EFFICIENT MULTI HOP ROUTING ALGORITHM FOR BLUETOOTH DEVICES

Ganesh Gupta¹, Shubham Gupta², Manish Kumar Singh³, Durbadal Chattaraj⁴

¹,²,³,⁴Department of Computer Science & Engineering,
Narula Institute of Technology, Agarpara, Kolkata, West Bengal, India
¹gupta.ganeshnit@gmail.com, ²shubhamnit@gmail.com, ³manishkumarsinghnit@gmail.com, ⁴durbadal.chattaraj@gmail.com

Abstract
On wireless ad hoc network (AHN), ad-hoc mode is a method for wireless devices to directly communicate with each other. Since AHNs are dynamic in nature, they require a dynamic routing protocol to send the message from one source node to destination. Several approaches have been proposed for designing multihop routing protocols in wireless AHNs. Flooding is one of them to discover a dynamic routing path. But Flooding of request packet in the route in wireless AHN creates a huge amount of traffic which leads to high probability of packet collisions. Also it causes significant over head in delivering of packet and hence is inappropriate for the dynamic routing. It also causes more time consumption.

In this paper we propose a efficient multi hop routing algorithm to deliver the message using the cache variable for intermediate to reduce a large number of path & hence reducing the traffic generated by normal flooding. We also apply a logic to share the dynamic routing table between different devices in neighborhood. We also propose a scheme at receiver side to receive only one valid message from the flooded message and discard all other flooded messages.

I. INTRODUCTION
Wireless Ad-hoc Networks (AHNs) is a set of wireless independent multihop nodes which does not require any preexisting infrastructure.
This network directly connects one wireless device to another without using any router or hubs. Instead, each node take part actively in routing by forwarding packet to other nodes, and so the determination forwarded packet is made dynamically based on the network connectivity.
So, There is no concept of a particular server, any node can be a server or client. It allows all wireless devices to discover and communicate in peer-to-peer fashion with other who are present within the range or outside the range (communication is done in multi hop fashion) without involving any central access points (wireless routers or hubs). Some of the nodes in ad-hoc network may not be able to communicate directly with each other and dependent on some other nodes to pass their message. Such networks are often known as multi-hop or store and forward networks. The intermediate node act as routers, which discover and maintain a table to forward the message to other nodes in the networks.
Ad hoc networks have played an important role in following
1) **Military Environment**: In case of battlefield adhoc network can be formed by tanks and planes to communicate with each other.

2) **Emergency Operations**: In case of rescue working operating in disaster area can form adhoc network to communicate with each other.

3) **Personal Area Networking**: Adhoc network can formed by Cell phone, laptop, wrist watch to form a PAN

4) **Education**: Used in Virtual classrooms, conferences

---

**Fundamental Protocol in Ad-hoc Networks**

Since Wireless Ad-hoc Networks are a new scheme in the field of networking and it is different from the wired networks. Thus the main issue in AHN is *self-organization* and *wireless* transport of information [4], [5]. Since the nodes in a Wireless Ad-hoc Network are mobiles in nature so they are free to move arbitrarily at any time at any position which results in dynamic and unpredictable topology. This makes routing process difficult because the topology can be changed any time. Traditional routing algorithm like link state routing ad distance vector routing is not suitable for adhoc networking due to the reasons mentioned before. So, proper design of the ad-hoc routing protocol is needed to overcome the problem. Several protocols have been proposed for routing in ad hoc networks [3–15, 18-22, 25]. These routing protocols can be classified as basically of following types-

1) **Proactive algorithm**
2) **Reactive algorithm**
3) **Hybrid algorithm**

1) **Proactive Algorithm**

Proactive algorithm maintains routing tables that contains the lists of destinations and their corresponding routes. The tables are updated periodically by sharing of tables with the adjacent node. The disadvantage of this algorithm is that, it requires large amount of data for maintenance and it has slow response to node failures and change topology.
2) **Reactive Algorithm**

This type of protocol is also known as on-Demand routing. Here the term on “on demand” means that the sender node will try to find the route of destination only if it has to send some data. The node will maintain the route as long as it is needed by it. Therefore, these protocols require less control overheads. The routing is based on the shortest path algorithm to determine the host.

3) **Hybrid (both pro-active and reactive) routing**

Hybrid protocol is combination of both reactive protocol and proactive protocol. Here each node first maintains the routing tables as in case of proactive protocol. The node will also participate in handling the demand of routing that comes from nodes through reactive protocol.

II. RELATED WORK

Flooding is a method of broadcasting the packets. Here a node first broadcast packet to all its neighbor nodes. The receiver node will again forward each incoming packet to its neighbor except the one from which the packet come from, until packet reaches the destination.

Several authors have proposed to use the basic ideas of flooding. A reliable broadcast protocol was proposed by Pagani [19] for networks that have unpredictable dynamic topology. It gives a routing algorithm that ranges between flooding and the traditional routing protocols.

A Controlled Flooding protocol was proposed by Lesser and Rom [16]. In this, a message is limited broadcast on the basis of flooding mechanism ie not throughout the network. Traffic is further limited by assigning a cost to each link and a wealth to each message. A message is sent on a link on the basis of priority ie low wealth packet will be sent upon receiving a message; an intermediate node subtracts the cost of the link from the message wealth.

We propose a multipath routing protocol known as Efficient Multi hop Routing. The protocol is proactive. It uses the basic features of flooding, but restricts packet propagation by using a cache variable which store packet id.

It restricts the transmission of duplicate packet having same id that it has received early. Here the intermediate node will also not transmit packet to those node from which it comes from starting from the sender. Hence flooding becomes optimized as it uses adaptive mechanisms for restricting flooding in the network.

III. **THE EFFICIENT ROUTING PROTOCOL: OVERVIEW & OUR SCHEME**

The traditional algorithm is suitable where rate of topological changes is low but it fails when high.

So when nodes changes its position, the best way is to flood packets in the network [13].so that at least one packet can be reached.

We proposed an algorithm which is based on optimized flooding. In this scheme we try to deliver packet to the sender at the most appropriate path. We assume each node to be host and receiver and
each maintains a routing table to route the packet. The Bluetooth address is used to uniquely identify each node in the network. In the routing table for every entry of a node, there is a list of nodes from which a particular node can reach the sender who wishes to send a message. First, check whether the destination exits in its coverage area or not. If it exists, then it will send the packet directly to it, otherwise, it finds the list of adjacent nodes from which the destination can be reached and broadcasts the message to all of them. Here the contains are id, destination name and the routing path. At the intermediate node, it first checks whether its cache contains the packet id or not. If it contains, then it simply discards it; otherwise, it will first find all the list of nodes to which the destination can be reached and broadcast all of them except the nodes which are already in the routing path of the packet. Secondly, it will store the packet id to its cache so that any future if it receives any packet with the same id, it will discard it. This process continues until the message is delivered to its destination. The receiver receives the packet and sends the acknowledged directly to the path from which the packet is received. We assume that each node has a routing table which can be updated dynamically, initially when a node starts it first searches for neighbors and adds them to its routing table. After that, it will broadcast its table to its neighbors. On receiving the broadcast message, the neighbor will update its table as given by the following example.

**Diagram of table 1**

Now suppose that two new nodes D, E are introduced into the network on which node D is within the range of the pre-existing network, and E is out of range. Now both devices start searching for neighbors and add them to their routing table. After that, it will broadcast its table to its neighbors. On receiving the broadcast message, the neighbors will update their tables as given by the following example.

**Diagram 2**

Now both start sharing the table. After that, the table will be shown in the following diagram.

**Diagram 3**

Now after a fixed time interval, each node will again search for devices and update their table with their neighbors. This will lead to dynamic entry in each routing table of nodes.

**Packet format**

Each node in the ad hoc network communicates with each other by exchange of messages. The messages consist of two parts: the header portion and the body portion.

The format of the message is shown below:

<table>
<thead>
<tr>
<th>Header</th>
<th>Body</th>
</tr>
</thead>
</table>
Message-id = this id is generated at the sender side to identify each packet uniquely.

Destination = Bluetooth address of the destination node

Path = this contains the routing path through which packet is coming the last node in the path will be the sender

Hop Count = contains the maximum number of hop the packet can travel it is used to avoid

Type: This define the type of message sent

Type 1: indicates the message contains sharing routing table

Type 2: indicate the normal message

Example

<table>
<thead>
<tr>
<th>Message-id</th>
<th>Type</th>
<th>Destination</th>
<th>path</th>
<th>Hop Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>1559</td>
<td>1</td>
<td>D</td>
<td>A,S,E</td>
<td>10</td>
</tr>
</tbody>
</table>

We have path A-S-E, this means that message comes from E, the sender via S-A to receiver D

With type 1 i.e. it contains the share routing table of E also it has unique id 1559

Following format

<table>
<thead>
<tr>
<th>Node</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>XX</td>
<td>Add_1</td>
</tr>
<tr>
<td>YY</td>
<td>Add_2,Add_1,Add_3</td>
</tr>
<tr>
<td>ZZ</td>
<td>Add_3,Add_2</td>
</tr>
</tbody>
</table>
Now suppose the sender S wants to send message to D it finds that it can go from D via A and B having A has highest priority.

IV. ALGORITHM

Routing in the Bluetooth Ad-hoc network is based on limited flooding. In this type of routing the sender first checks the entries in the routing table the message to all the adjacent node from which it has to send the message. The intermediate node will accept the message IF it is the destination of the message otherwise it will flood the message to all the adjacent node from which destination can be reached except to those nodes from which the message has been reached to it. The intermediate node also note down the messageID in its cache so that IF in future any message comes with that particular id could be neglected

Sender Side

Suppose sender S wants to send the message to D

Step 1: find the list of addresses based on hop count which D can be reached in the routing table

Step 2: three cases arises

Case 1: Single entry i.e. it can be reached only directly, in this case the following event occurs

a) Send the message to D
b) SET COUNTER =3
c) DO WHILE COUNTER !=0
d) wait for acknowledge received till timeout
e) IF acknowledge received, Terminate process

ELSE { 
    resend (message,D)

        COUNTER--;

    }

ENDIF

ENDLOOP

f) Remove D from the routing table as it is not reachable and print message destination not reachable

Case 2) Entries are all indirect then

Step2) sendAll (urlList)

Step 3) Wait for the acknowledgement
Step 4 IF receive acknowledge then

    Terminates the process

    ELSE

    Remove the entry in the table for D

Case 4) Entries contain both direct and indirect path then it

Step 1: first try to send directly i.e. send(message,D) and wait for acknowledge

Step 2: IF acknowledge received within Timeout then

    Terminate the process

    ELSE

    Flood the message to all possible multi hop path i.e. send All(message) & wait for acknowledgment

    IF any acknowledge received within time out then

        Terminate process

        ELSE

        Print msgdestination Unreachable & Remove that Entry from the table

        END IF

    ENDIF

Receiver side: suppose node S want to send the Message to D via A-B.

When the packet is received at the node can be of three types.

Case 1: node is the destination

Step 1: the node will simply Accept the message. And extract the body message & use it.

Step 2: Send the acknowledge to source through the path in the header portion through which the message has come.

Receiving Process

IF header type is equal to `1 then

    discard message

ELSE

    add the message id to cache

IF localcachecontains message ID then

    return ok

ELSE

    process(message, sender)

    return ok

END IF
ELSE
local cache contains message ID then
    IF destination equals the receiver then
        discard message
    ELSE
        IF local cache contains message ID then
            return ok
        ELSE
            print “message received”
            return ok
        ENDIF
        add the message id to cache
        ENDIF
    ELSE
        redirect(message, header)
    ENDIF
ENDIF

Process (message, header)

Suppose a node A maintains its table as follows

<table>
<thead>
<tr>
<th>Node</th>
<th>List</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>B,C</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>D</td>
<td>D</td>
</tr>
</tbody>
</table>

Now entry for B means that it can be reach from C or directly

Now suppose routing table comes from C in the following form

<table>
<thead>
<tr>
<th>Node</th>
<th>List</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>D</td>
<td>C</td>
</tr>
<tr>
<td>E</td>
<td>E</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>B</td>
<td>B,E,F</td>
</tr>
</tbody>
</table>

At the receiver side it first check whether an entry exist for the coming sender or not if not found i.e. the case when the sender node starts after completion of search of node of receiving node then it first search for the device and make entry in the table and process as follows
1) Search for each entry in the incoming table with the existing table except entry for node which receives the table.

2) IF matches found then

i) Check the corresponding list entry in the existing table against each entry in list of sender node in the existing table.

ii) IF no entry found then a) add the entry to the list

iii) Else add entry for the node, list in the table

3) End

For example for Node D entry in the table is

| D | D |

The incoming entry for D is

| D | C |

As the previous list entry for D does not contain entry for C there for list is added so it becomes

| D | D,C |

Similarly entry for E, F is not present in the A table so they will be added

But in case of B there is already entry of C in the existing table

| B | B,C |
| C | C |
| D | D,C |
| E | C |
| F | C |

It may also possible that there can be more that there would be more than one more than on entry for the sender in the existing table

Redirect (message, header)
Step 1: add the local address (i.e. node which is redirecting) to the beginning of the path field in the header

Step 2: find the list for the destination of the packet to be redirected

Step 3: send the packet to all the nodes form where the destination to be reached

Step 4: wait for acknowledgement

Step 5: if ack not received within time then return OK to the sender; Else return failed to sender

Step 6: END

V. PROTOCOL APPLICABILITY AND ASSUMPTIONS

The main assumptions underlying the proposed protocols are
1) Each node first discover the nodes in the neighbor zone during startup
2) Each nodes periodically shares its routing table with each other
3) If a node discover that on for its neighbor is no longer in its zone then it will updated its table and also informed its neighbor nodes.
4) We assume that the intermediate node has sufficient memory to relay the incoming message To be transmitted

VI. CONCLUSION

Our algorithm solved the problem of multihop adhoc network which optimized the routing and reduces a large number of path generated by flooding. Also each node will first prefer to send directly to node instead of broadcast the packet to all its neighbour .this will greatly reduce the overall traffic generated by flooding of packet.

REFERENCES
Effect Algorithm for Mobility (DREAM), Proceedings of the fourth Annual mobile computing and networking, October 1998.


