Comparison of D-Vine and R-Vine Techniques for Virtual Network Embedding Problem

Preksha Satish [1], Deeksha Lingraj [2], Anjan Kumar S[3], Keerthan Kumar T G [4]

[1][2][3] UG Scholar, [4] Assistant Professor
Department of Information Science and Engineering, Siddaganga Institute of Technology, Tumakuru
{[1]prekshas573, [2]deekshalingraj10, [3]anjansa35, keerthanswamy[4]}@gmail.com

Abstract. In Network virtualization, Virtual Network Embedding(VNE) is the process of mapping virtual nodes and links of a virtual network request(VNR) on a substrate network to fulfill the demands of the request. Embedding virtual network requests helps in achieving network virtualization efficiently. This paper presents Vineyard, a set of VN embedding algorithms namely D-Vine and R-Vine, to introduce a finer correlation between the node mapping and link mapping phases. Deterministic Virtual Network Embedding(D-Vine) algorithm is used to embed virtual nodes onto substrate network based on the capacity constraint when it is not satisfied the system takes the Randomised Virtual Network Embedding(R-Vine) algorithm to map the nodes based on location constraint. Subsequently, after the nodes are embedded the link mapping phase is done based on the distance constraint. A window-based virtual embedding algorithm(W-Vine) is also introduced to evaluate the effect of lookahead in Virtual Network Embedding. After the mapping of multiple virtual network requests, analysis is done to compare the node and CPU utilization in both the algorithms and the variations are conserved.

1. Introduction
In Cloud Computing, Network virtualization is a trending technology for the internet of the future. This situation is dealt with by using internet ossification where necessary and innovative solutions are used for sharing the physical network resources with proper accountability[1][2][13]. During Network Virtualization, the first object that is studied is the virtual network. The active and passive network components are mostly distributed over the substrate network. A virtual topology is formed by connecting all virtual nodes through virtual links[2][5]. It is the responsibility of the InP to operate and maintain the physical infrastructure whereas the SP is responsible for aggregating the resources from INPs to construct virtual networks and end-to-end services are provided to the users.[9] Introducing network virtualization decouples the management and service roles of SP by recognizing three distinct components: A Virtual Network Provider (VnP) gathers virtual resources from an InP or several InPs, Virtual Network Operators[10] (VNOs) implement, control, and operate the VNs based on the requirements from the SP, Service Providers are freed from management and can focus on business by using the VNs for providing custom services[2][3][4].
In the environment of network virtualization, multiple service providers can form different virtual networks to provide end-to-end services to end-users by subletting shared resources from one or more infrastructure providers without committing to physical infrastructure\[1\][4]. Substrate nodes and Virtual nodes exist within the NVE\[7\]. The virtual nodes are embedded on the different substrate nodes and are mutually connected by the lane in the substrate networks that correspond to the virtual links, this is called the Virtual Network embedding problem\[2\][6]. There is a possibility that more virtual nodes might share the same substrate node resources, ensuring efficient request mapping will increase substrate node utilization. For a successful and efficient mapping, the embedding problem for Virtual Network Embedding must be considered an important challenge\[5\][6][11][12]. After the successful mapping of virtual nodes, it is necessary to check whether virtual nodes correspond to substrate nodes based on the location and capacity constraints\[2\][6]. The overall system aims to develop and allocate the resources to the VN that satisfies the conditions of the substrate network and perhaps increases the substrate node utilization and also scheduling of requests is done so that upcoming requests are set in a queue for the allocation purpose.

2. Literature Survey

Virtual network embedding is the procedure of embedding virtual network requests onto substrate networks to satisfy the virtual network’s requirements. It is common for virtual network embedding algorithms to lack coordination between the node and link mapping and delay for packet transfer. In this context, various research papers focused on existing virtual network embedding algorithms and how to address the shortcomings of existing algorithms. During the study phase, however, it was found that a wide range of embedding algorithms was based on a variety of factors, including resource demands, cost considerations, and the provider's revenue.

In the paper\[2\], the author describes how the mapping of virtual nodes on substrate nodes is done using exact VNE algorithms. When used in small-scale scenarios, it provides near-optimal embedding of each Virtual Network request. As a consequence, these algorithms cannot be applied in continuous-time scenarios due to their huge computational complexity.

Paper\[4\] Author describes the mapping of virtual nodes using the Subgraph isomerism detection problem. An isomeric subgraph is found that accomplishes the request within the substrate network and using a hop-limit constraint applies to substrate paths used to map the request. As a result, this leads to getting stuck in local optimum positions far from the actual optimum.

In paper\[5\], the author describes that the embedding of virtual network requests is done only when the virtual network requests are static. Virtual network demands and resources are matched with the substrate networks and embedding is done. Virtual requests change dynamically; efficient allocation of resources cannot be done when the system is static.

In paper\[6\], Node mapping and link mapping are done using the uncoordinated VNE technique MCMCF provides a multipath routing solution for each virtual link based on optimal linear programming. The problem with this technique is that it drives up the price of single or multiple paths in the virtual link mapping phase, which leads to a low acceptance rate and low long-term revenue.

In paper\[8\], the Novel node ranking approach is used to rank all the substrate nodes and virtual nodes before embedding each virtual network request. Virtual link propagation delay is considered a link constraint for embedding. This algorithm uses an iterative approach to rank all the nodes that can be further improvised.

In paper\[12\], Linear programming derived from Mixed Integer programming is used to select the virtual nodes for embedding. The Markov Chain approach is used to select the candidate substrate nodes and complete the link configuration. This approach is only leveraged to splittable link mapping.
3. Experimental

A Virtual Network Embedding problem involves mapping virtual network requests, which have different conditions on virtual nodes and links, onto particular physical nodes and links within a substrate network that has countable resources. Figure 1 shows the flowchart of the proposed system. The vineyard algorithms, which contain Deterministic Virtual Network Embedding (D-Vine) and Randomized Virtual Network Embedding (R-Vine) are proposed, to contribute a better correlation between the node and link mapping. Virtual nodes are mapped against substrate nodes in a form that promotes the mapping of virtual links to physical paths during the succeeding phase. Augmented substrate graph construction is done where the cluster is formed for each virtual node, finding out the possibility of the number of substrate nodes for the embedding. Once the location constraints are satisfied for the substrate and the virtual nodes, only those nodes are moved ahead in the embedding process; the rest of the substrate nodes are considered for the next request. Deterministic and Randomized Rounding techniques are used for node and link mapping purposes where the capacity constraint is the foremost constraint that has to be satisfied to embed the node, deterministic algorithm mainly concentrates on mapping of virtual nodes to substrate nodes based on the capacity constraints those nodes which fail in the D-vine follows R-vine where location constraint is considered for embedding the node. Once the node embedding is done, the next part is to link the physical paths to the virtual paths. With the help of Euclidean distance, the distance is calculated between the two nodes in the substrate network and the virtual network. Based on the distance constraint, the link mapping is done. After the embedding process, the analysis is done for the virtual requests over the time obtained by plotting the graph and the observations are made.
3.1 Construction of Substrate Network: The substrate network is implemented as a weighted bidirectional graph, $G^S = (N^S, E^S)$, $N^S$ indicates substrate nodes set, and $E^S$ indicates a substrate link set. On a graph, each and every substrate node $n^s \in N^S$ is allocated with CPU capacity weight values $c(n^s)$ as well as location $\text{loc}(n^s)$ on a general coordinate system. Every substrate link $e^S(i, j) \in E^S$ between two substrate nodes $i$ and $j$ are allocated with the distance $d(x, y)$ value obtained by using the euclidean distance.
3.2 Construction of Virtual Network Request: The Virtual Network Requests implemented as a weighted bidirectional graph with each virtual network request having the form $G^v = (N^v, E^v)$, $N^v$ indicates virtual nodes set, and $E^v$ indicates a virtual link set. Each Virtual network requests (VNR) is having a positive value $D^v$ that indicates the distance of virtual node $n^v$ $\in$ $N^v$ will be embedded according to its favored location $\text{loc}(n^v)$. A landmark-based approach is used in our model to request virtual nodes so Infrastructure providers do not have to expose their network structures.

3.3 Augmented Substrate Graph Construction: As soon as the virtual network request arrives, each virtual node must choose the substrate nodes on which embedding can be done. With augment substrate graph construction, clustered nodes are created based on virtual node location constraints. The cluster is formed if the substrate node location $x$ is less than the location of the virtual node, otherwise, the particular substrate node is rejected. Prior to actual embedding, augmented substrate graph construction focuses on forming the cluster that includes substrate nodes for each virtual node.

$$x_{\text{loc}}(N^S) < x_{\text{loc}}(n^v) \in N^v$$

(1)

According to equation (1), the x coordinate location of the substrate node should be lesser than the x coordinate location of the virtual node.

3.4 D-Vine and R-Vine for embedding: The D-Vine algorithm is implemented once the augmented substrate graph construction is done. As inputs, the algorithms take the augmented substrate graph and the virtual network request. Using the D-Vine algorithm, we check if the virtual node is possible to embed on the substrate node based on capacity constraints if the capacity of the substrate node is lesser than the virtual node, the embedding is not possible, or else if the capacity of the substrate node is greater than that of the virtual node, the difference is calculated and the node with the minimal residual capacity is allocated for the virtual node. Those nodes which are not embedded still undergo the R-Vine technique and allocation is done based on the location constraints. After the residual capacity is calculated, there are chances that a single substrate node can be embedded on the virtual node, in that case, D-Vine follows to consider the substrate node which is having minimum residual capacity, so that unique embedding of the virtual node on to the substrate node property is conserved. Once the node embedding is completed, the system follows in embedding the links. The distance between the two nodes is calculated by using the euclidean distance.

$$R_N(n^S) = x - \sum c(n^v)$$

(2)

Where $x \uparrow y$ indicates that the virtual node name $y$ is introduced on substrate node $x$

$$R_E(c) = \min R_N(n^S)$$

(3)

Equation (2) indicates that residual capacity is obtained by calculating the difference between the substrate and the virtual node.

Equation (3) shows that the node with minimum residual capacity is selected.
\[ \sum_{\omega \in \Omega(m)} x_{\omega m} = 1, \quad m \in N \setminus N^S \] (4)

Equation (4) states that each virtual node should be embedded on a single substrate node.

\[ \sum_{\omega \in \Omega(m)} x_{\omega m} \leq 1, \quad m \in N^S \] (5)

Equation (5) indicates that a unique substrate node should be selected for embedding.

\[ d(x, y) = \sum_{i=1}^{n} (y_i - x_i)^2 \] (6)

\[ \text{distance}(x, y) = \sqrt{d(x, y)} \] (7)

Equation (6) and equation (7) are used to calculate the euclidean distance.

\[ \text{distance}(x, y)(E^V) \leq \text{distance}(x, y)(E^S) \] (8)

Distance Equation (8) indicates that the distance of the virtual link should be less than or equal to the distance between 2 nodes of the substrate network.

3.5 VNE Embedding with lookahead: Virtual requests don't arrive in time and there may be a clash in the incoming multiple virtual requests. The proposed system handles the multiple virtual requests in an efficient way, once the first virtual request starts processing and during the execution of the request, if other requests start arriving then automatically the system calculates the maximum waiting period for the particular request and is stored in the batch process. In order to select the virtual requests out of the queue, a window period mechanism is used where only those requests which are having the maximum waiting period less than window period are selected for execution, if there are a number of requests satisfying the time constraints, then the requests are sent for execution using first come first serve basis technique, in this way all the virtual requests are handled in an efficient way.

\[ G^V = (N^V, E^V)(t_w) < (W_p) \] (9)

\[ t_w = \text{waiting period} \quad W_p = \text{window period} \]

Equation (9) indicates that the virtual network request waiting period must be less than the window period.

4. Results

Figure 2 shows the analysis of Node utilization using R-Vine and D-Vine technique, D-vine algorithm mainly concerned with the CPU resource utilization, only those nodes which satisfy the capacity constraint is allocated with the nodes, hence the percentage of allocation of nodes using D-vine is lesser when compared to the R-Vine technique. R-Vine technique follows checking of location constraint, therefore the possibility of allocating the node using R-Vine is more with respect to the D-vine, the above graph shows the variation in the percentage of node allocation using R-Vine and D-Vine with the number of substrate nodes. From the above graph, it is indicating that node utilization in R-vine is more when compared with the D-vine and through the variation, one can note down the observations that R-vine is independent of the capacity constraints, only D-vine considers capacity constraints, based on that allocation is done.
Figure 2 shows the CPU utilization using R-vine and D-vine. The analysis is done based on finding the CPU capacity wasted after the execution of both D-vine and R-vine. It is observed that D-vine focuses on CPU resource utilization and it allows for the embedding only if the nodes satisfy the capacity constraints. The total substrate node capacity is compared with some of the capacity of the nodes embedded using D-vine and R-Vine, the analysis shows that in D-vine there is lesser wastage of CPU capacity compared with R-vine, Therefore D-vine is more efficient than R-vine in CPU utilization process.

5. Conclusion
In the process of making network virtualization, an intrinsic entity of the upcoming generation of internet design, systematic and pragmatic algorithms for embedding the virtual networks are very much important. In this paper, the algorithms that are proposed perform efficient embedding and establish a step ahead correlation between the node and link mapping juncture. The correlation notably hikes the solution stretch and the endowment of heuristic algorithms. We started with setting up the network and then used the algorithms to accept multiple virtual network requests and embed them on the substrate networks considering the residual capacity, location, and distance constraints. A window-based mechanism is used to store the requests in a batch system and allow the requests which satisfy the time constraints one after the other for execution based on a first come first serve basis. Both algorithms are compared with respect to the node and CPU utilization and the analysis is drawn. Considering the node utilization inspection, it is observed that R-Vine performance is better than D-vine because of the fact that R-vine only focuses on the location constraint, and allocation of the node is done irrespective of the capacity of the nodes. With the help of CPU utilization analysis, it is observed that D-vine is more efficient than R-vine because the CPU capacity wasted is less when compared with the R-vine and it is important that lesser the CPU wastage more is the efficient binding of nodes and that increases resource utilization and the overall performance of the system is increased.

References
[1] Fischer, A.; Botero, J.F.; Till Beck, M.; de Meer, H.; Hesselbach, X., "Virtual Network Embedding: A Survey," in Communications Surveys & Tutorials, IEEE, vol.15, no.4, pp.1888-1906, Fourth Quarter 2013. DOI: 10.1109/SURV.2013.013013.00155.

[2] M. Chowdhury, M. Rahman and R. Boutaba, "Vineyard: Virtual network embedding algorithms with coordinated node and link mapping", IEEE/ACM Trans. Netw., vol. 20, no. 1, pp. 206-219, Feb. 2012.

[3] N. M. M. K. Chowdhury, "Network virtualization: State of the art and research challenges", IEEE Commun. Mag., vol. 47, no. 7, pp. 20-26, Jul. 2009.

[4] T. Anderson, L. Peterson, S. Shenker, and J. Turner, “Overcoming the Internet impasse through virtualization,” Computer, vol. 38, no. 4, pp.34–41, 2005.

[5] J. Turner and D. Taylor, “Diversifying the internet,” in Proc. IEEE GLOBECOM, 2005, vol. 2, pp. 755-760

[6] N. Feamster, L. Gao, and J. Rexford, “How to lease the Internet in your spare time,” ACM SIGCOMM Computer Communication Review, vol. 37, no. 1, pp. 61–64, 2007

[7] J. Lu and J. Turner, “Efficient mapping of virtual networks onto a shared substrate,” Washington University, Tech. Rep. WUCSE-2006-35, 2006.

[8] Cheng, X.; Su, S.; Zhang, Z.; Wang, H.; Yang, F.; Luo, Y.; Wang, J.,“Virtual network embedding through topology-aware node ranking,” SIGCOMM Computer Communication Review, vol. 41, pp 38–47, April 2011. DOI: 10.1145/1971162.1971168.

[9] Even, S.; Itai, A.; Shamir, A., "On the complexity of time table and multi-commodity flow problems," Foundations of Computer Science, 16th Annual Symposium on, pp.184-193, 13-15, October 1975. DOI: 10.1109/SFCS.1975.21.

[10] K. Tutschku, T. Zinner, A. Nakao, and P. Tran, "Network virtualization: Implementation steps towards the future internet", Proc. Workshop on Overlay and Network Virtualization at KiVS, march 2009.

[11] A. Haider, R. Potter and A. Nakao, "Challenges in resource allocation in network virtualization", 20th ITC Specialist Seminar, vol. 18, pp. 20, 2009.

[12] Lingnan Gao, George N Rouskas, “Virtual Network Reconfiguration with Load Balancing Migration Cost Considerations”, IEEE Infocom 2018.

[13] Keerthan Kumar T G, H. K. Virupakshiaah and Nanda K V, "Ensuring an online Chat Mechanism with accountability to sharing the non-downloadable file from the Cloud," 2016 2nd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), 2016, pp. 718-721, DOI: 10.1109/ICATCCT.2016.7912093