Multicasting Sustained CBR and VBR Traffic in Wireless Ad-Hoc Networks

George D. Kondylis, Srikanth V. Krishnamurthy, Son K. Dao, and Gregory J. Pottie*
HRL Laboratories, LLC, 3011, Malibu Canyon Road, Malibu, CA 90265
*Electrical Engineering Department, University of California, Los Angeles
E-Mail: kondylis@wins.hrl.com, krish@hrl.com, skdao@hrl.com, pottie@icsl.ucla.edu
Tel: (310) 317-5785, Fax: (310) 317-5695

Abstract— **Wireless ad-hoc networks consist of mobile nodes forming a dynamically changing topology without any infrastructure. Multicasting in a wireless ad-hoc network is difficult and challenging. In this paper we propose a novel protocol, Wireless Ad-hoc Real-Time Multicasting (WARM), for multicasting real-time (CBR and VBR) data among nodes in a wireless ad-hoc network. The protocol is distributed, highly adaptive and flexible. Multicast affiliation is receiver initiated. The messaging is localized to the neighborhood of the receiving multicast member and thus the overhead consumed is low. The protocol enables spatial bandwidth reuse along a multicast mesh (a connected structure of multicast group members). The real-time connection is guaranteed quality of service (QoS) in terms of bandwidth. For VBR traffic, the trade-off between reserved and random-access bandwidth for a specific packet loss rate is studied. The protocol is self-healing in the sense that the mesh structure has the ability to repair itself when members either move or relays fail. We present simulation results to demonstrate features of the protocol and show that the throughput is above 90% for pedestrian environments.**

I. Introduction

A wireless ad-hoc network consists of a collection of "peer" mobile nodes, capable of communicating with each other, and forming a dynamically changing network with no infrastructure. In order to route packets to a destination node, each node in the wireless ad-hoc network has to use other nodes in the network as relays. It is therefore essential that the nodes in the network establish routing among themselves. The routes keep changing as the nodes move or the environment changes (due to fading/interference). A number of routing protocols have been proposed to enable routing in such an environment [1], [2], [3]. However, these protocols do not support real-time traffic, nor do they attempt to guarantee any kind of QoS. None of the protocols address multicast streaming of real-time data in a wireless ad-hoc network environment. In this paper we attempt to provide a coarse grained notion of dedicated service in a wireless ad-hoc network. Bandwidth will be reserved in terms of time division multiple access (TDMA) slots for a multicast session and the multicast structure will be built so as to avoid collisions and jamming of transmissions of mutlicast group members. CBR multicast data is streamed through the reserved channels from the source to the destination periodically, thereby ensuring a bound on delay jitter, while for VBR traffic we study the trade-off between reserved and random-access bandwidth in order to achieve a certain packet loss rate. Our approach is that of concurrent TDMA *collision-free slot scheduling* and route building, so that a multicast mesh is created with reserved bandwidth along the different routes.

A lot of work has been previously done on broadcast scheduling in ad-hoc networks, which is akin to our problem [4], [5], [6]. In [4], broadcast scheduling was shown to be a NP-complete problem. The broadcast scheduling problem can be stated as follows:

- In an ad-hoc network, how can one schedule transmissions among nodes, such that, when a node transmits a packet, every neighboring node receives it? Note that the scheduling protocol needs to ensure that when two nodes transmit, their packets do not collide at a third node (hidden terminal). It should also ensure that two neighboring nodes do not transmit at the same time, since they will be unable to listen to each other's transmissions.

Any of the algorithms developed for solving the above problem could be used to multicast data from a single source in an ad-hoc network. However, multicasting is different and has properties that can be exploited to improve the spatial re-use of bandwidth (slots):

- In multicasting, not all nodes need to transmit: only a subset of nodes relay packets to all other nodes.
- Neighboring nodes that relay multicast packets from a single session to different "children" can simultaneously transmit packets if they do not jam the reception of any other neighboring node. That is, nodes that relay at the same time the same packet, do not need to hear each other's transmissions, because these transmissions are not intended for each other.

In our protocol, it is up to the receivers to join the multicast session and bandwidth is reserved by means of signaling packets exchanged between one hop neighbors. In order to satisfy its bandwidth requirement, a node can connect to different "parent" nodes in different slots. Multicasting is therefore done on a more robust *mesh* [3], instead of a tree. Some links will eventually fail due to mobility, but by using the same signaling mechanism, receivers will change parent nodes, bandwidth will be reassigned and the multicast mesh will be re-configured for the new topology. Through simulations we find that at pedestrian speeds, throughputs above 90% are achieved, when our protocol is used.

II. TDMA Frame Structure

We assume half-duplex transceivers and hence, nodes cannot transmit and receive simultaneously. Therefore, since we are dealing with periodic traffic (either CBR or VBR), we assume that the slotted time is grouped into super-frames, consisting of two frames each, frame 0 and frame 1. In one of these frames a node receives data and in the other it transmits, if it is a relay. We denote the transmit frame of node N_i by F_i; F_i takes on a value of 0 or 1 depending on whether N_i can transmit in frame 0 or in frame 1 respectively. Figure 1 depicts an example, showing a super-frame and its constituent frames, the first one of which is a transmit frame while the second, a receive frame. The reserved and random-access portions of each frame are also shown.

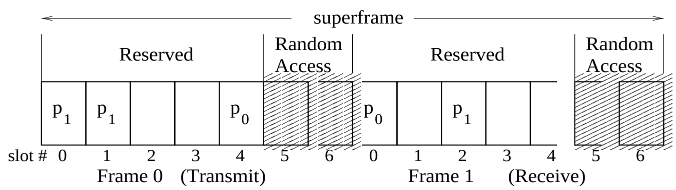

Fig. 1. TDMA frame structure. The node receives in frame 1 and transmits in frame 0 (hence $F = 0$).

Multiple slots can be reserved for a session by a node, and used in each frame. Packets transmitted by the source in each frame are numbered sequentially for that frame and each node marks its receive (transmit) slots with the *frame-sequence number* of the packet to be received (transmitted) in that slot. For example, Figure 1 shows that a node receives packets with frame-sequence numbers 0 and 1 in slots 0 and 2 and transmits packet 0 in slot 4 and packet 1 in slots 0 and 1 (multiple transmissions are for supporting different children that cannot receive in the same slot).

Packets in a frame that are in excess of the reserved number of slots are transmitted/received in the random-access portion of the frame (this would happen in the case of VBR traffic for packets in excess of the current mean source rate). The parent node notifies its children of the specific random-access slots they have to listen to by *appending* the relevant information to packets transmitted during the reserved portion of the bandwidth.

Transmit scheduling for signaling (i.e. control information) between nodes is done in a round-robin fashion on a separate channel. The type of control information exchanged will be described in detailed in a later section.

III. Channel Model and Connectivity

We assume that all nodes transmit with the same power, P_T. The power received by a node N_0 due to the transmission of a node N_j is given by $P_{j0} = G_{j0} P_T$, where G_{j0} denotes the power attenuation on the path between nodes N_0 and N_j. Here, we assume that G_{ij} follows a simple propagation decay law, i.e. $G_{ij} = d_{ij}^{-\eta}$, where d_{ij} is the distance between nodes N_i and N_j and η is the power loss exponent. The signal to interference ratio (SIR) at node N_0, when node N_j transmits, is given by:

$$\mathrm{SIR} = \frac{G_{j0}}{\sum_{i \neq j} G_{i0}} \tag{1}$$

where nodes N_i have frame numbers $F_i = F_j$, and transmit in the receive slot of node N_0. A packet is received correctly if its SIR is above a certain threshold γ. Finally, we assume that, in the absence of co-channel interference a node can communicate with nodes up to a distance $d_{\max}$ away. This distance typically depends on the transmit power P_T, the noise floor and the sensitivity of the receivers.

In the following sections it will be assumed that nodes have knowledge of the path losses to all of their one-hop neighbors. Indeed, the slow variations of the channel, due to path loss and shadow fading are easily tracked, in contrast to fast fading variations due to multipath which, usually, cannot be estimated easily [9].

Since nodes might have more than one reserved slot in a TDMA frame, they can rely on different parents to receive packets in the different slots. This adds a degree a flexibility to the reservation process, since a single parent node may not be able to relay certain packets in a frame, either because it does not have them, or because it does not have enough slots to transmit all the packets, or because some of its children may experience too high an interference in the slots where the parent can transmit the packets. Consider for example Figure 2. Let all the nodes shown in the figure have three slots reserved for receiving packets p_0, p_1, and p_2. Initially, node N_0 relays these packets to node N_2 in slots 2, 3, and 4. However, node N_1 that is relaying the same packets to node N_3 in slots 0, 1, and 2 strays into the neighborhood of node N_2, and thereby causes interference to node N_2's transmission in slot 2. Thus, node N_2 will no longer receive packet p_0. If node N_0 cannot relay p_0 to N_2 in some other slot (for any of the reasons mentioned above), N_2 can receive that packet from node N_1, and therefore should now attempt to become a child of that node also, with the intent to receive a packet p_0 from that node in slot 0. Thus, in this scenario nodes receive packets from multiple parents.

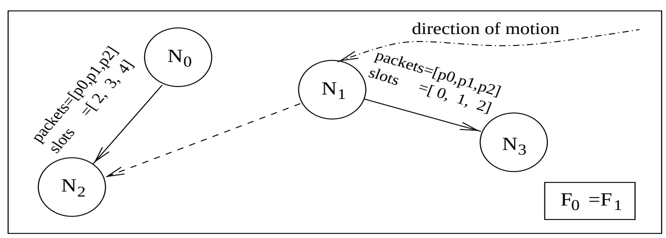

Fig. 2. Node N_1's transmissions interfere the reception at node N_2 in slot 2. Node N_2 can connect to node N_1 to receive packet p_0 in slot 0, if it cannot receive p_0 in another slot from N_0.

It should be noted here that, in the case of VBR traffic, there is a drawback in having multiple parents relay packets to a single child: if there are packets to be transmitted in random-access slots, these will be transmitted by all parents, although only one actually needs to transmit, for the packets to be received by the child. This will increase the possibility of collisions in the random-access slots. Considering Figure 2 for example, if nodes N_0 and N_1 are parents of node N_2, and they have packets that must be relayed to N_2 in random-access, both N_0 and N_1 will transmit these packets, whereas only N_0's transmission would be enough for the packets to be relayed to N_2.

IV. Overview of WARM

WARM deals primarily with the transmission scheduling problem for a multicast session in a wireless ad-hoc network. The mechanism will be presented, by which a node reserves TDMA slots and attaches itself to the multicast mesh, when at least one of its neighbors is a multicast member. We do not consider the case in which none of the neighbors is a multicast member. In such a case, a search might be initiated to find a route from the member to the existing multicast mesh and all the nodes along the route from that member to the mesh will be forced to join the multicast session as relays [3]. Our protocol is receiver initiated, i.e. it is up to the receiver to attempt to connect to the multicast mesh. No special routing information is needed at relay nodes; they would only need to maintain the IDs of their children, the slots with which they transmit to them, and the frame-sequence numbers of the packets they transmit in each slot. Of course, each node will try to attach itself to relays that are as close to the source as possible, but minimum hop routing is a secondary consideration, when a node cannot reserve enough bandwidth from minimum hop relays. Finally, maintenance information, which will be detailed below, is only exchanged among one hop neighbors and is not propagated to more distant nodes.

A. Data structures maintained at each node

A node is characterized by a set $\mathcal{Q}$ (for a particular multicast session), defined as:[1]

$$\mathcal{Q} = \{ID, HC, F, RS, RS_{\text{des}}, RS_{\min}, \mathbf{PID}, \mathbf{RxSlot}, \mathbf{RxSeq}, \mathbf{SIR}, \mathbf{G_p}, \mathbf{TxSlot}, \mathbf{TxSeq}, \mathbf{US}\} \quad (2)$$

where:

- ID: the node's unique ID;
- HC: the node's hop-count, i.e. the number of hops from the source;
- F: a bit which indicates the transmit frame of the super-frame (Figure 1);
- RS: the number of receive slots that are *currently* reserved by the node;
- RS_{des}: the desired number of reserved slots, as estimated by the node, based on the current traffic load;
- $RS_{\min}$: the minimum acceptable number of reserved slots. This depends on the acceptable packet loss rate[2]. Note that, RS and the available number of random-access slots determine the *current* packet loss rate at the node;
- $\mathbf{PID}$: a vector which represents the parents of the node, $PID[i]$, $i = 0, 1, \ldots, RS-1$. Note that all parents must have identical hop counts, and that some or all of the elements of $PID[i]$ might be identical (i.e. the same parent transmits multiple packets in each frame to its child);
- $\mathbf{RxSlot}$: a vector which lists the receive slots of the node, $RxSlot[i]$, $i = 0, 1, \ldots, RS-1$;
- $\mathbf{RxSeq}$: a vector indicating the frame-sequence in which packets are received in the receive slots, $RxSeq[i]$, $i = 0, 1, \ldots, RS-1$. The node receives packet with sequence number $RxSeq[i]$ in slot $RxSlot[i]$. This is required for sorting packets according to their frame-sequence numbers;
- $\mathbf{SIR}$: a vector containing the SIR of the node in its receive slots $\mathbf{RxSlot}$, $SIR[i]$, $i = 0, 1, \ldots, RS-1$;
- $\mathbf{G_p}$: a vector representing the path losses between the node and each of its parents, $G_p[i]$, $i = 0, 1, \ldots, RS-1$;
- $\mathbf{TxSlot}$: a vector which lists the slots in which the node transmits (empty, if the node is not a relay);
- $\mathbf{TxSeq}$: a vector indicating the sequence in which packets are transmitted in the transmit slots. That is, the node transmits packet with frame-sequence number $TxSeq[i]$ in slot $TxSlot[i]$;
- $\mathbf{US}$: a vector which lists the slot numbers of the transmit slots unusable by the node. These are slots in which the node cannot transmit since such a transmission would cause excessive interference to some neighboring node that receives in the same slot.

As long as a node has at least $RS_{\min}$ slots reserved and it successfully receives packets from its parents in these slots, it will consider itself to be connected to the multicast mesh. If the node receives less than $RS_{\min}$ packets in the reserved portion of the bandwidth, it becomes disconnected, and will set its $RS = 0$ and will attempt to reconnect to the multicast session.

B. Signaling information exchanged

Nodes participating in the multicast session take turns to transmit their status information on a signaling channel in a round-robin fashion[3]. Specifically, a node N_i will transmit the following set in its own signaling slot:

$$\mathcal{Q}_T = \{ID, HC, F, RS, \mathbf{RxSlot_1}, \mathbf{RxSeq_1}, \mathbf{PID_1}, \mathbf{TxSlot}, \mathbf{TxSeq}, \mathbf{US}\} \quad (3)$$

where $\mathbf{RxSlot_1}$, $\mathbf{RxSeq_1}$ and $\mathbf{PID_1}$ contain vectors $\mathbf{RxSlot}$, $\mathbf{RxSeq}$ and $\mathbf{PID}$ respectively, and potentially some additional terms, as explained below.

This information, collected by neighboring nodes, is dual purpose:

a) Node N_i, through its broadcast signaling message, can attempt to reserve more receive slots, if $RS < RS_{\text{des}}$. This includes the case when $RS = 0$, i.e. when the node is disconnected. The node then, would append the required extra receive slots to vector $\mathbf{RxSlot}$ of equation (2). It will also append the potential parents from whom it can receive in these slots and the corresponding frame-sequence numbers of the packets it is missing, to vectors $\mathbf{PID}$ and $\mathbf{RxSeq}$ respectively (if $RS = 0$, then these vectors are all empty). Thus, vectors $\mathbf{RxSlot_1}$, $\mathbf{PID_1}$ and $\mathbf{RxSeq_1}$ of (3) are created.

b) Neighboring nodes will update their *neighborhood databases* by means of the data broadcasted by N_i. The entry, NB_i, of the neighborhood data-base contains the fields of $\mathcal{Q}_T$, the estimated path loss between the two nodes ($NB_i.G$) and the time stamp which represents the last time that node N_i sent an update message ($NB_i.t$). In addition, neighboring nodes will update their corresponding packet schedules ($\mathbf{TxSlot}$ and $\mathbf{TxSeq}$), and the field representing their unusable slots ($\mathbf{US}$), as will be explained below. To this end, nodes maintain a *children database* and an *unusable slots data-base*. An entry, CH_i, of the children data-base has the fields:

- $CH_i.ID$: the child's ID;
- $CH_i.\mathbf{RxSlot}$: the vector of *receive slots* of the child, i. e. slots in which the child receives from the specific node;

[1] Here and throughout this work, bold typeface denotes a vector quantity.

[2] Note that packets that are transmitted in random-access slots may not be received correctly if the SIR at the receiver node is below a certain threshold. These packets will be considered lost.

[3] Since signaling information is transmitted in round-robin fashion, it is interference free and is received by nodes up to a distance $d_{\max}$ away.

- $CH.\mathbf{t}$: each component of this vector, $CH.t[i]$ denotes the last time at which the node transmitted to the child in the corresponding receive slot $CH.RxSlot[i]$.

An entry US_i of the unusable slots data-base has the fields:

- $US_i.s$: the position of the unusable slot in the frame;
- $US_i.t$: the time at which this entry was last refreshed.

The node can render a slot usable again, if the slot is not refreshed in the US before a timeout period.

C. Connection procedure

In order to explain the connection procedure of the protocol, assume that node N_0 has a smaller number of reserved slots than what is desired, but the node is not disconnected from the multicast mesh, that is $RS_{\min} \leq RS < RS_{\text{des}}$ for N_0. First, N_0 will determine the frame-sequence numbers of the packets from the multicast session that it is not receiving in the reserved portion of the bandwidth.[4] Then, using its neighborhood data-base NB, the node will look for neighbors that are already transmitting these packets (for supporting other nodes) whose hop-count is less than its own by one. This will be done by inspecting the fields $NB_j.\mathbf{TxSeq}$ in the database, for all neighbors N_j such that $NB_j.HC = HC - 1$. For each such neighbor N_j, that transmits one of the missing packets, say packet with frame-sequence number $NB_j.TxSeq[k]$ in slot $NB_j.TxSlot[k]$, node N_0 will estimate the SIR for N_j's transmission in that slot. If N_0 senses a receive power P in that slot, then an estimate of the SIR, for N_j's transmission would be:

$$\text{SIR} = \frac{NB_j.G \cdot P_T}{P - NB_j.G \cdot P_T}. \tag{4}$$

If the estimated SIR is larger than the threshold γ, node N_0 can receive the corresponding missing packet from node N_j. Therefore, it will append $NB_j.ID$, $NB_j.TxSlot[k]$ and $NB_j.TxSeq[k]$ to $\mathbf{PID}$, $\mathbf{RxSlot}$ and $\mathbf{RxSeq}$, respectively, and include these fields in the signaling packet. This process continues until either node N_0 meets its desired reservation bandwidth, RS_{des}, or there is no other neighbor that is already relaying the missing packet, and from which N_0 could receive that packet.

If after this procedure, node N_0 is still missing multicast packets, it will identify neighboring nodes with hop-count one less than its own that can add transmit slots to their $\mathbf{TxSlot}$, in order to relay the missing packets. To this end, for each neighbor N_j, N_0 will examine the set of slots left after subtracting the slots in $NB_j.\mathbf{TxSlot}$ as well the slots in $NB_j.\mathbf{US}$. For each one of the remaining slots (if there are any), N_0 will estimate the SIR, if N_j were to relay a packet to N_0 in that slot:

$$\text{SIR} = \frac{NB_j.G \cdot P_T}{P}. \tag{5}$$

where P is the power sensed in the slot. If the $SIR > \gamma$, N_0 can receive packets from N_j in the specific slot and will proceed to make N_j its parent. It will add N_j's ID, the specific slot and the frame-sequence number of the missing packet to vectors $\mathbf{PID}$, $\mathbf{RxSlot}$ and $\mathbf{RxSeq}$ respectively.

If node N_0 is disconnected from the multicast mesh ($RS = 0$), it also follows the above procedure, but it first tries to use as parents neighbors with the minimum hop count. If N_0 cannot find $RS_{\min}$ slots, to receive packets from these neighbors, it will try to obtain the packets from neighbors with a hop count greater than the minimum by one, and so on. It is worth noting that, since all parents of a node have the same hop count, routing cycles are precluded.

As mentioned earlier, node N_0 will broadcast a maintenance packet in its signaling slot. This packet will contain the fields shown in equation (3). When node N_0's neighbors receive this broadcast packet, they will first examine the field which lists the parent ID's of N_0, $\mathbf{PID_1}$. Each neighbor node N_j will find elements $PID_1[k]$ that contain its own ID. For each such element, N_j will check whether N_0 is in its children data-base, and is already receiving packets with frame-sequence number $RxSeq_1[k]$ in slot $RxSlot_1[k]$ from N_j. If so, N_j will simply refresh the pertinent time entry in its children data-base, i.e. $CH.\mathbf{t}$. If N_0 is not in its children data-base, node N_j will check if it already relays packets with frame-sequence number $RxSeq_1[k]$ in slot $RxSlot_1[k]$ to some other children. If this is the case, then N_j will add node N_0 to its children data-base as a child receiving the above mentioned packets in the above mentioned slot. Finally, if N_j is not transmitting in slot $RxSlot_1[k]$, and this slot is not unusable (i.e. it is not in the $\mathbf{US}$ of N_j), N_j will add N_0 as its child as in the above case. It will also add slot $RxSlot_1[k]$ to its transmit slots vector $\mathbf{TxSlot}$ and $RxSeq_1[k]$ to the transmit sequence vector $\mathbf{TxSeq}$.

Next, each neighbor node N_j will consider the elements $PID_1[k]$ of $\mathbf{PID_1}$, that do not contain its own ID. This would imply that node N_0 will be receiving packets from different neighbors in the corresponding slots. For each such element, for which $ID \neq PID_1[k]$, N_j will add the slot (of the transmit frame), with number $RxSlot_1[k]$ to its unusable slot vector $\mathbf{US}$, but only if it is not already using this slot to transmit. By adding slot $RxSlot_1[k]$ to $\mathbf{US}$, node N_j will preclude itself from using this slot in the future to relay packets, and hence potentially interfere with N_0's reception in the same slot. Note that, N_j might be able to use slot $RxSlot_1[k]$ for transmission, if N_0's SIR in that slot was sufficiently high in spite of such a transmission. In that case, N_j would have to be able to estimate N_0's SIR and that would entail N_0 broadcasting SIR *and* interference information for each one of its receive slots. While this variation is possible, it is not considered here.

It is to be noted that, if the network were to be static, the protocol is designed so as to ensure that nodes that are already connected to the multicast mesh are not jammed by new members that try to connect. Consider for example, Figure 3. Node N_0 broadcasts its maintenance packet at time t. Suppose that node N_1's ID is contained in $\mathbf{PID_1}$ for slot s, either because N_0 is already connected to N_1 and receives from N_1 in slot s, or because it requests for the first time to receive packets from N_1 in slot s. At that time, neighbor N_2, upon receiving N_0's message, will update (or add) slot s in its unusable slots data-base. Assume also that Node N_2 has not yet broadcasted its maintenance packet, when node N_3 transmits a maintenance packet in

[4] Note that packets with frame-sequence numbers from 0 to $RS_{\text{des}} - 1$ will be scheduled to be received in the reserved portion of the bandwidth, while packets with larger frame-sequence numbers will be received in the random-access portion of the bandwidth.

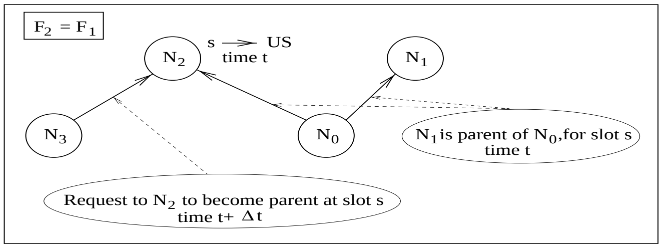

Fig. 3. The reception of node N_0 is protected from jamming since slot s is in the unusable slot vector of node N_2.

TABLE I
\# OF SLOTS PER SUPERFRAME AND \# OF RELAYS, VS. $d_{\max}$
($RS_{\text{DES}} = RS_{\min} = 1$).

$d_{\max}(m)$	Slots	Relays
250	8	19
300	8	14
350	6	9
400	6	9
500	4	6

which N_2 is identified as a parent which would transmit in the same slot s (this implies that N_2's signaling slot comes after that of N_3 in the round robin signaling period). Node N_2 will simply ignore this message since slot s is already in its unusable slot vector, and hence cannot be used for transmission. Thus, if node N_0 has successfully connected to node N_1, its connection is protected from being jammed. Node N_3 will learn of the updated unusable slots of node N_2 later in the same signaling period, and in the next signaling period may try to connect to N_2 by means of another slot, or may attempt to connect to another node. On the other hand, if N_0 was attempting to connect to node N_1 and this attempt failed for some reason (for example N_0 also had "stale" information with regard to slot s, and this slot is already in the unusable slot vector of N_1) slot s will not be updated in the unusable slots data-base US of N_2 for some time and will subsequently be rendered usable again.

Of course, under mobility, as nodes move into the vicinity of each other, excessive interference will be caused in reserved slots and nodes will lose connectivity with some of their parents. Then, following the procedures outlined earlier, they will try to re-affiliate to the multicast mesh.

V. PERFORMANCE OF WARM

Performance of WARM was evaluated using Parsec, a C based, discrete event parallel simulation language, developed at UCLA [11]. Fifty nodes were considered and were dispersed in an area of one square kilometer. In all cases, the minimum SIR required for a packet to be successfully received was set at $\gamma = 10$ dB and the path loss exponent η was set to four. We consider CBR traffic to begin with, and each node needs to reserve only one slot in the superframe, for receiving packets ($RS_{\text{des}} = RS_{\min} = 1$). The first simulation we performed assumed that all nodes were static. Our aim was to compute the number of slots, as well as the number of relays required in order to connect all the nodes to the multicast mesh. We computed this number for various values of the maximum transmission range $d_{\max}$. Results from one hundred different node positions were found and averaged, and are presented in Table I. The number of slots reported is per superframe. This table shows that, as expected (since for a given node the number of neighbors increases with $d_{\max}$), fewer relays are required for larger values of $d_{\max}$, in order to have complete connectivity, and hence fewer slots are sufficient to support the multicast session. As $d_{\max}$ decreases, the average degree of nodes decreases. Thus, more relays are needed to support the multicast session, and one expects that the number of slots required for complete connectivity increases. However, spatial re-use of TDMA slots increases with decreasing $d_{\max}$, since the co-channel interference also decreases in this case; for $d_{\max} = 250$ meters, 19 relays are needed and 8 slots are required, while for $d_{\max} = 500$ meters, 6 relays and 4 slots are enough.

Next, we evaluate the performance of WARM under mobility. We fix the values of $d_{\max}$ to be 250 meters. We also assume that there are ten slots per superframe assigned for the session, although from table I we see that only eight are needed on the average. The extra slots will improve performance in terms of received packets, as will be seen below. We assume that, out of the 50 nodes, 10 nodes are mobile and all mobile nodes move with the same speed. Nodes randomly select a direction (an angle is chosen uniformly between 0 and 360 degrees) and move along a straight line. They choose a new random direction after a random interval which is exponentially distributed with a mean value of one minute. Finally, nodes that reach the boundary of the roaming area simply bounce back into the area by choosing a new random direction. A signaling period is assumed to follow every 1000 data slots. Results were computed for 7 different combinations of mobile nodes (beginning from the same initial network) and an average of the combinations is presented in Figure 4 in the form of percentage of multicast data packets received (throughput) versus the speed of the mobiles. Two cases are considered, wherein, the source node is either static or mobile. Observe that, if the source node is static, the throughput remains very high (90%) even at high speeds. However, when the source node is mobile, the throughput suffers as speed increases.

Throughput improves with the number of slots available for reservation. Figure 5 presents throughput versus number of slots in the superframe assigned for the multicast session, for the case in which the source is mobile and the speed considered is 15 Km/h. As it can be seen from Table I and Figure 5, although only 8 slots are needed on the average for full connectivity of the static network, throughput under mobility would be poor when only 8 slots are assigned. However, by assigning 10 slots for the multicast session, we observe a 10% improvement in throughput.

Next, we consider the case of VBR traffic. Here, we assume that nodes can estimate accurately the mean traffic rate [7], [8][5],

[5] It must be noted that estimating the mean traffic rate of a VBR traffic source is not an easy problem. Also, during abrupt increases of the mean, packets will be queued and will be relayed mostly through random-access slots, until more bandwidth is reserved, thus increasing the packet dropping rate. We do not consider these problems in our simplified model.

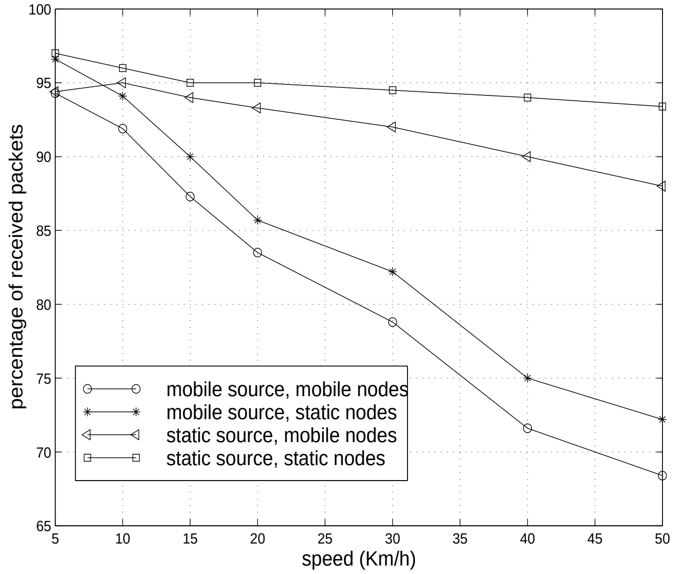

Fig. 4. Percentage of received packets versus speed, for 10 mobile nodes and the cases of the source being static and mobile.

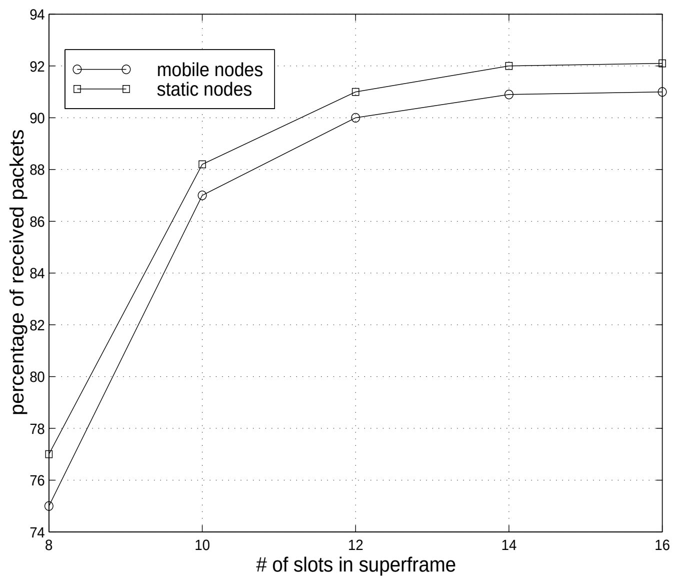

Fig. 5. Percentage of received packets versus # of slots in superframe, for mobile source and speed of 15 Km/h.

and focus on the problem of partitioning a total number of slots into reservable and random-access portions, such that a specific packet loss rate is achieved. Obviously, by assigning reserved slots to support the peak source rate, the packet loss rate will be zero (in a static network) but bandwidth utilization will be very poor. The question is, how much bandwidth above the mean do we have to assign to a multicast session in order to have an acceptable packet loss rate? In our simplified model, the packet arrival process at the source is assumed to be Poisson distributed, with mean rate of one packet per superframe (with the assumptions made above we can scale the mean to be one). Packets that are in excess of the reserved bandwidth are transmitted in the random-access portion of the transmit frame. There is no queuing and therefore delay jitter is bounded by the duration of one superframe (10 ms). We have set $RS_{\min} = RS_{\text{des}}$ and the total number of slots to be allocated for the multicast session to enable contention free scheduling is obtained by multiplying RS_{des} by 8 (which is the number required for $RS_{\text{des}} = 1$, as seen from Table I). Figure 6 presents the packet loss probability versus total available number of slots in a superframe, for different RS_{des}, for a static network of 50 nodes and with $d_{\max} = 250$. Statistics were gathered from nodes with the maximum hop-count, which are expected to have the worst packet loss rate. From Figure 6, note that the abscissa of the left-most point of each curve corresponds to the total number of reserved slots, which is a multiple of eight. The corresponding packet loss rate is the fraction of packets per frame that are in excess of RS_{des}, since there are no random-access slots in this case and these packets are dropped at the source. From Figure 6, it is obvious that assigning only one reserved slot ($RS_{\text{des}} = 1$) for each node, and receiving all packets in excess of the first one in the random-access portion, results in very severe packet loss rate. However, for $RS_{\text{des}} = 2$ (in which case, as can be seen from Figure 6, there are 16 reserved slots and about 10% of packets are transmitted using random-access), using 24 random-access slots, for a total of 40 slots in the superframe, the packet loss rate is slightly above 1%. Note here that $RS_{\text{des}} = 2$ corresponds to the mean source rate plus one standard deviation.

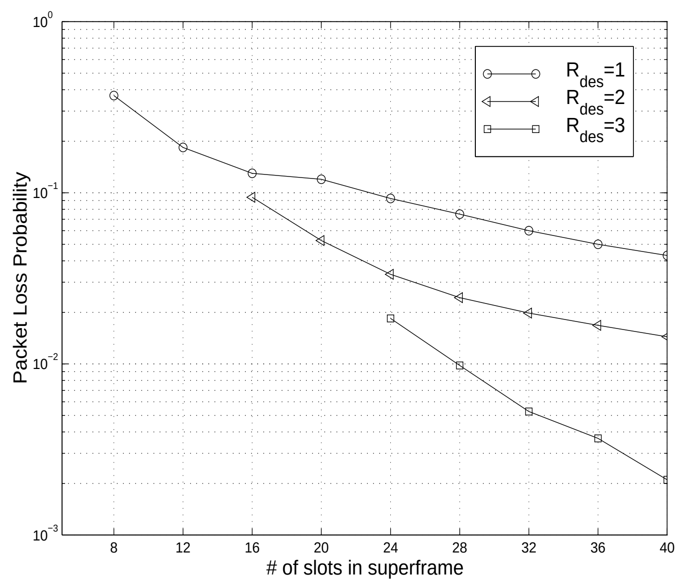

Fig. 6. Packet loss probability for a Poisson arrival rate with mean one packet per superframe vs. total number of slots in the superframe and for different number of reserved slots, RS_{des}.

VI. Conclusions-Future Work

In this paper we propose a novel protocol to stream real-time multicast data to nodes in a wireless ad-hoc network. The multicast structure is built such that collisions are avoided when data is being streamed. The protocol is distributed and the information is being exchanged only between one hop neighbors. The multicast structure which is built has self-healing features. Simulation results show that throughput above the 90%th percentile for pedestrian speeds ($\sim$ 10 Km/h) is achieved. The proportion of reserved and random-access bandwidth needed for VBR sources was also studied with a simplified model. Results quantify the trade-off between bandwidth reservation and packet loss rate for the multicast session.

A study of the exact signaling bandwidth needed for the exchange of maintenance packets is under way. Additionally, we are looking for ways to adapt the random-access strategies (like RTS/CTS or CSMA and its variants [10]) for the signaling channel, which would make our protocol completely scalable and more robust to mobility.

References

[1] C. E. Perkins and P. Bhagwat, “Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers”, in *Proc. ACM SIGCOMM’94*, London UK, pp. 234-44.

[2] J. Broch et al., “A performance comparison of multi-hop wireless ad hoc network routing protocols”, in *Proc MOBICOM’98*, Dallas TX, pp. 85-97.

[3] J. J. Garcia-Luna-Aceves and E. L. Madruga, “The core-assisted mesh protocol”, *IEEE J. Select. Areas Commun.*, vol. 17, pp. 1380-1394, Aug. 1999.

[4] A. Ephremides and T. V. Truong, “Scheduling Broadcasts in multihop radio networks”, *IEEE Trans. Commun.*, vol. 38, pp. 456-460, April 1990.

[5] S. Ramanathan and E. L. Lloyd, “Scheduling algorithms for multihop radio networks”, *IEEE/ACM Trans. on Networking*, vol. 1, pp. 166-177, 1993.

[6] I. Chlamtac and A. Farago, “Making transmission schedules immune to topology changes in multi-hop packet radio networks”, *IEEE/ACM Trans. on Networking*, vol. 2, pp. 23-29, Feb. 1994.

[7] S. Chong and J. Gosh, “Predictive dynamic bandwidth allocation for efficient transport of real-time VBR video over ATM”, *IEEE J. Select. Areas Commun.*, vol. 13, pp. 12-23, Jan. 1995.

[8] D. P. Connors, G. J. Pottie, B. Ryu and S. K. Dao, “A quality of service based medium access control protocol for real-time sources”, *Baltzer/ACM J. on Spec. Topics in Mobile Netw. and Appl. (MONET)*, to appear Fall 1999.

[9] G. J. Pottie, “System Design Choices in Personal Communications”, *IEEE Personal Commun.*, pp. 50-67, Oct. 95.

[10] D. Bertsekas and R. Gallager, *“Data Networks”*, 2nd ed., Prentice Hall 1992.

[11] `http://pcl.cs.ucla.edu/projects/parsec/`