A Delay and Delay Variation Constraint Multicast Algorithm in Heterogeneous Network

Mohamed Aissa[1], Adel Ben Mnaouer[2], Abdelfettah Belghith[3]
[1]University of Nizwa
Nizwa, Sultanate of Oman
m.issa@unizwa.edu.om

[2]Dar Al-Uloom University
Saudi Arabia
adelm@dau.edu.sa

[3]Hana Research Group, ENSI
University of Manouba - Tunisia
abdelfattah.belghith@ensi.rnu.tn

ABSTRACT: *In this paper, we study the problem of QoS group communication in a heterogeneous network, which consists of multiple MANETs attached to the backbone Internet. We propose a heuristic multicast algorithm called DDCMA (Delay and Delay Variation Constraint Multicast Algorithm). DDCMA is designed for solving the DVBMT (Delay- and delay Variation- Bounded Multicast Tree) problem, which has been proved to be NP-complete. The literature studies consider only end-to-end delay bound and minimizing delay variation. In this paper, we improve and extend previous well known from literature heterogeneous network algorithms to provide scalable and stable multicast services on the Internet. The algorithm defines QoS parameters as constraints on both delay and delay variation. Furthermore, we introduce a new Delay–Variation Estimation Scheme for heterogeneous networks, which can help DDCMA achieve better performance in terms of the multicast delay variation than some well known algorithms. Theoretical analysis is given to show the correctness of DDCMA and its performance in terms of the multicast delay variation.*

Keywords: Multicast delay variation, Heterogeneous network, QoS group communications, DDVCA, DDVMA

Received: 28 December 2010, Revised 4 February 2011, Accepted 18 February 2011

1. Introduction

The explosive growth of mobile communications has attracted interests in the integration of wireless networks with wireline ones and the Internet in particular. Providing mobile users the wireless access to the Internet is of major interests in today's research in networking [1]. For MANETs, extensive work has been done on extending IP connectivity of the mobile hosts (MHs) [1]. The heterogeneous network architecture will facilitate the current trend of moving to an all-IP wireless environment. In the heterogeneous network, gateway is a fixed node connecting an MANET to the Internet and each gateway serves only one MANET. Gateways forward data packets and relay them between MANETs and the Internet. When an MANET is connected to the Internet, it is important for the MHs to detect available gateways providing access to the Internet. Therefore, a gateway discovery mechanism is required [1]. Lots of efforts have been devoted to the problem of Internet gateway discovery and gateway forwarding strategies [2][3]. These works have provided the foundation for the practical implementation of our algorithm.

Without loss of generality, we assume a group communication scenario in the heterogeneous network: one team leader is the source and wants to multicast messages to several other team leaders called destinations. The source will transmit the messages to its Internet gateway called the source gateway using AODV protocol. Then the source gateway will transmit the messages to all the destination gateways using a multicast tree in the backbone Internet. Finally, each destination gateway will forward the messages to the destination in its MANET also using AODV protocol, separately [1].

Delay is a significant factor in a heterogeneous network and is taken as a constrained metric of the presented algorithm. In addition, *end-to-end delay* is definitely used rather than average delay or total of the whole tree, because each user is mostly concerned to receive information from the source as soon as possible. Besides, *inter-destination delay variation* is paid attention in this paper as well. It is an important factor in this situation. It is necessary that every participant to receive information from the source at the same time so that the fairness is guaranteed. There are several situations in which we need to limit the variation among the path delays by a certain given maximum bound. During a teleconference, it is important that a speaker is heard by all participants at the same time; otherwise, the communication may lack the feeling of an interactive face-to-face discussion.

In [4], Sheu and Chen studied the problem of the minimization of multicast delay variation under the multicast end-to-end delay constraint. As a result, they proposed the DDVCA (Delay and Delay Variation Constraint Algorithm) algorithm coming from the Core Based Tree (CBT) [5] and the minimum path algorithm [6]. It has a complexity of $(O|E||V|^2)$ (where V - the set of vertices and E - the set of edges in a graph). When a candidate of core node is several nodes, the DDVCA *randomly* choose a core node among candidates. Here, the delay variation was only minimized.

In [1], the authors proposed a heuristic multicast algorithm called DDVMA (Delay and Delay Variation Multicast Algorithm). DDVMA is designed for solving the DVBMT (Delay- and delay Variation-Bounded Multicast Tree) problem [7], which has been proved to be NP-complete. It can find a multicast tree satisfying the multicast end-to-end delay constraint and minimizing the multicast delay variation. Two concepts- the proprietary second shortest path and partially proprietary second shortest path are introduced, which can help DDVMA achieve better performance in terms of the multicast delay variation than DDVCA [4]. Unfortunately, in their heuristic, the delay variation was only minimized and not constrained.

In [8], an algorithm based on CBT [5] and having a complexity of $(O|E||V|^2)$ was proposed. It produces multicast trees with low multicast delay variation. The algorithm consists of two parts. In the first part, a core node is selected. In the second part, a multicast tree is constructed. The simulation results show that the proposed scheme obtains the better minimum multicast delay variation than the DDVCA.

In [9], the authors have considered the problem of determining a multicasting sub-network with end-to-end delay bound, and the tightest delay variation for multimedia applications on overlay network. Then they have presented their heuristic Chains, which achieves the tightest delay variation for a given delay bound. At the initial phase of their heuristic, they have used k shortest path technique proposed by [6] to find the paths for each destinations for which the delays are less than or equal to Δ. Then using these delays, they have determined the delay chain, which gives the minimum delay variation and constructed the multicasting sub-network by retrieving the paths from the delays. Unfortunately, in their heuristic, the delay variation was only minimized and not constrained.

Although, all the previous algorithms are attractive, we think that it is more important to constraint the delay-variation than minimize it. Consequently, we address the problem of improving the previous algorithms by constraining the delayvariation and decreasing their time complexity.

The main contribution of our work is the discovery of a simple yet effective heuristic that exhibits very good performance and that can be implemented in a wide range of heterogeneous networks. Furthermore, we extend DDVCA, DDVMA, Kim's algorithms and Chain by *(a)* adding the delay-variation constrain (rather than minimize it), *(b)* proposing an algorithm with lower time complexity and *(c)* introducing a new delay-variation estimation method, which to our best knowledge is the first to be adopted in heterogeneous networks.

Our results show that our algorithm significantly outperforms heuristics that are based on random selection such as DDVCA and by its lower complexity, outperforms DDVCA, DDVMA and Kim's algorithm.

In the remainder of this paper, section 2 will present the multi-constrained heterogeneous network problem definition. In section 3, we propose a new delay-variation estimation scheme. In section 4, we propose our Delay and Delay- Variation constrained Multicast algorithm (DDCMA). This section also includes some theorems and comparison of our proposed algorithm with some well-known algorithms from the literature. In section 5, we prove the correctness and time complexity analysis of our DDCMA algorithm. Finally, section 6 concludes the paper.

Next, we formulate our network model as it was defined in [1].

2. Network Model And Problem Specifications

The backbone network can be modeled as a weighted digraph $G(V,E)$, where V represents the set of nodes including gateways, and E represents the set of links between the nodes. For each link $l \in E$, a *link-delay* function $D: l \rightarrow r+$ is defined. A nonnegative value $D(l)$ represents the transmission delay on link l. Multicast messages are sent from the leader MH of the source MANET. Messages are first forwarded to the source gateway $v_s \in V$ through the route discovered by AODV, then arrive at a set of destination gateways $Z \in \square V-\{v_s\}$ through the multicast tree T in the backbone network, and finally are forwarded to the leader MHs of the destination MANETs through the wireless routes between each destination gateway and each leader MH, respectively. To guarantee the QoS of group communication, the multicast end-to-end delay between the leader MH of the source MANET and the leader MH of each destination MANET should not exceed the multicast endto- end delay constraint ", and the multicast delay variation among the leader MHs of destination MANETs should not exceed a tolerance δ. Let $P_T(v_s,v_w)$ denote the path from the source gateway v_s to a destination gateway $v_w \in \square Z$ on T. Then the transmission delay between v_s and v_w on T is defined as

$$Delay[v_w] = \sum_{l \in P(v_s,v_w)} D(l) \tag{1}$$

For each gateway $g \in \square \{s\} \cup Z$, we define a *gateway-delay* function $W: g \rightarrow r^+$. It assigns gateway g a nonnegative value $W(g)$, which represents the delay of the wireless route discovered between gateway g and the leader MH of the MANET g serves.

In our paper, the problem of QoS group communications in the heterogeneous network model is to find an optimal multicast tree $T^*(V_{T^*},E_{T^*}), \square \{vs\} \cup Z \subseteq V_{T^*}, E_{T^*} \subseteq E$, satisfying:

$$W(v_s) + \sum_{l \in P(v_s,v_w)} D(l) + W(v_w) \leq \Delta, \tag{2}$$

$$\left| \left(\sum_{l \in P_T(v_s,v_u)} D(l) + W(v_u) \right) - \left(\sum_{l \in P_T(v_s,v_w)} D(l) + W(v_w) \right) \right| < \delta, \quad \forall v_u, v_w \in Z \tag{3}$$

where T denotes any multicast tree spanning v_s and Z in $G(V,E)$.

If we assume $W(g)=0$ for each $g \in \square \{s\} \cup Z$, the problem turns to be the DVBMT problem, which has been proved to be an NP-complete problem [7]. Our problem is also NPcomplete because it contains, as a special case, the DVBMT problem. Hence, only heuristic algorithms can be developed for our problem. In this paper, our work focus on DVBMT problem in the heterogeneous networks, which consists of several MANETs attached to the backbone Internet.

3. The Proposed Delay Variation Estimation Method

To estimate the delay variation in the heterogeneous network, we propose an estimation based on the average-delay method as mentioned in [10]. For that purpose, the delay from the source v_s to each destination v_i, $Delay[v_i]$, is computed.

The network average-delay $AvDelay(T)$ can be calculated as:

$$AvDelay(T) = \frac{\sum_{v_i \in Z} (Delay[v_i] + W(v_i))}{|Z|} \tag{4}$$

where $|Z|$ the size of the multicast group (destination gateways), $W(v_i)$ the delay of the wireless route discovered between gateway v_i and the leader MH of the MANET v_i serves. Consequently, constraint (4) is substituted by the following simpler constraint:

$$\left| AvDelay(T) - (Delay[v_u] + W(v_u)) \right| \leq \delta, \quad \forall v_u \in Z \tag{5}$$

The delay and delay variation bounds are two conflicting objectives. The delay constraint dictates that short paths must be used. However, choosing short paths may lead to a violation of the delay variation constraint among nodes which are close to the source and nodes which are far away from it. Consequently, it may be necessary to select longer paths for some nodes in order to satisfy the latter constraint. A balance must be struck between the two constraints. Consequently, we address the problem of designing a multicast routing algorithm that overcomes these conflicts. For this purpose, we give the following observation:

Observation 1

To omit some short paths leading to the destination nodes, and select long paths instead, we propose a new computation for the average delay in an heterogeneous network as follows:

$$AvDelay(T) = \begin{cases} \Delta, & \text{if } Delay[v_i] + W(v_i) \leq \delta, \quad \forall v_u \in Z \\ \dfrac{\sum_{v_i \in Z} (Delay[v_i] + W(v_i))}{|Z|}, & \text{otherwise} \end{cases} \tag{6}$$

where *s* and *vi* the source and destination node respectively, Δ is the user's end-to-end delay bound, δ is the user delay variation tolerance and $|Z|$ the size of the multicast group. Hence (6) replaces (4) in (5). Next, we give a formal definition of our proposed DDCMA algorithm.

4. DDCMA Algorithm

Similar to DDVCA [4] and Kim's algorithms [8], our DDCMA algorithm **(**Fig. 1) basically comes from CBT [5], and the Dijkstra's shortest path algorithm [11]. CBT establish a multicast tree by choosing some Core Routers, which compose the Core Backbone. Afterwards, all node operations relating to join and leave the multicast group are based on issuing a request toward an appropriate Core Router. In our DDCMA algorithm (as in DDVCA), we select a Core Router addressed as a central node.

We denote *a core-selection algorithm* as *delay-bounded*, if the algorithm considers a given delay-bound for the group during the selection process, and the resulting core is such that there exists a path between each source-receiver pair in the group *which passes through this core* without violating the delay-bound [12]. Furthermore, we denote *a core-selection algorithm* as *delay-variation bounded,* if the algorithm considers a given delay variation tolerance for the group during the selection process, and the resulting core is such that the difference between the end-to-end delays along the paths from the source *s* to any two- destination nodes, which passes through this core, satisfies the delay variation tolerance.

Motivated by the simulation results provided in [13], we adopt a strategy similar to "Topological Center of Z in Z", which dictates that core candidates are restricted to be *multicast group members*. In link-state routing-based networks, where centralized algorithms can be used by individual routers, this restriction reduces the computational overhead to (O$|M|V|^2$) [13]. In our algorithm, rather than using topological center, we use a *QoS parameter center*. That is, based on the delay, we select a such core node among destination nodes having the least delay average. That is, by this value, it is situated the nearest to all the remainingdestination nodes in the *LDT*. It is as if it is situated at the of the remaining destination nodes. We will use the term *center member*, as a synonym for such a destination core node in our heuristic.

To avoid message retransmission and alleviate the network traffic, we adopt a strategy based on the hypothesis that if a message passes through a destination node first, then it is received immediately by this node. Thus, we avoid the needed time to reach the core node and then this core relays the message to each of the destination nodes.

Our DDCMA algorithm (Fig. 1) contains five stages. The first stage (lines 2-4) is the initialization. The second one (line 6), during which the Least Delay Tree (*LDT*) is computed by using Dijkstra's algorithm [11]. Subsequently, the user input data are verified (not shown in the algorithm for simplicity). If these data are too tight, then they are relaxed. The third one (lines 7-12) is the computation of the delay average in order to form an ordered set of candidate center members. This phase is described next. The fourth stage (lines 13-28) constitutes the center member selection verifying both delay and delay variation constraints. The fifth algorithm phase (lines 29-36) represents the multicast tree construction process. Next, we describe the third phase.

4.1 Proposed center member selection method

The center member selection is executed into two stages:

4.1.1Center member pre-selection method

The pre-selection operation (lines 7-12) consists in finding for every destination node v_i the following results: *(a)* The delay between v_i and every destination node v_j through the shortest delay and in *LDT* and then summate all these values. (line 9); and *(b)* the average delay value computed as follows (line 10)**:**

$$AverageD\left[v_i\right] = \frac{\sum\limits_{\substack{j=1 \\ j\neq i}}^{|Z|-1} D\left[v_i, v_j\right] + \sum\limits_{\substack{j=1 \\ j\neq i}}^{|Z|-1} W_j}{|Z|}, \qquad (7)$$

$$\text{where } v_i \text{ and } v_j \in Z \text{ and } v_i \neq v_j$$

The *AverageD* is calculated for every destination node v_i and then introduced in a priority queue *Q* (line 11).

Subsequently, *Q* is sorted in an increasing order (line 13) such that the first node *Q* has the least *AverageD* and therefore, has the highest priority. The first element is picked up (line 14) and is served as a first test in the first loop selection method execution, which we describe next.

4.1.2 Center member selection method

The center member selection constitutes the fourth stage algorithm execution. At this stage (lines 13-28), we test whether for the picked node from *Q,* the Dijkstra's shortest path from the source to any destination node passing through this picked node satisfies the delay bound Δ (3) and the delay variation tolerance (9). If it does, then it is selected as a center member ($v_c=v_i$) (line 26). Therefore, there is no need to treat the other remaining nodes in Q. Otherwise, we pick from *Q* the next candidate (line 20) and the same constraints (3) and (9) are tested. If all nodes in *Q* are treated and no one verifies the delay bound and the delay variation tolerance, then the source *s* is considered as the only candidate ($v_c=s$) (line 21). The fifth algorithm phase (lines 29-36) represents the multicast tree construction process. We first connect the source node with the center member *vc* (line 31), and then we connect to this center member all the remaining destination nodes (lines 32-33). If the source is selected as a center member, we apply Dijkstra's delay shortest path algorithm to compute the delay and delay variation bounded multicast tree rooted at the source *s* and spanning all destination nodes (line 36).

Observation 2

In our proposed DDCMA algorithm, *AvDelay* (*T*) and *AverageD[vi]* are updated when a new member joins or leaves the multicast group *Z.*

4.2 DDCMA algorithm operation

A detailed example in Fig. 2 is provided to show how the DDCMA algorithm works on the original graph depicted in Fig. 2(a). [1][4][9] used the same graph with the same settings but without delay variation constraint. We applied Kim's Algorithm on this graph. Our resulting multicast tree is shown in Fig. 2(c). In the original graph, each number *d* of numbers along any edge,

Input: *a backbone network $G=(V,E)$, a set of destination nodes (gateways) Z, a source node s, an upper bound Δ of end-to-end delay. δ- Delay variation tolerance.*

Output: *a delay and delay-variation bounded network$T=(V_T , E_T)$ $(T \subseteq G)$ spanning all nodes in Z.*

Step1 *Using AODV protocol to get the delay of the wireless routing path between each team leader and its gateway.*

Step2 *Multicast tree construction in the backbone Internet.*

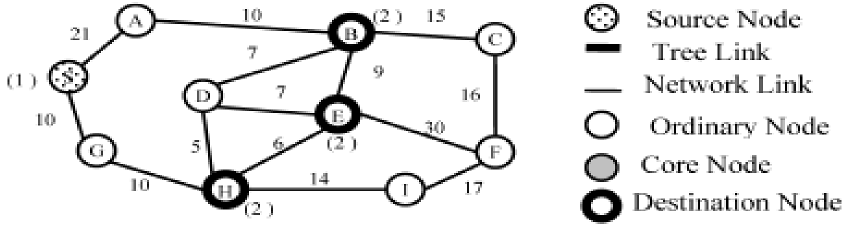
```
1  DDCMA( $G(V, E), s, Z, \Delta, \delta, D, W$ )
2    Begin
                                        /*   Initialization   */
3            $Delay[s]=0; T=\phi; Q \leftarrow \phi ; v_c \leftarrow \phi;$
4          for each vertex $u \in V$ -{s} do
5              {  $Delay[u]= \infty ;  AverageD[u]= \infty ;$ }
                     /* Computing the Least -Delay Tree (LDT)*/
6       Call Dijkstra's algorithm to compute the least delay tree (LDT). Then Compute AvDelay[T].
    /*For this LDT, the following loop computes D(vi,vj) of all paths connecting each destination node vi with another destination $v_j$,
      and it calculates the average delay for each destination node vi using (7) */
7        For each $v_i \in Z$ do {
8                     for each $v_j \in Z$ do
9                     {$Delay[V_j] \leftarrow Delay[V_i] + [V_i ,V_j] + W[V_j]$}
10            $AverageD[V_i] \leftarrow Delay[V_j] /Z$ ;
11           $Q \leftarrow AverageD[V_i]$; }
12              }
                               /*  Core node   selection process  */
13  Sort  Q in an increasing order
    /* Node with the least DELAY average is put the first in the queue */
                              $V_i \leftarrow Q$
14        /* Pick from Q the first destination node with the minimum AverageD. This node represents a candidate for a core node $v_c$ */
15      for each $v_j \in Z$ do
16        {
17           $Delay[V_j] \leftarrow Delay[V_i] + [V_i ,V_j] + W[V_j]$ ;
18            if $Delay[V_j] > \Delta$ and $|AVDelay[T]- Delay[V_j]| > \delta$
19            then
20               {  if $Q \neq \emptyset$ then {$Q \leftarrow Q \{V_i\}$}
21                 else {  $v_c$=s ;  Exit (step 35); }
22                          /* The source node is selected as a core node and exit from the loop For each*/
23            }
24             else    /* The destination node verifies the delay and delay variation bounds */
25             {
26                $v_c = v_i$;  /* $v_i$ is selected as a core node as it is validated */
27                                                    }
28   }  /* for loop end */

                               /*  Multicast Tree construction process  */s
29  if $v_c \neq s$
30     {
31        $T = T \cup \{L / L \in$ minimum delay path from s to $v_c\}$
32        for each $v_j \in Z$ do
33         $T = T \cup \{L / L \in$ minimum delay path from $v_j$ to $v_c$ $\}$
34     }
35   else          /* The source node is selected as a core node */
36   Call Dijkstra's Algorithm to compute the delay bounded and delay-variation constrained
     multicast tree spanning $Z \cup \{s\}$ and rooted at s.
37        return  T;
38  end  (of the Algorithm)
```

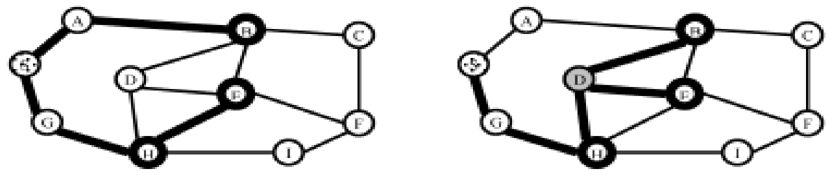
Figure1. Our DDCMA Algorithm

represent the delay *(d)* for that edge. *s* is set to be the source node. The delay bound Δ is set to 60 (as in [5]), the delay-variation tolerance ´ is set to 15 (our input data), and the set of destination nodes *Z* is set to: Z=*{B,E,H}*. The number in the parentheses near gateway g (including the source gateway and all the destination gateways) represents the corresponding wireless route delay W(g). Because the wireless route delay between the source leader MH and the source gateway is 1, th multicast end-to-end delay constraint used in DDCMA will be 59 (i.e., 60-1). Table 1 shows the procedure of selecting a central node in our algorithm DDCMA

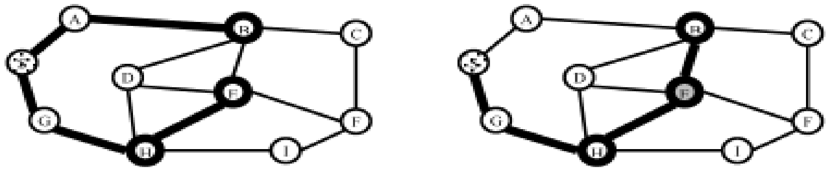
In order to find the average-delay of the multicast tree and verify the validity of the user's input data, the least delay tree (*LDT*) is computed using Dijkstra's shortest path algorithm. The result is shown in Fig. 2(b). In *LDT*, the delay of each destination node from the source is as follows: *Delay[s,B]*=31, *Delay[s,E]*=26, *Delay[s,H]*=20. Consequently the maximum delay is $\Delta_{max} = \Delta_T = 31$. As the user-defined delay bound Δ is greater than Δ_T , then it is accepted. The average delay is calculated using (3) as:



(a) The Original Graph

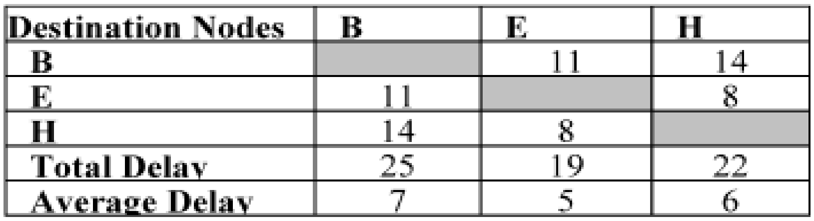


(b) LDT (d) DDVCA, DDVMA



(f) Chains (e) DDCMA (Our Algorithm), Kim's Algorithm,

Figure 2. Comparison Between DDCMA and Other Algorithms



Destination Nodes	B	E	H
B		11	14
E	11		8
H	14	8	
Total Delay	25	19	22
Average Delay	7	5	6

Table 1. Core Selection Method Using Average Delay

Destination Nodes	DDVCA	DDVMA	DDCMA
B	26	26	10
E	20	20	1
H	8	2	5

Table 2. Node selection Process In DDVCA, DDVMA And DDCMA

$$AvDelay(T) = \frac{Delay[s,B] + Delay[s,E] + Delay[s,H] + W_B + W_E + W_H}{3}$$
$$= \frac{31 + 26 + 20 + (2 + 2 + 2)}{3} = \frac{83}{3} \approx 27$$

Then the delay variation for every destination node is calculated using (4) is as follows:

$$\delta_B = |27\text{-}31| = 4,\ \delta_E = |27\text{-}26| = 1,\ \delta_H = |27\text{-}20| = 7.$$

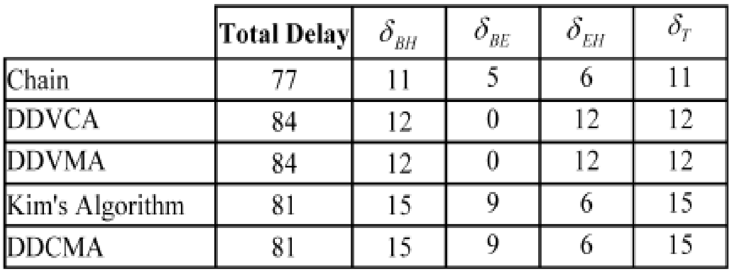
As a result, we have:
$\delta_{max} = \delta_T = \max\{\delta_B, \delta_E, \delta_H\} = \max\{1, 4, 7\} = 7$. As $\delta > \delta_T$ then our input data is also accepted. The execution is then transmitted to the following phase related to the formation of the ordered set of center member candidates. For that purpose, we apply (11). The results are shown in Table.1. According to the average delay values mentioned in this table, we form in an increasing order a priority Q={E,H,B}. We pick from Q the first destination node E, for which, we calculate the following delays and delay variation tolerances:

Delay[s,B]=Delay[s,E]+ D[E,B]+WB= 26+9+2=37,

Delay[s,E]=Delay[s,H]+ D[H,E] +WE = 20+6+2=28,

Delay[s,H]= 20+2=22, $\delta_B = |27\text{-}37| = 10 < \delta$,

$\delta_E = |27\text{-}28| = 1 < \delta$, $\delta_H = |27\text{-}22| = 5 < \delta$.



	Total Delay	δ_{BH}	δ_{BE}	δ_{EH}	δ_T
Chain	77	11	5	6	11
DDVCA	84	12	0	12	12
DDVMA	84	12	0	12	12
Kim's Algorithm	81	15	9	6	15
DDCMA	81	15	9	6	15

Table 3. Comparision Between DDCMA And Other Algorithms

As the destination node E satisfies the delay bound and the delay variation tolerance calculated from the source and to every destination node passing through node E, then E is selected as a center member. We connect the source node with E and subsequently, we connect all the destination nodes to it. The resulting tree is shown in Fig.2 (e).

Observation 3

In Table 2 we registered the calculated delay variations for the corresponding nodes B,E and H according to Table 1 and Table 2 in [1], along with our data. As it is observed, we can conclude that our selected core node can achieve the multicast tree with smaller multicast tree delay variation than DDVCA and DDVMA. Consequently, we can conclude that our method is more efficient.

4.3 Comparison with Other Algorithms

In Table 3, we compare the execution of the mentioned algorithms on the original graph depicted in Fig. 2 (a). The delay bound Δ is set to 60, the delay-variation tolerance δ is set to 15, and the set of destination nodes M is set to: $M=\{B,H,E\}$. In this table, we calculate the delay variation between every pair of destination nodes using (4). Then the maximum delay variation tolerance δ_T is fixed and calculated for every tree as follows:

$$\delta_T = \max_{v_i, v_j \in M} \left\{ \left| Delay[v_i] + W_i - \left(Delay[v_i] + W_j \right) \right| \right\} = \max\{\delta_{BE}, \delta_{EH}, \delta_{BH}\}$$

It is to be noticed that the tree constructed by our DDCMA algorithm is similar to that constructed by the Kim's algorithm. Both trees have the least total delay and the least delay variation tolerance and are *feasible trees* (these trees verify both delay and delay variation constraints).

5. Correctness Proof And Time Complexity Analysis of The DDCM A Alogorithm

We firstly prove the correctness of our DDCMA algorithm and thereafter, we prove its time complexity.

5.1 Correctness Proof

The correctness of the algorithm DDCMA results from the following theorem:

Theorem 1

The algorithm DDCMA always constructs a delay and delay variation-bounded multicast tree if such a tree exists.

Proof

Our algorithm tests whether the path from the source node to any destination node and incorporating the center member does violate neither the delay nor the delay variation constraints. If it does, then the center member candidate will be rejected and another one is extracted from *Q* and tested. From Fig.1 (line 36), it is obvious that the destination nodes situated on the path to the core node do not receive a feedback information from the core node. This fact decreases rather than increases the end-to-end delays. Therefore, the multicast tree T obtained by DDCMA definitely satisfies the multicast end-to-end delay constraint Δ. However, a center member may not be obtained and consequently the multicast tree either when none of the nodes satisfies the multicast end-to-end constraint. Here, it is claimed that when as long as there exists a multicast tree in the network satisfying the delay bound and the delay variation tolerance, DDCMA will absolutely capable of finding such a multicast tree. Since the entire destination nodes are checked one by one in line 17 of DDCMA and if no one is selected, then a source node *s* is also likely to be a centre member. In such a situation, a multicast tree is rooted at *s* and spanning all destination nodes through Dijkstra's shortest delay path. Obviously, if such a multicast tree still fails to meet the delay bound and the delay variation tolerance, we can conclude that a multicast tree, which satisfies the constraints regulated by the inputs, does not exist. That is, these inputs are too tight and should be relaxed.

5.2 Time complexity analysis

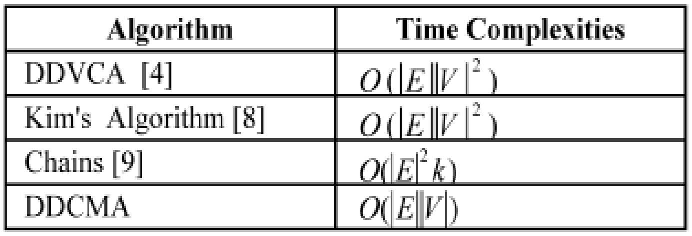
In the following, we analyze the time complexity of our proposed DDCMA algorithm.

Theorem 2

The time complexity of DDCMA is $O(|E||V|)$.

Proof

The DDCMA execution time is mainly spent on the loops. The initialization (Lines 3-5) takes $O(|V|)$. The line 6 calls the Dijkstra's algorithm. The time complexity of computing the *LDT* by Dijkstra's shortest path algorithm is $O(|E|\log|V|)$ assuming that the priority queue *Q* is implemented as a binary heap. Line 9 calculates the delay between any $v_i \in Z$ (where *E* initially contains *Z*) and every $v_j \in Z(v_i \neq v_j)$. This requires $O(|V|)$. Since this operation is executed once for every $Z\ v_i$ " , the time complexity of Lines 7-9 is $O(|Z||V|)$. Lines 13 through Line 28 select the center member. Since $O(|Z|)$ is required in Line 15, the time complexity, therefore of Lines 15-28 is $O(|Z|)$. The loop in Line 32 connects all the destination nodes to the center member. As Line 33 has a time complexity $O(|V|)$ the overall time complexity of Lines 32-33 is $O(|Z||V|)$. As a result, the time complexity of DDCMA is $O(|E|\log|V|) + 2\,O(|Z||V|) + O(|Z|) + O(|V|) = O(|E||V|)$. Table 4, proves that our DDCMA algorithm has better complexity than others well-known algorithms to which it is compared. In this Table, *E, V* and *Z* are as mentioned before, *k*-number of shortest paths.



Algorithm	Time Complexities
DDVCA [4]	$O(\|E\|\|V\|^2)$
Kim's Algorithm [8]	$O(\|E\|\|V\|^2)$
Chains [9]	$O(\|E\|^2 k)$
DDCMA	$O(\|E\|\|V\|)$

Table 4. Alogrithm Complexities

6. Conclusion

In this paper, we considered the problem of generating minimum delay multicast trees that satisfy 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 in a heterogeneous network. These constraints are imposed by the user process. Furthermore, extending previous works, we have proposed a new delay-variation estimation. This scheme is adjusted dynamically in response to the connection of new destination nodes. Therefore, based on the combination of CBT and the Dijkstra's shortest path algorithm, we proposed DDCMA with much lower time complexity $O(|E||V|)$ than DDVCA, DDVMA and Chains.

References

[1] Cheng, H. , Cao, J., Wang, X., Mullai, S(2005). A heuristic multicast algorithm to support QoS group communications in heterogeneous network, *In*: Proc. IEEE QShine.

[2] Ghassemian, M.., Hofmann, P., Prehofer, C. , Friderikos, C., Aghvami, H (2004). Performance analysis of Internet gateway discovery protocols in ad hoc networks, *In*: Proc. IEEE WCNC, p.120-125.

[3] Ruiz, P. M., NÌmez-Skarmeta, A. F. G (2004). Enhanced Internet connectivity for hybrid ad hoc networks through adaptive gateway discovery, *In*: Proc. IEEE LCN, p. 370-377.

[4] Sheu, Pi-Rong., Chen**,** Shan-Tai (2002)**.** A fast and efficient heuristic algorithm for the delay- and delay variation-bounded multicast tree problem, *Computer Communications* , 25 (8) 15 May 2002, p. 825- 833.

[5] Ballardie, A. (1996). Core Based Tree (CBT) multicast routing architecture, *Internet RFC 2201*, Sept.

[6] Prim, R. (1957). Shortest Connection Networks and Some Generalizations, *Bell Systems Tech. J*., 36. 1389-1401.

[7] Rouskas, G.., Baldine, I (1997). Multicast routing with End-To-End Delay and Delay Variation Constraints, *IEEE Journal on Selected Areas in Communications*, 15 (3).

[8] Kim, M., Bang, Y.C., Choo, H (2004). On core selection algorithm for reducing delay variation of many-to-many multicasts with delaybounds, NETWORKING 2004. p. 200-210.

[9] Shankar M., Banik, Sridhar Radhakrishnan, Chandra N. Sekharan, Multicast Routing with Delay and Delay Variation Constraints for Collaborative Applications on Overlay Networks, *IEEE Transactions on Parallel and Distributed Systems*, 18 (3) (March 2007) 421-431.

[10] Lee, Hun-Young., Youn, Hun-Young (2000). Scalable Multicast Routing Algorithm for Delay-variation Constrained Minimum residual bandwidth Tree, *In*: *Proc. of the IEEE ICC2000 conference*, June 2000, New Orleans, USA.

[11] Dijkstra, E. W. (1959). A Note on Two Problems in Connection With Graphs, *Numerische Mathematik,* 1 p. 269-271.

[12] Karaman, A., Hassanein, H. (2005). Core-selection algorithms in multicast routing – comparative and complexity analysis, *Computer Communications* 1-17.

[13] Fleury, E., Huang, E., McKinley, P. K. (2000). On the Performance and Feasibility of Multicast Core Selection Heuristics, *Networks*, 35 (2) p. 145-156, John Wiley.