International Journal of Computer Theory and Engineering, Vol. 3, No. 2, April 2011
ISSN: 1793-8201

A Survey on Congestion Adaptive Routing Protocols for Mobile Ad-Hoc Networks

Laxmi Shrivastava, G.S.Tomar and Sarita S. Bhadauria

***Abstract*—Congestion in mobile ad hoc networks leads to transmission delays and packet losses and causes wastage of time and energy on recovery. In the current designs, routing is not congestion adaptive. Routing may let a congestion happen which is detected by congestion control, but dealing with congestion in this reactive manner results in longer delay and unnecessary packet loss and requires significant overhead if a new route is needed. This problem becomes more visible especially in large-scale transmission of heavy traffic such as multimedia data, where congestion is more probable and the negative impact of packet loss on the service quality is of more significance. Routing should not only be aware of, but also be adaptive to, network congestion. Routing protocols which are adaptive to the congestion status of a mobile ad hoc network can greatly improve the network performance. Many protocols which are congestion aware and congestion adaptive have been proposed. In this paper, we present a survey of congestion adaptive routing protocols for mobile ad hoc networks.**

***Index Terms*—AODV, DSR, DLAR, CARM, Congestion, CRP, ECARP**

I. INTRODUCTION

In mobile wireless ad hoc networks the key issue is network congestion and traffic blocking. The congestion occurs in mobile ad hoc networks due to limited availability of resources [1]. In such networks, packet transmissions suffer from interference and fading, due to the shared wireless channel and dynamic topology. Transmission errors also cause burden on the network due to retransmissions of packets in the network. Recently, there has been increasing demand for support of multimedia communications in MANETs. The large amount of real-time traffic tends to be in bursts, is bandwidth intensive and liable to congestion. Congestion leads to packet losses and bandwidth degradation, and wastes time and energy on congestion recovery [2]. Although, it is not possible to get rid of congestion problem but it is possible to limit the impact of congestion on network efficiency by using some suitable procedures and rules for traffic flow. To minimize congestion in network routing algorithms are used. Different dimensions can be used to categorize routing algorithms in MANETs: proactive routing versus on-demand routing, or single-path routing versus multipath routing [3]. In proactive protocols, routes between every two nodes are established in advance even though no transmission is in demand. This approach is not suitable for large networks because many unused routes still need be maintained and the periodic updating may incur overwhelming processing and communication overhead. The on-demand approach is more efficient in that a route is discovered only when needed for a transmission and released when the transmission no longer takes place. However, when a link is disconnected due to failure or node mobility, which often occurs in MANETs, the delay and overhead due to new route establishment may be significant. To address this problem, multiple paths to the destination may be used as in multipath routing protocols [4]. An alternate path can be found quickly in case the existing path is broken. The trade-off, as compared to single-path routing, is the multiplied overhead due to concurrent maintenance of such paths. Furthermore, the use of multiple paths does not balance routing load better than single-pathing unless we use a very large number of paths which is costly and therefore infeasible [5]. There is another dimension for categorizing routing protocols: congestion-adaptive routing versus congestion-unadaptive routing. Most of the existing routing protocols belong to the second group. Some of the existing routing protocols are congestion-aware, and a very few are congestion-adaptive. In congestion-aware routing techniques, congestion is taken into consideration only when establishing a new route which remains the same until mobility or failure results in disconnection. In congestion-adaptive routing, the route is adaptively changeable based on the congestion status of the network [8]. Routing may let a congestion happen which is later detected and handled by congestion control. Congestion non-adaptiveness in routing in MANETs may lead to the following problems:

- *Long delay:* It takes time for a congestion to be detected by the congestion control mechanism. In severe congestion situations, it may be better to use a new route. The problem with an on-demand routing protocol is the delay it takes to search for the new route.
- *High overhead* : In case a new route is needed, it takes processing and communication effort to discover it. If multipath routing is used, though an alternate route is readily found, it takes effort to maintain multiple paths.
- *Many packet losses*: Many packets may have already been lost by the time a congestion is detected. A typical congestion control solution will try to reduce the traffic load, either by decreasing the sending rate at the sender or dropping packets at the intermediate nodes or doing

Manuscript received July 16, 2010

Laxmi Shrivastava is with the Madhav Institute of Technology, Gwalior 474005, Rajiv Gandhi Prodyogiki Vishwavidyala, Bhopal, M.P., India, (e-mail: lselex@yahoo.com)

Sarita S. Bhadauria is with the Madhav Institute of Technology, Gwalior 474005, Rajiv Gandhi Prodyogiki Vishwavidyala, Bhopal, M.P., India,(e-mail: saritamits61@yahoo.co.in.com)

G.S.Tomar is with Machine Intelligence Research Labs (MIR Labs), Gwalior 474011 India,(e-mail: gstomar@ieee.org)

both. The consequence is a high packet loss rate or a small throughput at the receiver.

The above problems become more visible in large-scale transmission of traffic intensive data such as multimedia data, where congestion is more probable and the negative impact of packet loss on the service quality is of more significance. In a dynamic network like a MANET, it is expensive, in terms of time and overhead, to recover from congestion [7]. This survey gives an overview of existing approaches that attempt to provide some congestion adaptive routings in mobile ad hoc networks. The existing approaches are systematically described, classified and compared. The approaches that have been selected for analysis are CARM, CRP, CAAODV, AODVM. While their main objective to make routing protocol congestion adaptive is common to all but adaptation and approach is different and have variations in basic characteristics.

II. ALGORITHMS:

There are many routing algorithms in mobile ad hoc networks for routing and congestion free networks. Some of them are explained below.

Dynamic Load-Aware Routing (DLAR)

In existing on-demand routing protocols such as DSR (Dynamic Source Routing) [3], AODV (Ad-hoc On-demand Distance Vector) [6] and TORA (Temporally Ordered Routing Algorithm); the shortest path routing criteria has been used. The route selection philosophy can lead to network congestion and long delays due to congestion in the network. DLAR [14] considers the load of intermediate nodes as the main route selection metrics and monitors the congestion status of active routes to reconstruct the path when nodes of the route have their interface queue overloaded. DLAR uses the number of packets buffered in the interface as the primary route selection criteria and DLAR builds routes on-demand. When a route is required but no information to the destination is known, the source floods the ROUTE REQUEST packet to discover a route. When nodes other than the destination receive a non-duplicate ROUTE REQUEST, they build a route entry the $\langle \text{source}, \text{destination} \rangle$ pair and record the previous hop to that entry (thus backward learning). This previous node information is needed later to relay the ROUTE REPLY packet back to the source of the route. Nodes then attach their load information (the number of packets buffered in their interface) and broadcast the ROUTE REQUEST packet. After receiving the first ROUTE REQUEST packet, the destination waits for an appropriate amount of time to learn all possible routes. In order to learn all the routes and their quality, the destination node accepts duplicate ROUTE REQUESTS received from different previous nodes. The destination then chooses the least loaded route and sends a ROUTE REPLY packet back to the source via the selected route. During the active data session, intermediate nodes periodically piggyback their load information on data packets. Destination nodes can thus monitor the load status of the route. If the route is congested, a new and highly loaded route is selected to replace the overloaded path. Routes are hence reconstructed dynamically in advance of congestion. The process of building new routes is similar to the initial route discovery process except that the destination floods the packet to the source of the route, instead of flooding to the destination. The source, upon receiving ROUTE REQUEST packets, selects the best route in the same manner as the destination. The source does not need to send a ROUTE REPLY, and simply sends the next data packet using the newly discovered route. Thus DLAR protocol considers intermediate node routing loads as the primary route selection metric. The protocol also monitors the congestion status of active routes and reconstructs the path when nodes of the route have their interface queue overloaded.

Route Selection Algorithms

Three schemes have been used in selecting the least loaded route.

DLAR Scheme1: simply adds the routing load of each intermediate node and selects the route with the least sum. If there is a tie, the destination selects the route with the shortest hop distance. When there are still multiple routes that have the least load and hop distance, the path that is taken by the packet which arrived at the destination earliest between them is chosen.

DLAR Scheme2: Instead of using the sum of number of packets queued at each intermediate node's interface as in scheme 1, scheme 2 uses the average number of packets buffered at each intermediate node along the path. The shortest delay can be used as a tie breaker if needed.

DLAR scheme3: considers the number of congested intermediate nodes as the route selection metric. Basically, it chooses the route with the least number of intermediate nodes that have their load exceeding the threshold value.

TABLE I: ROUTE QUALITIES BASED EACH SCHEME

	Scheme 1	*Scheme 2*	*Scheme 3*
Route *i*	20	5	2 (A and B)
Route *j*	19	6.67	2 (A and E)
Route *k*	21	5.25	1 (A)
Selection	Route *i*	Route *j*	Route *k*

To avoid producing bottlenecks and to use the most up-to-date route information when discovering routes, DLAR does not allow intermediate nodes to reply from cache. DLAR periodically monitors the congestion status of active data sessions and dynamically reconfigures the routes that are being congested. Using the least-loaded routes helps balance the load of the network nodes and utilize the network resources efficiently.

Simulation results showed that DLAR schemes outperform DSR which uses the shortest path and does not consider the routing load. DLAR protocols delivered more fraction of data packets, yielded shorter end-to-end delays, and generated nearly equal number of control packets as DSR.

Distance Vector Routing Protocol - Congestion Aware Distance Vector (CADV)

In a distance vector routing protocol, every host maintains a routing table containing the distances from itself to possible destinations. A mobile host in an ad hoc network can be viewed as a single server queuing system. The delay of sending a packet is positively correlated with congestion. In CADV [12], each entry is associated with an expected delay, which measures congestion at the next hop. Every host estimates the expected delay based on the mean of delay for all data packets sent in a past short period of time. Currently, the length of the period is equal to the interval between two periodical updates. The expected delay is computed as

$$E[D] = \frac{\sum D_i}{n} L,$$

where n is the number of sent packets and L is the length of MAC layer packet queue. $E[D]$ estimates the time a newly arrived packet has to wait before it is sent out. When a host broadcasts an update to neighbors, it specifies the delay it may introduce. A routing decision is made based on the distance to the destination as well as the expected delay at the next hop. CADV tries to balance traffic and avoid congestion by giving priority to a route having low expected delay. When making routing decisions, a function $f(E[D], distance)$ is used to evaluate the value of a route.

A CADV routing module consists of three components.

- *Traffic Monitor* monitors traffic going out through the link layer. Currently it keeps track of the average delay for sending one data packet in recent period of time. The time period is specified by the route maintenance component.
- *Traffic control* determines which packet is the next to send or drop, and reschedules packets if needed. At present, it supports a drop tail FIFO queue and provides functionality to re-queue packets.
- *Route maintenance* is the core component. Its functionalities include exchanging information with neighbors, evaluating and maintaining routes, managing the traffic monitor and traffic control components.

CADV outperforms AODV in delivery ratio by about 5%, while introduces less protocol load. CADV introduces higher end-to-end delay than AODV and DSDV do when the number of connections is greater than 10, because it may choose longer route to forward packets. The delay is rather stable with the increase of the number of connections. CADV consumes less power. For the movements of mobile hosts generated by the random waypoint model, the link change and route change are, with a very high probability, linear functions of the maximum speed, and linear functions of the pause time, respectively. The protocol load for the proactive routing protocols (such as DSDV) grows as the number of hosts increases, while that of the on-demand routing protocols (such as AODV) increases with the number of source-destination (S-D) pairs. The proactive approach performs better when the number of S-D pairs is close to the number of hosts. CADV is not congestion adaptive. It offers no remedy when an existing route becomes heavily congested.

Congestion – Aware Routing Protocol for Mobile Adhoc Networks (CARM)

A congestion aware routing protocol for mobile ad hoc networks which uses a metric incorporating data- rate, MAC overhead, and buffer delay to combat congestion. This metric is used, together with the avoidance of mismatched link data-rate routes to make mobile ad hoc networks robust and adaptive to congestion. A further cause of congestion is link reliability. If links break, congestion is increased due to packet retransmission. CARM [10],[11]applies a link data-rate categorization approach to prevent routes with mismatched link data-rates. The MAC overhead from (2) is a good me asure of congestion, being a combination of the two factors. In addition to MAC overhead, queuing delay is a useful measure of congestion. CARM is an on-demand routing protocol that aims to create congestion-free routes by making use of information gathered from the MAC layer. The CARM route discovery packet is similar to that in DSR where every packet carries the entire route node sequence. CARM employs the *Weighted Channel Delay* (WCD) metric to account for the congestion level. In addition, CARM adopts a route effective data-rate category scheme to combat the *mismatched data-rate route* (MDRR) problem. The combination of these two mechanisms enables CARM to ameliorate the effects of congestion in multi-rate networks. CARM uses the same route maintenance approach as that in DSR. In the first, only WCD metric it taken into account in DSR, which is called CARMdelay, In the second, both the WCD and the effective link data-rate category (ELDC) scheme are taken into account, which is called CARM. It has been noted that in DSR the routing load is dominated by RREP packets. However, in CARM due to the suppression of RREPs at intermediate nodes, CARMdelay and CARM work to exclude congested links via the use of the WCD. In CARM, ELDCs also contribute to congestion control. So, DSR yields lower overhead due to route discovery, it requires discovery more often due to congestion. In CARMdelay and CARM, the reduced number of congested links in established routes contributes to better performance in high traffic loads. CARM utilizes two mechanisms to improve the routing protocol adaptability to congestion. Firstly, the weighted channel delay (WCD) is used to select high throughput routes with low congestion. The second mechanism that CARM employs is the avoidance of mismatched link data-rate routes via the use of effective link data-rate categories (ELDCs). In short, the protocol tackles congestion via several approaches, taking into account causes, indicators and effectors. The decision made by CARM are performed locally. Our simulation results demonstrate that CARM outperforms DSR due to its adaptability to congestion.

A Hop-by-Hop Congestion-Aware Routing Protocol for Heterogeneous Mobile Ad-hoc Networks

A hop-by-hop congestion aware routing protocol employs a combined weight value as a routing metric, based on the data rate, queuing delay, link quality and MAC overhead. Among the discovered routes, the route with minimum cost index is selected, which is based on the node weight of all the in network nodes. The nodes are usually heterogeneous in realistic ad hoc networks. For example, in a battlefield network, portable wireless devices are carried by soldiers, and more powerful and reliable communication devices are carried by vehicles, tanks, aircraft, and satellites and these devices/nodes have different communication characteristics in terms of transmission power, data rate, processing capability, reliability, etc. Hence it would be more realistic to model these network elements as different types of nodes [1]. Such heterogeneous networks nodes are portable to transmit at different power levels and thus cause communication links of varying ranges. A congestion-aware routing metric for MANETs should incorporate transmission capability, reliability, and congestion around a link. A hop-by-hop congestion aware routing protocol[13] which employs the following routing metrics:

- Data-rate
- Buffer queuing delay
- Link Quality
- MAC Overhead

In this routing protocol, after estimating the above metrics, a combined weight value is calculated for each node. We select any multi path on-demand routing protocol, which discovers multiple disjoint routes from a source to destination. Among the discovered routes, the route with minimum cost index is selected, which is based on the node weight of all the in-network nodes for each packet successfully delivered from the source node to the destination node. The node's cost index is calculated in a backward propagating way. The cost indices of a node's possible downstream neighbors are obtained by the feedbacks of its downstream neighbors.

C. Link Quality Estimation

To be able to see that a node is moving and a route is about to break, we rely on the fact that communication is based on electronic signals. Because of that it is possible to measure the quality of the signal and based on that guess if the link is about to break. This can be used by the physical layer to indicate to the upper layer when a packet is received from a host, that is sending with a signal lower than a specific value and then indicate that that node is in pre-emptive zone [9],[10]. Thus, using the received signal strength from physical layer, link quality can be predicted and links with low signal strength will be discarded from the route selection. When a sending node broadcasting RTS(Request –To-Send) packet, it piggybacks its transmissions power P_t. On receiving the RTS packet, the intended node measures the signal strength received which holds the following relationship for free-space propagation model [11].

$$P_r = P_t(\lambda/4\pi d)^2\, G_t G_r$$

Where λ is the wavelength carrier, d is distance between sender and receiver, and receiving omni-directional antennas, respectively. The G_t and G_r are unity gain of transmitting effects of noise and fading are not considered.

So, the link quality $L_q = P_r$

D. Estimating MAC Overhead

In this network, we consider IEEE 802.11 MAC with the distributed coordination function (DCF). It has the packet sequence as request-to-send (RTS), clear-to-send (CTS), and data, acknowledge (ACK). The amount of time between the receipt of one packet and the transmission of the next is called a short inter frame space (SIFS). Then the channel occupation due to MAC contention will be

$$C_{OCC} = t_{RTS} + t_{CTS} + 3t_{SIFS}$$

Where t_{RTS} and t_{CTS} are the time consumed on RTS and CTS, respectively and t_{SIFS} is the SIFS period. Then the MAC overhead OHMAC can be represented as :

$$OHMAC = COCC + tacc$$

Where t_{acc} is the time taken due to access contention. The amount of MAC overhead is mainly dependent upon the medium access contention, and the number of packet collision s. That is, OH_{MAC} is strongly related to the congestion around a given node. OH_{MAC} can become relatively large if congestion is incurred and not controlled, and it can dramatically decrease the capacity of a congested link.

E. Estimating End to End Delay

There is a significant variation between the end-to-end delay reported by RREQ-RREP measurements and the delay experienced by actual data packets. We address this issue by introducing a DUMMY-RREP phase during route discovery. The source saves the RREP packets it receives in a RREP TABLE and then acquires the RREP for a route from this table to send a stream of DUMMY data packets along the path traversed by this RREP. DUMMY packets efficiently imitate real data packets on a particular path owing to the same size, priority and data rate as real data packets. H is the hop count reported by the RREP. The number of packets comprised in every stream is 2H. The destination computes the average delay D_{avg} of all DUMMY packets received, which is sent through a RREP to the source. The source selects this route and sends data packets only when the average delay reported by this RREP is inside the bound requested by the application. The source performs a linear back-off and sends the DUMMY stream on a different route selected from its RREP TABLE when the delay exceeds the required limit.

F. Estimating Data Rate

In heterogeneous ad hoc networks, throughput through a given route is depending on the minimum data rate of all its links. In a route of links with various data rates, if a high data rate node forwards more traffic to a low data rate node, there is a chance of congestion. This leads to long queuing delays in such routes. Since congestion significantly reduces the effective bandwidth of a link, the effective link data-rate is given by

$$D_{rate} = D_{size} / C_{delay}$$

Where D_{Size} is the data size and C_{delay} is the channel delay.

III. Congestion Aware Routing Protocol (CARP)

CARP is an on-demand routing protocol that aims to create congestion-free routes by making use of information gathered from the MAC layer. CARP employs a combined weight metric in its standard cost function to account for the congestion level. For establishing multiple disjoint paths, we adapt the idea from the Adhoc On-Demand Multipath Distance Vector Routing (AOMDV) [5]. The multiple paths are computed during the route discovery. We now calculate the node weight metric NW which assigns a cost to each link in the network and select maximum throughput paths, avoiding the most congested links. The NW for the link from node *i* to a particular neighboring node is given by

$$NW = (L_q * D_{rate}) / (OH_{MAC} * D_{avg})$$

A. Route Request

Let us consider the route

$$S - N1 - N2 - N3 - D$$

To initiate congestion-aware routing discovery, the source node S sends a RREQ. When the intermediate node *N1* receives the RREQ packet, it first estimates all the node weight metrics.

The node *N1* then calculates its node weight NW_{N1}

$$RREQ\mathrm{N1} \longrightarrow N2$$

N2 calculates *NW*N2 and forwards the RREQ packet

$$RREQ\mathrm{N2} \longrightarrow N3$$

Finally the RREQ reaches the destination node D with the sum of node weights

$$RREQ\mathrm{N3} \longrightarrow D$$

B. Route Reply

The Destination node D sends the route reply packet RREP along with the total node weight to the immediate upstream node N_3

$$RREQ\mathrm{D} \longrightarrow N3$$

Now N_3 calculates its cost C based on the information from RREP as

$$C\mathrm{N3}= (NW_{N1}+NW_{N2}+NW_{N3}) - (NW_{N1}+NW_{N2})$$

By proceeding in the same way, all the intermediate hosts calculate its cost.

On receiving the RREP from all the routes, the source selects the route with minimum cost value.

IV. Congestion Adaptive Routing in Ad hoc Networks

CRP[8],[9] protocol tries to prevent congestion from occurring in the first place. CRP uses additional paths (called “bypass”) to reduce packet delay, but tries to minimize bypass use to reduce the protocol overhead. Traffic is split over the bypass and the primary route probabilistically and adaptively to network congestion. Hence, 1) power consumption is efficient because traffic load is fairly distributed and 2) congestion is resolved beforehand and, consequently, CRP enjoys a small packet loss rate.

In CRP, every node appearing on a route warns its previous node when prone to be congested. The previous node then uses a “bypass” route bypassing the potential congestion to the first non-congested node on the route. Traffic will be split probabilistically over these two routes, primary and bypass, thus effectively lessening the chance of congestion occurrence. CRP is on-demand and consists of the following components:

(1) Congestion monitoring, (2) Primary route discovery, (3)Bypass discovery, (4) Traffic splitting and congestion adaptivity, (5) Multi-path minimization, and (6) Failure recovery.

C. Congestion Monitoring

A variety of metrics can be used for a node to monitor congestion status. Chief among these are the percentage of all packets discarded for lack of buffer space, the average queue length, the number of packets timed out and retransmitted, the average packet delay, and the standard deviation of packet delay. In all cases, rising numbers indicate growing congestion. Any of these methods can work with CRP in practice. We further classify the congestion status at a node into 3 levels: “green”, “yellow”, and “red”. A node is said to be “green” if it is far from congested, “yellow” if likely congested, or “red” if most likely or already congested. A bypass is a path from a node to its next green node. The next green node is the first green node at least two hops away downstream on the primary route.

D. Primary Route Discovery

To find a route to the receiver, the sender broadcasts a REQ packet toward the receiver. The receiver responds to the first copy of REQ by sending toward the sender a REP packet. The REP will traverse back the path that the REQ previously followed. This path becomes the primary route between the sender and the receiver. Nodes along this route are called primary nodes. To reduce traffic due to route discovery and better deal with congestion in the network, we employ two strategies: (1) the REQ is dropped if arriving at a node already having a route to the destination, and (2) the REQ is dropped if arriving at a node with a ”red” congestion status.

E. Bypass Discovery

A node periodically broadcasts to neighbors a UDT (update) packet. This packet contains this node’s congestion status and a set of tuples {destination R, next green node G, distance to green node m}, each for a destination R that the node has a route to. The purpose is that when a node N receives a UDT packet from its next primary node Nnext regarding destination R, N will be aware of the congestion status of Nnext and learn that the next green node is G which is m hops away on the primary route. If Nnext is yellow or red, a congestion is likely ahead if data packets continue to be forwarded on link N Nnext . Since CRP tries to avoid congestion from occurring in the first place, N starts to discover a bypass route toward node G - the next green node of N known from the UDT packet. This bypass search is similar to primary route search, except that: (1) the bypass request packet’s TTL is set to 2 × m, and (2) the bypass request is dropped if arriving at a node (neither N nor G) already present on the primary route. Thus, it is not costly to find a bypass and the bypass is disjoint with the primary route, except that they join at the end nodes N and G. It is possible that no bypass is found due to the way the bypass request approaches G. In which case, we continue using the primary route. However, [1] finds that the chance for a “short-cut” to exist from a node to another on a route is significant.

F. Traffic Splitting and Congestion Adaptability

At each node that has a bypass, the probability p to forward data on the primary link is initially set to 1 (i.e., no data is sent along the bypass). It is then modified periodically based on the congestion status of the next primary node and the bypass route (see Table I). The congestion status of the bypass is the accumulative status of every bypass nodes. The key is that we should increase the amount of traffic on the primary link if the primary link leads to a less congested node and reduce otherwise. An example is demonstrated by Figure 1, where the bypass from A is A→ X → Y→ C, from B is B→Y → Z→ E, and from D is D →W→F .

G. Multi-path Minimization

To reduce the protocol overhead, CRP tries to minimize using multiple paths. If the probability p to forward data on a primary link approaches 1.0, this means the next primary node is far from congested or the bypass route is highly congested. In this case, the bypass at the current node is removed. Similarly, if the next primary node is very congested (p approaches 0), the primary link is disconnected and the bypass route becomes primary. To make the protocol more lightweight, CRP does not allow a node to have more than one bypass. The protocol overhead due to using bypass is also reduced partly because of short bypass lengths. Each bypass connects to the first non-congested node after the congestion spot, which should be just a few hops downstream.

H. Failure Recovery

A desirable routing protocol should gracefully and quickly resume connectivity after a link breakage. CRP is able to do so by taking advantage of the bypass routes currently available. For instance, in Figure 1, if node C or D fails or moves away, B can take the bypass B →Y→Z →E.

The following highlights were concluded from our performance evaluation:

End-to-end delay: Consistently in simulation runs, CRP provided an average delay shorter than did AODV and DSR. In addition, delay standard deviation was smaller in CRP than in the other protocols, making CRP more suitable for real-time and multimedia applications.

Data packet delivery ratio: Both CRP and AODV successfully delivered more data packets than DSR. However, when the network was heavily loaded, whether the network was steady or highly mobile, CRP performed better than AODV. In the other cases (only a few), they performed similarly.

Protocol overhead: Both CRP and DSR were more lightweight than AODV. CRP was significantly better when the network traffic became heavier.

Energy efficiency: CRP and AODV were consistently better than DSR. CRP was more efficient than AODV, especially when the network traffic was heavier.

CRP, a congestion-adaptive routing protocol for MANETs has been proposed. CRP enjoys fewer packet losses than routing protocols that are not adaptive to congestion. This is because CRP tries to prevent congestion from occurring in the first place, rather than dealing with it reactively. A key in CRP design is the bypass concept. A bypass is a subpath connecting a node and the next noncongested node. If a node is aware of a potential congestion ahead, it finds a bypass that will be used in case the congestion actually occurs or is about to. Part of the incoming traffic will be sent on the bypass, making the traffic coming to the potentially congested node less. The congestion may be avoided as a result. Because a bypass is removed when the congestion is totally resolved, CRP does not incur heavy overhead due to maintaining bypass paths. The bypass maintenance cost is further reduced because a bypass is typically short and a primary node can only create at most one bypass. A short end-to-end delay is also provided by CRP. Indeed, since CRP makes the network less congested, the queuing delay is less. Furthermore, since recovery of a link breakage is realized gracefully and quickly by making use of the existing bypass paths, the delay due to new-route establishment is also low. Our ns-2-based simulation has confirmed the advantages of CRP and demonstrated a significant routing and energy efficiency improvement over AODV and DSR.

ECARP: An Efficient Congestion Adaptive Routing Protocol for Mobile Ad Hoc Networks

The proposed congestion control routing protocol outperform all the other routing protocols during heavy traffic loads.

The simulation experiment with five CBR traffic sources sessions between to common destination using AODV[7], DSR[6], DSDV and TORA were conducted. The performance metrics are Average Packet Delivery Ratio and Average End-to-End delay. For observation in as constraint situation we have considered only Average Packet Delivery Ratio, In normal case AODV outperforms better than other three routing protocols . The TORA performs better than DSDV. But under constraint situation of same routing protocols behaves differently. With six CBR traffic sources to a common destination, AODV suffers degradation up to 35% whereas DSR suffers only 10% compared to normal situation. TORA suffers degradation of 45% whereas DSDV suffers only 15%. On comparing their performances. It was observed that DSR performs better than other three routing protocols. The main reason for performance degradation in packet delivery rato is due to packet drops by the routing algorithm after being failed to transfer the data in the active routes. There are several reasons for packet drops such as network partitioning, link break, collision and congestion in the ad hoc networks. The main important property of routing algorithm is quick link recovery through efficient route maintenance. Therefore, the DSR routing protocol has fast reaction for kink recovery and finds alternative path (during congestion) in compared with AODV and other routing protocols in the given situation. AODV keeps only the active and removes the state ones. Therefore, unavailability of the alternate routes leads to route discovery by the source node. The congestion will be high when multiple CBR sources send data to a single destination. In AODV, the intermediate, nodes are unable to send the data packets, link break situation perceived by AODV sends route error or finding new route through source will result in packet drops resulting in degradation of packet delivery ratio, increase in Average End-to-End delay and increase in Routing overhead. In DSR, the routes caches heave more alternative routes and in the constrained environment when most of the routes are fresh, therefore the route repair is localized. DSR also has more provision of more than one mechanism for local route repairs such as replaying to Route Requests using Cached Routes.

DSDV is proactive routing protocol and has more alternate paths than TORA. Thus, performance of DSDV is better than TORA in a constraint environment. The delay for establishing route is less when compared to TORA, Routing overhead is very high in reactive protocols AODV and DSR when compared to DSDV and TORA.

The ECARP designed to ensure the high availability of alternative routes and reduce the rate of stale route . This can be achieved by increasing the parameters of routing protocols (especially in AODV) that normally take more time for link recovery. The parameters such as active route time-out, route reply wait time, reverse route life, TTL_start, TTL_increment, TTL_threshold and delete_period. In ECARP congestion control algorithm we propose some parameters of AODV to be increased. Thus AODV ensures the high availability of alternative routes and reduce the rate of broken route removal process.

I. Ecarp Congestion Control Algorithm

This algorithm provides solution to improve routing protocols due to constrained environment.

Step1: check the occupancy of link layer buffer of node periodically, Let C be the congestion status estimated.

Step2: Compute Cs = Number of packet buffered in Buffer Buffer size

Step 3: Set the status for Congestion. It can be indicated by three statues "Go", "Careful" and "Stop" . ["Go" indicates there is no congestion with Cs ½ "careful" indicates the status likely to be congested with ½ Cs ¾ and "Stop" indicates the status likely to be congested ¾ Cs 1.]

Step4: Invoke congestion control routine when link failed event has occurred in data transfer with using active route or ¾ Cs 1.

Step 5: Assume that neighbor will have alternate route or noncongested route to the destination.

Step 6: Make Query to non-congested neighbors for route to destination.

Step 7: After obtaining the routes from the neighbors, select route with minimum hops.

Step 8: Once route is finalized start sending the data packets through non-congested route.

Step 9: If there is no alternative route to destination then start splitting the traffic to the less congested route.

Step 10: Traffic splitting effectively reduces the congestion status at the next main node.

In normal case AODV better then DSR using packet delivery ratio and average delay. But in constraint situation of many CBR sources leading to same destination, DSR works better than AODV and DSDV was improved by using local corrective mechanisms which are quick reactive to local corrective mechanisms which are quick reactive to local route repairs to overcome the problem of congestion.

V. COMPARISON

Congestion is a dominant reason for packet drops in ad hoc networks [15]. Lu et al. [15] found that AODV is ineffective under stressful network traffic situations. They therefore proposed a modified version of AODV (called CADV) CADV, the difference being that a node with low routing load is favored to be included in the routing path during the route discovery phase which favors nodes with short queuing delays in adding into the route to the destination. While this modification may improve the route quality, the issues of long delay and high overhead when a new route needs to be discovered remain unsolved. Furthermore, CADV is not congestion adaptive. It offers no remedy when an existing route becomes heavily congested. This is probably the reason that CADV improves AODV in delivery ratio by only 5 percent in highly loaded networks. (CRP improves by 10 percent-28 percent.) A dynamic load-aware routing protocol (DLAR) was proposed in [12]. DLAR is similar to CADV, the difference being that a node with low routing load is favored to be included in the routing path during the route discovery phase.

CADV, DLAR, as well as most on-demand routing protocols, are a single-pathing. Multipath protocols may be used to shorten the delay due to new-route discoveries. Some of these protocols are multipath versions of existing on-demand single-path protocols, such as [3], [5] (extensions to AODV) and [4] (extension to DSR). Another multipath protocol, named MDVA, was proposed in [29].

CRP also sends packets on both bypass paths and primary routes simultaneously. However, CRP distributes incoming traffic over the bypass and primary routes dynamically based on the current network congestion situation. Congestion is subsequently better resolved.

In CRP, since a bypass is established from a node to the next noncongested node on the primary route, it is not costly to maintain and not time-consuming to discover. On the contrary, an alternate path in other multipath routing schemes is longer because it is destined all the way for the destination.

In ECARP congestion control algorithm some parameters of AODV such as active route time-out, route reply wait time, reverse route life, TTL_start, TTL_increment, TTL_threshold and delete_period are increased. Thus AODV ensures the high availability of alternative routes and reduce the rate of broken route removal process.

V. CONCLUSION

It is clear from the algorithms available for having adaptive solution for congestion in the network as due to vast payloads on network, which may be due to flooding of packets or may be due to repeat requests on the basis of error correction techniques. This is clear from the investigations that new set of solutions are needed to overcome the problem congestion in network. It is also clear that congestion is the problem associated with the network and has to be countered by having compromised solution rather than elimination.

REFERENCES

[1] S. Ramanathan, M. E. Steenstrup, "A survey of routing techniques for mobile communications networks, mobile networks and applications," Vol. 1, pp. 98–104, 1996.

[2] C. E. Perkins, "Ad hoc on demand distance vector (AODV) routing," IEFT Internet Draft, November 1998,

[3] D. B. Johnson, D. A. Maltz, and J. Broch, "Dynamic source routing for multihop wireless ad hoc networks," in Ad Hoc Networking,

edited by Charles E. Perkins, Addison-Wesley, Chapter 5, pp 139–172, 2001.

[4] Y. Lu, W. Wang, Y. Zhong, and B. Bhargava, "Study of distance vector routing protocols for mobile ad hoc networks," Proceedings IEEE International Conference Pervasive Computing and Comm. (PerCom), pp. 187–194, March 2003.

[5] [M. Marina and S. Das, "On-demand multipath distance vector routing in ad hoc networks," Proceedings IEEE International Conference Network Protocols (ICNP), pp. 14–23, 2001.

[6] J. Broch, D. Johnson, and D. Maltz, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks," IETF Internet draft, Oct.1999.

[7] C.E. Perkins, E.M. Belding-Royer, and I. Chakeres, "Ad Hoc On Demand Distance Vector (AODV) Routing," IETF Internet draft, Oct. 2003.

[8] H. Raghavendra and D.A. Tran, "Congestion Adaptive Routing in Ad Hoc Networks (Short Version)," Proc. ACM Int'l Conf. Mobile Computing and Networking (MOBICOM), Oct. 2004.

[9] H. Raghavendra and D.A. Tran, "Congestion Adaptive Routing in Ad Hoc Networks" IEEE Transactions on Parallel and Distributed Systems, Vol. 17, No. 11 , November 2006.

[10] D.A. Tran and H. Raghavendra, "Routing with Congestion Awareness and Adaptivity in Mobile Ad Hoc Networks," Proc. IEEE Wireless Comm. and Networking Conf. (WCNC), Mar. 2005.

[11] Xiaoqin Chen, Haley M. Jones†, A .D .S. Jayalath, "Congestion-Aware Routing Protocol for Mobile Ad Hoc Networks" in proceedings of IEEE conference on Vehicular Technology,pp.21-25, 2007, Doi.10.1109/VETECF.2007.21.

[12] Xiaoqin Chen, Haley M. Jones and Jayalath, "Congestion-Aware Routing Protocol for Mobile Ad Hoc Networks" IEEE 66th conference on Vehicle Technology, pp.21-25, October 2005.

[13] S. Santhosh baboo and B.Narasimhan, "A Hop-by-Hop Congestion-Aware Routing Protocol for Heterogeneous Mobile Ad-hoc Networks" (IJCSIS) International Journal of Computer Science and Information Security, Vol. 3, No. 1, 2009.

[14] S. J. Lee and M. Gerla, "Dynamic load-aware routing in ad hoc networks," Proceedings IEEE International Conference on Communications, pp. 3206–3210, June 2001 .

[15] T.G.Basavraju, S.K.Sarkar, C.Puttamadappa, M.A.Gautham, "ECARP: An Efficient Congestion Adaptive Routing Protocol for Mobile Adhoc Networks", ITS Telecommunication Proccedings, pp.715-718, 2006.

Laxmi Shrivastava – born in 1970 in Mathura, Uttar Pradesh, India, received B.Tech. degree in Electronics and Communication Engineering from Govind Ballabh Pant University of Agriculture and Technology, Pantnagar, Nainital, India in 1993, M.Tech. degree in Electronics and Comuunication (Microwave Engineering) from Madhav Institue of Technology, Gwalior, Rajiv Gandhi Prodyogiki Vishwavidyala, Bhopal, M. P., India in 2006 and pursuing Ph.D. from the same university. She is currently working as Lecturer(Selection Grade) in Electronics Engineering Department at Madhav Institue of Technology, Gwalior, Rajiv Gandhi Prodyogiki Vishwavidyala, Bhopal, M.P., India. She is associate member of Institution of Electronics and Telecommunication Engineers (IETE), India and life member of Computer Society of India (CSI), India. Her research interests are development of algorithm for congestion in Mobile Adhoc Networks and security of these networks.

Prof. Geetam Singh Tomar (IEEE M' 2002), received his UG, PG, and Ph. D. degrees in electronics engineering from reputed universities of India. He is presently Principal, Malwa Institute of Technology & Management, Gwalior, India. He is actively involved in research and consultancy with Machine Intelligence Research Labs (USA), India section, Gwalior. He is guiding Ph.D. Students in the area of air interface for cellular and mobile ad-hoc networks, Antenna design and fabrication, wireless sensors networks and underwater communication. He is actively involved in IEEE activities and is chief editor of international journals.

Sarita S. Bhadauria–born in 1961in Gwalior, Madhya Pradesh, India, received B.E. degree in Electronics Engineering from Madhav Institute of Technology& Science, Gwalior, Madhya Pradesh, India in 1985, M.Tech. degree in Computer Science & Tech. from University of Roorkee , Roorkee, Uttar Pradesh, India, in 1993 and Ph.D. in Computer Science and Engineering from Rajiv Gandhi Technological University, Bhopal, Madhya Pradesh, India.

She is currently working as Professor in Electronics Engineering Department at Madhav Institue of Technology, Gwalior, Rajiv Gandhi Technological University, Bhopal, M.P., India. She is life member of Institution of Electronics and Telecommunication Engineers (IETE), India, life member of Computer Society of India (CSI), India, life Member of the Indian Society for Technical Education (ISTE), India and associate member of the Institution of Engineers (IE), India. Her research interests are Image Processing, Networking.