ISSN 2348 – 9928
IJAICT Volume 3, Issue 7, July 2016 Doi:01.0401/ijaict.2014.06.10 Published on 05 (7) 2016

ENERGY EFFICIENT ALGORITHM FOR MULTICAST ROUTING TECHNIQUES THROUGH THE MEDIUM ACCESS CONTROL (MAC) LAYER IN MANET

Mr. D. Balaji
Assistant Professor,
Electronics and Communication Engineering,
Mahendra College of Engineering,
Salem, Tamilnadu, India

Mr. B. Vinod
Assistant Professor,
Electronics and Communication Engineering,
VMKV Engineering College,
Salem, Tamilnadu, India

Abstract— The basic idea behind the concept here is to address the congestion control multicast routing problem in wireless ad hoc networks through the medium access control (MAC) layer. This work mainly focuses on cluster-based routing protocol (CBRP) and its comparative analysis with two other on demand routing protocols Adhoc On Demand Distance Vector (AODV) and Dynamic Source Routing (DSR) which do not use cluster based mechanism for routing. The simulation results obtained in this scheme shows a good performance in the aspects such as fairness with TCP, robustness against misbehaving receivers, and traffic stability.

Keywords— Multicast Routing, Medium Access Control, MANET and Congestion Control.

I. INTRODUCTION

Existing multicast congestion control schemes generally fall into two categories: single-rate and multi-rate. Multi-rate schemes usually offer much more freedom to receivers in choosing appropriate receiving rate than singlerate schemes. Because the links of a multicast tree are usually heterogeneous, receivers in a multicast session may have diverse amounts of available bandwidth. Multirate schemes have a great advantage over singe-rate schemes in catering to every receiver in a multicast session. Proposed scheme presents a new multi-rate multicast congestion control scheme suitable for Mobile Ad-hoc Networks (MANETs). For transport protocols not specifically designed for MANETs, the main sources of problems in MANETs are high link error rates, limited bandwidth, link access delays, and hand-offs. Almost all existing multicast congestion control schemes will suffer from the same problems as TCP suffers in MANETs (e.g., unnecessarily reducing the transmission rate in response to link errors).

This is because they use losses as the indication of congestion but cannot distinguish between link-error losses and congestion losses. Another specific problem for multi-rate schemes is the link access delay in MANETs caused by access competition. Because of the inherent design of the IGMP protocol, the layer-drop latency is already a significant problem in wireline networks for multirate schemes. The link access delay in MANETs caused by competition will exacerbate the layer-drop latency problem, because pruning information can reach a upstream router only after the upstream link has been successfully accessed, and in congested situations, there is a significant delay before the upstream link becomes available. Although some schemes have made a significant progress in combating this problem, they usually introduce considerable control traffic overhead, which is a serious disadvantage inMANETs e.g., valuable bandwidth and power are wasted.) [1][2].

Besides the disadvantages specific to MANETs, most existing schemes still have problems in sharing bandwidth fairly with TCP and dealing with misbehaving receivers.

To deal with the above disadvantages of existing schemes, instead of depending on individual receivers to detect congestion and adjust their receiving rates, the scheme proposed here adjusts multicast traffic rate right at each bottleneck of a multicast tree. Specifically, when congestion occurs or is about to occur at a branch, some layers of the multicast sessions traversing the branch are “blocked” from entering the branch; when the branch is lightly utilized, some blocked layers are “released” to traverse the branch.

II. RELATED WORKS

Routing protocols are classified as either table-driven or on-demand. The table driven protocols, also referred to as proactive protocols, maintain current information on all reachable destinations in anticipation of the use of the information. The nodes are required to maintain one or more tables of routing information which are propagated throughout the network as changes in the topology occur. The protocols differ in the necessary tables related to routing and how the tables are exchanged. The on-demand methods, also known as reactive protocols, wait to determine the route packets will travel at the time the communication begins. When a route is needed, the source node initiates a route discovery process to the destination. Once established the route must be maintained until it is no longer needed or the destination node becomes inaccessible. Proactive and reactive protocols each have advantages and disadvantages. In a proactive protocol the information to determine the routes is immediately available so no additional time is needed to discover the hops in a route, thus the delay of the first packet does not include route discovery time. This is a significant advantage when many routes are needed within a short period of time. A disadvantage of a proactive method is that it requires periodic updating of the routing tables, so if only a few routes need to be determined then the overhead of table exchanges and maintenance may be substantial. The advantages to an on-demand protocol are related to the disadvantages seen in the proactive protocols. If only a few routes need to be determined, using an on-demand method would incur less overhead to discover the hops than the overhead associated with the proactive protocol's exchange of topology information.

The disadvantage of the reactive method is that it necessitates a longer delay in getting the packets to the destination since it must first discover the route. In wireless environments, losses are time-variant and caused by a variety of reasons like link quality, fading, mobility, transmission errors, congestion and hidden terminals. The underlying MAC protocols also have a big impact on performance. In particular, the use of contention-based MAC protocols combined with hidden terminal problems make MANETs much more sensitive to load and congestion than wired networks or even wireless cellular networks. In such an environment, we argue that multicast reliability cannot be achieved solely by retransmission of lost packets as is typically done in wired networks with protocols such as Scalable Reliable Multicast (SRM).

In order to achieve reliable multicast delivery in MANETs, we must consider jointly two components: reliability and congestion control. This paper differs from previous work in the following ways. First, we focus on the case of congestion control. Second, we focus on allocating achievable fair shares of bandwidth to individual flows instead of scheduling and queuing schemes.

III. CONGESTION CONTROL SCHEME

The proposed scheme operates in the following way. When multicast sessions traverse a link, the scheme agent starts to observe the output queue of the link and the traffic passing the link. When the number of packets in the queue, N_{QuPkt}, exceeds a threshold, QuThresh2, some layers of multicast sessions are blocked from entering the link. However, when N_{QuPk}t is below another threshold, QuThresh1, for a period of time, some blocked layers are released to traverse the link. In other cases, there is usually no layer adjustment. In this way, congestion can be alleviated while free bandwidth can also be claimed. This is only a profile of the scheme. Some important details are missing. For example:

- ❖ How is it ensured that the bandwidth of a bottleneck is shared fairly between TCP sessions and multicast sessions?
- ❖ How is the layer priority information communicated if the layers of a multicast session have different priorities?

The remainder of the present scheme is detail in the rest of this section.

3.1 Scheme Basics

The proposed scheme retrieves some information about the competing sessions at a bottleneck to assist its operation. Specifically, the number of TCP sessions (N_{TcpSes}), the number of multicast sessions (N_{MctSes}), the number of layers of each multicast session (Ni LiveLayer; 0 < i _ N_{MctSes}), the average per-flow rate of TCP sessions(R_{TcpAvg}) and the average per-flow rate of multicast sessions (R_{MctAvg}) are the information retrieved. In general, all the information can be obtained by analyzing the addresses of the passing packets. In some applications such as streaming media, a lower layer usually has higher priority than a higher layer.

The proposed scheme embeds the layer priority information into the addresses used by the layers of a multicast session. Specifically, in session i, the address of the j^{th} layer is lower than the address of the k^{th} layer if j is less than k ($AL_{ij} < AL_{ik}$ if j < k). Meanwhile, at a bottleneck the proposed scheme distinguishes the priorities of the layers of the same multicast session according to their addresses. Specifically, a layer with a lower address has higher priority than a layer with a higher address ($_{PLij} > PL_{ik}$ if $A_{ij} < A_{ik}$).Instead of using layer-add and layer-drop at receivers as in most existing schemes, the proposed scheme uses layer-block and layer-release at bottlenecks to solve congestion and to claim bandwidth, respectively. Layer-block is the modification of the multicast routing table to stop a layer from entering a congested link; layer-release is the modification of the routing table to allow a blocked layer to traverse a link.

When layer block is necessary, the multicast session with the maximum number of layers is selected to block a layer. Within this session, the layer with the lowest priority among the unblocked layers is blocked. However, when layer-release is required, the multicast session with the minimum number of layers is selected to release a layer. Within this session, the layer with the highest priority among the blocked layers is released. In addition, receivers also play a small role in layer adjustment: each of them maintains a single empty layer. An empty layer of a receiver is a layer that is blocked somewhere in the network and has no data flowing into the receiver.

3.2 The Adjustment of the Number of Multicast Layers

Here the procedures for adjusting the total number of multicast layers (N_{layer}) traversing a bottleneck is presented. The proposed scheme blocks or releases multicast layers at a bottleneck according to the state of the output queue of the link. The queue is classified into three phases: phase 1, phase 2, and phase 3. The phase of a queue is decided by the number of packets in the queue, N_{QuPkt}, and two specified thresholds, QuThresh1 and QuThresh2 (QuThresh1 < QuThresh2)[3][4].

If NQuPkt _ QuThresh1, then the queue is in phase 1; if QuThresh1 < NQuPkt _ QuThresh2, then the queue is in phase 2; if NQuPkt > QuThresh2, the queue is in phase 3. The layer adjustment rules are as follows. When the queue is in phase 1, it is checked if the queue has been in phase 1 for a period of time greater than $T_{Observe}$. If it has, a multicast layer is released. Otherwise, nothing is done. Phase 1 is designed to claim free bandwidth spared by TCP sessions.

When the queue is in phase 2, the average per-flow rate of TCP sessions (RTcpAvg) and the average per-flow rate of multicast sessions (RMctAvg) are checked. When RMctAvg < RTcpAvg, a multicast layers is released. Otherwise, no action is taken. When the queue is in phase 3, RTcpAvg and RMctAvg are also checked. If RMctAvg > RTcpAvg, a multicast layer is blocked. Otherwise, no action is taken. The purpose of phase 3 is to detect congestion.

IV. THE PROPOSED PLAN

This section analyzes the proposed scheme for fairness and link utilization, effectiveness in MANETs, and cost. The proposed scheme imposes no direct control over any unicast flows and assumes that each unicast flow is controlled by TCP or a similar protocol without the assistance of active queue management. A multicast source encodes its signal into multiple layers and then sends each layer to a separate multicast group. After a source chooses its layer size, it does not change the size for the rest of the multicast session. The intended receivers of the multicast source try to subscribe to all these groups. Packets for all or some of these groups then flow into each individual receiver [5].

4.1 Link Utilization

All the multicast packets traversing a link and originating from the same multicast source are called a “multicast flow” on the link in this paper. When multicast flows traverse the wireless link of a node, the node observes the output queue of its link at regular intervals $I_{CheckQu}$ (congestion events are always immediately reported irrespective of the observation intervals though). When the number of packets in the queue, N_{QuPkt}, exceeds a threshold, QuThresh2, some layers of multicast flows will be blocked from entering the link. However, when N_{QuPkt} is below another threshold, QuThresh1, for a specified amount of time, some blocked layers of multicast flows will be released and allowed to traverse the link. In other cases, there are usually no layer adjustments over multicast flows [6][7].

It is mandatory to know what happens when the data in a layer is blocked before introducing the scheme details. If the multicast application, such as video multicasting, can tolerate losses, the data in a blocked layer is usually not recovered and receivers thus obtain information at a lower resolution, such as lower quality of received video. On the other hand, if the multicast application requires total reliability in data delivery, the source needs to use a technique such as the digital fountain technique. In such a case, a receiver needs to receive enough packets before it can decode and obtain all the data from the source. A blocked layer, therefore, introduces latency in data delivery in this case [8][9].

Finally, we need to elucidate that the proposed scheme is mainly designed to effectively relieve congestion at bottlenecks with multicast traffic. In addition, it is designed to maintain general fairness in bandwidth sharing among the competing flows at a bottleneck. In particular, when a given bottleneck limits the rates of all flows that pass through it, all flows receive similar bandwidth shares at the bottleneck. However, in scenarios where the rates of some flows may be restricted by other bottlenecks, these flows may receive a lower share of the bandwidth at the given bottleneck. Finally, since the proposed scheme is not centralized, we do not expect it to meet the requirements of fairness criteria other than the one we consider [10][11].

4.2 Effectiveness in MANETs

The effectiveness of the proposed scheme in MANETs stems from several factors. Instead of waiting for receivers to request pruning and grafting as in existing schemes, the proposed scheme adjusts multicast traffic rate right at each bottleneck of a multicast tree. Therefore, it is not affected in its rate adjustment by the link access delay caused by link competition in MANETs, which can adversely affect existing schemes significantly in their rate adjustment (i.e., further increased layer-drop latency). Link errors also cannot decrease the performance of the proposed scheme (i.e., cannot cause it to wrongly block layers), since it uses the queue state at a bottleneck instead of the loss information at receivers as the metric to adjust the multicast traffic rate at the bottleneck. In addition, the proposed scheme only has very limited control traffic overhead. In existing schemes, either poor coordination among receivers or the design of the scheme itself results in frequent branching and pruning, which may produce significant control traffic overhead [12].

Although the receivers of a multicast session need to adjust their empty layers with the proposed scheme, the adjustments are few because the proposed scheme does not have frequent layer adjustment at bottlenecks. Furthermore, all receivers under a bottleneck are well coordinated by the multicast traffic that is effectively controlled at the bottleneck. Without penalty from link errors or link access delay and without excessive control traffic overhead, the proposed scheme works effectively and efficiently in MANETs [13][14].

Another feature of the proposed scheme is that misbehaving receivers can neither benefit them selves nor hurt other receivers, since with the scheme; the number of active layers a receiver can receive is solely controlled at the bottleneck along the path from the source to the receiver. In fact, if a receiver intentionally or accidentally subscribes to too many layers, the number of layers that have data flowing into the receiver will not change, because the bottleneck will block the excessive layers automatically. Other receivers under the same bottleneck are not affected either. The only consequence is that some limited bandwidth above the bottleneck is possibly wasted (see the next subsection for more details) [15].

4.3 Cost

The main cost of the proposed scheme arises from retrieving information about competing sessions. All the information can be obtained by analyzing the addresses of passing packets. Since addresses have to be analyzed anyway in packet forwarding, the extra cost introduced by the proposed scheme is arguably not significant. In fact, the forwarding process only needs to put the retrieved addresses of packets into a buffer and another separate process can analyze them to obtain the information needed by the proposed scheme [16].

Another kind of possible cost of the proposed scheme may come from the empty layer maintained by each receiver. When a receiver maintains an empty layer, some bandwidth above the bottleneck along the path to the receiver may be wasted if no other receiver above the bottleneck needs that layer. However, the maximum amount of bandwidth that may be wasted by session m at link i is limited to the difference between the average bandwidth share for each session at link i and the bandwidth actually used by session m at link i (assuming no free bandwidth at link i). Last, all the operations of the proposed scheme; in general, do not affect the queuing, scheduling, or forwarding policy of existing networks, so the proposed scheme will not affect existing network structure and applications if it is deployed.

V. PERFORMANCE ANALYSIS AND SIMULATION RESULTS

The wireless ad hoc network configured in this simulation, the propagation model is two-ray ground; the MAC protocol is IEEE 802.11; the ad hoc routing protocol is MOADV, and the link queue size is 50 packets. The wireless ad hoc network is in an area of 1500 × 1500 m and the transmission distance of a node is 250 m. Two nodes at a distance of 100 m form a shared wireless bottleneck. The senders of the competing flows are randomly placed within 100 m around one node of the shared bottleneck and the receivers of the competing flows are randomly placed within 100 m around the other node of the shared bottleneck. It is ensured though that all the senders must be more than 100 m away from all the receivers so that all competing flows traverse the shared bottleneck, which has a radio bandwidth of 2 Mbps.

5.1 Throughput

The two source routing protocols demonstrate high quality in delivering packets—more than 95% in the case of 50 nodes. AODV has difficulty when the nodes are moving fast (corresponding to smaller pause time), with a throughput less than 80%. Source routing reveals more information in one route discovery than AODV. Therefore, within the same time, more routes are discovered and so more packets can be delivered. AODV catches up when the mobility of the nodes gets lower. This is because routes become more stable, and so eventually everybody can find all the routes it ever needs. Between DSR and CBRP, CBRP has a better throughput for a larger network size. This better scalability comes from its largely reduced flooding for route discovery.

5.2 Delay

Among the three protocols, AODV has the shortest end-to-end delay of no more than 0.05 seconds. Besides the actual delivery of data packets, the delay time is also affected by route discovery, which is the first step to begin a communication session. The source routing protocols have a longer delay because their route discovery takes more time as every intermediate node tries to extract information before forwarding the reply. The same thing happens when a data packet is forwarded hop by hop. Hence, while source routing makes route discovery more profitable, it slows down the transmission of packets. CBRP is even more time-consuming because of its two-phase route discovery. The task of maintaining cluster structure also takes a piece of each host's CPU time.

5.3 Overhead

Without any periodic hello messages, DSR outperforms the other two protocols in terms of overhead. In most cases, both the packet overhead and the byte overhead of DSR are less than half of the overhead of CBRP and less than a quarter of AODV's overhead. AODV has the largest routing load (in the 50-node cases, as many as 6.5 routing packets per data packet and 2 routing bytes per data byte) because the number of its route discoveries is the most, and the discovery is network-wide flooding. CBRP has a much smaller flooding range; the number of its route requests and replies is constantly half that of DSR. But its hello messages outweigh this gain. And since the size of CBRP hello messages can be large, its byte overhead is still more than DSR's (in the 50-node cases, more than twice as much as DSR's). When there are more connections, more routing is needed, and so the proportion of hello messages in the total overhead becomes smaller. As the result, CBRP and AODV get closer to DSR.

The following fig.1 shows the transmission of messages between nodes in multicasting by using proactive protocol - DSR.

Fig 1 : Transmission of messages between nodes-DSR

The following fig. 2 shows the transmission of messages between nodes in multicasting by using pro active protocol-CBRP.

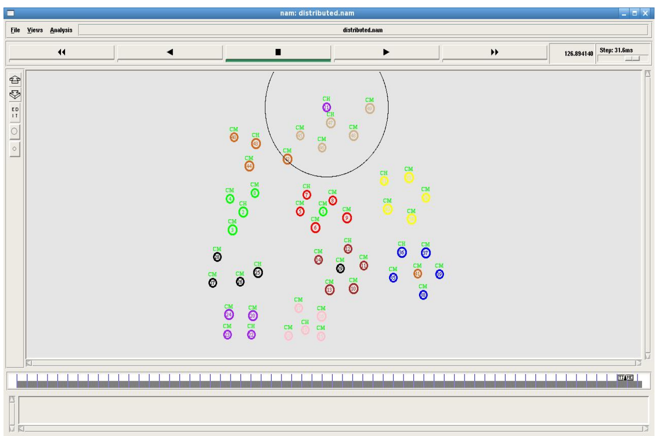

Fig 2 : Transmission of messages between nodes-CBRP

The fig.3 shows the relationship between the No of Clusters and transmission range in which the time in the X - axis and packets delivered in the Y - axis within the simulation time 27sec.

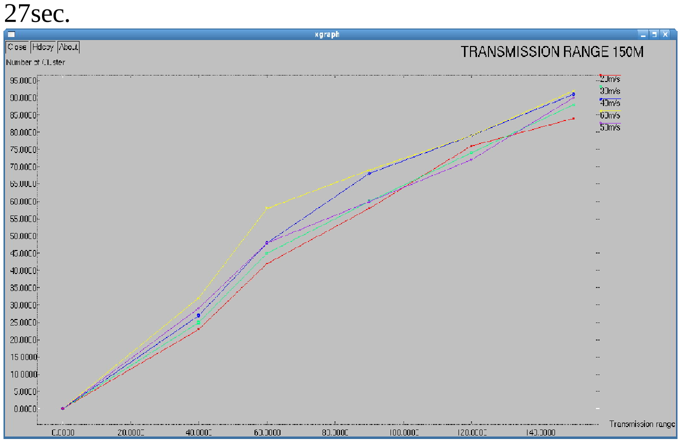

Fig 3 : Throughput Performance Evaluation

VI. CONCLUSION

The main focus of the work here is the routing problem in ad hoc networks. Routing in wireless mobile ad-hoc networks should be time efficient and resource saving. One approach to reduce traffic during the routing process is, to divide the network into clusters. The NS2 simulation results are used to compare three on-demand ad - hoc routing protocols (DSR, AODV, and CBRP), using a variety of workloads such as mobility, load, and size of the ad hoc networks. From the results, it is concluded that the two source routing–based protocols, DSR and CBRP, have very high throughput while the distance-vector-based protocol, AODV, exhibits a very short end-to-end delay of data packets. Furthermore, despite its improvement in reducing route request packets, CBRP has a higher routing overhead than DSR because of its periodic hello messages. DSR has much smaller routing overhead than AODV and CBRP, and AODV have the largest overhead among the three protocols.

References

[1]. S.McCanne, V.Jacobson, and M.Vetterli, “Receiver-driven layeredmulticast,” in Proc.of ACM SIGCOMM., Aug 1996, pp. 117–130.

[2]. L.Vicisano, L. Rizzo, and J. Crowcrof, “Tcp-like congestion control for layered multicast data transfer,” in Proc of IEEE INFOCOM., San Franciso, March 1998, pp. 996–1003 Vol. 3.

[3]. J. Byers, M. Frumin, G. Horn, M. Luby, M. Mitzenmacher, A. Roetter, and W. Shaver, “Flid-dl: Congestion control for layered multicast,” in Proceedings of NGC 2000, November 2000, pp. 71–81.

[4]. M.Luby and V. Goyal, “Wave and equation based rate control using multicast round trip time,” in Proc of ACM SIGCOMM., Pittsburgh, Pennsylvania, USA., Aug. 2002, pp. 191–204.

[5]. I. Rhee, N. Balaguru, and G. Rouskas, “Mtcp: scalable tcp-like congestion control for reliable multicast,” in Proc of IEEE INFOCOM., March. 1999.

[6]. L. RIZZO, “pgmcc: A tcp-friendly single-rate multicast congestion control scheme,” in Proc of ACM SIGCOMM., Stockholm, Sweden, Aug 2000, pp. 17–28.

[7]. J. Widmer and M. Handley, “Extending equation-based congestion control to multicast applications,” in Proc of ACM SIGCOMM., San Diego, CA, Aug. 2001.

[8]. R. Gopalakrishnan, J. Griffioen, G. Hjalmtysson, and C. Sreenan, “Stability and fairness issues in layered multicast,” in Proceedings of the NOSSDAV, June 1999, pp. 31–44.

[9]. A. Legout and E. W. Biersack, “Pathological behaviors for rlm and rlc,” in Proceedings of the NOSSDAV, North Carolina, USA, June 2000, pp. 164–172.

[10]. O.Papaemmanouil and U. Cetintemel, “Semcast: Semantic Multicast for Content-Based Data Dissemination,” Proc. Seventh Int’l Workshop Web and Databases (WebDB ’04), June 2004.

[11]. R.Jain, D. Chiu, and W. Hawe, “A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer Systems,” DEC Research Report TR-301, Sept. 1984.

Corresponding Author: Ms. V. N. Ilakkiya, Kalaignar Karunanidhi Institute of Technology, Coimbatore, Tamilnadu, India.

[12]. N. Sastry and S. Lam, "CYRF: A Theory of Window Based Unicast Congestion Control," IEEE/ACM Trans. Networking, vol. 13, no. 2, pp. 330-342, Apr. 2005.

[13]. The Network Simulator—ns-2, http://www.isi.edu/nsnam/ns/, 2008.

[14]. S. Sarkar and L. Tassiulas, "Back Pressure Based Multicast Scheduling for Fair Bandwidth Allocation," Proc. IEEE INFOCOM, 2001.

[15]. S. Bajaj, L. Breslau, and S. Hanker, "Uniform versus Priority Dropping for Layered Video," Proc. ACM SIGCOMM '98, Sept. 1998.

[16]. R. Gopalakrishnan, J. Griffioen, G. Hjalmtysson, C. Sreenan, and S. Wen, "A Simple Loss Differentiation Approach to Layered Multicast," Proc. IEEE INFOCOM '00, Mar. 2000.