D2-VND-CS: Delay and Delay Variation Constrained Algorithm Based on Variable Neighborhood Descent Algorithm for Core Selection Problem in Multicast Routing Protocol

Youssef BADDI, Mohamed Dafir ECH-CHERIF El KETTANI
Information Security Research Team - ISeRT
ENSIAS - Mohammed V-Souissi University
Rabat, Morocco
{baddi,dafir}@ensias.ma

ABSTRACT: *Within the development of network multimedia technology, more and more real-time multimedia applications arrive with the need to transmit information using multicast. These applications require a multicast routing protocol which has packets arriving at the multicast receptors within a specified delay and delay variation bound. D2-VND-CS (delay and delay variation Core Selection) problem consists in choosing an optimal multicast router in the network as the root of the shared path multicast tree (SPT) within a specified delay and delay variation bound. The choice of this designated router is the main problem concerning multicast tree construction; this choice influences multicast routing tree structure, and therefore influences performances of both multicast session and routing scheme. The determination of a best position of the core within a specified delay and delay variation bound is an NP complete problem: it needs to be solved through a heuristic algorithm. In this paper, we propose a new Core Selection CS algorithm based on Variable Neighborhood Descent algorithm, based on a systematic neighborhood changing. D2-VND-CS algorithm selects the core router by considering tree cost, delay and delay variation. Simulation results show that good performance is achieved in terms of multicast cost, end-to-end delay and tree construction delay and other metrics.*

Keywords: Core, D2-VND-CS, Multicast routing, PIM-SM, SPT

Received: 7 February 2012, Revised 2 April 2012, Accepted 9 April 2012

1. Introduction

Traditional applications using the Internet require a unicast communications. However, some important new emerging applications (videoconferencing) require simultaneous communication between groups of computers with quality of service (QoS) requirement. Therefore, Deering [11] proposed a technique called IP multicast routing, which entrusts the task of data duplication to the network: applications can send one copy of each packet and address it to the group of involved computers; the network takes care of message duplication to the receivers of the group. Multicast IP is bandwidth conserving technology that reduces traffic in the network, and by the many, the bandwidth consumption also.

Multicast communication is based on a multicast tree for data routing; multicast routing protocols are built using two kinds of multicast trees: source based trees and shared trees. With single source based trees, a separate tree is built for each source. With a shared tree, one tree is built for the entire group and shared among all senders; core based trees have a significant advantage in terms of routing resources more than source-based trees in that only one routing table entry is needed for the group [33].

Shared trees can be constructed using a core based tree: in this paper, we will focus on core based tree construction. It requires the selection of a central router called "*Rendezvous point*" RP in PIM-SM protocol [12] and "*core*" in CBT protocol [6]. Finding out an optimal position of this router is a very well knew problem called core selection problem.

Core selection directly impacts on the structure of the tree and the performance of multicast routing scheme.

With the proliferation of existing multimedia group applications, the construction of Shared multicast tree satisfying the quality of service (QoS) requirements is becoming a problem of the prime importance. Current implementations of PIM-SM [12] and CBT [6] protocols decide on the core router selection administratively [22], based only on priority and IP address of each candidate core. This leads to high cost, high delay, and high congestion.

Delay and Delay Variation core selection DDVCS correspond to the problem of finding a multicast tree with a core router satisfying the Delay and Delay Variation Bound quality of service from any source to any destination of multicast group. This problem first proposed by G. N. Rouskas and I. Baldine [23] but with a single source in the multicast tree, is an NP complete problem [23, 27, 2, 1], which needs to be solved with a heuristic algorithm. In this paper, we introduce a novel Delay and Delay Variation core selection algorithm D2-VND-CS that can improve the delay and delay variation constraint in a multi source shared tree.

This paper is organized as follows. In the next section, we describe the core selection problem. Section 3 presents the state of research of the core selection problem in the literature. Section 4 is devoted to the description of a mathematical modeling of core selection problem. Section 5 describes the proposed D2-VND-CS algorithm for the core selection problem. Simulation results are reported in Section 6. Finally, Section 7 provides concluding remarks.

2. Background

The main role of a multicast routing protocol consists in managing multicast groups and routing multicast messages through an optimal multicast tree in order to reach all group nodes, which facilitates the operation of multicast packet duplication. Constructing a unique multicast tree covering all multicast groups members (receivers and sources) at the same time is known by the minimum Steiner tree problem (MST) [20]; this problem is NP complete [33, 17, 9], and seeks to find a low-cost tree by minimizing cost and transmission delay. Because of the difficulties in obtaining SMT, especially in larger graphs, it is often deemed acceptable to use other optimal trees to replace SMTs through a heuristic algorithm. Multicast routing protocols are classified in two categories, as mentioned earlier (SBT and ST) [32].

Source based tree SBT or Shortest Path Tree SPT is composed of the shortest paths between the source as root and each receivers of multicast group. The main motivations behind using a source based tree SBT are the simplicity of building in a distributed manner using only the unicast routing information, and optimization of transmission delay between source and each receiver [17]. The main drawbacks of SBT are: additional costs for maintaining SPT trees, and the number of statements to be stored in the nodes complexity is $O(S * G)$ (S is the number of sources and G is the number of groups) [17]. The shortest path tree SPT is used by several multicast routing protocols such as DVMRP [31], MOSPF [21], and PIM-DM [25].

Source-based trees in general are mostly suitable for small-scale, local-area applications. The main motivation for their use is delay optimization during multicast forwarding. They are not adapted to sparse mode situation because of the additional overhead of tree maintenance; also the scalability of source-based protocols tends to degradation in terms of network resource consumption [17].

Core based trees are more appropriate when there are multiple sources in the multicast group. Under this approach, Core-based trees separate the concept of source from that of the tree root. One node in the network is chosen as the center, and the sources forward messages to the center. Like SBT tree, a shortest path multicast tree is constructed rooted at the selected core, offering better flexibility and extensibility. And only routers on the tree need to maintain information related to group members. It gives good performance in terms of the quantity of state information to be stored in the routers and the entire cost of routing tree [31, 21, 25].

Joining and leaving a group member is achieves explicitly in a hop-to-hop way along the shortest path from the local router to core router resulting in less control overhead, efficient management of multicast path in changing group memberships, scalability and performance [11, 29].

Several multicast routing protocols in the literature use core-based tree: Protocol Independent Multicasting-Sparse mode PIM-SM [12] and Core-Based Tree (CBT) [6]. Current implementation of PIM-SM [12] and CBT [6] protocols divide the tree construction problem into two sub-problems, the first is center selection problem and the second is routing problem. PIM-SM [12] and CBT [6] use for center selection a special router called Bootstrap router (BSR) fifined in RFC 5059 [7], which notifies a set of candidate cores. Every node uses a Hash function to map to one core according to the address of the group; this hash function is based on router priority and his IP address. Both of these parameters do not guarantee the selection of an optimal core with any delay and delay variation guaranties. This leads to high cost, high delay, and high congestion. This problem first proposed by G. N. Rouskas and I. Baldine [23], is an NP complete problem [23, 27, 2, 1], which needs to be solved through a heuristic algorithm.

In this paper, we propose a new Core Selection Algorithm VND-CS based on a "*Variable Neighborhood Descent*". VND algorithm has already been applied successfully to resolve a wide variety of NP-hard problems [30, 10, 26, 8, 15] to select a global optimal solution using several neighborhoods structures systematically, but not yet in core selection problem. VND-CS can simultaneously minimize the delay and cost of the multicast tree. It attempts to find the best core using a fitness function.

3. Literature Review

There are several proposals, algorithms and mechanisms for core selection problem in the literature. A variety of these algorithms are compared in [17]. Among proposed selection algorithms, we find the Random Source-Specific Tree (RSST) heuristic, in which, the center is chosen randomly among the sources. It is comparable to selecting the first source or the initiator of the multicast group, as proposed in PIM [12] and CBT [6] protocols.

Some Proposed algorithms select Core on the basis of basic heuristics and do not consider QoS constraints. This kind of Core selection can provide every member of the group with a cost function guarantee to the core iteratively selected, this set of algorithms is highly complex and the used cost function is hard to be guaranteed for new group members. From this algorithm set we mention OCBT proposed by Shields and Garcia-Luna-Aceves [18], Topology- Based Algorithm [9], which selects a single core closest to topology center by using the domain topology and subgraph constructed from the multicast group. In order to reduce the search area used by the Topology-Based Algorithm, and to select a distributed cores for all multicast groups in the network domain and close to the group members, [9] proposed group-based algorithm. Tournament-based algorithm proposed by Shukla, Boyer, and Klinke [29] executes a Distributed tournament between nodes to determine a center; Tabu Search algorithm for RP selection (TRPSA) [28] is a distributed core selection algorithm, based on dynamic meta-heuristic Tabu Search TS algorithms proposed first by Glover [28] to solve combinatorial optimization problems in PIM-SM protocol [12]. We cite also our algorithms VNS-RP [4], VND-CS [5] and GRASP-RP [3] based in VNS [15], VND [19] and GRASP [13] heuristics successively.

There are also many well-known approaches to select core router satisfying delay and delay-variation constraints.

Delay Variation Multicast Algorithm (DVMA) was proposed by G. N Rouskas, I. Baldine [23] to resolve the Delay and Delay Variation Bounded Multicasting Network (DVBMN) problem. DVMA [23] tries to find a sub-network given a source and a set of destinations that satisfies the QoS (Quality of Service) requirements on the maximum delay from the source to any of the destinations and on the maximum inter-destination delay variance: it starts with a source-based tree spanning some and not always all multicast members satisfying the delay constraint only. Then the algorithm searches through the candidate paths satisfying the delay and delay-variation constraints from a non-tree member node to any of the tree nodes. DVMA [23] is most classed in source-based tree then shared tree, and it assumes that the complete topology is available at each node. The computer simulation shows that the performance of DVMA [23] is good in terms of multicast delay-variation. However, it shows a high complexity i.e. $O(kldn^4)$ where k and l are the number of paths satisfying the delay bound between any two nodes; $|D| = d$ and $|N| = n$ represents number of multicast receptors node and total number of nodes in the topology network respectively.

Delay and Delay Variation Constraint Algorithm (DDVCA) was proposed by Sheu and Chen [27] based on the Core Based Tree (CBT) [6]: the main objective of DDVCA [27] is to find as much as possible core router spanning a multicast tree with a smaller multicast delay variation under the multicast end-to-end delay constraint. To do that, DDVCA [27] first calculates the delay of the least delay path from the destination nodes to all the nodes. The node that has the minimum delay-variation is selected as the core node. In comparison with the DVMA [23], DDVCA [27] Algorithm shows a significant lower complexity i.e. $O(dn^2)$. where m is the number of destination nodes and n is the total number of nodes in the computer network.

Kim et.al [19] has proposed another efficient core selection algorithm based also on CBT like DDVCA [27] to build a core based multicast tree under delay and delay-variation bound. First, AKBC [19] finds a set of candidate core nodes that have the same associated multicast delay-variation for each destination node. Then, it selects a final core node from this set of candidate core nodes that has the minimum potential delay-variation. AKBC [19] algorithm investigates candidate nodes to select the better node with the same complexity as DDVCA [27] i.e $O(dn^2)$.

All these algorithms (DVMA [23], DDVCA [27] and AKBC [19]) are only applied in the symmetric network environment that has no direction. To overcome this limitation, Ahn, Kim and Choo [1] proposed AKC (Ahn Kim Choo) to build a multicast tree with low delay-variation in a realistic network environment that has two-way directions. This algorithm works efficiently in the asymmetric network with the same complexity as DDVCA i.e $O(dn^2)$.

The last core selection algorithm, proposed by Sahoo and. al [24], is based on dynamic meta-heuristic Tabu Search TS algorithms, proposed first by Glover [16], to solve combinatorial optimization problems. Tabu Search algorithm for RP selection (TRPSA) [24] is a distributed core selection algorithm to find a local solution after a certain finite number of iterations by using memory structures that describe the visited solutions. The basic idea of the TRPSA [24] algorithm is to mark the best local solution obtained in order to prevent the research process to return back to the same solution in subsequent iterations using a data structure to store the solutions already visited, this structure is called tabu list. However, the method requires a better definition of stopping criterion and effective management of the tabu list, since the choice of stopping criterion and tabu list size is critical and influences the performance of the algorithm. According to [24], TRPSA has $O(|E| + (|S| + d)n^2)$ complexity with $|E|$ the nember of edge in the network topology.

However, these algorithms [23, 27, 19, 1] select the best core node out of a set of candidate core nodes that have the same associated delay-variation. Therefore, these algorithms are restricted only to selecting the best core node, which may not generate an optimal delay-variation-based multicast tree in many cases. Also TRSPA [24] doesnâ AZt overcome this limitation because it just selects a local optimal node which may not generate an optimal delay and delay-variation-based multicast tree in all topology networks.

Table 1 present a comparative study of existing core selection algorithms.

Algorithms	Complexity
DVMA [23]	$O(kldn^4)$
DDVCA [27]	$O(dn^2)$
AKBC [19]	$O(dn^2)$
AKC [1]	$O(dn^2)$
TRPSA [24]	$O(\|E\| + (\|S\| + d) * n^2)$

Table 1. Comparative study of Single-core selection algorithms

4. Mathematical Modeling

A computer network is modeled as a simple directed and connected graph $G = (N, E)$, where N is a finite set of nodes and E is the set of edges (or links) connecting the nodes. Let $|N|$ be the number of network nodes and $|E|$ the number of network links. An edge $e \in E$ connecting two adjacent nodes $u \in N$ and $v \in N$ will be denoted by $e(u,v)$, the fact that the graph is directional, implies the existence of a link $e(u,v)$ between v and u. Each edge is associated with two positive real value: a cost function $C(e) = C(e(u, v))$ represents link utilization (may be either monetary cost or any measure of resource utilization), and a delay function D(e) = $D(e(u,v))$ represents the delay that the packet experiences through passing that link including switching, queuing, transmission and propagation delays. We associate for each path $P(v0, vn) = (e(v0, v1), e(v1, v2), ..., e(vn-1, vn))$ in the network two metrics:

$$C(P(v_0, v_n)) = \sum_{0}^{n-1} C(e(v_i, v_{i+1})) \quad (1)$$

And

$$D(P(v_0, v_n)) = \sum_{0}^{n-1} C(e(v_i, v_{i+1})) \quad (2)$$

A multicast tree $T_M(S, C, D)$ is a sub-graph of G spanning the set of sources node S $\subset$ N and the set of destination nodes D Nwith a selected core C. Let | S | be the number of multicast destination nodes and | D | is the number of multicast destination nodes.

In Protocols using Core-based tree, all sources node needs to transmit the multicast information to selected core via unicast routing, then its well be forwarded to all receptors in the shared tree, to model the existence of these two parts separated by core, we use both cost function and delay following:

$$C(T_M(S, C, D)) = \sum_{s \in S} C(P(s, C)) + \sum_{d \in D} C(P(C, d)) \quad (3)$$

And

$$D(T_M(S, C, D)) = \sum_{s \in S} D(P(s, C)) + \sum_{d \in D} D(P(C, d)) \quad (4)$$

We introduce also a Delay Variation function defined as the difference between the Maximum and minimum end-toend delays along the multicast tree from the source to all destination nodes and is calculated as follows:

$$DelayVariation = \max\left(D\left(T_M\right)\left(S, C, d\right)\right) - \min\left(D\left(T_M\right)\left(S, C, D\right)\right) \quad \forall d \in D \quad (5)$$

Core selection problem tries to find an optimal node C in the network with an optimal function Opt_F by minimizing in the fist time the cost function $C(T_M(S, C, D))$ and in the second a Delay and delay variation bound as follows:

$$Opt_F(C, T_M) = \begin{cases} C(T_M(S, C, D)) \\ D(T_M(S, C, D)) < \alpha \\ Delay\ Variation < \beta \end{cases} \quad (6)$$

5. A Variable Neighborhood Descent for Core Selection VND-CS

5.1 Basic Variable Neighborhood Descent Search Algorithm

Many combinatorial optimization algorithms have been widely exploited for solving research problems (Hill climbing and tabu search algorithms). These algorithms allow selecting a local optimal point by trajectory following with a single neighborhood. Contrary to all others kind of meta-heuristics based on local search methods, Mladenovi´c and Hansen [19] proposed a recent meta-heuristics Variable Neighborhood Decent VND Algorithm based on the simple idea of a systematic neighborhood changing arbitrarily. VND has been applied successfully to a wide variety of NPhard problems to select a global optimal solution such as the travelling salesman problem [10], Job Shop Scheduling Problems [26], the clustering problem [15], Arc routing problems [30], and nurse rostering [8].

The use of more than one neighborhood provides a very effective method that allows escaping from a local optimum. In fact, it is often the case that the current solution, which is a local optimum in one neighborhood, is no longer a local optimum in a different neighborhood; therefore, it can be further improved using a simple descent approach.

As defined by Mladenovic and Hansen [19], in the VND paradigm, a finite set of neighborhoods structures N_k (k = 1,..., k_{max}) and an initial solution S are generated, starting from this initial solution, a so-called shaking step is performed by randomly selecting a solution S from the first neighborhood. If this solution (S) improve the weight function used one starts with the first neighborhood of this new solution ($S \leftarrow S'$); otherwise one proceeds with the next neighborhood. This procedure is repeated as long as a neighborhood structure allows such iteration.

5.2 A Variable Neighborhood Descent for Core Selection Problem

The main motivation behind the use of the VND search algorithm to solve core selection problem is the use of several neighborhoods to explore different neighborhood structures systematically. Our goal is to break away from a local minima, this use is based on three facts:

1. If node N_1 is a local minimum for one neighborhood structure N_k is not necessary so with another one N_k.

2. A global minimum solution S is a local minimum for all possible neighborhood structures.

3. For the core selection problem local minima for all neighborhood structures is relatively close and localized in the same place.

In this section we provide a detailed description of the Variable Neighborhood Descent algorithm for core selection Problem VNS-CS, and his three process phases: the initialization process, Stopping conditions phase and the shaking step. After, the source code will be discussed according to the features of core selection problem; we consider the solution object such as node and neighborhood like the set of neighboring nodes.

5.3 Initial Solution

The first step of variable neighborhood descent search is to define an initial solution. Many methods can be used to generate this solution; the simplest is to select randomly one node in the network as initial solution. There are other methods that try to reduce the selection area and generate an initial solution from an ordered set of multicast group members.

5.4 Neighborhood Structures

VNS-CS uses neighbor nodes concept to generate neighborhood structures: a node *u* is neighbor of another node *v* if an edge *e*(*u, v*) between *u* and *v* exists. We propose to compute a neighborhood structure N_j throught the following formula (with *neighbor* (*S*) a set of neighbor nodes of *S*):

(7)

5.5 Shaking

In the shaking step we explore the k^{th} neighborhood of an initial solution *S* to generate a local minimum solution *S'* related to this neighborhood structure $N_k(S)$.

After a random selection of a one local solution S in k^{th} neighborhood structure, *S* is compared to *S*. If *S'* is better than *S*, it replaces *S* ($S \leftarrow S'$) and the algorithm starts all over again with $k \leftarrow 1$. Otherwise, *k* is incremented and algorithm continues the shaking step with next neighborhood structure.

5.6 Stopping conditions phase

From an initial solution, the shaking and movement steps continues until a stopping condition is met. In case of small problem instances, where the best solution is usually found very quickly, the stopping conditions with a limit on the maximum number of iterations is sufficient. This stopping condition is not enough when large-scale problem is treated. Hence, many stopping conditions have been added for large-scale topology. These conditions are the maximum CPU time allowed and the maximum number of non-improving iterations.

5.7 VNS-CS algorithm and pseudo code

In this section as presented in Algorithm 1, a step by step VND-CS algorithm for core selection problem is presented, containing seven steps:

step 1. Set maximum iteration number, maximum number of iteration Without Improvement;

step 2. Select initial solution *S*;

step 3. Choose the kmax scalar, select the set of neighborhood structures N_k, for $k = 1,..., k_{max}$, that will be used in the search; choose a stopping condition;

step 4. Check the stopping criterion. If it is satisfied go to step 7; otherwise, go to nest step;

step 5. Take at random a solution S0 from $N_k(S)$;

step 6. Check if objective function value of solution *S'* is less than objective function value of solution *S*, then move to *S'*

solution and continue the search with N_1 ($k \leftarrow 1$) from step 4; otherwise, set $k \leftarrow k+1$ and also continue the search from step 4;
step 7. Output the best solution core selected.

Algorithm 1 D2-VNS-CS Pseudo code

```
Input : i = 0, Total_Iteration = 0, Current_Iteration = 0;
Input : Max_Iteration_Without_Improvement, Initial_Solution, Max_i teration;
Input : SolutionChk, Best_Solution;
1.  Best_Solution ← initial_Solution;
2.  while i < Max_Iteration_Without_Improvement && Total_Iteration < maxit do
3.    lastCost ← Opt_F (Best_Solution); k ← 0;
4.    while Current_Iteration < Max_i teration && k < kmax do
5.      if Total_Iteration > Max_i teration then
6.        break;
7.      end if
8.      getN_K (s) ;
9.      SolutionChk ← getRandomN_K (s) ;
10.     if Opt_F (SolutionChk) > Opt_F (Best_Solution) then
11.       k ← k + 1;
12.     else
13.       Best_Solution ← SolutionChk;
14.       k ← 0;
15.     end if
16.     Total_Iteration ← Total_Iteration + 1 ;
17.     Current_Iteration ← Current_Iteration + 1;
18.   end while
19.   if lastCost > Opt_F (Best_Solution)) then
20.     i ← 0;
21.   else
22.     i ← i + 1;
23.   end if
24. end while
25. return Best_Solution
```

5.8 Complexity Analysis

The complexity of VND-CS algorithm is explained line by line in the following. Line 1 is initialization statement. Their complexity is $O(1)$. Line 2 and 4 are a judgment statements of the while loop, and its complexity is $O(1)$. Line 3 is initialization statement. Their complexity is $O(1)$. Lines 5 7 are judgment statements and their complexity is $O(1)$. Line 8 generate the kth neighborhood structures, the complexity is $x(|neighborhoodstructures|k)$, the average value of $|neighborhoodstructures|$ is $(2|E|/|N|)$, then the total complexity is $O((2|E|/|N|)k) < O(|E|)$. Line 9 select randomly one solution from $N_k(S)$, their complexity is $O(1)$. Line 10 computes and compare weight function and their complexity is $O(|S|+|D|)$. Lines 11 - 17 correspond to assignment, their complexity is $O(1)$.

Lines 22 - 23 form a while loop. Generally it take less time than | N |, then the complexity is

$$O(|N|(O(1)+O(|E|)+O(1)+O(|S|+|D|)+O(1))) = O(3|N|+|N||E|+|N|(|S|+|D|))$$

Line 24 is a return statement, and its complexity is O(1). Therefore, the complexity of the algorithm is $O(1) + O(1) + O(3|N|+|N||E|+|N|(|S|+|D|))$, that is, $O(3|N|+|N||E|+|N|(|S|+|D|))$.

6. Simulation Results

In this section, we use simulation results to demonstrate the effectiveness of the proposed algorithm described above. To study the performance of our selection algorithm D2-VND-CS, we implement it in a simulation environment; we use the network simulator NS2. The random graph generator GT-ITM is used to generate a random different 100 networks, and we adopt Waxman [14] as the graph model. Our simulation studies were performed with a 100 runs. The values of $\alpha = 0.2$ and $\beta = 0.2$ were used to generate networks with an average degree between 3 and 4 in the mathematical model of Waxman.

To demonstrate the performance of this algorithm (D2-VND-CS), we compare it with the following algorithms, including random (*R*), Tabu RP selection (TRPS) [24] and DDVCA [27].

The main objective of our algorithm is to reduce delay and delay variation; therefore, we start the simulation results by comparing these two metrics. Figure 1 and 2 shows the simulation results of multicast delay-variations versus the number of nodes on a topology network. The multicast group member's size is 10% of the overall network nodes. Simulation results show that D2-VND-CS is the best among all the algorithms on average delay, with TRPS [24] and DDVCA [27] following it, and Random is the worst.

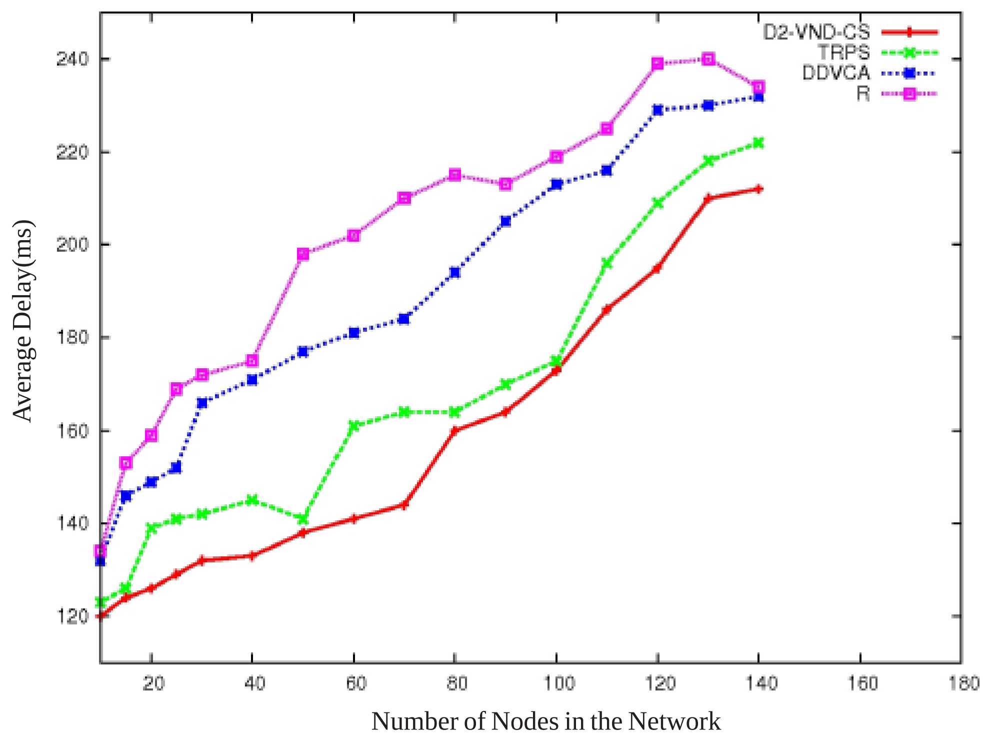


Figure 1. Comparison of Delay vs. Network size

Based on the cost function in the formula (3), Figure 3 presents a comparison study of multicast tree Cost generated by each algorithm, the performance of Random selection is the worst, followed by DDVCA [27] and TRPS [24], D2-VND-CS shows better performances, and it has the minimal cost.

The Figure 4 and 5 show the multicast delay and delay-variation for a network of 200 nodes. The multicast group size is between 10% and 80% of the overall nodes of the network. Simulation results show that multicast trees build by our proposed algorithm have an average multicast delay and delay variation better than TRSPA [24] and DDVCA [27] algorithm and support more multicast members.

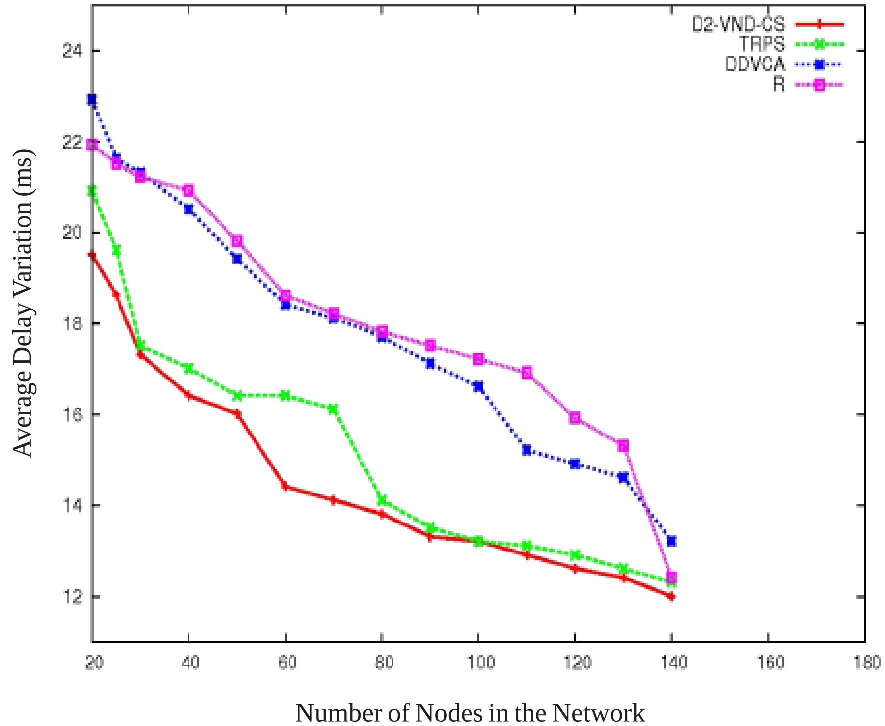


Figure 2. Comparison of Delay Variation vs. network size

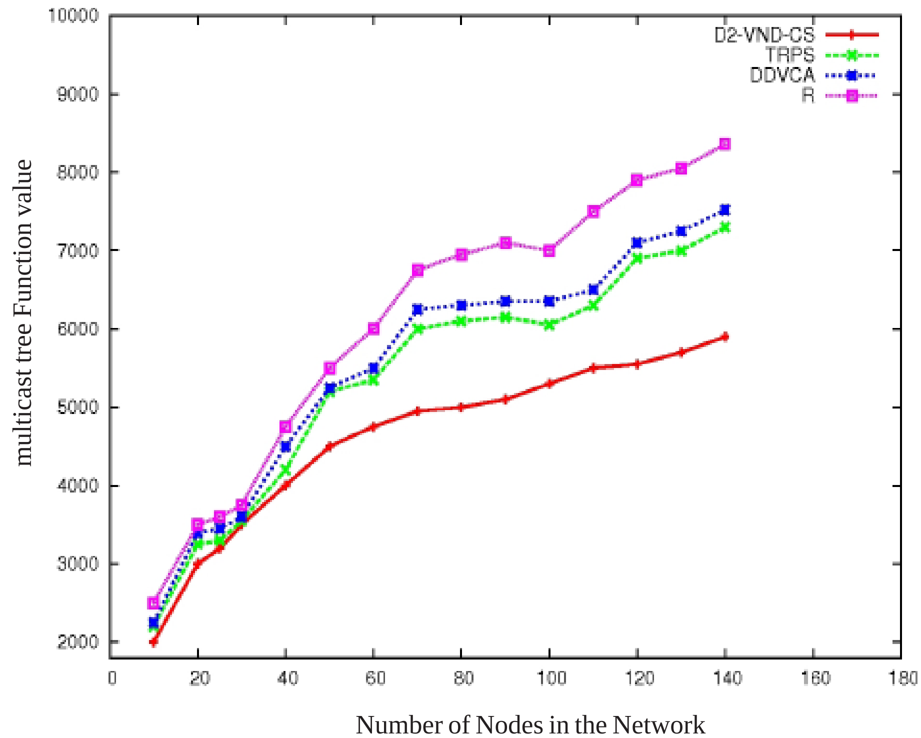


Figure 3. Comparison of Multicast Tree Cost vs. network size

The Figure 4 and 5 show the multicast delay and delay-variation for a network of 200 nodes. The multicast group size is between 10% and 80% of the overall nodes of the network. Simulation results show that multicast trees build by our proposed algorithm have an average multicast delay and delay variation better than TRSPA [24] and DDVCA [27] algorithm and support more multicast members.

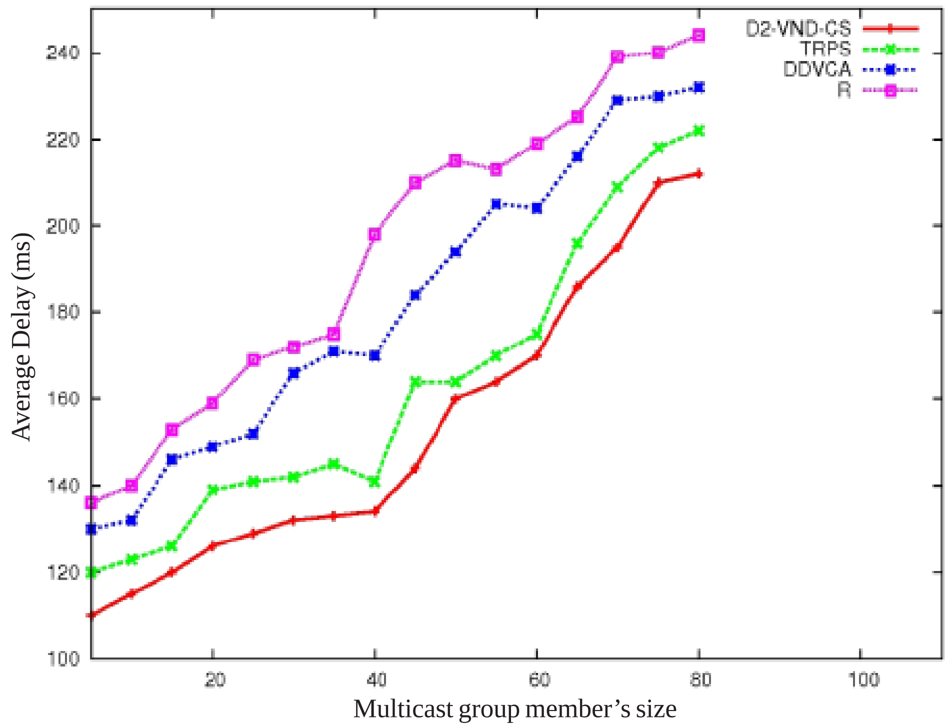


Figure 4. Comparison of Delay vs. multicast group member's size

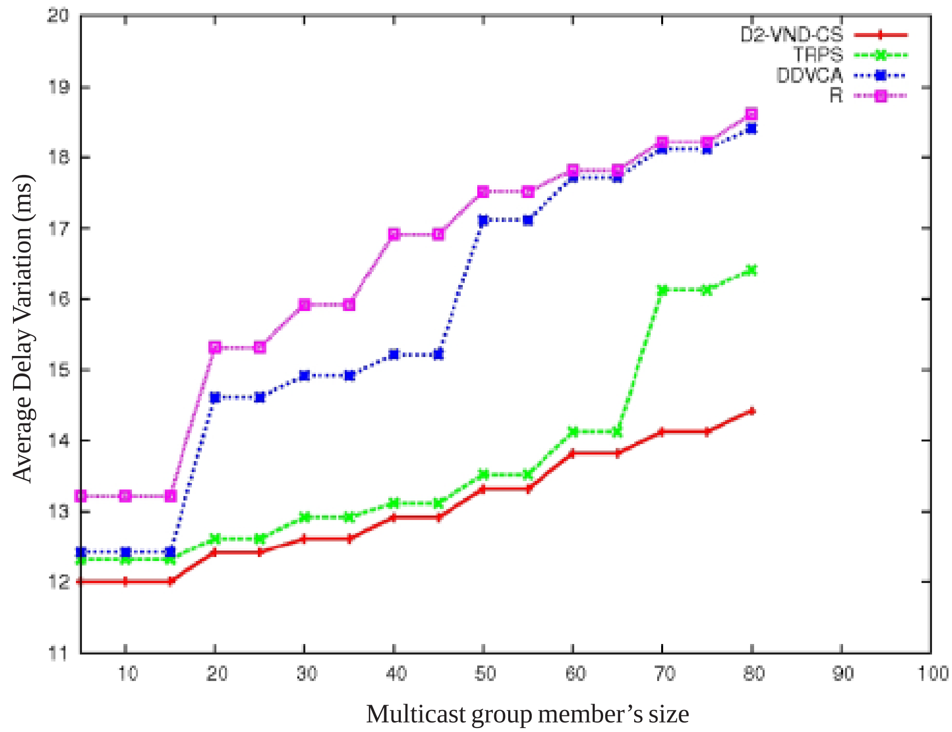


Figure 5. Comparison of Delay vs. multicast group member's size

7. Conclusion

In this paper we have investigated the problem of finding a good core router in a distributed manner. Core Selection CD problem

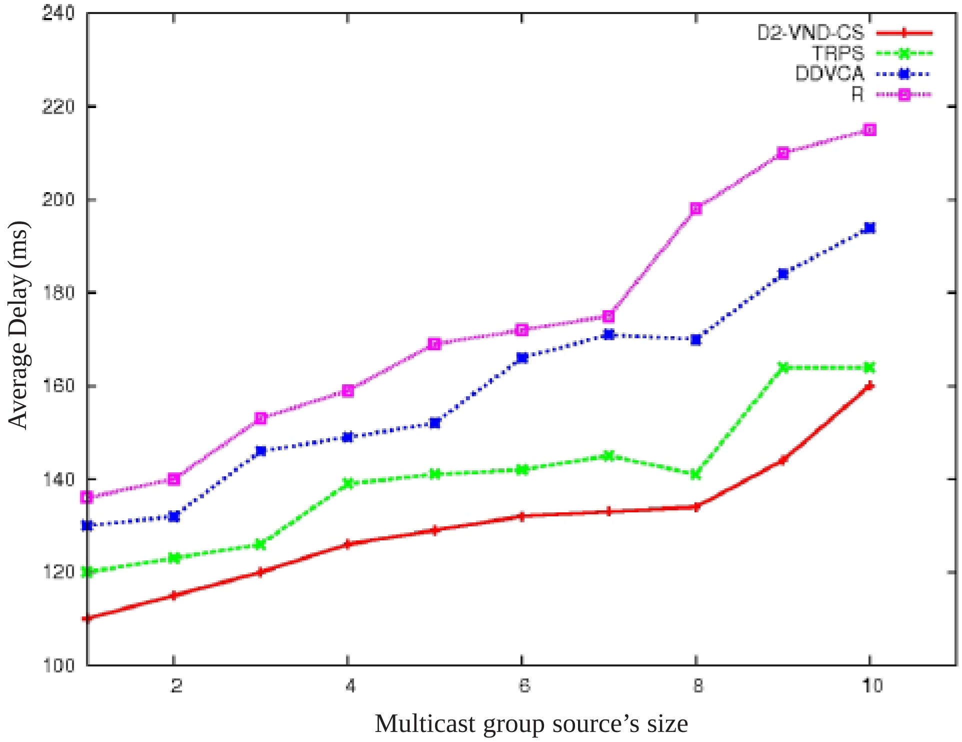


Figure 6. Comparison of Delay vs. multicast source's size

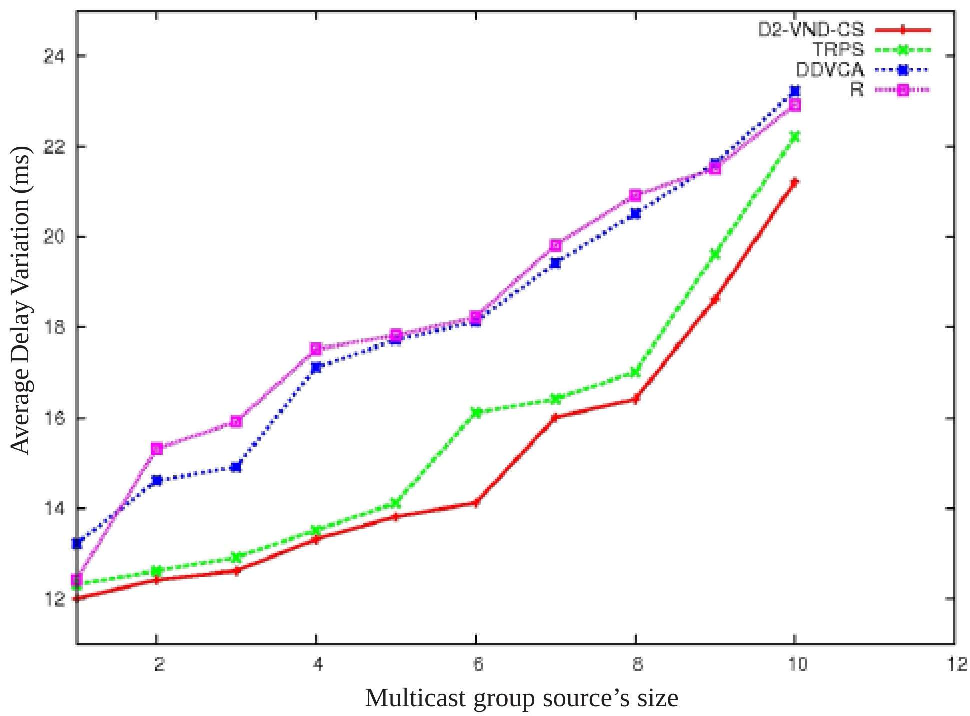


Figure 7. Comparison of Delay vs. multicast source's size

affects directly the structure of the tree and the performance of the routing scheme of multicast consequently. Current algorithms decide on core router administratively, which leads to high cost, high delay, and high congestion. To solve these problems, D2-

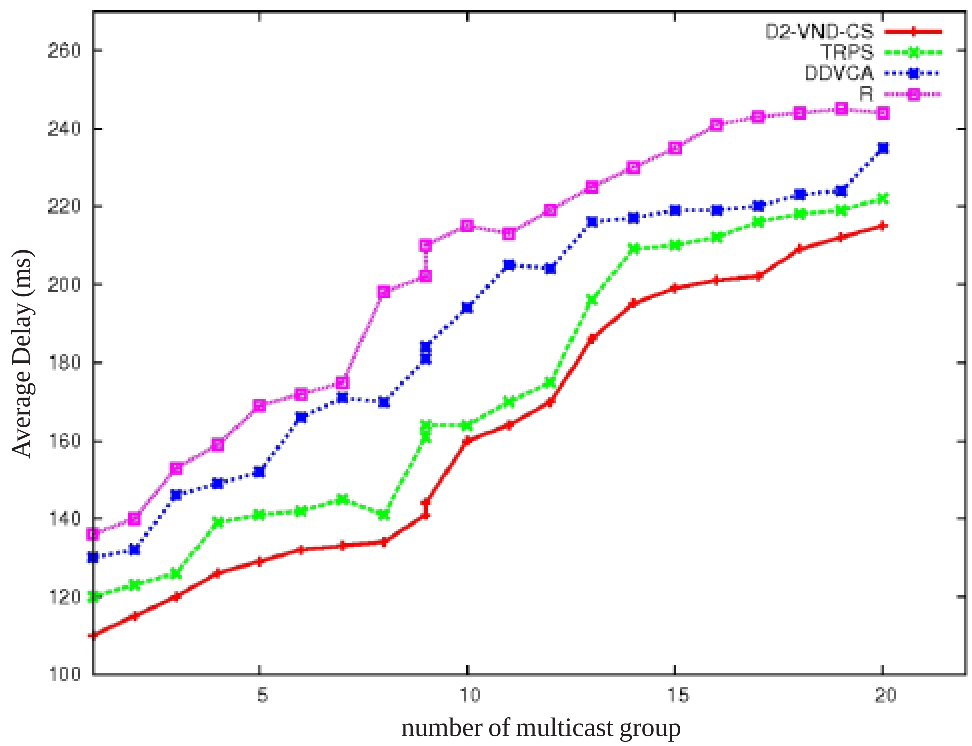


Figure 8. Comparison of Delay vs. Nember of group

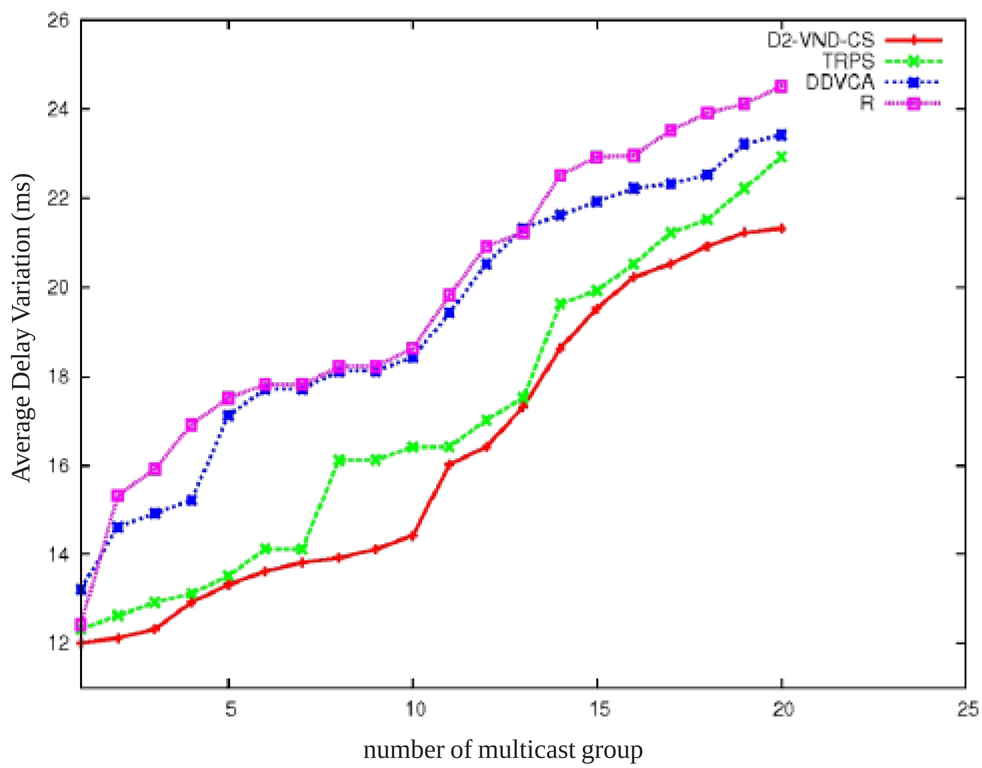


Figure 9. Comparison of Delay Variation vs. Nember of group

VND-CS is proposed based on VND heuristic algorithm. To present our proposition we started with a brief overview of multicast routing protocols and two types of multicast trees SBT and CBT. We reviewed and analyzed the cost, delay and delay variation functions. We reviewed the core selection algorithms studied in literature for their algorithmic structures. Simulation results show that our algorithm presents good performances in multicast cost, delay, delay variation and other aspects. Our future work is focused on extending this algorithm to support others multiple QOS criteria imposed by receivers across the network.

References

[1] Soobeen Ahn, Moonseong Kim, Hyunseung Choo, (2008). Efficient algorithm for reducing delay variation on delaybounded multicast trees in heterogeneous networks. *In:* WCNC, p. 2741–2746. IEEE.

[2] Youngjin Ahn, Moonseong Kim, Young-Cheol Bang, Hyunseung Choo, (2005). On algorithm for the delay- and delay variation-bounded multicast trees based on estimation. *In:* Proceedings of the First international conference on High Performance Computing and Communications, HPCC'05, p. 277–282, Berlin, Heidelberg. Springer-Verlag.

[3] Youssef Baddi, Mohamed Dafir El Kettani. GRAS-RP: greedy randomized adaptive search algorithm for rp selection in pim-sm multicast routing. *In: Seventh International Conference on Intelligent Systems: Theories and Applications*, Mohammedia, Morocco.

[4] Youssef Baddi and Mohamed Dafir El Kettani. VNS-RP algorithm for rp selection in multicast routing protocol pim-sm. *In:* The 3rd International Conference on Multimedia Computing and Systems (ICMCS'12), Tangier, Morocco.

[5] Youssef Baddi, Mohamed Dafir El Kettani, (2012). VND-CS: a variable neighborhood descent algorithm for core selection = problem in multicast routing protocol. *In:* Fourth International Conference on Networked Digital Technologies (NDT 2012), Dubai, India, April.

[6] Ballardie, A. (1997). Core based trees (cbt version 2) multicast routing – protocol specification –. RFC 2189.

[7] Bhaskar, N., Gall, A., Lingard, J., Venaas, S. (2008). Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM). RFC 5059 (Proposed Standard), January.

[8] Edmund Burke, Patrick De Causmaecker, Sanja Petrovic, Greet Vanden Berghe, (2004). Variable neighborhood search for nurse rostering problems, p. 153–172. Kluwer Academic Publishers, Norwell, MA, USA.

[9] Kenneth L. Calvert, Ellen W. Zegura, Michael J. Donahoo, (1995). Core selection methods for multicast routing. p. 638–642.

[10] Carrabs, F., Laporte, G., Cordeau, J. F., Québec. (2005). (Centre for Research on Transportation (Montréal. Variable neighborhood search for the pickup and delivery traveling salesman problem with LIFO loading. Publication (Centre for Research on Transportation (Montréal, Québec))). Centre for Research on Transportation = Centre de recherche sur les transports (C.R.T.).

[11] Stephen, E. Deering, David, R. Cheriton, (1990). Multicast routing in datagram internetworks and extended lans. *ACM Transactions on Computer Systems*, 8, 85–110.

[12] Fenner, B., Handley, M., Holbrook, H., Kouvelas, I. (2006). Protocol Independent Multicast- Sparse Mode (PIM-SM): Protocol Specification (Revised). RFC 4601 (Proposed Standard), August. Updated by RFC 5059.

[13] Thomas, A. Feo and Mauricio, G. C. (1995). Resende. Greedy randomized adaptive search procedures, *Journal of Global Optimization*, 6,109–133.

[14] Hansen, P., Mladenovic, N., Pierre Hansen, Nenad Mladenovic, Les Cahiers Du Gerad, (1999). Variable neighborhood search: Methods and recent applications. *In:* Proceedings of MICâ ØA ´ Z99, p. 275–280.

[15] Pierre Hansen, Nenad Mladenovic, (2001). Variable neighborhood search: Principles and applications. European Journal of Operational Research, 130 (3) 449–467, May.

[16] Wang Hua, Meng Xiangxu, Zhang Min, Li Yanlong, et al. (2010). Tabu search algorithm for rp selection in pim-sm multicast routing, *Comput. Commun.*, 33 (1) 35–42, January.

[17] Ay¸se Karaman, Hossam Hassanein, (2006). Core-selection algorithms in multicast routing-comparative and complexity analysis. Comput. Commun., 29 (8) 998–1014.

[18] Aye Karaman, Hossam Hassanein, (2003). Core-based approach in multicast routing protocols.

[19] Moonseong Kim, Young-Cheol Bang, Hyunseung Choo, (2004). On core selection algorithm for reducing delay variation of many-to-many multicasts with delay-bounds. *In:* Nikolas Mitrou, Kimon P. Kontovasilis, George N. Rouskas, Ilias Iliadis, Lazaros F. Merakos, editors, NETWORKING, vol. 3042 of Lecture Notes in Computer Science, p. 200–210. Springer.

[20] Kurt Mehlhorn, (1988). A faster approximation algorithm for the steiner problem in graphs. *Inf. Process. Lett.*, 27, 125–128, March.

[21] Moy, J. (1994). MOSPF: Analysis and Experience. RFC 1585 (Informational), Internet Engineering Task Force IETF, March.

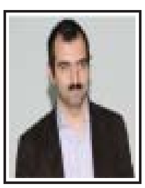
[22] Maria Ramalho, (2000). Intra- and inter-domain multicast routing protocols: A survey and taxonomy, *IEEE Communications Surveys and Tutorials*, 3 (1) 2–25.

[23] George N. Rouskas, Ilia Baldine, (1995). Multicast routing with end-to-end delay and delay variation constraints. Technical report, Raleigh, NC, USA.

[24] Satya Prakash Saho, Manas Ranajan Kabat, Asis Kumar Sahoo, (2011). Tabu search algorithm for core selection in multicast routing. *In:* Proceedings of the 2011 International Conference on Communication Systems and Network Technologies, CSNT 11, p. 17–21, Washington, DC, USA. IEEE Computer Society.
[25] Savola, P., Lehtonen, R., Meyer, D. (2006). Protocol Independent Multicast - Sparse Mode (PIM-SM) Multicast Routing Security Issues and Enhancements. RFC 4609 (Informational), October.
[26] Mehmet Sevkli, Mehmet Emin Aydin, (2006). Variable neighbourhood search for job shop scheduling problems. *JSW* 1 (2) 34–39.
[27] Pi-Rong Sheu, Shan-Tai Chen, (2001). A fast and efficient heuristic algorithm for the delay- and delay variation bound multicast tree problem. *In:* Proceedings of the The 15th International Conference on Information Networking, ICOIN '01, p. 611–, Washington, DC, USA. IEEE Computer Society.
[28] Clay Shields, Garcia-Luna-Aceves, J. J. (1997). The ordered core based tree protocol. In Proceedings of the INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution, INFOCOM '97, p. 884–, Washington, DC, USA. IEEE Computer Society.
[29] Shridhar B. Shukla, Eric B. Boyer, Eric Klinker, J. (1994). Multicast Tree Construction in Network Topologies with Asymmetric Link Loads. PhD thesis.
[30] Mariam Tagmouti, Michel Gendreau, Jean-Yves Potvin. (2010). A variable neighborhood descent heuristic for arc routing problems with time-dependent service costs. Comput. Ind. Eng., 59, 954–963, November.
[31] Waitzman, D., Partridge, C., Deering, S. E. (1988). RFC 1075: Distance vector multicast routing protocol, November. Status: EXPERIMENTAL.
[32] Liming Wei, Deborah Estrin, (1994). The trade-offs of multicast trees and algorithms.
[33] Zappala, Daniel., Fabbri, Aaron., Lo, Virginia Mary(2002). An evaluation of shared multicast trees with multiple cores. Telecommunication Systems, 19 (3-4) 461–479.

Author Biographies

Youssef BADDI received MASTER Thesis degree in computer science from ENSIAS School, University Mohammed V Souissi of Rabat, Morocco, in 2010. Since 2011 he is a Ph.D. Student ENSIAS School, member of Information Security Research Team. His research interests include multicast routing protocol, IPv6, mobile IP, Integration of Multicast IP and mobility IPv6 management, with emphasis on the algorithms for Core selection in the Shared Multicast trees protocol.



Mohammed Dafir ECH-CHERIF EL KETTANI received his and Ph.D. degree from EMI School, University Mohammed V Agdal of Rabat in 1997. He is currently a professor at ENSIAS School, University Mohammed V Souissi of Rabat, Morocco and Information Security Research Team leader. His main research interests are in the area of Security management and multicast routing.