Networking 2013 1569696409

p-Cycle-Based Node Failure Protection for Survivable Virtual Network Embedding

Abdallah Jarray
SITE, University of Ottawa
Ottawa, Ontario, K1N 6N5 Canada
Email: ajarray@uottawa.ca

Yihong Song
SITE, University of Ottawa
Ottawa, Ontario, K1N 6N5 Canada
Email: ysong@uottawa.ca

Ahmed Karmouch
SITE, University of Ottawa
Ottawa, Ontario, K1N 6N5 Canada
Email: Karmouch@site.uottawa.ca

***Abstract*—Network Virtualization offers to Physical Infrastructure Provider (PIP) the possibility to smoothly roll out multiple isolated networks on top of his infrastructure. A major challenge in this respect is the embedding problem which deals with the mapping of Virtual Networks (VNs) resources on PIP network. In literature, a number of research proposals have been focused on providing heuristic approaches to solve this NP-hard problem while most often assuming that PIP infrastructure is operational at all time. In virtualization environment a single physical node failure can result in single/multi logical link failure(s) as it effects all VNs with a mapping that spans over. Setup a dedicated backup for each VN embedding, i.e., no sharing, is inefficient in terms of resource usage. To address these concerns, we propose in this work an approach for VN mapping with combined physical node and multiple logical links protection mechanism (VNM-CNLP) that: (a) uses a large scale optimization technique, developed in our previous work, to calculate in a first stage a cost-efficient VN mapping while minimizing the effects of a single node failure in VN layer, and (b) proposes in a second stage link p-Cycle based protection techniques that minimize the backup resources while providing a full VN protection scheme against a single physical node failure and a multiple logical links failure. Our simulations show a clear advantage of VNM-CNLP approach over benchmark in terms of VN backup cost and resources utilization.**

***Index Terms*—Virtual network embedding, survivability, single physical node failure, multiple logical links failure, p-Cycle, Column Generation, Integer Linear Programming.**

I. Introduction

Network virtualization [1], [2] provides more flexibility in network provisioning as it offers to Physical Infrastructure Provider (PIP) the possibility to smoothly roll out multiple isolated networks on top of his infrastructure. Accordingly, network virtualization has attracted a lot of attentions in the past few years, by providing a new paradigm for scalable and on-demand provisioning of multiple virtual networks (VNs) over a shared substrate network. However, the requirement for the resiliency guarantees of substrate network also increases remarkably. Compared to link, the protection for node failure is more desirable, because the impact of node failure is more severe, especially in network virtualization environment, as it can prune several VNs simultaneously.

The most efficient approach to handle physical node failure is to establish a set of backup paths for primary paths in a proactive manner on substrate network. This enables fast recovery from the failure and transport of data is almost undisturbed. But the shortcoming, resources and capacity usage are large, so researchers [3] have adapted some alternative technologies, such as, ring, mesh and p-Cycle mechanisms to overcome this defect.

However, most often protection approaches proposed in literature deal with physical link failures [4]-[6]. A very few proposals [7]-[9] have been dedicated to provide backup mechanism in case of single physical node failure. Moreover, they omit to consider any logical linking among protected paths as they are proposed for optical and multi-protocol label switched (MPLS) networks. In fact, in optical networks the aim of protection is only to guarantee that physical nodes are still linked together in the presence of node failures (weak resiliency). However, in network virtualization environment, we need to ensure that all virtual links are intact in the presence of failures. In other words, for a VN request, we have to guarantee survivability of all the VN links simultaneously (strong resiliency).

Furthermore, in network virtualization environment, a physical node failure not only effects the physical layer, but also propagates to the virtual network layer, and if the virtualization is recursive, any number of upper logical layers [4]. Dropping such failing node will affect large number of upper VN layers. Accordingly, node failure protection mechanisms can be provided at different layers. In case of a physical layer protection mechanism, a node failure is detected directly at the physical layer. The affected VN mapping paths can be repaired by activating the backup paths or by rerouting affected VN traffic on the fly. This is in general transparent to the VN layers. On the other hand, in case of reactive failure protection at the VN layer, a physical failure propagates to the logical layer where it is detected. Routing paths in the VN topology are re-computed to avoid the failing virtual links, which is transparent to the physical layer.

It is well known that a single physical node failure is logically equivalent to multiple physical link failures in a network [10]. Moreover, in virtualization environment, a single physical node failure can also result into a multiple VN links failure (logical links failure) as virtual links belonging to the same VN can be mapped to physical paths that span over the failing physical node. Accordingly, our focus will be on the physical layer protection mechanism while taking into account constraints from the logical layer (virtual networks) in order to guarantee a full VN protection against a single physical node failure and multiple logical links failures. In other word, we want to provide a VN mapping backup against physical node failure that takes into account constraints from the logical layer related to the VN topology and QoS requirements.

To do so, we propose in this work an approach for VN mapping with a combined protection mechanism against physical node failure and a multiple logical links failure (VNM-CNLP) that: (a) uses a large scale optimization technique, developed in our previous work [17], to calculate in a first stage a cost-efficient VN mapping while minimizing the effects of a single physical node failure on the VN layer, and (b) proposes in a second stage a link p-Cycle based VN protection approaches that minimizes the backup resources while providing a full protection scheme against a single physical node failure as well as a multiple logical links failure.

The remainder of the paper is organized as follows. Next Section presents related works. Section III defines the VN embedding problem with resiliency guarantee. Section IV presents the VNM-CNLP approach. Section V lists the proposed performance evaluation metrics, followed by the numerical results. Finally, Section VI concludes the paper.

II. Related Works

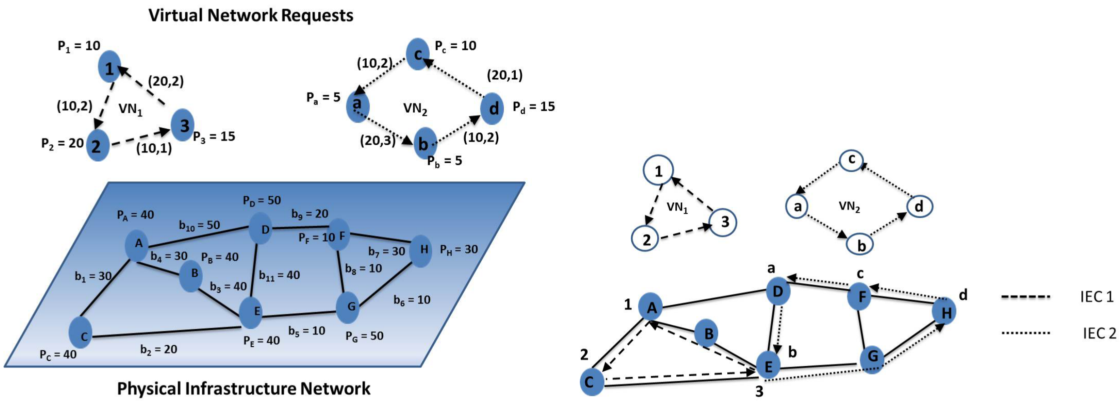
VN embedding problem is known to be NP-hard as it can be reduced to multi-way separator problem [16]. Accordingly, proposals in literature have been focused on restricting the problem in some particular scenarios. Assuming infinite network capacities in order to obvious admission control [11]. Ignoring either virtual link or node QoS requirements [11], [12]. Relaxing embedding path-splitting constraint [14]. Focusing on specific VN topologies [12]. Designing greedy embedding algorithms [12], [13]. More recently Chowdhury et *al.* [15] proposed a mathematical programming based scheme to coordinate node and link mapping. The proposed scheme handle online VNP demand and introduces a better correlation between node and link embedding phases. However, provided solution might have a gap compared to approach based-on a simultaneous node and link mapping. To handle these concerns, we proposed in recent previous work [17] a cost-efficient VN embedding approach that makes use of large scale optimization technique called Column Generation approach.

As aforementioned a great number of proposals have been focused on VN embedding problem, but most of them are done under the normal operation of substrate. A few works have been dedicated to provide resiliency for VN mapping. Resiliency of VN service is an urgent requirement for PIP. Indeed, service interruption due to an intermediate failing substrate node will result into the violation of Service Level Agreements (SLAs) contracted between PIP and VNs owners, resulting in economic penalties and revenue loss for PIP. To address resiliency concern, one solution could be the allocation of a dedicated backup for each VN mapping. However, it is not efficient in terms of resource usage from the PIP point of view. Indeed, it is expected that backup resources can be shared for the protection of several VNs keeping room for the mapping of more VN requests.

In [9], authors proposed a node and link failure protection approach in the context of optical networks. The main focus in this proposal is on the physical layer. There is no information about the upper logical layers. Accordingly, a physical failing node or link can have a severe effects on the logical layer which is the case in the network virtualization environment. Indeed, a single physical node failure can result in multiple logical links failure as virtual links belonging to the same VN can be mapped into physical paths that span over the failing node.

In [8], a combined node and span protection approach with a single or shared set of node-encircling p-Cycles (NPECs) is proposed. In other words, each node is protected by a single dedicated or a shared NPECs. The main concern as aforementioned in [9] is related to fact that the node protection proposal omit to consider any logical linking among protected paths. Accordingly, the proposal is not well adapted to the context network virtualization under consideration in the current paper.

In [7], authors proposed a node failure protection scheme. Again, the main focus in this node failure protection proposal is on providing a mechanism that can handle physical node failure omitting any consideration related to virtual networks in the upper layers. The main contribution of this proposal is an extension of ordinary span-protecting p-Cycles to include more candidates in order to increase the protection efficiency in terms of resources usage. The main idea consists of that in addition to that each p-Cycle has it own spans for which it provides intended span failure protection, every p-Cycles will also happen to intercept a number of working flows upstream and downstream of nodes on p-Cycles.



(a) Topologies of VNs and physical network

(b) Independent Embedding Configuration

Figure 1: Embedding VNs on substrate network

III. Survivable VN Mapping Problem

A. Substrate network

The physical infrastructure network S is represented by an undirected graph $G_s = (W_s, L_s)$, where W_s denotes the set of substrate nodes and L_s the set of bidirectional links. Figure 1a shows an example of a physical infrastructure network, where each physical link $l \in L_s$ offers a bandwidth capacity b_l (number over link) and each substrate node $u \in W_s$ has a Central Processing Unit (CPU) capacity p_u (number over node). We introduce a bandwidth unit cost c_l per each substrate link $l \in L_s$, for load balancing purpose. Similarly, we associate a CPU unit cost for each substrate node $u \in W_s$. We denote by Π^s the set of paths in graph G_s.

B. VN request

Similarly, a Virtual Network $n \in N$ is represented by a directed graph $G_n(A_n, E_n)$. The QoS requirements of each virtual link $e \in E_n$ belonging to class $j \in Q_1$ are defined by the couple (b^j, d^j). b^j is the required bandwidth and d^j is the maximum number of switching nodes as an indirect way to upper-bound the end-to-end delay. We assume that number of used links has a neglect effect on the end-to-end delay. Similarly, QoS requirements of each virtual node $a \in A_n$ belonging to class $j \in Q_2$ are defined by the couple (p_j, t_j). p_j is the required CPU and t_j is the potential nodal mapping locations. We denote by $c(a)$ (resp. $c(u)$) the QoS class of virtual link a (resp. node u). Π^e_{uv} is the set of all shortest paths between embedding nodes (u, v) assigned to virtual link $e \in E_n$. Figure 1a shows an example of two VN requests VN_1 and VN_2. We assume that embedding of each VN request n provides revenue P_n.

C. VN embedding

The embedding (called also mapping, we will exchange their use through the document) of each VN request can be decomposed into node and link mapping as follows.

1) Node mapping: Each virtual node $a \in A_n$ from the same VN request n is embedded to different substrate node $u \in W_s$ by mapping: $M_{\text{N}} : A_n \mapsto W_s$

2) Link mapping: Similarly, each virtual link $e \in E_n$ from the same VN request n is embedded to a substrate path $\pi^e_{uv} \in \Pi^s$ by mapping: $M_{\text{L}} : E_n \mapsto \Pi^s$, where (u, v) are substrate nodes assigned to virtual nodes (s, d) source and destination nodes of virtual link e respectively.

D. VN protection

In addition to mapping of primary flow of a VN request, it is required to allocate backup resources (paths with spare capacity) in order to guarantee the recovery in case of a single physical failure. Our focus will be on single node failure, as node failure is equivalent to multiple links failure [10]. Node failure means intermediate node failure, source and destination node failure are omitted. Because source and sink flows cannot be restored by any protection approach. The objective of the node failure recovery is to restore transit flows that pass through the failing node.

E. PIP objective

In this paper, we focus on CPU and bandwidth as the main substrate resources. We therefore calculate the PIP revenue of each VN request n as follows.

$$\text{REV}[G_n] = P_n - \text{COST}\left[M_{\text{N}}(A_n), M_{\text{L}}(E_n)\right] \quad (1)$$

Where the first term calculates the offered VN revenue regards VN request, and the second term calculates the cost of assigned resources. We note that resources used for protection of VN primary flow do not incur a direct profit. Backup paths are considered as QoS requirements. In other words, we assume that are implicitly included in the Service Level Agreement contracted between VNs owners and PIP.

IV. VN Mapping with Combined Protection against physical node and multiple logical links failures (VNM-CNLP)

To guarantee a resilient VN provisioning, we propose an approach for VN mapping with a combined protection against a physical node and multiple logical links failures (VNM-CNLP). This approach consists on combining a cost-efficient join node and link mapping using Column Generation technique and a combined node and a multiple logical links protection mechanism based on link p-Cycle concept. We define in following in more details this approach.

A. Small-batch VN provisioning

In a realistic virtualization scenario, VN requests may not usually arrive one after another [15] in regular interval of time. Thus, a realistic VN embedding scenario could be based on a periodical approach, where VN requests are queued and then process them in small-batches in order to optimize PIP profit's over time. To do so, we discretize the provisioning time into a set of consecutive short periods. Hence, VN demand can be expressed more precisely, let T be the set of planning periods of time and $R(0)$ the initial set of VN requests. The set of VN requests $R(t)$ indexed by $t \geq 1$ is defined as:

$$R(t) = R(t-1) + R_{\text{NEW}}(t) - R_{\text{DROP}}(t) \tag{2}$$

where $R(t-1)$ is the set of accepted VN requests at the ending of period $t-1$. $R_{\text{NEW}}(t)$ is the set of new incoming VN requests and $R_{\text{DROP}}(t)$ is the set of ending VN requests at the start of period t. Where NEW and DROP are randomly selected between for example, 10% and 40%, giving us a range of cases from slowly fluctuating (10%) to fast changing (40%) of VN demand.

B. VN embedding

To overcome the complexity issue of VN embedding problem, we will use the Column Generation (CG) formulation proposed in our previous work [17]. In this modeling, we reformulate the VN embedding problem in terms of Independent Embedding Configurations (IECs), where each of which defines the embedding solution of one VN request. An IEC is defined as a set of substrate nodes and links used to handle resource requirements of the granted VN request, see Figure 1b. We represent an IEC $c \in C$ by a vector $(a_n^c)_{n \in N}$ such that $a_n^c = 1$ if the IEC c serves VN request n and 0 otherwise. We denote by COST_c the cost of an IEC c that corresponds to the costs of used bandwidth and CPU for the embedding of the VN request granted by the IEC c. It is defined according to Equation (1) as follows.

$$\text{COST}_c = \sum_{l \in L_s} b^c(l) \times c_l + \sum_{u \in W_s} p^c(u) \times c_u \tag{3}$$

Where $b^c(l)$ and $p^c(u)$ are the used substrate bandwidth of link l and CPU resources of node u by IEC c respectively. As mentioned previously, each VN request n offers a revenue for the embedding of its sets of virtual nodes and links. Accordingly, an IEC c serving VN request n generates a PIP profit's defined as:

$$\text{REV}_c = P_n - \text{COST}_c \tag{4}$$

The use of the Column Generation technique means that the VN embedding problem is decomposed into a master problem (which includes constraints related to the availability of substrate resources) and a pricing problem (which includes the constraints related to the embedding of VN resources). We call this approach Join Node and Link Embedding using CG (JNLE-CG). We present the master and pricing problem formulations as following.

1) Master problem: It corresponds to the choice of a maximum of N IECs in order to maximize the PIP profit's. To so do, we use the following Integer Linear Program $ILP(M)$ based on the decision binary variable λ_c that takes the value 1 if IEC c is used in VNs embedding solution and 0 otherwise.

a) Objective Function:

$$\max \sum_{c \in C} \text{REV}_c \, \lambda_c \tag{5}$$

b) Constraints:

$$\sum_{c \in C} \lambda_c \, b^c(l) \leq b_l; \qquad \forall l \in L_s \qquad (\alpha_l) \tag{6}$$

$$\sum_{c \in C} \lambda_c \, p^c(u) \leq p_u; \qquad \forall u \in W_s \qquad (\beta_u) \tag{7}$$

$$\sum_{c \in C} \lambda_c \, a_c^n \leq 1; \qquad \forall n \in N \qquad (\gamma_n) \tag{8}$$

$$\sum_{c \in C} \lambda_c \leq N; \qquad (\eta) \tag{9}$$

$$\lambda_c \in \{0, 1\} \tag{10}$$

Equations (6) and (7) indicate that substrate link and node loads are no more than the residual bandwidth b_l and residual CPU capacity p_u respectively. Equation (8) indicates that a maximum of one IEC can be selected for the embedding of each accepted VN request $n \in N$. Equation (9) guarantees that a maximum of N IECs can be selected for the embedding of all VN requests, as optimally all VN requests are accepted where each VN request is granted by a distinguish IEC. Equation (10) expresses the integrality of the master variable λ_c.

2) Pricing problem: It corresponds to the problem of generating an additional column (IEC) for the constraint matrix of the current master problem. We will use the same pricing model proposed in our previous work [17] except, we add the following constraint (11) in order to ensue that embedding paths for each VN request must be link-disjoint. Accordingly, we minimize the affect of a physical node failure into the VN layer and guarantee more resiliency in the VN backup phase.

$$\sum_{e \in E_n} \sum_{(u,v) \in W_s^2} \sum_{\pi \in \Pi_{uv}^e} \delta_\pi^l x_\pi^e \leq 1; \; l \in L_s \qquad (11)$$

Where $\delta_\pi^l = 1$ if embedding path π uses substrate link l and 0 otherwise. x_π^e is the main decision variable of pricing problem and it is equal to 1 if embedding path π is used for the mapping of virtual link e and 0 otherwise.

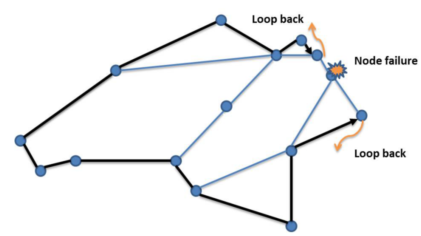
3) Solving JNLE-CG model: To solve the JNLE-CG model developed in the previous Section, we denote by $LP(M)$ the continuous relaxation of the master problem $ILP(M)$ obtained by exchanging the integrality constraint (10) by $\lambda_c \in [0,1]$ for any $c \in C$. Then the $LP(M)$ is solved using any linear programming solver (such as Cplex) until the optimal solution is reached. To check the effectiveness of this solution, we solve the pricing problem in order to identify variables with positive reduced cost. If such a variable exists, it is added to the master problem and solved again until the master problem has been solved to optimality. More details can be founded in our previous work [17].

C. VN survivability against physical node and multiple logical links failures

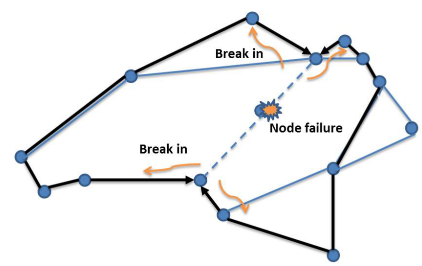
VN survivability can be seen as a generalization of the QoS guarantees that PIP commits to VN users. In this work VN survivability means VN backup plan to recover from a single physical node failure as compared to physical link failure, its impact is more severe. Indeed, it is well known [10] that a single physical node failure is logically equivalent to multiple physical link failures. Moreover, in virtualization environment, a single physical node failure can also result into multiple logical links failure as virtual links belonging to the same VN can be mapped to physical paths that share a span using the failing node.

Accordingly, our focus will be on the physical layer to provide a protection mechanism against a single physical node failure while taking into account constraints from the logical layer (virtual network layer) in order to guarantee a full VN protection against a multiple logical links failure. To do so, we proceed as follows: (*a*) VN mapping is done in a such way that virtual links belonging to the same VN are mapped to link-disjoint physical paths (see constraint (11)), and (*b*) input of VN protection phase will be the mapping topology (set of selected embedding paths) for each VN request instead of a set of independent VN embedding paths. Doing so, we will be able to constraint the VN backup in such a way that embedding paths used for embedding of virtual links belonging to the same VN should be node-disjoint in order to guarantee a node failure independent path protection scheme. We present in following Sections the concept of link p-Cycle and our three proposed link p-Cycle based VN protection techniques.

1) Link p-Cycle concept: A link p-Cycle is proactive protection scheme with pre-configured backup paths. Due to its high capacity efficiency and fast recovery speed, link p-Cycle, is regarded as a very promising approach, that can provide strong protection for node failure. A link p-Cycle provides two protection paths for each span/link that straddles the cycle along with one path protection for "on-cycle" nodes/spans, see Figure 2. Accordingly, straddling spans can have working bandwidth capacity but no spare capacity, which is a very unique propriety of link p-Cycle based protection approach [3]. We present in following our proposed protection techniques against a single node failure and a multiple logical links failure.



(a) protection "on-cycle" failure

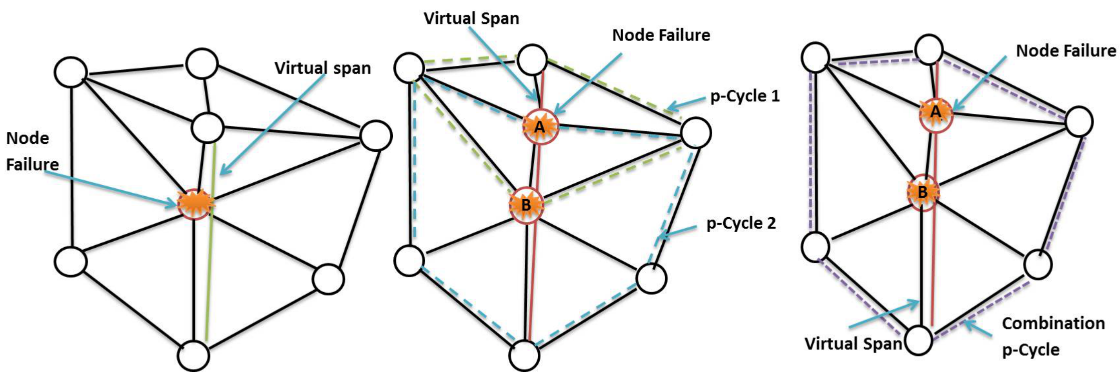


(b) protection a "straddling" failure

Figure 2: Link p-Cycle protection technique

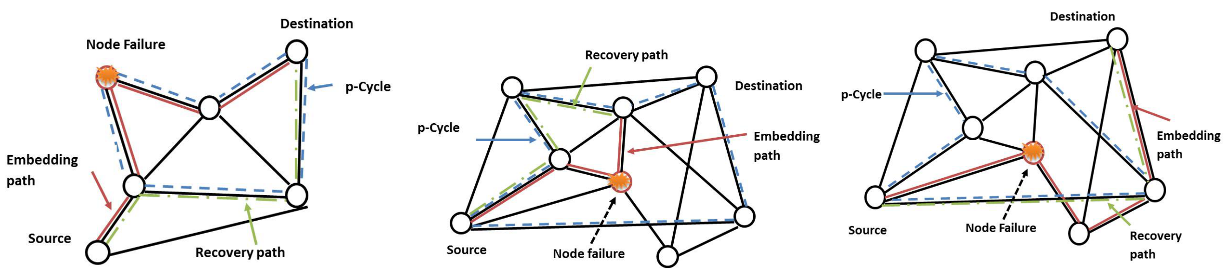
2) $p-$Cycle-based virtual span protection technique (p-Cycle-VSP): p-Cycle-VSP technique consists on the transfer of a node failure protection into a span failure protection, Figure 3a shows an example. To do so, we apply the two following phases.

a) Phase 1: For each substrate node used in a VN embedding path, we assume that there exists a *virtual span* that crosses this node from one of adjacent nodes



(a) Virtual span (b) Two virtual spans (c) Combination of two virtual spans

Figure 3: p-Cycle-based virtual span protection



(a) Partially on-span (b) Partially on-span and straddling (c) Partially straddling

Figure 4: p-Cycle-based span-and-path protection candidates

to another. In fact, a virtual span covers an intermediate node and two of its adjacent links belonging to this embedding path. In another word, we protect the VN embedding paths against node failure by finding the equivalent protecting set of virtual spans. Next, we can easily find candidate p-Cycles for protecting these "virtual span" by using Hongbo's algorithm [18].

b) Phase 2: We formulate the p-Cycle-VSP protection technique as an ILP defined as follows.

Parameters:

- S and P are the set of virtual spans and candidates protection p-Cycles respectively.
- T_N: set of all embedding topologies generated for N VN virtual networks. We note that an embedding topology of a VN n corresponds to the set of assigned embedding paths.
- S_n: set of virtual spans generated for the embedding topology calculated for the VN n in embedding phase.
- b_s: bandwidth requirement of span s.
- $\delta_s^u = 1$, if physical node u is an intermediate node of span s and 0 otherwise.
- $\alpha_p^l = 1$, if p-Cycle p uses link l and 0 otherwise.
- a_s^p represents the number of backup paths that a p-Cycle p offers to span s. Thus, $a_s^p = 2$ only if end-nodes of span s are in p-Cycle p (Figure 3c), $a_s^p = 1$ if end-nodes of span s are in p-Cycle p and at least one link uses cycle p (Figure 3b), and $a_s^p = 0$ otherwise.

Decision Variables:

- y^p: amount of bandwidth used on p-Cycle p for protection of VN embedding paths.
- y_s^p: amount of bandwidth of p-Cycle p used to protect virtual span s.
- $z_s^p = 1$ if span s is protected by p-Cycle p and 0 Otherwise.

Objective function

$$\min \sum_{p\in P} y^p \sum_{l\in L_s} c_l \alpha_p^l \tag{12}$$

Constraints

$$y_s^p \le y^p \ ; p \in P, s \in S \tag{13}$$

$$y_s^p \le M z_s^p \ ; p \in P, s \in S \tag{14}$$

$$b_s \le \sum_{p\in P} a_s^p y_s^p \ ; \ s \in S. \tag{15}$$

$$\sum_{s\in S_n} z_s^p \delta_s^u \le 1 \ ; \ p \in P. \tag{16}$$

$$L(p,s) \leq H^{c(s)}\ ;\ s \in S, p \in P \tag{17}$$

$$y_s^p, y^p \in Z_+, z_s^p \in \{0,1\}\,;\, p \in P, s \in S_n. \tag{18}$$

Where $L(p,s)$ is expressed as the sum of number of hops of backup and embedding paths minus the number of hops of used span. M is a large number at least equal to the largest QoS class bandwidth requirement.

The Objective function (12) minimizes the cost of p-Cycles used for VNs protection. Equation (13) and (14) express the relationship between decisions variables z_s^p, y^p and y_s^p. Equation (15) ensures that enough bandwidth should be provided by a p-Cycle that protect a virtual span. Equations (16) are used to protect VN embedding against a multiple logical links failure. Thus, spans relative to the same VN embedding topology sharing any protection p-Cycle should be node-disjoint. Constraint (17) expresses QoS requirement in terms of number of switching nodes. (18) is the variables domain constraint.

We also propose an improved version of p-Cycle-VSP technique, that combines multiple virtual spans together into a new virtual span as shown in Figures 3b and 3c. Then, we find a p-Cycle to protect this new virtual span instead of previously multiple virtual spans. Doing so, we increase the backup sharing efficiency. We call this improved approach p-Cycle-CVSP for combined virtual span based p-Cycle protection. We use the same ILP model as in p-Cycle-VSP approach except we modify constraint (15) as follows.

$$\max_{s \in S_u} b_s \leq \sum_{p \in P} a_s^p y_s^p\ ;\ u \in W_s. \tag{19}$$

where S_u denotes the span set that crosses node u. Constraint (19) guarantees that enough bandwidth is provided by protecting p-Cycles for the span with the largest bandwidth requirement in S_u set denoted by $\max_{s \in S_u} b_s$.

3) p-Cycle-based span-and-path protection technique (p-Cycle-SPP): This approach consists on enlarging the candidate protecting p-cycles set compared to those used in p-Cycle-VSP technique. To do so, we apply the two following phases.

a) Phase 1: Compared to p-Cycle-CVSP and p-Cycle-VSP techniques, the range of candidate p-Cycles is extended. The new range includes p-Cycles that have one of the following relationships with its respective protected embedding path against a given node failure.

1) Embedding path is partially on-span with used p-Cycle (Figure 4a).
2) Embedding path is partially on-span and on straddling with used p-Cycle (Figure 4b).
3) Embedding path is partially on straddling with p-Cycle (Figure 4c).

For the first two relationships, it's easy to find eligible p-Cycles by using Hongbo's algorithm [18]. For the third relationship, we will use Onguetou's [7] concept which consists on the following:
"The protecting p-cycle must intercept the affected flow in at least two nodes, that is, one node upstream and one node downstream of the node failure."

b) Phase 2: We formulate the p-Cycle-SPP protection technique as an ILP defined as follows.
Parameters:

- P: set of candidates protection p-Cycles.
- b_π: bandwidth requirement of embedding path π.
- a_π^p represents the number of backup paths that a p-Cycle p offers. Thus, $a_\pi^p = 2$ if only end-nodes of path π are in p-Cycle p, $a_\pi^p = 1$ if end-nodes of path π are in p-Cycle p and at least one link of path π uses p-Cycle p, and $a_\pi^p = 0$ otherwise.

Decision Variables:

- y^p: amount of bandwidth used on p-Cycle p for protection of VN embedding paths.
- $y_{u\pi}^p$: amount of bandwidth used on p-Cycle p to protect embedding path π, when node u fails.
- $z_{u\pi}^p = 1$ if embedding path π is protected by p-Cycle p against node failure u and 0 otherwise.

Objective function

$$\min \sum_{p \in P} y^p \sum_{l \in L_s} c_l \alpha_p^l \tag{20}$$

Constraints

$$\sum_{n \in N} \sum_{\pi \in T_n} y_{u\pi}^p \leq y^p\ ; p \in P, u \in W_s \tag{21}$$

$$y_{u\pi}^p \leq M z_{u\pi}^p\ ; p \in P, u \in W_s, \pi \in T_n, n \in N \tag{22}$$

$$b_\pi \leq \sum_{p \in P} a_\pi^p y_{u\pi}^p;\ u \in W_s, \pi \in T_n, n \in N. \tag{23}$$

$$\sum_{\pi \in T_n} z_{u\pi}^p \leq 1;\ p \in P, u \in W_s. \tag{24}$$

$$L(p,u,\pi) \leq H^{c(\pi)}\ ; u \in W_s, \pi \in T_n, p \in P \tag{25}$$

$$y_{u\pi}^p, y^p \in Z_+, z_{u\pi}^p \in \{0,1\}\,;\, u \in W_s, \pi \in T_n, p \in P. \tag{26}$$

Where $L(p,u,\pi)$ is expressed as the sum of number of hops of backup and embedding paths minus the number of hops of used p-Cycle.

The objective function (20) minimizes the cost of p-Cycles used for VNs protection. Constraints (21) and (22) express linking among variables, $z_{u\pi}^p$, $y_{u\pi}^p$, and y^p. Constraints (23) ensure that enough bandwidth should be provided by p-Cycles that protect embedding paths. Constraints (24) are used to protect VN embedding against a multiple logical links failure. Thus, backup paths of the same VN embedding topology sharing any protection p-Cycle should be node-disjoint. Constraint (25) expresses QoS requirement in terms of number of switching nodes. (26) is the variables domain constraint.

V. Numerical results

A. Simulation setup

To assess the efficiency of the proposed periodical VNM-CNLP approach, we carried out experimental assessments, on a EU metro backbone network that spans the 20 largest metropolitan areas in Europe, connected by 44 bidirectional links [12]. Since VNs are still not well-known in the industry, we generated synthetic VN topologies using our generator inspired from GT-ITM tool proposed in [19]. QoS requirements of new VN requests are randomly determined by a uniform distribution among $|Q_1| = 5$ QoS classes for VN nodes and among $|Q_2| = 5$ QoS classes for VN links. Unit costs are expressed in terms of $\$X$, which represents the price of 1 Mb of bandwidth or 1 unit of CPU.

B. Simulation scenarios

To evaluate the performance of our VNM-CNLP approach, we will evaluate the following scenarios:

1) VNM-CNLP-1: uses JNLE-CG approach for VN mapping and p-Cycle-VSP for VN protection.
2) VNM-CNLP-2: uses JNLE-CG approach for VN mapping and p-Cycle-CVSP for VN protection.
3) VNM-CNLP-3: uses JNLE-CG approach for VN mapping and p-Cycle-SPP for VN protection.

C. Benchmark

Since our main contribution in this work is the proposition of VN protection techniques, we propose as benchmark VNM-NPEC approach that uses: *(a)* a VN mapping based on our JNLE-CG approach, and *(b)* a VN protection technique based on the node encircling p-Cycle (NPEC) proposed in [8] that consists on finding a set of particular p-Cycles for each substrate failing node. A NPEC is defined for the protection of transiting flows through one specific node by containing all the immediate neighbor-nodes of the protected node but not the protected node itself. In case of there is no NPEC, a path that goes through a maximum number of neighbor-nodes is used for protection.

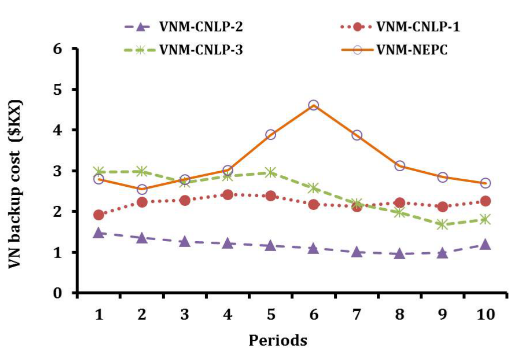


Figure 5: VN backup cost

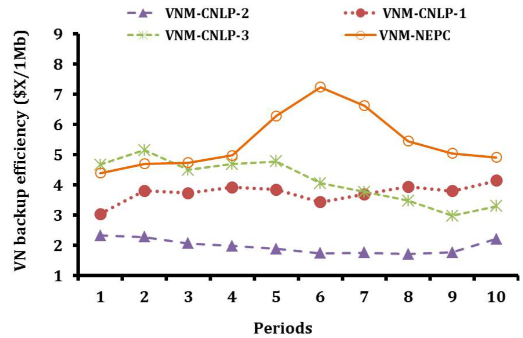


Figure 6: VN backup efficiency

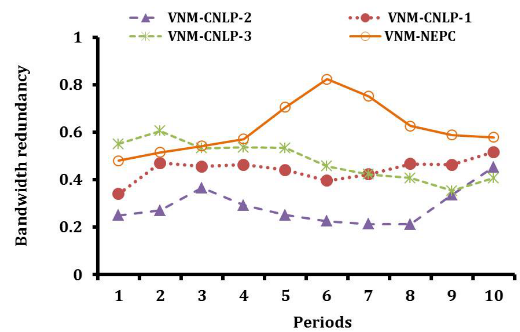


Figure 7: Bandwidth capacity redundancy

D. Performance evaluation metrics

To evaluate the performance of our VNM-CNLP approach, we are measuring the following metrics.

1) VN backup cost: measured as the cost of the used bandwidth for VN protection.

2) VN backup efficiency: measured as the ratio of VN protection cost to the amount of protected bandwidth used by VN embedding paths.

3) Bandwidth capacity redundancy: measured as the ratio of bandwidth used for VN protection to the bandwidth used by VN embedding paths.

E. Evaluation results

Our focus in this Section, is on quantifying the advantage of using link p-Cycle protection mechanism against physical node failure in terms of VN protection cost, bandwidth redundancy and VN protection efficiency.

1) Based link p-Cycle protection approaches vs. benchmark: Figures 5 plots the backup cost of protected bandwidth used by embedding paths vs. the allocation time periods. This Figure shows that link p-Cycle based protection approaches provide the lowest VN backup cost over a large mix of VN requests. We observe the same tendency, in terms of VN backup efficiency, i.e., cost of protecting 1 unit of bandwidth used by VN embedding paths, and bandwidth capacity redundancy as shown in Figures 6 and 7 respectively. The main reason behind this is that link p-Cycle based VN backup approaches compared to NEPC model use protection bandwidth resource with lower cost and maximize the backup bandwidth sharing among protected VN embedding paths. Indeed, averaging over the backup efficiency obtained over 10 periods shown in Figure 6, we conclude that benchmark NPEC uses on average 5.48 unit cost to protect 1 unit of protected bandwidth; while VNM-CNLP-1, VNM-CNLP-2 and VNM-CNLP-3 use on average 3.84, 1.95 and 3.68 unit cost respectively. Accordingly, these results confirm our expectation that protection against a single node failure using node encircling p-Cycle may result in high VN protection cost as it is unable to maximize the sharing of used VN backup resources. For example, in period 6 we can note that NEPC provides the worst results in terms of backup cost, backup efficiency and bandwidth capacity redundancy, this is related to the fact that NPEC setups almost a distinguish backup for each VN request. As shown in Figure 7, the capacity redundancy at period 6 is equal to 0.84, so there is roughly no resources sharing among VNs backups.

2) Why proposing three link p-Cycle protection techniques: As aforementioned, the three based link p-Cycle protection techniques provide on average better results than benchmark NPEC. However, each of them has its own advantages and shortcomings that can be summarized as following.

- p-Cycle-VSP has the lowest computation time, as it uses the lowest number of span p-Cycle candidates. However, as it provides protection by span, an embedding path can be protected by more than one p-Cycle. Accordingly, this technique has the largest failure recovery time.
- p-Cycle-CVSP has the lowest backup cost, the best efficiency and the lowest redundancy ratios. However it has the largest computation time, as it uses the largest number of span protection p-cycles. Moreover, the failure recovery time is comparable to p-Cycle-VSP but worst than p-Cycle-SPP.
- p-Cycle-SPP has the lowest failure recovery time, as it proposes a protection p-Cycle per embedding path. However, it provides the worst results in terms of backup cost, backup efficiency compared to span p-Cycle based techniques. This is related mainly to the fact that it uses a lower VN protection resources sharing ratio.

VI. Conclusion

We proposed a survivable VN embedding approach that provides a protection mechanism for the physical layer while taking into account constraints from the virtual network layer in order to guarantee a full VN protection against a single physical node failure and multiple logical links failures. We quantified the advantage of using link p-Cycle protection mechanism against physical node failure in terms of VN protection cost, bandwidth redundancy and VN protection efficiency. Experiments on large mix of VN requests confirms the superiority of our protection techniques over benchmark.

References

[1] S. Baucke *et al.*, *Virtualization Approach: Concept, 4WARD project*, Deliverable 3.1.0, 2009.

[2] M. Chowdhury, R. Boutaba, *A Survey of Network Virtualization*. In Computer Networks (COMNET). Elsevier, Vol. 54(5), pp. 862-876, April 2010.

[3] M. Kiaei, C. Assi and B. Jaumard, *A Survey on the p-Cycle Protection Method*, IEEE Communications Survey & Tutorials, Vol. 11, No. 03, pp.53-70, 2009.

[4] M. Rahman, and I. Aib, and R. Boutaba, *Survivable Virtual Network Embedding*, Networking, LNCS 6091, pp. 40-52, 2010.

[5] Y. Chen, J. Li, T. Wo and W. Liu, *Resilient Virtual Network Service Provision in Network Virtualization Environment*, IEEE International Conference on Parallel and distributed Systems, Shanghai, China, 8-10 December, 2010.

[6] T. Guo, N. Wang, K. Moessner and R. Tafazolli, *Shared Backup Network Provision for Virtual Network Embedding*, IEEE ICC, kyoto, Japon, 5-9 June 2011.

[7] D. P. Onguetou and W. D. Grover, *A new insight and approach to node failure protection with ordinary p-cycles*, in Proc. ICC '08, Beijing, China, 2008.

[8] J.E. Doucette, P.A. Giese, and W.D. Grover, *Combined Node and Span Protection Strategies with Node-Encircling p-Cycles*, in Proc. DRCN 2005, Ischia (Naples), Italy, pp. 213-221, 16-19 October 2005.

[9] A. Frikha, B. Cousin and S. Lahoud, *Extending Node Protection Concept of P-Cycles for an Efficient Resource Utilization in Multicast Traffic*, in Proc. LCN, Bonn, Germany, 4-7 Oct. 2011.

[10] W. Grover and D. Stamatelakis, *Cycle-oriented distributed preconfiguration: Ring-like speed with mesh-like capacity for self-planning network restoration*, in Proc. IEEE ICC'98, Atlanta, Georgia, USA, June 1998.

[11] Y. Fan and M. Ammar, *Dynamic topology configuration in services overlay networks- A study of reconfiguration policies*, in Proc. IEEE INFOCOM, pp. 1-12, 2006.

[12] J. Lu, and J. Turner, *Efficient mapping of virtual networks onto a shared substrate*, DCSE department, Washington University in St Louis, Technical Report, Vol.35, pp.1-11, 2006.

[13] Y. Zhu and M. Ammar, *Algorithms for assigning substrate network resources to virtual network components*, IEEE INFOCOM, Barcelona, Spain, April 23-29, 2006.

[14] M. Yu, *et al.*, *Rethinking Virtual Network Embedding: Substrate Support for path Splitting and Migration*, ACM SIGCOMM Comp. Communication, Vol. 38, Issue 2, pp. 17-29, 2008.

[15] N. Chowdhury, M. Rahman, and R. Boutaba, *Virtual Network Embedding with Coordinated Node and Link Mapping*. In Proc. of the 28th IEEE Conference on Computer Communications (INFOCOM). IEEE Press, pp. 783-791. Rio de Janeiro, Brazil, April 19-25, 2009.

[16] D. Anderson, *Theoretical approaches to node assignment*, Unpublished Manuscript, http://www.cs.cmu.edu/ dga/papers/ andersen-assign.ps, 2002.

[17] A. Jarray, and A. Karmouch, *Column Generation Approach for One-shot Virtual Network Embedding*, IEEE GlobeCom Workshop MENS, Anaheim, USA, 3-7 December 2012.

[18] H. Liu and J. Wang, *A new way to enumerate cycles in graph*, AICT/ICIWT, Guadeloupe, French Caribbean, 19-25 February, 2006.

[19] E. Zegura, K. Calvert, and B. Bhattacharjee, *How to model an inter-network* in Proc. IEEE INFOCOM, pp. 594-602, 1996.