This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.

A Near Optimal Localized Heuristic for Voice Multicasting over Ad hoc Wireless Networks*

G. Venkat Raju [a], T. Bheemarjuna Reddy [b], and C. Siva Ram Murthy [b]
[a] Yahoo! Software Development India Pvt. Ltd., Bangalore, India 560001
[b] Department of Computer Science and Engineering, Indian Institute of Technology Madras, India 600036
gvraju@yahoo-inc.com, arjun@cs.iitm.ernet.in, murthy@iitm.ac.in

***Abstract*— Providing real-time voice multicasting over multi-hop ad hoc wireless networks is a challenging task. The unique characteristics of voice traffic (*viz.* small packet size, high packet rate, and soft real-time nature) make conventional multicasting protocols perform quite poorly, hence warranting application centric approaches in order to provide robustness against packet losses and lower the overhead due to high packet rate. In this paper, we first show that the Optimal Voice Multicasting Tree (OVMT) problem is NP-complete and then propose a Localized Distributed heuristic for Minimum number of Transmissions (LDMT). By incorporating LDMT in ADMR protocol, extensive simulations are done in NS-2 framework to measure the performance of LDMT for voice applications. We observed that LDMT reduces the redundant transmissions in transmitting voice packets from the source to all multicast receivers (thus reducing the overall voice traffic considerably), thereby making it suitable for voice multicasting in AWNs.**

I. Introduction

An Ad hoc Wireless Network (AWN) is a collection of mobile nodes that dynamically form a temporary network without any pre-existing infrastructure. AWNs are characterized by high bit error rates and path breaks due to frequently changing network topology. As developments in AWNs continue, there is an increasing expectation of sending multimedia data to more than one receiver simultaneously. In this paper, we concentrate on voice multicasting as it is a key application in many group-oriented scenarios.

Voice application can tolerate packet losses up to 5% [1], but is highly delay sensitive (typically for interactive voice application, the *end-to-end delay* should be less than 200 *ms* [2]). All lately arrived packets are assumed to be lost. The unique characteristics of voice traffic, such as small packet size, high packet rate (typically 50 pkts/s to 100 pkts/s), and soft real-time nature make voice multicasting a very challenging issue in AWNs. The efficiency of AWNs for voice applications is poor due to small voice payloads (typically 20 bytes) and large packetization and synchronization overheads that are unique to wireless networks [3]. Thus, to make voice multicasting feasible in AWNs, the overall voice traffic (the total number of voice packets exchanged in the network due to voice multicast) must be minimized while reducing the end-to-end delay to the maximum possible extent. An important parameter in this connection is the Number of Forwarding Nodes (NFNs) in the multicast tree. We define forwarding nodes as those nodes which are not leaf nodes in the multicast tree. Note that a forwarding node can be a receiver or a non receiver. The term NFNs also specifies the total number of transmissions required to send a data packet to all the receivers in the multicast tree in AWNs. Fig. 1(b) shows a multicast tree in which we mark the forwarding nodes with double circles. In an ideal multicasting tree structure, the NFNs must be minimum. This reduces the overall network traffic and improves the throughput, making it feasible to send more number of voice packets per second. However, finding a tree with minimum NFNs is a difficult task in AWNs as we explain further.

Recently a multicast routing protocol, Adaptive Demand-Driven Multicast Routing protocol (ADMR) [4] was proposed for efficient multicast data packet delivery in AWNs. ADMR has the lowest normalized packet overhead (NPO, see Section IV-A for definition) compared to other multicast protocols, like ODMRP [6]. Basing on ADMR and exploiting the error resilient properties of Multiple Description Coding (MDC) and path diversity, a multiple tree video multicasting protocol, Robust Demand-driven Video Multicast Routing (RDVMR) protocol was proposed in [5]. RDVMR protocol uses a novel path based Steiner tree heuristic to reduce the number of forwarders in each tree, and constructs multiple (k) trees in parallel with reduced number of common nodes. Therefore, each receiver has k maximally node disjoint paths to the source, along which different MDC descriptions are sent. However, under high packet rates (as we show later in Section IV-A) the RDVMR protocol fails to perform well due to its large overhead and thus limiting its application for voice multicasting.

The problem of finding a minimum cost multicast tree is well known as Steiner Minimal Tree (SMT) problem. For an excellent survey on SMT refer [7], [8]. Karp [9] demonstrated that this problem is NP-complete even when every link has the same cost, by a transformation from the exact cover by 3-sets. There are some heuristic algorithms to compute SMT. For instance, the algorithm in [10] provides a 2-approximation, and Zelikovsky [11] proposed an algorithm which obtains a 11/6-approximation. These solutions are centralized, meaning the multicast source node needs the entire network topology information. Mobility of nodes, or changes in their activity status (from active to passive and vice-versa) may cause changes in any SMT based structure. Therefore, topology changes must be propagated to the multicast source for any centralized solution. This may result in extreme and unacceptable communication overhead in the case of AWNs. The distributed algorithms given in [12] have very high message passing overhead, take a long time to converge, and need to use beaconing for neighbor discovery. Hence, most of the existing protocols in AWNs use Shortest Path Trees (SPTs), which can

* This work was supported by the Microsoft Research University Relations India.

1-4244-0353-7/07/$25.00 ©2007 IEEE

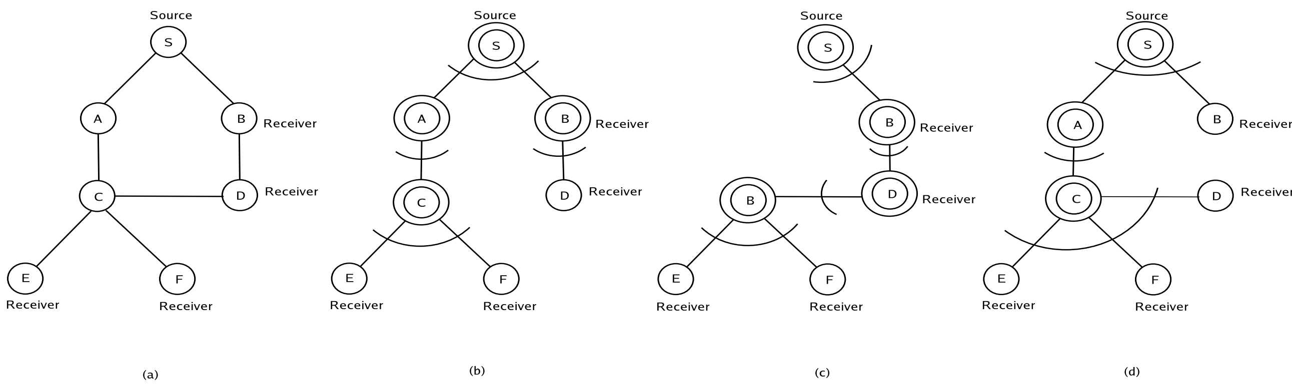

Figure 1. Multicast Trees (a) Original graph (b) Multicast tree generated by SPT heuristic (c) Steiner minimal tree (d) Optimal voice multicast tree; S is the source node and A, B, C, and D are receiver nodes. The number of non-leaf nodes in (b), (c), and (d) are 4, 4, and 3, respectively. Each arc indicates one transmission in the multicast tree and all forwarding nodes in the multicast trees are represented by double circles.

be computed in polynomial time. Lim and Kim [13] analyzed the problem of minimal multicast trees in AWNs, but they defined several heuristics based on the Minimal Connected Dominating Set (MCDS) which are only valid for flooding. In AWNs, when a node 'A' transmits a packet, all its neighbors that are within the transmission range of 'A' can receive the packet due to the broadcasting nature of the medium, $i.e.$, with a single transmission all neighbors of node 'A' can receive the data. This is known as Wireless Multicast Advantage (WMA). Given the broadcasting nature of AWNs, an SMT does not minimize the cost (defined in terms of NFNs) of the multicast tree (see Fig. 1(c)). The cost assignment function used in wired networks is not well-defined for AWNs. That is, by assigning a cost to each link of the graph, existing formulations have implicitly assumed that a given node v needs k transmissions to send a multicast data packet to k of its neighbors. Thus, SMT tries to reduce the overall link cost or node cost or both in the network which is not optimal for AWNs. In this paper we show that the SMT does not generally give an optimal solution. We then formulate the problem of Optimal Voice Multicast Tree (OVMT) problem that contains the minimum number of non-leaf nodes in the multicast tree. We show that the OVMT problem is NP-complete. We then propose a localized distributed heuristic for Minimum number of Transmissions (LDMT) that achieves superior performance compared to the related approaches in terms of voice frame delivery ratio, NPO, and perceived voice quality. The rest of the paper is organized as follows: Section II discusses the OVMT problem and its NP-completeness. We describe the proposed LDMT heuristic algorithm in detail in Section III. In Section IV, we evaluate the performance of LDMT heuristic algorithm through simulations and compare with related protocols. Finally, in Section V we conclude with possible future work.

II. Optimal Voice Multicasting Tree in AWNs

A. Graph Model

An AWN can be modeled by an undirected graph, $G = (V, E)$ where V represents the set of mobile nodes and E represents the set of edges in the network. An edge between two nodes v_1, $v_2 \in V$ exists iff $dist(v_1, v_2) \leq r$ ($i.e.$, v_1, v_2 are within the communication range r). We assume that all links are bidirectional, $i.e.$ if node v_i can communicate with v_j, then v_j can also communicate with v_i.

Definition Given a graph $G = (V, E)$, source node $s \in V$, and receiver set $R \subseteq$ V; R = $\{r_1, r_2, \cdots, r_i; 1 \leq i \leq M\}$, where M is the number of receivers, a tree T is said to be a **multicast tree** iff T has $\{s\} \cup R \subseteq V(T)$ where s and $V(T)$ are root and vertices of tree T, respectively; $V(T) \subseteq V(G)$.

B. Optimal Voice Multicast Tree Problem

Given a graph $G = (V, E)$, source node $s \in V$, and receiver set $R \subseteq V$; $R = \{r_1, r_2, \cdots, r_i; 1 \leq i \leq M\}$, where M is the number of receivers, the OVMT, T^* is defined as, among all multicast trees of G, denoted by $T_1, T_2, T_3, \cdots, T_k$; for some integer k, the number of non-leaf nodes in T^* is minimum $i.e.$, if $l_1, l_2, l_3, \cdots, l_j$, are leaf nodes of T^*; $2 \leq j \leq M$, then $\mid V(T^*) - \{l_1, l_2, l_3, \cdots, l_j\} \mid$ must be minimum compared to all other multicast trees of G. Note that T^* need not be unique. We call the problem of finding OVMT as optimal voice multicast tree problem.

C. Properties of the Optimal Voice Multicast Tree

As observed in Figure 1(d), the number of non-leaf nodes in the OVMT is minimum; which implies that the number of transmissions in the multicast tree is also minimum. Note that unlike SMT, OVMT is not optimal in terms of total number of nodes or links (actually radio links in an AWN) present in the multicast tree. OVMT improves the throughput by minimizing the total number of transmissions required for sending voice packets from the source to all multicast receivers making it highly suitable for high packet rate voice multicasting in AWNs. Each reduction in NFNs results in large number of transmission savings. To understand this fully, consider the following example. Suppose that in a typical voice multicast application the source sends around 15,000 voice packets to all its multicast receivers, $i.e.$, at each hop these 15,000 packets are to be forwarded (transmitted). If the NFNs is less by one, it means we can save 15,000 transmissions in the network which reduces the overall voice traffic in the network considerably and improves the throughput significantly. This also avoids unnecessary energy depletion of nodes in AWNs. We now show that the problem of finding OVMT is NP-complete.

Theorem 1. Given a graph $G = (V, E)$, source node $s \in V$, and receiver set $R \subseteq$ V, the problem of finding $OVMT$ is NP-complete.

Proof: Assume that the $OVMT$ problem can be solved in polynomial time with an algorithm called *OVMT_Algo*. Since $OVMT\_Algo$ can be applied for any number of receivers $1 \leq \mid R \mid \leq n$ where $n = \mid V \mid$. Let the receiver set, R contain all nodes of G, $i.e.$ $\mid R \mid = n$. Using $OVMT\_Algo$, and for all $\vartheta \in V$ as root node and R=V-$\{\vartheta\}$, construct $OVMT$. Let the $OVMT$ for each case be $T_1, T_2, T_3, \cdots, T_i$; where $1 \leq i \leq n$. Each tree T_i contains $\mid V \mid$ nodes, E_i edges, and R_i receivers where R_i=V-$\{\vartheta_i\}$ with ϑ_i as its root node. Of all the trees $T_1, T_2, T_3, \cdots, T_n$, find the trees where s is a non-leaf node. Let these trees be $M_1, M_2, M_3, \cdots, M_k$; where $1 \leq k \leq n$. Now from all the trees $M_1, M_2, M_3, \cdots, M_k$ find the tree (break the ties randomly) whose number of non-leaf nodes is minimum. Call that tree as M_j. M_j fits the definition of Optimal Broadcast Tree (OBT) as M_j contains minimum number of non-leaf nodes, $i.e.$, we solved the OBT problem in polynomial time which is a contradiction to the OBT problem, which has been proved to be NP-complete by Lim and Kin in [13]. Since a special case of $OVMT$ problem ($\mid R \mid = n$) is NP-complete, the $OVMT$ problem must be at least as hard as NP-complete. ■

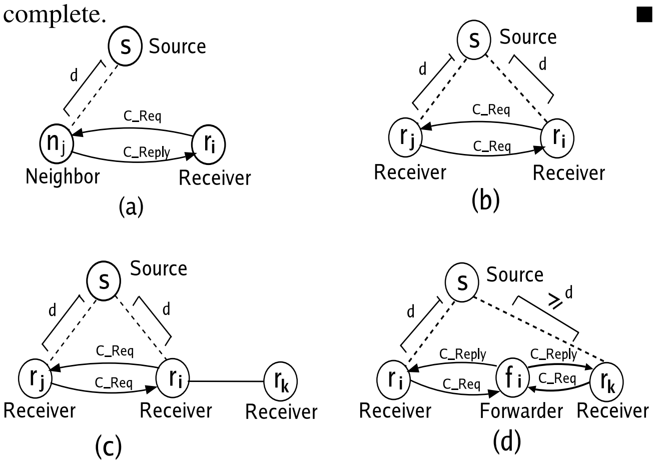

Figure 2. Different possible cases in LDMT heuristic.

Algorithm 1 Localized Distributed heuristic for Minimum number of Transmissions (LDMT)

$R \leftarrow$ all receiver node ids which are not within the transmission range (single-hop) of Source node S

$l_{m_i} \leftarrow$ minimum hop length from source seen by node i

$pl_{m_i} \leftarrow$ previous node along l_{m_i} path seen by node i

$Nrss_i \leftarrow$ number of receivers seen along l_{m_i} path by node i

$N(r_i) \leftarrow$ the set of neighbor nodes of receiver r_i

$N_r(n_i) \leftarrow$ the set of neighbor receivers of a node n_i, $i.e.$, all the receivers that are within the single hop distance to node n_i

$NC(n_i) \leftarrow$ the set of neighbor receivers that are connected to n_i, $i.e.$, all the receivers that are within the single hop distance to node n_i and connected to node n_i to receive multicast voice packets.

Begin

(1) **Source node** S broadcasts $SrcJoinAdvt$ packet: $SrcJoinAdvt(l_{m_i}$ = 0, pl_{m_i} = $S.id$, $Nrss_i$ = 0);

(2) **At node** n_i

if Received non_Duplicate $SrcJoinAdvt$ packet with better hop length, l_{m_i} **then**

Update the variables: l_{m_i}, pl_{m_i}, and $Nrss_i$

if $n_i \in R$ **then** $Nrss_i$ = $Nrss_i$ + 1 **end if**

end if

Forward the $SrcJoinAdvt$ packet: $SrcJoinAdvt(l_{m_i}$ = l_{m_i} + 1, pl_{m_i} = i, $Nrss_i$);

(3) Each receiver, $r_i \in R$, issues a 1-hop broadcast message C_Req with the following 4-tuple: $< r_i.id, l_{m_i}, pl_{m_i}, Nrss_i >$.

(4) Now each $n_j \in N(r_i)$ replies with C_Reply message with the following 6-tuple:
$< Nrss_j, l_{m_j}, pl_{m_j}, N_r(n_j), NC(n_j), n_j.id >$.

(5) Identify the **CONNECTING_NODE** of receiver $r_i \in R$

(a) **if** $r_i \in R$ gets a reply from only one neighbor **then** connect to it. /* this case is shown in Fig. 2(a) */

(b) **if** $r_i \in R$ gets a reply from a neighbor which is also a receiver (say r_j) and if $l_{m_i} = l_{m_j}$ **then**

if $|NC(n_i)| < |NC(n_j)|$ **then**

Connect r_i to r_j /* Since r_j has high connectivity than r_i. See Fig. 2(c) */

else

if $|NC(n_i)| = |NC(n_j)|$ **then**

/*Connectivities of both nodes n_i and n_j are equal: See Fig. 2(b)*/

if $Nrss(n_i) > Nrss(n_j)$ **then**

Connect r_j to r_i /* since r_i has higher $Nrss$ value than r_j */

else

Connect r_j to r_i /* giving more priority to min. node-id */

end if

end if

end if

end if

(c) **if** a receiver r_i has a neighbor f_j (where f_j is a node in the multicast tree which is neither source node nor a multicast receiver node) and if $N(f_j)$ contains a receiver r_k and f_j is at $d+1$ distance from the Source with pl_{m_j} = r_i, **then** r_k must be at least $d+1$ distance away from the Source. This information is useful to take a decision on to which r_i or r_k is to be connected. /* this case is shown in Fig. 2(d) */

(6) The receivers which are able to connect to their **CONNECTING_NODES** will send a JOIN message confirming their willingness to join with the **CONNECTING_NODES**.

(7) Now all **CONNECTING_NODES** act as receiver nodes and issue a 1-hop broadcast message to their neighbors and steps (3) - (6) will be repeated. Nodes that already issued a $JOIN$ message will not respond now.

(8) Step (7) is stopped when all **CONNECTING_NODES** are exhausted.

End

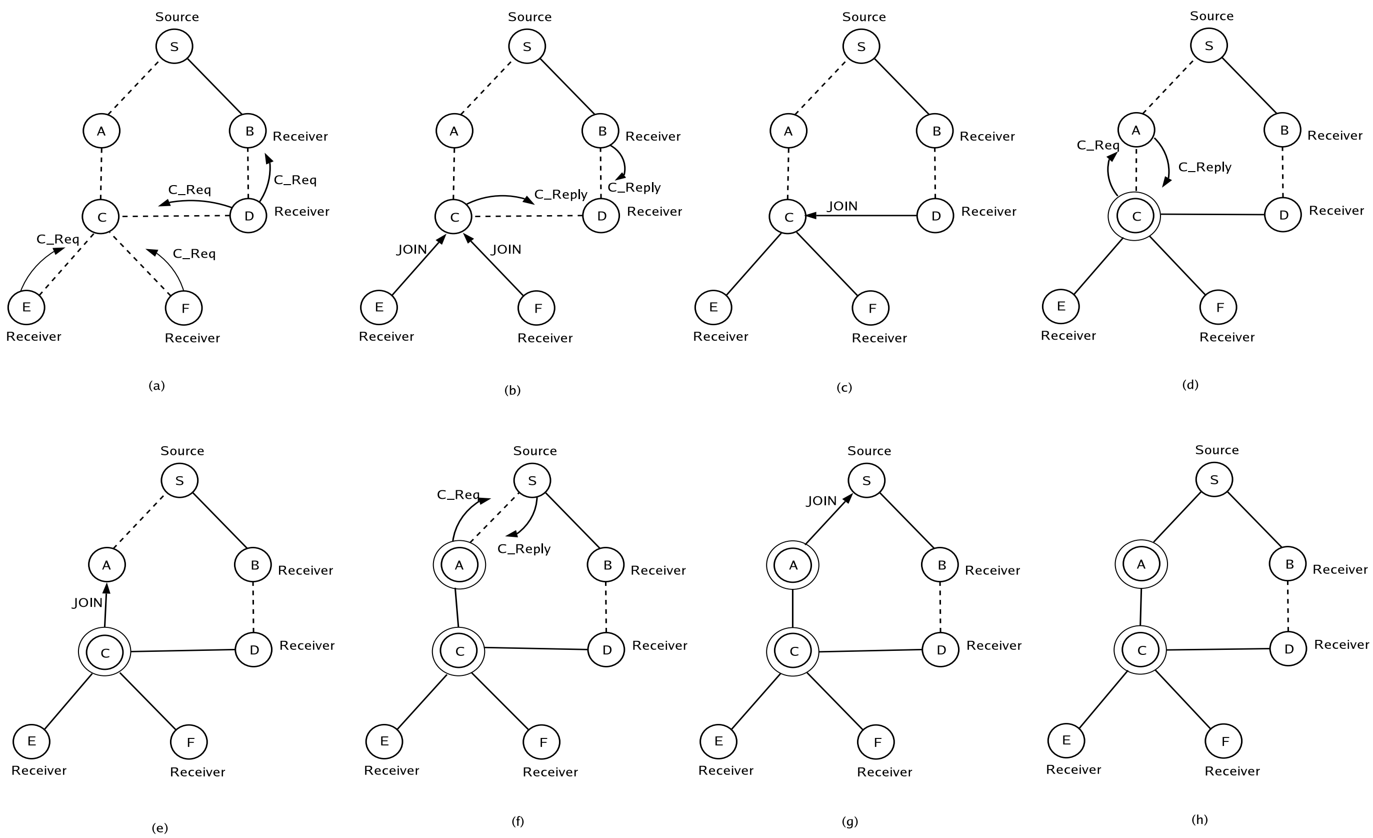

Figure 3. An example that illustrates the LDMT heuristic. S is the source and B, D, E, and F are multicast receivers. A solid line between two nodes indicates that the connection is established between them. A dotted line between two nodes indicates that the connection is not yet established between them. A double circled node represents that it is acting as a forwarding node.

D. Approximation Algorithm

We observe that there exists a closely related problem in graph theory for finding the OVMT, known as Connected Minimum Vertex Cover (CMVC) problem. A simple algorithm for solving the unweighted CMVC problem that gives a factor 2 approximation exists in the literature. That is, we can find a voice multicast tree whose optimality (in terms of NFNs) is at the maximum twice that of OVMT. The idea is as follows. A Depth First Search (DFS) is done on the graph, and all the non-leaf vertices are taken as the nodes in the vertex cover. This clearly induces a connected graph, and the approximation ratio is 2, as shown by Savage [14]. In practice, however this method gives large connected vertex covers and also it is a centralized algorithm making it unsuitable for AWNs. In the next section, we propose an efficient localized distributed heuristic algorithm for AWNs.

III. Localized Distributed Heuristic Algorithm to Approximate Optimal Voice Multicasting Tree

The basic idea of Localized Distributed heuristic for Minimum number of Transmissions ($LDMT$) is to reduce the number of non-leaf nodes in the tree so that it will minimize the total number of transmissions required for voice multicasting. Each receiver tries to connect to a forwarding node that is already feeding at least one another receiver node. The detailed step by step mechanism is given in Algorithm 1. We illustrate the working mechanism of $LDMT$ heuristic with an example. Assume that Fig. 1(a) represents an AWN. Let S be the source node and receiver set $R = \{B, D, E, F\}$. Initially source S broadcasts a $SrcJoinAdvt$ message announcing the availability of voice multicasting (step-1 in the Algorithm 1) service. This $SrcJoinAdvt$ packet contains three fields (i) Minimum hop length path, l_{m_i}, from the source, (ii) Previous node corresponding to the minimum hop path, pl_{m_i}, and (iii) Number of receiver nodes seen ($Nrss_i$) along the minimum hop path. On receiving a non-duplicate $SrcJoinAdvt$ message which has better l_{m_i}, each node saves the current values of l_{m_i}, pl_{m_i}, and $Nrss_i$ and updates them before forwarding the $SrcJoinAdvt$ packet (step-2 in the Algorithm 1). After some $STABILIZATION\_TIME$, assuming that $SrcJoinAdvt$ is received by all receivers in the network, each receiver r_i issues a 1-hop broadcast message C_Req to get the details of its neighbor nodes, in order to find its upstream $CONNECTING\_NODE$ (step-3 in the Algorithm 1). After a $SHORT\_SPAN$, each neighbor of r_i, $n_j \in N(r_i)$, sends C_Reply packet with $< Nrss_j$, l_{m_j}, pl_{m_j}, $N_r(n_j)$, $NC(n_j)$, $n_j.id >$ (step-4 in the Algorithm 1). We define $CONNECTIVITY$, $NC(n_i)$ of a node, n_i, as the set of receiver nodes lying in its transmission range and already connected to node n_i. After receiving the replies each receiver r_i uniquely identifies its $CONNECTING\_NODE$ and issues a JOIN message to its connecting node. There are 4 different cases depending on the type of nodes present in the neighborhood of a receiver node r_i. These sub-cases are shown in Figs. 2(a)-2(d) and are explained in steps 5(a)-5(c) of Algorithm 1. An important point to be considered

TABLE I
SIMULATION PARAMETERS

Parameter	Value	Parameter	Value
Terrain Area	1,200 m x 800 m	Tx. Range	250 m
Channel Capacity	11 Mbps	# of Nodes	75
Mobility Model	*Random Way Point*	Sim. Duration	900 s
Source data rate	50 *pkts/s*	# of Receivers	25
MAC Protocol	802.11 DCF	Voice Frame Size	253 bits
Traffic Type	CBR	*ipt*	20 *ms*

is that while choosing a $CONNECTING\_NODE$, priority must be given to [$NC(n_j)$, l_{m_j}, $Nrss_j$, NodeId] in the same order. That is, while choosing a $CONNECTING\_NODE$, a receiver r_j gives highest priority to $NC(n_j)$, then to l_{m_j}, and so on. Once all receivers get connected to their $CONNECTING\_NODES$, all the $CONNECTING\_NODES$ now act as receivers and the whole process is repeated till all $CONNECTING\_NODES$ are exhausted. Fig. 3 shows a step-by-step mechanism of computing the OVMT using LDMT heuristic. Fig. 3(a) shows when all receivers send a 1-hop broadcast message C_Req. Note that since receiver B is within the transmission range of source S, it will not broadcast any C_Req packet. Each neighbor, n_i, of these receiver nodes replies with $< Nrss_i, l_{m_i}, pl_{m_i}, N_r(n_i), NC(n_i), n_i.id >$ after a small time interval $SHORT\_SPAN$. As shown in Fig. 3(b), receivers E and F find that they have to connect to node C. This is because the pl_{m_i} of receivers E and F is node C and thus node C is the best choice for receivers E and F. Now receivers E and F send a $JOIN$ packet to node C. After receiving the C_Reply from node C, the receiver D finds that it has to connect to node C (since node C is already feeding receivers E and F) and thus it sends a $JOIN$ packet to node C (see Fig. 3(c)). Now node C becomes a receiver temporarily and follows the similar procedure to find its best connecting node. The process is shown in Figs. 3(d)-(g). The final multicast tree is shown in Figure 3(h).

IV. SIMULATION STUDIES

A. Simulation Framework

We use the NS-2 simulation framework [15] to evaluate the performance of LDMT. We compare its performance with ADMR [4] and RDVMR [5]. We compare our protocol with ADMR and RDVMR (for the single tree case) for various scenarios. For all experiments we set the parameters α, β, γ, and λ for RDVMR to 0.03, 0.8, 0.2, and 2, respectively as given in [5]. We evaluate the performance using the following metrics: 1) **Frame Delivery Ratio (FDR)** (the ratio of the average number of voice frames received by each receiver over the number of frames sent by the source), 2) **Number of Transmissions Needed For Multicast** (the total number of transmissions taken by protocol for sending a voice frame to all receivers in the multicast tree), 3) **Normalized Packet Overhead (NPO)** (the ratio of the total number of packets (control and data) exchanged over the total number of data packets received by all the receivers), and 4) **Measurement of Perceptual Evaluation Speech Quality - Mean Opinion Score (PESQ-MOS)**. The PESQ-MOS is evaluated as follows. At each receiver, the voice frames are decoded and the wide band version of ITU perceptual measurement algorithm, PESQ-MOS reference software tool [17] is used to measure their perceived voice quality. The PESQ-MOS reference software tool compares the degraded speech with the reference speech and computes the objective MOS value in a 5-point score ranging from -0.5 (worst) to 4.5 (best). With respect to a original raw voice frame, the voice quality scores of different voice frames are evaluated using PESQ-MOS reference software tool. The evaluated voice quality scores of (a) raw voice frame, (b) decoded AMR-WB voice frame, and (c) decoded bits of AMR-WB voice frame that corresponds to basic quality are 4.5 (Ideal Quality), 3.818 (Optimal), and 2.86, respectively. The optimal quality score (3.818) corresponds to the decoding of AMR-WB (lossy encoded) voice frame assuming no losses in the network.

We modified ADMR protocol to incorporate LDMT heuristic in the NS-2 version 2.1b8. The simulation parameters are shown in Table I. The source sends data throughout the simulation period and 25 of the total nodes are randomly chosen to be receivers. Each of these receivers joins at a random time instant, chosen uniformly from $(4, 450)$ seconds. The receivers do not leave the multicast session. All the results presented in this paper were averaged over 30 simulation runs and all the results conform to 95% confidence levels. Each node moves with some constant speed ($i.e.,$ min speed is equal to max speed) with zero pause time. The playback deadline is 200 ms, if a packet is not received within its playback deadline it is considered lost. We use AMR-WB (Adaptive Multi-Rate Wide Band) [16] speech codec with 12.65 Kbps bit rate with a sample size of 253 bits for sending the voice packets from the source to all the receivers in the multicast session.

B. Simulation Results

1) Number of Transmissions vs. Receivers: We fix the periodicity of flooding $SrcJoinAdvt$ to be 30 seconds in all protocols for uniformity sake with a static scenario (mobility = 0 *m/s*). As observed in Fig. 4, LDMT performs better than ADMR and RDVMR. This is because both ADMR and RDVMR protocols concentrate on finding shortest path between the source and receivers either by using Shortest Path Tree (in case of ADMR) or by using a variant of SMT (in case of RDVMR). More importantly these protocols do not consider Wireless Multicast Advantage and thus they limit themselves for further improvement.

2) Effect of High Packet Rate: Under static scenario, we measure the effect of high packet rate on all the three protocols. It can be seen from Fig. 5 that the FDR of both ADMR and RDVMR protocols decreases rapidly as the data rate increases beyond 30 pkts/s. Since LDMT reduces the NFNs, it can cope up with high packet rate up to 70 pkts/s well without significant reduction in FDR.

3) Effect of Mobility and PESQ-MOS: We set data rate at 50 pkts/s for this experiment. As observed in Fig. 6, the NPO of LDMT is less compared to that of ADMR and RDVMR protocols. This is due to the fact that the number of transmissions (and thus overall voice packet traffic) is lesser in LDMT compared to ADMR and RDVMR protocols. Thus it can sustain mobility induced packet losses more easily

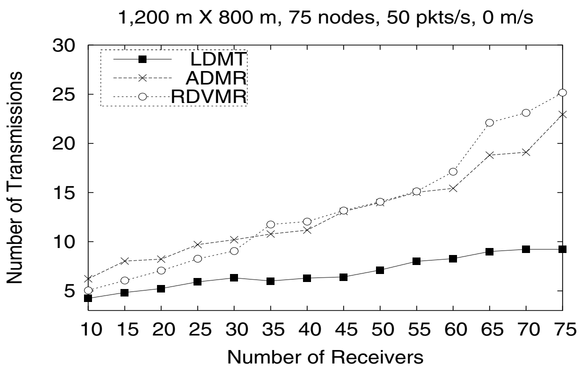

Figure 4. Number of Transmissions vs. Number of Receivers.

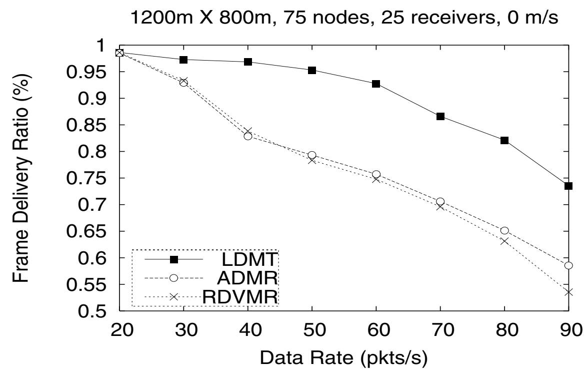

Figure 5. Frame Delivery Ratio vs. Data Rate.

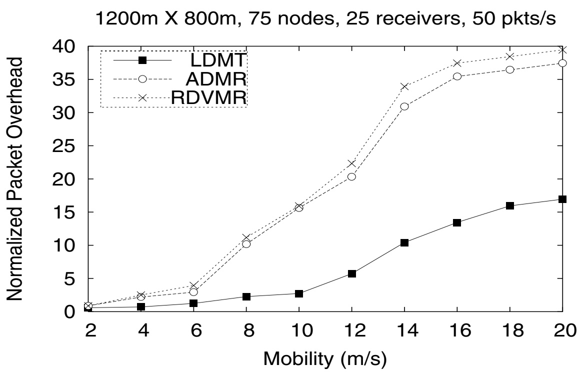

Figure 6. Normalized Packet Overhead vs. Mobility.

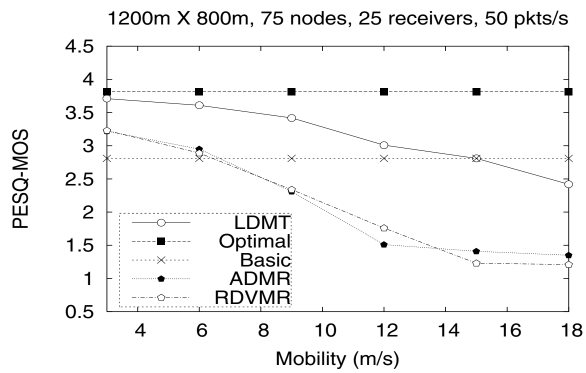

Figure 7. PESQ-MOS vs. Mobility.

compared to ADMR and RDVMR protocols. Fig. 7 shows the measured wide band PESQ-MOS at the destination for varying mobility. As observed in the figure, the LDMT algorithm when incorporated in ADMR outperforms all other protocols due to its minimal overhead (because of its minimum NFNs) at all times. Both optimal and basic scores are also shown in the same figure.

V. Conclusion and Future work

In this paper we proposed an effective low overhead, distributed heuristic called LDMT for voice multicasting in Ad hoc wireless networks. Simulation results showed that LDMT outperformed RDVMR and ADMR in terms of frame delivery ratio, NPO, and PESQ-MOS. For our future work, we explore novel ways of improving both FDR and voice quality by sending forward error protected data along with original voice frames. We also wish to address the congestion and receiver heterogeneity issues of voice multicasting along with an efficient MAC layer approach.

References

[1] N. Jayant and S. W. Christensen, "Effects of Packet Losses in Waveform Coded Speech and Improvements Due to an Odd-Even Sample-Interpolation Procedure", *IEEE Transactions on Communications*, vol. 29, no. 2, pp. 101–109, February 1981.

[2] ITU-T Recommendation, G.114: "One Way Transmission Time", February 1996.

[3] G. Venkat Raju, T. Bheemarjuna Reddy, Shyamnath Gollakota, and C. Siva Ram Murthy, "On Supporting Real-time Speech over Ad hoc Wireless Networks", *in Proc. of IEEE ICON*, pp. 421-426, September 2006.

[4] J. G. Jetcheva and D. B. Johnson, "Adaptive Demand-Driven Multicast Routing in Multi-hop Wireless Ad hoc Networks", *in Proc. of ACM MOBIHOC*, pp. 33–44, October 2001.

[5] D. Agrawal, T. Bheemarjuna Reddy, and C. Siva Ram Murthy, "Robust Demand-Driven Video Multicast over Ad hoc Wireless Networks", *in Proc. of IEEE BROADNETS*, October 2006.

[6] Sung Ju Lee, William Su, and Mario Gerla, "On-Demand Multicast Routing Protocol in Multihop Wireless Mobile Networks", *Mobile Networks and Applications*, vol. 7, no. 6, pp. 441–453, December 2002.

[7] P. Winter, "Steiner Problem in Networks: A Survey", *Networks*, vol. 17, no. 2, pp. 129–167, Summer 1987.

[8] Sudipto Guha and Samir Khuller, "Approximation Algorithms for Connected Dominating Sets", *in Proc. of European Symposium on Algorithms*, pp. 179–193, March 1996.

[9] R.-M. Karp, "Reducibility Among Combinatorial Problems", *in Complexity of Computer Computations*, Plenum Press, NY, USA, 1975.

[10] L. Kou, G. Markowsky, and L. Berman, "A Fast Algorithm for Steiner Trees", *Acta Informatica*, vol. 2, no. 15, pp. 141–145, June 1981.

[11] A. Zelikovsky, "An 11/6 Approximation Algorithm for the Network Steiner Problem", *Algorithmica*, vol. 9, no. 5, pp. 463–473, July 1993.

[12] F. Bauer and A. Varma, "Distributed Algorithms for Multicast Path Setup in Data Networks", *IEEE/ACM Transactions on Networking*, vol. 4, no. 2, pp. 181–191, April 1996.

[13] H. Lim and C. Kim, "Multicast Tree Construction and Flooding in Wireless Ad hoc Networks", *in Proc. of ACM MSWiM*, pp. 61–68, August 2000.

[14] C. Savage, "Depth-First Search and the Vertex Cover Problem", *Information Processing Letters*, vol. 14, no. 5, pp. 233–235, 1982.

[15] ns-2: network simulator, *http://www.isi.edu/nsnam/ns/*

[16] H. Dong, A. Gersho, J. Gibson, and V. Cuperman, "A Multiple Description Speech Coder Based on AMR-WB for Mobile Ad Hoc Networks", *in Proc. of IEEE ICASSP*, vol. 1, pp. 277–280, May 2004.

[17] Proposed Modification to P.862 to Allow PESQ to be Used for Quality Assessment of Wideband Speech, *ITU-T SG12 Delayed Contribution COM-D007-E*, February 2001.