Advances in Computer Science and Information Technology (ACSIT)
Print ISSN: 2393-9907; Online ISSN: 2393-9915; Volume 2, Number 3; January-March, 2015 pp. 249-251

http://www.krishisanskriti.org/acsit.html

Effect of the Core Selection Strategy in Determining Optimum BDMST in Multicast Networks

Sakshi Arora[1] and Sunanda[2]

[1]SMVD University Kakryal, Katra, J&K
[2]SMVD University Kakryal, Katra, J&K
E-mail: [1]sakshi@smvdu.ac.in, [2]sunanda.gupta@smvdu.ac.in

Abstract—*Choice of the centre or core is an issue that has been a major issue in the research in multicast networks, as the choice of centre greatly influences the delay and hence the quality of service in such networks. Various core selection methods exist in literature ranging from simple arbitrary core selection method to complex heuristics, involving the use of domain specific knowledge. An issue in this regard is, how complex the core selection process should be to ensure a reasonable trade-off between performance and the ensuing time-complexity. A case in this point are the heuristics which can be strategically used to leverage the domain specific knowledge, thereby yielding core selection methods that result in significantly better network performance and can keep the complexity under check. In this paper, we present a core selection method that can significantly improve the performance of the underlying multicast network while keeping the time complexity of the process lower than the existing core selection techniques by a factor of roughly n, where n is the number of nodes in the graph.*

1. INTRODUCTION

Selection of the centre of graph (tree) is an important topic both of practical and theoretical interest in multicast routing. The choice of the centre in such applications influences the shape of the network and thus influences the performance of the underlying heuristic [1], [2]. Choice of the centre or core therefore is an issue that has been a major issue in the research in multicast networks, as the choice of centre greatly influences the delay and hence the quality of service in such networks. Various core selection methods exist in literature ranging from simple arbitrary core selection method to complex heuristics, involving the use of domain specific knowledge. An issue in this regard is, how complex the core selection process should be to ensure a reasonable trade-off between performance and the ensuing time-complexity. The addition of topological information can be used to improve performance especially when nodes form a non uniform cluster in which the occurrence of outliers cannot be ruled out [3]**.** An issue in this regard is, how complex the core or centre choice method should be, to ensure a reasonable performance. It is argued, for example in [4], that the random choice of centre is better than arbitrary choice and that topologically informed choice of the centre leads to even better performance. Broadly, when there is little variation in the performance of different centres, an arbitrary choice is sufficient. When the variance is significant, more sophisticated methods to choose the core may be required [4].

2. RELATED WORK

Many new multimedia applications such as videoconferences or multiplayer games on the internet ask for the selection of a meeting point in the network, where each user sends his data to the meeting point. The role of this particular point is to gather the information received to create a single composite data flow, which is multicast back to the users. Investigations have been carried out to measure the relationship between the choice of the location of centre and the performance of the routing scheme [5] [6]. Wei and Estrin [7] however have been the first to point out the relationship between the choice of core and network performance. The performance of any such a scheme is usually evaluated in terms of delay and bandwidth consumption. If the objective is to minimize the average delay between every pair of nodes in the network, the optimal path is achieved on a shortest path tree rooted at the centre. In that case, each node-centre shortest path is used in both directions, and the optimal location of the centre is such as to minimize the sum of the shortest path lengths between the centre and every node. The problem of locating the centre node c under these two criteria, one being the sum of the edges between v and rest of the nodes and the other being the cost of the tree linking c to the rest of the nodes has been solved in [8] using an enumeration algorithm that evaluates the objective function at every potential core location, but the computational complexity of such an enumerative approach is extremely high. Besides, previous works considering the relationship between core choice and performance has focused primarily on worst case bounds [9], and performance with an optimal choice of core [10]. In this paper we extend the

Center_Determine( ) strategy proposed by the author in [11], that, makes the choice of the centre under the above two criteria but manages to keep the time complexity reasonably low.

3. PROPOSED STRATEGY

As the starting vertex significantly affects the weight of the generated trees therefore, in *Center_Determine()*[11], rather than choosing randomized centre (one centre if D is even and two otherwise), randomly from the vertex set '*V*' , we propose to choose the centre from a specified subset of '*V*' The vertex set '*V*' of the instance graph is partitioned into three subsets *Potential_Centres*, *Intermediate_V and Pendant_V* (refer Fig. 1), based on their potential to form the centre, the backbone of the bounded diameter minimum spanning tree (BDMST) and the auxiliary edges. The Centre is chosen from an informed subset '*Potential_Centres*' of '*V*'. In addition, while extending the tree by adding vertices, the next vertex is not chosen randomly from '*V*', but from these three subsets in order of merit.

In the subroutine *Center_Determine()*, the entries [i,j] of the matrix *Vertex_Potency* are initialized for the input graph *G* by creating an adjacency matrix of *G*. The entries *Vertex_Potency*[i][j] of this matrix represent the number of hops taken to reach from vertex *i* to each vertex *j*, on the least weighted path. Sum of the number of hops it takes to reach from a vertex *i* to every other vertex *j* gives an estimate of the measure of 'goodness' of the vertex *i* to be the centre. Let *S[i]* denote the vector storing this sum for all the vertices of the graph. Further, *S[i]* is sorted along with its vertex indices in ascending order, to reflect the relative potential of each vertex to act as the centre of the tree, or to be in the backbone of the tree.

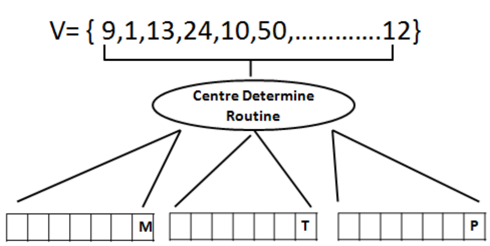

Fig. 1: Vertex Partitioning using Centre_Determine( ) Subroutine

If the objective is to minimize the average delay between every pair of users in the group, the optimal routing is achieved on a shortest path tree rooted at the centre. In that case, each shortest path from the centre to every other node is used in both directions, and the optimal location of the centre is such as to minimize the sum of the shortest path lengths between the centre node and every other node. For detailed algorithm, refer [12].

The average of the *S[i]* values is computed and the number of vertices having *S[i]* value less than the average, are stored in the variable '*l*'. The first *l* vertices of the sorted vector *S[i]*contend to populate the set *Potential_Centres*. The reason for determining the set *Potential_Centres* in this manner is that it can be safely assumed that the vertices which take less than average number of hops to reach all other nodes will form better centre and a backbone to which the rest of the vertices can be connected using lighter edges and lesser number of hops. A random number *Q* is then determined in the range:

Q = random (l-2) + 2

The first Q vertices out of *l* populate the subset Potential_Centres.

4. EXPERIMENTATION AND RESULTS

Based on a set of Euclidean instances of network graphs, a number of multicast scenarios are defined. For each individual node of an instance, the performance of a routing algorithm was measured when that node was selected as the core. To normalize these measurements so that the differences among various network instances are accounted for, the average performance of these cores are considered.

Table1: Summary of trials of OTTC, CBTC, RTC and DRGH on 100 Euclidian test instances.

Instance		D	OTTC		CBTC		RTC		DRGH	
			T_{mean}	σ	T_{mean}	σ	T_{mean}	σ	T_{mean}	σ
n	**100**	5	27.64	1.51	24.42	1.43	14.24	0.68	**10.80**	0.09
l	**6.48**	10	19.25	1.66	17.09	1.18	9.77	1.43	**6.27**	0.15
d	**42.67**	15	11.64	1.21	10.25	0.89	9.03	0.38	**8.89**	0.13
$min\ d$	**29.00**	25	9.13	0.22	**7.73**	0.29	9.86	0.31	8.11	0.07
n	**250**	10	54.67	4.27	51.27	2.69	20.89	0.37	**11.89**	0.15
l	**10.62**	15	43.29	3.13	39.24	3.45	15.30	0.29	**12.27**	0.12
d	**79.60**	20	35.43	2.75	31.15	2.19	14.85	0.32	**12.05**	0.11
$min\ d$	**48.00**	40	15.63	2.45	**11.26**	0.65	15.12	0.19	**11.23**	0.08
n	**500**	15	109.67	5.18	97.28	5.26	24.86	0.33	**17.12**	0.11
l	**14.75**	30	60.22	6.89	46.51	3.89	21.54	0.39	**16.34**	0.13
d	**123.90**	45	35.29	6.13	19.24	2.43	29.73	0.42	**18.13**	0.12
$min\ d$	**91.00**	60	19.47	3.83	**17.29**	0.87	23.65	0.34	17.71	0.13
n	**1000**	20	223.67	8.69	199.12	7.28	37.22	0.37	**22.27**	0.07
l	**20.84**	40	132.56	17.21	103.27	7.36	31.63	0.29	**23.47**	0.14
d	**202.13**	60	72.83	11.45	53.67	5.01	31.24	0.31	**25.23**	0.09
$min\ d$	**152.00**	100	29.95	3.84	**21.47**	0.63	30.79	0.26	26.75	0.14

Reasonable comparison can be drawn for different core selection techniques with relatively large graph instances are considered. Core-based routing algorithms offer good scalability and therefore we have used large network instances of up to 500 nodes for our study. The graph construction strategy is similar to that of [4] where, the first nodes in each instance are assigned random coordinates in a unit square. A link is then established between each pair of nodes with a probability $P_\alpha(d)$, where d is the Euclidean distance and α is the degree of connectivity which can be adjusted. *Center_Determine()* strategy has been incorporated in a BDMST construction routine DRGH [11].The proposed strategy assumes significance in light of the issues which arise in the construction of BDMSTs using greedy & semi-greedy strategies like OTTC[13], CBTC[14] and RGH[15].

This is because in case of existing heuristics such as RTC and OTTC the random choice of centre leads to significantly higher variance in the solution.

Consequently, the solution has to be normalized by running the heuristic n times, each time with a new vertex as centre which leads to enormously high computational complexity.

5. CONCLUSION

The proposed strategy overcomes these issues by choosing the centre from an informed subset of the vertices. The topological information of the instance graph guides the heuristic in choosing the centre, thus, reducing the sensitivity of the final solution to the initial choice of the centre. A measure of 'goodness' is calculated for the vertices to act as the centre or to be in the backbone, leading to generation of BDMSTs which have longer backbone edges and lighter stars.

REFERENCES

[1] Wei, L., & Estrin, D. (1994, Sepetember). The trade-offs of multicast trees and algorithms. In Proceedings of ICCCN (Vol. 94, pp. 902-926).

[2] Whitley, L.D., Starkweather, T., & D'Ann Fuquay. (1989, June). Scheduling problems and travelling Salesmen: The genetic edge recombination operator. In/CGA (Vol. 89, pp. 133-40).

[3] Van Hentenryck, P., & Michel, L. (2003, January). Control abstractions for local search. In *Principles and Practice of Constraint Programming–CP 2003* (pp. 65-80). Springer Berlin Heidelberg.

[4] Calvert, K. L., Zegura, E. W., & Donahoo, M. J. (1995, September). Core selection methods for multicast routing. In *Computer Communications and Networks, 1995. Proceedings., Fourth International Conference on* (pp. 638-642). IEEE.

[5] Fleury, E., Huang, Y., & McKinley, P. K. (1998, October). On the performance and feasibility of multicast core selection heuristics. In *Computer Communications and Networks, 1998. Proceedings. 7th International Conference on* (pp. 296-303). IEEE.

[6] Li, T. H., Lucasius, C. B., & Kateman, G. (1992). Optimization of calibration data with the dynamic genetic algorithm. *Analytica Chimica Acta, 268*(1), 123-134.

[7] Wei, L., & Estrin, D. (1993). A comparison of multicast trees and algorithms.*Submitted to INFOCOM*, *94*.

[8] Lucasius, C. B., & Kateman, G. (1993). Understanding and using genetic algorithms Part 1. Concepts, properties and context. *Chemometrics and intelligent laboratory systems*, *19*(1), 1-33.

[9] Wieselthier, J. E., Nguyen, G. D., & Ephremides, A. (2002). Energy-efficient broadcast and multicast trees in wireless networks. *Mobile Networks and Applications*, *7*(6), 481-492.

[10] Donahoo, M. J., & Zegura, E. W. (1996, October). Core migration for dynamic multicast routing. In *Proceedings of 5th IEEE ICCCN'96* (pp. 92-98).

[11] Arora, S., & Garg, M. L. (2009). New hybrid evolutionary algorithm for solving the bounded diameter minimum spanning tree problem.

[12] Arora, S., & Garg, M. L. (2011, June). Clustering the data points to obtain optimum backbones for the bounded diameter minimum spanning trees. In*Communication Systems and Network Technologies (CSNT), 2011 International Conference on* (pp. 303-307). IEEE.

[13] Deo, N., & Abdalla, A. (2000). Computing a diameter-constrained minimum spanning tree in parallel. In *Algorithms and Complexity* (pp. 17-31). Springer Berlin Heidelberg.

[14] Julstrom, B. A. (2009). Greedy heuristics for the bounded diameter minimum spanning tree problem. *Journal of Experimental Algorithmics (JEA)*, *14*, 1.

[15] Raidl, G. R., & Julstrom, B. A. (2003). Edge sets: an effective evolutionary coding of spanning trees. *Evolutionary Computation, IEEE Transactions on*,*7*(3), 225-239.