Computer Communications xxx (2016) xxx–xxx

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

Virtual Network Embedding for telco-grade network protection and service availability

Q1 Ashiq Khan[a], Xueli An[b,1,*], Shigeru Iwashina[a]

[a] Research Laboratories, NTT DOCOMO, Inc., Tokyo, Japan
[b] DOCOMO Communications Labs Europe GmbH, Munich, Germany

A R T I C L E I N F O

Article history:
Received 3 January 2015
Revised 18 February 2016
Accepted 14 March 2016
Available online xxx

Keywords:
Virtual Network Embedding
Network virtualization
Network protection
Telecom networks

A B S T R A C T

The decoupling of software from hardware by means of virtualization presents us with a unique opportunity to perform on-the-fly network deployment and reconfiguration. Virtual Machines could be instantiated and virtual links can connect these machines to form end-to-end virtualized networks on top of a physical network infrastructure. Virtual Network Embedding (VNE) algorithms could be used to map such virtual network on a physical network infrastructure. However, present VNE algorithms do not consider overall network protection. This is critical for a Telecom operator that needs to realize 99.999% network availability. Moreover, VNE is an NP-hard problem. In this paper, we propose a heuristics VNE aimed at protecting Telecom operator sites. Our objective is to develop advance counter-measures in the form of telco-grade redundancy to avoid large scale network failures such as the one observed during the tsunami in Japan in 2011. In designing our VNE algorithm, we distinguish server nodes from switching nodes as a server function cannot be embedded on a switching node. We also choose not to employ path splitting which is difficult to implement in real network operations. Along with detail modeling of our proposal, we also evaluate our scheme which shows its efficiency in realizing operators' valuable infrastructure protection while maintaining lower resource consumption.

1. Introduction

Network virtualization has gained large momentum in the research community over the recent years and is now moving towards commercialization. Extensive usage of network virtualization can be observed in the datacenters or clouds in the IT sector where service availability and the consequent protection requirements are not so stringent. Telecom operators are also becoming increasingly interested in network virtualization. Some potential use cases in the Telecom sector [1] are the ease of deployment of nodes like MME, S/P-GW [2], on-demand scaling of such nodes based on instant load rather than peak load based over-provisioning [3], dynamic topology reconfiguration for disaster avoidance and recovery [4], etc.

Network virtualization consists of computing node virtualization and communication link/path virtualization. This creates virtualized end-to-end networks on a Physical Network Infrastructure (PNI). Virtual Machine (VM) virtualizes computing nodes or network functions. For transport network virtualization, Software-Defined Networking (SDN) has become a prominent candidate for virtualizing communication links/paths. The key features of virtualization are isolation among virtualized entities i.e. VMs and Virtualized Links (VLs), and decoupling of software from hardware. Isolation among virtualized entities enables the coexistence of multiple Virtualized Networks (VNs) in the same PNI e.g. different generations of cellular networks [5]. The independence of software e.g. VM from the underlying hardware enables on-the-fly network creation, which takes years at present in physical network deployments. The PNI is a static entity on top of which, VNs with different topologies, computing and link resources can be created, operated and removed on demand. This enables faster network deployment, reduces occupying resources when not necessary, and thus improves resource usage efficiency of a PNI to maximize revenue. Such decoupling between software and hardware also enable VMs and VLs migration [6]. Such characteristics can be used to migrate critical network facilities, when virtualized, to safer location during natural disasters.

Such automated and on-demand VN generation can be performed by Virtual Network Embedding (VNE) techniques. A VNE request consists of a VN topology, necessary computing and link resources e.g. number of VMs per computing node in the requested topology, link Bandwidth (BW), delay and other requirements. A

Q2 * Corresponding author. Tel.: +49 1781320572.
E-mail addresses: khan@nttdocomo.com (A. Khan), an_de_luca@docomolab-euro.com, anxueli@gmail.com (X. An), iwashina@nttdocomo.com (S. Iwashina).
[1] Present address: European Research Center, Huawei Technologies, Munich, Germany.

http://dx.doi.org/10.1016/j.comcom.2016.03.017
0140-3664/© 2016 Published by Elsevier B.V.

VNE algorithm then finds out the best possible mapping i.e. embedding solution of such requests on a PNI. VNE is an NP-hard problem [7]. Many heuristics have been proposed with a view to finding a workable polynomial time solution. A major drawback is that network protection aspects are largely absent in the existing solutions. Many schemes are available for path protection in the form of multi-path redundancy. However, site e.g. datacenter protection schemes are not available. Network paths are stateless and if paths are lost, service does become unavailable. However, it does not destroy user or operational data. If a site is destroyed, uncountable amount of user and network operation data are lost. In this paper, a site refers to a datacenter or cloud, consisting of a collection of physical servers. Sites are assumed to be geographically distributed i.e. a Telecom operator or a service provider has multiple such sites in a country, connected by a core transport network. In this paper, we consider such sites hosting telecom node functions [2] in the form of VMs, which require high service availability. For modeling purposes, a site is sometimes referred to as a site node which is an abstraction of the whole site to a single network node providing computational and storage resources.

In this paper, we propose a polynomial time VNE algorithm for telecom operators site protection. Our objective is to design a solution which can provide protection for all sites in an operators network [8]. There is no less critical site for an operator, who is bound by regulatory constraint on service availability of 99.999% (five 9s). This results in a downtime of around 5 min a year [9]. The conventional 1 + 1 protection scheme employed by the Telecom operators is based on such a constraint. Unlike the available site protection schemes which address a single-site failure at a time, our objective is to provide solution for simultaneous multiple-site failures. Therefore, our aim is to develop a VNE algorithm which provides a 1 + 1 redundancy to all Telecom sites so that any number of site failures can be recovered without service interruption. This comes from our experience with the earthquake/tsunami in March 2011 in Japan, where Telecom operators experienced large-scale network failures over a prolonged period of time. We explicitly do not address path protection in this paper because of the availability of in-depth research in this area which has resulted in a number of path protection schemes [18–22]. We do address link embedding in this paper to the extent of the correlation between link embedding and node embedding to improve the overall VNE performance. Correlation between node embedding and link embedding is largely absent in existing VNE methods, where a two-step approach is adopted – first, selection of the nodes, then discovery of paths among the selected nodes to embed the requested links. In such approaches, a node having sufficient computational resources but not sufficient link resources can be selected as potential candidate to embed a virtual node. Such nodes are discarded in the second step where the link embedding algorithm discards such nodes with insufficient bandwidth. To avoid this inefficiency, we only select those nodes that have sufficient ingress/egress link bandwidth to host the consequent VLs. This correlation between node and link embedding is also an originality of our proposal.

The scope of this paper is to propose a polynomial time heuristic VNE for Telecom site protection, where a backup site is explicitly found for each site in a VNE request. Our objective is to put forward an efficient VNE which achieves the above but with higher resource usage efficiency compared to existing solutions. We restrict ourselves to the theoretical evaluation of the VNE algorithm itself in terms of its success rate, and how much network resources the VNE solution consumes. In this paper, we do not address the site recovery procedure i.e. exactly when and how a primary site is switched over to a backup site. Such decision depends on telecom operators operational principles, as well as the particular node backup mechanism involved (e.g. hot standby). In this paper, our objective is to ensure that a backup site is found which realizes a telco-grade 1 + 1 protection scheme for all Telecom sites. The consequent evaluation of switching mechanism to backup sites during failures is an important item to further evaluate the efficiency of our proposed solution in practical network operations, and we intend to address this in our future work.

The rest of the paper is organized as follows: Section 2 discusses related work. Section 3 presents our network model and formulates the VNE problem mathematically. In Section 4 , we present our site-protection VNE algorithm, and provide evaluation results in Section 5. In Section 6 we discuss how to use such VNE algorithms in a Network Functions Virtualization (NFV) context. Section 7 concludes the paper with a summary and areas for future exploration.

2. Related work

ViNEYard [7] proposes VNE algorithms where node mapping is coordinated with link mapping. Here, VNE is formulated as Mixed Integer Linear Program (MIP). As MIP is computationally intractable, they relax the integer constraints to obtain a Linear Program (LP) which can be solved in polynomial time. However, they use location as a requesting parameter in a VNE request. This limits the embedding location of a VNE request. It potentially overloads a certain locality of a physical network infrastructure whereas other parts of the infrastructure may remain underutilized. They also use Multi-Commodity Flow (MCF) which embeds a virtual link over multiple physical paths by path splitting, thus increasing the success rate of a VNE. However, path splitting is not supported in real commercial public network. Routers do not keep multiple paths to a certain destination and even if they do, they do not use them simultaneously for the same session. Besides, sending packets of the same session over multiple paths leads to packet reordering problem in the end host, as different packets arrives in a different order due to different lengths of the paths. Packets out of sequence are usually discarded.

Authors in [10] propose two Fault-Tolerant VN Embedding (FTE) approaches. These are FTE-PP for protection in the physical layer and, FTELP for protection in the logical or VN layer. The FTE-PP provides two protection paths for each link, and focuses on minimizing the backup resources necessary for such redundancy. FTE-LP augments a VN topology with redundant resources so that a VN survives physical link/node failure by using the redundant resources. In our proposal, we provide 1 + 1 backup for a site. However, it does not necessarily mean that the backup sites cannot be used for other purposes in the absence of a failure in the primary site e.g., allocating to low priority services. In [10], the authors assume that the substrate network (PIP) is not operational all the time. The authors re-emphasize our statement that although there are numerous physical link protection schemes available, little work has been done on node failure protection.

However, in this work, the node mapping and the link mapping remains uncorrelated which, as demonstrated by the authors in [7], leads to suboptimal VNEs as well as extra complexity for the embedding process itself. The most significant difference between the research presented in [10] and our solution is that the authors in [10], like most other related work, assume that there are no simultaneous multiple node or link failures. Our solution is entirely based on the assumption that there are indeed cases where simultaneously multiple nodes/sites fail, e.g. during large scale natural disasters. The proposal in [10] solely focuses on protection against single-node or link failure.

Authors in [11] propose a heuristic Virtual Network Embedding techniques that increase the survivability of the embedded virtual network by means of node migration and link remapping. They use the Artificial bee colony algorithm to achieve optimal Virtual Network Embedding. When a node fails (a node is a site in the

Please cite this article as: A. Khan et al., Virtual Network Embedding for telco-grade network protection and service availability, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2016.03.017

context of our proposal), it is migrated to a normal node. After that, affected links are all remapped by using the shortest path algorithm. However, as mentioned before, a mobile Telecom network which is the primary focus of this work, has a service availability requirement of five 9s. Searching for a new node after the failure and then migrating a heavy telco node virtual machine requires significant time [12]. To meet telecom service availability requirements that are also regulatory, we not only prepare a backup node beforehand, but also prepare it at a reasonable network distance so that switching over to the backup node once the primary node fails could be done within Telecom service availability requirements. For voice service, fail-over needs to be performed in less than a second.

Authors in [13] propose a reliability-aware heuristic VN embedding algorithm, where they try to minimize the over-provisioning of network resources necessary for such reliability, in other word, redundancy. They assume a heterogeneous failure rate for different elements in the Physical Network Infrastructure (PNI), which, we believe, is a realistic assumption. They then calculate the overall reliability based on the heterogeneity of the PNI elements. In their model, a VNE request comes with its reliability requirement which needs to be met during the VNE process. However, as the authors themselves point out, measuring the reliability of a VN is a daunting task. To reduce the problem space, they define protection-domains where, failure of one element in the domains leads to the failure of the whole domain. Such clustering reduces the number of elements needed to be considered in order to measure the overall reliability of a VN. However, while performing the VNE taking the overall required reliability of a requested VN, this proposal uses Multi-Commodity Flow (MCF), which we consider unrealistic in real network operation (see Section 1).

Authors in [14] consider regional disasters to be of stochastic nature and incorporate this when they perform VNE for improved reliability. They estimate risk values for different regions and consequently perform the VNE with such risk-awareness. We consider this work very suitable to disaster-prone regions like Japan where strong earthquakes are frequent. As explained in detail in Section 4.1, our backup site mapping considers a network distance from the primary site to its backup site so that both do not fail simultaneously. The proposal in [14] could be an efficient way to define such network distance so that the primary and backup sites are not mapped in the same failure region e.g. continental plates. However, we generalize this aspect rather than explicitly focus on disaster-prone regions so that our scheme could be used in any arbitrary site failure scenario.

Authors in [15] address the topic of this paper i.e. survivable VN design by means of protection against site node (‘facility node’ in their term) failure. They propose two heuristic schemes which extend the target VN for redundancy during embedding, and then improve resource usage efficiency by enabling resource sharing between primary and backup sites. The Extended Virtual Network (EVN) approach before embedding taken in [15] is similar to our approach where we extend the requested VN first for our objective of simultaneous multiple site failure protection. However, this work purely focuses on recovery from a single site node failure and proposes a resource efficient way to design an $N+1$ VN topology (N is the number of site nodes in the original VNE request). The proposed scheme in [15] will fail to provide protection during simultaneous multiple site failures, which is the main focus of our work.

Authors in [16] assume a flexible optical grid transport plane which is controlled by a Software Defined Network (SDN) controller. The controller takes a VNE decision, and performs the embedding to realize link protection and node (‘site’ in our context) protection. However, they consider a shared protection scheme which reduces resource consumption necessary for redundancy at the expense of the number of failures the system could recover from. From the node perspective, a node, which has sufficient resources to backup all other nodes, is selected as the shared backup. Creating such resource-heavy single site in a Telecom network is a formidable task. Further, the authors in [16] consider a particular transport network topology i.e. optical grid, whereas we do not restrict ourselves to any particular topology. Our network model is sufficiently robust to accommodate any present, as well as future network topology that could appear due to the flexibility provided by network virtualization.

Authors in [17] present a heuristic VNE algorithm which takes into account the substrate node reliability awareness during the embedding process. In their proposal, they first rank the substrate network nodes based on their reliability and resource load. They then chose the high ranking nodes to optimize reliability against resource consumption. However, the focus remains on minimization of node resource consumption rather than redundancy for failure recovery. The proposed scheme relies on the high reliability value of a chosen substrate node. In real-life network operation, all nodes fail—highly reliable or not. Besides, highly reliable nodes cannot avoid failure in natural disasters which is our focus area for improved protection and reliability.

Including the works presented above, most network protection and service availability schemes aim at optimizing network redundancy against resource consumption. This leads to an $N+K$ protection scheme where $N>K$. Here, N is the number of site nodes in the requested VN, and K is number of backup sites for N site nodes. Our proposal is an explicitly $N+N$ protection scheme which is a telco service requirement. And, an $N+N$ protection scheme can always be reduced to an $N+K$ protection scheme without any added complexity, but not vice versa. Extending an $N+K$ protection scheme to an $N+N$ protection scheme is not straightforward. If executed along the lines of the conventional approach described in Section 4, it becomes very inefficient from the point of view of resource consumption, as would be shown in Section 5 of this paper. The motivation to develop a robust $N+N$ protection scheme is further underlined by this issue.

In the existing literature, link protection schemes have been exhaustively investigated. Our focus in this paper is not on link protection; rather, the protection of high-availability Telco-sites, which hold data of millions of customers per-site. Existing link-protection schemes [18– 20] can be readily used with our scheme to realize link protection. However, in relation to multiple simultaneous link failures, we present two recent works which can become useful under such failure scenarios.

Authors in [21] confirm our finding that a live migration-based protection scheme would not suffice to limit the service downtime to below a reasonable value [9]. They propose Opportunistic Resilience Embedding (ORE) which proactively maps a virtual link to multiple physical paths for protection reasons. They also have a reactive step which tries to recover the lost capacity after a failure. Although we do not explicitly address link protection, but rather try to minimize link resource consumption during a VNE process, mapping a virtual link to multiple substrate paths will consume substantially more link resources than in our scheme. However, as the path redundancy level is quite high, ORE [18] can be a suitable technique to recover from multiple simultaneous link failures. In our view, this proposal can fit very well with ours where we ensure simultaneous multiple site failures, whereas this scheme is used to recover from simultaneous multiple link failures, both of which are observed during large scale natural disasters.

The work presented in [22] provides a modeling scheme for VNE, where the link availability constraints are added to the links in a VNE request. As VNE is an NP-hard problem, authors in [22] present a heuristic which meets the link availability requirement in a VNE request. The heuristic selects multiple physical

Please cite this article as: A. Khan et al., Virtual Network Embedding for telco-grade network protection and service availability, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2016.03.017

Table 1
Summary of related work.

Related work	Features					
	Link-node correlated embedding	No path splitting	Server-switch distinction	Limited site protection	All site protection	Proactive approach
ViNEYard [7]	Yes	No	No	No	No	Yes
FTE [10]	No	Yes	Partially	Yes	No	Yes
Migration-based [11]	No	Yes	No	Yes	No	No
Reliability aware [13]	No	No	No	Yes	No	Yes
Region failure [14]	Yes	Yes	No	Yes	No	Yes
Survivable VN [15]	Yes	No	No	Yes	No	Yes
Optical grid [16]	No	Yes	No	Yes	No	Yes
Selected protection [17]	No	Yes	No	Yes	No	Yes
Our proposal	Yes	Yes	Yes	Yes	Yes	Yes

Q3

paths, where the total availability over all the selected physical paths can meet the requested availability of one VL. Thus, one VL, depending on the physical links it is being mapped on, can have multiple backup paths. This is a realistic approach in ensuring link failure recovery as the conventional $1+1$ link protection may not suffice if the underlying physical links availability values are too low.

In Table 1, we present a summary of the related body of work focusing on the research that explicitly mentions site protection. In our view, the link protection schemes can be used with most site protection schemes. Table 1 clearly positions our proposal and shows its scope compared to other relevant work. The higher numbers of 'Yes' shows the proximity to our work in this paper. Proactive approach refers to the case where a protection scheme is determined and deployed beforehand. Table 1 should be viewed in conjunction with the detailed comparison provided above.

3. Modeling and problem formulation

In future commercial networks, a VNE request will come from a Virtual Network Operator (VNO) which wishes to provide service using the PNI. Fig. 1a shows an operational structure [5] of the process. Operators Network Operation System will receive such VNE requests, and embed them in the PNI. The PNI consists of two components: the sites and the core transport network. A site here is a datacenter/cloud (shown as a cloud in Fig. 1a), consisting of numerous physical server machines. These sites are geographically distributed over a large area e.g. a country, and the core transport network provides connectivity among these sites. We envision that the sites with computation and storage capabilities can be used to host services like 3GPP core network nodes/functions, e.g. MME, S/P-GW, etc., which can be deployed in the form of VMs on physical machines. Such mobile core nodes are conventionally deployed in a geographically distributed way to perform mobility management and user plane aggregation closer to mobile users. Upon a particular VN embedding request, these mobile core network nodes are embedded in the PNI. The graphical interpretation of the scenario explained above is shown in Fig. 1b. The Telecom sites in Fig. 1a are modeled as single-site nodes in Fig. 1 b. The numbers shown in Fig. 1b beside these represent their capacity, such as the number of VMs this site can accommodate at the moment, or the number of physical machines available etc. The core transport network consists of switching nodes and links connecting the site nodes. Links can be specified by their BW limitation and delay (d) values.

3.1. Network model

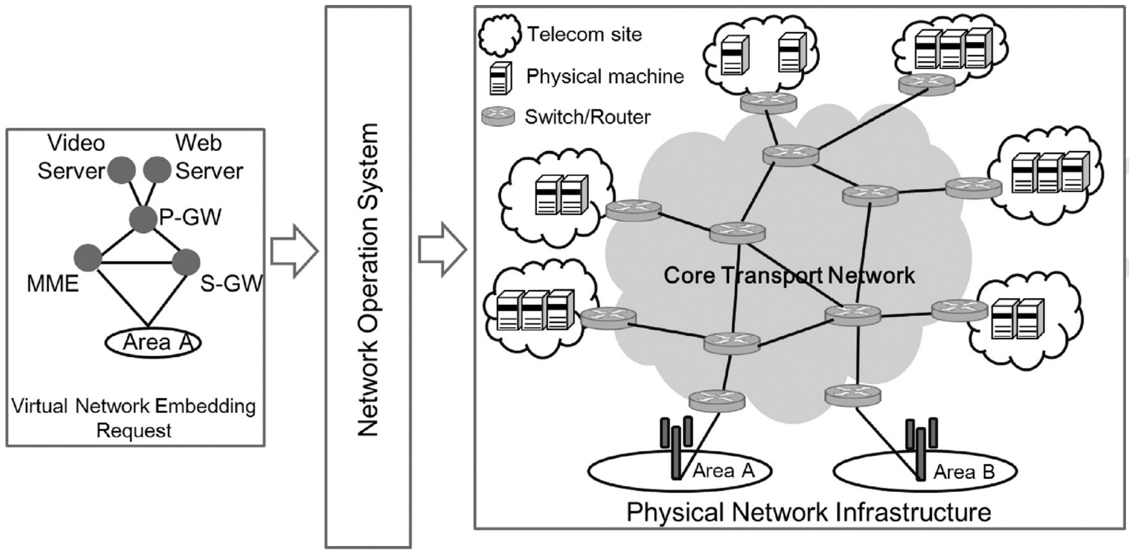
The entire PNI can be modeled as a graph $G_p = (V_p, E_p)$, where V_p and E_p represent the set of vertices/nodes and the set of links within the PNI, respectively. We use $|V_p|$ to denote the number of nodes in the set of V_p. We use n_p^i to refer to a node with ID i which belongs to set V_p, $\forall n_p^i \in V_p$. A link between node i and j is given by $e_p(i, j)$, and $\forall e_p \in E_p$. The adjacency matrix of G_p is given by A_p, which is a $|V_p| \times |V_p|$ matrix. If there is an incident link between nodei and j, $A_p(i, j) = 1$; otherwise, $A_p(i, j) = 0$. A bidirectional graph is assumed in this work, hence $A_p(i, j) = A_p(j, i)$. A path from source nodei and destination node j is denoted as $p(i \to j)$, which is the collection of the links along the path. The path length is given by $|p(i \to j)|$. $\bar{i}$ and $\bar{j}$ represent a set of sources and destinations, hence $p(\bar{i} \to \bar{j})$ is the set of paths containing all the combinations of the sources and destinations. $P_p$represents the set of all the feasible paths within G_p. Therefore, if the source and destination are physical nodes ($\forall n_p^i, n_p^j \in V_p$), then the notation $p(i \to j)$ represents a physical path, which can be represented as$p(i \to j) \in P_p$. The set of switching nodes within the core transport network is given by V_s and the set of site nodes are represented as V_r, hence we have $V_p = V_s \cup^{V_r}$ on condition that $V_s \cap^{V_r} = \phi$. Node capacity is specified by $a \times |V_p|$ arrayC_p, in which, the capacity of node i is $C_p(i)$. The link BW is specified by a $|V_p| \times |V_p|$ matrix B_p. For a link $e_p(i, j) \in E_p$, its available BW is given by $B_p(i, j)$. The available BW on a path $p(i \to j)$ is denoted by $f_p^{i \to j}$, the value of which is limited by the intermediate link that has the minimum BW as $f_p^{i \to j} = \min(B_p(i, v_1), \ldots B_p(v_{|p(i \to j)|-1}, j))$, where v_i for $\forall i \in [1, |p(i \to j)| - 1]$, is an intermediate node on the path.

3.2. Virtual Network Embedding request

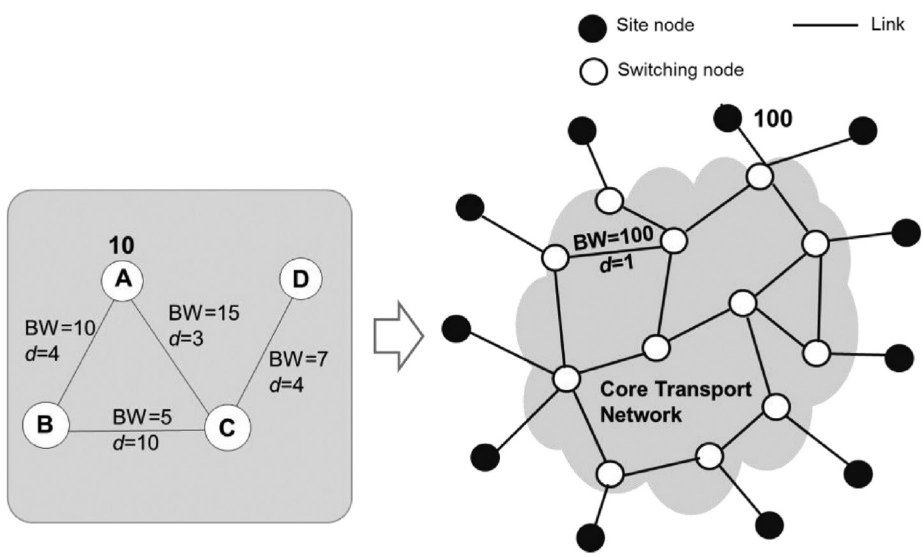
The VNE request can come in different granularities. If a VNO chooses to operate its virtual network at router level and above, it can send a request in that detail. As this work is about site protection, we assume that a VNO sends its embedding request at site granularity. A VNE request consists of VN nodes and links, which can also be represented by an undirected graph $G_v = (V_v, E_v)$ with adjacency matrix A_v. Beside the VN topology, the VNE request also provides the requirement constraints in terms of node capacityC_v, link BW B_v and delay limits D_v. Delay can be defined by round trip time (RTT) or hop count. We assume that the dominant factor of delay is the processing time when packets pass through a switching node, hence we use hop count as a parameter to model delay. Fig. 1b provides an example of a VNE request, which consists of four nodes with capacity, link BW, and delay requirements. Once a VNE request arrives, it will be mapped to the PNI, which means that a number of site nodes will be selected within the PNI, and the bandwidth among the site nodes will be reserved for hosting service and communication purposes.

To host Telecom services like 3GPP core network nodes in VNs, carrier-grade availability is compulsory. The embedded VNs should be survivable and recoverable from any number of simultaneous site failures. Providing $1+1$ backup for all the sites is one solution to achieve this required high availability. Therefore, along with

Please cite this article as: A. Khan et al., Virtual Network Embedding for telco-grade network protection and service availability, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2016.03.017



a. Network operation for VNE



b. Graphical representation of VNE

Fig. 1. VNE request embedding.

one primary site, one backup site is also needed for mapping a VN node. The backup site can be hot-standby and the running service states are synchronized between the primary and backup sites at all times, hence the backup site can take over the service immediately without interruption when the primary site goes down. However, the backup mechanism is out of the scope of this work. In this paper, we focus on how to select and inter-connect the backup and primary sites.

3.3. Problem formulation

The VNE problem can be considered as a process with two stages: VN node mapping and VN link mapping. In the first stage, VN nodes are mapped to site nodes in the PNI using function $M_n : V_v \mapsto V_r$. In order to achieve 1 + 1 site protection for VN node i, a primary site node $n_r^{i_p}$ and a backup site node $n_r^{i_b}$ are selected together for VN node mapping:

$$\mathfrak{M}_n(n_v^i) = \left\{n_r^{i_p}, \quad n_r^{i_b}\right\}, \quad \forall n_v^i \in V_v, \quad \forall n_r^{i_p}, \quad n_r^{i_b} \in V_r, \quad n_r^{i_p} \neq n_r^{i_b} \tag{1}$$

In the second stage, the feasible paths between all the mapped site nodes are established by using function $\mathrm{M}_1 : \mathrm{E}_v \to \mathrm{P}_\mathrm{P}$, where,

$$\mathfrak{M}_l(e_v(i, j)) = p(\mathfrak{M}_n(n_v^i) \mapsto \mathfrak{M}_n(n_v^j)), \forall n_v^i, \quad n_v^j \in V_v \tag{2}$$

To guarantee the seamless service migration in the site failure scenario, for instance for a VNE request with two nodes and one link as shown in Fig. 2a, we have to explicitly search for two primary and two backup nodes, and six links between all the primary and backup nodes to enable 1 + 1 site protection as illustrated in Fig. 2b. This would be the conventional method to handle 1 + 1 site protection. Hence, the total required BW to embed one VN link is the summation of the reserved BW on all six links. The required

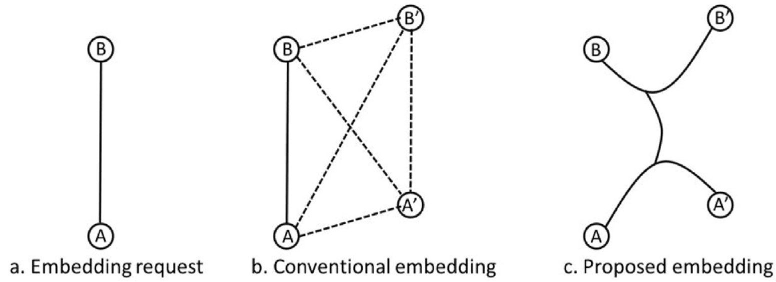


Fig. 2. 1 + 1 site protection illustration.

backup BW for the path between the primary and backup nodes might be fewer than the requested BW on the primary path, which depends on the selected backup mechanism as we mentioned before, or the explicit request from the VNO. For analysis simplicity, we assume that the required backup BW is the same as the requested BW for the primary path. $\mathfrak{M}_l(e_v(i,j))$ defined in (2) is a path set with two kinds of paths: data communication path set $\mathfrak{M}_l^c(e_v(i,j))$ and backup path set $\mathfrak{M}_l^b(e_v(i,j))$ where $\mathfrak{M}_l^c(e_v(i,j)) = \{p(i_p \to j_p), p(i_b \to j_b), p(i_p \to j_b), p(i_b \to j_p)\}$ and $\mathfrak{M}_l^c(e_v(i,j)) = \{p(i_p \to i_b), p(j_p \to j_b)\}$. The VNE request embedding issue can be formulated as an optimization problem as following:

Objective:

$$\min \sum_{e_v(i,j)\in E_v} B_v(i,j)\Bigg(\sum_{p(x\to y)\in\mathfrak{M}_l^c(e_v(i,j))} |p(x\to y)| + \sum_{p(x\to y)\in\mathfrak{M}_l^b(e_v(i,j))} |p(x\to y)|\Bigg) \quad (3)$$

Resource constraints:

$$C_v(i) \le \min\left(C_r(i_p), C_r(i_b)\right), \forall n_r^{i_p}, n_r^{i_b} \in V_r \quad (4)$$

$$f_p^{i_p\to j_b} \ge \sum_{j=1, j\neq i}^{|V_v|} B_v(i,j), \forall n_v^i, n_v^j \in V_v \quad (5)$$

$$f_p^{x\to y} \ge B_v(i,j), p(x\to y) \in \mathfrak{M}_l^c(e_v(i,j)) \quad (6)$$

$$|p(x\to y)| \le D_v(i,j), p(x\to y) \in \mathfrak{M}_l^c(e_v(i,j)) \quad (7)$$

$$\sum_{e_v(i,j)\in E_v}\Bigg(\sum_{p(x\to y)\in\mathfrak{M}_l^c(e_v(i,j))} B_v(i,j) 1_{p(x\to y)}(e_p(u,v)) + \sum_{p(x\to y)\in\mathfrak{M}_l^b(e_v(i,j))} \sum_{j=1, j\neq i}^{V_v} B_v(i,j) 1_{p(x\to y)}(e_p(u,v))\Bigg) \le B_p(u,v) \quad (8)$$

Node and link constraints:

$$x_{i,j} \in \{0,1\}, \forall n_v^i \in V_v, \forall n_v^j \in V_r \quad (9)$$

$$x_{i,i_p} = 1, x_{i,i_b} = 1, \quad \sum_{j=1, \quad j\neq i}^{|V_v|} x_{j,i_p} = 0 \quad \text{and} \sum_{j=1, j\neq i}^{|V_v|} x_{j,i_b} = 0 \quad (10)$$

$$x_{uv} \in \{0,1\}, \forall n_p^u, n_p^v \in V_p, \forall e_p(u,v) \in E_p \quad (11)$$

$$\sum_{i:e_p(i,k)\in E_p} x_{ik} - \sum_{j:e_p(k,j)\in E_p} x_{kj} = \begin{cases} -1, & k = s, n_r^s \in V_r \\ 1, & k = d, \quad n_r^d \in V_r, \quad p(s\to d) \in P_p \\ 0, & n_s^k \in V_s \end{cases} \quad (12)$$

Eq. (4) represents the capacity constraint from a VNE request, which means that the selected primary and backup site nodes should have sufficient capacity to host it. Eq. (5) is the BW constraint for the path between the primary and backup site nodes, which are mapped from VNE node *i*. It implies that the reserved BW for this path is the summation of all the incoming and outgoing traffic from VN node *i*. Eqs. (6) and (7) are the BW and delay constraints for the communication paths. Eq. (8) implies that the total BW of all the flows passing through the physical link $u \to v$ is limited by its available BW $B_p(u,v)$, in which $1_A(a)$ is an indicator function and $1_A(a) = 1$ if $a \in A$, otherwise, $1_A(a) = 0$.

In Eq. (9), $x_{i,j}$ is a binary variable, which is 1 if site node *j* is selected as a primary or a backup node for VN node *i*. Otherwise, it is zero. Eq. (10) ensures that one site node can only accommodate one VN node (primary or backup) for one VNE request. $\tilde{x}_{uv}$ introduced in Eq. (11) is also a binary variable which is equal to the indicator function $1_{p(x\to y)}(e_p(u,v))$ defined in Eq. (8). Eq. (12) limits that, for all the intermediate switching nodes on the path $p(s \to d)$, the number of incoming links is equal to the number of outgoing links.

4. Our proposal: VNE for telco site protection

As mentioned in the previous section, we aim to realize the telco-grade 1 + 1 protection scheme for all Telecom sites. 1 + 1 site protection is expensive in terms of bandwidth because the required bandwidth is not only reserved from the primary to primary sites, but also the path between primary to backup, and backup to backup sites. This is implemented in order to handle any number of primary site failures. Therefore, our main objective is to reduce the bandwidth consumption to embed the VNE requests.

At first, we select potential candidate site nodes in the PNI with enough resources to accommodate the requested nodes. In the second step, we form primary-backup site node pairs based on a predefined network distance between them. Once such candidate pairs for each VN node have been selected, the problem space is significantly reduced. Then, we embed the links among the pairs which satisfy both the BW and delay requirements from the VNE request. Multiple candidate pairs provide the flexibility to find out viable paths and optimize BW consumption.

Please cite this article as: A. Khan et al., Virtual Network Embedding for telco-grade network protection and service availability, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2016.03.017

4.1. Primary-backup pair searching

We consider all computing and storage resources (physical/virtual machines) connected to the same access router as one logical site node. Therefore, we model the PNI in such a way that each switching node does not attach more than one site node. To realize a $1+1$ protection scheme, a VN node needs to be mapped to two site nodes to form a primary-backup pair. If these two sites are connected to the same ingress/egress router, their network distance would be 2 hops. We consider this an unsuitable scenario where the primary and backup sites can reside very close to each other and increase the possibility of both being affected during large-scale disasters. Therefore, we propose that the distance between the primary and its backup site is at least 3 hops. Moreover, we limit the path length between the primary and backup site nodes by a threshold d_{th}, with a view to make sure that primary and backup sites are not too far away from each other. Hence, we have one more constraint for objective (3): For a candidate primary-backup pair of VN node i, we have,

$$3 \le |p(i_p \to i_b)| \le d_{th} \tag{13}$$

The primary-backup node distance constraint d_{th} is a network operational parameter which can be determined by a VNO. In order to geographically distribute primary and backup sites, a backup site node can also be considered even though the path length between the primary-backup pair is longer than d_{th}. However, a large value of primary-backup pair path length may increase the communication cost for service backup to achieve site protection and network downtime during migration from primary to backup sites due to a longer network distance. It should be noted that d_{th} is not a compulsory constraint and our proposed VNE algorithm can work without it. However, we believe that (13) helps in keeping a primary-backup pairs at the right distance. In practice, a Telecom operator knows how its PNI is deployed and what could be the right distance between a primary-backup pair. The operator can then use (13) to reflect the desirable distance in the VNE process or can advertise to a VNF. We will perform detailed investigation on (13) in our future work. In this paper, we assume distance in routing hops.

Searching all the candidate primary-backup pairs for all the VN nodes takes up time and computing resources, and is also unnecessary especially for a large scale PNI. Greedy node mapping algorithm is applied here for primary and backup site nodes mapping. For a VN node i, its capacity requirement is $C_v(i)$ and its BW requirement is the summation of the BW required from all its incident links $\sum_{j=1, j\neq i}^{|V_v|} B_v(i,j)$. We first sort the VN nodes according to their required capacity in decreasing order as $V_v = \{n_v^{1_c}, n_v^{2_c}, \ldots n_v^{|V_v|_c}\}$ and use this sequence to map the VN nodes to the site nodes in the PNI. The rationale behind this is that it is more difficult to embed a node with a high capacity requirement than a low capacity VN node. Site nodes with sufficient capacity are considered as candidates for VN node mapping and any two candidate site nodes form a candidate primary-backup pair. However, searching the optimized primary-backup pair is complex in terms of computing resource and time. Geographical constraints could be added to assist site node selection as in [7], but they are not considered here due to space constraints. We simply limit the number of selected site nodes per each VN node by a fixed number n_{nm} (e.g. $n_{nm} = 5$). In the next step, the candidate pairs are sorted according to the hop count in an increasing order. If several pairs have the same hop count, the pair with higher capacity is put on top of the pair with lower capacity. We choose the first n_{th} pairs as the selected primary-backup pairs. If the number of candidate pairs is smaller thann_{th}, all the pairs are selected. For a VN node i, it then has n_i candidate primary-backup pairs, where $n_i \le n_{th}$. All the site nodes in the candidate primary-backup pairs is given by $V_{pb}^i = \bigcup_{x=1}^{n_i} \{i_{p_x}, i_{b_x}\}$ where i_{p_x} and i_{b_x} represents the primary and backup nodes from the xth candidate pair respectively. Primary and backup nodes can be randomly decided within the primary-backup pairs or based on their capacity. For instance, a node with higher capacity is the primary node. We assume that each site node can only be mapped to one VN node within one VNE request. Therefore, after establishing V_{pb}^i, all the nodes within V_{pb}^i are reduced from the site nodes as $V_r = V_r \cap (V_{pb}^i)^c$, where $(V_{pb}^i)^c$ is the absolute complement ofV_{pb}^i.

Algorithm 1
PBPS algorithm pseudo-code. Q4

```
procedure PBPS (G_p, G_v)
  l_ns ← sort V_v (according to site capacity in decreasing order)
  for x ← 1, |V_v||V_v| do
    l_nm ← ϕ
    i' ← l_ns(x)
    for all n_r^i ∈ V_r do
      if C_r(i) ≥ C_v(i') then
        l_nm ← l_nm ∪^i
      end if
      if |l_nm| == n_nm then
        break
      end if
    end for
    if |l_nm| == 0 then
      PBPS Fails
    end if
    r ← 1, l_pb(x) = ϕ
    for i ← 1, |l_nm|, |l_nm| do
      for j ← i, |l_nm||l_nm| do
        if ≤ |p(l_nm(i) → l_nm(j))| ≤ d_th then
        l_pb(x) ← l_pb(x) ∪{ l_nm(i), l_nm(j)}
      end if
      if |l_pb(x)| == n_th then
        break
      end if
    end for
  end for
 end for
end procedure
```

The searching of primary-backup pairs for the rest of VN nodes continues within the updated set of V_r until all the VN nodes form their own primary-backup pair sets. Primary-backup paths are found in this step as described above. By combining them with the primary-primary paths, we ensure a complete connectivity between all nodes in two primary-backup pairs. The pseudo code of this Primary-Backup Pair Searching (PBPS) algorithm is shown in Algorithm 1.

4.2. VN link embedding

After selecting the primary-backup pairs for the VN nodes, the VN links between VN nodes are mapped to the paths on the PNI.

(a) *Link embedding based on link degree:* VN links are mapped sequentially based on the node degree of the VN nodes. First, we sort the VN nodes according to their node degrees in the decreasing order $V_v = \{n_v^{1_l}, n_v^{2_l}, \ldots, n_v^{i_l}, \ldots, n_v^{|V_v|_l}\}$. The VN links are mapped from higher node degree VN nodes to lower node degree VN nodes. We assume that the node degree of VN node i is k and the node IDs of its neighbors are denoted as $i^1, i^2, \ldots, i^k$. All the VN links that pass through the same VN node, e.g. $e_v(i, i^x)$ for $\forall x \in [1, k]$, have the same priority for mapping. In our algorithm, we sort i^x in increasing order and map $e_v(i, i^x)$ using the sorted neighbor sequence. In the next step, the shortest paths between all the candidate pairs are established. Constraint-based Shortest Path First (CSPF) [23] can be used to determine the path with sufficient BW between all pairs. If the path length of a candidate pair cannot

Please cite this article as: A. Khan et al., Virtual Network Embedding for telco-grade network protection and service availability, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2016.03.017

satisfy (13), this pair is removed from the candidate pair set. For a certain VN node, the BW constraint for its primary-backup pair is the summation BW of all links passing through this VN node.

This has two effects. Firstly, it correlates the node embedding with the link embedding. There is no point of selecting a node as the candidate which only has enough node capacity but not enough BW for ingress/egress links. Secondly, we only establish primary-primary paths in the PNI for inter VN node link embedding which is explained below.

(b) *Joint embedding of primary and backup paths:* Using the conventional $1+1$ site protection approach as shown in Fig. 2 b, a mesh needs to be created among primary and backup nodes to ensure connectivity during any number of site failures. For example, for the VNE request of two connected nodes of A and B, in the conventional approach, we need to find out A and B, their respective backup nodes A' and B' and then interconnect all four of these. In our approach, we only connect the primary-backup pairs A–A' and B–B', and then we connect the primary nodes A–B, as shown in Fig. 2 c.

This way, we reuse a part of the primary-primary path to ensure connectivity among all 4 nodes if both A and B fail simultaneously or if one of them fails. The rationale behind this is, if the primary and backup sites are not too far away from each other (which would perform badly for backup data synchronization and migration-based switch over), the paths between two primary nodes and their backups would overlap for a large section. Therefore, to map two VN nodes, identifying three paths would suffice. If B fails, the path can be switched to B' from the intersection point of A–B and B–B'. Path searching between primary sites is a standard routing problem, which can be solved by many existing shortest path algorithms [23]. Constraint-based Shortest Path First (CSPF) is used in this work for route discovery in the PN, in which the required BW is the constraint. If the shortest path found in this way does not meet the delay requirement, we conclude that other paths will not either.

The conventional approach practically embeds the VNE request twice, and then connects the nodes in the primary network with nodes in the backup network in a mesh. Therefore, the number of paths (P_{con}) for a VNE request $G_v = (V_v, E_v)$ in worst case would be, $P_{con} = 2E_v + \frac{2V_v(2V_v-1)}{2}$.

Contrary to this, in our scheme, we only find out the paths requested in a VNE request and additionally, the paths between the primary-backup pairs. Therefore, the number of paths (P_{new}) in our scheme would be, $P_{new} = E_v + V_v$, which is much smaller than P_{con} when V_v is large.

Please note that a primary path is interchangeable with its backup path in our scheme. For two given primary-backup pairs, the path with minimum length can be chosen to work as the primary path between these two pairs.

(c) *Embedded link adjustment:* After mapping the VN nodes, the VN link embedding process may not be successful in one trial. For instance, site nodes i and j are selected for VN node mapping, but there is no path that can be found between them after running CSPF because of BW scarcity. If this scenario occurs during the VN link embedding process, instead of using the BW constraint, we find out all paths between site nodes i and j that meet the delay constraint. We first take the shortest one and check if it overlaps with any previously embedded paths. If there are overlapping links, we check if releasing the previously embedded path would help releasing enough BW to embed the current path. If not, we go to the next shortest path between site nodes i and j, and do the same as above. If releasing the previously embedded path can help in embedding the current one, we release the previous one, embed the current VN link and re-embed the previous VN link as has been explained before. In order to avoid an algorithm loop, we mark all the overlapped links in the PNI. If further VN link embedding overlaps on such marked links again, we exit the algorithm and declare a failure. This is because releasing a previously embedded path going through the marked links means that when re-embedding this released path, it will have the same overlapping link(s), releasing which will take us into a loop. If all the paths satisfying delay constraints do not overlap with any previously embedded paths, path searching fails, which means that there is no path available in the PNI to meet the BW requirement.

Table 2
Simulation parameters.

Parameter	Value
Physical node capacity	[100, 300]
Physical link bandwidth	[100, 200]
Node capacity in VNE request	[1, 5]
Link bandwidth in VNE request	[1, 5]
Number of nodes in VNE request	[2, 5]
Link delay limit in VNE request	10 hops
Primary-backup node distance	4 hops
Link probability in VNE request	0.5

4.3. Algorithm complexity discussion

The algorithm complexity should be considered separately for node selection and link selection. For node selection, let us suppose that there are m virtual nodes and n physical nodes, in which we assume that there are cn resource node and $(1-c)n$ switching nodes. For each virtual node, we go through all possible resource nodes (with linear complexity $O(n)$), and for each selected resource node, we run the shortest path search to find a backup (with complexity $O(n^2)$). Therefore, the primary and backup site nodes searching for one virtual node has the complexity of $O(n)+O(n^2)$, which is $O(n^2)$. For m virtual nodes, we have complexity $m \times O(n^2)$ which is $O(n^2)$ when m is a constant number. For link selection, our proposed algorithm selects the shortest path between two primary nodes, therefore, it has the same complexity as the shortest path searching algorithm i.e. $O(n^2)$. Thus, the summation of the two parts is still $O(n^2)+O(n^2)$, which makes the complexity of our proposed VNE algorithm $O(n^2)$ and thus, can be solved in polynomial time.

5. Performance evaluation

In this section, we provide the simulation-based evaluation results of our proposal against the conventional approach to achieve $1+1$ site protection.

5.1. Simulation settings

(a) *Physical network:* To evaluate the performance of our algorithm, we have implemented a MATLAB based discrete event simulator. We randomly generate $|V_s|$ switching nodes in a circular area to form the core transport network. Each switching node selects the five closest neighbors in terms of distance as its direct neighbors. We also generate $|V_r|$ site nodes, which are randomly attached to one of the switching nodes. Any two site nodes cannot attach to the same switching node. The major simulation-related parameters are listed in Table 2. The capacity and bandwidth of each site node and switching node are uniformly distributed within the range [100, 300] and [100, 200] respectively. For capacity and bandwidth, we do not present any unit. For bandwidth, it could be K/M/Gbps. It could also be the numbers of λ in an optical network. For capacity, the unit could range from the number of physical machines, or VMs possible to accommodate to the numbers of CPU cores available. Network operators can decide in which granularity they want to perform capacity management.

Please cite this article as: A. Khan et al., Virtual Network Embedding for telco-grade network protection and service availability, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2016.03.017

Table 3
Comparison of best, conventional and our solution in terms of path length.

Number of switching nodes	13	16	18	22
Analytical model (best solution)	6.86	7.76	7.44	8.22
Conventional approach	9.3	9.7	10.06	10.78
New algorithm (our proposal)	9.3	9.7	10.06	10.78

(b) *VNE request:* For one VNE request, the number of requested VN nodes is set to a fixed number 3, and the probability of connectivity between every two VN nodes is 0.5. If all the nodes within the VNO graph are not connected via single or multiple hops, the VNO graph is regenerated. The bandwidth and capacity requirements are uniformly distributed within the range [1, 5]. The delay limit for a VNE request is set to 10 hops. The path length limit between a pair of primary and backup site nodes is 4 hops. The number of selected primary and backup candidate pairs n_{th} is set to 1.

5.2. Performance evaluation

In this section, we evaluate the performance of our proposed algorithm to achieve $1+1$ site protection. We simulate the arrival of VNE requests as discrete events. To test the maximum number of VNE requests that can be embedded in a PNI, we do not define the lifetime of a VNE request. Hence, once a VNE request is embedded in the PNI, it will remain in the network for the rest of the simulation. For averaging, different number of iterations are performed in different comparisons below. In general, 200 simulation runs are used, unless specified otherwise, e.g. in *(a) Overall comparison* below, we use 50 iterations as it is highly time- consuming to search the whole problem space in order to find out the best solution.

(a) *Overall comparison:* To evaluate the performance of our proposal, we use primary path length as an indicator which is the summation of the hop counts from all the primary paths of a VNE request. We compare the results averaged over *50 iterations* for the heuristic, with the optimal results derived from exhaustive search over the whole problem space to find out all the primary-backup possibilities in the analytical model. The analytical model is built on the conventional approach, which requires full mesh between all the primary and backup sites. In this study, the number of site nodes is fixed to 10, and the number of switching nodes varies between 13 and 22. Table 3 shows the results (*in terms of path length*) obtained from different types of approach, where we compare the embedded primary path lengths of the three types of approach. What should be noticed is that, to compare the bandwidth consumption efficiency from different VNE solutions, we use the same node mapping mechanism for primary-backup pair selection in both conventional and our proposed algorithms to eliminate the influence from the VN node embedding. This is also the reason why the primary path lengths for these two algorithms (conventional and our proposal) are the same as shown in Table 3. The optimal solutions are obtained by searching the complete problem space, and they therefore indicate the results that are not only optimized for node mapping but also for link mapping. The results from the analytical model are superior to the heuristic algorithms, but the required computing resources and computing time necessary to achieve so cannot be neglected.

(b) *Bandwidth consumption:* As emphasized before, BW usage is one of the main metrics to evaluate the VNE algorithm efficiency. Therefore, we compare the BW consumption for each VNE request by using both the conventional algorithm and our proposal. We set up a PNI with 50 switching nodes and 20 site nodes. In order to test the overall capability of the PNI to host the VNs, within one simulation run, we input 20 VNE requests to one PNI sequentially by using our proposal and the conventional approach. If one VNE request can be embedded, its requested site node capacity and link bandwidth are reserved for this VNE request. Otherwise this VNE request is dropped. Fig. 3a depicts the BW consumption for each VNE requests within one simulation run. A VNE request that is generated earlier is associated with a smaller sequence ID as shown in the *X*-axis in the figure. The *Y*-axis is the allocated BW within the PNI for each VNE request with backup solution. If a VNE request is rejected, the assigned BW is set to zero. The results shown in the figure indicate that our proposed algorithm consumes much less BW to embed a VNE request and it can also accept more VNE requests than the conventional algorithm due to its better resource allocation efficiency. To gain better understanding about the algorithm performance, the simulation was repeated 200 times. For all the embedded VNE requests, the average BW consumption for the embedded VNE request with the same sequence ID is shown in Fig. 3b, which indicates that the conventional algorithm reserves around three times the BW compared to our proposed algorithm. Moreover, the reserved BW while using the conventional algorithm has a higher fluctuation than our proposed algorithm.

(c) *Path length:* Based on the same simulation setup described in the previous subsection, the total primary path length (paths from primary site to primary site) and backup path length (paths from backup site to backup site) has been investigated for each VNE request. The results are shown in Fig. 4a. As shown in the figure, for both conventional and our proposed algorithms, the primary path length is not always shorter than the backup length. This is due to the fact that we setup the role of primary and backup nodes (based on the site node capacities) before link embedding. Therefore, the primary path length is not necessarily shorter than the backup path. However, as mentioned before, primary sites can be selected based on the shorter path as well, as primary and backup sites are interchangeable in our proposal. As we can see, by using our proposed algorithm, the path length of the backup path has a higher probability to be longer than the primary path. This phenomenon occurs because, after assigning the primary backup pairs for a VNE request, we first select the shortest path in the primary–primary site nodes which satisfies the BW requirement. By doing so, the primary path is always the optimized solution. Due to the fact that our proposed algorithm tries to reuse part of the primary path to reduce the bandwidth reservation for backup paths, the selection of backup paths might not be optimized. This may result in longer backup path lengths than the primary paths. In this work, we do not optimize the backup path length. During a failure and the consequent site switchover, the path to the backup PNI node could be optimized step by step. In such cases, a new primary backup site pair will be formed, or the failed primary site node can become the backup site node after its recovery and an unnecessary switchover back to it can be avoided. We will address this in our future work. The average path length for each embedded VNE request with the same sequence ID is shown in Fig. 4b, which is based on 200 simulation runs. It also indicates that the average backup path length of our proposed algorithm is longer than the average primary path length, which is not the case for the conventional algorithm.

(d) *VNE request acceptance ratio:* In order to gain better understanding about the above obtained results from Figs. 3b and 4b, the VNE request acceptance ratio is plotted in Fig. 5, which is defined as the ratio between the total number of accepted VNE requests (successfully embedded) and the total number of simulation iterations for this setup, which is the summation of successfully embedded and unsuccessfully embedded VNE requests. As shown in the figure, it indicates that our proposed algorithm sees a mild decrease in acceptance rate as the number of embedded VNs increases in the network. In comparison, the decrease shown by

Please cite this article as: A. Khan et al., Virtual Network Embedding for telco-grade network protection and service availability, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2016.03.017

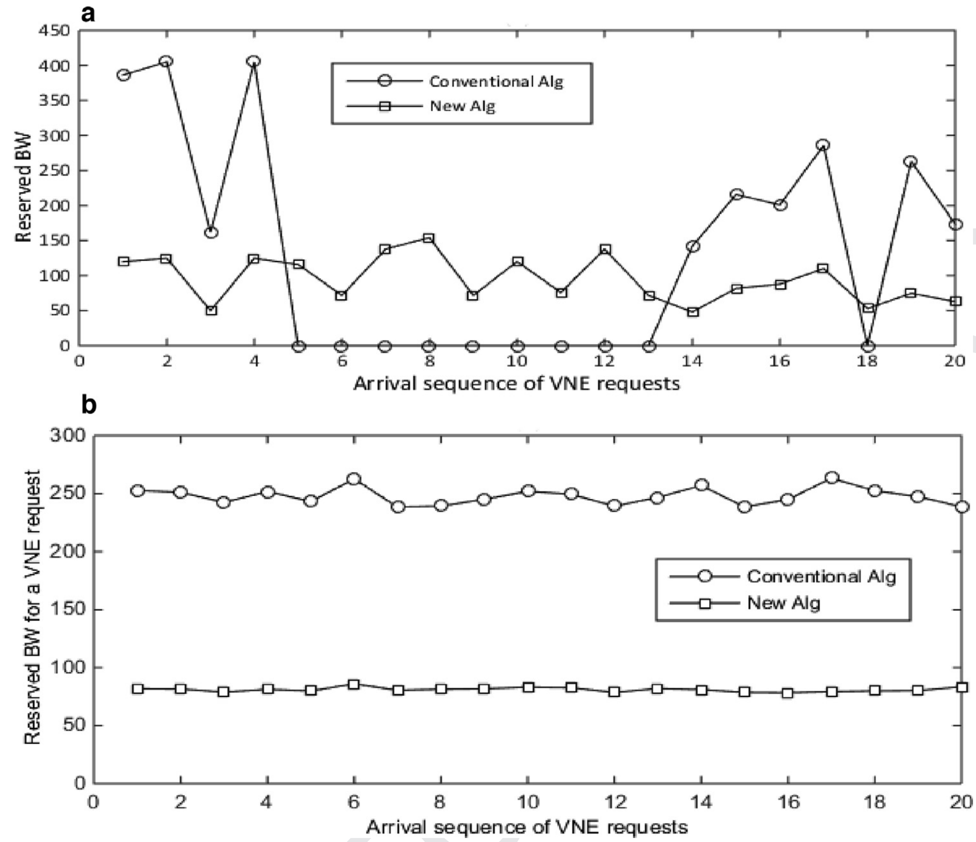


Fig. 3. Bandwidth consumption comparison between the conventional approach and our proposed algorithm.

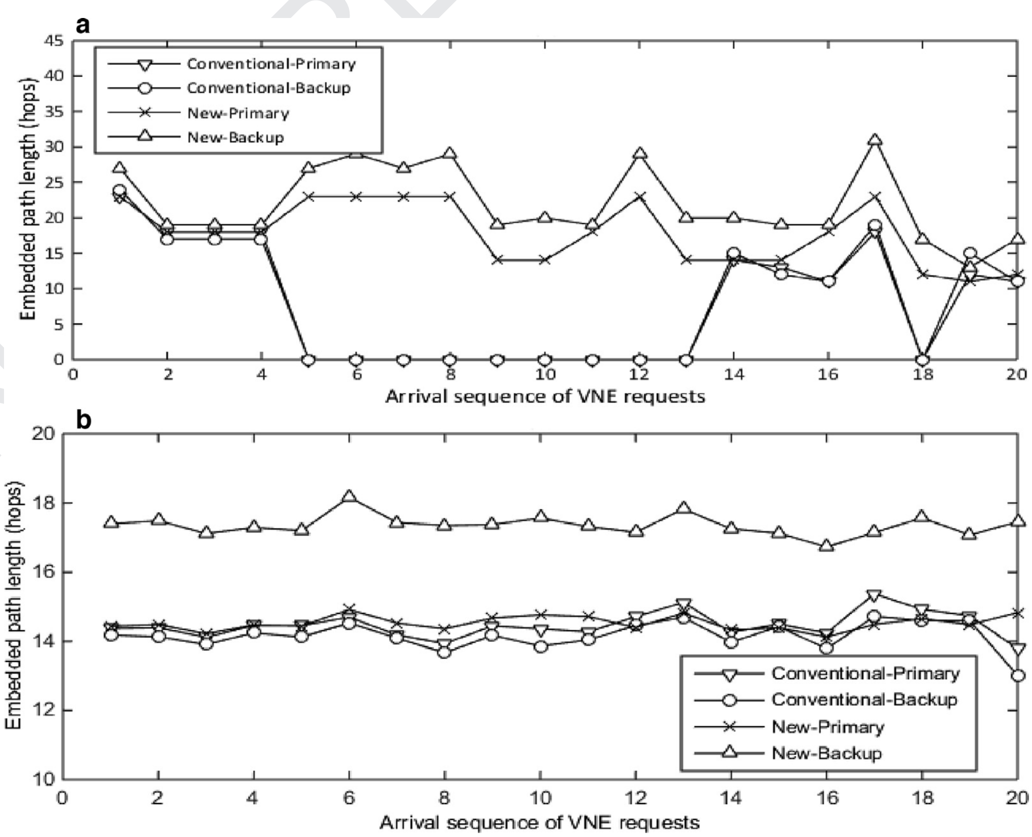


Fig. 4. Path length comparison between the conventional approach and our proposed algorithm.

Please cite this article as: A. Khan et al., Virtual Network Embedding for telco-grade network protection and service availability, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2016.03.017

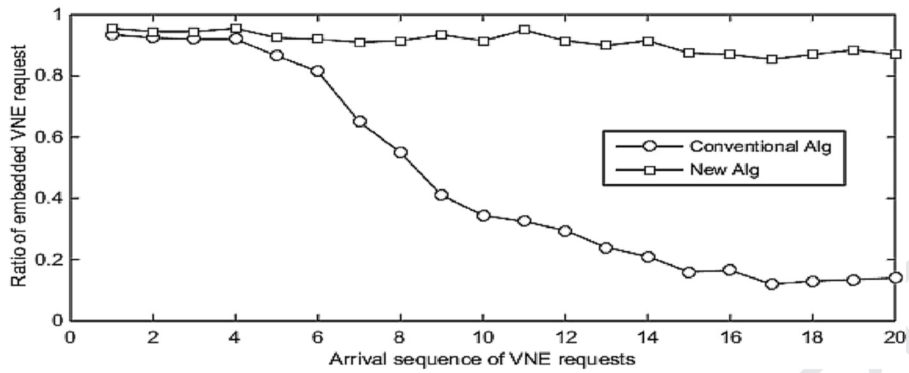


Fig. 5. VNE requests acceptance ratios for the conventional approach and our proposed algorithm.

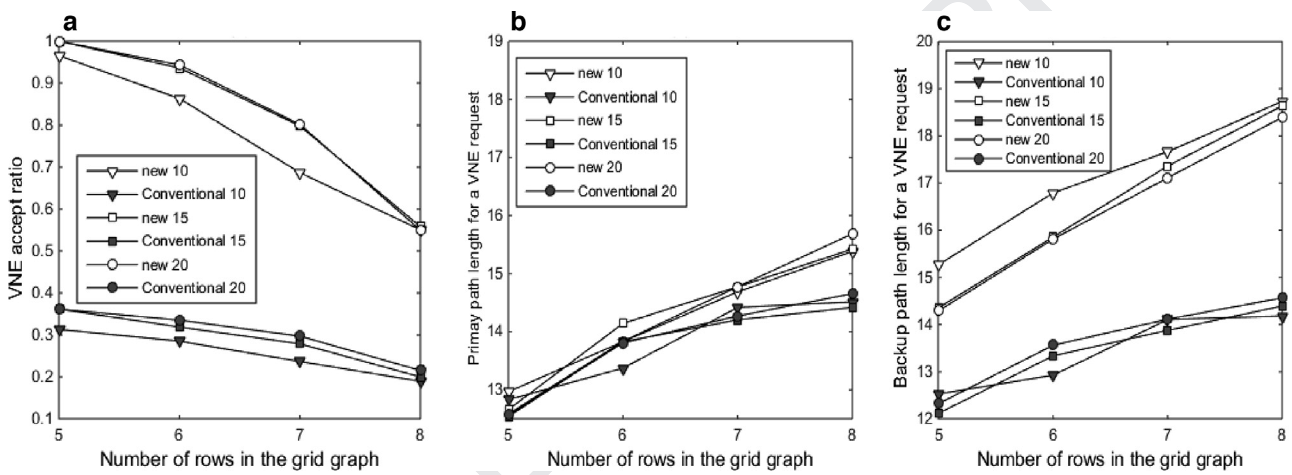


Fig. 6. Physical network scale influence on the embedding performance.

the conventional algorithm is more dramatic. This is because the more VNE requests are embedded in the PNI, the less resources are left to embed the later VNE requests (having higher sequence ID). In our scheme, we require much less resource than the conventional scheme, hence the VNE request acceptance ratio tends to be higher. In comparison, the conventional scheme fails to come up with an embedding solution when resources become scarce.

(e) *Physical network scale influence:* The size of a PNI also influences the VNE performance. To investigate the PNI scale influence, we generate a core transport network with a grid topology. The number of switching nodes is set to 5×5, 6×6, 7×7 and 8×8 in different simulation scenarios. Moreover, the number of site nodes is varied from 10, to 15 to 20, and they are randomly attached to the switching nodes. The results shown in Fig. 6 are the mean value after 100 iterations, in order to average the randomness effect. The number of VNE requests is set to 30 in each run of the simulation. We further compare the VNE request acceptance ratio under different scales of transport network and the results are shown in Fig. 6a. Our proposed algorithm has a much higher VNE request acceptance ratio than the conventional approach. When the network size increases, the acceptance ratios drop in both cases. There are two causes of VNE request drop. Once the core transport network becomes larger, the average distance between site nodes also increases when the number of site nodes is fixed. Hence, the first cause of VNE request drop is the fact that the path length between the selected site nodes cannot satisfy the delay constraint specified by the simulation. This is a direct influence from the PNI topology. The second cause of VNE request drop lies in the BW within the PNI not being sufficient to host the VNE requests. When we further investigate the causes of the VNE request drop by using these two algorithms, we find that the VNE request drop in our algorithm is mainly due to the first cause, and for the conventional one, it is mainly due to the second cause.

As shown in the figure, the results also indicate that the number of site nodes also has certain impact on the acceptance ratio, i.e. the more site nodes we have, the better the acceptance ratio. More number of site nodes puts them closer to each other which helps in finding out paths requiring lower latency. Once the site node number becomes higher, the reason for embedding failure shifts to the scarcity of transport network resources.

In Fig. 6b and c, we show the average primary and backup path length for the embedded VNE requests in different PNI sizes. As shown in the figures, compared to the backup path length, the difference of primary path lengths between our proposed algorithm and the conventional algorithm is smaller. In our scheme, the primary path is the shortest path whereas in the conventional scheme, all paths are the shortest paths. Therefore, path lengths are similar for the primary paths in both schemes, whereas the backup path is longer in our scheme. In relation to this, it should be noted that our proposal has higher success rate (acceptance ratio) in larger PNI which also makes the path length longer while averaged over the total number of successful embedding.

Although our backup paths are few hops longer (Fig. 6c), we still consume fewer link resources (Fig. 4b). This is because we require to find and establish fewer links to ensure the necessary connectivity (Fig. 2b and c). Along with the reasons explained above for network scale, this is also the reason why we achieve higher acceptance ratio. As more and more VNE requests are embedded and consequently resources run out, the algorithm which

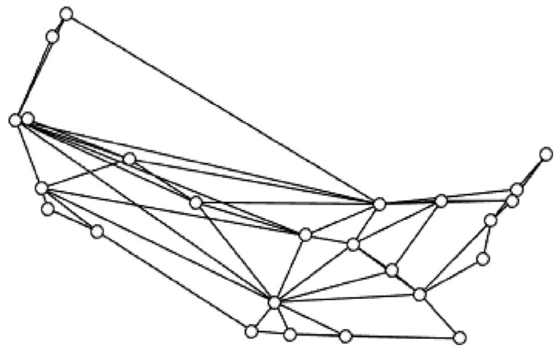


Fig. 7. US IP backbone topology.

consumes fewer resources achieves higher acceptance ratio i.e. is successful in the embedding when resources are scarce.

An implication of longer backup path is the usage of suboptimal paths after failure. Once a primary VN node is switched to its backup node, the path length to the backup node from other connected VN nodes will be suboptimal. These paths can slowly be optimized step by step once switching over to the backup node is completed without interfering with the intended service delivery.

(f) *Real network topology:* In order to understand our proposed algorithm performance in real networks, we use the data from topology zoo [25]. The used topology (Fig. 7) is from ATT North America with 25 switching nodes. We re-run the simulation with the same parameter setup as mentioned before (see Table 2). We show the Cumulative Distribution Function (cdf) of the embedded path length in Fig. 8a when the number of site nodes vary from 10, to 15, to 20.

In Fig. 8 a, total path length of all successful VNE is plotted. Although our scheme has longer backup paths, it still performs better as it realizes a requested VN with less number of paths than the conventional approach. As illustrated by the results, the variation of site nodes (i.e. from 10 to 20) has influence on the embedded path length for a VNE request by using our scheme compared to the results obtained by using conventional scheme. Fig. 8b reconfirms a result from Fig. 6a, which shows that VNE acceptance ratio increases with the number of site nodes available. However, as seen in Fig. 6a, this increase becomes less prominent after a certain number of site nodes (15 site nodes in Fig. 8b). After this point, the VNE failures occur due to the scarcity of transport resources.

6. Realization under NFV context

In this section, we explain the relevance of our work within the context of Network Functions Virtualization (NFV) as specified by European Telecommunications Standards Institute (ETSI) Industry Specification Group (ISG) NFV. ETSI ISG NFV has been formed by major operators and vendors to define global standard specifications for Telecom network virtualization. A virtualized implementation of a Network Function (e.g. mobile core nodes [2]) is called a Virtualized Network Function (VNF) which may contain multiple sub-components running in different VMs. These VMs could be located in the same datacenter i.e. site, or in multiple sites, according to the VNF deployment requirements and policies. In this paper, we have made the problem formulation such that a VNE request is embedded to geographically distributed sites connected by a Transport Network (TN). However, when a single VNF which is analogous to a VN node in our model, consists of multiple sub-components, the sub-components may need to be embedded within the same site. Besides, there could be a request to embed a chain of VNFs and each of the VNFs may contain multiple sub-functions as shown in Fig. 9. Whatever may be the scenario is, the VNE request can be formed as $G_v = (V_v, E_v)$ as explained in Section 3.2. In such a case, V_v represents the sub-components of a VNF, and E_v represents the intra-VNF links connecting those sub-components.

Fig. 10 is the NFV reference architecture proposed by ETSI ISG NFV [24]. Functional blocks such as the Orchestrator, Virtualized Infrastructure Manager and VNFs as shown in Fig. 10 are the major relevant entities for VNF embedding. The embedding request is sent to NFV Orchestrator (NFVO), which will trigger the embedding decisions on the VNF Managers (VNFMs). VNFMs manage the lifecycles of VNF instances. Virtualized Infrastructure Manager (VIM) controls and manages the compute, storage and network resources analogous to the PNI. A network within a site (e.g. datacenter) under the NFV context refers to the datacenter networks, which normally consist of different types of switches, e.g. core switches, aggregation switches and top of the rack switches, and all the switches are deployed according to certain pre-defined topology, e.g. two/three-tier tree, fat-tree, etc. If multiple sites are involved in deploying one VNF or a VNF chain, the TN (core transport network in Section 3) is also involved in the process. In this case, embedding needs to be done in two steps. Step 1 is the top-tier embedding, which selects suitable sites and links within the TN

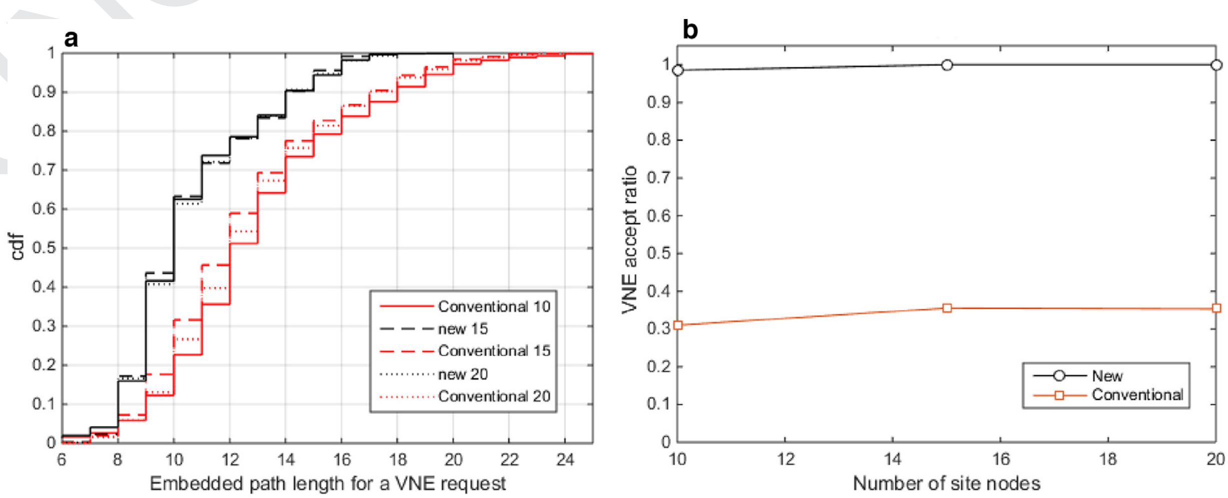


Fig. 8. Comparison of the conventional and our proposed algorithm in real network topology.

Please cite this article as: A. Khan et al., Virtual Network Embedding for telco-grade network protection and service availability, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2016.03.017

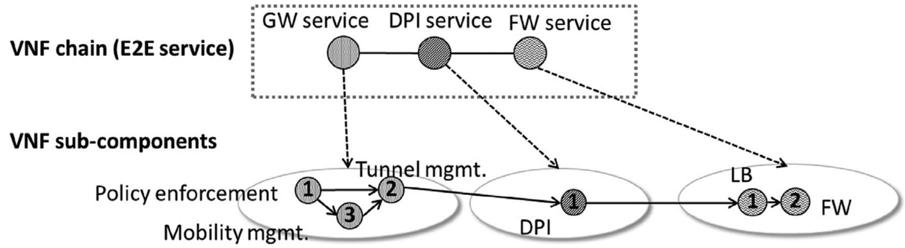


Fig. 9. VNF chain example.

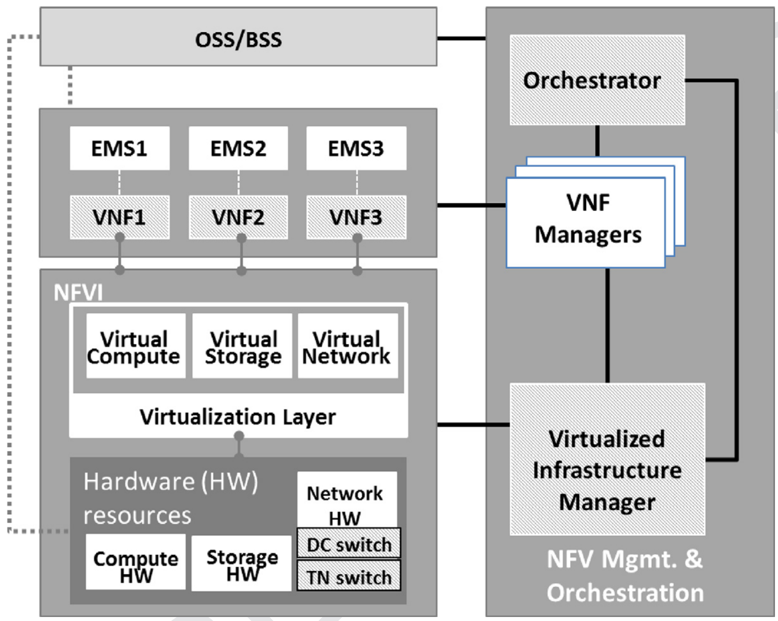


Fig. 10. NFV reference architectural framework.

that interconnect the sites. Step 2 is the local embedding: i.e. the sub-components of each VNF need to be within a site where multiple VMs are instantiated on different physical machines and connected through intra-site links. Therefore, the infrastructure graph for physical network $G_p = (V_p, E_p)$ should be formulated and updated accordingly by taking the switches of different types and capabilities inside the site into consideration. If the TN is not involved, the VNE algorithm can be run in VIM. If multiple VIMs and the TN between are involved in the VNE process, the top tier VNE will run in the Orchestrator and our proposed algorithm needs to be deployed in both the VIM and the Orchestrator to ensure overall protection.

7. Conclusions

In this paper, we have proposed and evaluated a heuristic algorithm which realizes VNE for $1+1$ site protection to meet Telco-grade network protection and service availability requirements. In approaching this issue, we do not relax the problem by using path splitting as that is unrealistic in operational networks. We also distinguish between server nodes and switching nodes, as a server node cannot be mapped on a switching node.

Our algorithm achieves sound correlation between node and link embedding, consumes less bandwidth and provides higher success rate than any conventional approach. Evaluation results also show that our algorithm performance in finding out a VNE solution is close to the theoretical ceiling. In our future work, we will address VN node mapping over multiple sites in order to further optimize site resources and backup path optimization in order to further reduce the consumption of link resources. In addition, we intend to further analyze the performance of the proposed VNE algorithm in failure scenarios focusing of service recovery latency.

References

[1] W. Kellerer, J. Widmer, A. Khan, D. Jurca, Future mobile network: use cases for network virtualization, in: Proceedings of the 9th Wuerzburg Workshop on IP: Joint ITG and EuroNF Workshop Visions of Future Generation Networks (EuroView'09), July 2009, Germany, 2009.
[2] 3GPP TS 23.401, General Packet Radio Service (GPRS) enhancements for Evolved Universal Terrestrial Radio Access Network (E-UTRAN) access, Technical Specification Group Services and System Aspects, 3rd Generation Partnership Project.
[3] Y. Takano, A. Khan, M. Tamura, S. Iwashina, T. Shimizu, Virtualization-based scaling methods for stateful cellular network nodes using elastic core architecture, in: Proceedings of the 6th IEEE International Conference on Cloud Computing Technology and Science (CloudCom'14), December 2014, 2014.
[4] T. Shimizu, T. Nakamura, Kiuchi M, A. Iwata, Y. Kubota, M. Ohhashi, An experimental evaluation of dynamic virtualized networking resource control on an evolved mobile core network, in: Proceedings of the Humanitarian Technology Conference (HTC'13), August 2013, IEEE, 2013.
[5] A. Khan, W. Kellerer, K. Kozu, M. Yabusaki, Network sharing in the next mobile network: TCO reduction, management flexibility, and operational independence, IEEE Commun. Mag. 49 (10) (2011) 134–142.
[6] S. Sharma, M. Chawla, A technical review for efficient virtual machine migration, in: Proceedings of the International Conference on Cloud & Ubiquitous Computing & Emerging Technologies (CUBE'13), November 2013, IEEE, India, 2013.
[7] M. Chowdhury, M.R. Rahman, R. Boutaba, ViNEYard: virtual network embedding algorithms with coordinated node and link mapping, IEEE/ACM Trans. Netw. 20 (1) (2012) 206–219.
[8] A. Khan, X. An, D.P. Caparros, W. Kiess, Virtual network embedding algorithm for one-to-one site protection, in: Proceedings of the IEEE Global Communications Conference (GLOBECOM'13), December 2013, Atlanta, GA, 2013.

Please cite this article as: A. Khan et al., Virtual Network Embedding for telco-grade network protection and service availability, Computer Communications (2016), http://dx.doi.org/10.1016/j.comcom.2016.03.017

[9] T. Yamasaki, A. Khan, M. Tamura, T. Shimizu, S. Iwashina, A database access scheme for elastic-core architecture, in: Proceedings of the IEEE APWiMob'14, August 2014, Bali, Indonesia, 2014.

[10] A. Jarray, A. Karmouch, Cost-efficient mapping for fault-tolerant virtual networks, IEEE Trans. Comput. 64 (3) (2015) 668–681.

[11] Z. Qiang, W.H. Qiang, F.G. Sheng, L.H. Wu, Heuristic survivable virtual network embedding based on node migration and link remapping, in: Proceedings of the IEEE 7th Joint International Information Technology and Artificial Intelligence Conference (ITAIC'14), December 2014, Chongqing, China, 2014, pp. 181–185.

[12] J.A. Johnson, Optimization of migration downtime of virtual machines in cloud, in: Proceedings of the IEEE 4th International Conference on Computing, Communications and Networking Technologies (ICCCNT'13), July 2013, India, 2013, pp. 1–5.

[13] Y. Chen, Ayoubi S, C. Assi, CORNER: COst-Efficient and Reliability-Aware Virtual NEtwork Redesign and Embedding, in: Proceedings of the IEEE 3rd International Conference on Cloud Networking (CloudNet'14), October 2014, Luxembourg, 2014, pp. 356–361.

[14] M. Pourvali, H. Bai, F. Gu, K. Shaban, M. Naeini, J. Crichigno, M. Hayat, S. Khan, N. Ghani, Virtual network mapping for cloud services under probabilistic regional failures, in: Proceedings of the IEEE 3rd International Conference on Cloud Networking (CloudNet'14), October 2014, Luxembourg, 2014, pp. 407–412.

[15] B. Guo, C. Qiao, J. Wang, H. Yu, Y. Zuo, J. Li, Z. Chen, Y. He, Survivable virtual network design and embedding to survive a facility node failure, IEEE J. Lightwave Technol 32 (3) (2014) 483–493.

[16] A.N. Patel, Z. Ye, P.N. Ji, C. Qiao, Survivable virtual infrastructure mapping with shared protection in transport software defined networks (T-SDNs), in: Proceedings of the OptoElectronics and Communication Conference and Australian Conference on Optical Fibre Technology (OECC/ACOFT '14), July 2014, Melbourne, Australia, IEEE, 2014, pp. 679–681.

[17] G. Liu, S. Su, Virtual network mapping algorithm with substrate node reliability awareness and shared-path protection, in: Proceedings of the IEEE 10th International Conference on High Performance Computing and Communications & Embedded and Ubiquitous Computing (HPCC_EUC '13), November 2013, Zhangjiajie, China, 2013, pp. 636–643.

[18] T. Guo, N. Wang, K. Moessner, R. Tafazolli, Shared backup network provision for virtual network embedding, in: Proceedings of the IEEE International Conference on Communications (ICC'11), June 2011, Japan, 2011.

[19] M.R. Rahman, I. Aib, R. Boutaba, Survivable virtual network embedding, in: Proceedings of the 9th International IFIP Networking Conference (NETWORKING'10), May 2010, Chennai, India, Springer, Berlin, 2010, pp. 40–52.

[20] H. Yu, Migration based protection for virtual infrastructure survivability for link failure, in: Proceedings of the Optical Fiber Communication Conference and Exposition and the National Fiber Optic Engineers Conference (OFC/NFOEC'11), March 2011, Los Angeles, CA, IEEE, 2011.

[21] R.R. Oliveira, D.S. Marcon, L.R. Bays, M.C. Neves, L.P. Gaspary, M.P. Barcellos, D. Medhi , Opportunistic resilience embedding (ORE): toward cost-efficient resilient virtual networks, J. Comput. Netw. (2015) (available online). Q6

[22] S. Herker, X. An, W. Kiess, A. Kirstaedter, Path protection with explicit availability constraints for virtual network embedding, in: Proceedings of the IEEE 24th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'13), September 2013, London, UK, 2013, pp. 2978–2983.

[23] F. Kuipers, P. Van Mieghem, T. Korkmaz, M. Krunz, An overview of constraint-based path selection algorithms for QoS routing, IEEE Commun. Mag. 40 (12) (2002) 50–55.

[24] European Telecommunications Standards Institute (ETSI), "network functions virtualization- introductory white paper ", October 2012 https://portal.etsi.org/nfv/nfv_white_paper.pdf . Q7

[25] The University of Adelaide, The Internet Topology Zoo, http://www.topology-zoo.org.