IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS), ISSN: 2249-9555
Vol. 2, No.6, December 2012

On Demand Node-Disjoint Multipath Routing Protocol with Low Overhead in Mobile Ad hoc Networks

Jayshree Tajne[1] (Student)
Department of Computer Science and Engineering
G.H. Raisoni college of Engineering
Nagpur university, Maharashtra, India
Jayshree_tajne@rediffmail.com

Veena Gulhane[2] (Asst. Prof.)
Department of Computer Science and Engineering
G.H. Raisoni college of Engineering,
Nagpur university, Maharashtra, India
veena.gulhane@raisoni.net

***Abstract*—Effective routing is critical in mobile ad hoc networks. Frequent link failures may occur in mobile ad hoc network because of dynamic topology and bandwidth constraints therefore, multipath routing is a very important research issue. In this paper, we propose the on demand node-disjoint multipath routing protocol finds all available node-disjoint routes from source to destination. The main goal is to discover multiple node disjoint paths with a low routing overhead during a route discovery, also improve the end-to-end delay and packet delivery ratio. Then the algorithm of analytical model is presented.**

***Keywords*- AODV, MANET, Multipath Routing, Node-disjoint**

I. INTRODUCTION

Mobile ad hoc networks are characterized by high node mobility, dynamic topology and low channel bandwidth. In these scenarios, it is essential to perform routing with maximal throughput and, at the same time, with minimal control overhead. In the networking research community, there is tremendous interest in MANETs Routing [1]. Routing protocols can be classified as either unipath or multipath based on the number of routes between the source-destination pair. Intuitively, network resources can be better utilized by multipath routing and multipath routing can offer performance improvements over unipath routing [2]. Now days there are many researches on multipath routing protocols for mobile ad hoc networks [3].

Multipath routing has been used as an attractive alternative for mobile ad hoc networking because it is able to provide the support for fault tolerance and load balancing. But the existing multipath routing protocols have some faults as larger routing overhead, less multipath route and more difficult in search for maximum irrelevant path. Node-disjoint multipath routing allows the establishment of multiple paths, each consisting of an unique set of nodes between a source and destination. We know that MANETs consist of mobile nodes that cause frequent link failures. This link failure causes two main problems. Firstly, when a route break occurs, all packets that have already been transmitted on that route are dropped and it decreasing the average packet delivery ratio (PDR). Secondly, the transmission of data traffic is halted for the time till a new route is discovered and it increasing the average end-to-end delay.

In this paper, we have designed a on demand node-disjoint multipath routing protocol. Our proposed approach minimizes the effect of link failure. Hence, the above mentioned two problems caused by frequent link failures are addressed. This protocol ensures that after a route is broken, the node can continuously send data without any delay, using one of the backup routes stored in its routing table during route discovery process.

The remainder of the paper is structured as follows. In Section II, related work is given by providing a brief description of existing multipath extensions of AODV routing protocol. The proposed protocol used for discovering multiple paths is presented in Section III. Finally, the conclusion is provided in Section IV.

II. RELATED WORK

In this section, we discuss the previous work done on multipath routing protocol. Multipath routing creates multiple paths between a source-destination pair. In case of the failure of first route, the backup routes are used for continues data transmission. In multipath routing protocols, the paths between a source and destination can be link-disjoint, node-disjoint or non-disjoint.

AOMDV [6] is a multipath extension of AODV [4] protocol to find out multiple disjoint loop-free paths between source and destination. It relies on the routing information already available in the AODV protocol, thereby limiting the control overhead incurred in discovering multiple paths. Long alternate paths are avoided by ignoring alternate paths that are more than one hop distance away. It has less number of route discoveries and more packet delivery ratio than AODV due to availability of alternate paths. The use of additional RREPs to form multiple forward paths to the destination increases the overhead for each route discovery but the overall overhead is less as compared to AODV due to less number of route discoveries. It does not provide scalability.

AODV-ABR [7] is an extension to AODV-BR which in turn is an extension to AODV. AODV-ABR tries to overcome the problems occurred in AODV-BR routing protocol. Overhearing of RREP and data packets makes the routing protocol more adaptable to changing topology without transmitting many extra control messages. Route maintenance is done by using a handshake process, which is accomplished by using two networks control signals: BRRQ and BRRP. The BRRP packet contains hop count field to solve the problem of choosing longer alternate path. Based on this hop count field, the node selects the shortest path available among the many alternate paths available. This can solve the congestion and collision problem occurring in AODV-BR. An aging technique is also going to be used for alternate route maintenance. AODV-ABR repairs the link failure by only using immediate neighbor nodes. So, it has less routing overhead and better throughput as compared to AODV.

SMORT [8] minimizes the route break recovery overhead by providing intermediate nodes on the primary path with multiple paths to the destination. It uses the idea of fail- safe multiple paths. There would be more fail safe paths as compared to node and link disjoint paths. Due to the usage of fail-safe paths, a link failure can be corrected at the intermediate node itself, thereby reducing the route recovery time and the number of route error packet transmissions. Fall safe multiple paths have higher fault tolerance to route breaks due to their higher availability

MP-AODV [9] discovers two routes for each pair of source-destination, a main route and a back-up route. Two RREQ messages are used to discover routes, each for one route. Whenever one route is broken, the other route is used for data packet transmission and a RREQ is flooded to maintain the broken route. This approach has two drawbacks: (1) MP-AODV contains higher overhead than the traditional AODV because it requires one RREQ flooding for one route and additional RREPs for node-disjoint paths and, (2) This approach is not able to find all the available node disjoint paths between a source-destination pair

NDM-AODV [10] finds all node-disjoint paths between source and destination also considers the residual energy of the nodes while selecting the routes. Multiple paths are created by using minimum routing overhead by making use of Destination Source Routing (DSR) protocol like source routing in route discovery process. Local connectivity is maintained by using Periodic Hello messages for all active routes during the route maintenance stage. The main disadvantage of this approach is that, as the size of the network increases, the size of the RREQ and RREP messages also increases because of the path accumulation function. Therefore, the size of routing table at destination node also increases due to the storage required to store multiple paths.

AODVM-PSE [11] presents multipath versions of AODV protocols, but the multiple paths identified in this approach are link-disjoint rather than node-disjoint. In this method, data transfer is started only after all multiple paths are discovered. Therefore there is an initial delay in data packet transmission. AODVM [12], AOMDV [13] presents multipath versions of AODV protocols. The multiple paths identified in these approaches are link-disjoint. The links which are created does not match with each other but may have node in common. Data transmission is started only after the discovery of all multiple paths therefore there is an initial delay in data packet transmission.

III. PROPOSED MULTIPATH ROUTING PROTOCOL

In this section, the proposed protocol is described. The main goal of this protocol is to find all available node-disjoint routes between a source and destination with minimum routing overhead. To achieve this goal, this protocol works in the following phases: (i) Route Discovery Phase, (ii) Route Selection Phase.

The MP-AODV uses multipath node-disjoint to reduce end to end delay, but it produces more control packets for finding two node-disjoint routes. In our protocol, we use only one route request packets for setup of multiple node-disjoint routes, which reduce control overhead in the network. In the other hand, this protocol setup backup route faster than MP-AODV, that reduces the end to end delay.

A. Route Discovery Phase

When a source node wants to transmit a data packet to destination, it checks its routing table for the next-hop towards the destination of the packet. If there is an active route entry for the destination in the routing table, then source forwards data packet to the next hop. Otherwise, the route discovery phase begins. In the route discovery phase, routes are discovered using two types of control messages: (i) Route request messages (RREQs) and (ii) Route reply messages (RREPs). The source node broadcasts the RREQ message into the network. Each intermediate node after receiving a RREQ packet, checks whether it is a fresh or a duplicate one by searching an entry in the Seen Table. Seen Table stores two entries (i.e. source IP address and RREQ broadcasting ID (bid_RREQ)) that uniquely identify a RREQ message in the network. If an entry of received RREQ message is present in the Seen Table, then it is considered as a duplicate RREQ message and discarded without broadcasting to its neighbour. Otherwise, the node creates an entry in the Seen Table and updates its routing table for forward path before broadcasting the RREQ message.

Source IP Address	Broadcasting ID	**Seen flag**
---	---	---

Fig. 1. Seen table structure

In this protocol, only the destination node can send RREPs upon reception of a RREQ message. The intermediate nodes are restricted to send RREPs even if they have an active route to destination. We have changed the data structure of seen table and RREP message as shown in figures 1, 2. This is done so as to get the node-disjoint routes. The destination node has to send a RREP message packet for each RREQ received, even if the RREQ message is a duplicate one. We change the data structure of both Seen Table and RREP message, by adding an extra field that works as a flag known as seenflag. At the beginning this flag is set to FALSE. The RREP messages initiated by destination node contain one extra field known as broadcast ID (bid_RREP).

Type	R	A	Reserved	Prefix size	Hop count
Destination IP Address					
Destination Sequence Number					
Source IP Address					
Source Sequence Number					
Broadcasting ID					

Fig. 2. RREP structure

The route discovery process is used to discover node disjoint paths. When a destination node receives a RREQ message packet, it creates the corresponding reply as RREP message. The destination node copies the bid_RREQ from the received RREQ message into the bid_RREP field of sent RREP message. This RREP is sent towards the source of the RREQ using the reverse route to construct the forward route. For every RREQ message received, the destination does the above mentioned process. When the RREP message has been received by the intermediate nodes in the reverse path, the intermediate nodes check the seenflag value in their Seen Table. If the seenflag is set to FLASE, this indicates that this is the first RREP message packet on the reverse path towards the source node. So, the intermediate node forwards the RREP towards the source and reset the value of seenflag. When the intermediate node receives a RREP message for the same RREQ message it got earlier, the node simply discards the RREP message on the basis of seenflag value. Due to this, the intermediate node's cannot take part in more than one route from the existing multiple routes.

Algorithm 1: Route discovery method when a node receives RREQ message

```
N = Node
S = Source Node
D = Destination Node
I = Intermediate Node
S_flag = FALSE //Initial value of seen flag seen table
X = FALSE
Count = 0
n_routes

if (S has data to send)
{
    if (S has route for D)
    {start data transmission
    }
else
  Initiate RREQ broadcasting
}
if (N receives a RREQ message)
{
    if (N = I or N=S)
    {
      X= check value of seen table for duplicate RREQs
      if (X==1)
        Discard RREQ without rebroadcasting
      else
        Rebroadcast RREQ
    }
  else
   {
    N is the destination
    Destination node unicast RREP on reverse route to
    create forward route
   }
}
```

Algorithm 2: Route discovery method when a node receives RREP message

```
if (N receives RREP)
{
  if ( N != S )
  {
    S_flag=check and return the value of seenflag from the
    seen table
    if (! S_flag )
    {
      Insert first route in routing table
      Reset the value of seenflag in seen table to detect
      duplicate RREPs
      Forward RREP to next hop towards source
    else
      Drop the duplicate RREP
    }
  }
  else
  {
      Count=count the numbers of active routes for
      destination in routing table
      if (Count < n_routes)
        Insert secondary routes and sort them in ascending
         hop count
      else
        Drop the RREP message
  }
}
```

A node follows the procedure as shown in algorithm 1 after getting a RREQ message. When a source node wants to send a data packet to destination, it checks its routing table for any active route available for destination. If an active route is available, data packet is forwarded to the next hop towards its destination. Else, it creates a RREQ message packet and inserts the entry in seen table about the request packet. The re-sending of RREQ messages is avoided by using the above process. Each node makes updating in its Seen Table to avoid duplicate broadcasting of the RREQ message. When a node receives RREQ message, the algorithm checks the node whether it is a source, intermediate or destination node. If the node is a source or intermediate node, then RREQ message is processed same as is done in the traditional AODV protocol. A destination node creates a RREP message after receiving the RREQ and copies the *bid*_RREQ value from RREQ into the extra field provided in RREP. Destination node does not check whether the received RREQ message is fresh or duplicate as is done in traditional AODV protocol. It replies to every received RREQ message to establish multiple paths.

In discovery process, when a node receives a RREP message, Algorithm 2 is used to discover multiple node-disjoint routes. The RREP message is received by the node then the algorithm checks whether the node is an intermediate or source node. If the node is an intermediate node then its seenflag is checked from its Seen Table. In a seen table if seenflag is FALSE then this indicates that it is the first RREP message for this particular source-destination pair. The algorithm resets the value of seenflag for particular node corresponding to the source-destination pair and the primary route for the destination node is inserted. The RREP message is then forwarded to the next hop towards source. The duplicate RREP message is discarded. This ensures that all the discovered routes are node-disjoint. If the RREP message is received by the source node then the discovered node-disjoint path is inserted as primary or secondary, based on the value of seenflag and the number of routes already present for this destination in routing table.

B. Route Selection Process and Data Packet Transmission

If the source node has data packets to send and there is no route available in routing table, the node starts the route discovery phase. The data packet transmission is started as soon as it gets the first route for destination node known as primary route. All the other node-disjoint routes that are discovered will be stored in the routing table as secondary routes. After storing the primary route and specified number of secondary routes in the routing table, all the other routes are not stored. All the other routes that have lower hop count for destination as compared to existing secondary paths can replace the existing ones. The route selection phase works in such a way that whenever a route is required for data packet transmission, it always selects the primary route if it is available in routing table. If the primary route is not active, then the route selection process selects the route with lowest hop count from the available secondary routes in the routing table.

IV. CONCLUSION

In this paper, we propose on demand node-disjoint multipath routing with low overhead. There is a frequent link failure in mobile ad hoc networks because of its features like dynamic topology and resource constraints. Our motivation is to provide the improvement of on-demand multipath routing method in terms of packet delivery ratio, average end-to-end delay, and control overhead. This routing establishes the first route and then the data packet transmission starts immediately. The multiple routes are discovered in this routing are node-disjoint so that each path consisting of a unique set of nodes between source and destination. In the future work, we will compare the proposed protocol with the existing node-disjoint multipath routing protocols.

REFERENCES

[1] M. Abolhasan, T. Wysocki, E. Dutkiewicz, “A review of routing Protocol for mobile ad hoc networks, Ad Hoc Networks,” Vol. 2, No. 1, pp.1-22, 2004.

[2] Shafqat Ur Rehman1, Wang-Cheol Song2 and Gyung-Leen Park. “Associativity-Based On-Demand Multi-Path Routing In Mobile Ad Hoc Networks”. KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS VOL. 3, NO. 5, pp.475-496, October 2009.

[3] S. Mueller, D. Ghosal, “Multipath Routing in Mobile Ad Hoc Networks: Issues and Challenges”, Lecture Notes in Computer Science, Springer, Berlin, pp. 209-2342004.

[4] Charles E. Perkins and Elizabeth M. Royer. Ad hoc on-demand distance vector routing. In Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, pp. 90–100, 1999.

[5] David B. Johnson, David A. Maltz and Yih-Chun Hu, The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks (DSR), INTERNET-DRAFT, 15 April 2003.

[6] Mahesh K. Marina and Samir R. Das, "Ad Hoc On-Demand Multipath Distance Vector Routing," Wiley InterScience, Wireless Communications and Mobile Computing, vol. 6, pp.969-988, 2006.

[7] Wei Kuang Lai, Sheng-Yu Hsiao, and Yuh-Chung Lin, "Adaptive Backup Routing for Ad-Hoc Networks," Elsevier, Computer Communications, vol. 30, pp. 453-463, 2007.

[8] L.Reddeppa Reddy and S.V. Raghavan, "SMORT: Scalable Multipath On-Demand Routing for Mobile Ad Hoc Networks," Elsevier, Ad Hoc Networks, vol. 5, pp. 162-188, 2007.

[9] Chang-Woo Ahn, Sang-Hwa Chung, Tae-Hun Kim, and Su-Young Kang,” A Node-Disjoint Multipath Routing Protocol Based on AODV in Mobile Ad hoc Networks”, Information Technology: New Generations (ITNG), 2010 Seventh International Conference on, pages 828 –833, April 2010.

[10] Shunli Ding and Liping Liu, “A Node-Disjoint Multipath Routing Protocol Based on AODV”, Distributed Computing and Applications to Business Engineering and Science (DCABES), 2010 Ninth International Symposium on, pages 312 –316, aug. 2010.

[11] Fubao Yang and Baolin Sun, “Ad hoc On-demand Distance Vector Multipath Routing Protocol with Path Selection Entropy”, Consumer

Electronics, Communications and Networks (CECNet), 2011 International Conference on, pages 4715 –4718, april 2011.

[12] Tsung-Chuan Huang, Sheng-Yu Huang and Lung Tang, "AODV-Based Backup Routing Scheme in Mobile Ad Hoc Networks", International Conference on Communications and Mobile Computing, 2010.

[13] YuHua Yuan, HuiMin Chen, and Min Jia., "An Optimized Ad-hoc On Demand Multipath Distance Vector (AOMDV) routing protocol", Communications, 2005 Asia-Pacific Conference on, pages 569 –573, oct. 2005.