and minimize the VNF and path usage according to multicast constraints;
25   **end**
26   **end**
27 **end**

Algorithm 3 Find Destinations Nodes for Multicast Sources

Input:
Multicast Services (R);
$dest\_path(d)$ from **Algorithm 1**;
Output:
source_path(s) for all multicast sources;
for $d = 1 : |dest\_path|$ **do**
  $all\_source\_set.add$(find all available sources from $dest\_path(d)$);
  $all\_dest\_set.add$(find all destinations from $dest\_path(d)$);
end
while *true* **do**
  if *is empty(all_dest_set)* **then**
    break;
  end
  $s =$ find source with the max count in all_source_set;
  $source\_path(s).add$(find all destinations which can be served by source s);
  remove $source\_path(s)$ from all_dest_set;
end

Algorithm 4 Find Available Paths With Reliability Constraints

Input:
Substrate Network (N and L) and Multicast Services (R);
$NUM_{rj}(\forall r, j)$, k_paths, $available\_vnf\_nodes$ from **Algorithm 1**;
Output:
$available\_paths$ for each destination;
for $p = 1 : |k\_paths|$ **do**
  if $available\_vnf\_nodes(d, p) < \sum_j NUM_{rj}$ **then**
    %% **add new VNF node into path** %%
    $new\_vnf\_nodes =$ find nodes close to source with enough computing resources;
    Generate new path with new_vnf_nodes;
    Replace p with new path in $available\_paths(d)$ if the delay constraints are met;
  end
  else
    $available\_paths(d).add(p)$;
  end
end

Algorithm 5 Find Central Path for Each Destinations Based on Link Weight

Input:
Substrate Network (N and L) and Multicast Services (R);
all_links, $available\_paths$, $source\_path$ from **Algorithm 1**;
Output:
$central\_path$ for each destination;
for $d = 1 : |source\_path(s).dest|$ **do**
  for $p = 1 : |available\_paths(d)|$ **do**
    $path\_weight(d, p) = \sum_{m \in available\_paths(p)} all\_links(m).weight$;
  end
  $central\_path(d) =$ find the path from $available\_paths(d)$ with largest weight.
end

In Algorithm 2, to find candidate source nodes for each destination in multicast service r, Dijkstra's shortest path algorithm is implemented. Therefore, the complexity for finding $dest\_path(d)$ and $source\_path(s)$ in Algorithm 2 is $O(|S_r||D_r|N^2) = O(N^4)$ ($|S_r| \leq N$ and $|D_r| \leq N$), where N is the number of nodes in network. Note that the complexity of VNF placement and multicast routing comes from K-shortest path algorithms for finding available/central paths. According to [17], the complexity could be calculated as $O(|source\_path||source\_path.dest|KN(M + NlogN)) = O(KN^3(M + NlogN))$ ($|source\_path| \leq N$ and $|source\_path.dest| \leq N$), where M is the amount of edges and K is the size of initial set of paths. In conclusion, the complexity of algorithm is $O(KN^3(M + NlogN))$ for the process of one multicast service.

IV. NUMERICAL RESULTS

In this section, we evaluate the performance of the proposed Reliability-aware Multi-source multicast hybrid Routing solutions with Delay constraints (RMRD-MILP and RMRD-Greedy). The performance of RMRD-MILP and RMRD-Greedy are compared with 1) Multi-source multicast hybrid Routing without reliability constraints (MR); 2) Reliability-aware Single-source multicast hybrid Routing solutions with Delay constraints (RSRD). We use CPLEX to solve the formulation of RMRD-MILP. All simulations are conducted on a physical machine equipped with an Intel 3.2 GHz processor and 24 GB RAM. Three different network topologies are considered, namely: 1) a small network shown in Figure 2 composed of 12 PNs (4 × 3), 2) a medium network composed of 20 PNs (5 × 4) and 3) a large network composed of 40 PNs (8 × 5). In simulations, we assume there are four different types of VNFs ($f1, f2, f3, f4$). And each multicast session will randomly choose two of them for VNF processing. The computing resource consumptions of these four VNFs are {2, 2, 4, 4} units. Each node is assumed to have a computing capacity of 20 units. And the reliabilities of nodes are randomly generated between 0.9 and 0.92. Each link is assumed to have available bandwdith and link delay of 20 and 10 units. Due to the complexity of the proposed RMRD-MILP solution, we only implement it on a small network hosting 1 multicast session with

TABLE 1. Routing results for a 12-node network.

Algorithm	Routing and VNFs assignment (VMs (VNFs))	Rel.	Del. (units)	Band. Util.	Comp. Util.	CPU time (s)
RMRD-MILP($1 \to 8$)	$1 \to 2(f_1) \to 6(f_3) \to 7(f_3) \to 8$	0.9830	40	2.06%	5%	5383
	$1 \to 5(f_1) \to 6(f_3)$					
RMRD-MILP($1 \to 11$)	$1 \to 2(f_1) \to 6(f_3) \to 7(f_3) \to 11$					
	$1 \to 5(f_1) \to 6(f_3)$					
RMRD-Greedy($1 \to 8$)	$1 \to 5(f_1) \to 6(f_3) \to 7(f_3) \to 8$	0.9830	40	2.06%	5%	0.1352
	$1 \to 2(f_1) \to 6(f_3)$					
RMRD-Greedy($1 \to 11$)	$1 \to 5(f_1) \to 6(f_3) \to 7(f_3) \to 11$					
	$1 \to 2(f_1) \to 6(f_3)$					

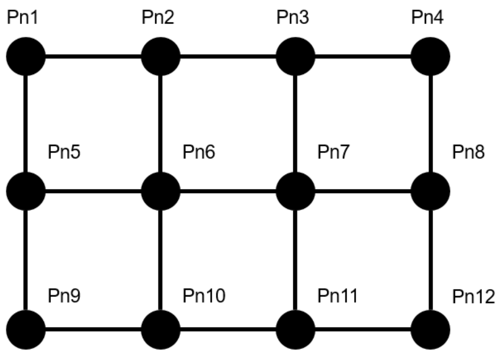


FIGURE 2. A small network composed of 12 PNs.

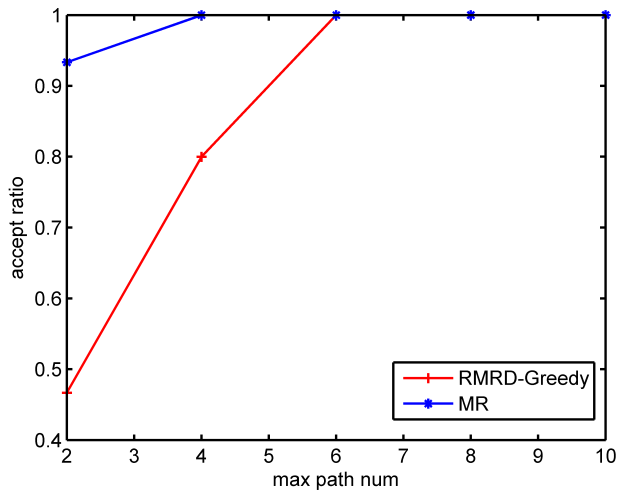


FIGURE 3. Percentage of admitted multicast destinations versus max path num (20-nodes network with 5 multicast services and a per-service reliability requirement of 0.98).

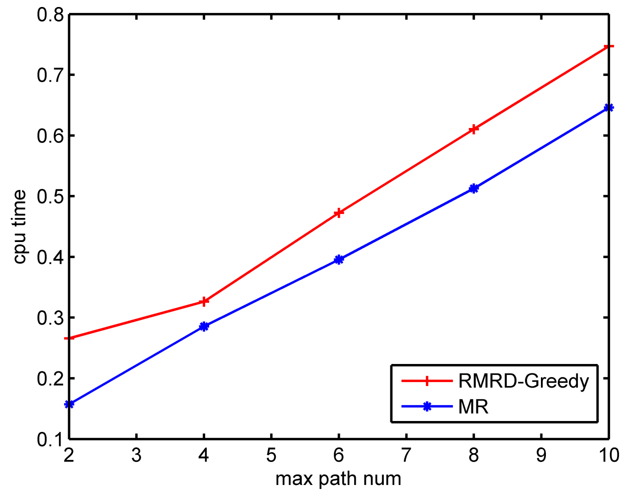


FIGURE 4. CPU time (s) versus max path num (20-nodes network with 5 multicast services and a per-service reliability requirement of 0.98).

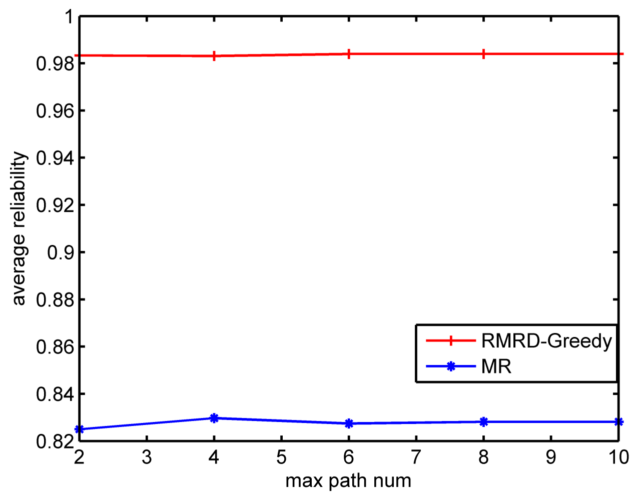


FIGURE 5. Average reliability versus max path num (20-nodes network with 5 multicast services and a per-service reliability requirement of 0.98).

2 sources and 2 destinations. In small network, to facilitate the demonstration of algorithm performance, the multicast sources and destinations are set as {1, 3}, {11, 8}, respectively. The end-to-end delay requirement for all destinations is 40 units. The reliability and bandwidth requirements of multicast session are repectively set to 0.98 and 2 units. The service chain is assumed as $f1 \to f3$. The performance comparisons of RMRD-MILP and RMRD-Greedy are shown in Table 1.

According to Table 1, both RMRD-MILP and RMRD-Greedy (K = 10) are able to find available hybrid paths and guarantee the reliability and delay of multicast destinations. And the performances of computing utilization and bandwidth utilization of RMRD-Greedy are same as RMRD-MILP. However, the CPU times of RMRD-MILP and RMRD-Greedy are 5383 seconds and 0.1352 seconds, respectively. We observe that the proposed RMRD-Greedy achieves the same performance to RMRD-MILP in 12-node network within a negligible runtime. Therefore, RMRD-Greedy will be used in the sequel for the performance evaluations for large network.

Next, we evaluate the performances of RMRD-Greedy in 20-node network with 5 multicast sessions. And each

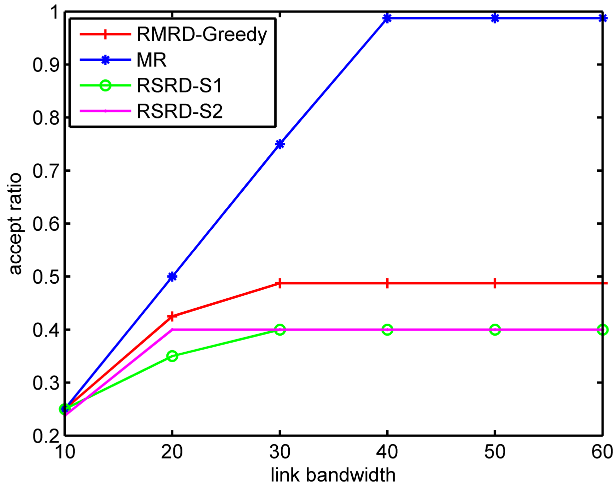


FIGURE 6. Percentage of admitted multicast destinations versus link bandwidth (40-nodes network with 20 multicast services and a per-service reliability requirement of 0.99).

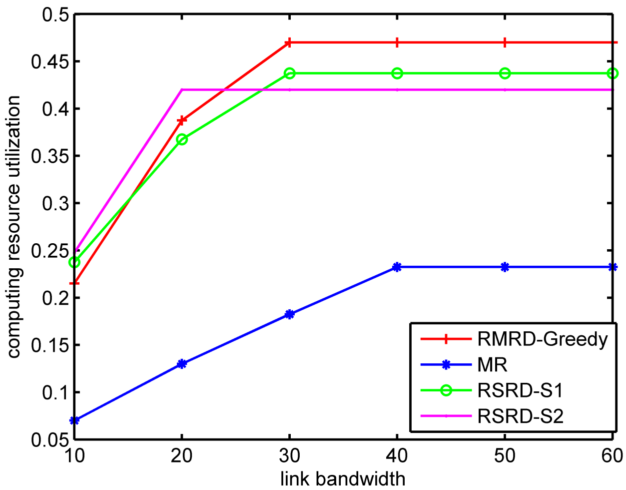


FIGURE 7. Computing resource utilization versus link bandwidth (40-nodes network with 20 multicast services and a per-service reliability requirement of 0.99).

multicast session is equipped with 2 sources and 3 destinations. For each destinations, the reliability and delay requirements are fixed to 0.98 and 40 units. We vary the *max_path_num* (K-shortest paths) in the range of 2 to 10. We define the accept ratio as the number of admitted multicast destinations divided by the number of all destinations. According to Figure 3, it is clear that more multicast services/destinations will be admitted with the increase of the *max_path_num*. If *max_path_num* $\geq$ 6 is selected, both RMRD-Greedy and MR achieve 100% acceptance ratio. In Figure 4, a longer CPU running time will be introduced if we select a larger value of *max_path_num*. Note that MR shows better performances of acceptance ratio and CPU running time. This is because MR needs no backup in service chain. And it needs less paths to do VNF placements. The average reliabilities of RMRD-Greedy and MR are shown in Figure 5. Our results show MR has a worse reliability since the reliability constraints are omitted.

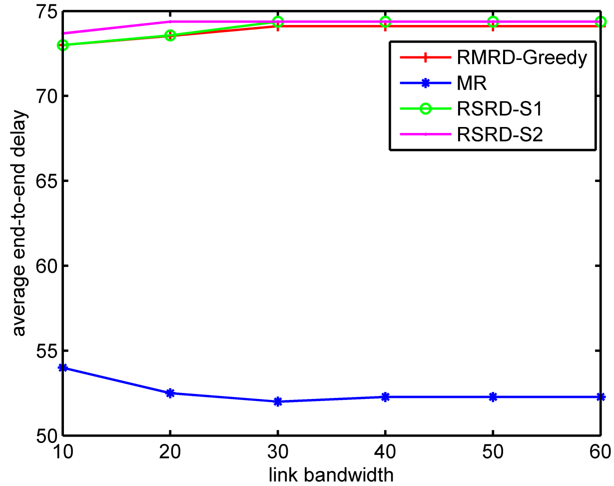


FIGURE 8. Average end-to-end delay versus link bandwidth (40-nodes network with 20 multicast services and a per-service reliability requirement of 0.99).

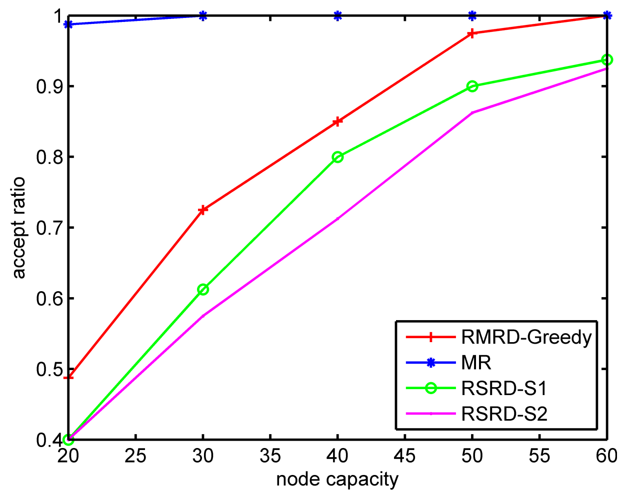


FIGURE 9. Percentage of admitted multicast destinations versus node capacity (40-nodes network with 20 multicast services and a per-service reliability requirement of 0.99, link bandwidth = 40).

In a large network with 40 PNs (*max_path_num* = 40), we first study the performance comparisons of limited link bandwidth. We randomly generate 20 available multicast services. And each of them is equipped with 4 destinations. The reliability and end-to-end delay are set as 0.99 and 80 units, respectively. To make the performance comparison clear, the number of sources in each multicast service is set as 2. And we introduce RSRD-S1 and RSRD-S2 as two different source node selection algorithms. The Figure 6 shows the percentage of admitted multicast destinations versus link bandwidth in the range of 10−60 units. Both RMRD-Greedy, MR, RSRD-S1 and RSRD-S2 achieve stable acceptance ratio when *link_bandwidth* $\geq$ 40 units. Similar to the results in 20-node network, MR has the best performance at the cost of a reliability loss. Due to the limitation of source selection, both RSRD-S1 and RSRD-S2 have less multicast destinations

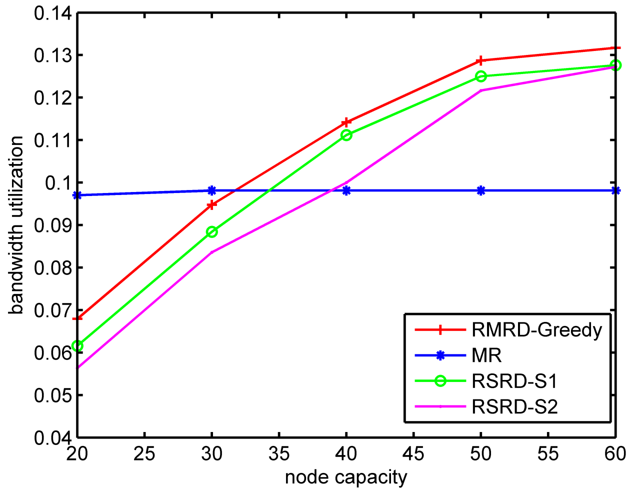


FIGURE 10. Bandwidth utilization versus node capacity (40-nodes network with 20 multicast services and a per-service reliability requirement of 0.99, link bandwidth = 40).

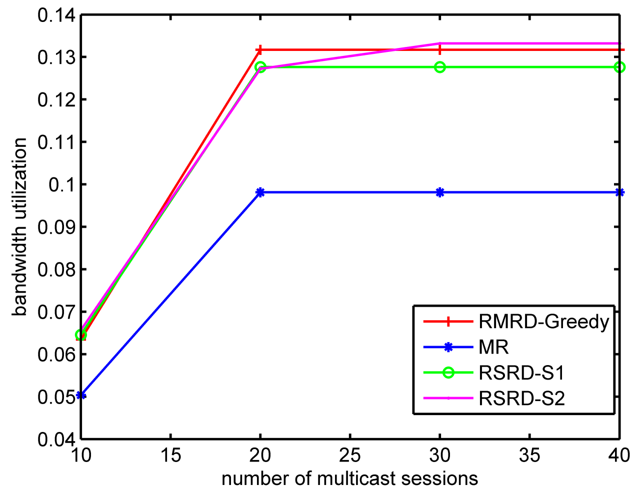


FIGURE 12. Bandwidth utilization versus number of multicast sessions (40-nodes network with 10 – 40 multicast sessions and a per-service reliability requirement of 0.99, link bandwidth = 40, node capacity = 60).

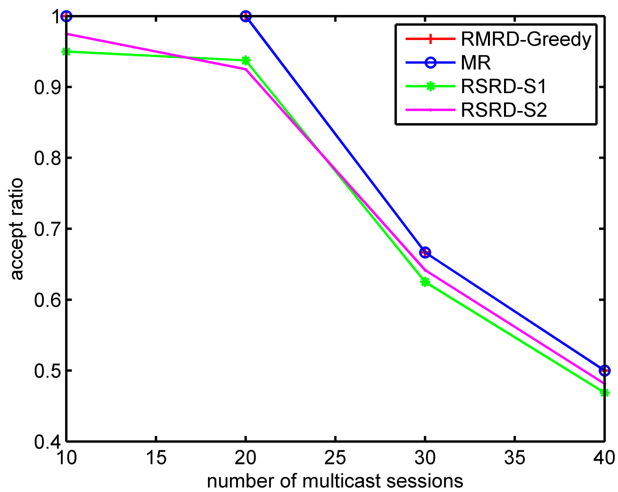


FIGURE 11. Percentage of admitted multicast destinations versus number of multicast sessions (40-nodes network with 10 – 40 multicast sessions and a per-service reliability requirement of 0.99, link bandwidth = 40, node capacity = 60).

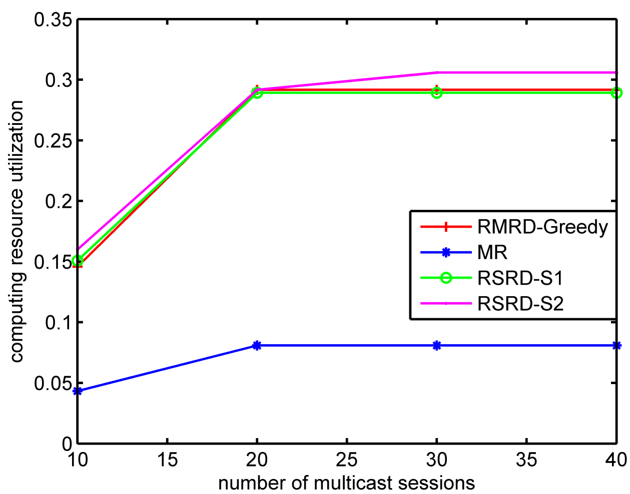


FIGURE 13. Computing resource utilization versus number of multicast sessions (40-nodes network with 10 – 40 multicast sessions and a per-service reliability requirement of 0.99, link bandwidth = 40, node capacity = 60).

been admitted compared to RMRD-Greedy. The computing resource utilization is shown in Figure 7. With the increase of link bandwidth, there are more multicast destinations which have been admitted for both four algorithms. RMRD-Greedy has the largest value of computing resource utilization when $link\_bandwidth \geq 30$ compared to RSRD. This is because RMRD-Greedy has larger value of acceptance ratio. In Figure 8, we show the average end-to-end delay of all four algorithms are smaller than the delay requirement 80 units. Due to the MR has the smallest number of VNFs, it is able to provide multicast service for more destinations with smallest end-to-end delay.

Next, we vary the node capacity in the range of 20 to 60 units and evaluate the performance of the proposed algorithms. To make a fair comparison, the link bandwidth is set as 40 units. In Figure 9, we show the percentage of admitted multicast destinations increased with the increase of node capacity for all four algorithms. Similarly, MR has the best result of acceptance ratio. And RMRD-Greedy outperforms RSRD-S1 and RSRD-S2. Since we fixed the link bandwidth, the comparison of bandwidth utilization is able to indicate the hybrid routing performance. In Figure 10, MR has the largest bandwidth utilization for node capacity in a range of 20 to 30. The reason is that MR has more multicast sessions which have been admitted (Figure 9). If we only consider the case of 100% acceptance ratio (RMRD-Greedy and MR at node capacity of 60), Figure 10 shows an extra 34.25% bandwidth resource consumption of RMRD-Greedy compared to MR. And the reason for a worse bandwidth utilization of RMRD-Greedy compared to RSRD is the larger value of acceptance ratio.

Finally, we fixed the link bandwidth and node capacity and vary the number of multicast sessions from 10 to 40. The results of acceptance ratio of four algorithms are shown in Figure 11. Note that MR and RMRD-Greedy have the same performance. With the increase of multicast sessions, some of them will eventually be dropped due to the lack of bandwidth and computing resources. We compare the bandwidth and computing resource utilization in Figure 12 and 13. The results show that the RMRD-Greedy and RSRD have the very close performances of bandwidth and computing resource utilization in the number of multicast sessions 10 - 20. Note that RMRD-Greedy has a larger value of acceptance ratio compared to RSRD in Figure 11. We conclude that multi-source multicast has a better routing selection compared to single-source multicast due to the more options of multicast sources.

V. CONCLUSION

In this paper, we presented a novel reliability-aware hybrid routing scheme for multi-source multicast in NFV-enabled network. The optimization routing model is formulated as a MILP. And a K-shortest path-based greedy algorithm is proposed to reduce the complexity of MILP formulation. Our numerical results show that the proposed reliability-aware multi-source multicast algorithm outperforms single-source multicast algorithm in terms of acceptance ratio and network resource consumption.

REFERENCES

[1] R. Zhu, D. Niu, B. Li, and Z. Li, "Optimal multicast in virtualized datacenter networks with software switches," in *Proc. IEEE INFOCOM-IEEE Conf. Comput. Commun.*, Atlanta, GA, USA, May 2017, pp. 1–9.

[2] Z. Xu, W. Liang, M. Huang, M. Jia, S. Guo, and A. Galis, "Efficient NFV-enabled multicasting in SDNs," *IEEE Trans. Commun.*, vol. 67, no. 3, pp. 2052–2070, Mar. 2019.

[3] N. Akhtar, I. Matta, A. Raza, L. Goratti, T. Braun, and F. Esposito, "Virtual function placement and traffic steering over 5G multi-technology networks," in *Proc. 4th IEEE Conf. Netw. Softwarization Workshops (NetSoft)*, Montreal, QC, Canada, Jun. 2018, pp. 114–122.

[4] F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and O. C. M. B. Duarte, "Orchestrating virtualized network functions," *IEEE Trans. Netw. Service Manag.*, vol. 13, no. 4, pp. 725–739, Dec. 2016.

[5] W. John, K. Pentikousis, G. Agapiou, E. Jacob, M. Kind, A. Manzalini, F. Risso, D. Staessens, R. Steinert, and C. Meirosu, "Research directions in network service chaining," in *Proc. IEEE SDN Future Netw. Services (SDN4FNS)*, Trento, Italy Nov. 2013, pp. 1–7.

[6] N.-T. Dinh and Y. Kim, "An efficient reliability guaranteed deployment scheme for service function chains," *IEEE Access*, vol. 7, pp. 46491–46505, 2019.

[7] B. Addis, D. Belabed, M. Bouet, and S. Secci, "Virtual network functions placement and routing optimization," in *Proc. IEEE 4th Int. Conf. Cloud Netw. (CloudNet)*, Niagara Falls, ON, Canada, Oct. 2015, pp. 171–177.

[8] S. Q. Zhang, Q. Zhang, H. Bannazadeh, and A. Leon-Garcia, "Routing algorithms for network function virtualization enabled multicast topology on SDN," *IEEE Trans. Netw. Service Manag.*, vol. 12, no. 4, pp. 580–594, Dec. 2015.

[9] B. Ren, D. Guo, G. Tang, X. Lin, and Y. Qin, "Optimal service function tree embedding for NFV enabled multicast," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Vienna, Austria, Jul. 2018, pp. 132–142.

[10] B. Yi, X. Wang, M. Huang, and A. Dong, "A multi-stage solution for NFV-enabled multicast over the hybrid infrastructure," *IEEE Commun. Lett.*, vol. 21, no. 9, pp. 2061–2064, Sep. 2017.

[11] Y. Ma, W. Liang, J. Wu, and Z. Xu, "Throughput maximization of NFV-enabled multicasting in mobile edge cloud networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 2, pp. 393–407, Feb. 2020.

[12] O. Alhussein, P. T. Do, J. Li, Q. Ye, W. Shi, W. Zhuang, X. Shen, X. Li, and J. Rao, "Joint VNF placement and multicast traffic routing in 5G core networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Abu Dhabi, United Arab Emirates, Dec. 2018, pp. 1–6.

[13] H. Ren, Z. Xu, W. Liang, Q. Xia, P. Zhou, O. F. Rana, A. Galis, and G. Wu, "Efficient algorithms for delay-aware NFV-enabled multicasting in mobile edge clouds with resource sharing," *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 9, pp. 2050–2066, Sep. 2020.

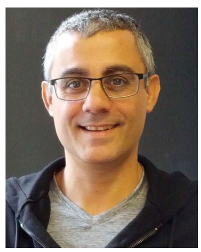
[14] L. Qu, C. Assi, and K. Shaban, "Delay-aware scheduling and resource optimization with network function virtualization," *IEEE Trans. Commun.*, vol. 64, no. 9, pp. 3746–3758, Sep. 2016.

[15] L. Qu, M. Khabbaz, and C. Assi, "Reliability-aware service chaining in carrier-grade softwarized networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 3, pp. 558–573, Mar. 2018.

[16] A. Tomassilli, N. Huin, F. Giroire, and B. Jaumard, "Resource requirements for reliable service function chaining," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Kansas City, MO, USA, May 2018, pp. 1–7.

[17] J. Y. Yen, "Finding the k shortest loopless paths in a network," *Manag. Sci.*, vol. 17, no. 11, pp. 712–716, Jul. 1971.

LONG QU (Member, IEEE) received the Ph.D. degree in communication and information system from Ningbo University, Ningbo, China, in 2015. He was a Visiting Ph.D. Student with Concordia University, Montreal, QC, Canada, from 2012 to 2013. He was a Senior Research Assistant with Qatar University, Doha, Qatar, from 2015 to 2016. He is currently a Lecturer with Ningbo University. His current research interests include cross-layer design in wireless communication systems and virtual network optimization.



CHADI ASSI (Fellow, IEEE) received the B.Eng. degree from Lebanese University, Beirut, in 1997, and the Ph.D. degree from the Graduate Center, The City University of New York, New York City, NY, USA, in April 2003. He is currently a Professor with Concordia University, where he holds a Tier I University Research Chair. Before joining Concordia, he was a Visiting Scientist at the Nokia Research Center, Boston, for one year 2002–2003, working on quality-of-service in optical access networks. He is also supervising a group of 14 Ph.D. students and four M.A.Sc. students, and has successfully supervised 18 Ph.D. students and 25 M.A.Sc. students. His students received very prestigious awards from NSERC and FQRNT. His current research interests are in the general areas of networks, network design and modeling, network optimization, resource virtualization and network, and cyber security. He received the prestigious Mina Rees Dissertation Award from The City University of New York, in August 2002, for his research on wavelength-division-multiplexing optical networks and lightpath provisioning. He held a Tier II University Chair at Concordia, from 2012 to 2017, in the area of wireless networks. He is on the Editorial Board of the IEEE Communications Surveys and Tutorials. He serves as an Associate Editor for the IEEE Transactions on Vehicular Technology, the IEEE Transaction on Communications, the IEEE Transactions on Mobile Computing, and the IEEE Transactions on Network and Service Management.