Analysis of Path Splitting and Migration in the Virtualisation of Mobile Substrates

Name SURNAME[1], Name SURNAME[2]
[1]*Organisation, Address, City, Postcode, Country*
Tel: +countrycode localcode number, Fax: + countrycode localcode number, Email:

Abstract: Network virtualisation is a way to share heterogeneous physical networks by different Virtual Operators that aim to run services in an isolated way. In order to share efficiently this Physical Substrate (PS) it is necessary to set up a coordinated procedure to provide the best fitted physical resources for the Virtual Network Requests (VNRs). Our work is focused on the development of mapping algorithms for mobile environments and the analysis of the embedding problem. This paper analyses the impact of mobility when no special techniques are applied, compared to the situation when Path Splitting and Migration (PSM) are applied. In addition to PSM, we present some specific modifications to the global embedding procedure for mobile environments, trying to make it more efficient. A simulation-based study lets us show the good results of applying such techniques in mobile substrates. This work has been carried out partially inside the European Project 4WARD (FP7 Project reference: 216041).

Keywords: Virtualisation, Mobile Substrate, Path Splitting, Path Migration.

1. Introduction

Network Virtualisation (NV) is a key concept for the construction of the Future Internet, because of its potential to allow multiple network architectures, each customized to a particular application or user community, to run on a common PS. NV is based on sharing a common PS by several Service Providers running their own services in a fully isolated and protected way. Such a new network paradigm has also a huge impact on the business aspects with the irruption of new actors in the game: the Substrate Network Provider (SNP, owner of the physical resources), the Virtual Network Provider (VNP, mainly responsible for the creation of virtual networks according to the requests of Virtual Network Operators), and the Virtual Network Operator (VNO, who offers final services to customers and request some specific configuration, VNRs, to be able to accomplish their services).

Several VNOs running on the same PS require efficient embedding techniques able to map the VNRs coming from different VNOs. The embedding problem becomes extremely difficult and it is mainly characterized by: node and link requirements, Admission Control and online requests with a certain duration time.

So far this embedding problem has been restricted to the mapping of VNRs over static PSs, which is actually a rather challenging problem. However, Future Internet requirements envisage a Network rather different to the current one, where heterogeneous networks work seamlessly. This heterogeneity implies different networks paradigms and concepts such as "the Internet of Things" [3] suggest the idea of a Future Internet in which most nodes are wireless and with the capability of motion.

In the embedding process, the availability of physical resources varies in time due to the dynamicity of the requests, which makes it very challenging the mapping of a new request in an efficient way. Moreover, if the PS is mobile, complexity increases considerably, and the embedding mechanism needs to be flexible enough to adapt the VN mappings to the changing location of the physical resources. Our work is focused on the development of embedding algorithms for mobile environments and assessing the impact of mobility on the embedding problem, taking [1] as a fundamental base.

This paper is organised as follows: Section 2 highlights the main objectives of our work; we developed and implemented embedding algorithms within a simulation environment, technically described in Section 3; Section 4 covers the definition of a set of scenarios (PSs) and simulations, describing which are the relevant metrics used; Section 5 shows the results of all simulations and the assessment of the benefits that can be concluded; and finally, Section 6 describes main conclusions and future work we envisage.

2. Objectives

The main goal of our work is focused on two ideas: the evaluation of the performance of PSM in mobile substrates; and the enhancement of PSM application algorithms in order to make the embedding process more efficient in mobile environments.

1. MAIN OBJECTIVE: To assess the performance of techniques such as PSM in the embedding algorithm of VNets, within a mobile PS from the VNP's perspective.
2. Other objectives can be derived and specifically aimed, from this principle:

- To maximize the flexibility (maintaining actual mappings whenever implied physical nodes are moving) and efficiency (map as many VNets as possible, in a PS where most nodes are mobile) of the embedding algorithm in mobile environments.
- To look for specific criteria which deal with mobile nodes within the mapping algorithm -> specific changes introduced to the previous simulator [2], in order to cope with problems derived from mobility (look for the shortest path to avoid incrementing the number of nodes susceptible to move and break the route, to update the PS status more often so as to have up-to-date information about node locations, etc…)

3. Technical description of the Mobile-Aware Embedding Algorithm

Our simulation environment is based on the previous work done by [1] and available in [2]. Before we start with the technical description of the different algorithms within the overall procedure, we are going to highlight the structure of the whole program. The simulator operation is divided into time windows, within which the different algorithms are applied in the following sequence.

- Previously mapped VNRs that have already been completed are released.
- The mobility management functions are started (section 3.3): location update procedure and repairing of affected VNRs due to lost links or nodes (dynamicity).
- The VNRs arriving inside that time window are attended in decreasing order of the amount of bandwidth (BW) they require. Node Mapping and Link Mapping run sequentially per VNR. Some VNRs do not succeed and are stored in a VNR Queue; they will be re-tried again in a certain interval's time (incremental periods). After 3 attempts, if the VNR cannot be mapped, then it is definitely discarded (rejected).
- If permitted, migration is applied periodically, every 10 completed and released VNRs; otherwise the procedure finishes.

3.1 Node Mapping Algorithm

We use a greedy algorithm to map the virtual nodes to the real ones, applying the criterion of "maximum availability in substrate resources". We also consider location restrictions for VNRs requiring a specific geographical area. In each time window, the system keeps track of the state of the PSs and the incoming VNRs are sorted according to the amount of BW they require, so the mapping process starts from the VNRs asking for maximum BW until all the virtual nodes have been mapped or rejected.

Expression (1) quantifies all parameters in the criterion to choose a certain n^s:

$$H(n^s) = \frac{CPU(n^s)\ x \sum_{l^s \in L(n^s)} \mathrm{bw}\ (\mathrm{l}^s)}{1+\alpha}, \tag{1}$$

where $L(n^s)$ is the set of all adjacent substrate links of n^s, $CPU(n^s)$ is the remaining CPU resource of n^s, and $bw(l^s)$ is the available BW resource for the substrate link l^s. α is called Mobility-factor and characterises the "grade of movement" of a node, updated by the system as it learns from the mobility history. This factor allows the VNP to choose in the first place nodes which are less likely to move in the next time slot, than others which are estimated as more likely to be moving. A node can have one of these three possible states: 0 - Stop (node which has not moved for a certain time period), 1 - Low Mobility (node which has just begun to move or has just stopped), and 2 - High Mobility (node which has been moving for a long time period, or at a high speed). The Mobility-factor, α, is calculated as the product of the correspondent state and speed of a node.

3.2 Link Mapping Algorithm with Path Splitting and Migration

Once the node mapping is done, the virtual links are embedded to physical links. As optimal embedding is not possible [4] [5], specially when mobility is present, we use a shortest path algorithm [6] with the possibility of path splitting if the VNR allows it (see section 4.2). We use a shortest path search for all the VNRs in a time window, i.e. sequentially the virtual links of every VNR are mapped onto the lowest number of physical links (hops) possible. As it is described in [1], Path Splitting allows the VNP to embed a virtual link of the mapping request in multiple substrate paths. This way, the embedding problem becomes more flexible, specially in those physical networks that are mobile and with scarce resources, maximizing the number of supported requests, and therefore, the succeeded revenue for the VNP.

Unlike [1], we do not apply path splitting as a Multicommodity Flow Problem (MFP) [7], but we deal with the link mapping sequentially, VNR after VNR and, trying to embed all of them without splitting in the first place. If this is not possible, then path splitting is applied to VNRs that allow it. MFP is more convenient for static PSs, but not for mobile PSs (many broken links because of the dynamics), where network changes require a less optimal but quicker (more effective) mechanism. In addition we avoid the congestion problem reported in [1].

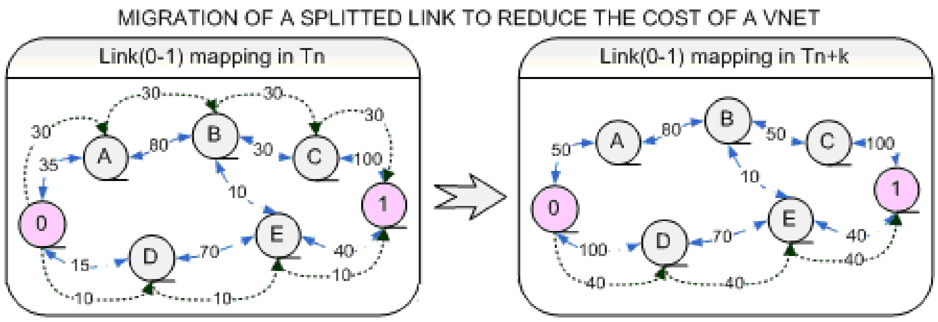

Figure 1: Path Splitting and Migration principle

Imagine that at a certain moment, T_n, our physical network is supporting such a number of requests that the available BW left is distributed as we can see in figure 1 (left). At this time, a new request arrives asking for 40 units of BW between nodes 0 and 1. Using a traditional embedding algorithm it is impossible to allocate the new virtual link because there is no path in our substrate with such spare capacity. However, Path Splitting allows the VNP to embed the VNR dividing the required BW into two separate paths (30 units through A-B-C, and 10 units through D-E).

The right side of figure 1 shows that after k time units, at T_{n+k}, the BW availability in the physical links has changed. If we recalculate the best option to map a virtual link from node 0 to 1, we obtain a lower cost of the path through (D-E) because it is the shortest one (3 hops instead of 4), so the migration of the link is produced. Periodically, in order to keep the mapping efficient in time and to increase the possibility of accepting future VNRs, this migration algorithm is used.

3.3 Mobility Management: Node and Link Remapping

Mobility makes the mapping process very unstable if no special considerations are taken into account. In addition to the previously described issues, after a VNR has been embedded, some monitoring is necessary to deal with broken links or location-dependant mappings. The system deals with:

- Node Remapping Algorithm: for location-aware VNRs, when the node in motion corresponds to a virtual node that is going out of the interest area.
- Link Remapping Algorithm: when the mobile node is out of reach of the previously connected neighbour node. There is a gap that needs to be repaired for the continuity of the VNR.

Link Remapping Algorithm	Node Remapping Algorithm
Step 1 - Look for lost old links. If there are not, stop. *Step 2* - Check if this link was mapped for any current VNR. If it is not, GoTo Step 1. *Step 3.1* - If the moving node corresponds to a virtual node, repair the gap mapping new physical links and nodes. *Step 3.2* – Else (middle node), try to repair the gap avoiding the moving node *Step 4* - If the virtual link has been repaired GoTo Step 1. *Step 5* – Else, try to repair the complete virtual link (start Link Mapping Algorithm). If repaired GoTo Step 1 *Step 6* – Else, release the mapped VNR and send it to the VNR Queue.	*Step 1* – Look for nodes whose area has changed. If there are not, stop. *Step 2* – Check if the node belongs to any allocated VNR. If it is not, GoTo Step 1. *Step 3.1* – If the new area is valid according to the VNR location restrictions, GoTo Step 1 *Step 3.2* – Else, release the virtual links associated to this virtual node. *Step 4* – Map the virtual node into a valid physical node (start Node Mapping Algorithm). *Step 5* – Else, try to map the virtual link into a substrate path, using Path Splitting or not, according to the VNR requirements. *Step 6* – Else, release the mapped VNR and send it to the VNR Queue.

Table 1: Mobility Management Algorithms

Figure 2 represents both types of remapping, combined at the same time. Node 0, which is a virtual node for a certain VNR, moves and gets out of the requested cell, so it is necessary to remap a virtual node in the same cell for the VNR; and node B (intermediate node) moves away, so its virtual links are lost and the affected VNR needs to repair the gap.

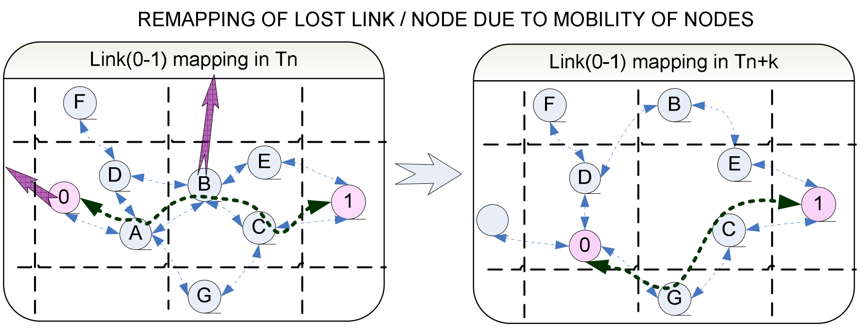

Figure 2: Mobility Management: Node and Link Re-mappings

4. Definition of Scenarios, VNRs and Simulations

The initial definition of our scenarios is based on [2], where some specific capabilities related to mobility have been added, such us: the cell-index (scenario size subdivided), the location parameter and the random mobility patterns for nodes.

All scenarios are generated with N=100 wireless nodes, placed randomly in a bi-dimensional square area. Each node has 100 units of CPU capacity at the beginning. We assume that two physical nodes will have connectivity, and could establish a wireless link, if the distance between them is less than their radio coverage. For all simulations, we have used either 1/2 or 1/4 of the length of a cell diagonal $(X_{cell}^2+Y_{cell}^2)^{1/2}$, as the radio coverage.

With respect to the VNRs used in the simulations, we defined a set of 500 requests for all cases. The number of requested virtual nodes in each request, n, is randomly determined by a uniform distribution between 3 and 12; the link requirements are as follows: the probability of asking for a link between each pair of nodes is 50%, which means that the number of requested links for a VNet is [n (n-1) / 4] on average. The requests arrival is modelled by a Poisson process with and average of 0.5 requests per time window.

Due to the introduction of mobility in the physical substrate, we found it interesting to evaluate the benefits of path splitting and migration according to the mobility pattern applied to the scenario. We aim at a certain relation between the benefits of these techniques depending on the grade of mobility. For this purpose, we run several simulations for the same scenario applying different grade of mobility to its nodes: Static (St-0% are mobile); Low Motion (LM-25% are mobile); Medium Motion (MM-50% are mobile); High Motion (HM-75% are mobile); and Full Motion (FM-100% are mobile).

Regarding location requirements, we considered three types of possible requests from the VNO, depending on the specificity of the location requirement:

1. Location for all the nodes: The VNO requests a specific location per node. This type of VNR is the most restrictive one because just a small subset of substrate nodes can fit.
2. Central area: In this case, the VNO requests a location for the whole VNR. The VNP will try to map all the nodes in that area, or the surrounding ones.
3. No location: This is the simplest case because the VNO does not require any location for the nodes. Our algorithm, with the objective of minimizing the mobility effects, tries to map a virtual link onto the physical path with fewer hops. For this purpose it is estimated which area has more available resources, and mark it as the central area of the VNR. The difference between this case and the second one is that the algorithm uses an increasing index of neighbouring cells until the request is finally accepted or rejected.

In all simulations there are 30% requests of type 1, 40% requests of type 2 and 30% requests of type 3.

In order to configure a set of coherent simulations and to compare results among them, we have assigned several combinations of values for the parameters described above. Table 2 gathers the relevant configuration values the simulations covered in our analysis:

Scenario	Nº of Nodes	Map Size	Nº of cells	Radio Coverage
A	100	200 x 200	4 x 4	40
B	100	200 x 200	4 x 4	20
C	100	200 x 200	8 x 8	40
D	100	400 x 400	4 x 4	40

Table 2: Configuration parameters for scenario simulations

For each scenario in table 2 we have run a subset of simulations incrementing the percentage of VNRs allowing Path Splitting, both for the case where the migration is permitted and for the case where it is not.

5. Analysis of Results

Once the simulation parameters and goals have been defined at previous sections, we will analyse here the obtained results. Results are displayed in different graphs, showing the most interesting data which will allow the reader to understand the benefits of using PSM in the virtualisation of mobile networks.

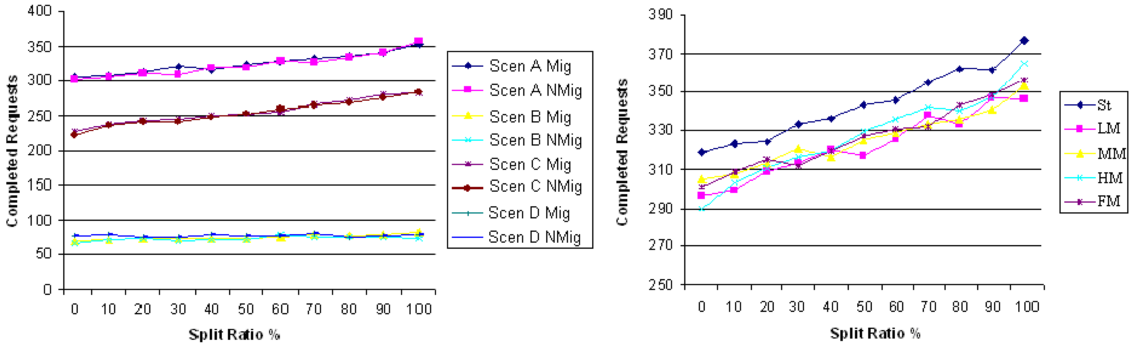

Figure 3: Completed VNRs for each scenario (left) and as a function of the grade of mobility (right)

On the left side of figure 3 we can see the amount of completed VNRs as a function of the splitting-ratio for scenarios in table 2, with (Mig) and without (NoMig) migration. All simulations were run with a medium grade of mobility. On the right side of figure 3 we have the number of completed VNRs for all the possible mobility grades (Static-St- to Full Motion-FM-) and splitting ratios, but specifically for scenario A.

1. “The benefits of using Path Splitting are extensible to mobile substrates”

The main conclusion we can extract from the obtained results in figure 3 (left) is that Path Splitting is a beneficial mapping technique, also for wireless mobile substrates, which helps the VNP to maximize the number of allocated VNRs, and hence the revenue obtained. There are a series of considerations that can be extracted from the results:

- A low relation between the map size and radio coverage (figure 3 -left-, scenarios B and D) makes the Path Splitting non-useful because there are not enough links per node. Few VNRs exhaust the PS since, having very few physical links, they all get fully occupied quickly and there is no margin for PSM to gain any benefit.
- The smaller the cell size is, the lower number of nodes per cell we have, which makes more difficult to map a specific node for location-aware VNRs. This is a reason why the number of completed VNRs is decreased, for example in the results of scenario C (1.56 nodes per cell on average) in figure 3 (left).

2. “The number of completed VNRs does not vary much with the grade of mobility”

As we expected, analysing a single scenario (A), the biggest amount of completed VNRs was reached for the static case. Without mobility, VNRs are not interrupted and stopped. We can also see in figure 3 (right) how the number of completed VNRs does not change a lot as mobility grade increases; graphs are quite close in the scale. That is because the Repairing-Ratio (explained next) does not decrease as the grade of mobility is incremented.

3. “The Repairing-Ratio increases with the grade of mobility”

The Repairing-Ratio is defined by equation 2 and it gets improved as we increment the grade of mobility for a certain scenario (see figure 4 –right-). “Stopped VNRs” are all requests that need to be interrupted by mobility issues; “Repaired VNRs” are those affected requests that can be repaired by the Remapping Algorithm; “Remapped VNRs” are those requests that could not be repaired, but that are completely remapped in following time-windows.

$$\text{Repairing - Ratio} = \frac{(\text{Remapped VNRs / Stopped VNRs}) \cdot \text{RepairedVNRs}}{\text{Completed VNRs}} \qquad (2)$$

Increasing mobility for a scenario makes the number of “Remapped VNRs” increase (more attempts needed to maintain a VNR), but the number of “Stopped VNRs” also goes up, so the proportion remains (roughly) constant. What is increased is the number of “Repaired VNRs”, and that is the main reason why the Repairing-Ratio grows up with mobility.
The percentage of VNRs allowing splitting is also a key factor showed in figure 4 (right). More splitted paths imply more affected links due to mobility, so the “Repaired VNRs” will increase, and the Repairing-Ratio grows with the splitting ratio.

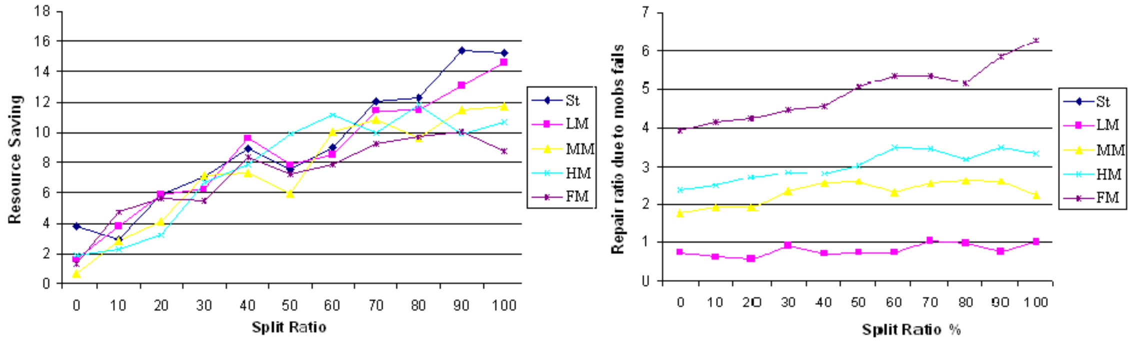

Figure 4: Resource saving (left) and Repairing-Ratio (right) as functions of mobility grade for Scenario A

4. “Migration improves the saving of resources on mobile substrates”
Migration reduces the cost of allocating VNRs on the substrate as it is displayed on figure 4 (left). The Resource saving shows the average difference of costs between the case where migration is applied and the one where it is not (for scenario A). This Resource saving is obtained as the total mapping-costs divided by the average mapping-duration (M^S) and by the number of completed VNRs.

$$C(n^s) = \frac{\sum_{l^v \in L^V} \sum_{p \in P^s(l^v)} \text{hops}(p)\,\text{bw}(l^v)}{Av(M^S) \cdot \text{Completed Requests}} \tag{3}$$

The splitting ratio increases this saving because the number of hops per path will significantly increase the allocating costs. This way, reallocations will have a higher margin of cost-reduction. The grade of mobility also affects the Resource saving (figure 4 – left-), negatively in this case: the higher the movement is, the higher the cost of the same VNR becomes; migration can reduce costs but not so significantly.

6. Conclusions and Future Work

It is a quite predictable fact that mobility introduces much more complexity to the provisioning of virtual networks. Otherwise, the Future Internet will comprehend a very heterogeneous environment, where interoperability is a key concept, and the global substrate network is expected to be conformed as well by fully mobile topologies, such us: Wireless Sensor Networks (WSNs), Mobile or Vehicular Ad-hoc NETworks (MANETs or VANETs), etc. Virtualisation of such PSs seems to be challenging, but with this paper we try to show that some steps can be taken although there is still much work to do.

Our analysis was focused on the performance of PSM in the embedding of VNRs on a mobile physical environment. In order to adjust and refine these techniques to be applied to mobile PSs, we proposed and implemented some enhanced algorithms (presented in section 3) and evaluated their impact on the embedding procedure. Our definition of simulations was aimed at the study of PSM in scenarios with different grades of mobility, comparing results given by simulations with several splitting ratios, and evaluating the resource saving (cost calculation) when migration was applied.

The analysis of results obtained with the whole set of simulations showed several and important conclusions, which can be summarised as follows:

- Using PSM in the embedding over mobile substrates increases the benefit for the VNP and for VNOs: the VNP obtains higher revenue in terms of number of completed requests, and VNOs obtain a better performance since most of their VNRs are served.
- Mobility does not drastically decrease the number of completed VNRs, but only if PSM techniques are applied in the embedding, for the same scenario.
- Applying PSM stabilizes the mapping as mobility increases: when simulations increment the grade of mobility in the PS, a decrease on the number of Repaired VNRs could have been expected, but results showed that the proportion of restored VNRs was maintained and so, the benefit of PSM is still notable even in fully mobile PSs.
- Migration improves the resource saving for the VNP on mobile PSs: even in full mobile scenarios, the migration algorithm reduces the cost of embedded VNRs for the VNP.

All our efforts have been focused on the study of mobile physical environments, and one of the real challenges for virtualisation here is the time scale problem. This is one key future line that we intend to work on. PSM has been proved to obtain beneficial results, but we can only assure it without looking at response times. In a fully mobile substrate where several VNRs are mapped, the effects of mobility can be quite a few in an extremely short time period. The process of interrupting affected VNRs and repairing links requires some time, which certain final services (real time) cannot afford waiting. Simulations should take this response time into account and introduce some constraints to the performance of the Remapping algorithms. Splitting and migration as well should be evaluated with respect to their time consumption.

Another important line of future work is the analysis and classification of different kinds of mobile PSs, where virtualisation would have full sense in terms of final services offered, and time constraints. The grade of mobility of the PS is probably a key factor to analyse user applications to be supported by virtualised networks, for instance.

Regarding the design of the simulation environment presented, we can also identify some future improvements. The Cost and Revenue estimation for the VNP could be enhanced to include more criteria: the number of attempts consumed to finally embed a successful VNR should increment the cost of it, the duration of a mapped VNR should count for the revenue obtained, VNRs could be attended according to a certain priority schema, i.e. “attend first the shortest VNR”, or “attend first the most location-restrictive VNR”.

References

[1] M. Yu, Y. Yi, J. Rexford and M. Chiang. Rethinking Virtual Network Embedding: Substrate Support for Path Splitting and Migration. *ACM SIGCOMM*, April 2008.

[2] Virtual Network Embedding Simulator. http://www.cs.priceton.edu/~minianyu/embed.tar.gz

[3] Internet Reports 2005: The Internet of Things. 2005, 7th edition.

[4] J. Kleinberg. Approximation algorithms for disjoint paths problems. PhD thesis, MIT, 1996.

[5] S. G. Kolliopoulos and C. Stein. Improved approximation algorithms for unsplittable flow problems. In IEEE Symposium on Foundations of Computer Science, pages 426–435, 1997.

[6] http://en.wikipedia.org/wiki/Shortest_path_problem

[7] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. Network Flows: Theory, Algorithms, and Applications. Prentice Hall, 1993.

[8] L. Peterson, S. Shenker, J. Turner. Overcoming the Internet impasse through Virtualization. ACM HotNets III, 2004

[9] GENI http://www.geni.net/

[10] I. Houidi, W. Louati and D. Zeghlache. A Distributed Virtual Network Mapping Algorithm. Submitted to IEEE International Conference on Communications (ICC 2008), 2008

[11] N. Feamster, L. Gao, J. Rexford. How to lease the Internet in your spare time. Georgia Tech Technical Report GT-CSS-06-10, August 2006