MC-TRACE: MULTICASTING THROUGH TIME RESERVATION USING ADAPTIVE CONTROL FOR ENERGY EFFICIENCY

Bulent Tavli and Wendi B. Heinzelman

Department of Electrical and Computer Engineering,
University of Rochester, Rochester NY

Abstract- In this paper, we present Multicasting through Time Reservation using Adaptive Control for Energy efficiency (MC-TRACE), which is an energy-efficient voice multicasting architecture for mobile ad hoc networks. MC-TRACE is a monolithic design, where the medium access control layer functionality and network layer functionality are performed by a single integrated layer. The basic design philosophy behind the networking part of the architecture is to establish and maintain a multicast tree within a mobile ad hoc network using broadcasting to establish the desired tree branches and pruning the redundant braches of the multicast tree based on feedback obtained from the multicast leaf nodes. Energy efficiency of the architecture is partially due to the medium access part, where the nodes can switch to sleep mode frequently; and partially due to the network layer part where the number of redundant data retransmissions and receptions are mostly eliminated. Furthermore, MC-TRACE achieves high spatial reuse efficiency by keeping the number of nodes taking part in multicasting operation minimal. We evaluated the performance of MC-TRACE through ns simulations and compared with flooding. Our results show that packet delivery ratio performance, energy efficiency and spatial reuse efficiency of MC-TRACE is superior to those of flooding.

INTRODUCTION

Both broadcasting and unicasting are special forms of a more general networking operation, which is multicasting. In multicasting, one or more source nodes convey information to the members of a multicast group, possibly through the use of non-multicast group member nodes within the network.

Multicast routing of voice traffic within a mobile ad hoc network has many applications, especially in military communications. For example, members of a medical or engineering unit within a larger formation of soldiers need a multicasting platform for their group communication needs. Furthermore, it is not possible to restrict the communication platform to a single-hop networking framework. In many situations a platform restricted to single-hop communications will not be enough to fulfill the connectivity requirements of a mobile group. For example, some of the members of a multicast group will not be in reach of a source which is beyond their single-hop transmit/receive range due to extended distance, obstacles or interference. Thus, the need for multi-hop voice multicasting is obvious within a wireless mobile ad hoc networking framework.

The first objective of a multicast protocol is to convey packets from a source to the members of a multicast group with an acceptable quality of service (QoS). QoS in voice communications necessitates (i) maintaining a high enough packet delivery ratio (PDR), which is defined as the ratio of the number of data packets received by the destination node to the number of data packets generated at the source node, (ii) keeping the packet delay low enough, and (iii) minimizing the jitter in packet arrival times. Actually, flooding, which is the simplest group communication algorithm, is good enough to achieve high PDR, provided that the data traffic and/or node density is not very high so that the network is not congested. However, flooding generally is not preferred as a multicast routing protocol due to its excessive use of the available bandwidth. In other words inefficiency of the spatial reuse of flooding prevents its use as an effective multicast routing protocol.

Thus, the second objective of a multicast routing protocol is to maximize the spatial reuse efficiency, which is directly related with the number of retransmissions required to deliver each generated data packet to all members of a multicast group with a high enough PDR. The third objective of a multicast protocol is to minimize the energy dissipation of the network. Minimizing the energy dissipation is crucial to keep the mobile users, equipped with lightweight battery-operated radios, connected to the network.

There are many multicast routing protocols designed for mobile ad hoc networks, which can be categorized into two broad categories: (i) tree-based approaches and (ii) mesh-based approaches. Tree-based approaches create trees originating at the source and terminating at multicast group members with an objective of minimizing a cost function. For example, the cost function to be minimized can be the distance between the source and every destination in the multicast group [1]. A multicast protocol for ad hoc wireless networks (AMRIS) [2] constructs a shared delivery tree rooted at one of the nodes with IDs increasing as they radiate from the source. Local route

This research is funded by CEIS, a NYSTAR designated center for advanced technology, and Harris Corp. RF Communications Division.

recovery is made possible due to this property of IDs, hence reducing the route discovery time and also confining route recovery overhead to the proximity of the link failure.

Mesh-based multicasting is better suited to highly dynamic topologies, simply due to the redundancy associated with this approach. In mesh-based approaches there is more than one path between the source and multicast group members; thus, even if one of the paths is broken due to mobility the other paths are available. On Demand Multicast Routing Protocol (ODMRP) [3] is a mesh-based scheme using a forwarding group concept, where only a subset of nodes forwards the multicast packets via scoped flooding. Instead of using a tree, ODMRP utilizes a mesh structure, which is redundant and robust, to compensate for the frequent route failures and trades-off bandwidth for stability, which comes with redundancy.

Although there are many protocols for multicasting in mobile ad hoc networks [4], to the best of our knowledge there is not a single protocol that jointly optimizes the QoS, spatial reuse efficiency, and total energy dissipation. Thus, in this paper we propose such a distributed algorithm, which is called MC-TRACE (MultiCasting through Time Reservation using Adaptive Control for Energy efficiency).

MC-TRACE is a cross-layer design that incorporates network layer and medium access control (MAC) layer functionality into a single layer; thus, it is a monolithic design. We previously designed MH-TRACE (Multi-Hop Time Reservation using Adaptive Control for Energy efficiency) [5], which is a MAC protocol that combines the advantageous features of both distributed protocols (*e.g.*, IEEE 802.11).and centralized protocols (*e.g.*, PRMA [6]). MH-TRACE achieves very high energy efficiency as well as support for voice QoS in a distributed fashion. While preserving the energy efficiency provided by the MAC layer in idle listening or unnecessary carrier sensing, MH-TRACE also improves the energy efficiency by minimizing the number of retransmissions as well as ensuring that nodes to not receive unnecessary data packets.

The remainder of this paper is organized as follows. Section II describes the MC-TRACE architecture. The simulation environment and results are presented in Section III. Conclusions are drawn in Section IV.

PROTOCOL ARCHITECTURE

MC-TRACE is a network architecture designed for energy-efficient voice multicasting. MC-TRACE is created though the integration of network layer multicasting with the MH-TRACE MAC protocol [5]. We present a brief description of MH-TRACE and a detailed description of MC-TRACE in the following subsections.

MH-TRACE

Multi-Hop Time Reservation using Adaptive Control for Energy efficiency (MH-TRACE) is a MAC protocol that combines advantageous features of fully centralized and fully distributed network protocols for energy-efficient real-time data broadcasting [5]. Figure 1 shows a snapshot of MH-TRACE clustering and medium access for a portion of a distribution of mobile nodes. In MH-TRACE, the network is partitioned into overlapping clusters through a distributed algorithm. Time is organized into cyclic constant duration superframes (T_{SF}) consisting of several frames. Each clusterhead (CH) chooses the least noisy frame to operate within and dynamically changes its frame according to the interference level of the dynamic network. Nodes gain channel access through a dynamically updated and monitored transmission schedule created by the CHs, which eliminates packet collisions within the cluster. Collisions with the members of other clusters are also minimized by the CH's selection of the minimal interference frame. However, inter-cluster interference is not completely eliminated in MH-TRACE due to the limited carrier sensing range of the radios, yet, the benefits of the coordination obtained with MH-TRACE (*e.g.*, high throughput, low energy dissipation, and low jitter) are superior to that which can be obtained with a CSMA type MAC protocol (*e.g.*, IEEE 802.11) [5].

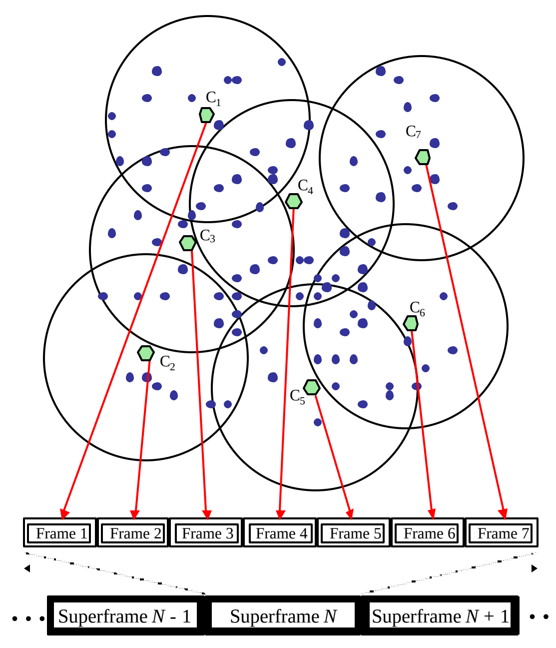

Figure 1. A snapshot of MH-TRACE clustering and medium access for a portion of a distribution of mobile nodes (C1 - C7 are CHs).

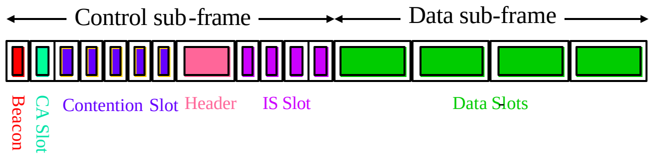

Figure 2. MH-TRACE frame structure.

Ordinary nodes are not static members of clusters, but they choose the cluster they want to join based on the spatial and temporal characteristics of the traffic, taking into account the proximity of the CHs and the availability of the data slots within the corresponding cluster. Each frame consists of a control sub-frame for transmission of control packets, and a contention-free data sub-frame for data transmission (see Figure 2). Beacon packets are used for the announcement of the start of a new frame; CH Announcement (CA) packets are used for reducing co-frame cluster interference; contention slots are used for initial channel access requests; the header packet is used for announcing the data transmission schedule for the current frame; and Information Summarization (IS) packets are used for announcing the upcoming data packets. IS packets are designed to be versatile, and they are crucial in energy saving. Each scheduled node transmits its data at the reserved data slot.

Nodes that are scheduled to transmit data send a short information summarization (IS) packet prior to data transmission. The IS packet includes information about the data packet, where the content of the IS packets can be modified to fit the requirements of different applications.

Instead of frequency division or code division, MH-TRACE clusters use the same spreading code or frequency, and inter-cluster interference is avoided by using time division among the clusters to enable each node in the network to receive all the desired data packets in its receive range, not just those from nodes in the same cluster.

MC-TRACE

MC-TRACE is built on the MH-TRACE architecture and is fully integrated with MH-TRACE, which makes MC-TRACE highly energy efficient. Although, MH-TRACE provides many advantageous features to MC-TRACE (*e.g.*, availability of controlled channel access, organization of the network into clusters) it also restricts the design of MC-TRACE in many ways.

There are five basic building blocks in MC-TRACE: (i) Initial Flooding (IFL), (ii) Pruning (PRN), (iii) Maintain Branch (MNB), (iv) Repair Branch (RPB), and (v) Create Branch (CRB). MC-TRACE creates a broadcast tree through flooding (IFL) and then prunes redundant branches of the tree using receiver-based (or multicast leaf node-based) feedback (PRN). It ensures every multicast node remains connected to the tree while minimizing redundancy and uses IS slots so nodes can keep track of their role in the tree (*e.g.*, multicast relay node) as well as the roles of their neighbors. Finally, MC-TRACE contains mechanisms for allowing broken branches of the tree to be repaired locally (MNB and RBP) and globally (CRB). The MC-TRACE architecture is designed for multiple multicast groups and it can support multiple flows within each multicast group. However, for the sake of clarity we will describe the architecture for a single multicast group with a single source and a single data flow.

1) Initial Flooding: The source node initiates a session by broadcasting packets to its one-hop neighbors. Nodes that receive a data packet contend for channel access, and the ones that obtain channel access retransmit the data they received. Eventually, the data packets are received by all the nodes in the network, possibly multiple times. Each retransmitting node acknowledges its upstream node by announcing the ID of its upstream node in its IS packet, which precedes its data packet transmission (see Figure 2). The source node announces its own ID as its upstream node ID. Initially all retransmitting nodes announce a null ID as their downstream node ID. However, when an upstream node is acknowledged by a downstream node, the node updates its downstream node ID by the ID of this node. The leaf nodes (*i.e.*, nodes that do not have any downstream nodes that are acknowledging them as upstream nodes) continue to announce the null ID as their downstream node ID.

At this point, some of the nodes have multiple upstream nodes (*i.e.*, multiple nodes that have lower hop distance to the source than the current node) and downstream nodes (*i.e.*, multiple downstream nodes acknowledging the some upstream node as their upstream node). A node with multiple upstream nodes chooses the upstream node that has the least packet delay as its upstream node to be announced in its IS slot. Since a retransmitting node indicates its hope distance to the source (HDTS) in its IS packet, it is possible to choose the node with the least HDTS as the upstream node; however, our primary objective is minimizing delay rather than minimizing the multicast tree size. A node updates its own HDTS by incrementing the least HDTS it hears within T_{HDTS1} time. The initial HDTS value is set to max_HDTS, and the HDTS value is again set to max_HDTS if a node does not receive any IS or data packet for more than T_{HDTS2} time, where T_{HDTS2} is larger than T_{HDTS1}.

Multicast group member nodes indicate their status by announcing their multicast group ID in the IS packet (see Figure 3). Nodes that are not members of the multicast group set their multicast group ID to the null multicast group ID. If an upstream node receives an acknowledgement (ACK) from a downstream multicast

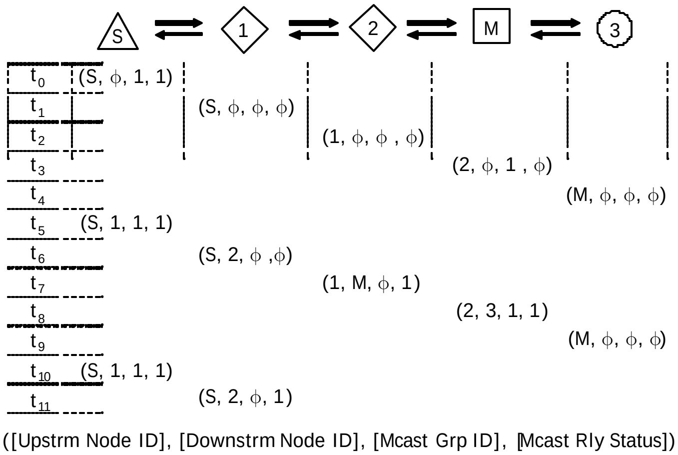

Figure 3. Illustration of initial flooding. Triangles, squares, diamonds, and circles represent sources, multicast group members, multicast relays, and non-relays, respectively. The entries below the nodes represent the contents of ([Upstream Node ID], [Downstream Node ID], [Multicast Group ID], [Multicast Relay Status]) fields of their IS packets (ϕ represent null IDs and t_i's represent time instants).

group member, it marks itself as a multicast relay and announces its multicast relay status by setting the corresponding status (*i.e.*, multicast relay bit) in the IS packet. The same mechanism continues in the same way up to the source node. In other words, an upstream node that gets an ACK from a downstream multicast relay marks itself as a multicast relay. Furthermore, a multicast group member that receives an ACK from an upstream multicast relay marks itself as a multicast relay also. Multicast relay status expires if no ACK is received from any downstream (for both members and non-members of the multicast group) or upstream (only for members of the multicast group) multicast relay or multicast group member for T_{RLY} time. For the sake of simplicity, we assume a link between any node pair is bidirectional at this point; however, this is not necessary for MC-TRACE to operate successfully. Initial flooding results in a highly redundant multicast tree, where most of the nodes receive the same data packet multiple times. Thus, a pruning mechanism is needed to eliminate the redundancies of the multicast tree created by the initial flooding.

2) Pruning: Actually initial flooding and pruning are two mechanisms working simultaneously; however, we describe these as sequential mechanisms to make them easier to understand. During the initial flooding, the multicast relays are determined in a distributed fashion. Pruning uses the multicast relays to create an efficient multicast tree. As described previously, a multicast relay node that does not receive any upstream or downstream ACK for T_{RLY} time ceases to be a multicast relay (for the sake of simplicity, we assume the multicast group members are always the leaf nodes). Furthermore, a node, which is not a multicast relay also ceases to retransmit the multicast data if it does not receive an ACK from any downstream node.

Figure 4 illustrates the operation of the pruning mechanism. After the initial flooding all the nodes receive the data packets and they determine their upstream and downstream nodes. Multicast relays are also determined. Nodes 4, 5, and M along with S are multicast relays. However, nodes 1, 2, and 3 are not multicast relays, because there is not a multicast group member connected to that branch of the network. Node-3 will cease retransmitting the packets that it received from its upstream node-2 T_{RLY} time after its first retransmission of data, because no node is acknowledging its data transmissions. However, until that time node-3 acknowledges its upstream node, which is node-2. Node-2 ceases retransmitting packets $2T_{RLY}$ times after its first data transmission. Note that node-2 acknowledges its upstream node (node-1) for $2T_{RLY}$ time. Node-1 ceases retransmitting $3T_{RLY}$ time after its first data transmission. Thus, the redundant upper branch, where no multicast group members are present, is pruned.

Unlike the upper branch, the lower branch is not pruned due to the fact that the lower branch has a multicast node as the leaf node. Node-M acknowledges the upstream node (node-5) upon receiving the first data packet. Since node-5 receives an ACK form its downstream node (node-M) and also node-M indicates its multicast group membership in its IS packet, node-5 marks itself as a multicast relay and announces its status in its following IS transmission. Upon receiving that IS packet from its downstream node (node-5), node-4 marks itself as a multicast relay also. Thus, the branch of the multicast tree consisting of node-4, node-5, and node-M is created in a distributed fashion. When compared to completion of the pruning of the upper branch the completion of the creation of the lower branch is realized in much shorter time.

Although in most cases initial flooding and pruning are capable of creating an initial efficient multicast tree, they are not always capable of maintaining the multicast tree in a mobile network. Thus, the need for additional mechanisms to repair broken branches is obvious. Maintain Branch, Repair Branch, and Create Branch

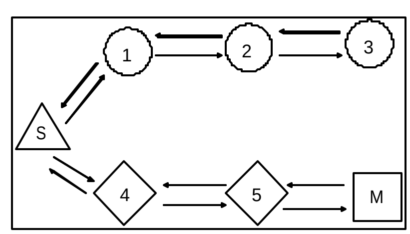

Figure 4. Illustration of pruning and multicast tree creation.

mechanisms are utilized to maintain the multicast tree.

3) Maintain Branch: Some of the multicast group members are not multicast relays. The upper panel of Figure 5 illustrates such a situation. Multicast node (node-M1) is a multicast relay, which is indicated by the two-way arrows; whereas node-M2 is not a multicast relay – it just receives the packets from the upstream node (node-2). Hence, node-M2 does not acknowledge node-2 (node-2 is acknowledged by node-M1. Note that any node can acknowledge only one upstream and one downstream node with a single IS packet. When node-M1 moves away from node-2's transmit range and enters node-1's transmit range, it either begins to acknowledge node-1 as its upstream node if the transition happens in less than T_{RLY} time (*i.e.*, node-M1's multicast relay status does not expire before T_{RLY} time) or just receives the data packets from node-1 without acknowledging node-1 if node-M1's transition takes more than T_{RLY} time. In any case, node-2 does not receive any ACK from node-M1, and starts to set its downstream node ID as the null ID. However, node-2 does not cease retransmitting data packets that it receives from its upstream node (node-1) instantly, because, a multicast relay does not resets its status for T_{RLY} time and continues to retransmit data packets.

Although node-M2 does not acknowledge any node, it monitors its upstream node through IS and data packets. When the upstream node of a multicast group member node (*i.e.*, node-M2) announces null ID as its downstream node ID, the multicast node (M2) starts to acknowledge the upstream node by announcing the ID of the upstream node (node-2) as its upstream node in its IS packet. Thus, node-2 continues to be a multicast relay and node-M2 becomes a multicast relay after receiving a downstream ACK from its upstream node (node-2). Actually, the situation illustrated in Figure 5 is just one example for MNB mechanism. There are several other situations that can be fixed by the MNB mechanism.

The MNB mechanism does not necessarily create a new branch, yet it prevents an existing operational branch from collapse. However, just maintaining the existing multicast relays is not enough in every situation. There are situations where new relays should be incorporated to the tree.

4) Repair Branch: After a node marks itself as a multicast relay, it continuously monitors its upstream node to detect a possible link break between itself and its upstream multicast relay node, which manifests itself as the interruption of the data flow without any prior notification. If such a link break is detected, the downstream node uses the RPB mechanism to fix the broken link. Figure 6 illustrates an example of a network topology where a branch of the multicast tree is broken due to the mobility of a multicast relay and fixed later by the RPB mechanism. The upper panel of Figure 6 shows a multicast tree formed by the source node, node-S, multicast relay nodes, node-1 and node-2, and the multicast group node, node-M, which is a multicast relay as well. Node-3 is neither a multicast relay node nor a multicast group member; however, it receives the IS packets from node-1, node-2, and node-M (*i.e.*, node-3 is in the receive range of all three nodes). After some time, as illustrated in the lower panel of Figure 6, node-2 moves away from its original position and node-1 and node-2 cannot hear each other; thus, the multicast tree is broken. At this point node-2 realizes that the link is broken (*i.e.*, it does not receive data packets from its upstream node anymore) and the RPB mechanism is used to fix the broken tree. Node-2 sets its RPB bit to one in the IS packets that it sends. Upon receiving a RPB indicator, all the nodes in the receive range start to retransmit data packets as they do in the initial flooding stage. One of these nodes, which is node-3 in this scenario, replaces node-2 as a multicast relay node and the multicast tree branch is repaired. We assumed node-3 remains in the transmit range of node-1, node-2, and node-M even after node-2 moved away from node-1's transmit range. However, even if node-3 was not in the transmit range of node-2, the tree can again be fixed. Since node-M does not receive any data packets from its upstream node (node-2), it sets its RPB bit to one and announces this in its IS packet. Upon receiving the RPB of node-M, node-3 starts

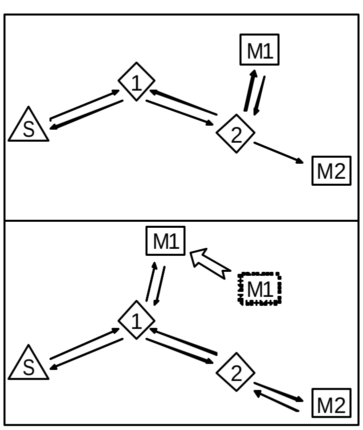

Figure 5. Illustration of the Maintain Branch mechanism.

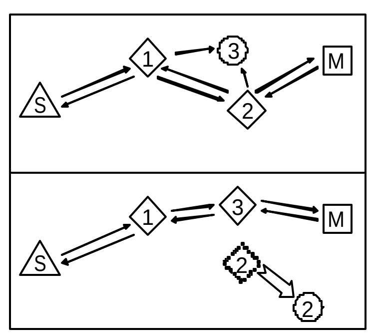

Figure 6. Illustration of the Repair Branch mechanism.

to relay data packets, and upon receiving an upstream ACK from node-M, marks itself as a multicast relay.

Both MNB and RPB are limited scope maintenance algorithms (*i.e.*, they can fix mostly one-hop tree breaks). However, in a dynamic network, limited scope algorithms are not capable of completely eliminating multicast tree breaks or, in some cases, the total collapse of the multicast tree. Thus, the create branch (CRB) mechanism is needed.

5) Create Branch: It is possible that due to the dynamics of the network (*e.g.*, mobility, unequal interference) a complete branch of a multicast tree can become inactive, and the leaf multicast group member node cannot receive the data packets form the source node. Figure 7 illustrates a network with one active branch, composed of the nodes S, 1, 2 and M1, and one inactive branch, composed of nodes 3, 4, 5, and M2. The double arrows indicate an active link with upstream and downstream ACKs. Dash-dotted arrows indicate an inactive link. The numbers below the nodes show their HDTS, which they acquired during previous data transmissions. One situation that can create such inactivity is that the upstream ACKs of nodes 8 and M1 are colliding and node-5 cannot receive any downstream ACK. Thus, node-5 ceases to relay packets, which eventually results in silencing all the upstream nodes up to the source (*i.e.*, if node-5 does not get any downstream ACKs it ceases acknowledging its upstream node, node-4, after T_{RLY} time, which results in silencing of node-4 in $2T_{RLY}$ time and node-3 in $3T_{RLY}$ time).

If a multicast group member, which is node-M2 in this scenario, detects an interruption in the data flow for T_{CRB} time, it switches to Create Branch status and announces this information via a CRB packet. A CRB packet is transmitted by using one of the IS slots, which is chosen randomly. Upon receiving a CRB packet, all the nodes in the receive range of the transmitting node switch to CRB status if their own HDTS is lower than or equal to the HDTS of the sender (*e.g.*, node-5, which has an HDTS of 4, switches to CRB status; however, node-10, which has an HDTS of 5, does not). When a node switches to CRB mode, it starts to relay the data packets if it has data

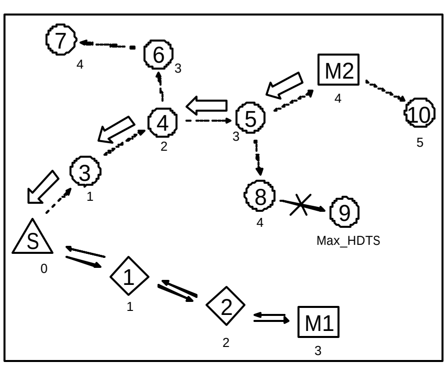

Figure 7. Illustration of the Create Branch mechanism.

packets for the desired multicast group. If it does not have the desired data packets, it propagates the CRB request by broadcasting a CRB packet to its one-hop neighbors. This procedure continues until a node with the desired data packets is found, which is illustrated by the block arrows in Figure 7. After this point, the establishment of the link is similar to the initial flooding followed by pruning mechanisms. However, in this case only the nodes in CRB mode participate in data relaying. Looking at the initial collapse of the branch, we see that node-8 does not participate in CRB due to its HDTS and it does not create interference for node-M2 in this case.

There are several mechanisms in MC-TRACE that provide energy efficiency: (i) nodes are in the sleep mode whenever they are not involved in data transmission or reception, which saves the energy that would be wasted in idle mode or in carrier sensing, and (ii) nodes can selectively choose what data to receive based on information from the IS packets, enabling the nodes to avoid receiving redundant data (*i.e.*, multiple receptions of the same packet). Note that each data packet has a unique ID, which is formed by combining the source node ID and the sequential packet ID. The sequence number need not be greater than that a few bits because data packets do not stay in the network for long due to the real-time requirements of the voice traffic. For example, with a packet drop threshold (T_{drop}) of 150 ms and packet generation period of 25 ms, there can be at most seven packets originated from a single source, simultaneously.

Although the mechanisms of MC-TRACE are fairly simple on their own, as a unified entity they create a robust architecture capable of handling complicated network dynamics, as it is shown by the simulation results.

SIMULATIONS

To test the performance of MC-TRACE and to compare with IEEE 802.11 based flooding, we ran simulations using the ns-2 simulator. We used the energy and propagation models discussed in [5]. Simulation parameters are presented in Table I.

We used the random way-point mobility model for nodes moving within a 1 km by 1 km area. Node speeds are chosen from a uniform random distribution between 0.0 m/s and 5.0 m/s with zero pause time. There are 100 mobile nodes in our scenario and the source node is located in the center of the network. The multicast group has five members excluding the source node.

A performance comparison of MC-TRACE and flooding is presented in Table II. Both the average and the minimum packet delivery ratios (PDR) of the multicast group members for MC-TRACE are 99 %, whereas those of flooding are 83 % and 82 %. Average PDR is the average PDR of the multicast group member nodes' PDRs.

Table I. Simulation parameters.

Variable	Description	Value
N/A	Number of nodes	101
N/A	Network Area	1 km × 1 km
N/A	Transmit range	250 m
N/A	Carrier sense range	507 m
T_{drop}	Packet drop threshold	150 ms
P_T	Transmit power	0.6 W
P_R	Receive power	0.3 W
P_I	Idle power	0.1 W
P_S	Sleep power	0.0 W
C	Channel rate	2 Mbps
S	Source rate	32 Kbps
N/A	Data packet payload	100 bytes
N/A	Data packet overhead	10 bytes
N/A	Control Packet size	10 bytes
N/A	Header packet size	22 bytes
T_{SF}	Superframe time	25 ms
T_{RLY}	Relay status expiration time	$5T_{SF}$
T_{CRB}	CRB time	$6T_{SF}$
T_{HDTS1}	HDTS decrement time	$20T_{SF}$
T_{HDTS2}	HDTS expiration time	$40T_{SF}$

Minimum PDR is the PDR of the multicast node with minimum PDR. The difference in PDRs is due to the high congestion and consequent collisions in flooding.

Both the average and minimum data packet delays of flooding are less than those of MC-TRACE due to the restricted channel access of MC-TRACE. On the other hand, jitter obtained with flooding is 15 times the jitter obtained with MC-TRACE.

Average multicast tree size (MTS_{AVG}) is an appropriate metric to evaluate the spatial reuse efficiency. We determine the MTS_{AVG} by dividing the total number of transmitted data packets from all nodes to the total number of transmitted data packets from the source node. MC-TRACE MTS_{AVG}, 11, is 13 % of MTS_{AVG} of flooding.

Table II. Performance comparison of MC-TRACE and Flooding.

	MC-TRACE	Flooding
PDR_{AVG}	0.99	0.83
PDR_{MIN}	0.99	0.82
$Delay_{AVG}$	49 ms	45 ms
$Delay_{MAX}$	78 ms	55 ms
$Jitter_{AVG}$	2 ms	30 ms
MTS_{AVG}	11	84
$ED_{MC\text{-}AVG}$	50.1 mJ/s	232.7 mJ/s
$ED_{MC\text{-}MAX}$	62.4 mJ/s	254.3 mJ/s
$ED_{AN\text{-}AVG}$	39.4 mJ/s	246.3 mJ/s
$ED_{AN\text{-}MAX}$	62.4 mJ/s	272.9 mJ/s
$TED_{AN\text{-}AVG}$	1.5 mJ/s	8.8 mJ/s
$RED_{AN\text{-}AVG}$	7.0 mJ/s	73.3 mJ/s
$CSED_{AN\text{-}AVG}$	7.4 mJ/s	133.6 mJ/s
$IED_{AN\text{-}AVG}$	15.5 mJ/s	30.6 mJ/s
$SEP_{AN\text{-}AVG}$	8.0 mJ/s	0.0 mJ/s

MC-TRACE average and maximum energy dissipations ($ED_{MC\text{-}AVG}$ and $ED_{MC\text{-}MAX}$) for the multicast nodes are 50.1 mJ/s and 62.4 mJ/s, respectively. Flooding average and maximum multicast node energy dissipations are 365 % and 307 % more than those of MC-TRACE.

Average and minimum energy dissipations for all nodes ($ED_{AN\text{-}AVG}$ and $ED_{AN\text{-}MAX}$) are 39.4 mJ/s and 62.4 mJ/s, respectively, for MC-TRACE and 246.3 mJ/s and 272.9 mJ/s, respectively, for flooding. The difference between the transmit energy dissipation ($TED_{AN\text{-}AVG}$) is directly related with the MTS. MC-TRACE receive energy dissipation ($RED_{AN\text{-}AVG}$) is 9.5 % of that of flooding due the packet discrimination (*i.e.*, redundant versions of the same packet are not received by the nodes in MC-TRACE by monitoring the IS packets). Carrier sense energy dissipation ($CSED_{AN\text{-}AVG}$) of flooding is the dominant energy dissipation term, which constitutes 54 % of the total energy dissipation. Idle energy dissipation ($IED_{AN\text{-}AVG}$) of MC-TRACE is approximately half of the energy dissipation of flooding. Flooding sleep energy dissipation ($SED_{AN\text{-}AVG}$) is zero because IEEE 802.11 never goes to sleep mode.

CONCLUSION

In this paper we described MC-TRACE, which is an energy efficient voice multicasting architecture for mobile ad hoc networks. We compared the performance of MC-TRACE with flooding in terms of packet delivery ratio, delay, jitter, spatial reuse efficiency, and energy dissipation through ns-2 simulations. Our initial results show that MC-TRACE performance is much better than the performance of flooding. Although flooding is not the best multicast algorithm, comparisons with flooding show the performance improvement we can obtain by using MC-TRACE. In the future we will compare MC-TRACE with existing multicast protocols like ODMRP and AMRIS.

REFERENCES

[1] D. Bertsekas and R. Gallager, *Data Networks*, Prentice Hall, 1992.

[2] C. W. Wu and Y. C. Tay, "AMRIS: a multicast protocol for ad hoc wireless networks," *in Proc. MILCOM*, vol. 1, pp. 25-29, 1999.

[3] S. J. Lee, M. Gerla, and C. C. Chiang, "On-demand multicast routing protocol," *in Proc. IEEE WCNC*, pp. 1298-1304, 1999.

[4] S. Athanassopoulos, I. Caragiannis, C. Kaklamania, and P. Kanellopoulos, "Experimental comparison of algorithms for energy-efficient multicasting in ad hoc networks," *LNCS*, vol. 3158, pp. 183-196, 2004.

[5] B. Tavli and W. B. Heinzelman, "MH-TRACE: Multi hop time reservation using adaptive control for energy efficiency," *IEEE J. Select. Areas Commun., vol. 22, pp. 942-953, 2004*

[6] D. J. Goodman and S. W. Wei, "Efficiency of packet reservation multiple access," *IEEE Trans. Vehic. Technol.*, vol 40, pp. 170-176, Feb. 1991.