

# A Cluster-based Overlay Network Embedding using Ruckpack++ Algorithm in Distributed Environment

**Thiraviaselvi G**

Assistant Professor, Department of Information Technology, Francis Xavier Engineering College, Tirunelveli, Tamil Nadu, India.

Email: thiraviaselvi@gmail.com

**Gnana Thiraviam G**

UG Student, Department of Mechanical Engineering, Anna University Regional Campus, Tirunelveli, Tamil Nadu, India.

Email: thiraviam477@gmail.com

---

## ABSTRACT

Network Virtualization is one of the emerging fields of Research as it becomes pillar of the future Network. Here, Overlay Network Embedding (Virtual Network Embedding) is the major problem to be addressed to have optimized resource utilization. The main purpose of Overlay Network embedding is to efficiently utilize the physically networked resources to offer the supporting of virtual nodes. While most of the Overlay Network Embedding algorithms consider it as a mapping problem, Ruckpack++ algorithm uses cluster-based and greedy approach to select the optimized physical node. In the proposed algorithm, topology awareness is also included. Extensive simulations have shown that the proposed algorithm produces better Acceptance Ratio and node Utilization when compared to other state-of-the-art algorithms.

**Keywords - Distributed Computing, Network Virtualization, Overlay Network Embedding, Ring Topology**

---

Date of Submission: Sep 03, 2020

Date of Acceptance: Sep 20, 2020

---

## I. INTRODUCTION

Network Virtualization is the process of networking virtual resources deployed on networked physical resources by using encapsulated tunneling. This logical network topology is often referred to as an 'Overlay Network' [1]. Network virtualization is designed to allow network optimization of data transfer rates, flexibility, scalability, reliability and security. It automates many network administrative tasks, which disguise a network's true complexity. Optimization of Scalability is one of the challenging tasks in Network Virtualization. Scalability means, embedment of virtual networks on the given physical network based on the requirements of Network Requests called Overlay Network Requests (ONR).

In distributed computing, a single problem is divided into many parts, and each part is solved by different nodes. If the nodes are networked, then they can communicate with each other to solve the problem. If so, the nodes perform like a single entity. [2]

Ring Topology is very organized. Also, in ring topology all the traffic flows in only one direction at very high speed. Even when the load on the network increases, its performance is better than that of Bus topology. Each computer has equal access to resources [3].

To maximize the aggregate performance across the VNs, an algorithm called Ruckpack ++ is proposed to dynamically adapt ONRs for a customized physical network [6]. As it is known, single ONR could not be processed by a single virtual node, for that a network of virtual nodes are needed. So, once the ONR is received by

ONR repository, it is distributed among different virtual nodes.

Each physical node hosts several virtual nodes to form a cluster, in such a way; they are connected to each other in Ring Topology. Cluster Heads will communicate with the cluster head hosted in next physical node. As Ring Topology offers high bandwidth, Link Capacity is not considered, only Processing capacity for a time period is considered for implementing the algorithm.

Once the ONR is received, the repository Manager allocates virtual nodes in such a way the sum of Processing Capacity of virtual nodes for the requested time period is equivalent to the requested Processing Capacity of ONR. Fig.1 depicts the architecture of Overlay Network Deployment. In this architecture, each physical node hosts a cluster of virtual nodes forms a Ring topology.

Fig.1 depicts the architecture of Overlay Network Deployment. Here, requests that are stored in the ONR repository are handled by ONR Manager.

The functionality of ONR Manager is to allocate resources according to the received request's Processing Capacity for the requested service Time period. Once the service time gets over; the Processing Capacity of virtual nodes will be reallocated. The rest of the paper is organized as follows: Related work is discussed in Section II. In Section III, Problem Specification is discussed. In next section, the optimized network virtualization embedding problem for evolving networks is explained. The effectiveness of the Ruckpack++ algorithm is validated by simulations evaluation in Section V. Finally, conclusions are presented in the last section.

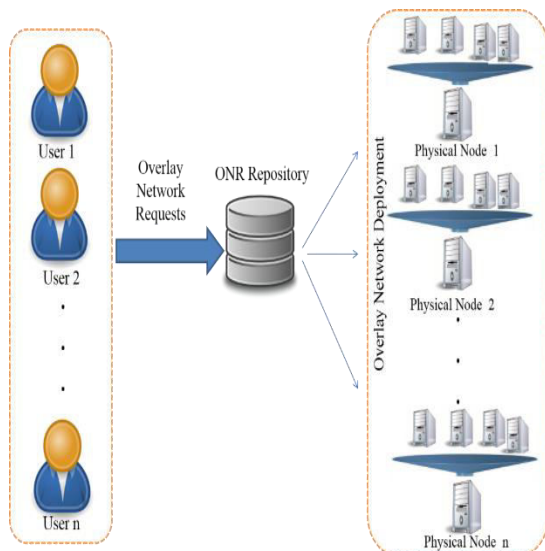


Fig.1: Architecture of Overlay Network Deployment

## II. RELATED WORK

There are lot more of research works have been done in the field of virtual network embedding. Each study focuses on different perspectives based on the problem formulation.

In [4] CPU capacity for node attributes, and link delay for link attributes are considered for optimal virtual network embedding using Integer Programming. The problem of opportunistically sharing substrate resources among multiple virtual networks is studied in [5].

The algorithm coarsens VNs, uses Heavy Edge Matching (HEM) technique to minimize its size and to minimize the total required bandwidth before embedding them on best-fit sub-substrate networks is implemented in [6]. A survey [7], [8] made on Virtual Network Embedding elaborates the properties of static vs dynamic, Distributed Vs Centralized, changes on substrate Network, changes on Virtual Network Elements, etc.

The objective of the virtual network mapping is to obtain the maximum benefit of the virtual network, which is based on the premise of satisfying the conditions, thus formulated and solved as constraint satisfaction problem in [9].

A fast-convergent discrete Particle Swarm Optimization algorithm is devised in [10] whereas Integer Linear Programming is formulated when path splitting is not allowed. Topological attributes of substrate and virtual networks through multiple characteristics, demonstrate that our algorithms improve the long-term average revenue, acceptance ratio, and revenue/cost ratio and study better coordinates node and link embedding in [11].

The employment of physical topologies that contain regions with high connectivity significantly contributes to the reduction of rejection rates and improved resource usage is demonstrated in [12]. Additionally, through an extensive analysis of denied requests, the main rejection causes related to both ISP and data center networks is provided with strong evidence.

For an ISP it is important to know how many additional Virtual Networks (VNs) of a specific

application (e.g. web, streaming, P2P, and VoIP) are map able into the current resource substrate with a certain probability. In [13] this probability is calculated to consider side effects based on remapping of VNs (e.g. due to reduced link delay).

## III. PROBLEM SPECIFICATION

This section introduces the system model including physical network and overlay network and specify the general characteristics of virtual network embedding problem. In the last subsection, evaluation index of virtual network embedding is simulated and measured.

### 1) Physical Network

The physical network is drawn as undirected graph and it is denoted as  $G_p = (N_p, L_p, P_n, P_l, S_p)$  in which  $N_p$  stands for the collection of physical nodes,  $L_p$  stands for the collection of physical links,  $P_n$  stands for the properties set of physical nodes resources, such as their locations, the speed of switching, the storage capacity and the Processing capacity and so forth,  $P_l$  stands for the properties set of physical links resources, such as the propagation delay time, the available bandwidth and so on and  $S_p$  stands for the profit gain by the node  $N_p$ .

### 2) Overlay Network

The overlay network is drawn as undirected graph and it is denoted as  $G_o = (N_o, L_o, P_{no}, P_{lo})$  in which  $N_o$  stands for the collection of virtual nodes,  $L_o$  stands for the collection of Virtual links,  $P_{no}$  stands for the properties set of virtual node resources, such as their locations, the speed of switching, the storage capacity and the Processing capacity and so forth,  $P_{lo}$  stands for the properties set of virtual link resources, such as the propagation delay time, the available bandwidth and so on.

### 3) Overlay Network Request

Overlay Network Request (ONR) is represented as  $T_{ONR} = (A_t, P_c, P_r, P_t)$  in which  $A_t$  stands for Arrival time of ONR,  $P_c$  stands for Processing Capacity of Virtual Node  $N_o$ ,  $P_r$  stands for Precedence of the ONR, that depends upon the Application for which ONR is received (E.g: Online Streaming has higher precedence than Printer request) and  $P_t$  denotes the Service time (amount of time the particular virtual node is in use).

### 4) The problem of Overlay Network Embedding

ONR Manager handles the  $T_{ONR}$  tuple stored in the ONR repository, at first by sorting the tuples in the non-increasing order based on the precedence of the application by Algorithm1.

---

**Algorithm1: Sorting Algorithm for ONRs**

---

```

1: Choose zeroth attribute of ONR as Arrival Time (At)
   from ONR tuple
2: Choose second attribute of ONR as Precedence from
   ONR tuple
3: for each ONR
4:   if arrival time (At) of successive ONRs are equal do
5:     Sort the ONRs by Precedence (Pr) in non-
       increasing order
6:   end if
7: end for
8: Return the sequence of ordered ONRs

```

---

As it is a cluster-based algorithm, the virtual nodes hosted on a physical node forms a cluster with leading virtual Node as Cluster Head. From this scenario, it is understood, each physical node hosts one cluster Head. According to [14], clusters help to perform computation simultaneously with better outcome.

Obviously, the number of Cluster Heads and the number of physical nodes is equal. As they are connected in bus topology, Cluster Heads can communicate with each other. Cluster Head in turn, can communicate with other nodes easily as they are connected in Ring Topology. Ring Topology is the main reason for excluding link capacity as it offers high bandwidth and it is explained in Algorithm2.

---

**Algorithm2: Formation of Cluster Head among Virtual Nodes**

---

```

1: for each physical node (Np)
2:   Host five virtual nodes (No) each of different
       capacity with their sum equivalent to the
       capacity of underlying physical node (Pc ∈
       Np)
3:   Choose the leading virtual node as Cluster
       Head
4:   Arrange rest of the virtual nodes to be in Ring
       Topology with the Cluster Head
5: end for
6: return a Cluster Head from each physical node
   (Np).

```

---

After Cluster formation, the next module is about allocation of virtual nodes (No) for the current ONR request based on the Processing Capacity requested.

Allocation starts from Cluster Heads followed by Cluster Members, if remaining capacity is to be allocated even after the completion of one Cluster, it can be extended to next Cluster and so on. Once allocated, the allocated virtual nodes (No) cannot accommodate other requests, since their Processing Capacities (Pc) are nullified.

---

**Algorithm3: Allocation of Processing Capacity**

---

```

1: for each sorted ONR
2:   choose first attribute as Processing Capacity (Pc)
   needed
3:   choose third attribute as Service time (Pt) needed
4:   for each Processing Capacity (Pc) of ONR
5:     Allocate virtual nodes whose sum surpassed by
       Pc of ONR starting from Cluster Head
6:     if Pc of ONR exceeds sum of Pc of one Cluster
       do
7:       complete current cluster and extend allocation
           starting from Next Cluster Head until
           Pc of ONR gets over
8:     end if
9:     nullify the Pc of each Virtual node once allocated
10:  end for
11: end for

```

---

ONR Manager will wait for the completion of given Service Time (Pt) for the requested ONR. Once it gets over, the Processing Capacity (Pc) of allocated virtual nodes (No) will be reallocated and that can be used for other ONRs. The above process is explained in Algorithm 3&4.

---

**Algorithm4: Reallocation of Processing Capacity (Pc)**

---

```

1: for each sorted ONR
2:   find the difference between successive ONR's
       arrival time (At)
3:   for each nullified virtual node
4:     if service time is equal to arrival time difference
       do
5:       Reallocate the virtual nodes with its original
           Processing Capacity (Pc ∈ No)
6:     end if
7:   end for

```

---

As optimization technique are used [15] in various fields of computation, Multiple Knapsack Problem is utilized in Ruckpack++ algorithm. Multiple Knapsack Problem (MKP) is a classical variation of KP. In MKP, there are set of knapsacks  $M:=\{1,...,i,...,m\}$  each with positive capacities  $b_i$ , and a set of items  $N:=\{1,...,j,...,n\}$  each with size  $s_j \geq 0$  and profit  $w_j \geq 0$ . The goal is to find a subset of the  $n$  items of maximum profit which can be packed into the  $m$  knapsacks. The problem formulation can be written as

$$\text{Maximize } \sum_{i=1}^m \sum_{j=1}^n w_j x_{ij} \quad (1)$$

Subject to Eqs. (2)-(4):

$$\sum_{i=1}^m s_j x_{ij} \leq b_i \quad 1 \leq j \leq n \quad (2)$$

$$\sum_{i=1}^m x_{ij} \leq 1 \quad 1 \leq j \leq n \quad (3)$$

$$x_{ij} \in \{0, 1\}, 1 \leq i \leq m, 1 \leq j \leq n \quad (4)$$

where variable  $x_{ij} = 1$  if item  $j$  is packed into knapsack  $i$  and zero otherwise. Algorithm5 combines all the modules into a single algorithm called Ruckpack++.

---

**Algorithm5: Ruckpack++**

---

- 1: Choose zeroth attribute of ONR as Arrival Time (At) from ONR tuple
  - 2: Choose second attribute of ONR as Precedence from ONR tuple
  - 3: sort the ONR by Algorithm1
  - 4: Create Cluster Head by Algorithm2
  - 5: **for** each ONR
  - 6:     Select the Physical Node ( $N_p$ ) with maximum Profit gain  $S_p$
  - 7:     Allocate the resources based on the processing capacity ( $P_c \in N_o$ ) by Algorithm3
  - 8:     Reallocate the virtual nodes once the service time is reached by Algorithm4
  - 9: **end for**
  - 10: Calculate the Acceptance Ratio.
  - 11: Calculate the Average node utilization.
- 

Fig.2: depicts the arrangement of nodes in both Physical and Overlay Network. Each physical node hosts 5 virtual nodes, sum of their processing capacity must be equivalent to the processing capacity of the underlying physical node.

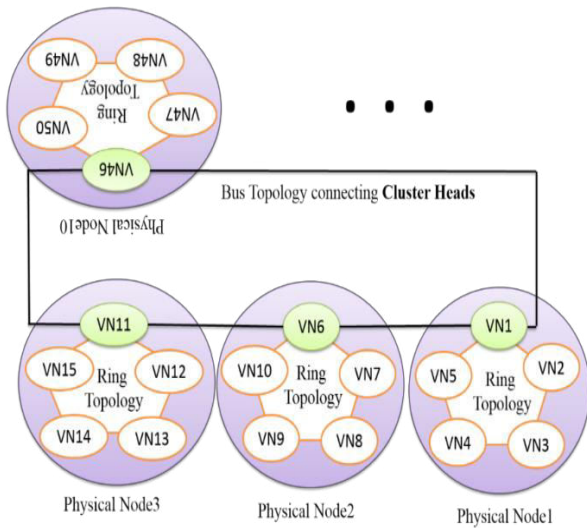


Fig.2: Motivational example of Virtual Node Arrangement

Moreover, virtual nodes hosted on a physical node forms a Ring Topology. Ring topology offers high bandwidth, so link capacity allocation is not taken into consideration. Among five, one is acting as a cluster Head. It communicates with the virtual nodes within the physical

node and with the cluster head of nearest neighbor. All cluster heads are connected to each other through Bus Topology.

### 5) The objectives of Overlay Network Embedding

In this section, two kinds of evaluation indexes such as revenue/cost ratio and acceptance ratio are included. No matter which evaluation index is used, the goal is to take advantage of infrastructure resources efficiently by means of mapping the overlay network requests into the shared physical network infrastructures while satisfying the resources constraints of nodes.

#### 5.1 The Acceptance Ratio of ONR

The acceptance ratio of ONR is the ratio of the number of overlay network request which have been successfully mapped and the total number of requests. The calculation formula is as follows:

$$Acceptance\ Ratio = \frac{\sum_{t=0}^T \# \text{ of Accepted ONRs}}{\sum_{t=0}^T \# \text{ of Requested ONRs}} \quad (5)$$

In the above formula, the numerator represents the number of overlay network request that have been successfully mapped, and the denominator represents the total number of overlay network requests.

#### 5.2 The Node Utilization of ONR

Node Utilization is the ratio between the number of substrate nodes utilized at that moment and the total number of substrate nodes available.

$$Utilization_{node} = \frac{N_{used}}{N_{total}} \quad (6)$$

where  $N_{used}$  in Expression (3) represents count of substrate node resources occupied after the VNR embedded successfully and  $N_{total}$  is gross count of node reserves.

## IV. SIMULATION RESULTS AND ANALYSIS

In this section, the simulation environment settings and the simulation results are discussed. We use the measurement index of performance metrics which proposed in section 2.4 to evaluate Ruckpack++ algorithm. Two state-of-the-art algorithms which called as *Greedy* [16] and *RW* [17] are compared with the proposed algorithm, respectively. In this paper, Ruckpack++ algorithm did not support path splitting, so we did not consider the case of supporting path splitting.

### 1) Experimental environment settings

NetBeans IDE 8.2 is used to implement Ruckpack++ algorithm. The specific parameter settings are shown in Table 1.

**Table1: Setting of Parameters**

Parameters	Scale
<b>Physical Network</b>	
# of physical nodes	10
Processing Capacity	120-300 (Uniform Distribution)
Topology	Bus Topology
Profit Gain	1-10
<b>Overlay Network</b>	
# of Virtual Nodes	55 (5 on each physical node)
Processing capacity	50 – 80 (in Uniform distribution)
Topology	Ring within each physical node & Bus among cluster Heads
<b>ONR</b>	
Arrival Rate	Poisson Distribution with $\lambda=12$

## 2) Experimental results and analysis

In order to evaluate the performance of Overlay Network embedding algorithms with different CPU capacity requirements, the CPU capacity of virtual nodes are uniformly distributed in [50-80]. The experimental results are shown in Fig. 3 and Fig.4.

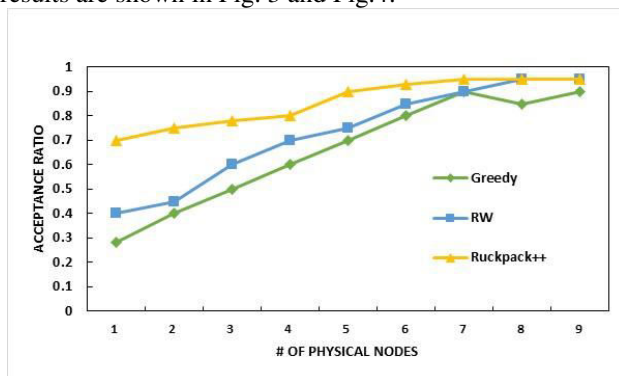


Fig.3: The Acceptance Ratio with increasing number of CPU.

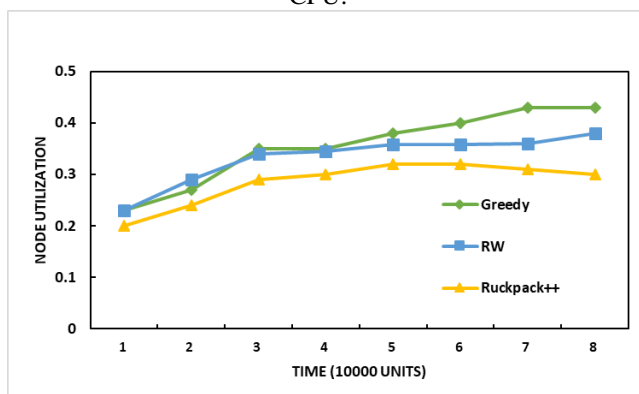


Fig.4: The Node Utilization with increasing time units

## 3) Acceptance ratio comparison

As illustrated in Fig. 3, the acceptance ratio will be gradually increasing along the increment of the Processing capacity requirements, through deep investigation and analysis, it is found that the available Processing capacity of physical nodes are gradually reducing with the number of overlay network requests (ONRs) which have been successfully mapped increases during the mapping process, As demonstrated in Fig. 3, proposed approach performed better when the demand of Processing requirements was increased. From the Fig. 3, it is observed that the performance of proposed algorithm called Ruckpack++ is superior to the other two algorithms.

## 4) Node Utilization comparison

As illustrated in Fig. 4, the Node Utilization will be gradually increasing along the time units, through deep investigation and analysis, it is found that the utilization of physical nodes are gradually increasing with the number of overlay network requests (ONRs) which have been successfully mapped, As demonstrated in Fig. 4, proposed approach performed better when the demand was increased. From the Fig. 4, it is observed that the performance of proposed algorithm called Ruckpack++ is superior to the other two algorithms as it utilizes a smaller number of physical nodes.

## V. CONCLUSION

In the era of network virtualization, virtual network embedding is one of the most difficult tasks to address the problem of network ossification. In the proposed Ruckpack++ algorithm, node allocation is done based on the Clusters, and its extensive simulations are done in NetBeans IDE 8.2. Result analysis shows that our proposed algorithm Ruckpack++ improves the performance of Overlay Network embedding procedure in terms of acceptance ratio and node utilization. The merits of the provided Ruckpack++ algorithm are as follows: (i) It uses the cluster-based approach to embed the virtual nodes (ii) It has omitted the parameter called bandwidth allocation or Link Capacity as the virtual nodes are connected in Ring Topology. It can be further optimized to enhance the energy efficiency of overlay network embedding algorithm.

## REFERENCES

- [1]. Z. He and G. Liang, "Research and Evaluation of Network Virtualization in Cloud Computing Environment," 2012 Third International Conference on Networking and Distributed Computing, Hangzhou, 2012, pp. 40-44, doi: 10.1109/ICNDC.2012.18.
- [2]. Javier Prieto, Abbes Amira, Javier Bajo, Santiago Mazuelas, Fernando De la Prieta, "IoT Approaches for Distributed Computing", Wireless Communications and Mobile Computing, vol. 2018, Article ID 9741053, 2 pages, 2018.

- [3]. GurpreetKaur, M.S.Patterh "Performance investigation of ring network topology in the presence of semiconductor optical amplifiers", Volume 124, Issue 22, November 2013, Pages 5366-5368.
- [4]. Zhiping Cai, Fang Liu, Nong Xiao, Qiang Liu, Zhiying Wang "Virtual Network Embedding for Evolving Networks", IEEE Globecom 2010 proceedings, 2010.
- [5]. Sheng Zhang, Zhuzhong Qian, Jie Wu, Sanglu Lu, and Leah Epstein, "Virtual Network Embedding with Opportunistic Resource Sharing", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 25, NO. 3, MARCH 2014.
- [6]. Ashraf A. Shahin, "Virtual Network Embedding Algorithms Based on Best-Fit Subgraph Detection", Computer and Information Science; Vol. 8, No. 1; 2015.
- [7]. Andreas Fischer, Juan Felipe Botero, Michael Till Beck, Hermann de Meer, and Xavier Hesselbach, "Virtual Network Embedding: A Survey", IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 15, NO. 4, FOURTH QUARTER 2013.
- [8]. Sandra Herker, Ashiq Khan, Xueli An, "Survey on Survivable Virtual Network Embedding Problem and Solutions", The Ninth International Conference on Networking and Services, ICNS, 2013.
- [9]. LaiMin, Liu Zhengliang, "Research on Algorithm for Virtual Network Embedding Based on Integer Programming", 3rd International Conference on Machinery, Materials and Information Technology Applications (ICMMITA 2015) proceedings, 2015.
- [10]. Xiang Cheng, Sen Su, Zhongbao Zhang, Kai Shuang, Fangchun Yang, Yan Luo, Jie Wang, "Virtual network embedding through topology awareness and optimization" Computer Networks 56 (2012) 1797–1813.
- [11]. Min Feng, Jianxin Liao, Jingyu Wang, Sude Qing, Qi Qi, "Topology-aware Virtual Network Embedding based on Multiple Characteristics", IEEE ICC 2014 - Next-Generation Networking Symposium, 2014.
- [12]. Marcelo Caggiani Luizelli, Leonardo Richter Bays, Luciana Salete Buriol, Marinho Pilla Barcellos, Luciano Paschoal Gaspary, "How physical network topologies affect virtual network embedding quality: A characterization study based on ISP and datacenter networks", Journal of Network and Computer Applications, 70 (2016) 1–16.
- [13]. Johannes Inf'uhr\_, David Stezenbachy, Matthias Hartmann y, Kurt Tutschkuy and G'unther R. Raidl, "Using Optimized Virtual Network Embedding for Network Dimensioning", IEEE conference on Networked Systems (NetSys), June 2013.
- [14]. He Hong, Wang Ning, "The Research of Group Mobility Model Based on Spectral Clustering Algorithm in Opportunistic networks", Int. J. Advanced Networking and Applications, Vol: 11 Issue: 05 Pages: 4378-4385(2020).
- [15]. Prashant Sangulagi, Ashok V. Sutagundar, Tabrunisa Abdul Rashid, "Network Lifetime Optimization in Sensor Cloud", Int. J. Advanced Networking and Applications, Vol 11, Issue:02, PP: 4198 -4204 (2020).
- [16]. M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: Substrate support for path splitting and migration," ACM SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, pp. 17\_29, 2008.
- [17]. X. Cheng et al., "Virtual network embedding through topology-aware node ranking," ACM SIGCOMM Comput. Commun. Rev., vol. 41, no. 2, pp. 38\_47, Apr. 2011.