A Location-and-Mobility Aware Routing Protocol for Bluetooth Radio Networks

Sheng-Wen Chang[a], Prasan Kumar Sahoo[b], Li-Ling Hung[c], Chih-Yung Chang[a]

[a]Tamkang University, Tamsui, Taipei, Taiwan,

[b]Vanung University, Chung-Li, Taiwan,

[c]Aletheia University, Tamsui, Taipei, Taiwan

swchang@wireless.cs.tku.edu.tw, pksahoo@vnu.edu.tw, llhung@mail.au.edu.tw, cychang@mail.tku.edu.tw

***Abstract*—Bluetooth is a most promising technology designed for the wireless personal area networks for the cable replacement. In this paper, a location aware mobility based routing scheme for the Bluetooth scatternet is proposed that constructs the links dynamically. Our proposed routing protocol requires location information of the nodes and constructs the route between any source and destination and reduces the number of hops. Besides, the network routing problems are analyzed and role switch operations are proposed to mitigate the problems. Moreover, the roles switch and route optimization operations are also proposed to improve route performance. Rigorous simulation works are done to evaluate the performance of our protocol in terms of mobility speed and number of mobile nodes and to compare our results with similar Bluetooth routing protocols. It is observed that our protocol outperforms in terms of energy consumption and transmission packet overheads as compared to similar Bluetooth routing protocols.**

I. INTRODUCTION

Bluetooth [1] ad-hoc network is a cutting edge technology that provides the short-range communication among the battery-operated portable radio devices such as personal digital assistant, headsets and notebooks. The underlying Bluetooth technology can support the connection-oriented and connectionless links to provide both voice and data transmission among the devices, typically located in the range of 10 meters. It can be classified into a single hop piconet or a multi-hop scatternet and a typical Bluetooth piconet consists of at most eight active devices, including one master and maximum up to seven active slaves. Both master and slaves hop over 79 channels with a speed of 1600/sec, and the time division duplex is employed for the sequential medium access. The master monitors the scheduling of the slaves and each piconet utilizes the frequency hopping spread spectrum (FHSS) to avoid interference and packet collision among the slaves. Different piconets employ different frequency hopping code-division multiple-access (FH-CDMA) channels to prevent mutual interferences. Hence, multiple piconets can co-exist in a common area and each piconet can be interconnected by means of some bridge nodes to form a bigger ad-hoc network known as scatternet. The bridge node can be a master in one piconet and slave in another or bridge between two or more piconets.

In Bluetooth ad hoc networks, it is obvious that nodes will enter and exit from the existing piconet time to time, thereby affecting the routing path. Though several papers propose the routing schemes for the static nodes, very limited papers talk about the mobility based routing of the Bluetooth scatternet. The authors in [2] propose a mobility model of mobile units that randomly move around a grid. The dynamic source routing protocol is used to calculate an appropriate multi-hop route through the Bluetooth personal area network (PAN) and may be suitable for the power-limited, multi-hop, ad hoc mobile devices. An on-demand routing protocol [3] for the Bluetooth scatternets is proposed that detects the mobility of the devices and establishes the routes in a mobile scatternet to cope with both power consumption and device mobility. However, the number of hop counts in this routing algorithm is not optimum. The authors in [4] propose a cluster based routing algorithm to construct and repair the routing path among different group of scatternets. However, the route length is also not optimum and the proposed algorithm costs addition time to reestablished the route. Though, considerable research works are done in the area of routing in Bluetooth ad hoc networks, maintenance of routing path due to frequent mobility of the nodes is an important research issue and has not been studied extensively. It is highly essential to maintain the existing routing path, if any one of the links of the routing path is broken. Hence, we propose here the mobility based routing algorithm that simultaneously constructs the shortest routing path and reserves a back-up path to maintain the routing due to mobility of the nodes.

The rest of the paper is organized as follows. The overviews of the related works are discussed in Section 2. Section 3 describes the system model and definitions of few related terms. Our Location Aware Mobility based routing Protocol (LAMP) is discussed in Section 4 of the paper. Performance evaluation of our protocol and comparison of the results with few standard routing protocols are discussed in Section 5 and concluding remarks are made in Section 6.

II. REALATED WORK

In this section, we analyze some standard routing protocols for the Bluetooth ad hoc networks. As discussed in section 1, though several protocols propose the routing mechanism for the Bluetooth technology, we consider here the Routing Vector Method (RVM) [5], relay reduction and route construction protocol (LORP) [6], and Bluetooth Master-Managed Routing (BMR) [7] protocol, as they have special relation to our proposed work.

The Routing Vector Method (RVM) [5] proposes the construction of routing path in Bluetooth scatternet between any source and the destination devices. The paper proposes a new packet forwarding method and discoveries the routing paths with the intermediate relay nodes. According to RVM, a source node broadcasts the SEARCH packet that accumulates the list of intermediate nodes along the routing path from the source to the destination. Upon receiving several broadcast packets, the destination device considers the first SEARCH packet of search process and unicasts a REPLY packet to the source along the path used for the SEARCH process.

For example, as shown in Figure 1, M_1, M_2, and M_3 are the master nodes for the piconets P_1, P_2, and P_3, respectively. Node C is the master for the piconet P_4 as well as a bridge between P_3 and P_4. Node A is the bridge between piconets P_1 and P_2, and B is the bridge between P_2 and P_3. If the packet is routed from source S of piconet P_1 to the destination node D of piconet P_4, according to RVM, the final routing path could be $S \rightarrow M_1 \rightarrow A \rightarrow M_2 \rightarrow B \rightarrow M_3 \rightarrow C \rightarrow D$ that requires 7 hops to route

978-1-4244-5228-6/09/$26.00 ©2009 IEEE

the packet from the source to the destination. However, we feel that the routing path in RVM is longer due to more number of hops, thereby increasing the latency and consuming more power and network bandwidth.

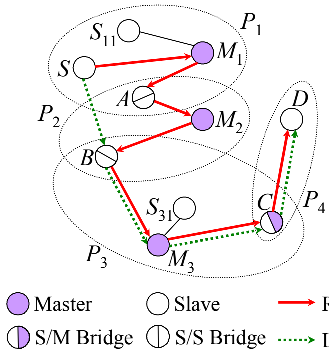

Figure 1: An example of routing paths constructed by RVM and LORP between the source S to the destination D.

A so-called relay reduction routing protocol (LORP) [6] for the Bluetooth scatternet is proposed to reduce the number of hops and to improve the drawbacks of RVM. In this work, the authors have proposed the relay reduction and disjoint routes construction algorithms for the Bluetooth scatternet. As per LORP, the network topology can be adjusted dynamically by reducing number of unnecessary relay nodes. Considering the physical distance of the nodes located in different piconets, numbers of hops are reduced and two disjoint routes for any pair of source and destination nodes are created. For example, as shown in Figure 1, though node S and B are within communication range (10 meters) of each other, still source S routes the packets through M_1, A, M_2 and finally to B, which requires 4 hops. According to LORP, since S and B can communicate directly, the packet can be routed through S, B, M_3, C and D and number of hops between the source and destination can be 4 instead of 7, as in RVM. But we still find some drawbacks in LORP, such as route length is still not shortest and some slave nodes require participating the path reduction, if a master asks its idle slaves to try to connect to destination or a relay in order to reduce the path length. So it may be just an overhead to the route construction thereby consuming more bandwidth and energy.

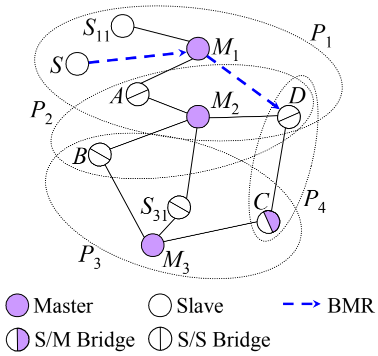

Figure 2: An example of the routing path constructed by BMR between the source S to the destination D.

A table-driven routing protocol named as Bluetooth Master-Managed Routing (BMR) [7] is proposed for the mobile Bluetooth ad hoc networks. The so-called BMR protocol is relied on robust scatternet in which a node having more or at most 7 neighbors can become a master and constructs the links with its nearby nodes. In BMR protocol, the scatternet has sufficient bridges to guarantee the existence of back-up routes. In order to select the shortest path from the source to destination, each master maintains the up-to-date information of the scatternet topology. For example, as shown in Figure 2, since node M_1 has more neighbors and node D is a neighbor of node M_1 when the scatternet is forming, node M_1 becomes the master and constructs a link with node D. Consequently, master M_1 selects the route $S \rightarrow M_1 \rightarrow D$ from the source S to the destination D according the information of the scatternet topology and number of hops between the source and destination can be 2 instead of 4, as in LORP. However, this routing algorithm works, if the nodes are static and fails for the mobility of the nodes. Though the authors have considered the mobility of the nodes, the up-to-date information is notified to each master of the scatternet, thereby resulting large number of control packets and consuming much bandwidth and energy.

In this paper, we propose a route reduction protocol that requires the location information of the nodes and shows a significant improvement over the RVM, LORP and BMR. Our protocol, which supports the mobility based routing still reduces the number of hops as compared to RVM and LORP and minimizes the control packets overhead as compared to BMR.

III. SYSTEM MODEL

In our proposed mobility based routing protocol, we consider a connected scatternet of Bluetooth enabled handheld devices. It is assumed that each node of the scatternet knows its location information through the LANDMARC [8] or Bluetooth Location Networks (BLN) [9]. The source node of any piconet can communicate with the destination node of another piconet, whose ID is known, but location information is unknown. Besides, it is assumed that each master of any piconet knows the ID, clock offset and location information of its active slaves. Each master also gets the location information of the intermediate nodes between the source and destination, when control packets are routed to construct the routing path. We introduce few definitions to explain our routing protocol as described in Section 4.

Definition: *Device ID (ID)*

Each Bluetooth node has a unique 48-bit Bluetooth device address (BD_ADDR). In our protocol, we assign one or two characters *Device ID (ID)* to each node of the scatternet, which is different from the unique BD_ADDR of a node. For example, A, S_3, M_{12} etc. are ID of the nodes, which are totally different from their BD_ADDR.

Definition: *Location (LOC)*

Location *(LOC)* of any node is its position in the scatternet, which is expressed in Cartesian co-ordinate (x, y).

Definition: *Initial Forwarding Node (IFN) set*

Set of nodes through which control packet is forwarded along the initial shortest path during the route search phase as described in Section 4 is termed as *Initial Forwarding Node (IFN) set*.

Definition: *Final Forwarding Node (FFN) set*

Set of nodes through which control packet is forwarded along the final shortest path during the route reply phase as described in Section 4 is termed as *Final Forwarding Node (IFN) set*.

Definition: *Final Backup Nodes (FBN) set*

Set of nodes through which control packet is forwarded along the final backup path during the route reply phase as described in Section 4 is termed as Final Backup Nodes (FBN) set.

Definition: *Last Forwarding Node (LFN)*

The intermediate node that is located in the communication range of the destination, but not connected to the destination is known as a *Last Forwarding Node* (*LFN*). In this case, the distance between the intermediate node and destination must be $\leq$ 10m (typical Bluetooth communication range) and therefore it can construct a link with the destination.

Definition: *Communicable Node Table (CNT)*

Any node, irrespective of its location in the same or different piconet can be element of the *Communicable Node Table (CNT)* of node A, if it lies within communication range of A. It is to be noted that in our protocol each master node of the scatternet maintains its *CNT* and entry in that table is 1, if a node is located in its communication range, else the entry is 0. It is assumed that each master knows location information of the intermediate nodes during route reply phase and estimates if any of them lies within its communication range. Besides, it updates the entry of *CNT* time to time, if any node is entered in or exit from the piconet due to its mobility.

Definition: *Equation of Ideal Path (EIP)*

Let $S\ (x_1, y_1)$ and $D\ (x_2, y_2)$ be the locations of the source and destination nodes, respectively. Then equation of the straight line connecting those two points is called the *Equation of Ideal Path (EIP).*

Definition: *Deviation from Ideal Path (DIP)*

The normal distance of the location of any node from the Equation of Ideal Path is termed as *Deviation from Ideal Path (DIP).*

IV. LOCATION AWARE MOBILITY BASED ROUTING PROTOCOL (LAMP)

Our Location Aware Mobility-based routing Protocol (LAMP) is divided into several phases such as route search, route reply and route construction phases, as described in this section. In our protocol, the initial shortest routing path is constructed by taking the ID and location information of the nodes and a backup routing path is also constructed side by side to maintain the path due to mobility of the nodes. Details of our LAMP algorithms are described as follows.

4.1 Route Search Phase

If a node of any piconet wants to transmit packet to another one, it has to go to the route search phase. It is assumed that the source node knows the ID of the destination in priori. Then, it floods a Route Search Packet (RSP) appending its own ID and LOC to the IFN field of the packet. Besides, ID of the destination node is also appended to the RSP and LOC of the destination is kept as NULL, as it is unknown to the source node. When the RSP is forwarded from one node to another, LOC and ID of all intermediate nodes are also appended to the IFN field of the packet.

Upon receiving an RSP, the master of the piconet forwards it to all of its bridge nodes and also the bridge nodes follow the same procedure by appending their own ID and location information to the respective IFN field of the packet. Ultimately, several RSPs are flooded at the destination through different possible routes from the source. Considering an example of the routing path $S \rightarrow M_1 \rightarrow A \rightarrow M_2 \rightarrow B \rightarrow M_3 \rightarrow C \rightarrow D$, as shown in Figure 1, the IFN set $\{S, M_1, A, M_2, B, M_3, C, D\}$ is constructed after the destination receives the RSP.

4.2 Route Reply Phase

In this phase, the final shortest and backup routing paths are constructed between the source and the destination. Due the mobility of the nodes, since there is every chance that the constructed route may be broken, the construction of backup path is highly essential to maintain the routing and to avoid the data loss. In our protocol, we construct a disjoint backup path along with the shortest path such that the two paths are not broken simultaneously. In case of mobility of nodes, the source node can use that disjoint backup path to replace the broken one without restarting the route search phase.

Upon receiving several RSPs through different routes, the destination node initiates this procedure. The destination node collects the location information of the source and all intermediate nodes between the source and itself from the ID and LOC fields of the RSP. Then it forwards the Route Reply Packet (RRP) to the next hop master/bridge node. The RRP has several different sub-fields in the payload field of the packet such as equation of ideal path (EIP), Final Forwarding Node (FFN) set that contains the list of nodes belongs to the current shortest routing path, and Final Backup Node (FBN) set that contains the list of nodes belongs to the current backup path.

In order to construct the final shortest and backup paths rapidly, the FFN and FBN each maintains the ID, LOC, BD_ADDR and CLK_offset of the nodes of current shortest routing and backup paths, respectively. It is to be noted that the destination node maps the ID of the nodes with their corresponding hop counts and only considers the packet with least number of hop counts out of all received RSPs. Then, it copies the order of ID and LOC pairs present in the IFN field of the RSP to the corresponding FFN field of the RRP and appends its BD_ADDR and CLK_offset to the corresponding FFN field. Thus, the FFN set $\{S, M_1, A, M_2, B, M_3, C, D\}$ is constructed. The destination node derives the EIP between the source and the destination and appends it to its RRP. In this phase, the destination node acts as if a source node and the RRP is routed along the same path as created during the route search phase. It is to be noted that each master knows its slave's location and ID. The backup path rule is executed to construct the disjoint backup path, the reduction rule is applied to reduce the path length by replacing some new nodes and the replacement rule is used to search the shorter path. The final shortest and backup paths between the source and the destination are obtained from the backup path, reduction and replacement rules as described below.

4.2.1. Backup Path Rule: The different steps of the Backup Path Rule are given as follows.

Step 1: Master node n_m scans EIP from the RRP and estimates the DIP for each of its slaves and itself.

Step 2: Master n_m verifies if itself or any of its slave n_l is LFN as per the definition 10 of Section 3.

Step 3: If any of its slave or itself satisfies the condition: It selects the LFN with minimum DIP value and copies the current FFN=$\{n_1, ..., n_d\}$ set to the FBN set.

Step 4: According to remaining routing path of the master n_m, n_m replaces the FFN=$\{n_1, ..., n_m, n_d\}$, where LFN is n_m, the FFN=$\{n_1, ..., n_m, n_l, n_d\}$, where the LFN= n_l or the FFN=$\{n_1, n_d\}$, where the LFN=n_1.

Step 5: Master node n_m executes the reduction rule for the FFN and FBN sequentially. Otherwise, only the current FFN is used for the reduction rule by the master.

4.2.2. Reduction Rule: The detail procedure of the reduction rule is explained as follows.

Step 1: Master verifies, if any of its slave node or itself can communicate with any two nodes, say n_i and n_j of FFN or FBN=$\{..., n_i, ..., n_j, …\}$ set, where $1 \leq i$ and $i+2 < j \leq k$.

Step 2: If the master or any of its slave satisfies the condition: It selects the node $n_{\min}$, which does not increase the number of common nodes between FFN and FBN and has least DIP.

Step 3: Nodes with index from n_{i+1} to n_{j-1} are replaced by the node $n_{\min}$ and new FFN or FBN={..., n_i, $n_{\min}$, n_j, …} set, where $1 \leq i$ and $j=i+2$ are stored in RRP.

Step 4: If node $n_{\min}$ is a bridge node and is a next hop of the routing path, the master appends its ID, LOC to the corresponding FFN or FBN fields. Otherwise, it appends the ID, LOC, BD_ADDR and CLK_offset of the node $n_{\min}$ to the corresponding FFN or FBN fields.

Step 5: After checking all node sets, the master applies the replacement rule for the FFN and FBN sequentially, if the FBN is not an empty set. Otherwise, only the current FFN is used for the replacement rule by the master.

4.2.3. Replacement Rule: The various steps of the replacement procedure are given as follows.

Step 1: Master checks CNT table to verify if any of its slave nodes is within communication range of its last forwarding node (LFN) and also with the next forwarding node in the FFN or the FBN.

Step 2: If so, it selects the slave node which does not increase the common nodes between FFN and FBN and has the least DIP.

Step 3: Master appends ID, LOC, BD_ADDR and CLK_offset of the slave node to the corresponding FFN or FBN fields instead of its own information.

Step 4: After checking all node sets, the master compares length of the shortest and backup paths if the FBN is not empty.

Step 5: If the backup path length is less than the shortest path length, the FFN and the FBN are exchanged.

If the destination node is a master or S/M bridge, it executes the above said three rules sequentially. Otherwise, it forwards the RRP to the next hop, which ultimately reaches to the source. Upon receiving the RRP, the S/S bridge node checks, if it is recorded in the FFN or FBN. If so, it appends its BD_ADDR and CLK_offset to the corresponding FFN or FBN fields and then forwards the RRP to the next hop of the routing path. Otherwise, it simply forwards the RRP to the next hop. However, the master or the S/M bridge nodes apply three rules sequentially upon receiving the RRP and then execute the same operations as the S/S bridge node. This process is continued until the source node receives the RRP. If the source node is a master or S/M bridge, it executes three rules sequentially. Then, it checks whether the shortest and backup paths are disjoint. If so, the source node obtains the final shortest and backup paths between the destination and itself in a reduced form. Otherwise, it only gets the final shortest path.

For example, as shown in Figure 3, destination node D does not execute the three rules, since it is a slave node. Thus, it only forwards the RRP to S/M bridge node C. Upon receiving the RRP, node C checks three rules sequentially, since it is a master. However, no other node qualifies the three rules and then node C appends its BD_ADDR and CLK_offset to the FFN field, since it is recorded in the FFN. Then, it forwards the RRP to master M_3. Master M_3 executes the backup path rule to check if any of its slave or itself can construct a backup path. It scans the EIP from the RRP and estimates the DIP for slave S_{31} and bridges B, C and itself. However, it finds that no node is the LFN and then it executes only reduction rule for the FFN set. By applying the reduction rule, Master M_3 checks the path connectivity to reduce the number of hops and finds that only bridge B can be connected with nodes S and M_3 to reduce the path length. Then, it selects bridge B, which does not increase the common nodes between FFN and FBN sets and has least value of DIP and deletes the information of nodes M_1, A and M_2 in FFN set. Since, bridge B is the next hop of the routing path, Master M_3 appends the information of bridge B to the FFN field and applies the replacement rule to check if any of its slaves can form the shorter route for the nodes FFN set.

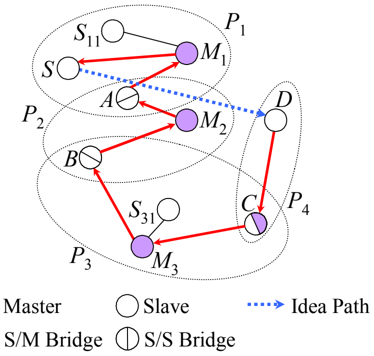

Figure 3: RSP is forwarded along the path from the destination D to the source S.

From its CNT, it finds that only slave S_{31} can be connected with nodes C and B. Hence, it selects slave S_{31}, which does not increase the common nodes between FFN and FBN and appends slave S_{31}'s information to the FFN field to replace master M_3. Since, FBN is empty, master M_3 does not compare the length of the shortest and backup paths and then forwards the RRP to the bridge node B. Bridge B checks that it is recorded in the FFN and appends its BD_ADDR and CLK_offset to the corresponding FFN field and forwards the RRP to master M_2. Now Master M_2 executes the backup path rule and estimates the DIP for itself and bridge nodes B and C. Then, it finds that only itself is the LFN and copies the current FFN={S, B, S_{31}, C, D} set to the FBN. Since, finding the new shortest path from the remaining routing path can help to reduce common nodes between FFN and FBN, master M_2 replaces the FFN={S, M_1, A, M_2, D} set according to the remaining routing path $A \rightarrow M_1 \rightarrow S$ of master M_2. Consequently, S and D become the common nodes and so as the current shortest and backup paths become disjoint. Then, master M_2 executes the reduction rule for the FFN. It finds that both bridges A and B can connect to nodes S and M_2 to reduce the path length. Since, bridge B increases the number of common nodes between FFN and FBN, master M_2 selects bridge A which does not increase the common nodes between FFN and FBN and deletes the information of node M_1 from FFN. Then, master M_2 only appends the information of bridge A to the FFN field, since bridge A is next hop of the routing path. After that, master M_2 executes the reduction rule for the FBN and finds that no node can reduce the backup path length since the shortest and backup path cannot be disjoint.

After master M_2 has checked all node sets, it executes the replacement rule for the FFN and the FBN sequentially and finds that no slave node can satisfy the condition since the shortest and backup path cannot be disjoint. Then, master M_2 estimates that the shortest path length is less than the backup path length. Therefore, the FFN and FBN should not be exchanged. Next, master M_2 appends its BD_ADDR and CLK_offset to the FFN field since it is recorded in the FFN and then forwards the RRP to bridge A. Bridge A checks that it is recorded in the FFN and appends its BD_ADDR and CLK_offset to the corresponding FFN field and forwards the RRP to master M_1. By applying the backup path rule, master M_1 finds that only itself is the LFN and then copies the current

FFN={S, A, M_2, D} set to the FBN and replaces the FFN={S, M_1, D} set. After that, it executes the reduction rule for the FFN and FBN sequentially and finds that no node can satisfy the rule. Master M_1 continuously executes the replacement rule for the FFN and FBN sequentially and still finds that no node can satisfy the condition. Then, master M_1 estimates the FFN and FBN is not exchanged since the shortest path length is less than the backup path length. Finally, master M_1 appends its BD_ADDR and CLK_offset to the FFN field and then forwards the RRP to source S. Since source S is a slave, it does not execute the three rules. Finally, it finds the final shortest and backup paths are disjoint and completes the route reply phase. For different nodes in the routing path, the corresponding FFN and FBN are shown in Figure 4.

Node	Corresponding FFN	Corresponding FBN
D	$S, M_1, A, M_2, B, M_3, C, D$	
C	$S, M_1, A, M_2, B, M_3, C, D$	
M_3	S, B, S_{31}, C, D	
B	S, B, S_{31}, C, D	
M_2	S, A, M_2, D	S, B, S_{31}, C, D
A	S, A, M_2, D	S, B, S_{31}, C, D
M_1	S, M_1, D	S, A, M_2, D
S	S, M_1, D	S, A, M_2, D

Figure 4: The FFN and FBN set of each node along the routing path.

4.3. Route Construction Phase

The route construction phase is executed after the route search and route reply phases are over. In this phase, source node sends the final FFN and FBN to the next forwarding nodes along the shortest and backup paths so that next forwarding nodes can correctly construct the final shortest and backup paths. Source node verifies the number of links between itself and the next forwarding nodes. If only one link is established, source node enters to page state to construct another link. However, if no link is established, source node enters to page state to construct the link of the shortest path. After constructing the link, source node enters to page state again to construct the link of the backup path. Upon receiving the final FFN and FBN sets, the forwarding nodes continuously send them to the next ones. Then, the forwarding node enters to page scan state if no link is existed between itself and the last node and completes the link construction. Each forwarding node executes the same operations to check the existence of link between itself and the next hope node. Upon receiving the final FFN and FBN, finally the destination node follows the same procedure to check its link with the last forwarding node. If only one link is established, destination node enters to page scan state to construct another one. However, if no link is established, destination node enters to page state to construct the link of the shortest path. Once the construction of the shortest path is over, it enters to page scan state again to finish the construction of the backup path. Then, the forwarding node of the shortest path sends the final FFN set to the next one after constricting the link. Finally, the forwarding node constructs the link with the destination to finish the construction of the shortest path and destination node does not enter to page scan state again after constructing the link. Moreover, the nodes of the shortest path actively inform to the source node to transmit data through the backup path while the shortest path is broken.

For example, as shown in Figure 5, let there exists disjoint shortest path $S\rightarrow M_1\rightarrow D$ and backup path $S\rightarrow A\rightarrow M_2\rightarrow D$. First source S sends the final FFN and FBN sets to nodes M_1 and A. Then, it checks existence of link and enters to page state to construct the link with node A. Upon receiving the final FFN and FBN sets, master M_1 forwards the final FFN and FBN sets to bridge A and then verifies the existence of links. Then, it enters to page state to construct the link with the destination D. Bridge A continuously sends the final FFN and FBN sets to Master M_2 and then checks the links existence. It enters to page scan state to finish the link construction with source S. Therefore, source S and bridge A become the master and the slave in the newly formed piconet, respectively. Now, Master M_2 sends the final FFN and FBN sets to destination D and checks the links existence. Then, it enters to page state to construct the link with destination D. Upon receiving the final FFN and FBN sets, destination D verifies the existence of links and then enters to page scan state to finish the construction of the shortest path. Therefore, destination D becomes the slave of master M_1. The destination node D enters to page scan state again to finish the construction of backup path and becomes the slave of master M_2. It is to be noted that the number of hops of the shortest path between the source and the destination are reduced to 2, as shown in Figure 5, which are least as compared to LORP [6] and RVM [5]. Besides, source S can use the backup path $S\rightarrow A\rightarrow M_2\rightarrow D$ to continuously transmit data if the shortest path $S\rightarrow M_1\rightarrow D$ is broken due to mobility.

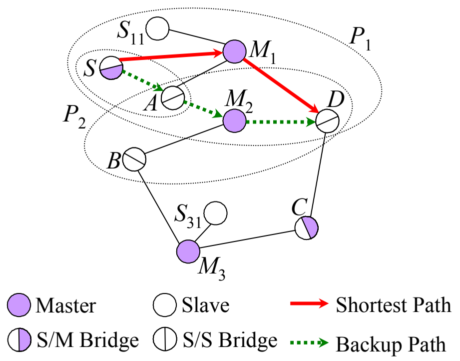

Figure 5: Final route construction phase in LAMP.

V. SIMULATION

In this section we rigorously analyze the performance of our mobility based routing protocol and compares our location aware mobility based routing protocol (LAMP) with some standard Bluetooth routing protocols such as RVM [5], LORP [6] and BMP [7].

In our work, we use C++ programming to simulate our protocol. In our simulation, initially a connected scatternet with fixed numbers of 100 Bluetooth nodes are taken, which are randomly distributed over a squared area of 50m×50m and 50 pairs of source and destination nodes are randomly selected to construct the route using RVM, LORP, BMR and LAMP. The communication range and mobility model is set by 10m and random waypoint model, respectively. The Constant Bit Rate (CBR) model is used to generate the traffic load for each route and the traffic arrival rate is kept at 100 Kbps. The energy consumption for transmitting or receiving one bit of data is set by 0.0763×10^{-6} J. In RVM, a new routing path is searched when the current route is broken. On the other hand, new shortest and backup paths in LAMP and LORP are searched, if the backup path is broken. 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 routing path length is estimated for different numbers of mobile nodes. In BMR, which is a table driven routing protocol, the master of the source knows to which piconet the destination belongs and finds the shortest path to destination. If the scatternet is changed due to nodes mobility, the up-to-date information is notified to each master. Thus, the master of the source can select the new shortest path when the current shortest path is broken.

As shown in Figure 6, the average hop counts for the different number of mobile nodes are simulated with different routing protocols that we have considered. The average speed of each mobile node is considered as 1.5m/s in the simulation. From the simulation results, it is observed that the average hop counts of the proposed protocol are less than that of RVM and LORP and similar to BMR. In RVM, LORP and LAMP, new and worse routes are found after reestablishing the routes as a result of which average hop counts are raised in these protocols, when the number of mobile nodes is increased. The route length of LORP and LAMP is less than that of RVM, since they try to shorten the route length while constructing the shortest and backup paths. Moreover, LAMP can reduce efficiently the route length by applying reduction and replacement rules. However, LAMP in some situations cannot construct the shortest path in order to construct the disjoint backup path. Therefore, the route length of LAMP is a little higher than BMR, which can select the new and worse shortest path.

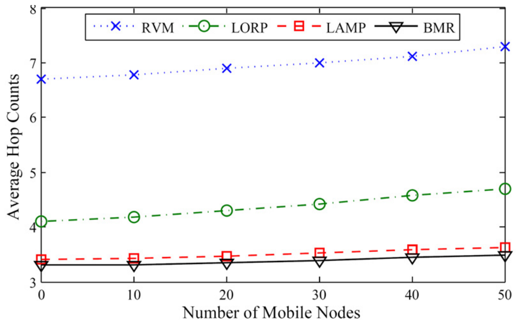

Figure 6: The average hop counts in different protocols for the different number of mobile nodes.

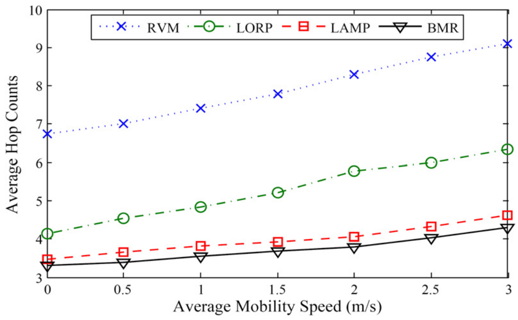

Figure 7: The average hop counts in different protocols for the different average mobility speed.

The average hop counts for the different average mobility speed of the mobile nodes are shown in Figure 7. All nodes in the scatternet are mobile in the simulation. It is observed that the proposed protocol gives tremendous improvement in terms of hop counts for different average mobility speed and is closer to BMR. In RVM, LORP and BMR, they initialize their protocols to find new and worse routing paths when search the routing paths. Since, the new shortest path in RVM and BMR and the backup path or the new shortest and backup paths in LORP are longer than the broken route, the average hop counts of all protocol are increased while the average mobile speed is added and the link of the route is broken more rapidly. However, the reduction and replacement rule of LAMP can significantly improve the hop counts of the shortest and backup paths. Therefore, the route length of LAMP is increased slightly than BMR, which can often select the shortest route.

Figure 8 investigates the average number of control packets by varying average mobility speed. It is founded that RVM, LORP and LAMP protocols outperform BMR and the control traffic of all protocols is raised when average mobility speed is increased. Since, the higher average speed of the mobile nodes results that scatternet topology is changed frequently; BMR creates large number of control packets to maintain the information of the scatternet topology than RVM, LORP and LAMP. Moreover, since the higher average speed of the mobile nodes also causes large number of broken links, RVM requires creating more control packets to reconstruct the routes than LORP and LARP, which have constructed the backup paths. Furthermore, LORP is higher than LAMP when the average mobility speed is larger than 2m/s. This is because there are more nodes to join the routes construction in LORP when the route length becomes longer such that LORP creates more control packets than LAMP.

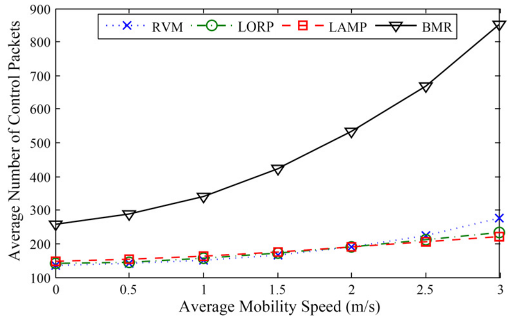

Figure 8: The average number of control packets in different protocols for the different average mobility speed.

VI. Conclusion

In this paper, we propose a location aware mobility based routing protocol for an ad hoc Bluetooth network. We consider location information of the nodes to minimize the number of hop between the source and the destination. Besides, we propose algorithm how to construct the backup paths and to maintain the shortest routing path due to mobility of nodes. From the simulation result we find that our protocol outperforms in terms of energy and bandwidth consumption to RVM, LORP and BMR. Since, our protocol supports mobility to construct routing path, it can used in different mobility based applications in shopping malls, supermarkets and mobile e-commerce scenarios.

References

[1] The Bluetooth Specification, *http://www.bluetooth.org*

[2] A. B. Mirza and J. K. Pollard, "Dynamic Source Routing in Bluetooth Personal Area Networks," http://www.ducati.doc.ntu.ac.uk/uksim/journal/Vol-5/No 1&2/MIRZA.pdf.

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

[4] S Z. Jian, Z. Fei and W. Bin,"Cluster Based Routing in Bluetooth Ad-hoc Network," in *Proc. of International Conference on Wireless Communications Networking and Mobile Computing*, Sep. 2005.

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

[6] C. Y. Chang, G. J. Yu, C.-F. Lin, and T.-T. Wu, "Relay Reduction and Route Construction for Scatternet over Bluetooth Radio Systems," *Journal of Network & Computer Applications*, vol. 30, no. 2, Apr. 2007, pp. 728-749.

[7] O. Song, C. Lim, and C.-H. Choi, "Mobility Management in Bluetooth ad hoc networks," in *Proc. of 14th Joint Conference on Communications & Information,* 2004.

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

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