Location Aware Route Maintenance Protocols for the Mobile Bluetooth Radio Networks

Prasan Kumar Sahoo
Dept. of Information Management
Vanung University
Chungli, Taiwan, 32045, R.O.C.
Email: pksahoo@msa.vnu.edu.tw

Chih-Yung Chang, Sheng-Wen Chang
Dept. of Computer Science
and Information Engineering
Tamkang University, Taipei, Taiwan, R.O.C.
Email: cychang@mail.tku.edu.tw
Email: swchang@wireless.cs.tku.edu.tw

***Abstract*— Bluetooth is a low-cost, low-power and short range communication technology, which operates in 2.4 GHz ISM band. The important research issues in Bluetooth are scatternet formation and routing, since nodes can arrive and depart at arbitrary time. In this paper, novel route maintenance algorithms are proposed for the Bluetooth scatternet that supports mobility of the nodes. Our protocols guarantee the connectivity among nodes and reconstruct the routes dynamically by taking their location information. Besides, we propose how to reduce the number of hops and to form the shortest route between the source and the destination due to addition of nodes. Performance analysis of our work shows that it outperforms in terms of end to end transmission delay, bandwidth consumption and route maintenance as compared to similar Bluetooth routing protocols.**
Keywords: Bluetooth, scatternet, route maintenance, mobility.

I. INTRODUCTION

Bluetooth technology [1] is based on a centralized master-slave communication, where the master allocates transmission slots to the slaves. One master with maximum up to 7 active slaves constitutes a piconet, which employs different frequency hopping code-division multiple-access techniques to prevent mutual interferences. Bluetooth technology can be extended to interconnect multiple piconets to form a large ad hoc scatternet consisting of hundreds of mobile devices. The rely nodes can be a master in one piconet and slave in another or bridge between two or more piconets. A blue-tree scatternet formation algorithm [4] is proposed to build a self routing scatternet to minimize the routing overhead. But, it does not mention how to construct the scatternet, if nodes are not within the proximity of each other. In [5], authors define a routing scheme for Bluetooth scatternets, which is based on the Zone Routing Protocol and explain how the scheme takes into account the specifics of the Bluetooth MAC layer and also provide simulation results showing the performance of the scheme. The authors in [6] propose a so-called Blueline algorithm to reduce the time and path length in routing, in which two Bluetooth nodes should communicate directly.

Since, Bluetooth is a short-range communication technology, we feel that its indoor applications are more than the outdoor applications. The typical example is the m-commerce scenario [7], in which customers walk around a large commercial area or shopping mall carrying wireless PDA and Bluetooth enabled wireless devices. Considering the recent technological developments for the m-commerce environments, we assume that location information can be transferred to the Bluetooth enabled handheld devices by several means. For example, LANDMARC [8], a location sensing prototype system that uses RFID technology for locating objects inside buildings and it improves the overall accuracy of locating objects by utilizing the concept of reference tags. Besides, the Bluetooth Location Networks (BLN) [9] transmits location information to the service servers without user's participation. The authors in [10] propose a route reduction protocol to reduce the number of hops as compared to the works in [2] and [3], taking location information of the nodes. In [11], authors propose an on-demand routing protocol for the Bluetooth scatternet, which can detect the device movement and establishes routes in a mobile scatternet to cope with both power consumption and device mobility issues.

Though, considerable research works are done in the area of routing in Bluetooth ad hoc networks, constructing and maintaining the route due to mobility of the nodes is an important research issue and have not been studied extensively. Hence, the contribution of our in this paper can be summarized as follows:

- We propose MObility based Location Aware Route maintenance *(MOLAR)* algorithm that maintains the route due to mobility of the nodes in a scatternet.
- We have developed algorithms, which reconstruct links, if nodes are entered to or left from the piconets.
- We propose algorithms to reduce the routing path due to addition of nodes and to create subrouting paths to route data efficiently.

Rest of the paper is organized as follows. Section II discusses some important routing algorithms related to our protocol. Our mobility based location aware route maintenance algorithms are proposed in Section III. Simulation results and performance analysis of our protocols are discussed in Section IV and concluding remarks are made in Section V of the paper.

II. OVERVIEW OF ROUTING PROTOCOLS

Let us consider a scatternet, as shown in Fig. 1, nodes S, S_{41} and D are pure slaves, M_1 and M_4 are pure masters for the piconets P_1 and P_4, respectively. Node B_{12} is the master

0742-1303/07 $25.00 © 2007 IEEE
DOI 10.1109/LCN.2007.144

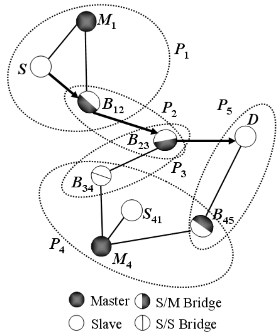


Fig. 1. Routing path formed by LARP [10] algorithm to route data from node S to D.

for the piconet P_2 and bridge between piconets P_1 and P_2. Node B_{23} is the master for the piconet P_3, as well as a bridge between P_2 and P_3. Node B_{45} is the master for the piconet P_5 as well as a bridge between P_4 and P_5. Node B_{34} is the bridge between piconets P_3 and P_4. In this scatternet, let node S has to route data to the destination device D.

According to RVM [3], the final routing path could be $S \longrightarrow M_1 \longrightarrow B_{12} \longrightarrow B_{23} \longrightarrow B_{34} \longrightarrow M_4 \longrightarrow B_{45} \longrightarrow D$, which requires 7 hops to route the packet from the source to the destination. However, the authors in LORP [2], find the path $S \longrightarrow B_{12} \longrightarrow B_{34} \longrightarrow B_{45} \longrightarrow D$, which requires 4 hops instead of 7 hops. In LARP [10], the authors propose a location aware routing algorithm that can route along the path $S \longrightarrow B_{12} \longrightarrow B_{23} \longrightarrow D$, which requires only 3 hops for the same scatternet, as shown in Fig. 1. Though, the routing path in LARP is shorter than LORP and RVM, LARP does not consider mobility of the nodes and corresponding routing path. Due to mobility of nodes, it may possible that the routing path may be increased, thereby increasing the number of hops. Hence, we propose a mobility based routing protocol (MOLAR) that considers the location information and mobility of the nodes and develop algorithms to reduce the routing path further.

III. Mobility based Location Aware Routing (MOLAR) Protocol

In our work, it is assumed that each device of the scatternet knows its location information through RFID [8] and BLN [9] and each node has a unique ID different from its BD_ADDR. The source node of one piconet intends to communicate with the destination node of another, whose ID is known, but location is unknown. Besides, it is assumed that each master knows the ID, $clock\_offset$ and location information of its slaves during the scatternet formation phase. The master can get this information about its slaves during connection phase of the piconet. The intermediate nodes can get location information of the source and the destination, when control packet is routed from the source to the destination during the route search phase. Before proceeding to our protocols and algorithms, we introduce here some definitions, which are used in our protocol.

A. Definitions

Routing Master(M_i): Any master M of i-th piconet is known as a routing master M_i, if any of its slaves or master itself is a member of the initial shortest path between the source and the destination. It is to be noted that initially a shortest path is formed between the source and destination and the route is reconstructed due to addition or deletion of the nodes. Each routing master stores route information, including the BD_ADDR, $clock\_offset$, and location information (LOC) of the members, those who participate in the routing.

As shown in Fig. 1, nodes M_1, B_{12} ,B_{23}, and B_{45} are the routing masters of the whole scatternet, since their slaves or themselves are member of the initial shortest routing path between the source and the destination.

Routing Piconet(P_i): Any piconet that contains a routing master M_i is known as a routing piconet P_i. As shown in Fig. 1, piconets P_1 , P_2, P_3 and P_5 are the routing piconets. If any slave joins in any routing piconet, BD_ADDR, $clock\_offset$, and LOC of that node should be forwarded to the routing master of that routing piconet.

SNR Threshold: The ratio of the received signal to the noise is called the signal to noise ratio (SNR), which should be $\geq \rho$, where ρ is a user defined threshold and is fixed for all nodes of the scatternet. According to definition,
$SNR = \frac{received\_power}{interference\_power} \geq \rho$

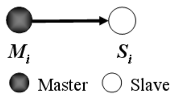
Weak Node: The node whose SNR value is less than the SNR threshold (ρ) is termed as a weak node. It is to be noted that a weak node must be a receiver.

Weak Link: A link connecting to any node with a weak node is termed as weak link. It is to be noted that a weak link may connect to two weak nodes or may connect to one weak node with another node. In our protocol, since, a sender is not aware of its receiver, whether it is a weak node or not, the weak node notifies its sender that it becomes a weak node and the link between the sender and the receiver is a weak link.

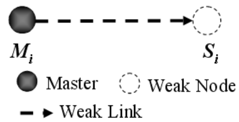
As shown in Fig. 2(a), initially, let there be a link between nodes M_i and S_i. If node S_i moves towards right, as shown in Fig. 2(b), the distance between nodes M_i and S_i is increased. Since, the signal strength received by the receiver S_i may be reduced due to its mobility, SNR value of node $S_i < \rho$. Hence, node S_i becomes a weak node and the link between M_i and S_i becomes a weak link, which is notified by S_i to its sender M_i.

Member Collection Procedure (MCP): In this procedure, a weak node S_i requests its routing master M_i and M_{i+1} of i-th and $i+1$-th piconet, respectively to return the BD_ADDR, $clock\_offset$ and LOC of the nodes of the routing piconets P_i and P_{i+1}. When a weak node goes for the *MCP*, it forwards a member collection packet to the routing masters.

As mentioned in the related work, the location aware routing protocol (LARP) [10], reduces the routing path as compared to RVM [3] and LORP [2]. However, we develop algorithms to accommodate new nodes to the original scatternet to reduce the routing path and reconfigure it as described below.



(a) Example of a normal link



(b) Example of a weak link and weak node due to mobility of node S_i

Fig. 2. Illustration of weak node and weak link

TABLE I

ALGORITHM 1: Node Add Procedure
Notation :
1. S_{new}: Newly added node to the piconet;
2. M_i: Routing master of i-th piconet;
3. P_i: Routing piconet;
4. S_i or S_j: Slave i or j;
Node Add Procedure(S_{new}, Role switch operation)
1. **Step 1:** If: a node S_{new} newly connects to any P_i
2. M_i, which is connected to S_i calculates:
3. If: S_{new} can reduce routing path, with any one of the members S_1, S_2,...,S_{i-1}, S_i, S_{i+1},...,S_n.
4. **Step 2:** If: S_{new} can connect to S_k, for any $k \geq i+3$ or can connect to S_j, for any $j \leq i-2$
5. Routing path can be reduced;
6. **Step 3:** If: S_{new} can reduce routing path,
7. M_i executes connecting (S_j, S_{new}) || (S_{new}, S_k);
8. **Step 4:** S_{new} executes piconet combination operation;

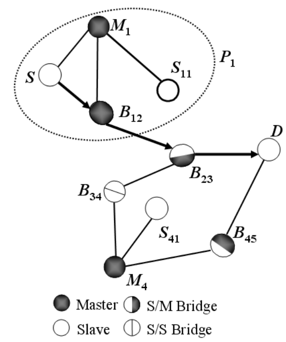


Fig. 3. Addition of new node S_{11} to the routing master M_1

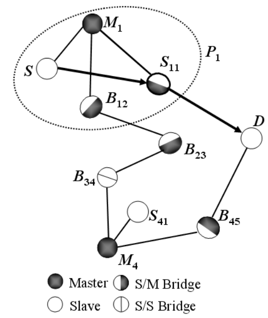


Fig. 4. The new routing path after addition of node S_{11}.

B. Node Add Procedure

In an ad hoc Bluetooth network, it is possible that new nodes may be added to the original scatternet. If a new slave S_{new} joins to a routing piconet, the routing master M_i calculates, whether or not the new slave S_{new} can reduce length of the routing path. In order to verify the path reduction, the routing master M_i carries out the *Node Add* procedure, as given in Table I. The above algorithm can be explained with an example, as shown in Fig. 3 and Fig. 4. Let, a new slave S_{11} joins to the routing master M_1 of piconet P_1. Then, the routing master M_1 initiates *Node Add* procedure. Since, M_1 knows the BD_ADDR, $clock\_offset$ and LOC of nodes S, B_{12}, B_{23} and D, it calculates if S_{11} can connect to S so that the routing path is reduced to S, S_{11}, and D. The routing master M_1 executes the procedure connecting (S, S_{11}) and connecting (S_{11}, D), as given in Table III. It notifies nodes S_{11} and D to enter to page scan state and node S to enter to page state. Then, node S establishes a link with node S_{11} and node S_{11} enters page state. After that, node S_{11} constructs a link with node D and executes the piconet combination operation to reduce the number of hops and piconets. Thus, nodes S, S_{11} and D construct a piconet, as shown in Fig. 4, where node S_{11} plays the role of a master and number of hops are reduced to 2. It is to be noted that the final routing path as per LARP [10] is $S \longrightarrow B_{12} \longrightarrow B_{23} \longrightarrow D$, as shown in Fig. 1, whereas in our mobility protocol, the routing path could be $S \longrightarrow S_{11} \longrightarrow D$, as shown in Fig. 4.

C. Node Leaving Procedure

The node leaving procedure comprises three policies to maintain the route due to mobility of a node. The first policy is the *Node Replacement* policy that finds another node to replace the weak node. The second policy is the *Link Replacement* policy that finds other links to replace the weak link and finally the *Local LARP* policy that reconstructs a sub-routing path due to mobility of a node. In our protocol, we suggest that *Node Replacement* policy has higher priority over *Link Replacement* policy, since *Node Replacement* policy maintains a shorter route. Similarly, the *Link Replacement* policy has higher priority over *Local LARP* policy, since *Link Replacement* policy costs less control overhead than the *Local LARP* policy.

1) Node Replacement Algorithm: In this procedure, a routing master or weak node intends to select one of the devices to replace the weak node. In the selection process, the first priority is given to the slave over the master and a master is given priority over a bridge. Normally, a slave node is preferred to be selected by the routing master or by the weak node, since it does not raise any cost in guard time and its traffic overhead is lower than the master. The details of the node replacement algorithm is described in Table II. It is to be noted that after selecting any node from the piconets based on the algorithm given in Table II, either the routing master or the weak node goes for the connecting procedure, as given in Table III.

As shown in Fig. 5, let, initially there exists a routing path

TABLE II

ALGORITHM 2: Node Replacement Procedure
$Notation$:
1. P_i: *Routing piconet*;
2. M_i: *Routing master*;
3. S_i: *Weak node (may be pure slave or bridge)*;
4. S: *Any device*;
1. **Node Replacement**(S_i, M_i)
2. { If(S_i is connected to one M_i and has one weak link)
3. { If(M_i selects one of S located in P_i
such that S can connect to S_{i-1} and S_{i+1})
4. { M_i executes connecting procedure (S_{i-1}, S) and (S, S_{i+1}); }
5. Else
6. GO to *LINK REPLACEMENT* procedure; }
7. Else
8. {If(S_i selects one of devices S located in P_i || P_{i+1}
such that S can connect to S_{i-1} and S_{i+1}1)
9. {S_i executes connecting procedure (S_{i-1}, S) and (S, S_{i+1});}
10. GO to LINK REPLACEMENT Procedure }}

TABLE III

ALGORITHM 3: Connecting Procedure
$Notation$:
1. M_i: *Routing master*;
2. S_i: *Weak node*;
3. d_i: *Any type of node* i;
1. **Connecting** (d_1 ,d_2) **Procedure**
2. {Routing master M_i or weak node S_i notifies device d_1 and d_2;
3. Device d_1 goes to page state;
4. Device d_2 goes to page scan state;
5. Device d_1 constructs a link with d_2; }

$S \longrightarrow S_{11} \longrightarrow D$ in the scatternet. If node S_{11} moves towards right, a weak link (S, S_{11}) is formed. Since, node S_{11} becomes a weak node, it informs M_1 to initiate the node leaving procedure (S_{11}, M_1), after waiting for a random back-off time. If any weak node S_i has only one weak link, routing master M_1 executes the node replacement (S_{11}, M_1) procedure. As shown in Fig 5, since, node S_{12} can connect to nodes S and D, routing master M_1 selects S_{12} to replace S_{11} along the new route. Finally, routing master M_1 executes the connecting procedure to connect (S, S_{12}) and (S_{12}, D). Then, node S_{12} executes the piconet combination operation to reduce the number of piconets and becomes the master of

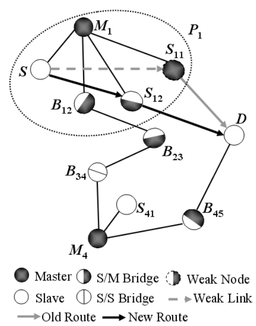


Fig. 5. Replacement of node due to mobility of S_{11}.

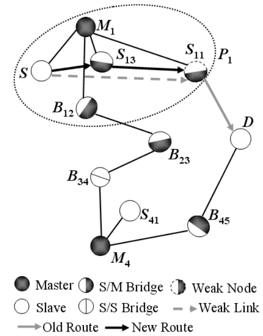


Fig. 6. One weak link is formed due to mobility of S_{11}, which is replaced by the bold lines.

nodes S and D. As shown in the figure, it is observed that the number of hops of the new route is same as the old one.

2) Link Replacement Algorithm: As mentioned earlier, the link replacement algorithm is executed, if any weak link is formed due to mobility of a node. The link replacement can be categorized into three cases, as described below.

Case 1: *If a weak node has only one weak link and is connected to one routing master.* In this case a weak node does not execute member collection procedure. To explain the link replacement algorithm given in Table IV, an example is given in Fig. 6. Let, there exists a routing path $S \longrightarrow S_{11} \longrightarrow D$ in the scatternet, as shown in Fig. 6. If node S_{11} moves to right, it creates a weak link (S, S_{11}). Since, node S_{11} becomes the weak node, it informs M_1 to initiate the node leaving procedure(S_{11}, M_1), after waiting for a random backoff time. If any weak node S_i has only one weak link, routing master M_1 executes the node replacement(S_{11}, M_1) procedure. However, here node replacement procedure is failed since, no device can be connected to S and D. In this case, routing master M_1 executes the link replacement(S_{11}, M_1) procedure and finds a slave S_{13}, which can connect to nodes S and S_{11}. Then, routing master M_1 executes the connecting procedure(S, S_{13}) and (S_{13}, D). Finally, node S_{13} executes the piconet combination operation to reduce the number of piconets and becomes the master between nodes S and S_{11}. In this case, the length of the new routing path is longer and more than one hop as compared to old one.

Case 2: *If a weak node has two weak links and is connected to one routing master.* In this case a weak node executes the member collection procedure. The replacement of such link is shown in Fig. 7. As shown in Fig. 7, let initially there exists a routing path $S \longrightarrow S_{11} \longrightarrow D$ in the scatternet. If node S_{11} moves right and up, it creates weak links (S, S_{11}) and (S_{11}, D), as shown in Fig. 7. By measuring the SNR value, both nodes S_{11} and D become weak nodes and notify to their senders S and S_{11}, respectively. Since, node S_{11} has two weak links, it initiates the node leaving procedure (S_{11}, M_1) after waiting for a random backoff time. The weak node S_{11} executes the member collection (M_1, B_{45}) procedure. Weak node S_{11} sends a member collection packet

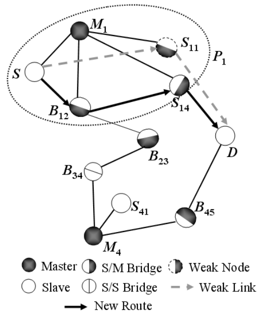


Fig. 7. Two weak links are formed due to mobility of S_{11}, which are replaced by the links shown in bold lines.

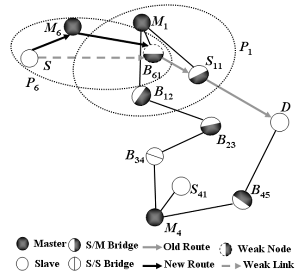


Fig. 8. Weak link is formed due to mobility of B_{61}, which is later replaced as shown by bold lines.

to the routing master M_1 and routing master B_{45} through node D. Let, S_1 and S_2 be the set of devices located in the routing piconet P_1 and P_5, respectively. $S_{11} = \{M_1, S, B_{12}, S_{11}, S_{14}\}$ and $S_2 = \{B_{45}, D\}$ and hence $S_{left} = \{M_1, B_{12}\}$ and $S_{right} = \{S_{14}\}$. Weak node S_{11} checks whether or not there exists any device in $S_1 \cup S_2$ such that node replacement or link replacement can be applied. Weak node S_{11} executes node replacement(S_{11}, M_1) procedure, but is failed, since no device can connect to nodes S and D. Then, weak node S_{11} executes the link replacement(S_{11}, M_1) procedure and finds B_{12} along S_{left} and S_{14} in S_{right} such that B_{12} can connect to S_{14}. Weak node S_{11} executes connecting procedure (S, B_{12}), (B_{12}, S_{14}) and (S_{14}, D). Then, node B_{12} executes the piconet combination operation to reduce the number of piconets and becomes the master of nodes S and S_{14}. in this case, the length of the new route is more than one hop than the old one.

Case 3: *If a weak node has one or two weak links, but is connected to two routing masters.* In this case a weak node executes member collection procedure. The link replacement of this case is shown in Fig. 8. As shown in Fig. 8, let there exists a routing path $S \longrightarrow B_{61} \longrightarrow S_{11} \longrightarrow D$ in the scatternet. If node B_{61} moves towards right, it generates a weak link (S, B_{61}), as shown in Fig. 8. By measuring the SNR value, node B_{61} becomes a weak node and notifies to its sender S. In this case, weak node B_{61} connects to two routing masters and therefore executes the member collection procedure (M_6, M_1). Weak node B_{61} sends a member collection packet to the routing masters M_6 and M_1. Let, S_1 and S_2 be the set of devices located in the routing piconets P_1 and P_6, respectively. $S_1 = \{M_6, S, B_{61}\}$ and $S_2 = \{M_1, B_{61}, B_{12}, S_{11}\}$. Weak node B_{61} checks whether or not there exists any device in $S_1 \cup S_2$ such that node replacement or link replacement can be applied. Weak node B_{61} executes node replacement(B_{61}, M_6) procedure, but is failed, since no device can connect to nodes S and S_{11}. Then, weak node B_{61} executes link replacement(B_{61}, M_6) procedure and finds M_6 can connect to S and B_{61}. Weak node B_{61} executes connecting (S, M_6) and connecting (M_6, B_{61}). Then, node M_6 executes the piconet combination operation to reduce the number of piconets. In this case, the length of the new route is more than one hop as compared to the old one.

A node, after becoming weak one, notifies its status to its connecting nodes, waits for a random backoff time and then executes the node leaving procedure or notifies its routing master to execute the same. If a relay (bridge) node has two weak links along its forwarding route, the relay node and its receiver, both become weak nodes. In this case, both of them individually execute the node leaving procedure after waiting for a random backoff time. This creates two sub-routing paths or length of the routing path is increased further. In order to avoid this situation, the weak node having two weak links waits less backoff time and first executes the node leaving procedure by itself. In order to repair the two weak links efficiently, the weak node requires information about other nodes located in its routing piconet and the routing piconet connected by its receiver. The weak node sends a member collection packet to its routing master and to the routing master connected by its receiver, before going for the member collection procedure. Upon receiving a member collection packet, its receiver node stops repairing the routing path. The details of the link replacement algorithm is described in Table IV.

Finally, the algorithm for the node leave procedure that combines all types of link replacements is summarized in Table V.

3) Local LARP: If S_i is weak node, in this procedure, node S_{i-1} executes LARP [10] algorithm to construct a sub-routing path with node S_{i+1}. As shown in Fig. 9, let there exists a routing path $S \longrightarrow B_{61} \longrightarrow S_{11} \longrightarrow D$ in the scatternet. If node S_{11} moves right and up, it creates the weak links (B_{61}, S, S_{11}) and (S_{11}, D), as shown in Fig. 9. Both nodes S_{11} and D become the weak nodes and notify to their senders B_{61} and S_{11}, respectively. Since, node S_{11} has two weak links, it initiates the node leaving procedure(S_{11}, M_1) after waiting for a short random backoff time. Moreover, weak node S_{11} has two weak links. Hence, it executes the member collection (M_1, B_{45}) procedure. Let, S_1 and S_2 be the set of devices located in the routing piconets P_1 and P_5. $S_1 = \{M_1, B_{61}, B_{12}, S_{11}\}$ and $S_2 = \{B_{45}, D\}$ and hence $S_{left} = \{M_1, B_{12}\}$ and $S_{right} = \{\Phi\}$. Then, weak node S_{11} executes the node replacement(S_{11}, M_1) procedure and finds that no device can connect to B_{61} and D. After this, weak node S_{11} executes the link replacement(S_{11}, M_1) procedure and also finds that no two devices can connect to each other.

TABLE IV

ALGORITHM 4: Link Replacement Procedure

$Notation:$
1. P_i: *Routing piconet*;
2. M_i: *Routing master*;
3. S_i: *Weak node*;
4. S: *Any device*;

1. **Link Replacement**(S_i, M_i)
2. $\{S_{left}$: $\{S \mid S$ is able to connect with both S_{i-1} and $S_i\}$;
3. S_{right}: $\{S \mid S$ is able to connect with both S_i and $S_{i+1}\}$;
4. If(S_i has one weak link and connects to one M_i)

CASE 1

5. $\{$If(M_i selects one of devices S located in P_i such that S can connect to S_{i-1} and S_i)
6. $\{M_i$ executes connecting procedure (S_{i-1}, S) and (S, S_i);$\}$
7. Else
8. GO to execute Local LARP;$\}$
9. Else
10. $\{$If(S_i has one weak link and is connected to two M_i)

CASE 3

11. $\{$If(S_i selects one of the devices S located in P_i || P_{i+1} such that S can connect to S_{i-1} and S_{i+1})
12. $\{S_i$ executes connecting procedure (S_{i-1}, S) and (S, S_{i+1});$\}$
13. Else
14. GO to execute Local LARP;$\}$
15. Else
16. $\{$If(S_i has two weak links and connects to one or two M_i)

CASE 2 and 3:

17. $\{$If($\exists\ S_1 \in S_{left}$ and $S_2 \in S_{right}$ such that S_1 can connect to S_2)
18. $\{S_i$ executes connecting procedure (S_{i-1}, S_1) and (S_1, S_2); and connecting (S_2, S_{i+1})$\}$
19. Else
20. GO to execute LOCAL LARP; $\}\}\}\}$

TABLE V

ALGORITHM 5: Node Leaving Procedure

$Notation:$
1. P_i: *Routing piconet*;
2. M_i: *Routing master*;
3. S_i: *Weak node (may be pure slave or bridge)*;
4. S: *Any device*;

1. **Node Leaving**(S_i, M_i)
2. CASE 1: If(S_i connects to one M_i and has one weak link)
   //Node Replacement
3. Step 1: M_i executes Node Replacement(S_i, M_i) Procedure;
4. If(Node Replacement fails)
5. M_i proceeds to Step 2;
   // Link Replacement
6. Step 2: M_i executes Link Replacement(S_i, M_i) Procedure;
   If (Link Replacement fails)
7. M_i proceeds to Step 3;
   //Executes Local LARP to repair sub-path
8. Step 3: M_i executes Local LARP(S_{i-1}, S_{i+1});
9. CASE 2: If(S_i connects to one M_i and has two weak links)
10. Step 1: S_i executes member collection(M_i, M_{i+1}) procedure and proceeds to Step 2;
    //Node Replacement
11. Step 2: S_i executes Node Replacement(S_i, M_i) Procedure;
12. If(Node Replacement fails)
13. Weak node S_i proceeds to Step 3;
    // Link Replacement
14. Step 3: Weak node S_i executes Link Replacement(S_i, M_i) Procedure
15. If(Link Replacement fails)
16. S_i proceeds to Step 4;
    //Executes Local LARP to repair sub-path
17. Step 4: S_i executes Local LARP(S_{i-1}, S_{i+1});
18. CASE 3 : If(S_i connects to two M_i and having one or two weak links)
19. Step 1: GO to Step 1 of Case 2;

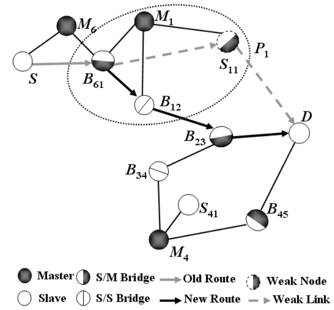


Fig. 9. Execution of Local LARP after movement of node S_{11}

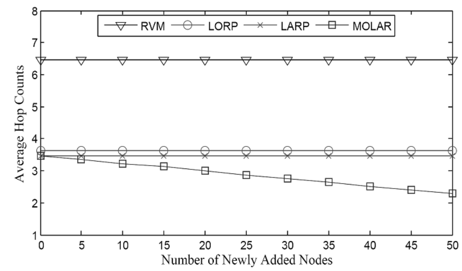


Fig. 10. Average number of hop counts for different number of newly added nodes

Weak node S_{11} initiates local LARP(B_{61}, D) and then the new routing path $B_{61} \longrightarrow B_{12} \longrightarrow B_{23} \longrightarrow D$ can be constructed. Node B_{23} executes the piconet combination operation and becomes the master of nodes B_{12} and D. Thus, node B_{12} becomes a S/S bridge node.

IV. Performance Analysis

In our simulation, initially a connected scatternet is taken with fixed numbers of 100 Bluetooth devices, which are randomly distributed over a squared area of $50 \times 50\ m^2$. The routes are chosen based on the transmission of control packets from the source to the destination and initial routing paths are generated using C++ programming. New routing paths are regenerated by adding or taking away of nodes. The control packets are sent from one node to another and all possible successful paths between the source and the destination are simulated taking mobility into consideration. Thus, the average end-to-end delay and routing path length are estimated for different number of mobile nodes. Finally, the performance results of our mobility based routing protocol (MOLAR) is compared with RVM [3], LORP [2] and LARP [10] as follows.

As shown in Fig. 10, average number of hop counts for different number of newly added nodes are simulated with different routing protocols that we have considered. It is observed that our protocol outperforms in terms of number of hop counts as compared to RVM [3], LORP [2] and LARP [10], when more new nodes are added to the existing scatternet. As per our algorithm, we got the most expected results as number of hops are reduced in MOLAR due to addition of new nodes. Fig. 11 shows the average number

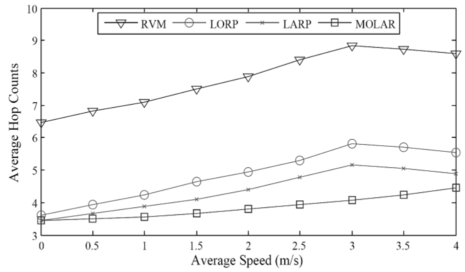


Fig. 11. Average number of hop counts for different average speed of the nodes.

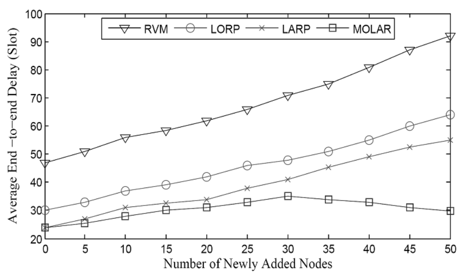


Fig. 12. Average end-to-end delay for different number of newly added nodes

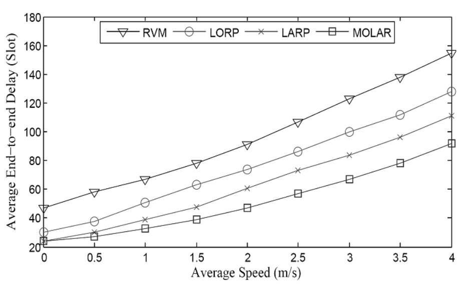


Fig. 13. Average end-to-end delay for different average speed of the nodes

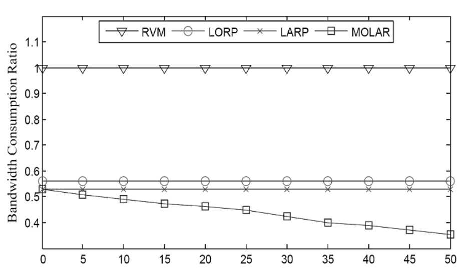


Fig. 14. Bandwidth consumption ratio for different number of newly added nodes.

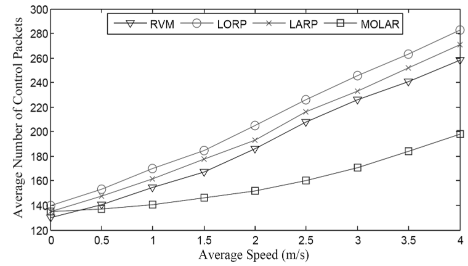


Fig. 15. Required average number of control packets for different average speed of the mobile nodes.

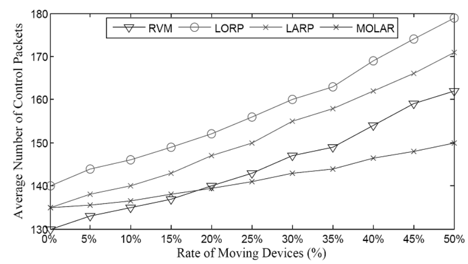


Fig. 16. Average number of control packets for different rate of mobile nodes.

of hop counts for different average speed of the nodes. It is observed that our protocol gives tremendous improvement in terms of hop counts for different average speed of the nodes. In RVM, LORP and LARP, they initialize their protocol to find a new and worse routing path from the source to the destination if a link of the routing path is broken. Therefore, the average hop counts of RVM, LORP and LARP are increased, when average speed is less than or equal to 3 m/s. However, when average speed is larger than 3 m/s, the scatternet topology is changed and thereby RVM, LORP and LARP may find a better route than the original one. Besides, MOLAR executes the node replacement, link replacement or local LARP policies to maintain or increase the route length, if a link of a routing path is broken. Therefore, if average speed is larger, the movement of a node easily causes to break two links simultaneously. Hence, MOLAR easily executes the link replacement or local LARP policies to increase the hop counts of the routing path. In our simulation, we have analyzed the average number of control packets for different routing protocols such as RVM,

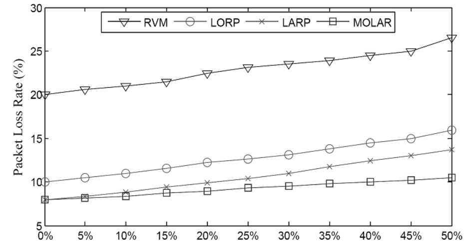


Fig. 17. Rate of packet loss for different rate of mobile nodes.

LORP etc.

From Fig. 12, it is observed that the average end-to-end delay of our protocol is less than that of the RVM, LORP and LARP. When the number of newly added nodes is less than or equal to 30, the average end-to-end delay of RVM, LORP, LARP and MOLAR is increased, since newly added nodes extend the polling time of the masters. However, we find that the average end-to-end delay of MOLAR is decreased when the number of newly added nodes is more than 30. This is because the shortened routing paths improve the average packet transmission delay. Fig. 13 shows average end-to-end delay of different protocols for different average speed of the mobile nodes. In the simulation, the end to end delay includes the maintenance time that RVM, LORP and LARP construct a new and worse route and that MOLAR executes node leaving procedure to repair the broken links. It is observed that the average end-to-end delay of our protocol is less than that of the RVM, LORP and LARP, since MOLAR locally executes the route maintenance algorithms to repair the links. Besides,it is to be noted that the higher average speed causes the links broken easily and thereby raising the delay time in executing the route maintenance. Although route length of RVM, LORP and LARP may be shortened when average speed is larger than 3 m/s and the scatternet topology is changed, their route maintenance still require larger value of delay time.

In Fig. 14, we have compared the bandwidth consumption ratio for different number of newly added nodes for different routing protocols. It is found that our protocol outperforms over RVM, LORP and LARP. Since, the newly added nodes help to shorten the routing path length, it is obvious that the bandwidth consumption in MOLAR is reduced. In Fig. 15, it is observed that our protocol consumes least number of control packets as compared to LARP, LORP and RVM. Since, higher average speed of the nodes improves larger number of the broken links, RVM, LORP and LARP create more control traffic overhead to maintain a route than MOLAR. Besides, since LORP and LARP tries to shorten the route length, their control packets are larger than the control packets of RVM. Moreover, the route length of LARP is shorter than LORP and hence control packets of LARP are less than that of LORP.

The average number of control packets that are required for different protocols with different rate of moving devices is shown in Fig. 16. The rate of the moving devices is defined as the percentage of nodes those are moved out of the total number of nodes of the scatternet. From the figure, it is observed that our protocol requires least number of control packets as compared to LARP, LORP and RVM, when rate of moving devices is larger than 20%. Since, higher number of mobile devices cause larger number of broken links, RVM, LORP and LARP require more number of control packets to maintain the existing route. However, when rate of moving devices is less than 20%, RVM outperforms to MOLAR. Since, RVM uses least control packet to construct the original route and the broken links are fewer, when rate of mobile devices is less, the average number of control packet of RVM is less than that of MOLAR. Besides, as LORP and LARP require to shorten the routing path, their control packets are larger than control packets of RVM. Fig. 17 depicts the packet loss rate for different rate of mobile devices of different protocols. As shown in the figure, our protocol outperforms RVM, LORP and LARP. Since, moving devices break more links, the packet may be lost and the packet loss rate is increased in all protocols, if a new device is added.

V. Conclusion

In this paper, we propose mobility based location aware route maintenance protocols. Initially, we consider an existing connected and constructed Bluetooth scatternet and maintain the routing path due to addition or deletion of new nodes to it. We have developed algorithms to reduce the number of hops by adding new nodes and have proposed several algorithms to reconstruct the sub-routing path, if any node moves away. From our simulation studies it is observed that our protocol outperforms in terms of hop counts, end-to-end delay and bandwidth consumption as compared to other routing protocols. Hence, our protocol can be applicable to several real applications such as in big shopping malls, supermarkets and specifically in mobile e-commerce scenarios.

References

[1] The Bluetooth Specification, http://www.bluetooth.org 1.0b and 1.1

[2] G.-J. Yu, C.-Y. Chang, K.-P. Shih, S.-C. Lee, "Relay Reduction and Route Construction for Scatternet over Bluetooth Radio Systems," *Journal of Network and Computer Applications*, Vol 30, 728-749, 2007.

[3] P.Bhagwat and A.Segall, "A Routing Vector Method (RVM) for routing in Bluetooth Scatternets," in *Proc. of IEEE Workshop MOMUC*, pp. 375-379, Nov. 1999.

[4] M. Sun, C.K. Chang and T.H. Lai, "A Self-Routing Topology for Bluetooth Scatternets," T*he International Symposium on Parallel Architectures*, Philippines, May, 2002.

[5] R. Kapoor, M. Gerla, "A zone routing protocol for Bluetooth scatternets", in *Proc. of IEEE Wireless Communications and Networking Conference*, pp. 1459-1464, 2003.

[6] R.-S. Chang, and M.-T. Chou, "Blueline: A Distributed Bluetooth Scatternet Formation and Routing Algorithm", *Journal of Information Science and Engineering*, 21, 479-494, 2005.

[7] U. Vershney, R.J. Vetter, and R. Kalakota, "Mobile commerce: A new frontier", *Computer*, 10, pp. 32-38, 2000.

[8] L. M. Ni, Y. Liu, Y. C. Lau and A. P. Patil, "LANDMARC: Indoor Location Sensing Using Active RFID", in *Proc. of First IEEE International Conference on Pervasive Computing and Communications (PerCom'03)*, 2003.

[9] F.J. Gonzalez-Castano, J. Garcia-Reinoso, "Bluetooth Location Networks," in *Proc. of Globecom*, 2002.

[10] C.-Y. Chang, P. K. Sahoo, and S.-C. Lee, "A Location-Aware Routing Protocol for the Bluetooth Scatternet", *Wireless Personal Communications*, vol. 40, no. 1, pp. 117-135, Jan. 2007.

[11] C-H Yang and J-W Ruan, "On-demand Routing for Bluetooth Scatternets Subject to Device Mobility", in *Proc. of 19th International Conference on Advanced Information Networking and Applications (AINA'05)*, 2005.