Multicasting with quality of service guarantees in Wireless ad hoc networks

Dr. R. Madhanmohan

Associate Professor
Department of Computer Science and Engineering
Annamalai University, Annamalainagar, India.

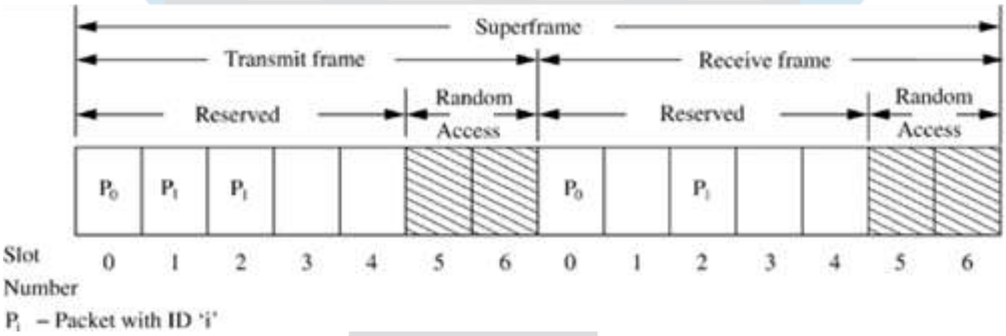
***Abstract*: Provisioning quality of service (QoS) implies providing guarantees such as deterministic end-to-end delay, availability of a fixed amount of bandwidth, buffers, and computational resources to the multicast session. Two multicast protocols, the wireless ad hoc real-time multicasting protocol and the multicast priority scheduling protocol that attempt to provide QoS guarantees are described in this paper.**

***Keywords*: WARM (Wireless ad hoc real-time protocol), Constant Bit rate, Variable Bit rate.**

1. Wireless Ad Hoc Real-Time Multicasting Protocol

Wireless ad hoc real-time multicasting (WARM) protocol [1] enables spatial bandwidth reuse along a multicast mesh. Bandwidth is guaranteed for real-time [constant bit rate (CBR)] traffic. The protocol uses periodic message exchanges, but the messaging is localized to the neighborhood of the receiving multicast member and hence the control overhead is low. It mainly deals with the transmission scheduling problem for a multicast session. A receiver node reserves time division multiple access (TDMA) time-slots and attaches itself to the multicast mesh through a neighbor node which is a multicast member. The protocol deals with two types of traffic: CBR traffic and variable bit rate (VBR) traffic. Time is slotted and is grouped into super frames. Each super frame consists of two frames, one for transmitting data and the other for receiving data. Each frame consists of two parts: a reserved part and a random-access part, as shown in Figure.1.. Both the frames are slotted. Packets transmitted by a node are numbered sequentially for that frame. Each node marks its receive (transmit) slots with the frame-sequence numbers of the packets to be received (transmitted) in those slots. A node may transmit the same packet in more than one slot, for supporting different children that may not be able to receive the packet in the same slot it can be seen that the same packet P1 is transmitted in slots 1 and 2). Packets in a frame that are in excess of the reserved number of slots are transmitted or received in the random-access portion of the frame.

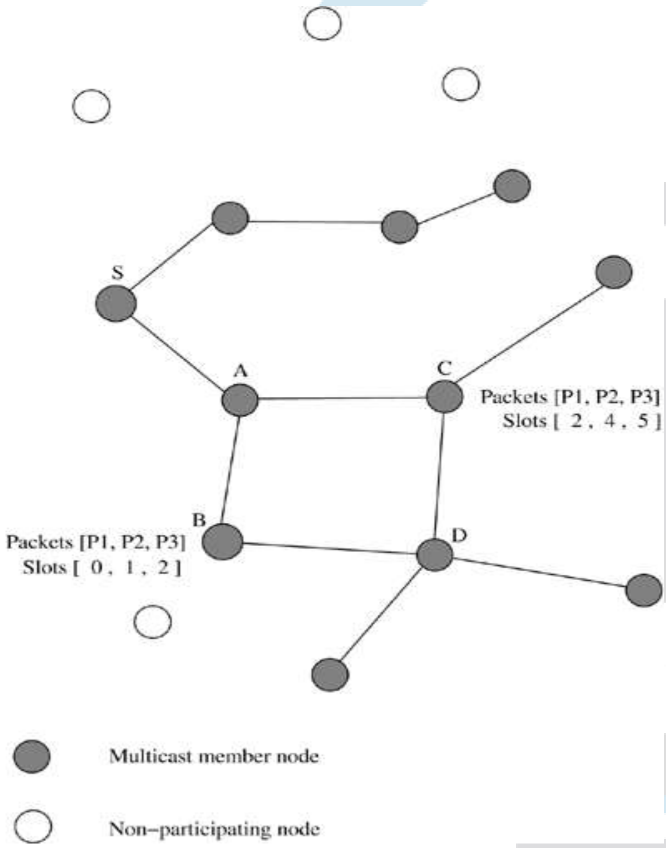
Figure 1. TDMA frame structure in WARM.



Each node maintains the following information for a particular multicast session: *ID* (the node's unique identifier), *HC* (the node's hop count, *i.e.*, the number of hops from the multicast source), *F* (a bit which indicates which of the two frames in the superframe is the transmit frame), *RS* (the number of receive slots currently reserved by the node), RS_{des} (the desired number of receive slots as determined by the node, based on the current traffic load), RS_{min} (the minimum acceptable number of reserved slots), *PID* (a vector representing the parents of the node), *RxSlot* (a vector which lists the receive slots of the node), *RxSeq* (a vector indicating the frame-sequence in which packets are received in the receive slots), *SIR* (a vector containing the signal-to-interference ratios of the node in its receive slots), G_P (a vector representing the path losses between the node and each of its parents), *TxSlot* (a vector which lists the transmit slots of the node), *TxSeq* (a vector indicating the frame-sequence in which packets are transmitted in the transmit slots), and *US* (a vector listing the slot numbers of the transmit slots unusable by the node). A node considers itself to be connected to the multicast mesh as long as $RS \geq RS_{min}$. If the node receives fewer than RS_{min} packets in its reserved receive slots, it becomes disconnected and sets its *RS* to zero. It then attempts to reconnect to the multicast session. Each node periodically transmits its status information consisting of *ID*, *HC*, *F*, *RS*, *RxSlot*, *RxSeq*,*PID*, *TxSlot*, *TxSeq*, and *US*. Transmission of this information is done on a separate signaling channel in a round-robin fashion. In case *RS* is less than RS_{des}, the node attempts to receive more receive slots through the signaling message. Here the node appends the required extra receive slots, the potential parents from which it can receive in these slots, and also the frame-sequence numbers of packets it is missing, to the signaling message. Each node maintains a neighborhood database containing information regarding all nodes from which it receives signaling information. Connection establishment when

$RS_{min} \le RS < RS_{des}$ is done as below. First, the node determines the frame-sequence numbers of the packets it is missing. It then looks into the neighborhood database to find nodes that are already transmitting those packets and whose hop-count is less than its own by one. For each such neighbor, it determines the *SIR* for the concerned slot. If the *SIR* is above a certain threshold value, then the node can receive data from the new parent node in that slot, and so it adds the new parent node's ID, slot number, and the frame-sequence to the appropriate vectors *PID*, *RxSlot*,and *RxSeq*, respectively. If, even after this process, the node misses packets, it identifies neighboring nodes with hop-length one less than its own and that can add transmit slots to their *TxSlots* in order to relay the missing packets. If the *SIR* it computes on these slots is acceptable, it makes these nodes its parents. Consider Figure 2, where node S is the source. Packets transmitted by node A are received without any collisions at node B and node C. Node B transmits packets P1, P2, and P3 in slots 0, 1, and 2, respectively. Node C transmits the same packets P1, P2, and P3 received from node A in slots 2, 4, and 5, respectively. Node B is the parent of node D. Packets P1 and P2 transmitted by node B are received by node D in slots 0 and 1 without any error. But in slot 2, both nodes B and C transmit packets simultaneously. Hence, the packets transmitted by them collide during slot 2 at node D. Node D therefore is unable to receive packet P3 from node B. It searches for other possible parent nodes transmitting packet P3. It finds that node C transmits packet P3 in slot5. If slot 5 at node B is free and if the *SIR* on slot 5 is acceptable, it makes node C its parent and receives packet P3 in slot 5.

Figure 2.. Example of WARM.



In case a node gets disconnected from the multicast mesh ($RS = 0$), it follows the same above procedure, but it tries to use neighbor nodes with the minimum hop-count as its parents. If RS_{min} slots cannot be found with these parents, it tries to obtain packets from neighbor nodes with hop-counts greater than the minimum by one, and so on.

2. Multicast Priority Scheduling Protocol

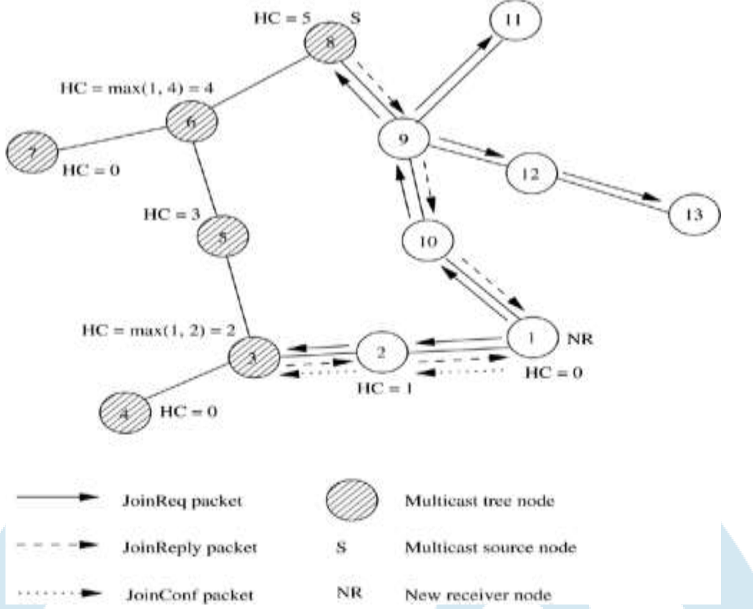
Multicast priority scheduling protocol (MPSP) [2] is a packet scheduling mechanism for multicast traffic in ad hoc wireless networks. The main objective of MPSP is to provide improved multicast packet delivery with bounded end-to-end delays. It is based on the DLPS protocol [3] which uses laxity-based priority scheduling and guarantees higher packet delivery for unicast traffic under bounded end-to-end delay conditions. Packet transmissions in a multicast session are broadcast in nature and are not protected by the RTS-CTS exchange. Therefore, the RTS, CTS, and ACK packets are not available for carrying piggy-backed priority information as in DLPS. MPSP modifies the DLPS protocol to suit the characteristics of multicast transmissions. MPSP can work with both tree-based as well as mesh-based multicast protocols. But since tree-based protocols are more efficient when compared to mesh-based protocol, the following discussion uses a tree-based multicast protocol for describing the operation of MPSP. Each node maintains a scheduling table (ST) which contains information about packets to be transmitted by the node and information about packets in the neighbor nodes (which is obtained by overhearing data packets carrying such information transmitted by neighbor nodes), sorted according to their *priority index* values. Priority index expresses the priority of a packet. The lower the priority index, the higher the packet's priority. The performance of MPSP was studied in [2] using the WBM protocol [15]. MPSP consists of four main components, namely, feedback mechanism, priority index calculation, scheduling table updates, and a back-off mechanism. The feedback mechanism is used to convey, to the multicast source, information regarding the percentage of packets delivered at the multicast receiver nodes. This task would be relatively simple in unicast routing as there exists only one receiver node, receiving packets from the source node through a single path. MPSP uses *soft acknowledgments* (described below) for this purpose. A leaf node *l*, on receiving a data packet from its parent node *p*, increments the count of packets received *countPkts* for the corresponding session, maintained by it. It sends back an acknowledgment (ACK) packet to its parent node *p* carrying the value of *countPkts*. If multiple leaf nodes are connected to a parent node, then that parent node receives an ACK from each of the leaf nodes. Now, parent *p* computes *avgCount*,the average of the received *countPkts* values.The value of *avgCount* is piggy-backed on each data packet transmitted by node *p*. This data packet is heard by node *pp*, which is the parent of node *p*.Node *pp* hears data packets transmitted by each of its child nodes. It computes the new *avgCount* from the *avgCount* values on the data packets heard. This new *avgCount* is piggy-backed by node *pp* on each DATA packet it sends. Thus, in this manner the value of *avgCount* moves up the multicast tree and reaches the source node after a small initial delay. This *avgCount* value is used in the calculation of priority index of packets, which is explained below.After transmitting each multicast packet, a node increments the count of multicast packets transmitted for that session *txCount*. The average count of packets received at the multicast receiver nodes *avgCount* is available at each multicast tree node. Each transmitted packet carries its end-to-end delay target *deadline*, that is, the time by which it should reach the destination. Using the above quantities, the priority index *PI* of a packet is calculated. It is given by

$$PI = \frac{PDR}{M} \times ULB$$

Here, $PDR = \frac{avgCount}{txCount}$ is the packet delivery ratio of the flow to which the packet belongs, $ULB = \frac{deadline - currentTime}{remHops}$ is the uniform laxity budget of the packet (*currentTime* denotes the current time according to the node's local clock), and *M* is a user-defined parameter representing the desired packet delivery ratio for the multicast session. *remHops* is the maximum number of hops remaining to be traversed by the multicast packet. It is obtained in the following manner. In WBM, when a new receiver node wants to join the multicast group, it initiates a *JoinReq*. A multicast tree node, on receiving this *JoinReq*, responds by sending back a *JoinReply* packet. The receiver node may receive multiple *JoinReply* packets. It chooses one of them and confirms its selection by transmitting a *JoinConf* to the corresponding multicast tree node.

This *JoinConf* packet is utilized by MPSP. Before transmitting the *JoinConf* packet, the new receiver node initializes a*hopCount* field on the *JoinConf* packet to zero. An intermediate node forwarding the *JoinConf*increments the *hopCount* by one. Hence, all nodes on the path to the node at which the new receiver node joins the multicast group (the node which initiated the *JoinReply* packet) know the remaining number of hops to be traversed by a multicast packet transmitted by it. The remaining number of hops*remHops* at node *n* is given by the maximum of *hopCount* to each receiver node through node *n*. A node piggy-backs this *remHops* value on each data packet it transmits. The node's parent, on hearing the transmitted data packet, extracts the piggy-backed value and computes its own *remHops* value, which it piggy-backs on the data packets it sends. Thus, the value of *remHops* moves up the multicast tree and finally reaches the multicast source node. Since each node in the multicast tree has a *remHops*value for the multicast session, the *ULB* of a packet queued at a node can now be computed. The above procedure is illustrated in Figure 3. Here, the new receiver node NR (node 1) initiates route discovery by transmitting a *JoinReq* packet. It reaches multicast tree nodes 3 and 8. Each of these two nodes responds by sending back a *JoinReply* packet. Node 1 chooses the *JoinReply* sent by node 3. It sends a *JoinConf* destined to node 3, with the *hopCount* value (*HC*) set to zero. When the packet reaches node 2, node 2 increments the *HC* by one and piggy-backs it on the next data packet it transmits. Node 3, on hearing this packet, retrieves the *HC* value one and computes the new *HC* value for node 1 as two. It is already connected to node 4, and so the *HC* from node 3 to node 4 is one. The new *HC* value for the multicast session at node 3 is computed as max(1, 2) = 2. This value is piggy- backed by node 3 on the next multicast packet it transmits, which would be heard by node 5. Nodes 5 and 6 compute the *HC* values in a similar fashion.When node 6 transmits a packet with the *HC* set to 4, node 8, the multicast source, would hear the packet and calculate the maximum number of hops remaining though node 6 as five.

Figure 3. Propagation of *hopCount* in MPSP.



Priority information regarding the highest priority ready packet to be transmitted next is piggy-backed on each data packet. A neighbor node, on hearing this packet, updates its scheduling table with the piggy-backed priority information. Hence, each node would know the priority of its own packet compared to the priorities of packets queued at its neighbor nodes. The back-off mechanism used in MPSP is the same as that used in DLPS. The objective of the back-off mechanism used in DLPS is to reflect the priority of the node's highest priority packet on the back-off period to be taken by the node. If *r* is the rank (the rank of an entry is the position of that entry in the scheduling table of the node), in ST of the node, of the current packet to be sent, *n* is the number of retransmission attempts made for the packet, and *nmax* is the maximum number of retransmission attempts permitted, then the back-off interval is given by

$$\text{back}-\text{off} = \begin{cases} Uniform[0,(2^n \times CW_{min})-1] \\ \quad \text{if } r=1 \text{ and } n \leq nmax \\ \\ \frac{PDR}{M} \times CW_{min} + \ Uniform[0, CW_{min}-1] \\ \quad \text{if } r>1 \text{ and } n=0 \\ \\ ULB \times CW_{min} + \ Uniform[0,(2^n \times CW_{min})-1] \\ \quad \text{otherwise} \end{cases}$$

where *CWmim* is the minimum size of the contention window, and *M* is the desired packet delivery ratio. If the packet has the highest rank in the broadcast region of the node, then it has the lowest back-off period according to Equation and faces very less contention. Else, if it is the first time the packet is being transmitted, the back-off distribution follows the second scheme as in Equation , where the back-off is more than that for the first case. Here the current *PDR* of the flow affects the back-off period. If *PDR* is very low, then the first term would be low, and if it is high, then the first term would be high and the node would have to wait for a longer time. Finally, if the packet does not fit into these two categories, then the back-off value is as per the third scheme as in Equation and is the longest of the three cases. The higher the value of *ULB*, the longer the back-off period.The above-described mechanisms aid MPSP in providing better multicast packet delivery compared to other regular multicast protocols, under bounded end-to-end delay condition.

3.Conclusion

The challenges faced by multicast routing protocols for ad hoc wireless networks are much more complex than those faced by their wired network counterparts. In this chapter, the problem of multicast routing in ad hoc wireless networks was studied. After identifying the main issues involved in the design of a multicast routing protocol, a classification of the existing multicasting protocols was given. Several of these multicast routing protocols were described in detail with suitable examples. The advantages and drawbacks involved in each protocol were also listed. Some energy-conserving multicasting routing protocols were presented, as most of the nodes in ad hoc wireless networks are battery-operated. Reliable multicasting has become indispensable for the successful deployment of ad hoc wireless networks as these networks support important applications such as military battlefields and emergency operations. Real-time multicasting that supports bounded delay delivery for streaming data is also a potential avenue for research. Security is another necessary requirement, which is still lacking in ad hoc multicast routing protocols, as multicast traffic of important and high security (*e.g.*, military) applications may pass through unprotected network components (routers/links). Thus, multicasting in ad hoc networks is a significant problem that merits further exploration.

References

[1] G. D. Kondylis, S. V. Krishnamurthy, S. K. Dao, and Gregory J. Pottie, "Multicasting Sustained CBR and VBR Traffic in Wireless Ad Hoc Networks," *Proceedings of IEEE ICC 2000*, pp. 543-549, June 2000.

[2] I. Karthigeyan, B. S. Manoj, and C. Siva Ram Murthy, "Multicast Priority Scheduling Protocol for Ad Hoc Wireless Networks," *Technical Report*, Department of Computer Science and Engineering, Indian Institute of Technology, Madras, India, January 2004.

[3] I. Karthigeyan, B. S. Manoj, and C. Siva Ram Murthy, "A Distributed Laxity-Based Priority Scheduling Scheme for Time-Sensitive Traffic in Mobile Ad Hoc Networks," to appear in *Ad Hoc Networks Journal*.

[4] L. Briesemeister and G. Hommel, "Role-Based Multicast in Highly Mobile But Sparsely Connected Ad Hoc Networks," *Proceedings of ACM MOBIHOC 2000*, pp. 45-50, August 2000.

[5] H. Zhou and S. Singh, "Content-Based Multicast (CBM) in Ad Hoc Networks," *Proceedings of ACM MOBIHOC 2000*, pp. 51-60, August 2000.

[6] Y. B. Ko and N. H. Vaidya, "Geocasting in Mobile Ad Hoc Networks:
Location-Based Multicast Algorithms," *Proceedings of IEEE WMCSA 1999*, pp. 101-110, February 1999.

[7] C. C. Chiang, H. K. Wu, W. Liu, and M. Gerla, "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel," *Proceedings of IEEE SICON 1997*, pp. 197-211, April 1997.

[8] R. Dube, C. D. Rais, K. Y. Wang, and S. K. Tripathi, "Signal Stability- Based Adaptive Routing for Ad Hoc Mobile Networks," *IEEE Personal Communications Magazine*, vol. 4, no. 1, pp. 36-45, February 1997.

[9] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," *Mobile Computing, Kluwer Academic Publishers*, vol. 353, pp. 153-181, 1996.

[10] S. Murthy and J. J. Garcia-Luna-Aceves, "An Efficient Routing Protocol for Wireless Networks,"*ACM Mobile Networks and Applications Journal:*
Special Issue on Routing in Mobile Communication Networks, vol. 1, no. 2, pp. 183-197, October 1996.