Self-managed Resources in Network Virtualisation Environments

Rashid Mijumbi, Joan Serrat and Juan-Luis Gorricho
Universitat Politècnica de Catalunya, 08034 Barcelona, Spain
rashid@tsc.upc.edu, serrat@tsc.upc.edu and juanluis@entel.upc.edu

Abstract—**Network virtualisation is a promising technique for dealing with the resistance of the Internet to architectural changes. This is achieved by enabling a novel business model in which infrastructure management is decoupled from service provision. One of the main challenges in network virtualisation is efficient sharing of physical network resources by the different virtual networks. This work contributes to efficient resource sharing in network virtualisation by dividing the resource management problem into three sub-problems: virtual network embedding (VNE), dynamic resource allocation (DRA), and virtual network survivability (VNS); and then proposing a solution for each one of them. Specifically, we propose a path generation-based approach for VNE, machine learning-based self-management approaches for DRA, and a multi-entity negotiation algorithm for VNS. Through simulations, all our proposals are compared with related approaches, showing improvements in resource utilisation efficiency, which would directly result into better profitability for physical resource owners.**

Keywords*—*Future Internet, network virtualisation, virtual network embedding, dynamic resource allocation, network survivability, resource management.

I. INTRODUCTION

The evolution of the Internet in the last decades has led to a shift from its conception as a mere connectivity network to a content based network [1]. Along with this evolution, Internet users are now concerned not only with being able to communicate, but also getting the right information at the right time and at an affordable price. The expectations and demands of Internet users have risen to levels that are very difficult to achieve with the traditional architectural approach of the same connectivity infrastructure for any type of service offer. Therefore, a specialisation of resources and protocol stacks is a must for such diversified service provisioning scenarios. This requires modifications in the current "one size fits all" architecture of the Internet [2]. However, the existence of multiple stakeholders with competing objectives makes it very difficult, if not impossible, for any architectural changes to be made on the Internet. This is the so called ossification of the Internet [3], and can be observed, for example, from the difficulties that have been encountered in the deployment of IP Multicast [4] and IPv6 [5].

Network virtualisation - which is now a subject of various research teams both in academia as well as industry [6] - has been proposed as a Future Internet enabler technique to not only allow for the de-ossification of the current Internet [7] but also to facilitate new and specialised service deployment [3]. In a network virtualisation environment (NVE), the traditional role of Internet service providers (ISPs) is split into two: infrastructure providers (InPs) and service providers (SPs) [2]. The InPs concentrate on deployment and management of physical networks known as substrate networks (SNs), while SPs lease resources from InPs to create virtual networks (VNs) which are used to provide services to end-users. Multiple VNs can share the same SN and indeed a given VN can be hosted by multiple SNs. Since the profitability of InPs depends on how many VNs are able to be allocated simultaneously onto the SN, the success of network virtualisation will depend, in part, on how efficiently VNs utilise SN resources.

This dissertation contributes to efficient resource management in NVEs by splitting the problem into three sub-problems: VNE, DRA, and VNS. VNE involves the mapping of all nodes and links in a given VN to nodes and links respectively of a SN following a set of constraints [8]. The constrained VNE problem is NP-Hard [9]. As a result, to simplify the solution, many existing approaches either solve the problem in two steps or propose heuristics that make assumptions (e.g. a SN with infinite resources), some of which would not apply in practical environments. This dissertation proposes an improvement in VNE by proposing a one-shot (both node and link mapping performed in one step) VNE algorithm which is based on path generation (PG). The PG approach starts by solving a restricted version of the problem, and thereafter refines it to obtain a final solution. The objective of a one-shot mapping is to achieve better resource utilisation, while using PG significantly enhances the solution time complexity.

In addition, current approaches are static in the sense that after the VNE stage, the resources allocated are not altered for the entire lifetime of the VN. The few proposals that do allow for adjustments in original mappings allocate a fixed amount of node and link resources to VNs throughout their life time. Since network load varies with time due to changing user demands, allocating a fixed amount of resources based on peak load could lead to an inefficient utilisation of overall SN resources, whereby, during periods when some virtual nodes and/or links are lightly loaded, SN resources are still reserved for them, while possibly rejecting new VN requests. The second contribution of this dissertation are a set of self-management algorithms in which the SNs use time-difference learning techniques that perform dynamic resource allocations which ensure that resources are efficiently utilised, while at the same time making sure that the QoS requirements of VNs are not violated.

Finally, while some research has already studied multi-domain VNE, the available approaches to survivable VNs have focused on the single InP environment. Since in the more practical situation a NVE will involve multiple InPs, and because an extension of network survivability approaches from the single to multi domain environments is not trivial [10], this dissertation proposes a distributed and dynamic approach to survivability in multi-domain NVEs. This is achieved by using a multi-agent-system that uses a multi-attribute negotiation protocol and a dynamic pricing model which allow InPs to form coalitions to support resource backups. The ultimate objective is to ensure that InPs maximise profitability by minimising penalties resulting from QoS violations.

Overview of Research Objectives and Contributions

Our focus is the development of algorithms for management of resources in NVEs. The primary objectives of the developed algorithms −beyond the obvious goals of utilisation efficiency, and autonomic allocations of physical resources− are: (1) to minimise the time complexity of carrying out VNE in one-shot; (2) to dynamically adjust resources allocated to VNs according to perceived needs; (3) to minimise QoS violations resulting from failures in SN resources. It is our humble opinion that achieving these objectives, either fully (as an orchestrated solution) or in part (each of them independently) would constitute a significant contribution to the very important problem of resource management for the Future Internet, and specifically in NVEs. In this context, the main contributions of this dissertation are as follows:

- A path generation-based approach that significantly improves the time complexity of the one-shot VNE compared to an optimal formulation [11], [12], [13], [14], [15].

- A set of distributed learning-based self-management algorithms that allocate resources to virtual nodes and links dynamically, leading to better substrate resource utilisation [8], [11], [13], [14], [15], [16], [17], [18].

- A negotiation protocol that ensures VNS with minimum communication message overhead and a dynamic resource pricing model that ensures efficient utilisation of resources [11], [13], [14], [19].

To the best of our knowledge, the work in this dissertation is the first application of path/column generation to VNE. It is also a novel contribution of this dissertation to apply machine learning techniques to DRA in NVEs [20]. Finally, our automated negotiation and pricing proposal is the first foray into network survivability for multi-domain VNs.

The rest of this extended abstract is organised as follows: Sections II, III and IV summarise the technical solutions for the three sub-problems addressed in the dissertation, namely, VNE, DRA and VNS, as well as the results obtained from our proposals respectively. The paper is concluded in Section V.

II. VIRTUAL NETWORK EMBEDDING

VNE involves mapping[1] of VNs onto a SN and is initiated by a SP specifying resource requirements for both nodes and links to the InP. The specification of VN resource requirements is usually represented by a weighted undirected graph denoted by $G_v = (N_v, L_v)$, where N_v and L_v represent the sets of virtual nodes and links respectively. Similarly, a SN can be modelled as an undirected graph denoted by $G_s = (N_s, L_s)$, where N_s and L_s represent the sets of substrate nodes and links, respectively. Nodes and links from the SN and VNs have properties such as CPU, bandwidth, delay, queue size, e.t.c [8].

The VNE problem involves the mapping of each virtual node to one of the possible (based on resource constraints of both virtual and substrate nodes) substrate nodes. For a successful VNE, each virtual node must be mapped and any given substrate node can map at most one virtual node from the same request. Similarly, all the virtual links have to be mapped to one or more substrate links connecting the nodes to which the virtual nodes at its ends have been mapped without violating resource demand/availability constraints [8].

The VNE problem, with fixed constraints on virtual nodes and links, reduces to the multi-way separator problem which is known to be NP-Hard [9], [21]. Even when all nodes have already been mapped, the problem of performing link mapping for unsplittable flows [22] is still NP-Hard [23]. Therefore, most approaches to the VNE problem have mainly been through splitting the problem into two stages (node mapping followed by link mapping), and then using heuristics for each of the two stages. Performing the VNE in two separate steps can lead to blocking or rejecting of resource requests at the link mapping stage and hence a sub-optimal substrate resource utilisation. Even when the two embedding steps are coordinated [9], the embeddings are still sub-optimal. If the embedding is performed in one step, the embedding efficiency can be significantly improved, but the computation time is so high.

A. Path Generation-based VNE

The VNE problem can usually be represented as a mathematical optimisation problem [9]. However, for one-shot unsplittable flow VNE, even medium sized problems are intractable. This dissertation proposes a PG-based heuristic to improve this time complexity while still ensuring a high quality embedding solution. PG is a method that solves mathematical programs with a large number of variables efficiently. The main idea is to solve a restricted version of the program (the restricted primal problem [24]) - which contains only a subset of the variables, and then (through the use of the dual problem[24]) add more variables as needed [25]. Usually, path generation involves creating an *initial solution* (*restricted* set of variables) which are used in the solution for the *restricted* primal problem. Then, solving pricing problems (which are determined from the dual problem), allows for adding more variables to improve the initial solution, until either a final optimal solution is found, or a stopping condition is reached.

In this dissertation, we formulated the one-shot unsplittable flow VNE problem as a mathematical program, called the primal problem. Then, by using duality theory [26], [24], we derived the corresponding dual problem. The duality theorem states that the objective function value of the dual at any feasible solution is always greater than or equal to the objective function value of the primal at any feasible solution [27]. The fundamental idea behind duality is that every feasible solution for the primal problem gives a bound on the optimal value of the objective function of the corresponding dual problem

---

[1] This paper, like other related works, uses the terms mapping and embedding synonymously.

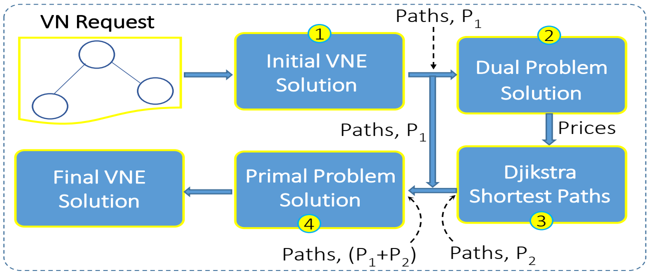
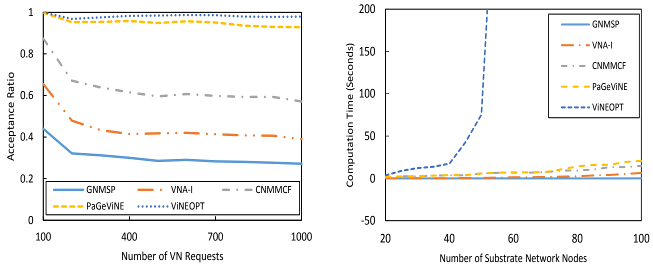


Fig. 1. Path Generation-based Virtual Network Embedding



(a) Acceptance Ratio (b) Computation Time

Fig. 2. Evaluation of Path Generation-based VNE

[28]. In our proposal, the dual problem was formulated by constructing the dual functions as by-products [11]. Since PG requires an initial solution, we also proposed a two-step node and link mapping approach for determining the initial solution. This is performed for each virtual link and involves first creating an augmented SN [9], and then through sequencial node and link mapping, creating substrate paths from one end of the link to the other. This consititutes a feasible solution for the VNE problem, which while requires less computation effort, is not efficient for resource utilisation.

The proposed approach can be summarised by the four steps shown in Fig. 1: We start by creating an initial set of paths (P_1) using a two stage node and link mapping. We then use these paths to solve a dual problem, and use the pricing problems (shortest path problems) to determine a set of paths (P_2) to add to the initial solution i.e. paths that can improve the initial solution. These paths are then used to solve a restricted primal problem to obtain a final solution. It can be noted that our proposal avoids the usual iteration required in a path generation approach where the primal and dual problems are solved sequentially, many times, instead preferring only to perform a single iteration.

B. *Results*

We implemented a discrete event simulator in Java, using Brite [29] to generate SN and VN topologies, and ILOG CPLEX 12.4 [30] to solve mathematical programs. We compared the performance of our proposal (PaGeViNE) with three representative solutions from the state-of-the-art. The first, GNMSP [31], performs node and link mapping separately; the second, CNMMCF [9], coordinates the two steps; and the third, VNA-1 [32], performs a one shot mapping. We also formulated and implemented, ViNE-OPT [11], a baseline formulation of the optimal one shot mapping. Evaluations included acceptance ratio and computation time [11].

From Fig. 2(a), we see that PaGeViNE achieves an average acceptance ratio about 94% of that obtained by the optimal solution ViNE-OPT while outperforming state-of-the-art solutions by atleast 34% in terms of average acceptance ratio. The fact that CNMMCF is underperforming PaGeViNE with respect to the average acceptance ratio and resource utilisation can be attributed to the fact that CNMMCF is using more resources at the link mapping stage since it performs node and link mappings separately. For VNA-1, while the node and link mapping is done in one shot, they are carried out sequentially, considering specific clusters of the SN each time. It is therefore expected that the results would not be as good as those achieved by a global solution based on mathematical programming. With respect to time complexity, the graphs in Fig. 2(b) show that the running times of GNMSP and VNA-1 are comparatively lower than those of PaGeViNE. Once again, this can be explained by the fact that these two solutions do not solve a mathematical program as PaGeViNE does. We also note that the computation time of PaGeViNE is slightly higher (only by about 2.5%) than that of CNMMCF. This can be attributed to the fact that PaGeViNE solves three mathematical programs [11], while CNMMCF solves only two. Moreover, it is expected that solving the problem in one shot requires more computation than solving it in two stages, since some of the mathematical programs solved in PaGeViNE are binary. With regard to ViNE-OPT we see that the computation time quickly grows exponentially. In fact, for 50 SN nodes, the computation time of ViNE-OPT is 1300% higher than that of PaGeViNE, while for 60 SN nodes or more, ViNE-OPT could not find a VNE solution in 1 hour.

III. DYNAMIC RESOURCE ALLOCATION (DRA)

The VNE approach proposed in Section II (or indeed in most VNE approaches [20]) is static in that it allocates a fixed amount of SN resources to any given VN for the entire duration of the VN. However, as the resources allocated to nodes and links are meant for use by end users, and because Internet traffic is not uniform, reserving a fixed amount of resources for virtual nodes and links throughout their lifetime could lead to inefficient resource utilisation and hence limit the revenue of InPs, especially if other VN requests are rejected while reserving resources for VNs that are lightly loaded. Therefore, DRA as proposed in this dissertation involves monitoring the actual usage of resources allocated to VNs, and making opportunistic use of these resources based on perceived need for them. The opportunistic use of resources involves carefully taking advantage of unused virtual node and link resources to ensure that VN requests are not rejected when resources reserved to already embedded requests are idle. However, this is performed carefully to ensure that quality of service parameters such as packet drop ratio and delay for the VNs are not affected.

A. *Learning-based Dynamic Resource Allocation*

We represent the SN as a multiagent system in which each substrate node and link is represented by a node agent $n_a \in \mathcal{N}_a$ and a link agent $l_a \in \mathcal{L}_a$, where $\mathcal{N}_a$ and $\mathcal{L}_a$ are the sets of node agents and link agents respectively. The node agents manage node resources such as queue size while the link

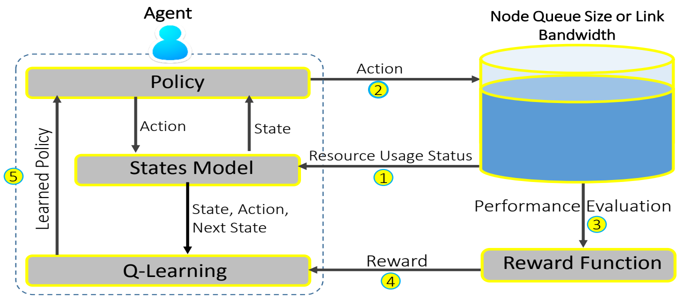
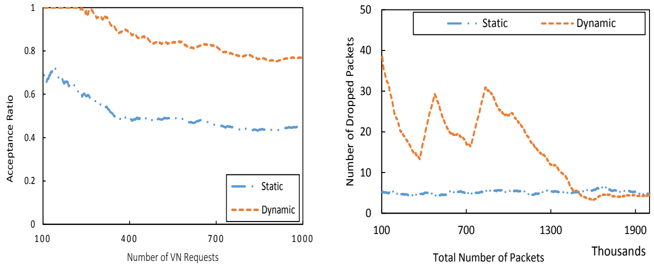


Fig. 3. Learning System Modelling: Case of a single substrate node/link



(a) Acceptance Ratio (b) Node Packet Drop Ratio Variation

Fig. 4. Evaluation of RL-based Dynamic Resource Allocation in NVEs

agents manage link resources such as bandwidth. The agents dynamically adjust the resources allocated to virtual nodes and links, ensuring that while enough resources are always available to serve user requests, they are not left under utilised.

As can be noted from Fig. 3, the proposed self-management system is made up of five steps. The agent starts by getting a resource usage status. For any given substrate node/link x and virtual node/link y, the resource status is a 3-tuple $RS^y_x = (R_y,\ R^v_x,\ R^s_x)$, where R_y is the percentage of total resource demand of y that is allocated to it (e.g. a node with demand 50 units may be allocated 25units at a given point, giving $R_y = 0.5\ or\ 50\%$), R^y_x is the percentage of R_y which is un utilised by y, and R^y_x is the percentage of total resources of x that are unused. These continuous variables are then discretised by the states model by matching them against a predefined set $S = \{S_0 = (000, 000, 000), S_1 = (000, 000, 001), ..., S_{510} = (111, 111, 110), S_{511} = (111, 111, 111)\}$ of 512 possible discrete states. It can be noted that these states result from discretising the variables R_y, R^v_x and R^s_x into 8 levels. The next step is for the agent to take an action. We define a set $A = \{A_0 = -50.0\%, A_1 = -37.5\%, ..., A_7 = +37.5\%, A_8 = +50.0\%, \}$ of 9 possible actions. Each action A_n represents a net percentage change in the resources allocated to a virtual node/link with respect to its total demand. In order for an agent to chose which action to take, it has a policy. The policy is implemented by means of a lookup table which, for each state, maintains an updated evaluation of all the possible actions. Given the evaluation of all the possible actions of a given state, the agent uses the softmax action selection criterion [33], i.e. it takes a random action A_n while in state S_m with a probability $\mathcal{P}(A_n|S_m)$ as defined in equation (1).

$$\mathcal{P}(A_n|S_m) = \frac{\exp\{Q(A_n|S_m))/\tau\}}{\sum\limits_{\hat{A_n} \neq A_n} \exp\{Q(S_m, \hat{A_n})/\tau\}} \tag{1}$$

where $Q(s, a)$ is an evaluate of a given action a while in state s, and τ is a positive parameter called the temperature. High temperatures cause the actions to be almost equiprobable. The SNs and VNs are then monitored to determine a reward $r(y)$ for each virtual node/link y. This is dependent on the percentage resource allocation R_y, the percentage resource utilisation R_u,, link delay D_{ij} in case of $l_a \in \mathcal{L}_a$ and the the number of dropped packets P_i in the case of $n_a \in \mathcal{N}_a$. We then use the reward function $r(y)$ to determine a reward which is fedback to the agent.

$$r(y) = \begin{cases} -100 & \text{if } R_y \leq 0.25 \\ \nu R_u - \Big(\kappa D_{ij} + \eta P_i\Big) & otherwise \end{cases}$$

where ν, κ and η are constants aimed at adjusting the influence of the variables R_u, D_{ij} and P_i to the overall reward. The objective of the reward function $r(y)$ is to encourage high virtual resource utilisation while punishing $n_a \in \mathcal{N}_a$ for dropping packets and $l_a \in \mathcal{L}_a$ for having a high delay. We also assign a punitive reward of -100 to resource allocations below 25% to ensure that this is the minimum allocation to a virtual resource and therefore avoid adverse effects to QoS in cases of fast changes from very low to high VN loading.

Finally, the agent uses the provided reward and the Q-learning equation (2) to adjust the evaluation/value of its previous action. This process continues until the agent has learnt optimal actions for all possible states.

$$Q(s_p, a_p) \leftarrow (1 - \alpha)Q(s_p, a_p) + \alpha \left\{ r_p + \lambda \max_{a \in \mathcal{A}} Q(s_n, a) \right\} \tag{2}$$

where $Q(s_p, a_p)$ is the new value of state s_p corresponding to action a_p, r_p is the reward obtained from taking the action a_p while in state s_p and s_n is the next state resulting from taking the action a_p while in state s_p, implying that $Q(s_n, a)$ is the value associated with the action a of the state s_n. The parameters $0 \leq \alpha \leq 1$ and $0 \leq \lambda \leq 1$ are referred to as learning rate and discount factor respectively. The value of α determines how fast learning occurs, while λ models the importance that is attached to future rewards in comparison to immediate rewards.

B. *Results*

We added a network virtualisation module to NS3 [34]. The implementation is such that every time a VN request is accepted by the SN, the VN topology is created in NS3, and a traffic application starts transferring packets over the VN. Real traffic traces from CAIDA anonymised Internet traces [35] were used for the evaluations. We compare our DRA proposal to a static approach that allocates a fixed amount of resources to a VN through out its lifetime.

As can be seen from Fig. 4(a), the dynamic approach out performs the static one in terms of VN acceptance ratio by about 30%. This can be attributed to the fact that in the dynamic approach the SN always has more available resources than in the static case, as only the resources needed for actual transfer of packets is allocated and/or reserved for VNs. Fig. 4(b) shows that the static approach has an almost constant packet drop rate while that for the dynamic approach is initially high, but gradually converges to that of the static approach. The reason for this is that at the beginning of the learning

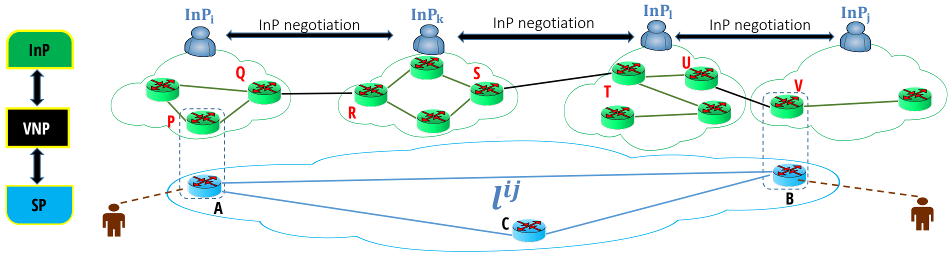


Fig. 5. Multi-Domain Survivable Virtual Network Embedding

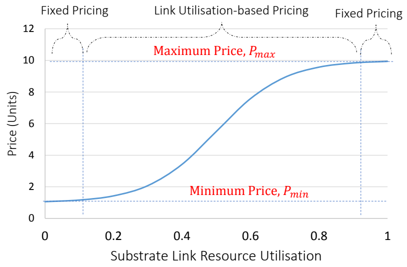


Fig. 6. Dynamic Substrate Resource Pricing

process, the dynamic approach varies the queue sizes quite considerably leading to more packet drops. This reduces as the agent learns from its errors and is finally able to make resource re-allocations without negatively affecting the network QoS. To even improve the rate of convergence of the learning as well as generalisation efficiency of our DRA proposal, we extended it using neural networks and neuro-fuzzy systems [17], [18]. In the extensions we achieved an even better acceptance ratio (up to 20% more) and a much faster convergence of the quality of service parameters.

IV. VIRTUAL NETWORK SURVIVABILITY (VNS)

In practice, physical networks do not remain operational at all times [36], hence making the provisioning of resources for backups and/or restorations an inevitable part of any survivable network resource management approach. Survivability in network virtualisation [37] involves consideration that substrate links and nodes can fail, and in ensuring that the virtual nodes or links mapped onto the failed substrate resources are not disrupted. This is usually achieved either by backing up secondary resources (proactive survivable virtual network embedding) before failures have actually occurred or provisioning the resources upon substrate resource failures (reactive survivable virtual network embedding) [38]. While proactive virtual network embedding avoids the delays and possible data loss that may be encountered if resources have to be provisioned upon failures, reserving some physical resources for un foreseen failures could result into inefficient resource utilisation for the substrate network.

In NVEs, survivability has only been considered for the single InP environment. In this dissertation, we propose a multi-entity based negotiation approach which allows InPs to minimise costs resulting from QoS violation penalties. We consider that the business model involves a virtual network provider (VNP) as a resource broker between SPs and InPs. Fig. 5 represents a general work flow of the proposed negotiation algorithm. Consider that a VNP wants to provision backup resources for the virtual link l^{ij}. We assume that the virtual link l^{ij} has already been mapped, with its two ends A and B being mapped by InPs InP_i and InP_j respectively. The VNP starts by determining an initial set of InPs to which the request can be sent. This initial set of InPs is such that it includes InPs that performed the initial mapping (and/or their direct neighbours) of the virtual link under consideration. For virtual link l^{ij}, the initial set would include the InPs InP_i and InP_j. With the InP set determined, the VNP sends the same mapping request (request to provision backup resources for a given virtual link) to each of the InPs in the set. The request includes the identity of the InPs that are mapping each end of the virtual link. On reception of a request from the VNP, a given InP begins by determining if it is able to complete the mapping on its own, i.e. if both ends of the virtual link are mapped with in its domain, and it has enough substrate link resources to provision the link. If the InP can perform the mapping on its own, then, it uses the pricing model (3) to determine the price, and then sends a proposal to the VNP. However, in the example of Fig. 5, InP_i is not able to complete the mapping on its own since one end of the virtual link is mapped by a different InP. In this case, InP_i would forward the request to (its direct neighbour) InP_k.

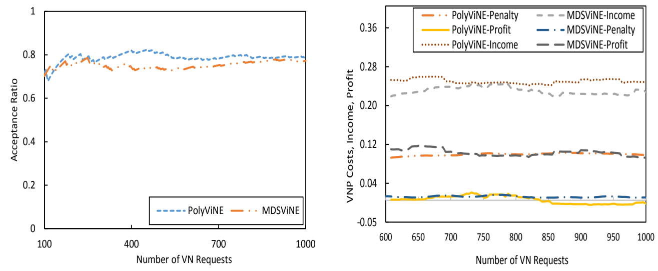
When an InP receives a forwarded request from one of his neighbours, it starts by ensuring that the inter-domain link connecting them has enough capacity to support the mapping being requested. If the inter-domain link does not have this capacity, then, the mapping cannot be completed, and the VNP will be informed about the failure. In our case, this means that InP_i must be able to provision link resources from node P (which maps one end of the virtual link), to node R (in the InP where the request has been forwarded). For instance, these resources could be along the substrate path PQR. At this point, since InP_k already has a connection to the node A of the virtual link (through the path PQR), the request issued from InP_k will include InP_k as the *most recent connection* to the virtual node. Therefore, the requests forwarded by InP_k will be a provisioning request for a link starting from InP_k to InP_j. Following a similar procedure, InP_l and InP_k will collaborate to create the connection RST, and finally, InP_l and InP_j will create the final path TUV. At this point, InP_j will send back its price to InP_l, who would, after adding his own cost forward his proposal to InP_k, and so on, until a final mapping proposal is delivered to the VNP. On reception of a proposal, the VNP may accept or reject it based on its own evaluation[2]. This is repeated for all the virtual links in the VN.

It is worth mentioning that just like related works [10], [37], our proposal only focusses on single substrate link failures. This is reasonable since network link failures occur about 10 times more than node failures [39], and given that about 70% of unplanned link failures are single link failures [36]. It should however be noted that any node failure can be considered as a failure of links adjacent to the node [10], and as such, our proposal can be extended to cover multiple link failures, and hence node failures.

Pricing Model: In order for InPs to generate proposals in response to a mapping request, they should be able to

---

[2] Due to space restrictions, we refer the reader to [11] for a description of how the InP evaluates a proposal, and the detailed negotiation protocol



(a) Acceptance Ratio (b) Average Costs, Income, Profits

Fig. 7. Evaluation of Negotiation for Survivable VNE approach

determine prices for their resources. We have chosen to use a hybrid pricing function that is based on the logistic function. This pricing model represents a dynamic pricing scheme that is based on the level of resource utilisation for the substrate network, which is restricted at either end by maximum and minimum allowed prices for the substrate resource in question. This pricing model has advantages over the constant pricing model that has been used in most network virtualisation proposals such as [20], [10] and [40], as it does not only allow prices to reflect network loading (hence encouraging better resource utilisation, and minimising network failures from over loading), but also ensures that resources have reserve prices (to cater for minimum fixed costs), and maximum prices to ensure competitiveness. Therefore, the price per unit of flow $P(s)$ on a substrate link s is determined as shown in Fig. 6, which is based on (3).

$$P(s) = l_x^s\left(P_{min}^s + \frac{P_{max}^s - P_{min}^s}{1+exp\Big(c_1 - c_2 u(s)\Big)}\right) \tag{3}$$

where P_{min}^s is the minimum acceptable price for s whose resource utilisation level is $u(s)$ and length l_x^s, and P_{max}^s is the maximum allowed price. c_1 is a constant aimed at shifting the pricing function horizontally (and hence affecting the levels of resource utilisation where the *minimum* and *maximum* prices come into effect), and c_2 is a constant that determines the slope of the pricing function (and hence the rate at which pricing changes from minimum pricing to maximum price). Therefore, the total price C^s that should be paid for all the secondary flows i.e. flows over backup resources f_v^s is given by (4)

$$C^s = \sum_{all f_v^s} f_v^s P(s) \tag{4}$$

A. Results

We extended the Java Agent Development Framework (JADE) [41] to implement a discrete event simulator. The proposed negotiation protocol is implemented as an extension to the ACLMessage [41], which is compliant to the FIPA 2000 specifications [42]. For each substrate link, the mean time between failures (MTBF) and mean time to repair (MTTR) were based on a characterisation of link failures in a real ISP backbone performed in [43] and both followed a Weibull distribution [44]. Detailed simulation parameters can be found in [11]. We compare our multi-domain survivable VNE (MDSViNE) proposal with PolyViNE [40] which performs multi-domain embedding without consideration for survivability. We also note that it is this approach which is used for performing the initial VNE before our survivability algorithms are initiated. Node mapping is performed using the greedy approach in [32] while link mapping is performed by formulating the problem as a multicommodity flow (MCF) [45] and solving the resulting linear program using CPLEX12.6 [30].

Fig. 7(a) shows that that PolyViNE has a marginally better acceptance ratio compared to MDSViNE. This is expected since MDSViNE commits some of the link resources for failures, and hence has less resources to accept VN requests. For this reason, we note in Fig. 7(b) that the total income of PolyViNE is slightly higher than that of MDSViNE. However, it can be seen from the same figure that the costs (QoS violation penalties) incurred by PolyViNE are much higher than those of MDSViNE, which results into MDSViNE having a significantly higher profit (about 4 times) than PolyViNE.

V. CONCLUSION

One of the fundamental requirements in network virtualisation is the assignment of physical network resources to virtual networks. Because it determines how many virtual networks can share a given set of physical resources at any given point, resource management directly affects the profitability and hence attractiveness of network virtualisation to infrastructure providers. This dissertation makes contributions to this very important part of network virtualisation. To this end, the resource management problem was split into three clear sub-problems; (1) virtual network embedding (VNE), (2) dynamic resource allocation (DRA), and (3) virtual network survivability (VNS). We proposed a path-generation approach for VNE, self-management learning techniques for DRA, and milti-entity-based negotiation algorithms that use dynamic resource pricing for VNS. Through simulations, we show that our proposals can bring significant improvements to VN acceptance ratio, and hence InP profitability.

However, our proposals for each of the three problems may still be improved. For VNE, there is room to enhance the computational complexity further, say, by formulating linear relaxations of the mathematical programs. In addition, the slightly poor performance of the proposed learning techniques at the beginning of the learning may be tackled in future by employing an initial offline learning step, such that actual online resource allocations start from optimal policies. Finally, our VNS proposal only considered single substrate link failures. Future works will involve extending the approach to consider multiple link failures, and hence node failures. For all three sub-problems, it may also simplify resource management in NVEs, by employing SDN [46].

DISSERTATION MATERIAL

The contributions of this research have been published in: [8], [17], [12], [18], [19], [15], [16], [14], [13]. Some of these contributions have also been part of technical documents and/or deliverables of some research projects such as [47] and [48]. The full dissertation can be downloaded from: [11]. The dissertation was accepted and successfully defended at the Universitat Politecnica de Catalunya (UPC) on November 06 2014, attaining both Cum Laude and International Doctor mentions.

ACKNOWLEDGMENT

Rashid was partly funded by a grant from the Spanish Ministry of Economy and Competitiveness (MINECO), FLAMINGO, a Network of Excellence project (318488) supported by the European Commission under its Seventh Framework Programme, the EVANS project (PIRSES-GA-2010-269323) and project TEC2012-38574-C02-02 from Ministerio de Economia y Competitividad.

REFERENCES

[1] A. Carzaniga and A. L. Wolf. Content-based networking: A new communication infrastructure. In Birgitta Knig-Ries, Kia Makki, S. A. M. Makki, Niki Pissinou, and Peter Scheuermann, editors, *Infrastructure for Mobile and Wireless Systems*, volume 2538 of *Lecture Notes in Computer Science*, pages 59–68. Springer, 2001.

[2] N.M. M. K. Chowdhury and R. Boutaba. A survey of network virtualization. *Computer Networks*, 54(5):862 – 876, 2010.

[3] J. S. Turner and D. E. Taylor. Diversifying the internet. In *GLOBECOM*, page 6. IEEE, 2005.

[4] C. Diot, B. Neil, L. Bryan, and K. D. Balensiefen. Deployment issues for the ip multicast service and architecture. *IEEE Network*, 14:78–88, 2000.

[5] J.G. Jayanthi and S.A. Rabara. Ipv6 addressing architecture in ipv4 network. In *Communication Software and Networks, 2010. ICCSN '10. Second International Conference on*, pages 461–465, Feb 2010.

[6] A. Khan, A. Zugenmaier, D. Jurca, and W. Kellerer. Network virtualization: a hypervisor for the internet? *IEEE Communications Magazine*, 50(1):136–143, 2012.

[7] T. E. Anderson, L. L. Peterson, S. Shenker, and J. S. Turner. Overcoming the internet impasse through virtualization. *IEEE Computer*, 38(4):34–41, 2005.

[8] R. Mijumbi, J.L. Gorricho, J. Serrat, M. Claeys, F. De Turck, and S. Latre. Design and evaluation of learning algorithms for dynamic resource management in virtual networks. In *Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS)*, NOMS2014. IEEE, 2014.

[9] M. Chowdhury, M.R. Rahman, and R. Boutaba. Vineyard: Virtual network embedding algorithms with coordinated node and link mapping. *Networking, IEEE/ACM Transactions on*, 20(1):206 –219, feb. 2012.

[10] M. R. Rahman and R. Boutaba. Svne: Survivable virtual network embedding algorithms for network virtualization. *Network and Service Management, IEEE Transactions on*, 10(2):105–118, 2013.

[11] R. Mijumbi, J. Serrat, and J. L. Gorricho. *Self-Managed Resources in Network Virtualisation Environments*. PhD thesis, Universitat Politecnica de Catalunya, 2014. URL: http://www.maps.upc.edu/rashid/files/Rashid_PhD_Dissertation.pdf.

[12] R. Mijumbi, J. L. Gorricho, J. Serrat, and R. Boutaba. Path generation-based virtual network embedding. *Networking, IEEE/ACM Transactions on*. Submitted.

[13] R. Mijumbi and J. Gorricho, J. L.and Serrat. Multi-agent based resource allocation in next generation virtual networks. In *Proceedings of Barcelona Forum on Ph.D. Research in Communication and Information Technologies*, 2012.

[14] R. Mijumbi, J. L. Gorricho, and J. Serrat. Autonomic resource management in virtual networks. In *Scalable and Adaptive Internet Solutions (SAIL) Summer School Work in Progress Session*, 2012.

[15] R. Mijumbi, J. L. Gorricho, and J. Serrat. Contributions to efficient resource management in virtual networks. In A. Sperotto, G. Doyen, S. Latr, M. Charalambides, and B. Stiller, editors, *Monitoring and Securing Virtualized Networks and Services*, volume 8508 of *Lecture Notes in Computer Science*, pages 47–51. Springer Berlin Heidelberg, 2014.

[16] R. Mijumbi, J. L. Gorricho, and J. Serrat. Learning algorithms for dynamic resource allocation in virtualised networks. In *Proceedings of Workshop on Management of Large Scale Virtualized Infrastructures: Smart Data Acquisition, Analysis and Network and Service Management in the Future Internet*, EUCNC2014, 2014.

[17] R. Mijumbi, J. L. Gorricho, J. Serrat, M. Claeys, F. De Turck, and J. Famaey. Neural network-based autonomous allocation of resources in virtual networks. In *Proceedings of the European Conference on Networks and Communications (EuCNC)*, EuCNC2014, June 2014.

[18] R. Mijumbi, J. L. Gorricho, J. Serrat, M. Shen, K. Xu, and K. Yang. A neuro-fuzzy approach to self-management of virtual network resources. *Expert Systems with Applications*, 42(3):1376 – 1390, February 2015.

[19] R. Mijumbi, J. L. Gorricho, J. Serrat, J. Rubio-Loyola, and R. Agüero. Survivability-oriented negotiation algorithms for multi-domain virtual networks. In *Proceedings of 10th International Conference on Network and Service Management (CNSM). Short Paper*, CNSM2014. IEEE, 2014.

[20] A. Fischer, J.F. Botero, M. Till Beck, H. de Meer, and X. Hesselbach. Virtual network embedding: A survey. *Communications Surveys Tutorials, IEEE*, 15(4):1888–1906, Fourth 2013.

[21] David G. A. Theoretical approaches to node assignment. Unpublished Manuscript, December 2002.

[22] K. Elbassioni, N. Garg, D. Gupta, A. Kumar, V. Narula, and A. Pal. Approximation Algorithms for the Unsplittable Flow Problem on Paths and Trees. In Deepak D'Souza, Telikepalli Kavitha, and Jaikumar Radhakrishnan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2012)*, volume 18 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 267–275, Dagstuhl, Germany, 2012. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[23] S. G. Kolliopoulos and C. Stein. Improved approximation algorithms for unsplittable flow problems. In *FOCS*, pages 426–435. IEEE Computer Society, 1997.

[24] M. X. Goemans and D. P. Williamson. Approximation algorithms for np-hard problems. chapter The Primal-dual Method for Approximation Algorithms and Its Application to Network Design Problems, pages 144–191. PWS Publishing Co., Boston, MA, USA, 1997.

[25] L. R. Ford, Jr. and D. R. Fulkerson. A suggested computation for maximal multi-commodity network flows. *Manage. Sci.*, 50(12 Supplement):1778–1780, December 2004.

[26] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. P. Savelsbergh, and P. H. Vance. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, 46(3):pp. 316–329, 1998.

[27] S. Bradley, A. Hax, and T. Magnanti. *Applied Mathematical Programming*. Addison Wesley, 1977.

[28] D. P. Bertsekas and D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, September 1999.

[29] A. Medina, A. Lakhina, I. Matta, and J. Byers. Brite: An approach to universal topology generation. In *Proceedings of the Ninth International Symposium in Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, MASCOTS '01, pages 346–353, Washington, DC, USA, 2001. IEEE Computer Society.

[30] IBM ILOG CPLEX Optimizer. http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/about/. Accessed: 2014-07-13.

[31] M. Yu, Y. Yi, J. Rexford, and M. Chiang. Rethinking virtual network embedding: Substrate support for path splitting and migration. *SIGCOMM Comput. Commun. Rev.*, 38(2):17–29, March 2008.

[32] Y. Zhu and M. Ammar. Algorithms for assigning substrate network resources to virtual network components. In *INFOCOM. 25th IEEE International Conference on Computer Communications.*, pages 1–12, 2006.

[33] R. S. Sutton and A. G. Barto. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998.

[34] Network Simulator 3. http://www.nsnam.org/. Accessed: 2014-02-17.

[35] The CAIDA Anonymized Internet Traces 2012 - 20 December 2012, equinix sanjose.dirB.20121220-140100.UTC.anon.pcap.gz. http://www.caida.org/data/passive/passive_2012_dataset.xml. Accessed: 2014-02-17.

[36] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C.N. Chuah, and C. Diot. Characterization of failures in an IP backbone. In *INFOCOM 2004*, 2004.

[37] S. A. Khan Herker and X. An. Survey on survivable virtual network embedding problem and solutions. In *International Conference on Networking and Services*, ICNS 2013, March 2013.

[38] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee. Survivable wdm mesh networks. *Lightwave Technology, Journal of*, 21(4):870–883, April 2003.

[39] P. Gill, N. Jain, and N. Nagappan. Understanding network failures in data centers: Measurement, analysis, and implications. *SIGCOMM Comput. Commun. Rev.*, 41(4):350–361, August 2011.

[40] F. Samuel, M. Chowdhury, and R. Boutaba. Polyvine: policy-based virtual network embedding across multiple domains. *Journal of Internet Services and Applications*, 4(1):6, 2013.

[41] F. L. Bellifemine, G. Caire, and D. Greenwood. *Developing Multi-Agent Systems with JADE.* Wiley, 2007.

[42] FIPA ACL Message Structure Specification. http://www.fipa.org/specs/fipa00061/. Accessed: 2014-04-02.

[43] A. Markopoulou, Gianluca Iannaccone, Supratik Bhattacharyya, Chen-Nee Chuah, Y. Ganjali, and C. Diot. Characterization of failures in an operational ip backbone network. *Networking, IEEE/ACM Transactions on*, 16(4):749–762, Aug 2008.

[44] R. B Abernethy. *The New Weibull handbook: reliability and statistical analysis for predicting life, safety, supportability, risk, cost and warranty claims; 5th ed.* Dr. Robert B. Abernethy, 2006.

[45] C. Barnhart, N. Krishnan, and P. H. Vance. Multicommodity flow problems. In C. A. Floudas and P. M. Pardalos, editors, *Encyclopedia of Optimization*, pages 2354–2362. Springer US, 2009.

[46] R. Mijumbi, J. Serrat, J. Rubio-Loyola, N. Bouten, F. De Turck, and S. Latre. Dynamic resource management in SDN-based virtualized networks. In *Network and Service Management (CNSM), 2014 10th International Conference on*, pages 412–417, Nov 2014.

[47] End-to-end Virtual Resource Management across Heterogeneous Networks and Services (Evans). 7th FP of the European Union. http://www.fp7-evans.eu/.

[48] Management of the Future Internet (Flamingo). 7th FP of the European Union. http://www.fp7-flamingo.eu/.