Virtual Network Mapping for Reliable Multicast Services with Max-Min Fairness

Xiujiao Gao[1], Weida Zhong[1], Zilong Ye[1], Yangming Zhao[2], Jingyuan Fan[1],
Xiaojun Cao[3], Hongfang Yu[2], Chunming Qiao[1]
[1]Dept. of Computer Science & Engineering, SUNY-Buffalo, Buffalo, NY USA 14260
[2]School of Communication and Information Engineering, UESTC, Chengdu, Sichuan, China 611731
[3]Dept. of Computer Science & Engineering, Georgia State University, Atlanta, GA USA 30302
{xiujiaog, weidazho, zilongye, jfan5,qiao}@buffalo.edu, cao@cs.gsu.edu, {zhaoyangming.uestc, yuhf2004}@gmail.com

***Abstract*— Network Function Virtualization (NFV) provides an effective way to reduce the network provider's cost by allowing multiple Virtual Networks (VNs) to share the underlying physical infrastructure. In the NFV environment, especially when supporting multicast service over the VNs, reliability is a critical requirement in the process of VN mapping since the failure of one virtual node can cause the malfunction of all the subsequent nodes that receive multicasting data from it. In this paper, for the first time, we study how to efficiently map VNs for reliable multicast services, while taking into consideration the max-min fairness of the reliability among distinct VNs. We propose a Mixed Integer Linear Programming (MILP) model to determine the upper bound on the max-min fairness reliability. In addition, an efficient heuristic, namely Uniform Reliability Mutation based Genetic (URMG) algorithm, is developed to address reliable multicast VN mapping with a low computational complexity. By encoding multicast tree construction and link mapping into path selection, taking into consideration the max-min reliability fairness goal, and the networking reliability factors during mutation, URMG can globally optimize the reliability and its fairness of all the multicast VN requests. Through extensive simulations, we demonstrate that URMG achieves close to the optimal reliability fairness with a much lower time complexity than the MILP and yields a significant performance improvement in terms of reliability fairness, bandwidth consumption and transmission delay comparing with other heuristic solutions.**

***Keywords*—** Multicast; Virtual Network Mapping; Reliability; Mixed Integer Linear Programming (MILP); Max-Min fairness

I. INTRODUCTION

Network Function Virtualization (NFV) provides an efficient and flexible way to deploy network services by implementing network functions in software that can run on standardized high volume servers/switches/storage. A set of network services can be provisioned through a *Virtual Network* (VN) (or service function chain [1-2]) consisting of virtual nodes and virtual links. The process of mapping a VN onto a *Substrate Network* (SN) generally includes two interrelated processes: virtual node mapping and virtual link mapping. The former maps each virtual node onto a physical node (servers/switches/storage) that can provide sufficient resources while the latter maps each virtual link to a physical path with sufficient bandwidth resources. With VN mapping in NFV, multiple diverse VNs can coexist on a common SN to share the physical resources, thus reducing the network provider's cost.

Many schemes have been proposed to address the general VN mapping problem for unicast services (e.g., [3-5]). Reliability has also been considered in such VN mapping [6-8] as a single physical node or link failure may affect several VNs. However, few work focused on designing efficient strategies to accommodate multicast service-oriented VNs [9-12]. In fact, many big data applications, distributed file systems (e.g., Map-Reduce), point-to-multipoint real-time and interactive applications (e.g., video-conferencing and IPTV) prefer multicast communications in order to improve the utilization of the physical resources. Unlike the unicast service where data packets are transmitted between a single sender and a single receiver, multicast services require that the same data packet flows to a selected group of destinations (or receivers), which can share the data transmission along the common links.

Recently, constructing reliable multicast tree routing directly in traditional fixed physical networks was investigated [13-17]. The authors in [13] minimized the routing cost in the multicast routing problem with delay and reliability constraint. The study in [14] introduced a reliable multicast routing algorithm based on reliability test in the multimedia communication, which minimizes the network resource utilization for different reliability requirements of multicast requests. The authors in [15] investigated the problem of finding a *k*-hop multicast strategy with maximum reliability in directed tree networks and extended it into general graph with an exponential time complexity. The work in [16] focused on developing high-throughput algorithms for reliable multicast routing in multi-hop wireless mesh networks. The work in [17] tried to find the one-to-many and many-to-many multicast tree with maximum reliability in a fixed topology.

However, none of the above work considers how to efficiently map virtual networks for reliable multicast services, while taking into consideration the max-min fairness of the reliability among distinct VNs. Particularly, these works do not fully consider the multicast routing together with the virtual network mapping and multicast tree design, which limits the mapping selection of reliable nodes. Reliable multicast VN mapping involves two types of sharing: data transmission sharing among multiple receivers within the same VN and SN sharing among multiple VNs. Hence, the reliable unicast VN mapping and reliable multicast routing schemes in traditional fixed networks cannot be directly applied to efficiently mapping VNs for reliable multicast services. In fact, the

978-1-4799-5952-5/15/$31.00 ©2015 IEEE

problem of reliable mapping multicast VNs has a few unique aspects: i) the physical multicasting source and destination nodes are not fixed which makes the problem more challenging; ii) the same multicasting streams over different virtual links may go through the same substrate link (e.g., fiber) to save bandwidth; iii) multicast VN mapping allows different multicast tree design (i.e., determining which destination nodes can be relay nodes to other destination nodes) in order to maximize network resource utilization; and iv) source node and relay nodes mapping need to be more reliable since they will affect many destination nodes.

In this paper, we study the reliable multicast virtual network mapping problem with the objective of maximizing the reliability of the request that has the lowest reliability (which is often called the Max-Min fairness). We only consider the node reliability since the link reliability issues can be straightforwardly converted into node reliability ones. We propose a Mixed Integer Linear Programming (MILP) model to obtain the upper bound of the max-min reliability fairness. To solve the problem efficiently, we design a Uniform Reliability Mutation based Genetic (URMG) algorithm, which has the following features: 1) URMG can jointly optimize node mapping, multicast tree construction and link mapping by encoding them in one gene; 2) URMG converts multicast tree construction and link mapping into path selections to simplify the encoding process; 3) URMG combines the max-min reliability fairness goal with genetic evaluation objective so that it can obtain results close to the optimal solution; 4) URMG adopts a uniform reliability based mutation mechanism, which can highly improve the mutation efficiency. We conduct comprehensive simulations to evaluate the proposed solutions in terms of running time, max-min reliability, bandwidth consumption and transmission delay.

The reminder of this paper is organized as follows. We introduce the problem description in Section II and present the MILP model in Section III. Then we describe the URMG algorithm in Section IV. The performance evaluation is presented in Section V and finally we conclude the paper in Section VI.

II. Virtual Network Mapping for Reliable Multicast Services with Max-min Fairness

In this section, we present the problem of reliable multicast virtual network mapping.

A general substrate/physical network can be modeled as a graph $G_p = (V, E)$, where V is the set of physical nodes and E is the set of physical links. Each physical node v is equipped with $C(v)$ units of computation resources and a reliability constant $r_{pn}(v)$, while each link has a bandwidth of $B(e)$.

For a given multicast VN request $MR_i = (s_i, D_i, b_i), i \in R$, s_i is the virtual source node, D_i ($|D_i|>1$) is the virtual destination node set, and b_i is the requested bandwidth in the multicast group. We assume that a given node $v \in \{s_i, D_i\}$, requires $c(v)$ computing resources and can only be mapped onto a subset of physical nodes denoted by $S(v)$.

For each multicast request, we need to map the virtual nodes, construct a multicast tree (i.e., decide which nodes are "spilt and copy" nodes), and determine the routing, such that the reliability of the mapped multicast request is maximized. For each MR_i, we normalize the expectation $E(D_i)$ of the number of live destination nodes D_i in each multicast request as $E(D_i)/|D_i|$ to evaluate its reliability $R(MR_i)$. The primary reasons that we use $E(D_i)/|D_i|$ to measure reliability of MR_i) include: 1) for any multicast request, the scenario that only source node survives is meaningless; 2) partial reliability (subset of destination nodes work) is acceptable in some multicast applications; and 3) the number of destination nodes are different for different multicast requests.

Given a multicast request MR_i and its tree topology mapping, the time to calculate $E(D_i)$ is exponential according to the definition of $E(D_i)$ as shown in Equation (1):

$$O(|V|*(C_{|D_i|}^1 + C_{|D_i|}^2 \dots + C_{|D_i|}^{|D_i|})) = O(|V|*(2^{|D_i|} - C_{|D_i|}^0)) \quad (1)$$

where $C_{|D_i|}^i = \frac{|D_i|!}{i!*(|D_i|-i)!}$. To reduce the calculation complexity, we apply the linearity of expectation to obtain:

$$E(D_i) = \sum_j E(d_{ij}) = \sum_j \prod_{v \in TN(d_{ij})} r_{pn}(v) \quad (2)$$

where $E(d_{ij})$ is the expectation of node $d_{ij} \in D_i$ that does not fail, and $TN(d_{ij})$ is the mapping tree path from source node s_i to destination node d_{ij}. The time complexity of calculating $\sum_j E(d_{ij})$ in the worst case is $O(|V|)$.

For example, Fig.1 (a) shows a multicast request MR_1 with three destination nodes, Fig.1 (b) shows that s_1, d_{11}, d_{12} and d_{13} can be mapped onto node B, F, C and D, respectively. Node F is the mapping node for d_{11} which is the "split and copy" node. The virtual link s_1-d_{11} is routed on link B-F; s_1-d_{12} is routed on links B-F and F-C and s_1-d_{13} is routed on links B-F and F-D. The reliability $R(MR_1)$ of this mapping is $(E(d_{11})+E(d_{12})+E(d_{13}))/3$, where $E(d_{11})$=0.9*0.9, $E(d_{12})$=0.9*0.9*0.8 and $E(d_{13})$=0.9*0.9*0.7.

Since multiple multicast requests can share the same SN, it may not be fair to simply maximize the sum of the reliability of all the multicast requests when the SN resources are limited. In the following sections, we propose solutions to maximize the reliability of the request that has the smallest reliability to achieve the max-min fairness among all the multicast requests.

III. Mixed Integer Linear programming (MILP)

In this section, we develop a Mixed Integer Linear Programming (MILP) model to mathematically formulate the

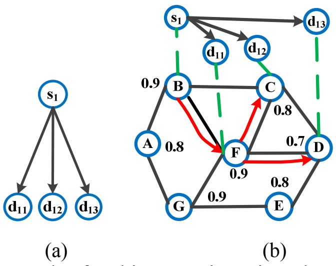


Fig.1 An example of multicast service-oriented VN mapping

problem of reliable multicast VN mapping with max-min fairness. For each physical node, we pre-calculate K shortest paths to all the other nodes and the corresponding path reliabilities. The notations are listed in Table I.

Variables

$$\sigma_{mn,i} = \begin{cases} 1, \text{if } MR_i \text{ uses link } mn \in E_{AG} \\ 0, \text{otherwise} \end{cases}$$

$$\eta_{k,ij} = \begin{cases} 1, \text{if the traffic flow to node } d_{ij} \in D_i \\ \quad \text{goes through path } k \in |V|*(|V|-1)*K \\ 0, \text{otherwise} \end{cases}$$

Objective

The objective is to maximize the smallest reliability of all multicast requests.

$$\text{Max} \min_{i \in R} R(MR_i) \tag{3}$$

where

$$R(MR_i) = \frac{\sum_j E(d_{ij})}{|D_i|} = \frac{\sum_{d_{ij} \in D_i} \sum_{k \in |V|*(|V|-1)*K} (r_p(k) * \eta_{k,ij})}{|D_i|} \tag{4}$$

Constraints

One-on-one node mapping

TABLE I. Notations

Notation	Physical Meaning
$G_p = (V, E)$	a graph representing the physical SN
$v \in V$	a physical node
$C(v)$	the computing capacity of node v
$r_{pn}(v)$	the reliability of physical node v
$mn \in E$	the physical link between m and n
$B(mn)$	the bandwidth capacity on physical link mn
$\|V\|*(\|V\|-1)*K$	the number of paths pre-calculated inside G_p
V^k	the set of nodes along the k^{th} path
E^k	the set of links along the k^{th} path
ps_k	the source node of the k^{th} path
pd_k	the destination node of the k^{th} path
$r_p(k)$	the path reliability of the k^{th} path which is the product of $r_{pn}(v), \forall v \in V^k$
$MR_i = (s_i, D_i, b_i)$	a multicast request i
s_i	the source node of MR_i
D_i ($\|D_i\|>1$)	the destination node set of MR_i
b_i	the request bandwidth of MR_i
$V_i = \{s_i\} \cup D_i$	the node set for request MR_i
d_{ij}	the j^{th} destination node in D_i
$c(v_i)$	the computation resource requirement of node $v_i \in V_i$
$S(v_i)$	the set of candidate physical mapping nodes of virtual node $v_i \in V_i$
E'	set of links from virtual source to its physical candidate mapping nodes and links from destination's candidate physical mapping nodes to corresponding destination
$E_{AG} = E \cup E'$	the augmented link set
$V_{AG} = V \cup (\cup_i V_i)$	the set of nodes from all requests and physical network
$G_{AG} = (V_{AG}, E_{AG})$	the augmented graph

$$\sum_{n \in S(s_i)} \sigma_{s_i n,i} = 1, \forall i \in R \tag{5}$$

$$\sum_{m \in S(d_{ij})} \sigma_{m\, d_{ij},i} = 1, \forall d_{ij} \in D_i, \forall i \in R \tag{6}$$

$$\sum_{m \in V_i} \sigma_{mn,i} \le 1, \forall i \in R, \forall n \in V \tag{7}$$

Equation (5) and (6) ensure that each virtual node is mapped to one and only one physical node, and Equation (7) ensures that multiple virtual nodes from the same request cannot be mapped to the same physical node.

Node capacity constraint

$$\sum_{i \in R} \sigma_{s_i v,i} * c(s_i) + \sum_{i \in R} \sum_{d_{ij} \in D_i} \sigma_{v\, d_{ij},i} * c(d_{ij}) \le C(v), \forall v \in V \tag{8}$$

Equation (8) specifies that the total allocated computing resources on any physical node $v \in V$ cannot exceed its capacity $C(v)$.

Link capacity constraint

$$\sum_{i \in R} \sigma_{mn,i} * b(i) \le B(mn), \forall mn \in E \tag{9}$$

Equation (9) ensures that the total allocated bandwidth on any physical link $mn \in E$ cannot exceed its bandwidth capacity $B(mn)$.

Multicast Tree Construction

$$\sum_{k \in |V|*(|V|-1)*K} \eta_{k,ij} = 1, \forall j \in \{1...|D_i|\}, \forall i \in R \tag{10}$$

$$\sum_{j \in \{1...|D_i|\}} \eta_{k,ij} \le 1, \forall k \in |V|*(|V|-1)*K, \forall i \in R \tag{11}$$

$$\sigma_{mn,i} = \begin{cases} 1, \text{if } \eta_{k,ij} = 1, \forall i \in R, \forall j \in \{1...|D_i|\}, \\ \qquad \forall k \in K, \forall mn \in E^k \\ 0, \text{otherwise} \end{cases} \tag{12}$$

$$\sigma_{s_i ps_k,i} \ge \eta_{k,ij}, \forall j \in \{1...|D_i|\}, \forall i \in R \tag{13}$$

$$\sigma_{pd_k d_{ij},i} \ge \eta_{k,ij}, \forall j \in \{1...|D_i|\}, \forall i \in R \tag{14}$$

Equation (10) shows that one and only one path will be selected for each source and destination pair in any multicast request. Equation (11) represents that the same path cannot be selected by more than one source destination pair in the same multicast request. In Equation (12)-(14), a link is marked as used by multicast request i if the link belongs to any path that is used by multicast request i.

It is intuitive that when K is large enough the above linear programming model can obtain the optimal solution for the reliable multicast VN mapping problem. However, due to the computational complexity, the MILP model is infeasible for scenarios in large-scale networks. Hence, in the next section, we propose heuristic algorithms.

IV. Uniform Reliability Mutation based Genetic (URMG) Algorithm

In this section, we propose a Uniform Reliability Mutation based Genetic (URMG) algorithm that jointly optimizes node mapping, multicast tree construction and link mapping while considering the physical network reliability during the mutation process to achieve max-min reliability fairness. In the following subsections, we present the encoding mechanism, the fitness function, URMG design and the convergence condition for URMG.

A. Genetic Encoding and the Fitness Function

We encode each *gene* as the provisioning for a single multicast request. An *individual* composed by a set of different *genes* represents the provisioning for all multicast requests and a *population* is a set of individuals. Specifically, we encode each $Gene_i$ as $\{\{M(s_i, D_i)\}, \{R_{s_i d_{ij}}, d_{ij} \in D_i\}\}$ for a multicast request MR_i, where

- $\{M(s_i, D_i)\}$ represents the physical node mapping for virtual nodes (s_i, D_i);
- $\{R_{s_i d_{ij}}, d_{ij} \in D_i\}$ represents the physical routing path for virtual link (s_i, d_{ij});
- $R_{s_i d_{ij}} \in P_{s_i d_{ij}}$, $P_{s_i d_{ij}}$ is the set of physical candidate paths for virtual link (s_i, d_{ij}) given node mapping in $\{M(s_i, D_i)\}$.

We calculate K most reliable paths as defined in Section III between each node pair (m, n) in the physical network and denote them as PS_{mn}. When s_i is mapped onto physical node m and $d_{ij} \in D_i$ is mapped onto physical node n, we have $P_{s_i d_{ij}} = PS_{mn}$.

For each MR_i, we randomly select a node mapping $\{M(s_i, D_i)\}$, and then we randomly pick the routing paths $\{R_{s_i d_{ij}}, d_{ij} \in D_i\}$ for its multicast tree construction and link mapping. We apply this process for each multicast request to obtain an individual. Each request then can select a different node mapping and routing paths randomly (i.e., new gene is created) to generate more individuals and those different individuals are grouped together to form a population of size P.

To achieve max-min fairness, we assign the fitness of each individual as F, which is the smallest reliability among all the genes (request mapping). The individual with larger fitness value has a higher chance to survive in the evolution thus URMG can obtain max-min fairness reliability when it converges.

B. Design of URMG

The procedure of URMG is shown in **Algorithm 1**. At the beginning of URMG, the first generation G of size P is initialized randomly, and then G goes into the evolution phase which includes selection, crossover and mutation operations. Specifically, a fixed number of individuals denoted as G_S are randomly selected from the population G, then the tournament selection is applied within the individual set G_S (as shown in Step 2) and the winner of each tournament (i.e., the fittest one in the competing group) is selected to evolve to the crossover phase. In Step 3, we randomly pair all the winners as parents for multipoint gene level crossover to get offspring. For each parent pair, we randomly choose $|R|*p_c$ (where p_c is the cross rate) number of genes to swap. We then select P fittest individuals from the parents' generation population and their offspring pools to keep the population size constant (as shown in Step 4). The chosen P fittest individuals then go into the mutation phase.

Algorithm 1: Uniform Reliability Mutation based Genetic (URMG) Algorithm

1: Initialize the first generation G with population size P and calculate the fitness value for each individual;
2: Select a subset G_S of G to participate tournament selections;
3: Pair all the winners from tournament selection randomly for crossover to generate children;
4: Select P fittest individuals from parents and children, and then the chosen children go to the mutation phase;
5: For each chosen child, call MURW to mutate the chosen genes;
6: Use the mutated child with increased fitness value to replace the individual with lowest fitness value until all the satisfied individuals are replaced while keeping the population size P constant to get new generation G';
7: If it converges or reaches a preset threshold of iteration number, go to **8**; otherwise go to **2** with G';
8: Provide the reliable VN mapping according to the fittest individual in G', terminate the process.

In the mutation phase, a number of genes are randomly selected by mutation ratio $|R|*p_m$ (where p_m is the mutation rate) for each offspring. For each chosen gene, we propose a Mutation based on Uniform Reliability Weight (MURW) strategy as described in **Algorithm 2** to generate new mutated gene (as shown in Step 5 of URMG). The main idea of MURW is selecting $\{M(s_i, D_i)\}$ for the chosen $Gene_i$ according to the Uniform Reliability Weight $URW(v)$ of each physical node $v \in V$, and then randomly selecting $\{R_{s_i d_{ij}}, d_{ij} \in D_i\}$ to finish the mutation. $URW(v)$ is the production of a uniform random value between (0, 1) and the sum of the path reliabilities of all the candidate paths from node v to all the other nodes in physical node set V.

The mutated child with the increased fitness value will replace the individual with the lowest fitness value to keep population size unchanged. URMG then goes to the next evolution stage with this new generation. Once it converges, URMG will map all multicast requests according to the genetic encoding of the best fitness individual in the last generation.

Note that we adopt a self-adaptive strategy [18] to dynamically adjust the crossover rate p_c and mutation rate p_m based on the individuals' fitness as shown in Equation (15) and (16):

$$p_c = \begin{cases} \dfrac{F_{\max} - F'}{F_{\max} - F_{\text{mean}}}, F' \geq F_{mean} \\ 1, \textit{otherwise} \end{cases} \tag{15}$$

$$p_m = \begin{cases} \dfrac{F_{\max} - F_p}{F_{\max} - F_{\text{mean}}}, F_p \geq F_{mean} \\ 0.5, \textit{otherwise} \end{cases} \tag{16}$$

where F_p is the fitness of individual p; $F_{\max}$ is the largest fitness value in the population; F_{mean} is the average fitness value and $F^{'}$ is the larger fitness value of two crossover individual p_1 and p_2.

C. Convergence Condition

To evaluate URMG's convergence performance, we modify the degree of diversity [19] as in Equation (17):

$$D_P = \frac{2}{P(P-1)} \sum_{p1=1}^{P-1} \sum_{p2=p1+1}^{P} \frac{|D_F(p1, p2)|}{F_{\max}} \qquad (17)$$

where $|D_F(p_1, p_2)|$ is the absolute difference of the fitness of individual p_1 and p_2; and F_{max} is the maximum fitness value in the generation. If D_p is lower than a certain threshold for 5 generations or more [20], we say the algorithm has converged. We stop URMG when it converges or the number of iteration reaches a preset threshold.

V. Performance Evaluation

A. Experiment Setting

We evaluate the proposed MILP model and heuristic algorithms on a 14-node, 22-link NSF network. By default, each physical node can provide 10,000 units computing resource, while each physical link has 4,000 units bandwidth resource. Each physical node has a random reliability with uniform distribution between 0.9 and 0.999. For each multicast VN request, the source node and destination node set are randomly generated, while the number of destination is uniformly distributed between [2, 8]. The computing demand of each node is less than 100 units with a uniform distribution. The set of candidate mapping nodes $S(v)$ (where $|S(v)|$ is within [3,14] following a uniform distribution) for each node v in multicast request is randomly selected from the substrate node set. The bandwidth demand of each multicast request is uniformly distributed within 10-100 *Gb/s*. The population size P is 50 and tournament size is set to 0.35*P. The iteration number of URMG is 500 and $D_p = 1.0 \times 10^{-5}$.

We use IBM CPLEX to solve the MILP and Visual Studio to implement the heuristic algorithms. All simulations are run on a computer with 2.5 GHz Intel Core i5-3210 CPU and 12 GB RAM. For the MILP, the simulation will be terminated if the optimal solution is obtained or the running time of 5 hours is reached. Each statistic result is the average result of 20 simulations.

To show the benefits of MURW, we implement another heuristic algorithm called No-MURW, which has the same steps as URMG but uses a random mutation to replace the MURW in Step 5 of URMG. For the comparison, we also implement a random mapping algorithm, namely Rand-Map, which randomly maps nodes and links without considering reliability.

B. Performance Analysis

As shown in Fig.2 (a) (all the figures in Fig.2 share the same solution labels as in Fig.2 (b)), we first evaluate the max-min reliability of different solutions when the number of multicast requests increases. We can see that MILPK (K is the number of pre-calculated shortest paths between each physical node pair and K=1, 2, 3) obtains a higher reliability values as K increases because the larger K is the closer the MILP is to optimal solution. More specifically, MILP2 (MILP3) improves the max-min reliability by at most 0.4% (0.08%) compared to MILP1 (MILP2). We take the results from MILP3 as the approximate upper bound since the improvement ratio compared with MILP2 is small enough. It can also be observed that the max-min reliability achieved by URMG is close to the approximate upper bound obtained by the MILP3 (0.2%-4% decrease ratio), while increasing the reliability of No-MURW (Rand-Map) by 0.4%-3% (0.7%-4%).

In Table II, we list the running time of different solutions when the number of multicast requests varies. We can see that URMG's running time is much smaller than that of the MILPs (which cannot find the optimal solution in a reasonable time when the problem become large). Hence, URMG can also be applied to the case with dynamic requests because of its low time complexity, particularly when the services are time-sensitive.

We also evaluate the bandwidth consumption when the

Algorithm 2: Mutation based on Uniform Reliability Weight (MURW)

1: For each physical node $v \in V$, generate a uniform random value between (0, 1) and update $URW(v)$;

2: For the chosen *Gene$_i$*, select the physical node v' which has the largest $URW(v')$ value from $S(s_i)$ as the mapping node for s_i, and add node v' into $\{M(s_i, D_i)\}$;

3: Select destination mapping nodes:

1) Find a physical node m in $V \setminus \{M(s_i, D_i)\}$ which has the largest $URW(v)$ and mark it as the eligible candidate mapping node for some destination node $d_{ij} \in D_i$;

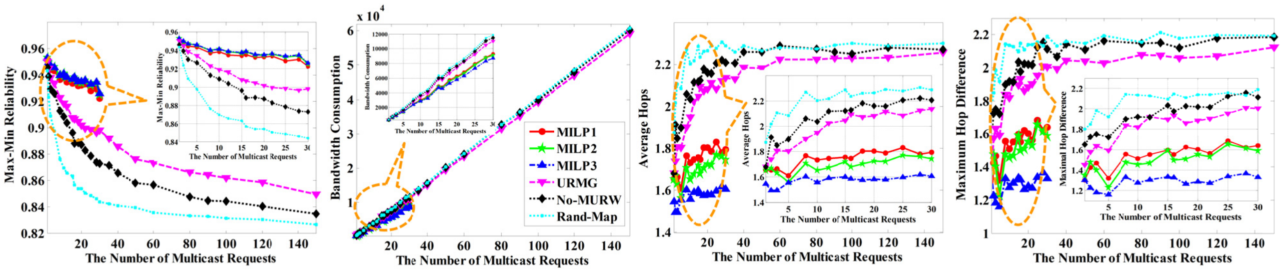
2) If there is more than one destination node using physical node $m \in V$ as the candidate mapping node, select node d_{ij} with the least physical candidate mapping nodes to map first;

3) Map node d_{ij} onto physical node m, let $D_i = D_i \setminus d_{ij}$ and store node m into $\{M(s_i, D_i)\}$;

4: If $D_i \neq \Phi$, go to step **3**, otherwise, randomly select $\{R_{s_i d_{ij}}, d_{ij} \in D_i\}$ and return $\{\{M(s_i, D_i)\}, \{R_{s_i d_{ij}}, d_{ij} \in D_i\}\}$.

TABLE II. Computation time (seconds) of different solutions

	5	10	20	30	80	150
MILP1	1.07	3.36	12.72	29.91	*	*
MILP2	3.25	10.72	45.53	114.56	*	*
MILP3	6.67	23.82	104.41	268.20	*	*
URMG	0.37	0.79	1.58	2.52	5.97	7.18
No-MURW	0.37	0.77	1.53	1.96	4.45	6.89
Rand-Map	0.01	0.02	0.04	0.07	0.18	0.35



(a) Max-Min Reliability (b) Bandwidth Consumption (c) Average Hops (d) Maximum Hop Difference

Fig.2 Results of reliable VN mapping with a max-min fairness over 14-node NSF network

network resources are sufficient for different number of multicast requests. From Fig.2 (b), we can observe that URMG requires 8% more bandwidth compared with MILPs, and consumes 5% and 10% less bandwidth compared with No-MURW and Rand-Map, respectively. The reason is that the more reliable a service is, the fewer hops it traverses, thus leading to a smaller amount of bandwidth consumption. In addition, for delay-sensitive multicast service applications such as video-conferencing and distributed database replication, the transmission delay and the jitter among multiple destinations should be as small as possible. We use the average hops and the maximum hop difference among multiple destinations in the same multicast group to measure the delay bounds and jitter. From Fig.2 (c) and (d), we can observe that average the number of hops achieved by URMG is 5% (8%) smaller than that of No-MURW (Rand-Map) and the maximum hop difference is 6% (13%) smaller than that of No-MURW (Rand-Map), respectively. Hence, URMG is a computational efficient solution that can achieve high max-min reliability fairness while requiring less bandwidth resources and generating smaller transmission delay and jitter than other heuristic solutions.

VI. Conclusion

In this paper, we have investigated the problem of reliable multicast VN mapping. We have proposed a MILP model that can achieve the upper bound of reliability with max-min fairness. In addition, we have proposed an efficient heuristic algorithm called Uniform Reliability Mutation based Genetic (URMG) algorithm, which can jointly optimize the processes of virtual network mapping and multicast tree construction to obtain the max-min fairness among multiple multicast VN requests. Our simulation results have shown that the proposed URMG can achieve the max-min reliability fairness that is close to the one obtained by the MILP and outperforms other heuristics in terms of max-min reliability fairness, bandwidth consumption and transmission delay.

References

[1] Network Function Virtualization white paper, "http://portal.etsi.org/NFV/NFV White Paper.pdf".

[2] Alcatel Lucent, "Network Functions Virtualization-Challenges and Opportunities", 2013.

[3] Z. Cai, F. Liu, N. Xiao, Q. Liu, and Z. Wang, "Virtual Infrastructure embedding for evolving networks," in Proc of GLOBECOM, pp.1–5, 2010.

[4] M. Yu, Y. Yi, J. Rexford and M. Chiang, "Rethinking Virtual Infrastructure Embedding: Substrate Support for Path Splitting and Migration," SIGCOMM Computer Communications Review, vol.38, no.2, pp.17-29, 2008.

[5] N.M. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual Infrastructure embedding with coordinated node and link mapping," in Proc. of IEEE INFOCOM, pp.783-791, 2009.

[6] O. Soualah, I. Fajjari, N. Aitsaadi, A. Mellouk, "A reliable virtual network embedding algorithm based on game theory within cloud's backbone," in Proc. of IEEE ICC, pp.2975-2981, June 2014.

[7] Q. Zhang, M.F. Zhani, M. Jabri, R. Boutaba, "Venice: Reliable virtual data center embedding in clouds," in Proc. of IEEE INFOCOM, pp.289-297, 2014.

[8] W. Yeow, C. Westphal, U.C. Kozat, "Designing and Embedding Reliable Virtual Infrastructures," ACM SIGCOMM Computer Communication Review, vol. 41, no.2, pp.57-64, Apr. 2011.

[9] M. Zhang, C. Wu, M. Jiang, and Q. Yang, "Mapping multicast service-oriented virtual networks with delay and delay variation constraints," in Proc. of GLOBECOM, pp.1–5, 2010.

[10] X. Gao, Z. Ye, W. Zhong, C. Qiao, X. Cao, H. Zhao, H. Yu and V. Anand, "Multicast Service-oriented Virtual Network Mapping over Elastic Optical Networks," in Proc. of IEEE ICC, pp.6796-6801,2015.

[11] Y. Miao, Q. Yang, C. Wu, M. Jiang, and J. Chen, "Multicast Virtual Infrastructure mapping for supporting multiple description coding-based video applications," Computer Networks, vol.57, no.4, pp.990-1002, Mar. 2013.

[12] P. Lv, Z. Cai, J. Xu, and M. Xu, "Multicast Service-Oriented Virtual Infrastructure Embedding in Wireless Mesh Networks," IEEE Communications Letters , vol.16, no.3, pp.375-377, Mar. 2012.

[13] C. Yang, X. Zhao, "A multicast routing algorithm based on reliable transmission," in Proc. of IEEE IMSNA , pp.679-682, Dec. 2013.

[14] X. Qi, L. Liu, S. Liu, J. Qiao, "A reliable multicast routing algorithm based on reliability test in multimedia communication," in Proc. of IEEE. ICCCAS, vol.1, pp.596-600, Jun. 2004.

[15] M.I. Andreica, N. Tapus, "Maximum reliability k-hop multicast strategy in tree networks," in Proc. of IEEE ISCE, pp.1-4, Apr. 2008.

[16] X. Zhao, J. Guo, C. T. Chou, A. Misra, S.K. Jha, "High-Throughput Reliable Multicast in Multi-Hop Wireless Mesh Networks," IEEE Transactions on Mobile Computing, vol.14, no.4, pp.728-741, Apr. 2015.

[17] W. Ding, X. Wei, X. Nie, M. Zhou, "Algorithms for Finding Maximum Reliability Many-to-Many Multicast Routing under a Fixed Topology," in Proc. of IEEE ICNDC, pp.124-127, Oct. 2012.

[18] M. Srinivas, and L. M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," IEEE Transactions On Systems, Man, and Cybernetics Society, vol.24, pp.656- 667, Apr. 1994.

[19] L. Gong, X. Zhou, W. Lu, and Z. Zhu, "A two-population based evolutionary approach for optimizing routing, modulation and spectrum assignments (RMSA) in O-OFDM networks," IEEE Communications Letters, vol.16, no.9, pp.1520–1523, Sept. 2012.

[20] J. Koza, "Genetic Programming: On the Programming of Computers by Means of Natural Selection," Cambridge, MA: MIT Press, 1992