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

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

Daegu Gyeongbuk Institute of Science and Technology (DGIST)
Deoksan-Dong 110, Jung-Gu, Daegu, 700-742, Korea
{ydkim, pobbylee, dhlee}@dgist.org

**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:** Ad hoc networks, Routing protocols, Load balancing.

## 1 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

F. Stajano et al. (Eds.): ICUCT 2006, LNCS 4412, pp. 130–139, 2007.

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 ties 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.

The rest of this paper is organized as follows. In Section II, we review two protocols DSR, DLAR [4]. In Section III and IV, we illustrate the motivation and detail operation of our proposed protocol. Performance evaluation by simulations is presented in Section IV. Finally, concluding remarks are given in Section VI.

## 2 Related Works

### 2.1 DSR (Dynamic Source Routing Protocol)

DSR is an on demand routing style protocol for ad hoc networks. Every source node knows a complete route to a destination and maintains a route cache containing the source routes that it is aware of. Each node updates the entries in the route cache if there is a better route, when it learns about a new one. Two main mechanisms of DSR are route discovery and route maintenance.

The route discovery procedure is initiated in an on-demand basis when a source node requires a route to a destination for routing. At first, if there is no route available in the route cache, the source broadcasts a Route Request (RREQ) packet which is flooded throughout the entire network. Each RREQ packet contains a record of listing the address of each intermediate node as well as initiator (source) and target identifier (destination) of the RREQ. If a node receiving the RREQ packet is the destination or an intermediate node having a path to the destination node in its cache, it can reply to the RREQ by sending a Route Reply (RREP) packet which contains the route information between the source and the destination. When the source node receives this RREP, it stores this route in its route cache for sending subsequent data packets to this destination.

In route maintenance procedure, when a node detects that its descendant node in the route is unreachable either by no packet receipt confirmation from the descendant node or no link level acknowledgement in the link layer, it sends a Route Error (RERR) packet to the source node. The RERR packet contains addresses of two end nodes of the broken link. During the propagation of the RERR packet to the source, every intermediate node in the route as well as the source removes the broken route entry in its own route cache and the source invoke the route discovery process again to construct a new route.

### 2.2 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. Similar to DSR, DLAR is also an on-demand routing protocol and has two main mechanisms of route discovery and route maintenance. Figure 1 illustrates the protocol operation of DLAR for route selection.

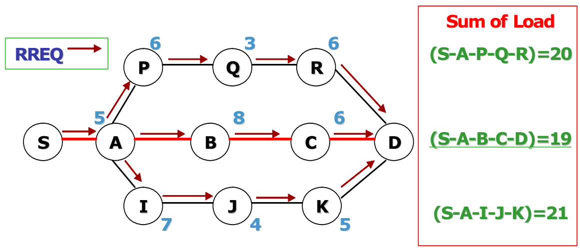


**Fig. 1.** Operation of DLAR

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

In the route maintenance procedure of DLAR, intermediate nodes that are in an active data session periodically piggyback their load information on data packets to report the load status of the active path. If the active path is believed to be congested, the source node reinitiates the route discovery procedure and finds an alternative route. When the intermediate node finds a broken link, it sends a RERR packet to the source node and the source node restarts the route discovery procedure. Elements of the figure described in the caption should be set in italics, in parentheses, as shown in this.

### 2.3 Other Routing Protocols with Load Balancing

There are other routing protocols that consider load balancing as the primary route selection criterion. However, their protocol operations are similar to that of

DLAR or DSR. Thus, we only present their main differences without describing the protocol operations. In LBAR (Load-Balanced Ad hoc Routing) [5], the network load is defined as total number of active routes passing through the node and its neighbors. During the route discovery procedure, load information on all paths from the source to a destination is forwarded to the destination node. In TSA (Traffic-Size Aware Routing) [6], the network load is defined as traffic sizes of routes, which is presented in bytes, not in number of packets because the packet sizes may vary. In MCL (Routing Protocol with Minimum Contention Time and Load Balancing) [7], the network load is defined as the number of neighbors which content with a source node. In CRP (Congestion-adaptive Routing Protocol) [8], although the number of packets currently buffered in interface is also defined as network load, the congestion is classified into three statuses, which are red (very likely congested), yellow (likely congested), and green (far from congested). If a node is aware of congestion symptom, it finds a bypass route which will be used instead of the congested route.

## 3 Motivation

As discussed in the previous section, DLAR is a load-balancing protocol which establishes a route with minimum load. However DLAR only monitors the number of packets buffered in a node's interface and monitoring the number of buffered packets does not directly reflect the situation of network congestion. Figure 2 illustrates this problem.

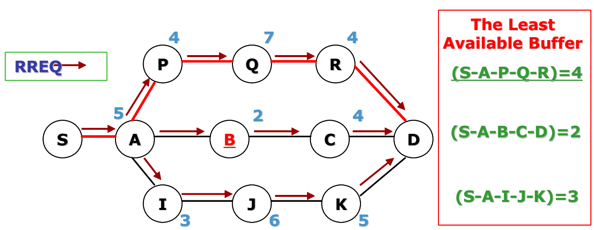


**Fig. 2.** Operation of DCAR

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. When a node does not have enough space to accommodate data packets originated from the new route, the routes including the node should be excluded from the route selection. 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.

The problems addressed above clearly motivates us to devise a new protocol that considers the minimum available buffer size as one of the primary route selection criteria to avoid the most congested nodes and to achieve load balancing in ad hoc networks.

## 4 Proposed Protocol

In this section, we present the proposed protocol, referred to as DCAR (Dynamic Congestion Aware Routing Protocol), to improve the performance by avoiding the congested nodes during the route discovery procedure in mobile ad hoc networks.

### 4.1 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. When a source node wants to send data packets, the source starts a route discovery process by broadcasting a RREQ packet to the entire network. To find the most congested node in the discovered routes, we define, $Q_{min}$, the minimum available buffer size among the nodes in the route. Each RREQ packet includes a unique identifier and $Q_{min}$ fields. In the proposed protocol, if an intermediate node receives duplicate RREQ packets that have bigger $Q_{min}$ than the previous one, it can rebroadcast the RREQ packets because the new route consists of less congested nodes. Otherwise, it drops the duplicated ones. When the intermediate node receives the first RREQ packet, it compares $Q_{min}$ 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 $Q_{min}$, the node replaces it with its own information and floods the RREQ packet.

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

After the same operation is done in node Q and R, the destination node D finally receives the RREQ packet containing $Q_{min}$ of 4 through the route [S-A-P-Q-R-D]. Node D also receives RREQ packets from other routes: the route

[S-A-B-C-D] having $Q_{min}$ of 2 and [S-A-I-J-K-D] having $Q_{min}$ of 3. Once the first RREQ packet has arrived at node D, it sends a RREP packet to node S by using the reverse path. If node D receives a duplicate RREQ packet with bigger $Q_{min}$, 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 RREQ packets.

When node D selects an optimal route, it considers the minimum hop count as well as the traffic load. The detail of the route selection algorithm is described in the following section.

During the route discovery procedure, our protocol does not allow intermediate nodes to send the RREP packet using its own route cache, because all RREQ packets have to be delivered to the destination to check the congestion status of the entire route. If the intermediate nodes can send the RREP packet, the route obtained from the route cache may be stale, especially when the nodes are highly mobile. Thus, by prohibiting intermediate nodes from sending the RREP packet, we can obtain fresh route information.

### 4.2 Route Selection Algorithm

When the destination node receives multiple RREQ packets, the route selection algorithm is used to choose an optimal route. The main operation is to select the route with biggest $Q_{min}$ value among the received RREQ packets. However by selecting a route with only load information, the route length may be long, which result in high delivery latency. So we define two thresholds which can find out whether the route should be selected by the load information or the hop count metric. 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 $Q_{min}$ 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.

### 4.3 Route Maintenance

Route maintenance procedure in DCAR is similar to DSR. If a node detects link breakdown, it sends a Route Error (RERR) packet to the source node along the active path. When a node receives the RERR packet, it removes this broken link from its route cache and performs a packet salvaging process, which attempts to salvage the data packet rather than dropping it. In the packet salvaging process, the node sending a RERR packet searches its own Route Cache for a route from itself to the destination node. If the source node receives the RERR packet from its neighbor node, it will restart the route discovery process to find an alternative route to the destination node.

# 5 Performance Evaluation

## 5.1 Simulation Environment

To evaluate the performance of the proposed protocol, we used the ns-2 simulator (version 2.28) [9] with the IEEE 802.11b DCF using RTS/CTS. There are 50 mobile nodes that are assumed to be randomly placed in a 1500m x 300m rectangle network area. All mobile nodes moved freely at the given maximum speed of 10m/s with the pause time of 0 during the simulation time of 300 seconds. The radio propagation range for a node 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 pair of source and destination nodes of a connection is randomly selected without duplicate sources. Each source generates constant bit rate (CBR) traffic with packet size of 512 bytes. The maximum buffer size of each node's interface is set to 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.

## 5.2 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 provides 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

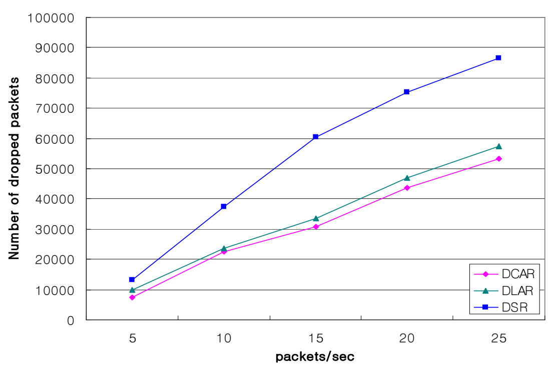


**Fig. 3.** Number of dropped packets

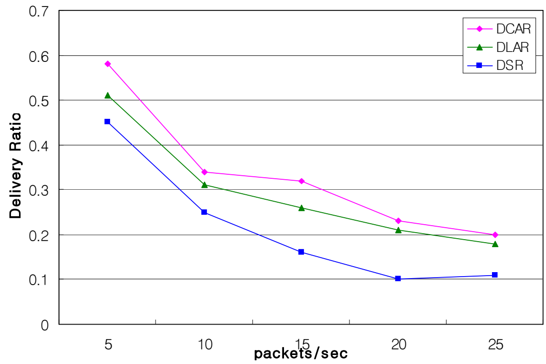


**Fig. 4.** Packet delivery ratio

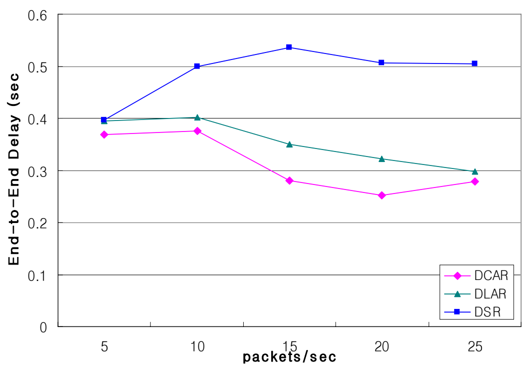


**Fig. 5.** End-to-end delay

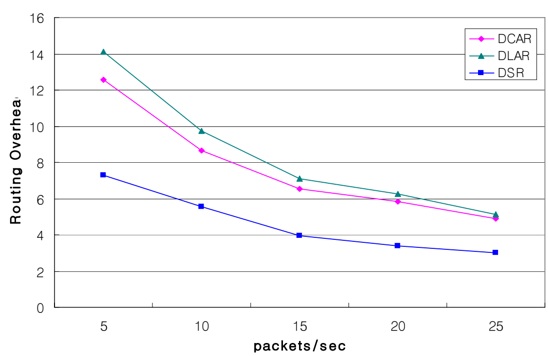


**Fig. 6.** Normalized routing overhead

does not know the most congested nodes in routes. However, 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 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 1.** Various threshold values of DCAR with 5 packets/sec

| Threshold | | 5 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 2.** Various threshold values of DCAR with 20 packets/sec

| Threshold | | 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 |

## 6 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 because the RREQ packet of DCAR contains the minimum available buffer size among the nodes in a discovered route. We also defined two buffer thresholds to choose the route selection metric between the traffic load and the minimum hop count. Simulation study shows that DCAR shows a good performance in terms of packet delivery ratio, end-to-end delay, routing overhead when a network is heavily loaded.

## References

1. C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination Sequenced Distance-vector Routing for Mobile Computers," Comp. Commun. Rev., 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. H. Hassanein and A. Zhou, "Routing with Load Balancing in Wireless Ad hoc Networks," ACM MSWiM, July 2001
6. Abdulranhman H. Altalhi and Golden G. Richard, III, "Load-Balanced Routing through Virtual Paths," IPCCC 2004
7. Bong Chan Kim, Jae Young Lee, Hwang Soo Lee and Joong Soo Ma, "An Ad-hoc Routing Protocol with Minimum Contention Time and Load Balancing," IEEE Global Telecommunications Conference, 2003
8. Tran, D.A. and Raghavendra, H., "Routing with congestion awareness and adaptivity in mobile ad hoc networks," IEEE WCNC 2005
9. S.McCanne and S. Floyd, "NS network simulator," URL: http://www.isi.edu/nsnam/ns