Volume 6, Issue 7, July 2016 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Design a Token Based On-demand Hybrid Multipath Routing Algorithm for Mobile Ad- hoc Network

Rohan Verma* — Deptt. of Information Technology, H.N.B Garhwal Central University, Uttarakhand, India

Manish Deoli — Deptt. of Information Technology, H.N.B Garhwal Central University, Uttarakhand, India

Himanshu Suyal — Deptt. of Computer Science, H.N.B Garhwal Central University, Uttarakhand, India

Abstract— In Mobile Ad Hoc Network (MANET) it's a challenging to design an efficient routing protocol, due to the dynamic network topology and resource constraint. AODV and DSR are most widely studied on-demand ad hoc routing protocols because of their low overheads. This paper explores the network redundancy through multipath routing. The paper is based on Token On-Demand Multipath Routing Protocol .The main purpose of the project is to reduce flooding in network. The path is only set between the nodes which have token. The algorithm is to overcome the disadvantages of AODV and DSR algorithm. TOHMR algorithm is used to reduce the flooding and network traffic.

Keywords— TOHMR, On-demand Routing, MANET, Node Disjointed, Braided multipath

I. INTRODUCTION

A Self-determining system of mobile nodes and related hosts link by the wireless routes is termed as mobile ad hoc networks (MANET). It does not contain any base stations. Alternatively, a function of routing is included in every mobile nodes and multi-hops probably will be essential to permit one node to interact with another node over the ad hoc network owing to the restricted transmission range. There is a possibility that wireless topology of the network may get varied randomly and quickly as the routes travel arbitrarily and systemize themselves randomly. Hence, MANETs are illustrated as active, multi-hop and continuously modifying topology [1].

In most MANETs, multipath protocols are needed to facilitate efficient connectivity between transmitters that are not necessarily within each other's wireless range. MANET routing protocols are divided into the following categories:

- Flat Routing Protocols
    - Proactive Routing (Table-Driven)
    - Reactive Routing (On-Demand)
    - Hybrid Routing (blend of Reactive and Proactive)
- Hierarchical (Zone/Cluster-Based) Routing Protocols
- Geographic Position Assisted Routing Protocols

II. MULTIPATH ROUTING PROTOCOLS

Most routing protocols maintain routing tables to store the next hop towards the desired destination. Many routing protocols preserve a caching mechanism by which multiple routing paths to the same destination are stored. Multipath routing is essential for load balancing and offering quality of service. Other benefits of multipath routing include [2]. the reduction of computing time that routers' CPUs require, high resilience to path breaks, high call acceptance ratio (in voice applications) and better security. Special attention should be given to transport layer protocols as duplicate acknowledgments (DUPACKs) could occur, which might lead to excessive power consumption and congestion.

Multipath routing in Reactive Protocols On-demand routing protocols are inherently attractive for multipath routing, because of faster and more efficient recovery from route failures. MSR "Multipath Source Routing Protocol"[3]

Multipath Routing in Proactive Protocols Proactive routing algorithms, such as DSDV "Destination Sequenced Distance-Vector Routing" [4], maintain route updates among all nodes all the time. In fact, many proactive protocols tend to offer shortest path to each destinations. This is done by continuously monitoring the network topology. Unlike reactive routing algorithms, proactive routing protocols are capable of repairing broken routes in a short time. This is done by collecting network topology continuously.

Multipath Routing in Hybrid Protocols- Hybrid routing protocols incorporate the merits of both on-demand and proactive routing protocols. An example of this category is Zone Routing Protocol "ZRP", which is similar to a cluster with the exception that each node acts as a cluster head and a member of other clusters. The routing zone forms a few mobile ad hoc nodes within one, two or more hops away where the central node is located. The fact that both reactive and proactive schemes are found in the functionality of hybrid routing protocols, better performance is expected. However, due to hierarchical nature of the schemes more memory will be required compared to the identical reactive or proactive scheme [5]

III. DISJOINTS PATHS IN MULTIPATH ROUTING PROTOCOLS

Multiple paths can also provide load balancing and route failure protection by distributing traffic among a set of disjoint paths. Paths can be disjoint in two ways: (a) link-disjoint and (b) node-disjoint. Node-disjoint paths do not have any nodes in common, except the source and destination, hence the do not have any links in common. Link-disjoint paths, in contrast, do not have any links in common. They may, however, have one or more common nodes. In order to use multiple paths simultaneously they need to be as independent as possible. So not only do they need to be disjoint, also route coupling must be taken into account, because routes can interfere with each other. Route coupling takes place when a path crosses the radio coverage area of another path. There is a protocol that uses this property of radio broadcast to create backup-routes, but in the case of multiple-path data transport route coupling is unwanted. Routes may be link- or even node-disjoint but still interfere with each other due to route coupling. Consider the node-disjoint routes of figure1 again. In the situation of figure 3, when node a for example sends data to node b (both route1), node d on the other route cannot transmit data to e on route 2, since the nodes (and thus routes) are in each other‟s radio coverage area and interfere with each other. Since none of the routing protocols take the route coupling into account, we will ignore it in the sequel. Disjointness will be the only measure used for path independence

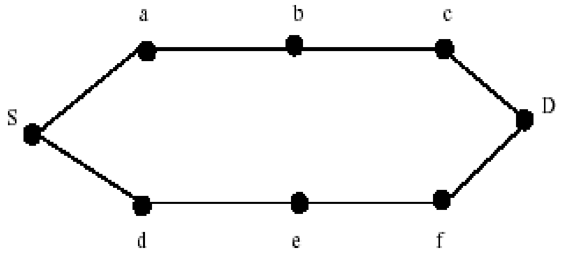

Fig. 1 Two node-disjoint paths from source S to destination D.

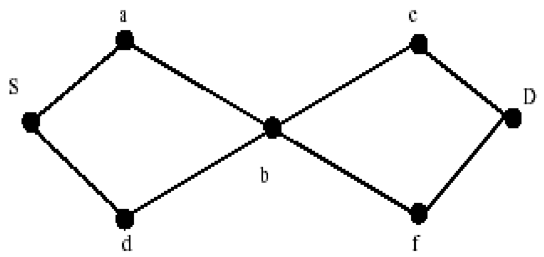

Fig. 2: Two link-disjoint paths from source S to destination D. Note that they are not node-disjoint, since they share node b.

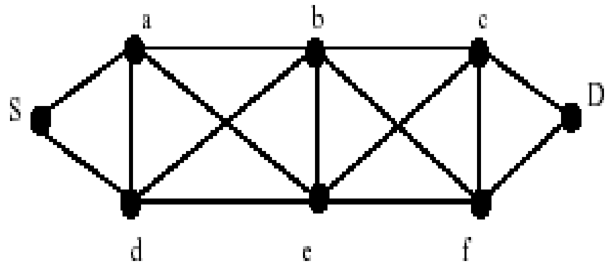

Fig 3:The two node-disjoint paths from fig 1, when they are in each other‟s radio coverage.

IV. RELATED WORK

OHMR to identify multiple paths during route discovery.OHMR is designed is primarily for highly dynamic ad hoc networks in which link failures or node failures occur on a frequent basis.When a single path routing protocol AODV is used in such networks,a new discovery must be launched each time when a route breaks.each route discovery induces high overheads and latency.this inefficiency can be avoided by maintaining multiple redundant paths such that the new route discovery is initiated only when all of the paths of destination are broken.OHMR searches for hybrid multipaths comprising both node disjointed and braided multipaths.

- THE OHMR protocol searches for node disjoint and braided routes using a single flooded query in order to provide sufficient redundancy with minimum overheads.
- This algorithm has two parts:
- First part-when a node receives a RREQ packet
- Second part-when a node receives a RREP packet

FIRST PART: Received RREQ packet

Step 1 : if(the packet is received for first time)

- Set up the first reverse path to the source using the previous hop of the packet as the next hop in the reverse route table.
- If(the node is destination)Generate a RREP & initialise its flag to “FALSE” and send the RREP to the next hop in the first reverse path.
- c) else broadcast the packet to the neighbouring nodes.

Step 2 : else if (the packet is received for the second time)

- set up the second reversed path to the source in the reverse route table.
- b) if the node is destinationgenerate a RREP &initialise the flag to “true” and send the RREP to the next hop in the second reverse path.

Step 3 : else discard the packet.

SECOND PART: RECEIVED RREP PACKET

Step 1 : if(the node is the source)
Set up the forwarding path to the destination using the previous hop of the packet as the next hop in its routing table.

Step 2 : else if (the packet is received for the first time)
Save the packet's flag to the alternate path field in the routing table and send it to the next hop in the reversed path.

Step 3 : else if (the packet's field is TRUE and alternate field is FALSE)
Reset he packet's flag to false and return it to previous hop.

Step 4 : else if (the packet 's field is FALSE and alternate field is TRUE)
Reset he packet's flag to false and return it to next hop in reverse path.

Step 5 : else discard the packet.

V. TOHMR ALGORITHM

Requesting and releasing the CS:

When node i requests access to the CS, it enqueues its own identifier on Q and sets status to WAITING. If node i does not currently hold the token and i has a single element on its queue, it calls ForwardRequest() to send a Request message. If node i does hold the token, i can set status to CRITICAL and enter the CS, since it will be at the head of Q. When node i releases the CS, it calls GiveTokenToNext() to send a Token message if Q is non-empty,and sets status to REMAINDER.

- Status: Indicates whether node is in the WAITING, CRITICAL, or REMAINDER section. Initially, status = REMAINDER.
- N: The set of all nodes in direct wireless contact with node i.Initially, N containsallofnode i's neighbours.
- MyHeight:A three-tuple (h1,h2,i) representing the height of node i.Links are considered to be directed from nodes with higher height toward nodes with lower height, based on lexicographic ordering.
- height[j]: An array of tuples representing node i's view of myHeightj for all j $\in$N(i). Initially, height[j] = myHeight(j), for all j $\in$ N(i). In node i's viewpoint, if j $\in$ N, then the link between i and j is incoming to node i if height[j] > myHeight, and outgoing from node i if height[j] < myHeight.
- tokenHolder: Flag set to true if node holds token and set to false otherwise. Initially, tokenHolder = true if i = 0, and tokenHolder = false otherwise.
- next: When node i holds the token, next = i, otherwise next is the node on an outgoing link. Initially, next = 0 if i= 0, and next is an outgoing neighbour otherwise
- Queue: Queue containing identifiers of requesting neighbours. Operations on Q include Enqueue(), which enqueues an item only if it is not already on Q, Dequeue() with the usual FIFO semantics, and Delete(), which removes a specified item from Q, regardless of its location
- receivedLI[j]: Boolean array indicating whether LinkInfo message has been received from node j, to which a Token message was recently sent. Any height information received at node i from a node j for which receivedLI[j] is false will not be recorded in height[j]. Initially, receivedLIi[j] = true for all j $\in$ N(i).
- Forming [j]: Boolean array set to true when link to node j has been detected as forming and reset to false when first LinkInfo message arrives from node j. Initially, forming i[j] = false for all j $\in$ N(i).
- formHeight[j]: An array of tuples storing value of myHeight when new link to j first detected. Initially, formHeighti[j]=myHeight i for all j $\in$ N(i)

When node i requests access to the CS:

- status := WAITING
- Enqueue(Q; i)
- if (not tokenHolder)
- if (|Q| = 1) ForwardRequest()
- else GiveTokenToNext()

When node i releases the CS:

- if (|Q| > 0) GiveTokenToNext()
- status := REMAINDER

When Request(h) received at node i from node j:

// h denotes j's height when message was sent

- if (receivedLI[j])
- height[j] := h
  // set i's view of j's height
- if (myHeight < height[j]) Enqueue(Q,j)

```
•  if (tokenHolder)
•  if ((|Q| > 0) and (status = REMAINDER))
•  GiveTokenToNext()
•  else // not tokenHolder
•  if (myHeight < height[k], $\forall$ k $\in$ N)
•  RaiseHeight()
•  else if ((Q = [j]) or ((|Q| > 0)
   and  (myHeight < height[next])))
•  ForwardRequest()      // reroute request
```

When Token(h) received at node i from node j: // h denotes j's height when message was sent

```
•  tokenHolder := true
•  height[j] := h
•  Send  LinkInfo(h.h1, h.h2  -1,i)
   to all outgoing k $\in$ N and to j
•  myHeight.h1 := h.h1
•  myHeight.h2 := h.h2 – 1   // lower my height
•  if (|Q| > 0)   GiveTokenToNext()
•  else next := i
```

When failure of link to j detected at node i:

```
•  N := N – {j}
•  Delete(Q,j)
•  receivedLI[j] := true
•  if (not  tokenHolder)
•  if (myHeight < height[k],  $\forall$ k $\in$ N)
•  RaiseHeight()        // reroute request
•  else if ((|Q|> 0) and (next $\neq$ N))
•  ForwardRequest()
```

When formation of link to j detected at node i:

```
•  Send LinkInfo(myHeight) to j
•  forming[j] := true
•  formHeight[j] := myHeight
```

Procedure ForwardRequest():

```
•  next := l $\in$ N  :  height[l] $\leq$ height[j],  $\forall$ j $\in$ N
•  Send Request(myHeight) to next
```

Procedure GiveTokenToNext():

```
•  next := Dequeue(Q)
•  if (next $\neq$ i)
•  tokenHolder := false
•  height[next]:=(myHeight.h1,myHeight.h-1,next)
•  receivedLI[next] := false
•  Send Token(myHeight) to next
•  if (|Q | > 0) Send Request(myHeight) to next
•   else    // next = i
•  status := CRITICAL
•  Enter CS
```

Procedure RaiseHeight():

```
•  myHeight.h1:=1+ min$_{(k \in N)}${height[k].h1}
•  S := {l $\in$ N  : height[l].h1=myHeight.h1}
•  if  (S $\neq \emptyset$)  myHeight.h2 := min(l $\in$ S) {height[l].h2} -1
```

- Send LinkInfo(myHeight) to all k ∈N // raising own height can cause links to be outgoing
- for (all k ∈N such that myHeight > height[k])
- Delete(Q,k)
  // reroute request if queue non-empty,
  // just had no outgoing links
- if (|Q| > 0) ForwardRequest()

VI. CONCLUSIONS

In this paper, we have proposed TOHMR algorithm pursuits for node –disjointed and braided routes using a token based query in order to provide adequate redundancy with low overhead. The power feeding by the alternative route of the braided multipath as related to primary path. This algorithm is to overcome the disadvantages of adaptive multi-path routing protocol. Token is allocated to reduce the flooding and network. When the source node wants to forward the data packet to the destination, it utilizes the reactive route discovery technique where the multiple paths are conventional using multi-path TOHMR algorithm. The alternative path of the node-disjointed multipath are not affected by the node failure on the primary path .TOHMR approach is a key reduction in the frequency of route discovery flooding in on-demand hybrid multipath routing protocols.

REFERENCES

[1] J.Premalatha and P. Bala Subramanie, "*Enhancing the quality of service in MANETs by effective routing*", ICWCSC, 2010.

[2] C. Siva Ram Murthy and B. S. Manoj, "*Ad Hoc Wireless Networks Architecture and Protocols*", Prentice Hall 2004

[3] L. Wang, L. Zhang, Y. Shu and M. Dong, "*Multipath Source Routing in Wireless Ad Hoc Networks*", CCECE 2000, 7-10 March 2000

[4] C.Perkins and P.Bhagwat, "*Highly Dynamic Destination-Sequenced Distance-Vector Routing for Mobile Computers*", In Proceedings of the Symposium on Communication Architectures and Protocols, ACM SIGCOMM, 1994

[5] S. Buruhanudeen, "*Overview of Ad Hoc Routing Protocols*", University of Bradford, UK, Nov. 2002

[6] P. Rama Devi and Dr. D. Srinivasa Rao," *Congestion Adaptive Hybrid Multi-path Routing Protocol for Load Balancing in Mobile Ad Hoc Networks*" ijcst, vol3,issue12 ,2012

[7] S. Taneja and A. Kush, " *A Survey of Routing Protocols in Mobile Ad-Hoc Networks*", International Journal of Innovation, Management and Technology, Vol. 1, No. 3, 279-285, August 2010

[8] Gary Breed Editorial Director, "*Wireless Ad-Hoc Networks: Basic Concepts*", High Frequency Electronics, March 2007.

[9] H. Deng, W. Li, and D.P. Agrawal, *"Routing Security in Wireless Ad Hoc Networks"* IEEE Communications Magazine • October 2002

[10] S.Mohseni.; Hassan, R.; Patel, A.; Razali, R, *"Comparative review study of reactive and proactive routing protocols in MANETs"*, 4th IEEE International Conference on Digital Ecosystems and Technologies, 304-309, 2010.

[11] H. Bakht, " *Survey of Routing Protocols for Mobile Ad-hoc Network*", International Journal of Information and Communication Technology Research, 258-270, October 2011

[12] C. W. Ahn, and R. S. Ramakrishna, "A "Genetic algorithm for Shortest Path Routing Problem and the Sizing of Populations" *EEE Transactions on Evolutionary computation, vol.6, no.6,* Dec. 2002

[13] M.Abolhasan ,T.Wyosocki ,E. Dutkiewicz , " *A review of routing protocols for mobile ad hoc networks*," Journal of ad hoc networks 1: pp. 1- 22, 2004

[14] C.C.Sue,C.Y.Hsu,Y.C.lin,"*Design and analysis of hybrid on-demand multipath routing protocol with multimedia application on MANETs*"10th Asia pacific network operation & management symposium,2007,japan,oct10-12,2007

[15] R. Ogier, F. Templin and M. Lewis, "*Topology Dissemination Based on Reverse-Path Forwarding (TBRPF)",* IETF Request for Comments (RFC) 3684, 2004

[16] C. Perkins, S. Ratliff S and J. Dowdell, "*Dynamic MANET On-demand (AODVv2) Routing*", IETF Internet-Draft draft-ietf-manet-dymo-26, 2013

[17] M. U. Farooq, N. Tapus,"*Performance comparison of on-demand routing protocols for Mobile Ad hoc Networks*", Proc. 12th RoEduNet International Conference Networking in Education and Research, 2013 pp. 1-5