

Multi-Path Routing Protocol to Control Congestion for Load Balancing in Mobile Adhoc Networks

M.V.Narayana^{1*}, J Rajeshwar²

^{1,2}Dept. of CSE, Guru Nanak Institutions Technical Campus, Hyderabad, India

Available online at: www.ijcseonline.org

Accepted: 19/Jan/2019, Published: 31/Jan/2019

Abstract— The multi-path routing protocol is one of the approaches used to have less overhead, better bandwidth cost through the load distribution among a set of paths. However, due to the interference of the paths during the communication, it increases the end to end delay and do not work well under a highly congested network. We have proposed the congestion aware multi-path routing protocol with load balancing. The paper explains the problem associated with the end-to-end delay. It deals with the problem of channel interference during the Multi-path communication. It suggests the congestion control mechanisms to improve the protocol performance. It proposes an additive increase/ multiplicative decrease and fast retransmits and recovery process and also it suggest the algorithm for load balancing scheme. In order to minimize the congestion formation, it selects the node disjoint paths for communication. The correlation factor among the selected multiple paths are calculated. It calculates the congestion formation in the network. It compares the proposed MPR protocol with ordinary DSR through the simulation and the simulation results are discussed.

Keywords—Multi-Path Routing, Congestion, Correlation Factor, Load Balance, Disjoint Path.

I. INTRODUCTION

The multi-path routing protocol is one of the approaches used to have less overhead, better bandwidth cost through the load distribution among a set of paths. However, due to the interference of the paths during the communication, it increases the end to end delay and do not work well under a highly congested network.[1][3] To maintain and allocate network resources effectively and fairly among a collection of users is a major issue. The resources shared mostly are the bandwidth of the links and the queues on the routers or switches. The data packets are queued in these queues awaiting transmission. When too many packets are contending for same link, the queue overflows and packets have to be dropped. When such drops become common events, the network is called congestion.[4][6] Hence, the multi-path communication is an optimized solution for the congestion minimization (Ron Banner and Ariel Orda, 2009; B. Sasthiri and T. Prakash, 2012). In this paper, the congestion aware multi point relay protocol is proposed.

1.1 Source node

Initially, the source node does not have the location information of the destination node. Therefore, it broadcasts the RREQ message for the route discovery. The RREQ carries the source ID, destination ID, and a path vector which

contains the relaying node ID, and the amount of the traffic the relaying node has delivered. The RREQ is again forwarded from the neighbouring nodes till the destination node is reached; this mechanism is known as flooding. [7]After broadcasting the RREQ message, the source node waits for RREP packets till predefined amount of time. Once this amount of time has expired after broadcasting RREQ packet, and no RREP packet has received the source node again broadcasts the RREQ message. Once the source node receives RREP packet, it comes to know that a route has been built and starts transmitting data packets via received route.

1.2 Intermediate Nodes

When the intermediate node receives this RREQ packet, it first checks its routing table, whether it's any of the neighbouring nodes active for that instant of time. If yes, then it drops the RREQ packet, else adds its node ID, load information in the path vector of the RREQ packet and again rebroadcast it. When an intermediate node receives the RREP packet, it first updates its routing table and the TTL values are listed in the RREP packet along with the cache information and then unicast this RREP packet and waits for the data packets to arrive. When the intermediate node receives the data packets, the \$ttl value replaces the TTL value.[7] Once this \$ttl time is out the flag bit is reset. When an intermediate node relays a data packet, then for every data packet being relayed it updates its routing table using the respective load information and \$ttl value being defined.

1.3 Destination Node

Once the destination node receives the RREQ message, it first records the TTL value defined in the RREQ packet for that respective route and waits for a predefined amount of time to collect other RREQ messages. Soon after the collection of route information, the destination node chooses the best node whose path load is least among all the available paths. The destination node, then simply swaps the path vector of the chosen route and after attaching the calculated \$ttl\$ value in the time field of the RREP packet forwards the RREP message and simultaneously set its flag bit.[7] When an intermediate node receives this RREP packet, it first sets its activity flag, i.e. A flag = 1 then after recording the information in its route cache, unicast this RREP packet to the next hop node defined in the path vector of the RREP packet. When the source node receives this unicast RREP packet, it comes to know that a route has been built and then first set its flag bit and then starts data transmission. When any of the intermediate relaying nodes moves away from the transmission range of its upstream neighbour due to mobility, its upstream neighbour informs the source node through the RERR message. The source node upon receiving a RERR message again floods the RREQ message to obtain the optimal route and process of route construction is repeated. The network load balancing is achieved through node and link disjoint path communication.

II. CONGESTION CALCULATION

The paths that are less congested can carry more packets. In MPR, the congestion values are computed through the measurements of queue data packets, number of packets dropped at each node, and number of Ready to Send (RTS) packet used to send data packets.[10][11] The following equation used to find the congestion metric:

$$\text{Avg_q_length} = 1 - \text{Avg_q_len} / (n * \text{q_length}) \quad (1)$$

$$\#\text{Drop} = 1 - \text{no_Drop_Packet} / \text{no_SentPkt} \quad (2)$$

$$\#\text{RTS} = 1 - \text{no_RTS_Ignore} / (7 * \text{no_SentPkt}) \quad (3)$$

Where n is the number of times queue length is calculated and 7 is the maximum retransmission.

$$\text{Node_congestion} = (\text{Avg_q_length} + \#\text{Drop} + \#\text{RTS})/3.0 \quad (4)$$

Whenever node sends the RREQ packet, each RREQ packet contains a congestion field to store the congestion attribute. Whenever the intermediate node receives the RREQ packet, it multiplies the congestion value in the RREQ packet using the individual node congestion value. The result is linear value and it is converted into non linear using the following equation:

$$\text{New Congestion} = 1 - (1 - \text{current congestion})^3 \quad (5)$$

The length effect the quality of the path, the shortest and least congested routes are compromised: Final QoS = (New Congestion)^L, where L is the length of the route. Final QoS is used at source nodes as a measure of the best path. The

higher the final QoS means the better in congestion and length.

III. CONGESTION CONTROL THROUGH LOAD BALANCING THODOLOGY

The congestion control methods can be a router centric or host/node centric. In existing congestion control methods, the source node is informed about the congestion in the network. Therefore, either it may slow down the packet transmission rate or find an alternate route which may not necessarily be an optimal route(Tom Goff et al. 2001). It must be pointed out that all the congestion control methods are able to inform the source about the congestion problem. It tries to balance the traffic load (Siuli Roy et al. 2003; Fengfu Zou et al 2007). The predominant example of end-to-end congestion control in use today that implemented using TCP. The essential strategy of TCP is to send the packets into the network without a reservation. The TCP assumes only First in First out (FIFO) queuing in the network routers, but also works with fair queuing.

3.1 Additive Increase/Multiplicative Decrease

The TCP maintains a new state variable for each connection, called congestion window to limit how much data is allowed to have in transit at a given time. The congestion window is congestion controls counterpart to flow control's advertised window. The TCP is modified such that the maximum number of bytes of unacknowledged data allowed, but it is the minimum of the congestion window and the advertised window.

$$\text{Max Window} = \text{MIN} (\text{Congestion Window}, \text{Advertised Window}) \quad (6)$$

$$\text{Effective Window} = \text{Max Window} - (\text{LastByteSent} - \text{LastByteAcked}). \quad (7)$$

That is, the max window replaces advertised window in the calculation of effective window. Thus, a TCP source is allowed to forward no faster than the slowest component of the network or the destination host can accommodate. The problem is that how the TCP comes to learn an appropriate value for congestion window. Unlike the advertised window which is forwarded from the receiving side of the connection, there is no one to forward a suitable congestion window to the sending side of TCP. The TCP does not wait for entire windows of ACKs and add one packet worth to the congestion window. The congestion window is incremented as follows, each time an ACK arrives:

$$\text{Increment} = \text{MSS} \times (\text{MSS}/\text{Congestion Window}) \quad (8)$$

$$\text{Congestion Window} += \text{Increment} \quad (9)$$

That is, the MSS bytes increments the congestion window. A fraction of MSS is increased for every time an ACK is received. The important concept to understand about AIMD is that the source is willing to reduce its congestion window at a much faster rate than it is willing to increase its congestion window.

3.2 Fast Retransmit and Fast Recovery

The mechanisms described to add the congestion control to the TCP. It is soon discovered, however, the coarse-grained implementation of TCP timeouts leads to long periods of time. Hence, a new mechanism called fast re-transmit is added to TCP. The fast retransmit is a heuristic that sometimes triggers the retransmission of a dropped packet sooner than the regular timeout mechanism.

3.3 Path Selection

The proposed algorithm may generate some paths with length longer than the used one and because of sharing channels between paths; the end to end delay is increased than DSR. Several criteria have been considered in selecting paths when attempting to forward the data packets. The first criteria in selecting path are node disjoint paths, second criteria is to calculate the correlation factor (CF) between paths.

3.4 Determination of node disjoint paths

The broadcasting of the RREQ is similar to the ordinary DSR, but instead of replying from the intermediate node cache, the RREQ is stored in the cache table when the request has been seen before. Otherwise, it broadcasts the packets. When a RREP is generated, the redirection field is set to true if the generated path is a disjoint path with respect to the stored paths in the destination node. Otherwise, the field is set to false. When the intermediate node receives the RREP packets, it redirects it to the next node in the RREP path if redirection field is set to true. Otherwise, it starts to search for the shortest path from the cache table whose first hop is different from the first hop of the RREP path.

3.5 Correlation Factor

Several criteria have been considered in selecting paths when attempting to send data. The first criteria in selecting path are node disjoint paths, second criteria is to calculate the CF between paths. The CF is defined as the number of links connecting two paths. In order to find the CF and the neighbour nodes for each node in each path, the attribution of minimum link determination relates to the rest of the paths. Figure 1 shows two paths with 7-related links.

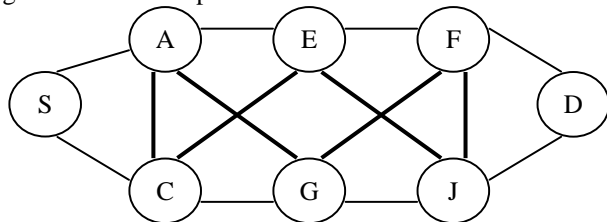


Figure 1 Two disjoint paths

The lower the CF is, the better the end to end delay is. The following equation is used to find the CF between two paths: If p_1, p_2 are two paths, then

$$CF = \text{No. of links between } p_1, p_2 / \text{hop Count } [p_1] * \text{hop Count } [p_2]. \quad (10)$$

The third criteria is select paths with length less than or equal to the previous communication paths. Also the difference between shortest path and alternative ones must be small. Hence, if the difference is large this leads to having many unordered packets received at destination and need time to order them which increase queuing delay. Before the source node starts to transmitting the data packet, it needs to apply the third criteria then apply the CF equation. The CF threshold value is used to find the final paths that are used to forward data packet are related to each other.

IV. PERFORMANCE EVALUATION

The performances of MPR are evaluated using ns2 simulator. The traffic pattern, the scenario description and the metrics are described in table 1, which are used for the experiments. The scenarios can also be exported for the network simulators ns-3, GloMoSim/QualNet, COOJA, MiXiM, and ONE.

Table 1 List of Simulation parameters for MPR Performance Evaluation

Simulation Parameter	Value
Simulator	NS-2
Node Movement Model	Random Waypoint
Speed	0-25m/s
Traffic Type	CBR
Bandwidth	2Mb/s
Transmission Range	250m
Buffer Size	64 Packets

4.1 Simulation Environment

The size of environment is 500 x 500 m², and every node moves at random as well as its position. Radio transmission range of the node is 250 m and its way of wireless communication is free space. In addition, MAC protocol is set to IEEE 802.11. The number of nodes is variable for different measurement, which is illustrated specially.

4.2 Mobility Pattern

The mobile movement is set as per random way point model. In the node mobility, the mobile nodes select the random way point to move, and a node stays its location for a pause time before the next move. The simulation is varied under different size and mobility model. The varied pause time of mobile nodes is 600 and 300 seconds and node velocity is 0-25 m/s.

4.3 Traffic Pattern

The data traffic is generated using CBR. The number of source and destination pairs is varied. The buffer size is 64. The transmission range of the communication node is 250 m.

V. SIMULATION RESULTS

The simulation results are illustrated in the aspects of PDF and AED. The effect on the PDF and AED is described.

5.1 Effect on packet delivery ratio

The figure 2 describes the packet delivery fraction for MPR and DSR. The packet delivery fraction is the fraction of the number of receiving data packets received to the number of forwarded packets. Packet delivery fraction reduces as the pause time decreases from 100 to 900m/s mobility. This is due to the mobility of the network and the probability of link failures increases as the mobility increases. It is observed that the MPR Protocol maintains a better packet delivery ratio than the existing DSR. The MPR selects the alternative path to the communication route without the congestion formation. Hence, the communication does not interrupt. It improves the packet delivery Ratio under a network with highly dynamic network. From the simulation results, the packet delivery fraction for MPR is 0.45 over high mobility, and low mobility, it is 0.89%. The packet delivery fraction for DSR is 0.13 over high mobility, and low mobility, it is 0.81%.

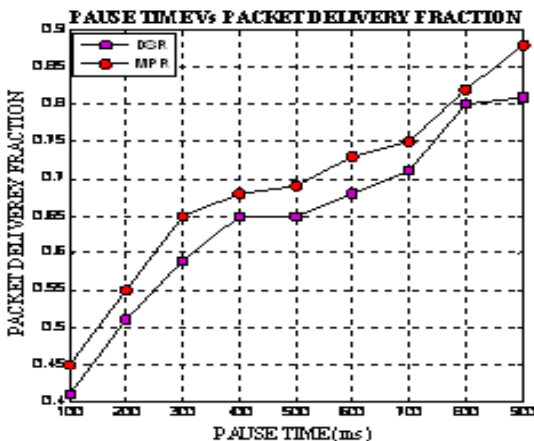


Figure 2 Pause Time Vs Packet Delivery Fraction

5.2 Effect on End-to-End delay

The figure 3 describes the end-to-end delay for DSR and MPR. The delay time is high for MPR. It consumes more time to rediscover the routes when the congestion is occurred. The increased number of nodes also increases the data delivery delay. The time taken to reach the destination from the source is defined as end-to-end delay. As the mobility increases, the probability of link failures also increased and hence the end-to-end delay increases.

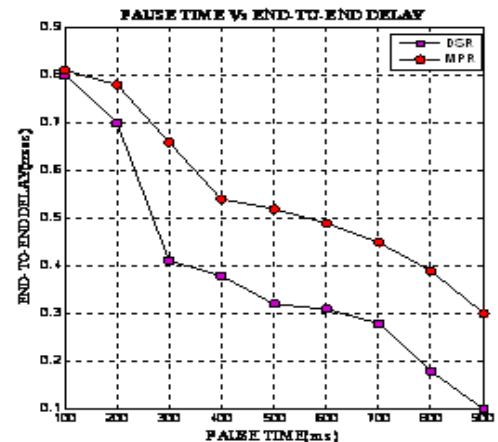


Figure 3 Pause Time Vs End-to-End Delay

In MPR, the data packets are delivered using alternative route when the congestion is occurred. However, the link failure of alternative routes incurs the data delay, but it is higher than the packet delay of DSR. From the simulation results, the end-to-end delay for DSR is 0.8 ms over high mobility and low mobility, it is 0.1 ms. The end-to-end delay for MPR is 0.8 ms over high mobility and for low mobility, it is 0.3 ms.

VI. CONCLUSION

The paper clearly explained the performance of multi point relays. Initially, the network congestion is explained. It clarified the performance of source, intermediate, and the destination node in the network when the data packets are transmitted through the communication path. It successfully calculated the node congestion. It provided the techniques of congestion minimization through the load balancing. The path selection for balanced load traffic is clearly explained. The determination of node disjoint paths and the correlation factor for intermediate nodes are calculated to minimize the congestion formation. Finally, the congestion minimized protocol MPR is proposed. The performance of the proposed MPR is compared with the existing DSR through the network simulation and it proved that the MPR provides better packet delivery ratio, but, the end-to-end delay of MPR is higher than the DSR.

REFERENCES

- [1] Elizabeth M Royer., and Chai-Keong Toh. A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks. IEEE Personal Communications, Apr. 2003, 46-55.
- [2] Sung-Ju Lee and Mario Gerla, "Split Multipath Routing With Maximally Disjoint Paths in Ad Hoc Networks", Proceedings of IEEE ICC'01, June 2001, vol 10, 3201-3205.
- [3] Ye, Z., Krishnamurthy, S.V., Tripathi, S.K.: A Framework for Reliable Routing in Mobile Ad Hoc Networks. IEEE INFOCOM 2003, vol 1, 270-280.

- [4] Shekhar H. M. P. and K.S Ramanatha. Mobile Agents based Congestion Aware Routing in Mobile Ad Hoc Networks. In Proceedings of the IEEE International Conference on "3G & Beyond", 3G 2005, November 2005), pp. 47-53.
- [5] L. Shrivastava, S. S. Bhadoria, and G. S. Tomar, "A load-balancing approach for congestion adaptivity in MANET," in Proceedings of the International Conference on Computational Intelligence and Communication Networks (CICN '11), pp. 32–36, October 2011.
- [6] L. Shrivastava, S. S. Bhadauria, and G. S. Tomar, "Influence of traffic load on the performance of AODV, DSR and DSDV in MANET," International Journal of Communication Systems and Network Technologies, vol. 1, no. 1, pp. 22–34, 2013.
- [7] B. Ramesh and D. Manjula, "Congestion adaptive AODV routing protocol for streaming video in mobile ad hoc networks," International Journal of Communications, Network and System Sciences, vol. 4, pp. 285–385, 2008.
- [8] A. Valarmathi and R. M. Chandrasekaran, "Congestion aware and adaptive dynamic source routing algorithm with load-balancing in MANETs," International Journal of Computer Applications, vol. 8, no. 5, article 1, 2010.
- [9] A. Gaba, S. Voulgaris, and M. van Steen, "Towards congestion-aware all-to-all information dissemination in mobile ad-hoc networks," in Proceedings of the 2010 IEEE Globecom Workshops (GC '10), pp. 1690–1695, Miami, Fla, USA, December 2010.
- [10] S. Soundararajan and R. S. Bhuvaneshwaran, "Adaptive multi-path routing for load balancing in mobile Ad Hoc networks," Journal of Computer Science, vol. 8, no. 5, pp. 648–655, 2012.
- [11] S. Jain, S. Kokate, P. Thakur, and S. Takalkar, "A study of congestion aware adaptive routing protocols in MANET," Computer Engineering and Intelligent Systems, vol. 3, no. 4, p. 64, 2012.