Proceedings of the International Conference on Computer and Information Science and Technology
Ottawa, Ontario, Canada, May 11 – 12, 2015
Paper No. 110

A QoS-based Resource Selection Approach for Virtual Networks

Saba Behrouznia, Rachida Dssouli
Concordia University
1455 De Maisonneuve, Montreal, Quebec, H3G 1M8, Canada
saba.behrouznia@gmail.com; rachida.dssouli@concordia.ca

May El Barachi
Zayed University
Khalifa City B, Abu Dhabi, United Arab Emirates
may.elbarachi@zu.ac.ae

Abstract- The key challenge in virtual networks is the problem of assigning virtual nodes and links to physical resources. Virtual network mapping/embedding consists in finding the most suitable physical nodes and links in the physical network in order to map Virtual Network requests with certain constraints on virtual nodes and links. The goal of this paper is to design and implement substrate network resource selection scheme to increase the overall efficiency of the virtual network embedding process and satisfy the set of predefined resource constraints. This work assumes the existence of a Virtual Infrastructure Provider requesting virtual networks from Physical Infrastructure Providers and proposes a selection algorithm based on service-oriented architecture. Our proposed virtual network embedding algorithm is a heuristic algorithm that considers online requests and static attributes along with dynamic attributes of nodes and links as well as end-to-end QoS constraints.

***Keywords*:** Virtual networks, resource selection, virtual network embedding, service-oriented solution, QoS.

1. Introduction

In short period of time, the Internet has gained a phenomenal success and it became a critical infrastructure for accessing information and global commerce. The Internet has established a new way of communication by providing new ways of connecting people but its stunning success caused new limitations. The Internet consists of various network infrastructure providers with different objectives which makes emerging of new technologies or major architectural changes that require cooperative agreements, relatively impractical (Anderson et al., 2005). While the current Internet architecture is not suitable for supporting many types of applications, network virtualization is considered as promising, yet challenging solution of these limitations. Network virtualization separates the role of traditional internet service providers (ISPs) into physical infrastructure providers (PIPs),responsible for deploying the physical infrastructure and service providers (SPs) offering end-to-end services such as layer 3 VPNs and VOIP to end users. Another motivation for network virtualization is the possibility to add value in the virtualization layer aiming to make use of new technologies (e.g. QoS schemes) and customizing existing technologies to adapt specific services (i.e. customizable networks). This provides the means to run multiple virtual networks on a shared substrate network simultaneously while each virtual network is customized for a specific use. (El Barachi et al., 2010).

Previous studies in virtual network embedding were limiting the problem scope in different ways. Some assumed that all the requests are known in advance (Andreas Fischer, 2013), some works don't consider the node constraints (I. Houidi, 2008) and admission control (Andreas Fischer, 2013)(Lu, 2006) or they only consider a specific topology (Andreas Fischer, 2013). The virtual network embedding approaches can be categorized to static or dynamic approach. In static approaches, once the VN request has been embedded onto the substrate network, the allocated physical resources will be fixed regardless of the actual usage during the lifetime of the VN. In the dynamic approach, the VN requirements are dynamically adjusted to adapt to the probable changes.

We have previously proposed a service-oriented hierarchical network virtualization architecture, which promotes the idea of "network as a service" by defining different levels of services to which networking resources are mapped (El Barachi, 2012). In this architecture, five business roles are introduced: 1) *The Physical Infrastructure Provider (PIP)* that owns and manages a physical network infrastructure and can partition its resources using some virtualization technology. The services offered by the PIP are essential bearer services. 2) *The Service Provider (SP)* that has a business agreement with the subscriber and offers value added services, which could be simple or composite (i.e. formed by combining service building blocks); 3) *The Virtual Infrastructure Provider (VIP)* that finds, negotiates, leases, and aggregates virtual resources from one or more PIPs, deploys any protocols/technologies in the instantiated VNet, and operates it as a native network. The VIP supports SPs or other VIPs with service enablers and service building blocks and has no direct business agreement with consumers; 4) *The Consumer* who acts as the subscriber and the end user of value added services; and 5) *The Services and Resources Registry (SRR)* acting as information broker by providing information to find other parties and the services/resources they offer.

Building on this architecture, in this paper we propose an end-to-end virtual resources' selection and mapping algorithm that takes into consideration QoS requirements/constraints, during the selection of virtual resources. One of the contributions of our work is that a new class of global constraints has been proposed for the virtual network mapping problem. The other is dealing with the online VN embedding problem using admission control. We assume that the requests are unknown in advance and arrive using a certain probability, and stay in the network for certain amount of time. Besides, we propose a node power-degree in our algorithm and consider the capacity of the substrate nodes in order to make the best use of the substrate network. We have also taken into consideration the end-to-end QoS attributes in the network, and perform the path selection based on the paths with highest ranks.

2. Virtual Network Embedding Model

Our proposed selection and embedding algorithm for virtual networking environments is based on the service-oriented network virtualization architecture presented in (El Barachi et al., 2012) and is a heuristic algorithm which considers online requests and static attributes along with dynamic attributes of nodes and links and end-to-end QoS constraints. The resource broker proposed as part of this architecture allows the different roles to publish their resources and discover other role's resources. The broker has to make use of an efficient mapping algorithm in order to discover the most suitable resources based on properties requested in the VN request and select the resources accordingly. The selection algorithm is used in two levels within the network virtualization hierarchy. At the first hierarchical level, the VIP wishes to instantiate the VNet and send the request to the broker and the broker who has the information of the PIPs selects the best resources using the algorithm. At the second level, the SP wishes to build VNet and use the service building blocks offer by the VIP. In this level the QoS attributes should be considered. For the second level, the algorithm ranks the services based on their QoS parameters and the selection is performed depending on the type of the requested service. For example some services need low delay while some services require high throughput. The flowchart of the algorithm is given in Fig.1.

1. Input set for virtual network mapping algorithm
2. Matching step which considers functional attributes of nodes and links
3. Filtering which considers non-functional attributes of nodes an links
4. Aggregation algorithm in which the link mapping occurs using K-shortest path and the characteristics of the paths are computed using specified function of the application
5. Ranking is related to the service the virtual network has to offer and the requests are ranked based on the service they reference in their requirements.

6. Selection which is the final stage of the algorithm and the selected substrate nodes and links will be identified.

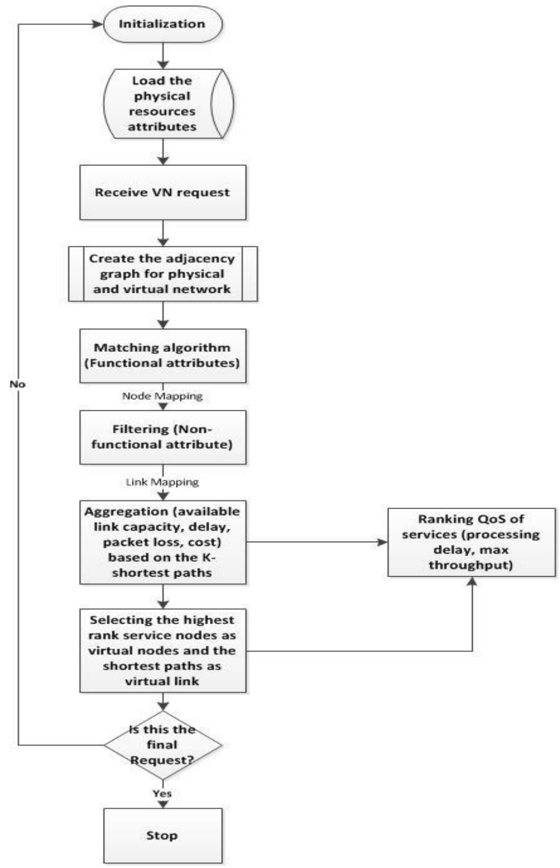

Fig. 1. Flowchart of A QoS-based Resource Selection Approach for Virtual Networks.

The algorithm in this work models requests dynamically arriving for virtual network instantiation on the substrate network. Each request specifies the topology of the virtual network, along with functional and non-functional attributes as well as the QoS requirements which includes time limit to instantiate it.

2. 1. Substrate Network Model

The substrate network is represented by an undirected graph denoted by $G^s(N^s, L^s, C_s{}^n, C_s{}^l)$ in which the substrate nodes (NS) are modelled as the vertices of the graph and the substrate links (L^s) as the edges. Each node in the substrate graph has attributes denoted as $C_s{}^n$ as follows: CPU $C(n^s)$ which denotes the available capacity of the physical node ns, location $loc(n^s)$ which is a coordinate of the physical nodes. The link attributes denoted as $C_s{}^l$ are as following: bandwidth $b(l^s)$ which denotes the available bandwidth capacity of the physical link ls, delay $d(l^s)$ and packet loss $pl(l^s)$ denote the delay and packet loss percentage of the physical link respectively.

Consider P^s a set of substrate paths in the substrate network G^s, the available bandwidth capacity AC(P) related to a substrate path P∈ PS between two substrate nodes is calculated as the minimum residual capacity of all the links in the substrate path:

$$AC(l^S) = b(l^s) - \sum_{l^v|L^s} b(l^v) \quad (1)$$

$$AC(P) = \min AC(l^S), \quad l^S \in P \quad (2)$$

2. 2. Virtual Network Model

The virtual network is also represented as $G^v(N^{V,} L^V, C_v^{\,n}, C_v^{\,l})$ in which virtual nodes are presented as N^V and virtual links as L^V .Each node in the VN requests is associated with certain characteristics as follows: CPU $C(n^V)$, Bandwidth $b(l^V)$, the maximum delay allowed in the virtual links $d(l^V)$ and location of each node $loc(n^V)$. All the node attributes and link attributes in the virtual network request are denoted as $C_v^{\,n}$, $C_v^{\,l}$ respectively.

Location in our model can be considered as specific policy for VIPs that may require the physical node to be in specified area. For this element we have taken into consideration the location of physical nodes and the desired location of the virtual nodes. We define a *Distance* denoted by D^V in the request which specifies how far a virtual node can be located from the given location. Each virtual link $l^v \in L^V$, is associated with bandwidth $b(l^V)$, delay $d(l^V)$ and packet loss $pl(l^v)$.

2. 3. Virtual Network Embedding Model

Virtual network embedding is defined as a mapping denoting by *MAP* from G^v to a subset of G^s such that the constraints in G^v are covered. *MAP*: $(G^v, C_v^{\,n}, C_v^{\,l}) \rightarrow (G^s, C_s^{\,n}, C_s^{\,l})$

The mapping consists of node mapping and link mapping stage as following:

Node mapping: MAP^N: $(N^v, C_v^{\,n}) \rightarrow (N^s, C_s^{\,n})$
Link mapping: MAP^L: $(L^v, C_v^{\,l}) \rightarrow (L^s, C_s^{\,l})$

The left side of Fig. 2 shows an example of the VN request and the right side of it shows the VN mapping solution. As the picture shows the nodes of the request are mapped to substrate nodes 1, 3 and 5. The virtual links (a, b) are mapped to the physical paths (1, 5) and (a,c) are mapped to physical path (1, 2, 3) and (b, c) are mapped to (3, 4, 5) with respect to CPU and bandwidth.

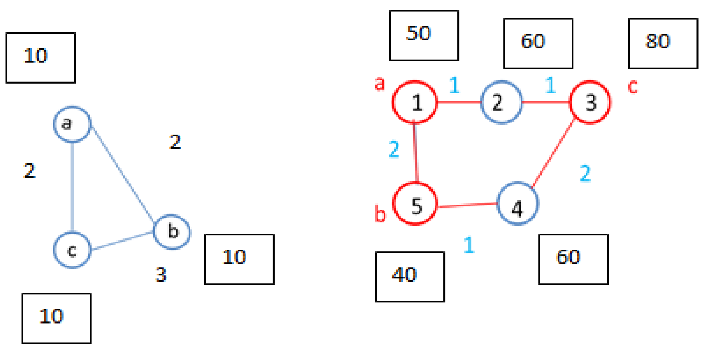

Fig. 2. Example of virtual network mapping

2. 4. Objective Description

The mapping algorithm may have multiple objectives. In this paper, our objective is to find a set of substrate nodes subject to the requirements in the VN request so that the total acceptance ratio and the revenue are maximized and the cost of the mapping is minimized. We have also taken the runtime into account and it is used in evaluation section to estimate the algorithm's efficiency.

3. End-To-End Virtual Resources' Selection And Mapping Algorithm

Our proposed mapping algorithm consists of five different steps and in each step certain criteria have been considered. The five steps are as following: Matching, Filtering, Aggregation, Ranking and Selection that will be explained in detail in next section. Applications that the quality of the end-to-end services is important can benefit from the proposed algorithm such as customizable networks, layer 3 VPNs and VoIP.

3. 1. Matching

Candidate matching is finding a set of available substrate resources based on functional attributes that comply with the requirements specified by the VN request. In a VN request the virtual links and nodes topology and their requirements are given specifically**.** A virtual node could possibly be a virtual host or a virtual router/switch. A virtual link can map over multiple physical links forming a physical path. A virtual link is usually mapped to multiple physical paths so it can satisfy some of the constraints of the virtual link such as bandwidth constraints that cannot be satisfied using a single path (Munstasir Raihan Rahman, June 2013)**.** In this phase only functional attributes which include typically the static parameters like operating system type, network stack, node/link type and virtualization platform are considered. The non-functional attributes which include real-time parameters (e.g. processing power, memory, and actual capacity) will be considered in the next steps of embedding.

In the first step of the algorithm, we partition and organize the resource descriptions. We have divided the resource matching part into two sections, *substrate resource adder* and *substrate resource finder*. The substrate resource adder builds a tree based on the resource description by the PIP in such an order so it is known that which specification is being dealt in each layer of the tree. We can easily add or remove resources whenever the PIP add or remove its resources. First we need to import the PIP resource specifications in the framework and then search through the data to select the most appropriate ones.

Upon receiving a request the substrate resource finder search the tree based on the requested attributes. Then in each children of the tree it finds the nodes with the required specifications and stores the potential candidates. At the end, the intersection of all the potential candidates with required specifications is our output for this phase. The VIP uses this framework to discover and match available resources using the VN request.

A. The Matching Algorithm (substrate resource adder)

1. Create the structure array with the specified resources and fill it according to their functional attribute as shown in Fig.3.

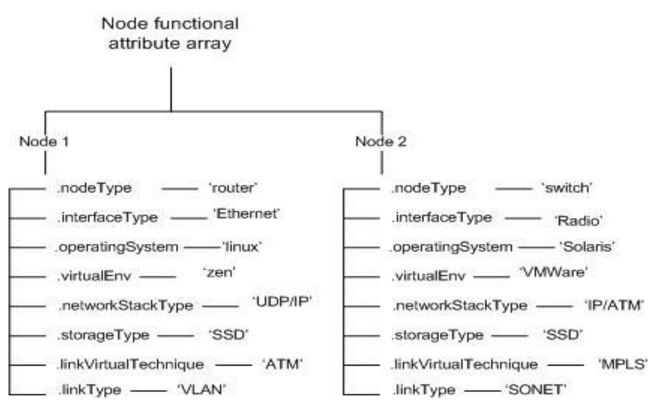

Fig. 3. Substrate resource adder in matching step

2. At any time that a resource has been added to the 'service and registry' we have to add it so it will be added among the resources for the coming virtual network request.

B. The Matching Algorithm (virtual resource finder)

1. Upon receiving the request, Virtual Resource Finder start searching through the structure array and for each specification it stores the candidate resources.
2. After searching the whole attributes, the intersection of all the candidate resources are the resources that have all the specifications that has been requested.
3. The output is all the resources with the requested functional attributes.

For our model we consider the following functional attributes for node. Node type: virtual router, virtual switch, virtual access point, virtual base station, interface type: Ethernet, Optical/Fiber, Radio(Wifi, WiMax), ATM, frame relay, operating system :linux, windows, solaris, UNIX, virtual environment type: VMWare, ZEn, KVM, network stack type: TCL/IP, UDP/IP, IP/ATM, IP/Ethernet, storage type: SSD , HDD. The link functional attributes are as follows: link virtual technique: ATM, MPLS, Ethernet 802.1g, VLAN, link type: VLAN, SONET, 802.11.

3. 2. Filtering

When a virtual network request arrives, the substrate network has to determine whether to accept the request or not. If the request is accepted, then the substrate network selects the suitable resources for the VN and allocates network resources on the selected substrate nodes and paths. Substrate network resources need to be measured and compared with the VN requirements.

Generally the physical nodes are used incompletely by some of the virtual requests and it is essential to calculate the remaining capacities of the substrate resources. AC (n^s) denotes the available CPU capacity of the substrate node ns ∈ Ns

$$AC\ (n^S)= C(n^s) - \sum_{n^v|n^s} c(n^v) \tag{3}$$

Where $n^V|n^S$ denoted that virtual node n^V is hosted on the substrate node n^S.

The aim of node mapping is to select the node which has more collective bandwidth of outgoing links among multiple similar nodes (in terms of resource availability). The amount of available resource for substrate node n^s is defined by:

$$AR(n^s)= AC\ (n^S) \sum_{l^s \in L(n^s)} AC(l^s) \tag{4}$$

Where $L(n^S)$ is the set of all adjacent substrate links of ns, $AC(n^S)$ is the available capacity of ns and $AC(l^S)$ is the available BW resource for the substrate link l^S. Available capacity of a substrate link lS ∈ LS could be derived from equation (5)

$$AC(l^S)= b(l^s) - \sum_{l^v|L^s} b(l^v) \tag{5}$$

Where $l^v|L^s$ denoted that virtual link l^V is hosted on the substrate link l^S. The available bandwidth capacity of a substrate path P∈ PS is

$$AC(P)= \min AC(l^S)\ ,\ l^S \in P \tag{6}$$

The VN assignment has been decomposed into two major components node mapping and link mapping that will be explained in the next part.

A. Node mapping

In node mapping each virtual node from the VN request should map to a different substrate node In other words, it's not possible to map two virtual nodes on the same physical node. The constraints for node mapping are given in the following equations:

$$C(n^V) =< AC\ (MN(n^V)) \tag{7}$$

$$dis(loc(n^V), loc(MN(n^V))) =< D^v \tag{8}$$

Where dis(x,y) measures the distance between the location of two substrate nodes x and y.

Algorithm 1: Virtual Network Node Mapping:

1. *Upon receiving a VN request* // Read the VN request attributes and substrate network attribute from database
2. *Extract* C_v^n, C_v^l and D^V *from the VN request*
3. **While** $N^v(i) \neq 0$ **do** // while there are non-assigned nodes
4. $\forall\, n^v \in N^v(i)$ **do**
5. j=1
6. **Sort** ($N^s(j)$) *// sort the selected nodes from matching algorithm according to node available resource (4)*
7. **If** $C(n^V) =< AC\,(C(n^s(j)))$ & $dis(loc(n^V), loc(M_N(n^V))) =< D^V$
8. $\forall\, n^s \in N^s(j)$
9. K=0
10. **While** $(AC(C(n^s)) >= C(n^V))$
11. K=K+1
12. **End while**
13. $n^v \rightarrow n^s$ // n^v assign to k-th n^s
14. $AC\,(n^S) = C(nS) - \sum_{nV|nS} c(nV)$
15. $AR(n^s) = AC\,(n^S) \sum_{lS \in L(nS)} AC(l^s)$*// 14 and 15 update the available capacity and resource for the node ns*
16. $N^v(i) = N^v(i) - n^v$ *// remove the node nv from list of requested nodes*
17. **Else**
18. j= j+1 *// go to the next node*
19. **go to** *6*
20. **end if**
21. **end while**

3. 3. Aggregation

The next step is to map an edge ($l^V \in L^V$) on a substrate path (P^S) containing one or more than one links. For link mapping we have used k-shortest paths algorithm for finding suitable edges.

A. Link mapping

In edge mapping each virtual link is mapped to a substrate path (or multiple paths). The substrate paths are those that connect the substrate nodes that have been preselected to host the virtual nodes. The constraint for link mapping is given by the following equation:

$$AC(P^s) >= b(l^v) \qquad (9)$$

In Aggregation, edge mapping solution assigns all edges to the substrate paths and each virtual link can be mapped on one or more physical paths. Edge mapping starts by finding edge disjoint k-shortest path for each edge of the VN graph. Afterwards, resources as well as QoS attributes of links (delay, packet loss) on each of these paths are calculated and the paths having sufficient resources (satisfying the QoS according to the service in demand) are selected. At this point the algorithm calls the ranking algorithm and finally it selects the path which has the highest rank. Ranking is based on end-to-end service level attributes (max throughput, end-to-end delay) if there are sufficient paths resources to satisfy the request of all the edges then the VN is completely mapped and the request is satisfied.

Algorithm 2: Virtual Network Edge Mapping:
1. Take the request which has successfully passed the virtual node mapping stage. 2. Create the adjacency cost matrix for the substrate network graph 3. Search the k-shortest paths between the pair of nodes connected by the edge (mapped on nodes by the node mapping algorithm). The output of this algorithm is 4. a. [shortestpaths]: the list of K shortest paths 5. b. [totalcost]: cost of the K shortest paths 6. After obtaining the k shortest paths, the links of these paths along with their attributes are extracted from substrate network. 7. For all the links of the paths calculate the available BW and delay and packet loss 8. If AC(Pi S) >= b(l^v) & ∑ d(li S) <= d(l^v) & ∑pl (li S) <=pl (l^v) 9. Select the ith shortest path which satisfies the edge demand (according to the edge mapping function, SPM). 10. Call the Rank Function which ranks the paths based on end-to-end service level attribute (packet loss, end-to-end delay) and select the paths with the highest ranks 11. If the edge request is satisfied, update available link capacities of the selected path according to (6) and GOTO 12 12. If this was the last request, then, stop, else, call "node mapping algorithm".

3. 4. Ranking

In our work, we don't monitor the network to measure the network parameters. We assume that the application providing QoS will require a performance monitoring and that the data acquired by such monitoring system could be used in simulation of the algorithm.

We propose the ranking for end-to-end QoS attributes for our algorithm. Our purpose is to select paths with high quality; therefore, we assign related weight to the QoS parameters according to the requested service. After selecting the preliminary nodes and links we calculate the ranks for the links in instantiated virtual network and choose the ones with the highest rank to proceed with the embedding. We consider max throughput, end-to-end delay as the parameters in our simulation but here we give a general equation for the ranking concept based on (Jawwad Shamsi).

An application with 'n' QoS constraints is represented as $P_1, P_2, \ldots, P_n$, and each is assigned to a specific weight, $w_1, w_2, .., w_n$ according to the importance of the constraints. Note that the addition of all the weights should be equal to 1. ($w_1+w_2+\ldots+w_n=1$)

By assigning weights to the end-to-end attributes, the intention is that each attribute related to the constraints contributes to the quality of the path in proportion to its weight. For example VoIP and online gaming require low end-to-end delay while bulk file transfer require high throughput. According to the constraints (P_i) and the requested value (R_i) the equation 10 shows the quality of the path (Q).

$$Q=\sum_{i=1}^{i=n}\frac{Pi}{Ri}\,w_i \tag{10}$$

Q represents the ratio of the QoS attributes received by an application over the QoS attributes requested. The purpose of the ranking stage is to ensure that the algorithm favors high values of the metrics that are more important for the requested application. For example minimum end-to-end delay is the most important QoS attribute for VoIP applications and should acquire the highest weight among other attributes.

3. 5. End-to-End Algorithm

Given the detailed description of each stage of the algorithm, here are the main steps of the algorithm:

Algorithm 3: End-to-end selection and mapping algorithm:

1. Match all the functional attributes of the VN nodes and links and specify the candidates.
2. If no candidate is selected reject the request, go to step11
3. Filter all the nodes and links from the substrate topology that meet the application non-functional constraints
4. If there is no node reject the request and go to step11
5. Store the nodes and links of substrate nodes that fulfill the requirements
6. From the VN request, select a node to perform the embedding
7. Find the shortest edges using k-shortest path among nodes discovered in step5 and compute the path characteristics using the aggregate functions
8. If no link exists reject the request and go to step 11
9. Compute the quality of the k-shortest path using equation (10) and based on the quality of the direct routes rank all the existing paths
10. Select the nodes with highest degree (if multiple nodes exist) and the path with highest rank in terms of QoS attributes.
11. If there is more request, go to step1, otherwise go to step 12
12. If no request left, stop and return the selected nodes and links.

4. Conclusion

Virtual resources' selection is one of the key research issues in network virtualization and utilizing the substrate resources in the most efficient way is the main challenge. The desired approach should consider resources' dynamic attributes along with their static characteristics and the online VN requests. In this work we concentrated on these characteristics and also take into consideration the QoS constraints in order to be able to select the best PIP to satisfy a VN request.

References

Ammar, Y. a. (2006). Algorithms for Assigning Substrate Network Resources to Virtual. IEEE INFOCOM.

Andreas Fischer, J. F. (2013). Virtual Network Embedding: A Survey. *IEEE Communications Surveys.*

I. Houidi, W. L. (2008). A distributed virtual network mapping. IEEE ICC.

Jawwad Shamsi, M. B. (n.d.). QoSMap: QoS aware Mapping of Virtual Networks for Resiliency and Enfficiency.

Jim'enez, J. C. (n.d.). Network virtualization: a view from the bottom. In Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architecture.

Lu, J. a. (2006). Efficient mapping of virtual networks onto a shared substrate. Washington University in St. Louis, Tech. Rep.

May El Barachi, N. K. (2012). Open Virtual Playground: Initial Architecture and Results. *in Procedding of the 9th IEEE Consumer Comminications and Networking Conference.*

May El Barachi, N. K. (Dec. 2010). Towards a service-oriented network virtualization architexture. *3rd ITU-T Kaleidoscope Event 2010*, (pp. 1-7).

Munstasir Raihan Rahman, R. B. (June 2013). Survivable Virtual Network Embedding. IEEE Transactions on network and service management.

Thomas Anderson, L. P. (2005). Overcoming the Internet Impass through Virtualization. 31-41.