On Core Selection Algorithm for Reducing Delay Variation of Many-to-Many Multicasts with Delay-Bounds*

Moonseong Kim[1], Young-Cheol Bang[2], and Hyunseung Choo[1]

[1] School of Information and Communication Engineering
Sungkyunkwan University
440-746, Suwon, Korea +82-31-290-7145
{moonseong,choo}@ece.skku.ac.kr

[2] Department of Computer Engineering
Korea Polytechnic University
429-793, Gyeonggi-Do, Korea +82-31-496-8292
ybang@kpu.ac.kr

Abstract. With the proliferation of multimedia group applications, the construction of multicast trees satisfying the quality of service (QoS) requirements is becoming a problem of the prime importance. In this paper, we study the core selection problem that should produce the improved delay-bounded multicast tree in terms of the delay variation that is known to be NP-complete [8]. A solution to this problem is required to provide decent real-time communication services such as on-line games, shopping, and teleconferencing. Performance comparison shows that our proposed scheme outperforms that of DDVCA [18] that is known to be most effective so far in any network topology. The enhancement is up to about 11.1% in terms of normalized surcharge for DDVCA. The time complexity of our algorithm is $O(mn^2)$.

1 Introduction

New communication services involving multicast communications and real time multimedia applications are becoming prevalent. In multicast communications, messages are sent to multiple destinations that belong to the same multicast group. These group applications demand a certain amount of reserved resources to satisfy their quality of service (QoS) requirements such as end-to-end delay, delay jitter, loss, cost, throughputs, and etc. Since resources for multicast tree are reserved along a given path to each destination in a given multicast tree, it may fail to construct a multicast tree to guarantee the required QoS if a single link cannot support required resources. Thus an efficient solution for multicast communications includes the construction of a multicast tree that has the best chance to satisfy the resource requirements [1, 4, 9–12, 19, 21, 22].

* This paper was supported in part by Brain Korea 21 and University ITRC project. Dr. H. Choo is the corresponding author.

Algorithms for the tree construction in multicast protocols can be categorized as follows: source-based algorithms (SBA) and core-based algorithms (CBA) [20]. SBA constructs a tree rooted at source that originates and sends messages to each destination in the multicast group. SBA is currently being used as the tree construction algorithm for Distance Vector Multicast Routing Protocol (DVMRP) [14], Protocol Independent Multicast Dense Mode (PIM-DM) [7], and Multicast Open Shortest Path First (MOSPF) [13]. On the other hand, CBA that is used for many-to-may multicasts selects a core node as a root of the multicast tree we want to determine at the first step. Then a tree rooted at the core node is constructed to span all members in the multicast group. Thus it is very important to select the best core node as much as possible. To send messages originated at source, messages are sent to the core and distributed to destinations along the path to the core node. Once messages are reached at the core node, messages are sent to remaining destinations. Multicast protocols that use CBA as a tree construction algorithm include Protocol Independent Multicast Sparse Mode (PIM-SM) [5, 7] and the Core-Based Tree (CBT) protocol [2, 3].

For multicast communications such as a teleconference, it is very critical that the current speaker must be heard by all participants simultaneously, but otherwise the communication may lose the feeling of an interactive face-to-face discussion. Another similar dispute can be easily found in on-line video games. These are all related to the multicast delay variation problem [17]. In this paper, we introduce a novel core selection algorithm that can improve the delay and delay variation constraint algorithm (DDVCA) known to be the best algorithm [18]. In the DDVCA, the selection of a core node over several candidates (possible core nodes) is overlooked in the a core node is randomly selected among candidates. Meanwhile we investigate candidate nodes to select the better node with the same time complexity of DDVCA. Our algorithm with the tree construction part of DDVCA was empirically compared with DDVCA, and evaluation determined that the delay variation of the tree constructed by our algorithm is smaller than that of the tree constructed by DDVCA. The enhancement is up to about 3.6%∼11.1% in terms of the normalized surcharge for DDVCA. Our main contribution in this research is as follows. We propose a new core selection algorithm that produces the multicast tree with the better delay variation in comparison with DDVCA. The time complexity of our algorithm is the same as that of the DDVCA.

The rest of the paper is organized as follows. In section 2, we study related works, and section 3 presents details of the proposed algorithm. Then, in section 4, we evaluate the proposed algorithm by the simulation model. Section 5 concludes this paper.

2 Related Works

We consider a computer network represented by a directed graph $G = (V, E)$ with n nodes and l links or arcs, where V is a set of nodes and E is a set of links (arcs), respectively. Each link $(i, j) \in E$ is associated with delay $d_{(i,j)}$. The delay of a link is the sum of the perceived queueing delay, transmission delay, and propagation delay over that link. We assume that the delay on each arc is asymmetric in general. Given a network G, we define a path as sequence of nodes $u,\ i,\ j,\ \ldots,\ k,\ v$, such that $(u,i),\ (i,j),\ \ldots,\ (k,v)$, belongs to E. Let $P(u,v) = \{(u,i),\ (i,j),\ \ldots,\ (k,v)\}$ denote the path from node u to node v. If all elements of the path are distinct, then we say that it is a simple path. We define the length of the path $P(u,v)$, denoted by $n(P(u,v))$, as a number of links in $P(u,v)$. Let $\preceq$ be a binary relation on $P(u,v)$ defined by $(a,b) \preceq (c,d) \leftrightarrow n(P(u,b)) \leq n(P(u,d))$, $^{\forall}(a,b),\ (c,d) \in P(u,v)$. $(P(u,v), \preceq)$ is a totally ordered set. For a given source node $s \in V$ and a destination node $d \in V$, $(2^{s \Rightarrow d}, \infty)$ is the set of all possible paths from s to d.

$$(2^{s \Rightarrow d}, \infty) = \{\ P_k(s,d) \mid \text{all possible paths from } s \text{ to } d,\ ^{\forall}s,\ d \in V,\ ^{\forall}k \in \Lambda\ \}$$

where Λ is a index set. The delay of arbitrary path P_k is assumed to be real function from $(2^{s \Rightarrow d}, \infty)$ to nonnegative real number $\mathcal{R}^+$. Since $(P_k, \preceq)$ is a totally ordered set, if there exists a bijective function f_k then P_k is isomorphic to $\mathcal{N}_{n(P_k)}$.

$$P_k = \{(u,i),(i,j),\ldots,(m,v)\} \xrightarrow{f_k} \mathcal{N}_{n(P_k)} = \{1,2,\ldots,n(P_k)\}$$

We define,

$$\text{function of delay along the path } \phi_D(P_k) = \sum_{r=1}^{n(P_k)} d_{f_k^{-1}(r)}\ ,\ ^{\forall}P_k \in (2^{s \Rightarrow d}, \infty)\ .$$

$(2^{s \Rightarrow d}, supD)$ is the set of paths from s to d for which the end-to-end delay is bounded by $supD$. Therefore $(2^{s \Rightarrow d}, supD) \subseteq (2^{s \Rightarrow d}, \infty)$. For multicast communications, messages need to be delivered to all receivers in the set $M \subseteq V \setminus \{s\}$ which is called multicast group, where $|M| = m$. The path traversed by messages from the source s to a multicast receiver, m_i, is given by $P(s, m_i)$. Thus multicast routing tree can be defined as $T(s,M) = \bigcup_{m_i \in M} P(s, m_i)$, and messages are sent from s to destination of M using $T(s,M)$.

In the following we now introduce two important qualities of service metrics in multicast communications [17]. The multicast end-to-end delay constraint, $supD$, represents an upper bound on the acceptable end-to-end delay along any path from the source to a destination node. This metric reflects the fact that the information carried by the multicast messages becomes stale $supD$ time units after its transmission at the source. The multicast delay variation, δ, is the maximum difference between the end-to-end delays along the paths from the source to any two destination nodes.

$$\delta\ =\ max\{\ |\phi_D(P(s,m_i)) - \phi_D(P(s,m_j))|,\ ^{\forall}m_i, m_j \in M,\ i \neq j\ \}$$

The issue first defined and discussed in [17] is of minimizing multicast delay variation under multicast end-to-end delay constraint. The authors referred to this problem as Delay- and Delay Variation-Bounded Multicast Tree (DVBMT) problem. The DVBMT problem is to find the tree that satisfies

$$min\{ \ \delta_\alpha \ | \ \ ^\forall m_i \in M, \ ^\forall P(s,m_i) \in (2^{s \Rightarrow m_i}, supD), \ ^\forall P(s,m_i) \subseteq T_\alpha, \ ^\forall \alpha \in \Lambda \ \}$$

where T_α denotes any multicast tree spanning $M \cup \{s\}$, and is known to be NP-complete [17]. There are two well known approaches to construct multicast tree for the DVBMT problem. One is DVMA (Delay Variation Multicast Algorithm) [17] and the other DDVCA [18]. The algorithm DDVCA proposed by Pi-Rong Sheu and Shan-Tai Chen is based on the Core Based Tree (CBT) [2, 3]. It has been shown that DDVCA outperforms DVMA in terms of the delay variation of the constructed tree. Moreover, the time complexities of algorithms are $O(mn^2)$ and $O(klmn^4)$ for DDVCA and DVMA, where k and l is the number of paths at line 3 and line 11 in Fig. 3, Ref. [17], respectively; m represents the number of destination nodes, while n represents the number of nodes in the computer network.

3 Description of the Proposed Algorithm

In this section, we describe our proposed novel algorithm to construct multicast trees that is superior to DDVCA. In order to define a multicast tree, the basic idea of the proposed algorithm is based on CBT [2, 3]. The method used in CBT for the establishment of a multicast tree is first to choose some core routers which compose the backbone. We also select a core router addressed as a core node.

3.1 The Basic Concept

The goal of this paper is to propose an algorithm which produces multicast trees with low multicast delay variation. In this subsection, we present our proposed algorithm. The proposed algorithm consists of a core node selection part and the multicast tree construction part. Hence we take an interest in a core node selection. When candidate of core node is several nodes, the DDVCA randomly choose a core node among candidates but the proposed algorithm presents lucid solution. To implement the our idea, the following data structures are employed.

- *Input* : A directed graph $G(V, E)$, M is the multicast group with $m = |M|$, a source node s, a end-to-end delay bound $supD$
- *Output* : The multicast tree T such that $\phi_D(P(s, m_i)) \leq supD$, $^\forall P(s, m_i) \subseteq T$, $^\forall m_i \in M$, and has a small multicast delay variation
- *candidate* : the candidates of core node
- *compare* : the max difference delay between core nodes and visited destinations

- $Dij(m_k, v_i)$ = Calculate the minimum delay between m_k and v_i
- $pass(s, v_i, m_k)$: $Dij(s, m_k)$ when any destination node m_k is visited in the path from s to v_i
- $max_i = max\{Dij(v_k, v_i) \mid {}^{\forall}v_k \in M\},\ {}^{\forall}v_i \in V$
- $min_i = min\{Dij(v_k, v_i) \mid {}^{\forall}v_k \in M\},\ {}^{\forall}v_i \in V$
- $diff_i = max_i - min_i$

In selecting such a core node, we use the minimum delay path algorithm [6]. Fig. 1 checks whether any destination node is visited in the path from source node to each other node. If any destination node is visited, then the proposed algorithm records in '*pass*' data structure.

```
For ∀l ∈ {the minimum delay path from s to v_i} Do
/* l : the nodes in minimum delay path form s to v_i */
   If l = m_k, ∀m_k ∈ M
     then pass(s,v_i,m_k) = Dij(s,m_k)
     else pass(s,v_i,m_k) = 0
If diff_i < diff_min and Dij(s,v_k) + max_i ≤ supD
   then diff_min = diff_i ; c = i
```

Fig. 1. A partial amendment of the DDVCA - I

Fig. 2 conforms *supD* and select nodes with the minimum delay variation as the candidates of core node. Next, our algorithm chooses the core node with $min\{\ \phi_D(P(s, c_i)) - min\{\ pass(s, c_i, m_j)\ \}\ \}$ in Fig. 3.

```
For ∀v_i ∈ V  Do
   If diff_i = diff_c and Dij(s,v_i) + max_i ≤ supD
     then candidate = candidate ∪ v_i
```

Fig. 2. A partial amendment of the DDVCA - II

For $\forall c_i \in candidate$ Do

  If $pass(s, c_i, m_k) = 0,$ for every $m_k \in M$

    then $compare_i = Dij(s, c_i) - min\{pass(s, c_i, m_j) \mid \text{positive and } \forall m_j \in M\}$

$c = min\{i \mid compare_i\}$

Fig. 3. A partial amendment of the DDVCA - III

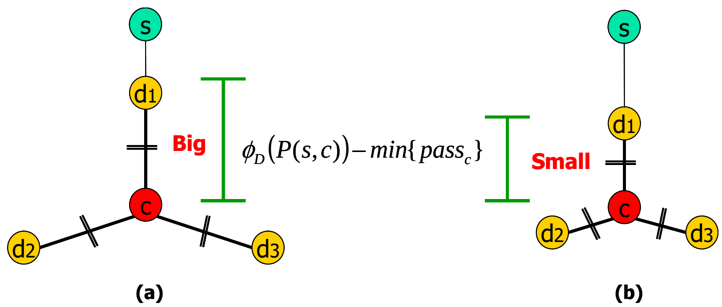


Fig. 4. The basic concept of the proposed algorithm

As you shown in Fig. 4, our algorithm overcomes DDVCA's weaknesses. In construction of a multicast tree, we follow the DDVCA. The time complexity of the proposed algorithm is $O(mn^2)$, which matches the complexity of the DDVCA.

3.2 An Illustration

In this subsection, we illustrate an example with explanation of the algorithm. Fig. 5 (a) shows a given network topology with link delays specified on each link. Suppose that the multicast end-to-end delay constraint $supD$ is 11. Fig. 5 (b) represents the ultimate multicast tree obtained by the DDVCA. Fig. 5 (c) shows the path constructed by the proposed algorithm.

From Table 1, we know that nodes with the smallest multicast delay variation are v_3, v_4, and v_8. However, since we must consider the delay bound $supD$, the node v_3 is ignored. The DDVCA randomly selects the node v_4, but the proposed algorithm selects the node v_8 as a core node. Because the proposed algorithm calculates the minimum among $compare_{v_4} = 8-4 = 4$ and $compare_{v_8} = 6-4 = 2$ in Fig. 3, we take the node v_8 as core node. Finally, the DDVCA's multicast delay variation is 7, but the proposed algorithm's multicast delay variation is 5. Fig. 6 shows that proposed algorithm chooses v_8 in Fig. 6 (b) in case of the same $diff_i$ value when it selects a core node.

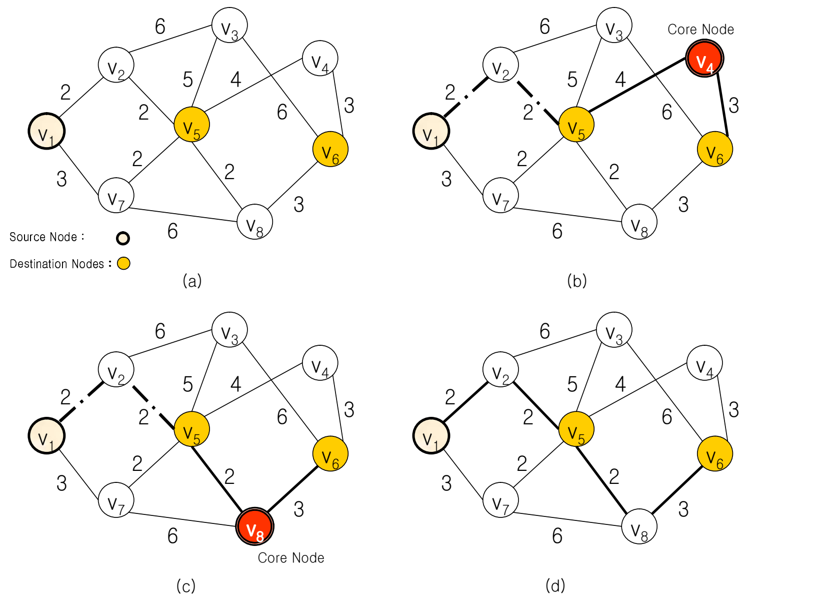


Fig. 5. (a) A Given network $G(V,E)$ and link delays are shown to each link, (b) DDVCA and $\delta_{DDVCA} = 7$, (c) Proposed Algorithm and $\delta_{Proposed} = 5$, (d) Optimal tree and $\delta_{Optimum} = 5$

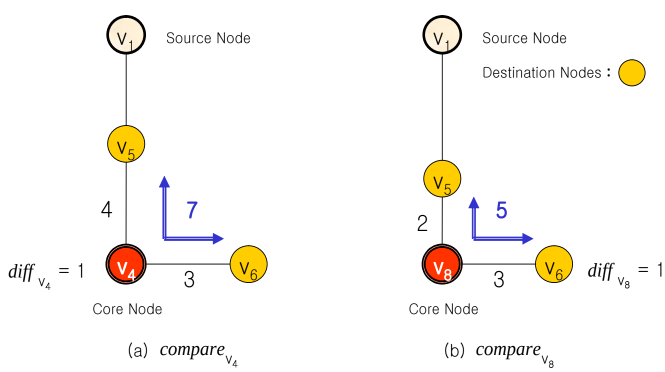


Fig. 6. The candidate nodes

4 Experimental Results

We compare our proposed algorithm with the DDVCA in terms of multicast delay variation. We describe the generation of random network topologies for the evaluation and the simulation results based on the network topology generated.

Table 1. The method by which proposed algorithm selects a core node

		v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8
source	v_1	0	2	8	**8**	4	9	3	**6**
pass	v_5	0	0	0	**4**	4	4	0	**4**
	v_6	0	0	0	**0**	0	9	0	**0**
destination	v_5	4	2	5	4	0	5	2	2
	v_6	9	7	6	3	5	0	7	3
max_i		9	7	6	4	5	5	7	3
min_i		4	2	5	3	0	0	2	2
$diff_i$		5	5	**1**	**1**	5	5	5	**1**
$compare_i$					4				**2**

4.1 Random Graph Generation and Simulation Parameters

The details of the generation for random network topologies are as follows. The method uses parameters n - the number of nodes in networks, and P_e - the probability of edge existence between any node pair [15, 16]. Let us remark that if a random graph models a random network then this graph should be connected. Hence, the graph should contain at least a spanning tree. So, firstly a random spanning tree is generated. As we know, we consider cases for $n \geq 3$. A tree with 3 nodes is unique, and thus we use this as an initial tree. And we expand to a spanning tree with n nodes. After adjusting the probability P_e, we generate other non-tree edges at random for the graph based network topology. Let us calculate the adjusted probability P_e^a. By $Prob\{event\}$ denote a probability of the event. Suppose e is a possible edge between a couple of nodes, then we have

$$P_e = Prob\{\ e \in spanning\ tree\ \} + Prob\{\ e \notin spanning\ tree\ \} \cdot P_e^a$$
$$P_e = \frac{n-1}{n(n-1)/2} + (1 - \frac{n-1}{n(n-1)/2}) \cdot P_e^a$$
$$\therefore\ P_e^a = \frac{nP_e - 2}{n-2}\ .$$

Let us describe a pseudo code for random network topologies. Here A is an incident matrix, r is a simple variable, and $random()$ is a function producing uniformly distributed random values between 0 and 1.

Graph Generation Algorithm

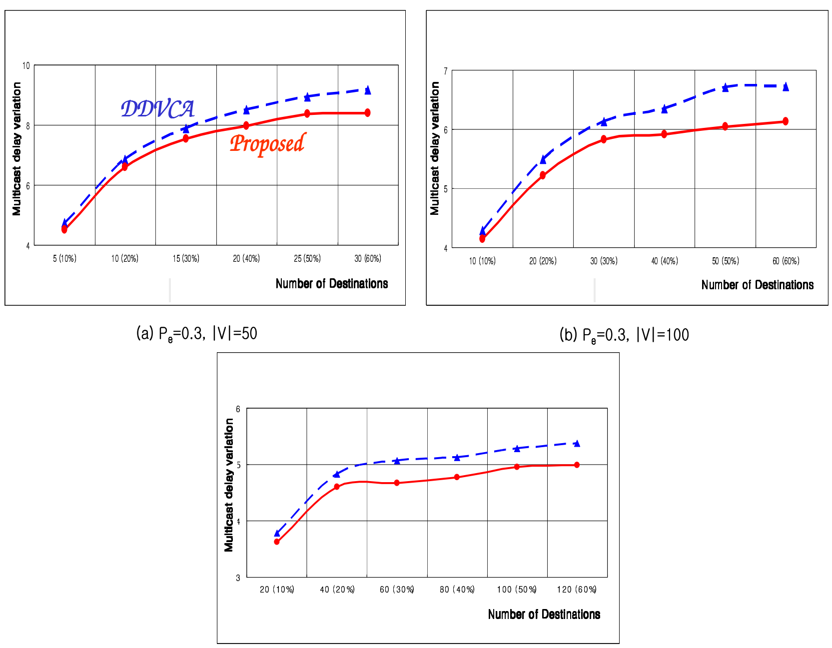
Begin
$A_{1,2} = A_{2,1} = A_{2,3} = A_{3,2} = 1$
For $i = 4$ to n **Do**
  $r = (i-1) \times random() + 1$
  $A_{r,i} = A_{i,r} = 1$
For $i = 1$ to $(n-1)$ **Do**
  For $j = (i+1)$ to n **Do**

If $P_e > random()$ **Then** $A_{i,j} = A_{j,i} = 1$

End Algorithm.

4.2 Discussion of Results

We now describe some numerical results with which we compare the performance of the proposed scheme. The proposed algorithm is implemented in C++. We randomly selected a source node. We generate 10 different networks for each size of given 50, 100, and 200. The destination nodes are picked uniformly from the set of nodes in the network topology (excluding the nodes already selected for the destination). Moreover, the destination nodes in the multicast group will occupy 10, 20, 30, 40, 50, and 60% of the overall nodes on the network, respectively. We randomly choose $supD$ such that $supD \geq min\{\phi_D(P_k(s,m)) \mid s \text{ is source node}, \ ^\forall m \in M, \ ^\forall k \in \Lambda\}$. Delays are uniformly random integer values between 0 and 10. We simulate 1000 times ($10 \times 100 = 1000$) for each $|V|$ and $P_e = 0.3$.



(a) P_e=0.3, |V|=50

(b) P_e=0.3, |V|=100

(c) P_e=0.3, |V|=200

Fig. 7. The multicast delay variations of the three different networks

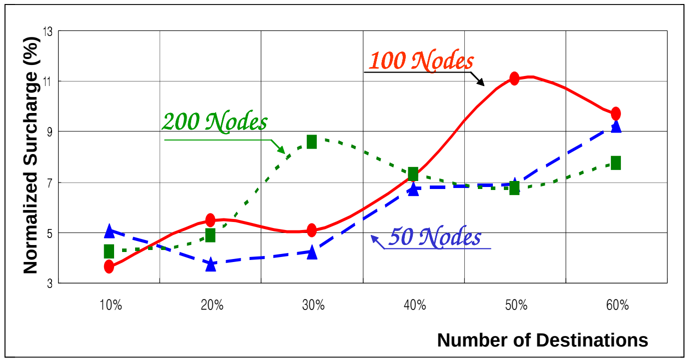


Fig. 8. Normalized Surcharges versus number of nodes in networks

For the performance comparison, we implement the DDVCA in the same simulation environment. We use the normalized surcharge, introduced in [11], of the algorithm with respect to our method defined as follows:

$$\bar{\delta} \;=\; \frac{\delta_{DDVCA} - \delta_{Proposed}}{\delta_{Proposed}} \quad .$$

In our plotting, we express this as a percentage, *i.e.*, $\bar{\delta}$ is multiplied by 100. Fig. 7 (a), (b), and (c) show the simulation results of multicast delay variations. As indicated in Fig. 8, it is easily noticed that the proposed algorithm is always better than the DDVCA. The enhancement is up to about 3.6%∼11.1% in terms of normalized surcharge for the DDVCA.

5 Conclusion

In this paper, we consider the transmission of a message that guarantees certain bounds on the end-to-end delays from a source to a set of destinations as well as on the multicast delay variations among these delays over a computer network. There are two well known approaches for constructing a multicast tree with the DVBMT problem, which is known to be NP-complete. The one is the DVMA [17]. Although it provides smart performance in terms of the multicast delay variation, its time complexity is as high as $O(klmn^4)$. As we all know, a high time complexity dose not fit in large scale high speed networks. The other is the DDVCA [18]. It has been shown that the DDVCA outperforms the DVMA slightly in terms of the multicast delay variation for the constructed tree. Moreover, the time complexity of the DDVCA is $O(mn^2)$. In the meantime, the time complexity of the proposed algorithm is $O(mn^2)$, which is the same as that of the DDVCA. Furthermore, the comprehensive computer simulation results show that the proposed scheme obtains the better minimum multicast delay variation than the DDVCA.

Future work is that we examine the problem of constructing minimum cost multicast trees which guarantees certain bounds on the end-to-end delay from the source to the destination nodes and the inter-destination delay variations between paths from the source to the destination nodes.

References

1. K. Bharath-Kumar and J. M. Jaffe, "Routing to multiple destinations in computer networks," IEEE Trans. Commun., vol. COMM-31, no. 3, pp. 343-351, March 1983.
2. T. Ballardie, P. Francis, and J. Crowcroft, "Core based trees (CBT) : An architecture for scalable inter-domain multicast routing," Computer Commun. Rev., vol. 23, no. 4, pp. 85-95, 1993.
3. A. Ballardie, B. Cain, and Z. Zhang, "Core Based Trees (CBT Version 3) Multicast Routing," Internet draft, 1998.
4. Y.-C. Bang and H. Choo, "On multicasting with minimum costs for the Internet topology," Springer-Verlag Lecture Notes in Computer Science, vol. 2400, pp.736-744, August 2002.
5. S. Deering et al., "Protocol Independent Multicast-Sparse Mode (PIM-SM): Motivation and Architecture," Internet draft, 1998.
6. E. W. Dijkstra, "A note on two problems in connexion with graphs," Numerische Mathematik, vol. 1, pp. 269-271, 1959.
7. D. Estrin et al., "Protocol Independent Multicast (PIM) Sparse Mode/Dense Mode," Internet draft, 1996.
8. M. R. Garey, R. L. Graham, and D. S. Johnson, "The complexity of computing steiner minimal trees," SIAM J. Appl. Math., vol. 32, no. 4, pp. 835-859, June 1977.
9. E. N. Gilbert and H. O. Pollak, "Steiner minimal tree," SIAM J. Appl. Math., vol. 16, 1968.
10. S. L. Hakimi, "Steiner's problem in graphs and its implication," Networks, vol. 1, pp. 113-133, 1971.
11. V. P. Kompella, J. C. Pasquale, and G. C. Polyzos, "Multicast routing for multimedia communication," IEEE/ACM Trans. Networking, vol. 1, no. 3, pp. 286-292, June 1993.
12. L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for steiner trees," Acta Informatica, vol. 15, pp. 141-145, 1981.
13. J. Moy, "Multicast Extension to OSPF," Internet draft, 1998.
14. T. Pusateri, "Distance Vector Routing Protocol", draft-ietf-idmr-dvmrp-v3-07, 1998.
15. A.S. Rodionov and H. Choo, "On generating random network structures: Trees," Springer-Verlag Lecture Notes in Computer Science, vol. 2658, pp. 879-887, June 2003.
16. A.S. Rodionov and H. Choo, "On generating random network structures: Connected Graphs," International Conference on Information Networking 2004, Proc. ICOIN-18, pp. 1145-1152, February 2003.
17. G. N. Rouskas and I. Baldine, "Multicast routing with end-to-end delay and delay variation constraints," IEEE JSAC, vol. 15, no. 3, pp. 346-356, April 1997.
18. P.-R. Sheu and S.-T. Chen, "A fast and efficient heuristic algorithm for the delay- and delay variation bound multicast tree problem," Information Networking, Proc. ICOIN-15, pp. 611-618, January 2001.

19. H. Takahashi and A. Matsuyame, “An approximate solution for the steiner problem in graphs,” Mathematica Japonica, vol. 24, no. 6, pp. 573-577, 1980.
20. B. Wang and J. C. Hou, “Multicast Routing and its QoS Extension: Problems, Algorithms, and Protocols,” IEEE Networks, Jan./Feb, 2000.
21. B. W. Waxman, “Routing of multipoint connections,” IEEE JSAC, vol. 6, no. 9, pp. 1617-1622, December 1988
22. Q. Zhu, M. Parsa, and J. J. Garcia-Luna-Aceves, “A source-based algorithm for near-optimum delay-constrained multicasting,” Proc. IEEE INFOCOM’95, pp. 377-385, March 1995.