International Journal of Innovative Research in Information Security (IJIRIS) ISSN: 2349-7017(O)
Issue 06, Volume 3 (September 2016) ISSN: 2349-7009(P)
www.ijiris.com

Optimization Multicast Routing Problem by Enabled Virtualization Rendezvous Point

Haider Nafaa N[1], Fang Liu[2]
School of computer science and technology [1,2],
Huazhong university, Wuhan Hubei [1,2],
China430074 [1,2],

Abstract*— Multicast provides effective communication and transmission, optimizes performance and enables truly distributed applications, also simultaneously optimizes cost of the multicast tree, average delay and maximum end-to-end delay. We analysed the shared tree algorithms that build a single tree to be used by all the multicast sessions. The tree contains a single point called core or virtualization rendezvous point (VRP) so that all the packets will be forwarded to VRP before reaching destination. In this way, we give a set of optimal solutions, based on Dijkstra's and Bellman-Ford algorithm then finding the Steiner Tree for constructs the minimum cost and get the maximum routing capacity through multicast rendezvous point by the knapsack problem with combinatorial optimization. According to simulations, the results show that algorithms realized best solution quickly and has a feasible, effective and scalability.

Keywords*— multicast routing, Dijkstra's algorithm, Bellman-Ford algorithm, virtualization rendezvous point

I. INTRODUCTION

The require of many applications send data from a source to multiple destinations through a telecommunication network. So that it is very necessary to define a multicast tree of minimum cost to connected the source node to the destination nodes Which are subject to delay restrictions. For this reason, multicast communication is becoming increasingly popular to save a huge amount of bandwidth [1]. Several basic multicast routing techniques have been developed for construction of multicast trees. Finding the optimal tree with the minimum cost is equivalent to finding the Steiner Tree. We propose such an algorithm to jointly determining the placement of virtualization nodes and constructing a multicast topology to connect the source with every end user through virtualization nodes. This single point called core or virtualization rendezvous point (VRP) so that all the packets will be forwarded to VRP before reaching destination [2] [3]. However, the selection of the optimal VRP is NP-hard problem. The exact solution algorithm based on Dijkstra's and Bellman-Ford algorithm then finding the Steiner Tree using Minimum Spanning Tree for constructs the minimum cost.

The knapsack problem (KP) is a problem in combinatorial optimization, we use it to compute the maximum routing capacity through multicast rendezvous point. When we have given a set of items, every have a weight and a value, determine the number of each item to include in a collection so that the total weight is lesser than or similar to given limit and the total value is as large as possible. That seeks for a better solution through many other solutions. A set of n items and a set of m resources are given. Each an item $j(j = 1, \bullet\bullet\bullet, n)$ has give a profit pj . a resource consuming value rij for every resource $i(i = 1, \bullet\bullet\bullet, m)$. The problem is to identify a subset of all item that lead to the high total profit and doesn't exceed the resource upper bound bi. The 0/1KP can be formulated as:

subject to maximize $f = \sum_{j=1}^{n} pjxj$ (1) $\sum_{j=1}^{n} rijxj \le bi$ (2)

$i = 1, \cdots, m, \quad xj \in \{0,1\}, j = 1, \cdots, n.$

The variable xj is an indicator of item j, if xj is set to 1, it mean item j is selected, or 0 mean item j isn't selected for $j = 1, \cdots, n$. Eq.1 represents the total profit of selection item and Eq.2 the m resource constraints [4].

A well stated 0/1 KP assumes that $pj > 0$ and $rij \le bi \le \sum_{j=1}^{n} rij$ for all $i = 1, \cdots, m,$ and $j = 1, \cdots, n.$

Multicast routing techniques issue schemes have been proposed in literature. In context of theoretical approach, they have always been among the simplest studied combinatorial optimization issues.

© 2014-16, IJIRIS- All Rights Reserved

The maximum routing capacity in network routing problem has be extensively studied as an optimization problem, using total link utilization and a bandwidth as objectives. Authors in [5] presents a new multicast algorithm based on the Strength Pareto Evolutionary Algorithm (SPEA), which together optimizes the cost of the tree, the maximum end-to-end delay and the average delay from the source node to each destination node. In [6], researchers presented optimize multiple quality of service parameters proposed with Bandwidth, Delay, Jitter, Packet Success Rate (PSR) and Packet Loss parameters based on genetic to finding optimal solution and global convergence with heuristic crossover and mutation operators can be achieved. Reference [7] propose a weighted fuzzy clustering algorithm for the mixed information data set indicating two different routing metrics. The schema first normalized the related routing metrics to include the feature vector, and so weighted the options in the fuzzy membership matrix identical to their related importance to the routing optimization for the better clustering performance. Wei Qun L et al. [8] study problem of constructing multicast trees to meet the quality of service requirement of real time period applications where it is necessary to provide bounded delays among the source and all destinations with keeping overall cost of the multicast tree low. The rest of the paper is organized as follows. Section II reviews problem statement of this research. Describing algorithm for solving VMN problem, In section III. While, section IV represent network design strategy next, section V presents experimental data and performance evaluation finally, section VI concludes the paper and presents future directions.

II. PROBLEM STATEMENT

Mathematically, we model the network as a graph $G = (V, E)$, which consists of a set of nodes V and undirected links E. We define $H \subseteq V$ as a set of candidate nodes on which VRP may be deployed. Each multicast session involves a source host $s \in V$ and a set of destinations $D \subseteq V$ .In this context, we define a multicast topology as a subgraph $G0 \subseteq G$. For each $G0$, there exists a mapping function $fG0 : D \rightarrow H0$, that maps each destination $d \in D$ to a VRP node $h \in H0$, where $H0 \subseteq H$ denotes the set of VRP nodes used by $G0$. Therefore, $G0$ delivers the multicast content to every $d \in D$ using two paths, one from s to a VRP node $fG0(d)$, the other from $fG0(d)$ to d. To define the cost of a multicast topology, we assume there is an non-negative fixed cost $w(e)$ for using each edge $e \in E$. Moreover, we assume running a VRP node on each machine $h \in H$ incurs an activation cost $c(h)$ in terms of resource usage and performance overhead. Therefore, the total cost of constructing a multicast topology consists of the activation cost of each VRP node and the sum of the link costs. Our goal is to find a multicast topology $G0$ that ensures each multicast flow traverses through the VRP node(s) before reaching the destination, while minimizing the total topology construction cost. Specifically, we define the total link cost of a subgraph $U \subseteq G$ as:

$$C^l(U) = \sum_{e \in U} w(e), \qquad (1)$$

and define total VRP activation cost as:

$$C^h(U) = \sum_{h \in U} c(h), \qquad (2)$$

then our goal is find a multicast topology U to minimize the sum of the total link cost and VRP activation cost:

$$C(U) = C^l(U) + C^h(U). \qquad (3)$$

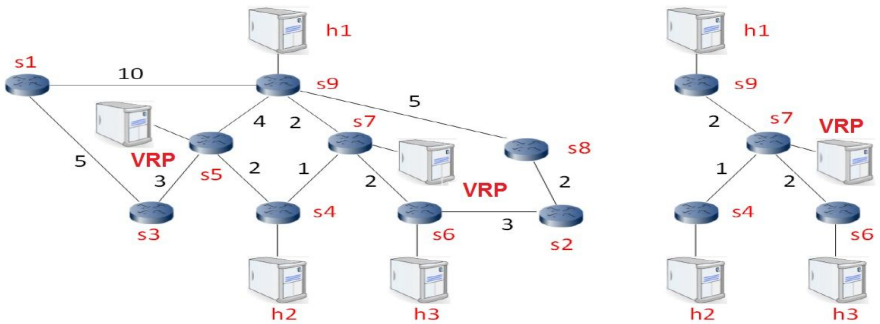


Fig. 1 Example of VMN

© 2014-16, IJIRIS- All Rights Reserved

We call this problem the VRP in multicast network *(VMN)*. A concrete example of (VMN) is provided in Fig. 1 (a). Consider a real-time video streaming service that delivers a trans coded version of original video stream from $h1$ to $h2$ and $h3$. Two nodes $\{5, 7\}$ are available to place the trans coder with activation cost $c(5) = 3$ and $c(7) = 1$ respectively. In this case, the goal of RPMN is to find the minimum cost multicast topology as shown in Fig. 1(b).

III.ALGORITHM FOR SOLVING VMN

In this section, we propose algorithms for solving VMN. We first searches for a VRP node which is used by all the destination users by using an algorithm that achieves shortest path problem. It computes the length of the shortest path from the source to VRP and each of the vertices on the plot by use an exact solution algorithm based on Dijkstra's and Bellman-Ford algorithm. Then it constructs the minimum cost is equivalent to finding the Steiner Tree using Minimum Spanning Tree among the end users. The traffic flow will first traverse through selected VRP node and then multicast to each end user using the MST it constructs.

Algorithm for Solving VMN

1: $G_{opt} = \infty$ optimal multicast topology of VMN
2: Let:graph G=(V,E) set of (node, undirected link)
3: Define $H \subseteq V$ is candidate node VRP node and destination $D \subseteq V$
4: H,D is inside range NO go 3
5: Yes for each (H,D) pair $\in H \times D$ do
6: Find shortest path form S to H No go5
7: Construct shortest path graph P_{opt} by connected S to H
8: Find minimum spanning tree from H to D No go 7
9: Construct minimum spanning node graph T_{opt}
10: Connect P_{opt} with T_{opt} to gather call the combined graph G_{opt}
11: Calculate $G_{opt} = P_{opt} + T_{opt}$ it's include the activation VRP node and total cost of link

---

IV.NETWORK DESIGN

In graph theory, a weighted undirected graph G is a dual group G = (V, E). The network under consideration is represented as a linked graph with N nodes. The metric of optimization is the cost of path between the nodes. The total cost is the sum of the individual jumps costs. The goal is to find the path with minimum total cost between source node and destination node through VRP node. That comes by compare two scenarios imposed for optimum transmission of data between the source and destination .

1- Dijkstra and Prim's algorithm
2- Bellman-Ford and Prim's algorithm

Each node in the graph indicates a domain and each domain are linked by undirected edges. The network graph shown below is represented as a text file which indicates the advertised information such as the domain capacities and the adjacent domains. all domains advertise their domain capacities over the network. It is noteworthy that unlike the traditional multicast tree problem, in VMN each link may be traversed more than once. For instance, in Fig. 2, assume the only VRP node is located at node 4 and $D = \{2, 3, 5\}$, $s = 1$. The multicast routes are:

1) From 1 to 3 : (1, 6, 4, 6, 3)
2) From 1 to 5 : (1, 6, 4, 6, 5)
3) From 1 to 2 : (1, 6, 4, 6, 3, 1, 2)

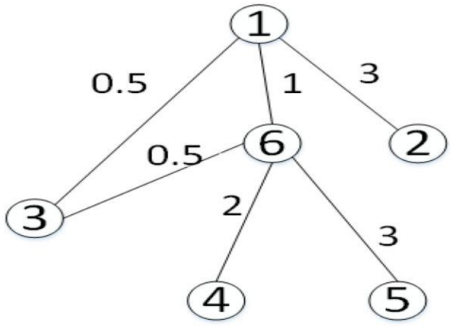


Fig.2 An example of VMN solution.

© 2014-16, IJIRIS- All Rights Reserved

In this case, the total cost of this multicast topology is $C(U) = w(1, 6) + w(6, 4) + w(6, 4) + w(6, 5) + w(6, 3) + w(3, 1) + w(1, 2) + c(4)$. The cost of link (6, 4) is counted twice since the flow traverses the link (6, 4) twice. And the cost of the RP node only counts once. The connection matrix (a square matrix of dimension N x N that represents a connection between each node-pairs) of the Fig. 3 network is:

M =

0	1	1	0	0	1
1	0	0	0	0	0
1	0	0	0	0	1
0	0	0	0	0	1
0	0	0	0	0	1
1	0	1	1	1	0

Fig. 3 The connection matrix of network in Fig.2.

The cost matrix *CM* for the network shown in Fig. 4 is in the following form:

CM=

0	3	0.5	0	0	1
3	0	0	0	0	0
0.5	0	0	0	0	0.5
0	0	0	0	0	2
0	0	0	0	0	3
1	0	0.5	2	3	0

Fig. 4 The cost matrix network in Fig. 2

The network is created using the following MATLAB command:

```
W = [1 2 3 3 .5 .5];
DG = sparse([1 6 6 1 6 1],[6 4 5 2 3 3],W)
UG = tril(DG + DG')
view(biograph(UG,[],'ShowArrows','off','ShowWeights','on'))
```

The first four row vectors in the sparse command represent source nodes, destination nodes, and equivalent costs respectively and then view the topology as shown in Fig. 4.

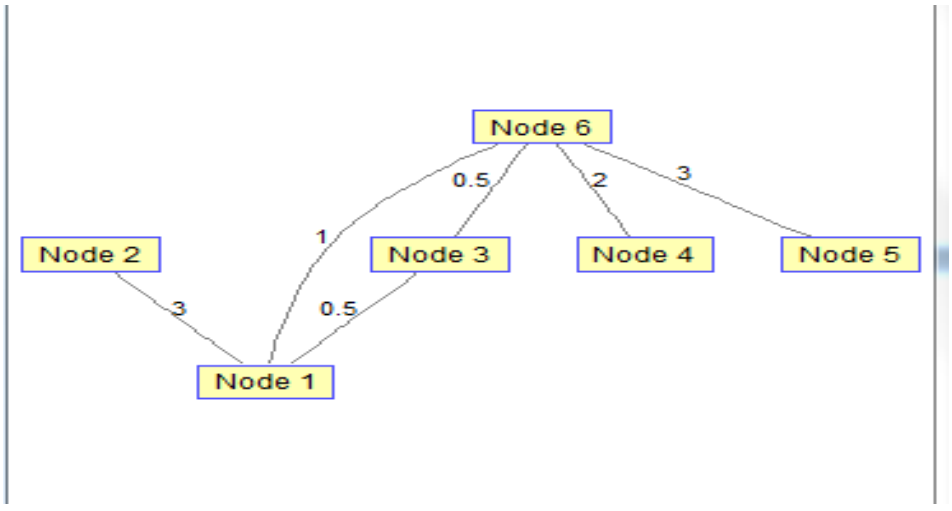


Fig. 4 Topology Network.

© 2014-16, IJIRIS- All Rights Reserved

A- DIJKSTRA AND PRIM'S ALGORITHM

The MATLAB command graphshortestpath is executed to find the shortest path. This command applies Dijkstra's algorithm as the default one to find the optimal solution .also graphminspantree is executed to find the minimum spanning tree

```
tic
[dist,path,pred] = graphshortestpath(DG,1,4)
[ST,pred] = graphminspantree(UG)
toc
T=dist+sum(ST(:))
```

We obtain the following results, i.e. **shortest path= {1-6-4}, shortest path cost = 3, MST cost=9 and total cost=12 .**Biography shows the multicast routing network. Elapsed time for this network is **0.006951** sec using **tic** and **toc** commands.

B- BELLMAN-FORD AND PRIM'S ALGORITHM

The MATLAB command graphshortestpath(DG,1,4,'Method','Bellman-Ford') is executed to find the shortest path. This command applies Bellman-Ford algorithm to find the optimal solution .also graphminspantree is executed to find the minimum spanning tree

```
tic
[dist,path,pred] = graphshortestpath(DG,1,4,'Method','Bellman-Ford')
[ST,pred] = graphminspantree(UG)
toc
T=dist+sum(ST(:))
```

We obtain the following results, i.e. **shortest path= {1-6-4}, shortest path cost = 3, MST cost=9 and total cost=12 .**Biography shows the multicast routing network. The Elapsed time for this network is **0.005652** sec Fig. 5 shows the Matlab results for routing network.

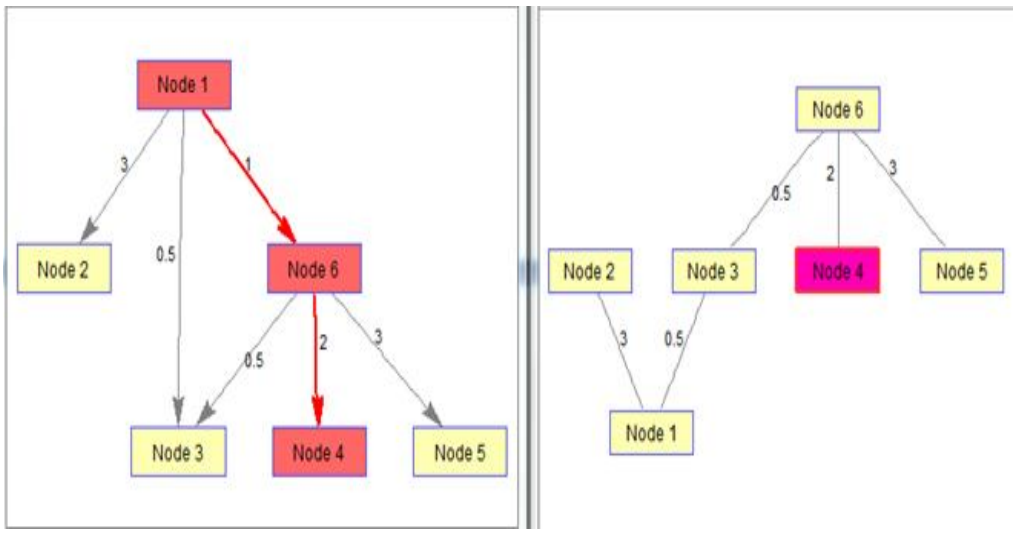


Fig.5 Matlab results for routing network

V. EXPERIMENTAL DATA AND PERFORMANCE EVALUATION

A. EFFICIENT THE MINIMUM COST MULTICAST TOPOLOGY

We create a large randomly generated matrix using MATLAB to test the goal of VMN is to find the minimum cost multicast topology . We use each N x N matrix for each algorithm .Each Dijkstra's Algorithm and Bellman-Ford algorithm and Prim's algorithm are implemented based on the pseudo code from the previous chapter. Also the compiled once for every number of vertices from 1 to 600 . Then the number of vertices is increased and new graph topology each time is created with random square matrix. Fig. 6,Fig. 7 and Fig. 8 are line graphs from runtimes of both major shorter path and minimum spanning tree. Fig. 6 Note expansion in the vector path with increasing time .Fig. 7 cross the vertices in the graph reach around 420 vertices while Fig. 8 shows runtimes for 1 to 600 vertices. the Bellman-Ford algorithm vertices increasing with time more than Dijkstra's algorithm vertices.

© 2014-16, IJIRIS- All Rights Reserved

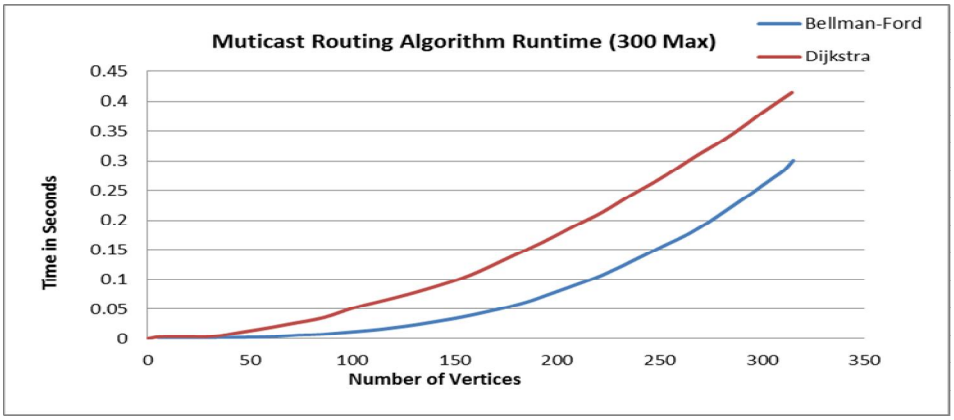


Fig. 6 Runtime results for Dijkstra vs. Bellman-Ford (1 to 300 vertices)

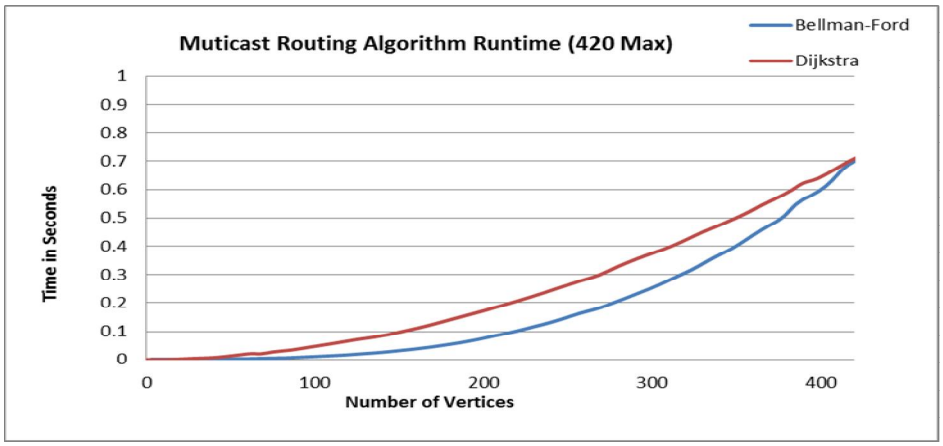


Fig. 7 Runtime results for Dijkstra vs. Bellman-Ford (1 to 420 vertices)

After running the two single-source shortest path algorithms, it can be seen that the Bellman-Ford algorithm is more efficient than Dijkstra's algorithm up until the number of vertices in the graph reach around 420. From this point, Dijkstra's algorithm shows to be much more efficient as the number of vertices approach infinity.

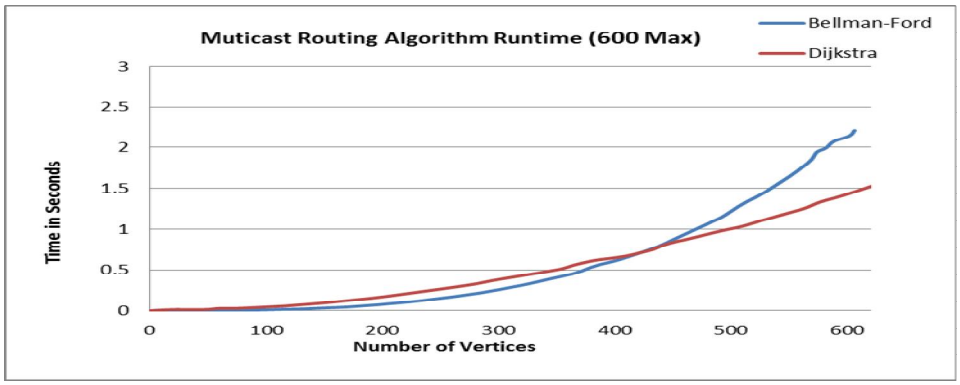


Fig. 8 Runtime results for Dijkstra vs. Bellman-Ford (1 to 600 vertices)

The Bellman-Ford algorithm and Dijkstra's algorithm proved to be much more efficient, With Dijkstra proving to run in the least amount of time for very large networks. It appears that for complete, fully meshed networks, the Bellman-Ford algorithm actually ran faster than Dijkstra's algorithm for networks of size 420 or less. However, Dijkstra showed to be considerably more efficient beyond networks with more than 420 vertices.

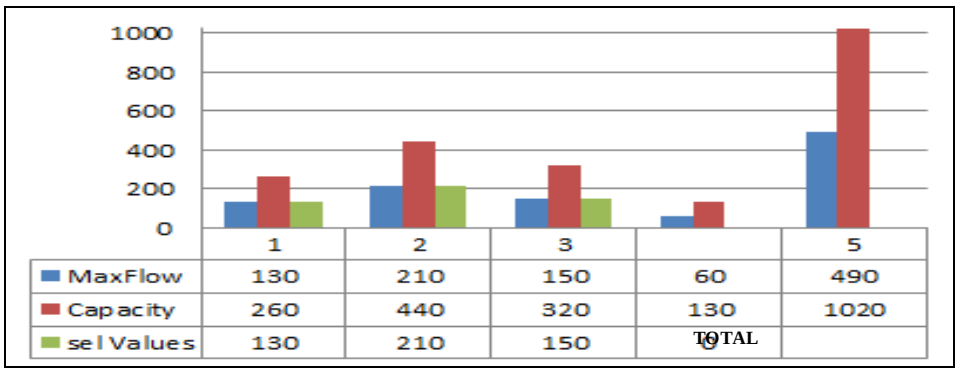
B. PERFORMANCE EVALUATION USING 0-1 KNAPSACK PROBLEM

We can suppose the maximum routing capacity of VRP in multicast network *(VMN)* presented by the total Knapsack size and the total number of items (number of multicast paths) should fill in the knapsack (should be routed by *VMN*) is four.

© 2014-16, IJIRIS- All Rights Reserved

1. The selected items represented by selected multicast paths.
2. The maximum KP represented by maximum routing capacity of VMN.
3. The total number of items represented by the number of multicast paths.
4. The weight of item represented by amount of multicast data per path.

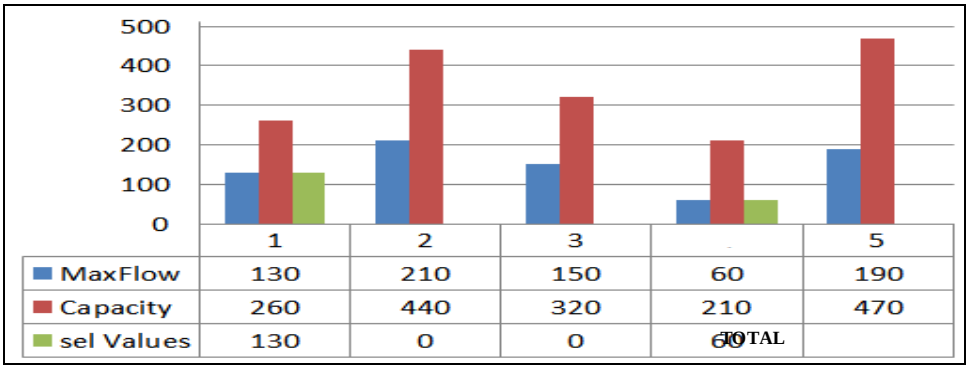
Fig. 9 shows the paths selected by VMN (sel. values) when the maximum number of multicast paths should be routed is four, found three of them been routed according to data amount (weight) per path in combinational optimization selection manner. Mean that the total amount of data must route on 4 paths is 1024 Mb/s data units, found that 490 Mb/s data units been routed. Fig. 10 and Fig. 11 shows the effect decreasing of the maximum routing capacity of VMN in the total amount of data to 512 Mb/s and 256 Mb/s.



	1	2	3		5
MaxFlow	130	210	150	60	490
Capacity	260	440	320	130	1020
sel Values	130	210	150	0	

TOTAL

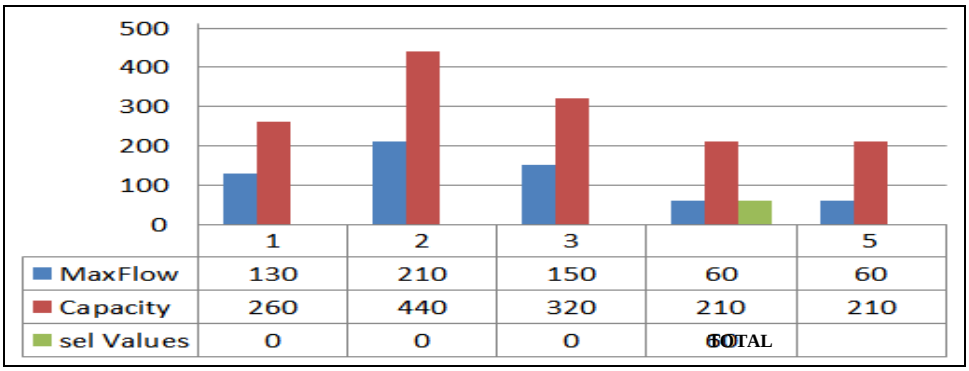
Fig. 9. KP size =1024 & number of Items=4.



	1	2	3		5
MaxFlow	130	210	150	60	190
Capacity	260	440	320	210	470
sel Values	130	0	0	60	

TOTAL

Fig. 10. KP size =512 & number of Items=4.



	1	2	3		5
MaxFlow	130	210	150	60	60
Capacity	260	440	320	210	210
sel Values	0	0	0	60	

TOTAL

Fig. 11. KP size =256 & number of Items=4.

The multicast rendezvous point schemes increase efficiency of multicast traffic in networks and it adds new entity to the network and more addressing complexity.

© 2014-16, IJIRIS- All Rights Reserved

VI. CONCLUSION

From the information provided in our Experimental data, it is obvious to see that the Dijkstra's algorithm proved to be much more efficient than Bellman-Ford algorithm, with Dijkstra proving to run in the least amount of time for very large networks. The knapsack algorithm gives us a new idea to improve the performance of AB multicasting in Virtualization systems and also to ensure the forwarding data to their destinations in combinational optimize selection manner, and acceptable throughput. I believe I reached a successful analysis and look forward to more mathematical research endeavors in the future.

ACKNOWLEDGMENT

The authors gratefully acknowledge the support from the national natural science foundation of china under (grant u1233119).

REFERENCES

[1] A. Younes, An Ant Algorithm for Solving QoS Multicast Routing Problem, International Journal of Computer Science and Security (IJCSS), Volume (5) : Issue (1) : 2011.
[2] Mukherjee, R. ; Atwood, J.W. "Rendezvous point relocation in protocol independent multicast - sparse mode". 10th International Conference on Telecommunications, 2003. ICT 2003.
[3] Macq, J.-F. ; Wolsey, L.A. ; Macq, B. "A rendezvous point selection algorithm for videoconferencing applications". 2002 IEEE International Conference on Multimedia and Expo, 2002. ICME '02. Proceedings.
[4] QIAN Fubin , DING Rui ,"Simulated Annealing for the 0/1 Multidimensional Knapsack Problem" ,University of Science & Technology of Suzhou,Supported by the National Natural Science Foundation of China (No. 10201026, 10672111).
[5] Jorge C, Benjamín B, A Multicast Routing Algorithm Using Multiobjective Optimization, Universidad Nacional de Asunción Asunción –Paraguay.
[6] Sukhvinder S, Manoj K, Dr. Sanjeev K, Multiple Constraints QoS Multicast Routing Optimization Algorithm based on GA for MANET, International Journal of Engineering Technology, Management and Applied Sciences9 (ijetmas), Volume 3, Special Issue, ISSN 2349-4476,2015.
[7] Qi Bing, Jun Lu, Yan Long, A Weighted Fuzzy Clustering Algorithm for Multicast Routing Optimization, Fuzzy Systems and Knowledge Discovery, ISBN: 978-0-7695-3305-6 ,pp: 71-75,2008.
[8] Wei Qun L, Xiang Kui C, A Delay-Constrained Multicast Routing Balanced-Optimization Algorithm of Link-Sharing, Applied Mechanics and Materials ,Volumes 152-154, 1834-1839,2012.

© 2014-16, IJIRIS- All Rights Reserved