논문 2006-01-02

DCAR: Dynamic Congestion Aware Routing Protocol in Mobile Ad Hoc Networks

Young-Duk Kim*, Sang-Heon Lee and Dong-Ha Lee

Abstract : In mobile ad hoc networks, most of on demand routing protocols such as DSR and AODV do not deal with traffic load during the route discovery procedure. To achieve load balancing in networks, many protocols have been proposed. However, existing load balancing schemes do not consider the remaining available buffer size of the interface queue, which still results in buffer overflows by congestion in a certain node which has the least available buffer size in the route. To solve this problem, we propose a load balancing protocol called Dynamic Congestion Aware Routing Protocol (DCAR) which monitors the remaining buffer length of all nodes in routes and excludes a certain congested node during the route discovery procedure. We also propose two buffer threshold values to select an optimal route selection metric between the traffic load and the minimum hop count. Through simulation study, we compare DCAR with other on demand routing protocols and show that the proposed protocol is more efficient when a network is heavily loaded.

Keywords : Mobile Ad Hoc Networks, Routing Protocol, Load Balancing

I. Introduction

A mobile ad hoc network (MANET) is a self-configuring network of mobile hosts connected by wireless links without fixed infrastructure such as base station. In MANETs hosts are free to move randomly, and thus network topologies may change rapidly and unpredictably. Devising an efficient routing protocols for MANETs has been a challenging issue and DSDV (Destination Sequence Distance Vector) [1], DSR (Dynamic Source Routing) [2], AODV (Ad-hoc On-demand Distance Vector) [3] are such protocols to tackle the issue.

Recently, the requirement for real time and multimedia data traffic continues growing. In this situation, the occurrence of congestion is inevitable in MANETs due to limited bandwidth. Furthermore, by the route cache mechanism in the existing protocols, the route reply from intermediate node during the route discovery procedure leads to traffic concentration on a certain node. When a node is congested, several problems such as packet loss by buffer overflows, long end-to-end delay of data packets, poor packet delivery ratio, and high control packet overhead to the reinitiate the route discovery procedure can occur. In addition, the congested node consumes more energy to route packets, which may result in network partitions.

In this paper, we propose the DCAR (Dynamic Congestion Aware Routing Protocol) which tries to distribute traffic load and avoid congested nodes during the route discovery procedure. DCAR monitors number of packets in an interface queue and defines traffic load as the minimum available buffer length among the nodes in the route. By avoiding the node with minimum available buffer length in the route, we can achieve load balancing, and improve performance in terms of packet delivery ratio and end-to-end delay, etc.

In Section 2, we review and DLAR [4] and give a motivation. In Section 3, we illustrate

* Corresponding Author
Manuscript received May 16, 2006 ; accepted June 13, 2006.
Y.D.Kim, S.H.Lee, D.H. Lee : DGIST

the detail operation of our proposed protocol. Performance evaluation by simulations is presented in Section 4. Finally, concluding remarks are given in Section 5.

Ⅱ. Related Work and Motivation

1. DLAR (Dynamic Load Aware Routing Protocol)

DLAR [4] is a DSR based load balancing routing protocol that uses the traffic load information of the intermediate nodes as the main route selection criterion. Figure 1 illustrates the protocol operation of DLAR for route selection.

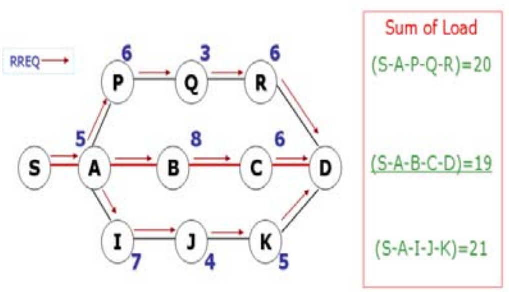

Figure 1. Operation of DLAR

In route discovery procedure of DLAR, the source S broadcasts RREQ (Route Request Packet) to its neighbors. When an intermediate node receives RREQ, it sums and attaches its own load information, then rebroadcasts the RREQ packet. The load information of the node is defined as the number of packets that is currently buffered in its interface. All nodes in the network monitor this load information. Unlike DSR, an intermediate node does not send a RREP (Route Reply Packet) on behalf of the destination in order to deliver fresh entire load information of the route to the destination. The destination node D can receive multiple RREQs from different routes for some amount of time. After receiving RREQs, D selects a best route presumed to be the one having the least load and sends RREP to the source via the reverse path. In the figure, the route S-A-B-C-D is chosen because the route has the least sum (19).

2. Motivation

DLAR only monitors the number of packets

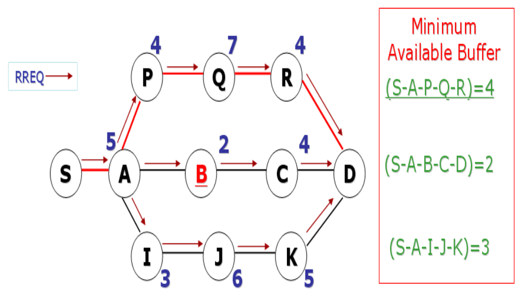

Figure 2. Operation of DCAR

buffered in a node's interface and monitoring the the number of buffered packets does not directly reflect the situation of network congestion. Figure 2 illustrates this problem.

The Figure 2 is the same topology as Figure 1 except that it additionally includes the number of remaining packets in each node's interface queue.

The maximum size of each buffer is assumed to be 10. When the number of currently buffered packets are used a primary key for selecting a route, like DLAR, the destination node D selects the route [S-A-B-C-D] which has the least sum. However, if we look at the remaining available buffer size, node B in the route selected by DLAR is most likely to be congested because its remaining buffer size is only 2. In Figure 2, the route containing node B, which is selected as the best route by DLAR, should be avoided. Another problem of DLAR is that it does not consider the minimum hop count metric significantly. In DLAR, a destination node uses the hop count to select a route only when two or more routes have the even load sums. Lastly, we must consider a case when the buffer size of each node varies, because the packet processing capacity of each node is different from another. In such a case, DLAR can not measure the exact traffic load in every node.

III. Proposed Protocol

1. Route Discovery and Selection Procedure

DCAR is an on-demand routing algorithm and assumes that every node in the network is aware of its own traffic load by monitoring the available buffer size of its interface. To find the most congested node in the discovered routes, we define, Qmin, the minimum available buffer size among the nodes in the route. Each RREQ includes a unique identifier and Qmin fields. If an intermediate node receives duplicate RREQs that have bigger Qmin than the previous one, it can rebroadcast the RREQs because the new route consists of less congested nodes. Otherwise, it drops the duplicated ones. When the intermediate node receives the first RREQ, it compares Qmin in the received RREQ with its own traffic load, represented by the available buffer size. If the traffic load of intermediate node is smaller than received Qmin, the node replaces it with its own information and floods the RREQ.

As shown in Figure 2, the route discovery procedure of the proposed protocol can be described as follows. The source S floods a RREQ packet to find a route to the destination D. When node A receives the RREQ, it updates Qmin with 5 and rebroadcasts RREQ. Then the next node P receives the RREQ and compares Qmin (=5) with its own remaining buffer size (=4). Since Qmin in the RREQ packet is greater than node P's remaining buffer size, it replaces Qmin with its remaining buffer size (=4).

After the same operation is done in node Q and R, D finally receives RREQ containing Qmin of 4 through the route [S-A-P-Q-R-D]. Node D also receives RREQs from other routes: the route [S-A-B-C-D] having Qmin of 2 and [S-A-I-J-K-D] having Qmin of 3. Once the first RREQ has arrived at node D, it sends RREP to node S by using the reverse path. If node D receives a duplicate RREQ with bigger Qmin, it immediately sends the RREP packet again to node S to change the active route with less congested nodes. Otherwise, it simply drops the duplicate RREQs.

During the route discovery procedure, DCAR does not allow intermediate nodes to send RREP using its own route cache, because all RREQs have to be delivered to the destination to check the congestion status of the entire route. If the intermediate nodes can send RREP, the route obtained from the route cache may be stale.

2. Route Selection Algorithm

To choose an optimal route between traffic load information and minimum hop count, we define two thresholds. The first threshold is Max-Threshold (T_{max}) which defines congestion criteria in a node. For example, when T_{max} is 30, we believe that Qmin with more than 30 is not congestion environment. Thus the destination node selects the route with minimum hop count metric. The second threshold is Diff-Threshold (T_{diff}) which is a numerical difference between Q_{min} values of two routes. For example, if T_{diff} is 5 and the difference between two routes is less than 5, we believe that the two load information is almost same. Thus the destination node chooses the route with shortest distance.

IV. Performance Evaluation

1. Simulation Environment

To evalute the performance of the proposed protocol, we used the ns-2 simulator (version 2.28) [5] with the IEEE 802.11b DCF using RTS/CTS. 50 mobile nodes are assumed to be randomly placed in a 1500m x 300m network area. All mobile nodes moved at the given maximum speed of 10m/s with the pause time of 0. The radio propagation range is set to 250m. 20 data connections are established with 5 different packet rates of 5, 10, 15, 20, and 25 to represent different network traffic load. Each source generates constant bit rate (CBR) traffic with packet size of 512 bytes. The maximum buffer size of each node is 50 and 3 different buffer Max-threshold values of 45, 20, 10 and 3 different Diff-threshold values of 5, 3, and 2 are used for the simulation study.

2. Simulation Result

Figure 3 shows the averaged number of dropped packets in a node's interface queue by buffer overflows. As shown in the figure, DCAR provids less buffer overflows because during the route discovery procedure DCAR can avoid congested nodes and can achieve load balancing in the network while the other protocols have frequent packet drops by buffer overflows, which eventually leads to route breakdowns.

Figure 4 shows the packet delivery ratios of DCAR, DLAR and DSR as a function of traffic load. The delivery ratio of DCAR is better than those of DLAR and DSR due to less frequent buffer overflows. Although DLAR also can avoid the congested routes, the performance of DCAR is better because DLAR doesn't know the most congested nodes in routes. Howerver, when the packet rate is over 25, delivery ratios of all the protocols are saturated because the entire network is congested.

Figure 5 shows the packet end-to-end delay as a function of traffic load. When the network traffic load increases, the end-to-end delay of DSR also increases. However, the delays of DCAR and DLAR decreases because these protocols can avoid congested nodes and congested routes. In DSR, the end-to-end delay decreases when the packet rate is above 15. When the traffic load is high and the intermediate nodes are congested, the RREQ packets are also dropped by buffer overflows, so the congested nodes can not forward RREQ packets as well as data packets to the destination. Thus DSR can avoid the congested nodes automatically during the route discovery procedure. In the figure, when compared to DLAR, we can see that the overall performance of DCAR is improved about 10% in terms of the packet delivery ratio and the end-to-end delay.

Figure 6 shows the normalized routing overhead which is the number of the control

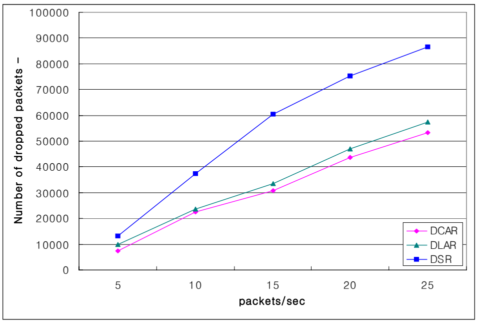

Figure 3. Number of dropped packets

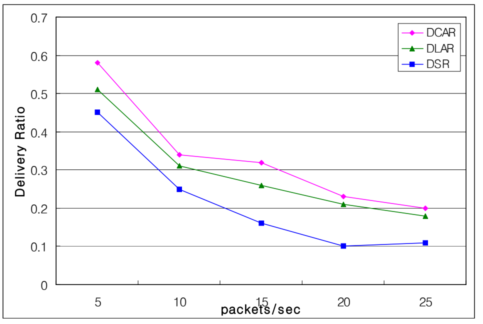

Figure 4. Packet delivery ratio

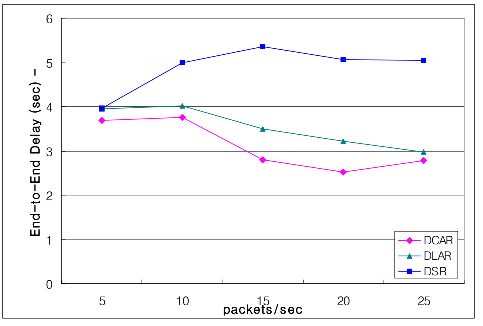

Figure 5. End-to-end delay

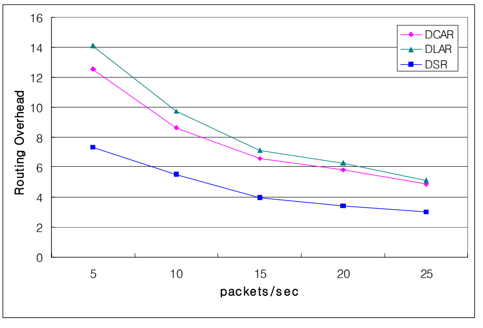

Figure 6. Normalized routing overhead

packets transmitted per data packet successfully delivered at the destination node. We can see that routing overhead of DCAR is larger than that of DSR because DCAR does not allow an intermediate node to send a RREP packet using its own route cache. Thus all RREQ packets are delivered to the destination node by flooding, which results in increased number of control packets during the route discovery process. This is same reason why DLAR has also high control packet overhead. However, the overhead of DLAR is a little bit higher than DCAR because DLAR has more frequent buffer overflows as shown in Figure 3. And we can see that as the traffic load increases, there are more buffer overflows, which leads the control packet overhead to decrease by dropping RREQ packets

Finally, Table I and Table II show the comparison of the performance with different buffer threshold values (Max-threshold and Diff-threshold) of DCAR in order to find the most efficient route. Although it is not easy to select the optimal values, we can see that the buffer threshold value affects the protocol's performance by setting differently. In both scenarios of different packet rates, we can find that DCAR shows the best performance when T_{max} is 20 and T_{diff} is 3, which are approximately correspond to 50% and 5% of the total buffer size, respectively.

Table I. Various thresholds with 5packets/sec

Theshold		20 packets/sec		
T_{max}	T_{diff}	Delivery Ratio	End-to-End Delay	Overflow Dropped
45	5	0.54	3.89	9424
20	3	0.58	3.69	7518
10	2	0.58	3.7	7602

Table II. Various thresholds with 20packets/sec

Theshold		20 packets/sec		
T_{max}	T_{diff}	Delivery Ratio	End-to-End Delay	Overflow Dropped
45	5	0.21	3.28	46117
20	3	0.23	2.52	44502
10	2	0.23	2.94	45513

V. Conclusion

In mobile ad hoc networks, congestion can lead to performance degradation such as many packet losses by buffer overflows and long end-to-end delay. However, existing load balancing protocols do not consider the available buffer size in node's interface queue. That is, they do not consider a certain congested node. In this paper, we have proposed DCAR (Dynamic Congestion Aware Routing Protocol) which can monitor the most congested node in route and can avoid it during the route discovery procedure. We also defined two buffer thresholds to choose the route selection metric between the traffic load and the minimum hop count.

References

[1] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination Sequenced Distance -vector Routing (DSDV) for Mobile Computers," Comp. Commun. Rev., pp. 234 - 44, October. 1994

[2] David B. Johnson, David A. Maltz, Yih-Chun Hu, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR)," Internet Draft, IETF Mobile Ad hoc Networks (MANET) Working Group.

[3] C. E. Perkins and E. Royer, "Ad-hoc on-demand Distance Vector Routing," Proc.2nd IEEE Wksp. Mobile Comp. Sys. App., February 1999.

[4] Sung-Ju Lee and Mario Gerla, "Dynamic Load-Aware Routing in Ad hoc networks," Proceedings of IEEE ICC 2001

[5] S.McCanne and S. Floyd, "NS network simulator," URL:http://www.isi.edu /nsnam/ns

Biography

Young-Duk Kim

2004, BS. Dept. of Computer Education in SungKyunKwan University. 2006, MS. Dept. of Computer and Communication Engineering in POSTECH. Now, Research engineer in DGIST. Interesting: Mobile Computing, Embedded Software.

Email: ydkim@dgist.org

Sang-Heon Lee

1994, BS. Dept. of Electronics in Kyungpook national university. 1996, MS. Dept. of Electronics in Kyungpook national university. Now, Senior research engineer in DGIST. Interesting: Embedded software, Fla-sh Memory, Sensor networks.

Email: pobbylee@dgist.org

Dong-Ha Lee

1985, BS. Dept. of Electronics in Kyungpook national university. 2001, MS. Dept. of Electronics in Kyungpook national university. 2005, Ph.D. Dept. of Electronics in Kyungpook national university. Now, Principal research engineer in DGIST. Interesting: Embedded software, Flash Memory, Sensor networks.

Email: dhlee@dgist.org