Wireless Pers Commun
DOI 10.1007/s11277-008-9498-x

A Location Aware Mobility based Routing Protocol for the Bluetooth Scatternet

Sheng-Wen Chang · Prasan Kumar Sahoo · Chih-Yung Chang

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.

Keywords Bluetooth · Scatternet · Routing protocol · Mobility · Location aware

1 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. It is the core representative of wireless personal area networks (WPAN), which is being further evolved by IEEE 802.15 [2] task group and operates

S.-W. Chang · C.-Y. Chang
Tamkang University, Tamsui, Taipei, Taiwan
e-mail: swchang@wireless.cs.tku.edu.tw

C.-Y. Chang
e-mail: cychang@mail.tku.edu.tw

P. K. Sahoo (✉)
Vanung University, Chung-Li, Taiwan
e-mail: pksahoo@msa.vnu.edu.tw

in the unlicensed 2.4 GHz ISM band. The Bluetooth Personal Area Networks (PANs) are becoming increasingly popular in connecting people, their personal devices and surrounding networks of Bluetooth-enabled handheld devices. 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 m. 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 1,600/s, 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.

As per the Bluetooth specification, a device of any piconet has to be in any one of the Standby, Intermediate, or Connection state at a time. Besides, a device can stay in one the following seven sub-states: inquiry, inquiry scan, inquiry response, page, page scan, page response, and master response. Even after joining to a piconet, the Bluetooth devices can move in and out of these states, and sub-states through commands from the Bluetooth link manager or from internal signals in the link controller. The performance of the connected scattered is highly relied on the number of bridge nodes present in it. Scatternet that contains a large number of bridge nodes will be benefited from the advantages including low probability of disconnection, short routing path and fast flooding, but will suffer from the drawbacks including consumption of active member address, creating a large amount of packets in flooding and difficulties in synchronization among the piconets. Moreover, a higher degree of relay nodes have to switch frequently among the participated piconets, increasing the difficulties of scheduling and the packet loss probability. To mitigate such problems, the relay reduction and route construction protocol (LORP) [3] proposes how to retain the suitable relays and remove other nodes.

Bluetooth scatternet is considered as a special type of ad-hoc network. So the routing protocols for Bluetooth can be categorized into two types, such as: table driven and on-demand routing protocols. In the table driven routing protocols [4], each node actively maintains a routing table irrespective of message to send or not. The main disadvantage of such protocol is the maintenance overhead of the routing table at each node. Also the table driven protocol may require more memory, as the size of the routing table is proportional to the size of the network. In case of the on-demand routing protocols [5], a node first floods a query message to learn the route to the destination before it can send a message. Some drawbacks in an on-demand routing protocols are due to the delay incurred by the query phase and flooding of the query signals. A position based routing scheme is analyzed in [6], in which a message is to be sent from a source node to a destination node in a given wireless network. The destination node is known and is addressed by means of its location. However, this routing scheme is only applicable to the mobile ad hoc networks consist of wireless hosts that communicate with each other in the absence of a fixed infrastructure. As analyzed in [7], all ad hoc networks routing protocols can be classified as proactive or reactive schemes. Proactive being when the network topology is always known by each node through regular refresh, and reactive being when the current topology is only found when a node needs to send data. However, the ad hoc network is normally decentralized, where all network activities including discovering the topology and delivering messages must be executed by the nodes themselves. In contrast, all slaves of a scatternet are controlled by the masters of the respective piconets which have different hopping sequences and routing of information from one slave to another can be passed through the master only. Hence, the standard ad hoc routing protocols cannot be applicable to Bluetooth scatternet routing.

A blue-tree scatternet formation algorithm [8] is proposed to build a self routing scatternet to minimize the routing overhead. But, the authors do not mention how to construct the scatternet, if nodes are not within the proximity of each other. Moreover, the number of hops between the source and the destination nodes of this so called blue-tree based scatternet are more, which incur more delay to dispatch the packets. The link formation time of current Bluetooth specification is too long for mobile devices. Hence, a dynamic source routing scheme [9] of Bluetooth scatternet is proposed by introducing the new packet format and network layer. The paper proposes that the source device delivers P-REQ page packet to find the destination and the destination node appoints each node either as a master or slave. Upon receiving the P-REQ packets, the destination node passes the P-REP packet through the nodes. 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 [10] 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 [11] 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 [12] 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.

To the best of our knowledge, no work considers the location information of the nodes to shorten the routing path and to maintain it due to mobility of the nodes. However, many users-positioning solutions have been proposed recently, though they are based on the specialized devices not supported by commercially available data terminals [13–15,17]. Such location aware protocols [16] propose how to establish a cooperative location network among the Bluetooth devices and intend to cover the two-dimensional target areas. Since, Bluetooth is a short-range communication technology; we feel that its indoor applications are more than outdoor applications. The typical example is the m-commerce scenario [17,18], in which customers walk around a large commercial area or shopping mall carrying wireless PDA and Bluetooth enabled wireless devices. In such scenarios, a customer is supposed to purchase items, request information and also receives store coupons and advertisements. It is to be noted that now-a-days the mobile phones and PDA are equipped [19] with Radio Frequency Identification (RFID) [20,21], which is highly useful to m-commerce scenarios. 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 [22], 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, Bluetooth Location Networks (BLN) [23] transmits location information to the service servers without user participation and its base technology is supported by the existing commercial handhelds [24]. 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 Sect. 2. Section 3 describes the system model and definitions of few related terms. Our Location Aware Mobility based routing Protocol (LAMP) is discussed in Sect. 4 of the paper. Enhancement algorithms to our mobility based routing path are proposed in Sect. 5. Performance evaluation of our protocol and comparison of the results with few standard routing protocols are discussed in Sect. 6 and concluding remarks are made in Sect. 7.

2 Related Work

In this section, we analyze some standard routing protocols for the Bluetooth ad hoc networks. As discussed in Sect. 1, though several protocols propose the routing mechanism for the Bluetooth technology, we consider here the Routing Vector Method (RVM) [25], relay reduction and route construction protocol (LORP) [3], and Bluetooth Master-Managed Routing (BMR) [26] protocol, as they have special relation to our proposed work.

The Routing Vector Method (RVM) [25] 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 Fig. 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 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.

A so-called relay reduction routing protocol (LORP) [3] 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 Fig. 1, though node S and B are within communication range (10 m) 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.

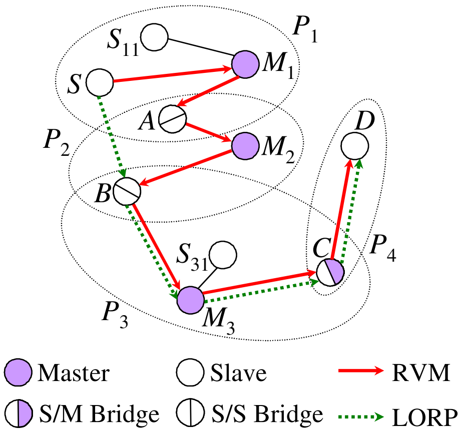

Fig. 1 Example of routing paths constructed by RVM and LORP between the source S to the destination D

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.

A table-driven routing protocol named as Bluetooth Master-Managed Routing (BMR) [26] 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 Fig. 2, since node M_1 has more neighbors and node D is a neighbor of node M_1 when the scatternet is formed, 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 Daccording 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. Besides, we propose the route enhancement algorithms that help to construct and optimize the routing paths due to mobility of the nodes.

3 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 [22] or Bluetooth Location Networks (BLN) [23]. The source node of any piconet can communicate with the destination node of another

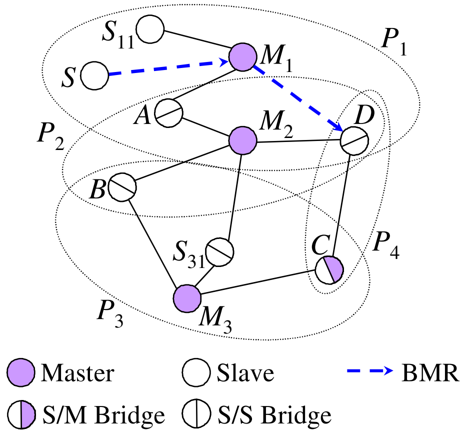

Fig. 2 Example of the routing path constructed by BMR between the source *S* to the destination *D*

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 Sect. 4.

3.1 Definition 1: 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.

3.2 Definition 2: Location (LOC)

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

3.3 Definition 3: 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 Sect. 4 is termed as *Initial Forwarding Node (IFN) set*.

3.4 Definition 4: 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 Sect. 4 is termed as *Final Forwarding Node (IFN) set*.

3.5 Definition 5: 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 Sect. 4 is termed as *Final Backup Nodes (FBN) set*.

3.6 Definition 6: 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 $\leq 10\,\text{m}$ (typical Bluetooth communication range) and therefore it can construct a link with the destination.

3.7 Definition 7: 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.

3.8 Definition 8: 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*).

3.9 Definition 9: 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*).

4 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. The format of the RSP is shown in Fig. 3, which is similar to the Bluetooth baseband packet, where the payload field contains LOC and ID of the source and destination. When the RSP is forwarded from one node to another, LOC and ID of all intermediate nodes are also appended to the IFN

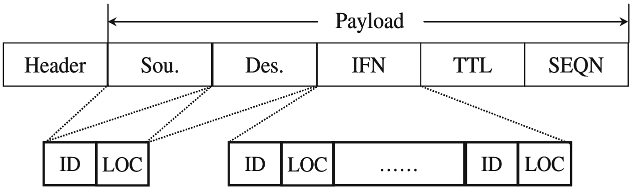

Fig. 3 Format of the Route Search Packet

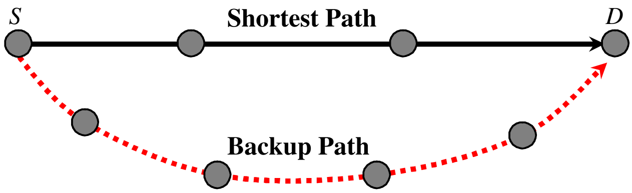

Fig. 4 An example of shortest and backup path between the source S and destination D

field of the packet. The time to live (TTL) field in IFN indicates the life of the RSP, which is dropped after the TTL duration is expired. Each packet contains a sequence number in the SEQN field of the RSP to maintain the uniqueness 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 $S \to M_1 \to A \to M_2 \to B \to M_3 \to C \to D$, as shown in Fig. 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.

As shown in Fig. 4, an example of shortest and backup routing paths between the source S and destination D is given.

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 seven different sub-fields in the payload field of the packet such as locations and IDs of the source and destination, equation of ideal path (EIP), Final Forwarding Node (FFN) set that contains the list of nodes belongs to the current shortest routing path, Final Backup Node (FBN) set that contains the list of nodes belongs to the current backup path, time to live (TTL), and sequence (SEQN) field. The format of the RRP is shown in Fig. 5.

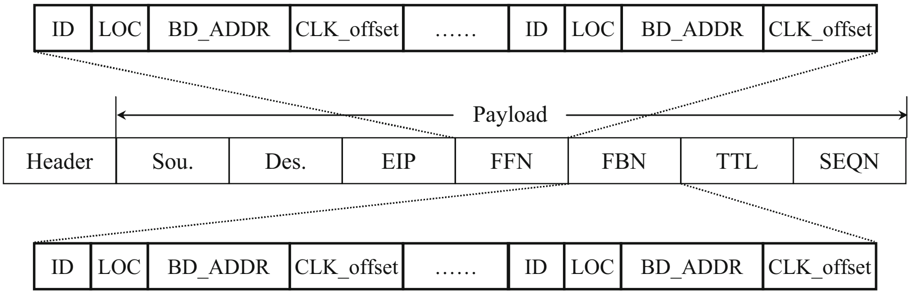

Fig. 5 Format of the Route Reply Packet (RRP)

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 Sect. 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 = $\{n_1, \ldots, n_d\}$ set to the FBN set.
- Step 4: According to remaining routing path of the master n_m, n_m replaces the FFN = $\{n_1, \ldots, n_m, n_d\}$, where LFN is n_m, the FFN = $\{n_1, \ldots, n_{m,} n_l, n_d\}$, 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 $= \{\ldots, n_i, \ldots, n_j, \ldots\}$ 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 $= \{\ldots, n_i, n_{\min}, n_j, \ldots\}$ 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 is 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 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 Fig. 6, 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

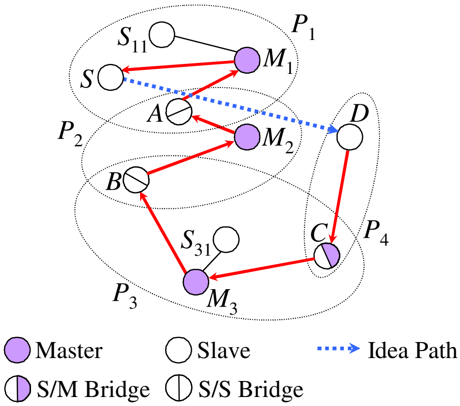

Fig. 6 RSP is forwarded along the path from the destination D to the source S

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.

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. Bridge 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

Table 1 The FFN and FBN set of each node along the routing path

Node	Corresponding FFN	Corresponding FBN
D	$S, M_1, A, M_2, B, M_3, C, D$	
C	$S, M_1, A, M_2, B, M_3, C, D$	
$M3$	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

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 Table. 1.

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

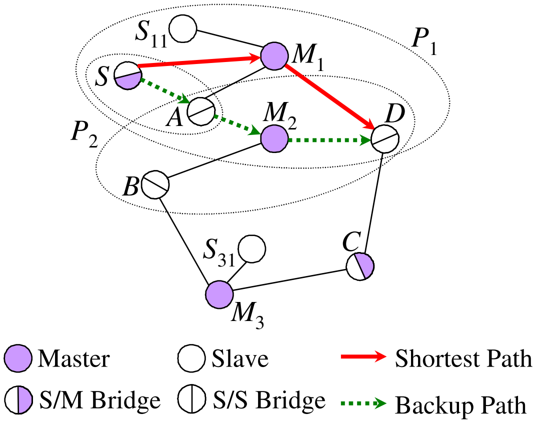

Fig. 7 Final route construction phase in LAMP

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 Fig. 7, 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 Fig. 7, which are least as compared to LORP [3] and RVM [25]. Besides, source Scan 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.

5 LAMP Enhancement Scheme

In order to enhance the route construction phase and optimize the route length, we propose here the route enhancement policies. In the Route Construction Phase, if a node enters to page scan state, it may participate the construction of the routing path. However, if a master node enters to page scan state, two problems may arise. The first problem is the network bottleneck and the second one is the limitation of the number of slaves. Hence, we describe here the problems and propose the route optimization and piconet combination operations as described below.

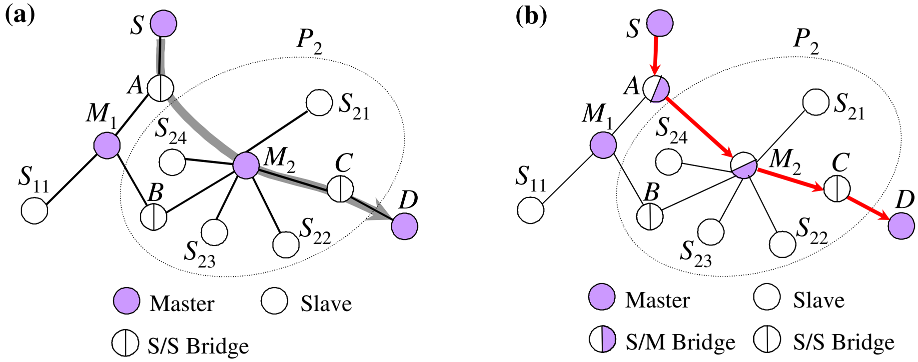

Fig. 8 The example of network bottleneck problem (**a**) M_2 enters to page scan state in the route construction phase and slaves of P_2 cannot transmit data. (**b**) M_2 plays a slave role with A as its master and halts data transmission of all slaves of P_2

5.1 Network Routing Problem

The network routing problem is encountered in the scatternet due to limitation of the number of slaves and change of role of a S/M bridge. The data transmission is blocked due to unavailability of a master node through which data should be transmitted. Analyzing all possible problem of the routing, we describe here the bottleneck and limitation of slave problems as follows.

5.1.1 Bottleneck Problem

As we know, few nodes of the scatternet serve as the M/S bridge nodes and they have to change their role during the route construction phase. A master node may enter to page scan state to construct a link with the node of another piconet that has entered to page state. Then, the master joins to the new piconet and plays a slave role when the link is constructed. In this case, the master has to suspend its operation of the original piconet and at that moment, if the slaves located in the original piconet have large number of data to send, the master cannot provide service to them, thereby causing the network bottleneck. For example, as shown in Fig. 8a, S and D are the source and destination node, respectively. The gray line represents the routing path, which will be constructed. Since, M_2 participates the route construction phase, it enters to page scan state. At the same time, the slaves in P_2 cannot transmit data through M_2, thereby decreasing throughput of the piconet. Once the routing path is constructed as shown in Fig. 8b, M_2 becomes a S/M bridge and serves as a slave role in the piconet, where it was serving as a master. Besides, when M_2 plays the slave role with its master as A, the slaves in P_2, cannot transmit data to other slaves of the same or different piconet.

5.1.2 Limitation of Slave Problem

In route construction phase, nodes joining the route construct a link with the next node of the route. If a node is playing master role and is connected with 7 active slaves, it cannot be connected with other new slaves. As a result of which, the construction of routing path is failed. As shown in Fig. 9, the gray line represents the routing path which will be constructed and therefore M_2 establishes a link with A in the route construction phase. However, M_2 is

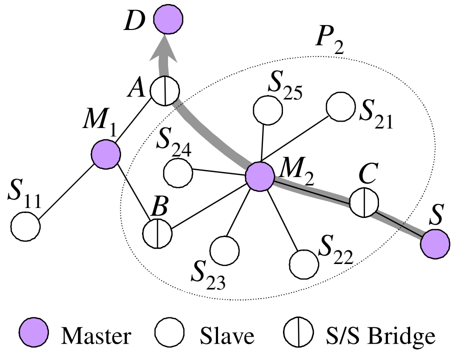

Fig. 9 M_2 cannot establish a link with A due to limitation of nodes

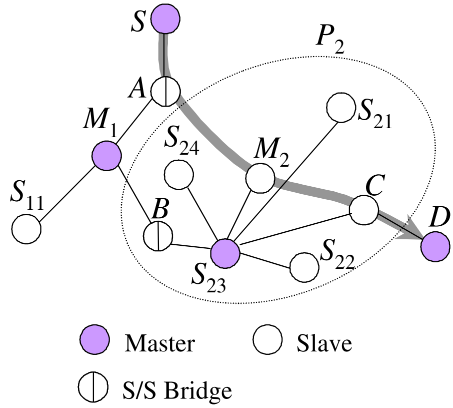

Fig. 10 Example of takeover operation

already connected with 7 active slaves in P_2 and therefore cannot be connected to A. As a result of which, the construction of routing path is failed.

If a master participates in the route construction phase, since two problems mentioned above are caused due to improper role of the node and limitation of the number of slaves, we use the role switch operations or the park operation to adjust the structure of the piconet in the route reply phase. The role switch operations can be applied to solve the two problems but the park operation merely deals with the second problem as described below.

5.1.2.1 Takeover Operation. The master that faces the network bottleneck or limitation of slave problem can apply the takeover operation in which the master changes its role to an idle slave to communicate with other slaves of the piconet. As shown in Fig. 10, M_2 executes the takeover operation when constructing link with A and notifies S_{23} to serve as the master. Then, other slaves in P_2 connect to master S_{23}. As a result of which, M_2 becomes the slave and then can establish the link with A. Consequently, the network bottleneck problem is solved.

5.1.2.2 Split Operation. In this operation, the master executes the split operation to form the new piconet and then apply the takeover operation to reduce the number of connected slaves with the new master. As shown in Fig. 11, M_2 executes the split operation when it constructs the link with A and informs S_{23} to create the new piconet P_3 so that M_2 can become S/M bridge. In order to reduce the traffic overhead of M_2, it applies takeover operation so

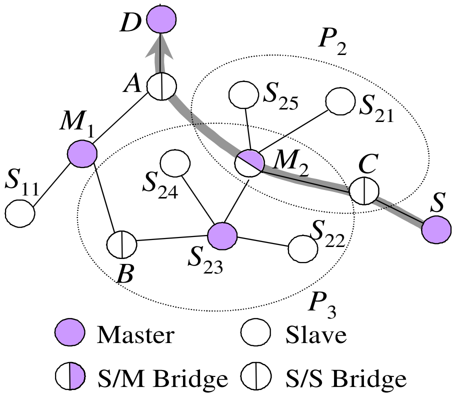

Fig. 11 Example of split operation

that some slaves like S_{22}, S_{24} and B can participate with piconet P_3 based on the location information. The split operation not only reduces the overhead of M_2 but also prevents M_2 from encountering the limitation of number of slaves in route construction phase.

5.1.2.3 Park Operation. This operation is also used to solve the limitation of slave problems. The park operation of the Bluetooth technology is used to allow an active salve to enter sleep mode so that a new slave can be connected and hence the route can be construed. Two policies are developed to establish the new link between the new slave and master. The first one is that master parks an idle active slave based on its traffic record when it constructs link with the next forwarding node. As shown in Fig. 12a, M_2 parks the idle slave S_{25} which has less traffic in the past when it constructs link with A and then can construct link with A. On the other hand, the second approach is to park the original bridge node during construction of links with the next forwarding node and then can establish a new link to new bridge which will be the next forwarding node in the route. As shown in Fig. 12b, M_2 parks bridge B when it constructs link with A and then constructs the link with A so that connection of P_2 and P_1 is maintained.

5.2 Piconet Combination Operation

In route construction phase, since link establishment of the routing path is based on the page/page scan mechanism, it forms new piconets. For example, as shown in Fig. 13a, there exists two piconets P_1 and P_2 in the original scatternet and slave S_{12} which connects to master M_1, and constructs link with master M_2 to form the new piconet. Then, S_{12} and M_2 become S/M bridges. In order to solve this problem, we use piconet combination operation to combine two piconets to single one. If a slave or an S/S bridge constructs a link with a master or an S/M bridge and form a new piconet, the master or the S/M bridge executes piconet combination operation to eliminate the newly formed piconet when finishing the link construction with a slave or an S/Sbridge. Thus, the newly formed piconet can be combined to the piconet which has existed. Moreover, the slave or the S/S bridge and the master or the S/M bridge serves as the slave and a master in the combined piconet, respectively. As shown in Fig. 13b, M_2 executes the piconet combination operations to combine its and newly formed piconets after finishing the link construction with slave S_{12}. Therefore, the newly formed piconet is eliminated. Moreover, S_{12} and M_2 become the S/S bridge and the master, respectively.

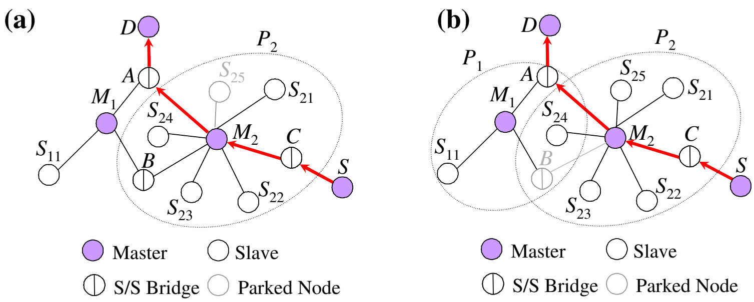

Fig. 12 Example of park operation. (**a**) M_2 parks the idle slave S_{25} which has less traffic flow in the past and then can connect to A. (**b**) M_2 parks bridge B and then constructs the link to A so that connection of P_2 and P_1 is maintained

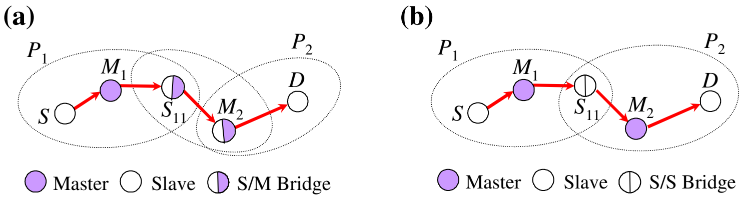

Fig. 13 Example of the piconet combination operation. (**a**) Slave S_{12} constructs the link with master M_2 to form the new piconet. (**b**) M_2 executes the piconet combination operation to combine piconets

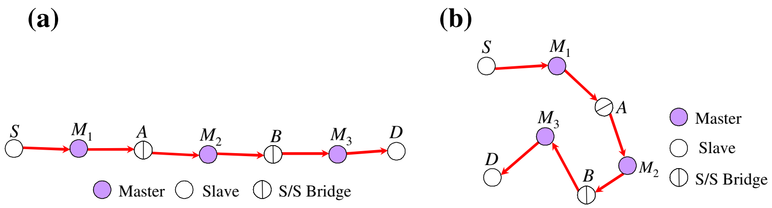

Fig. 14 The basic idea of route optimization. (**a**) The original topology of the route. (**b**) The new topology of the route

5.3 Route Optimization Operation

Let us assume that the routing path is the same as shown in Fig. 14a even after the mobility of nodes. Then, as shown in Fig. 14b, as node M_3 moves into M_1's communication range, the route length can be further reduced. However, since node M_1 and M_3 have different hopping sequence, M_1 cannot find M_3's existence.

In order to optimize the route, the basic idea is to piggyback the new location of each forwarding node in the data packet. Each node checks the locations of all forwarding nodes and applies path reduction operation, if it finds a new forwarding node in its communication range. However, there may exist more than one node executing the path reduction operation and hence causes the independent path problem. Therefore, when nodes in the route receive the new location of other node, they check its location and determine if it can reduce the routing path and record the shorter path information in data packet. After receiving data packet, the destination node notices the nodes in the shorter routing path to construct the connection.

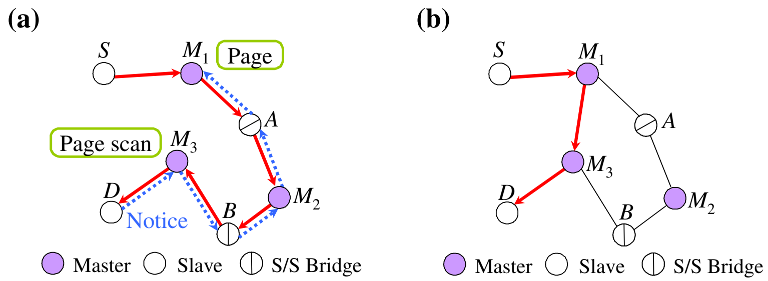

Fig. 15 The example of route optimization. (**a**) Node D finds there exist the shorter routing path and notifies $M_3$and M_1 to enter page scan and page states, respectively. (**b**) M_1 constructs connection with M_3 and hence the routing path can be optimized

Since nodes in the route move slowly, they will not create any weak link and the route is broken. Moreover, each forwarding node periodically puts its location in the data packet so that the data packet always does not store extra information. For example, as shown in Fig. 15a, node M_3 finds M_1 in its communication range and routing path can be reduced to $S \rightarrow M_1 \rightarrow M_3 \rightarrow D$. Hence, it puts the path reduction information of M_1, M_3 in the data packet. Upon receiving the data packet, node D finds that there exist the shorter routing path and notices M_3 and M_1 to enter to page scan and page states, respectively. Finally, as shown in Fig. 15b, M_1 constructs the connection with M_3 and hence the routing path can be optimized.

6 Performance Evaluation

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 [25], LORP [3] and BMP [26].

6.1 Simulation Setups

In our work, we use C++ programming to simulate our protocol. The parameters used in our simulation are listed in Table 2. 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 $50\,\text{m} \times 50\,\text{m}$ and 50 pairs of source and destination nodes are randomly selected to construct the route using RVM, LORP, BMR and LAMP. 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.

Table 2 Simulation parameters

Parameter	Value
The number of nodes	100
Network size	$50 \times 50\,m^2$
Communication range	10 m
Pairs of source and destination	50
Traffic model	CBR
Traffic arrival rate	100 kbps
Energy consumption	0.0763×10^{-6} J/bit
Mobility model	Random waypoint model

6.2 Simulation Result

It is to be noted that the proposed LAMP considers the mobility of nodes to construct the backup path and optimizes the route. Therefore, performance of LAMP is examined in terms of average number of hop counts, average number of control packets, total bandwidth consumption ratio and total energy consumption ratio based on the number of mobile nodes and average mobility speed. The simulation results are given in detail, as follows.

6.2.1 Average Hop Counts

As shown in Fig. 16, 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 1.5 m/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.

The average hop counts for the different average mobility speed of the mobile nodes are shown in Fig. 17. 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.

6.2.2 Control Packets Overhead

In our simulation, we have analyzed the average number of control packet for different number of mobile nodes as shown in Fig. 18. It is observed that RVM, LORP and LAMP protocols

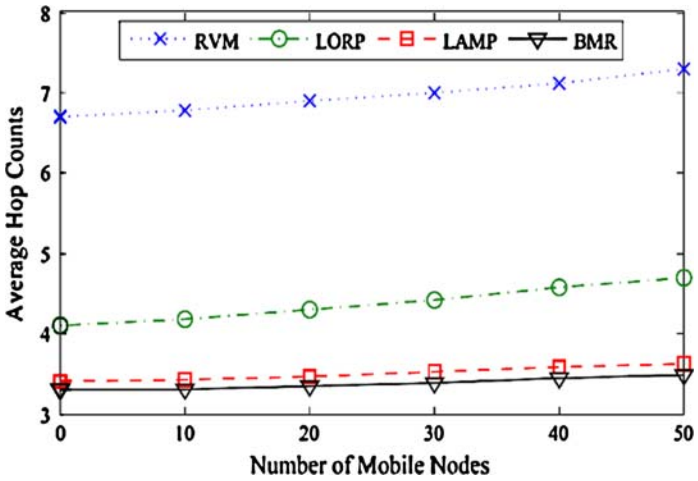

Fig. 16 The average hop counts in different protocols for the different number of mobile nodes

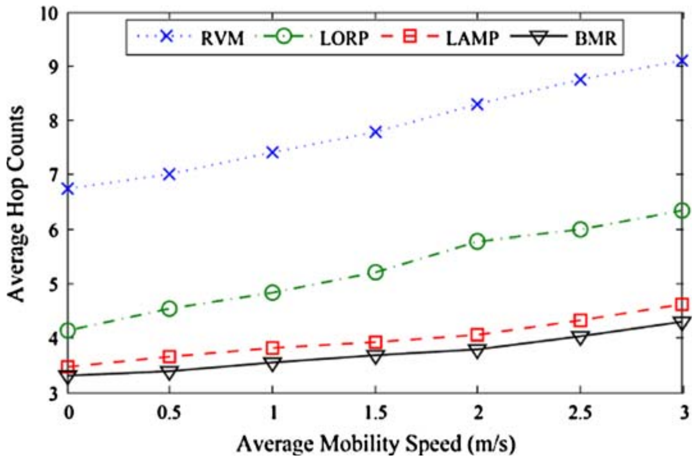

Fig. 17 The average hop counts in different protocols for the different average mobility speed

outperform in terms of the number of control packet as compared to BMR. This is because each master in BMR has to maintain the up-to-date information of the scatternet topology in order to construct the shortest path. Thus, BMR costs more control packets so that each master has the same and newest information of the scatternet topology. However, more mobile nodes result that the scattenet structure is easily changed. As a result, BMR requires more and more control packets when the mobile nodes are increased. On the other hand, RVM, LORP and LAMP also require more number of control packets to reconstruct the route since the higher number of mobile nodes causes that the route is easily broken. Although the backup paths constructed by LORP and LAMP can avoid to immediately reconstruct the routes and hence reduce the control packets, RVM does not cost more control packets to reconstruct the route when broken routes are fewer. Furthermore, LORP and LARP use additional control packets to shorten the routing path. As a result, RVM outperforms to LAMP and LORP. Besides, since there are fewer nodes to join to the routes construction in LORP when the

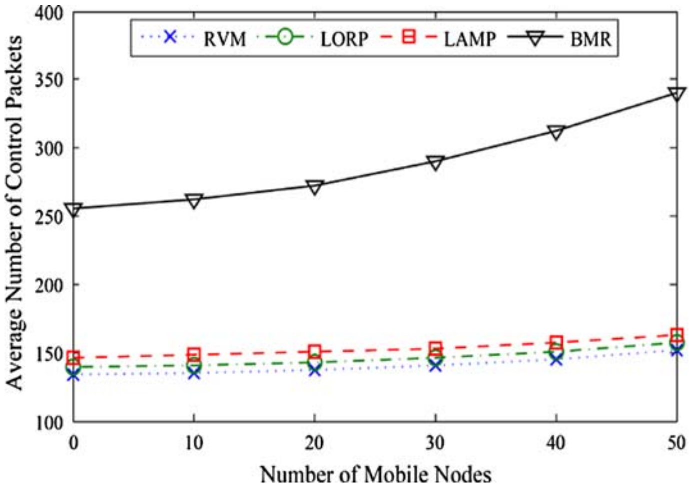

Fig. 18 The average number of control packets in different protocols for the different number of mobile nodes

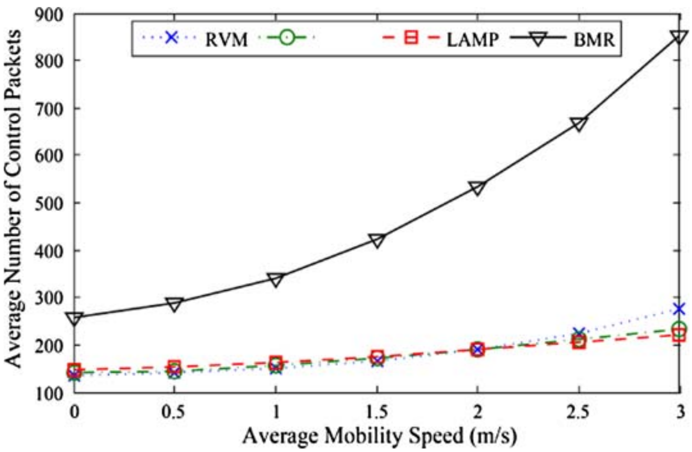

Fig. 19 The average number of control packets in different protocols for the different average mobility speed

route length is shorter, LORP costs fewer control packets to construct the routes and is lower than LARP.

Figure 19 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 2 m/s. This is because there are more nodes to join the routes construction in LORP

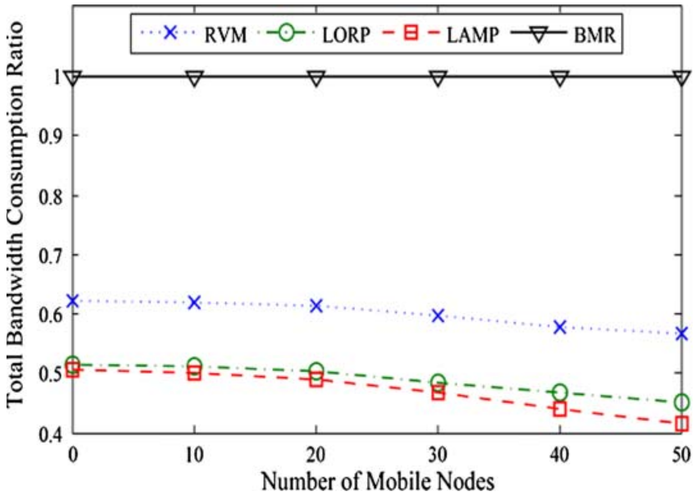

Fig. 20 The total bandwidth consumption ratio in different protocols for the different number of mobile nodes

when the route length becomes longer such that LORP creates more control packets than LAMP.

6.2.3 Total Bandwidth Consumption Ratio

In Fig. 20, we have compared the total bandwidth consumption ratio in different routing protocols for the different number of mobile nodes. The total bandwidth consumption contains the used bandwidth of control and data packets. It is observed that LAMP outperforms BMR, RVM and LORP. Since, more control packets has the higher bandwidth consumption and less hop counts has the lower bandwidth consumption, LAMP and LORP which create similar control packets as RVM and have shorter the route paths can efficiently reduce total bandwidth consumption. Moreover, LAMP is lower than LORP, since its route length is shorter than the one of LORP. Since, RVM has highest number of hop counts, its total bandwidth consumption cannot be efficiently reduced and is higher than LAMP and LORP. On the other hand, BMR has most total bandwidth consumption due to large number of control packets. Moreover, its total bandwidth consumption is raised when the number of mobile node is increased such that the ratios of LAMP, LORP and RVM are decreased.

From Fig. 21, it is observed that the total bandwidth consumption of the proposed protocol is less than that of the RVM, LORP and BMR for the different average mobility speed. This is because LAMP has quite shorter route length and less control overhead no matter the mobility speed is increased. Besides, the total bandwidth consumption ratio of LAMP and LORP are significantly improved when the average mobility speed is increased. This is because BMR has large number of control packets and LORP and LAMP have shorter route length.

6.2.4 Total Energy Consumption Ratio

Figure 22 measures the total energy consumption ratio in different protocols for different number of mobile nodes. The total energy consumption contains the energy consumption of

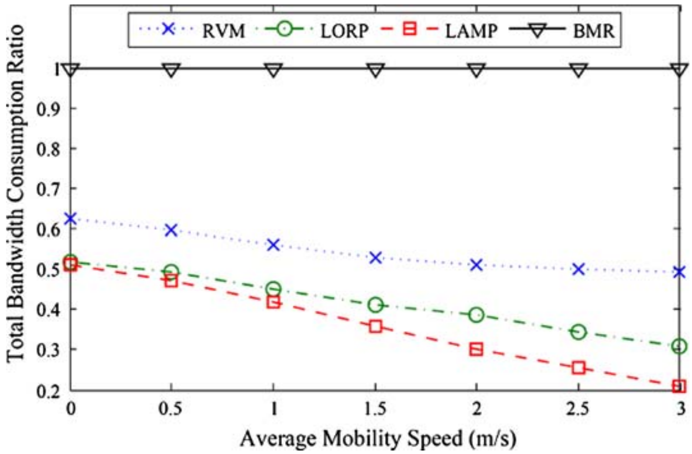

Fig. 21 The total bandwidth consumption ratio in different protocols for the different average mobility speed

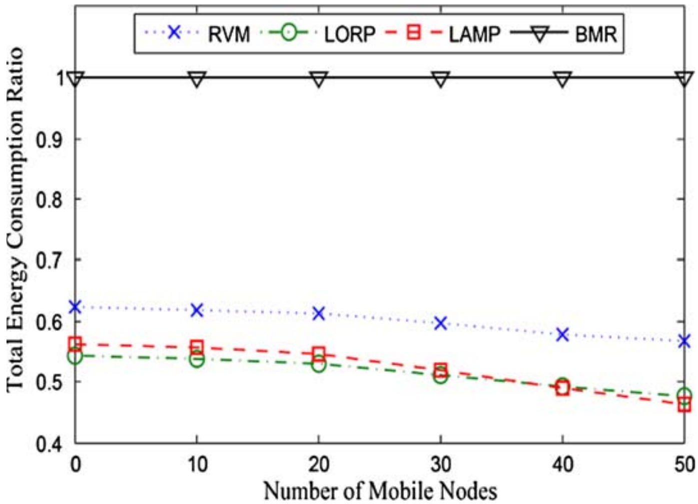

Fig. 22 The total energy consumption ratio in different protocols for the different number of mobile nodes

control and data packets. It is found that the total energy consumption of LORP is less than that of the RVM, LORP and LARP when the number of mobile nodes is less 40, since LORP has less number of control packets and hop counts and control packets which contain the FFN and FBN in LAMP consume more energy. However, the total energy consumption of LAMP is least when the number of mobile nodes is equal or larger than 40, since the number of control packets is similar between LAMP and LORP and LAMP has shorter routes than LORP. Although, RVM uses least number of control packets, it has higher number of hop counts and therefore its total energy consumption is slightly improved and is higher than LAMP and LORP. Besides, BMR has the highest total energy consumption since it requires large number of control packets. Moreover, the number of control packet in BMR heavily increases with the number of mobile nodes. Therefore, the total energy consumption ratios of LAMP, LORP and RVM reduce with the number of mobile nodes.

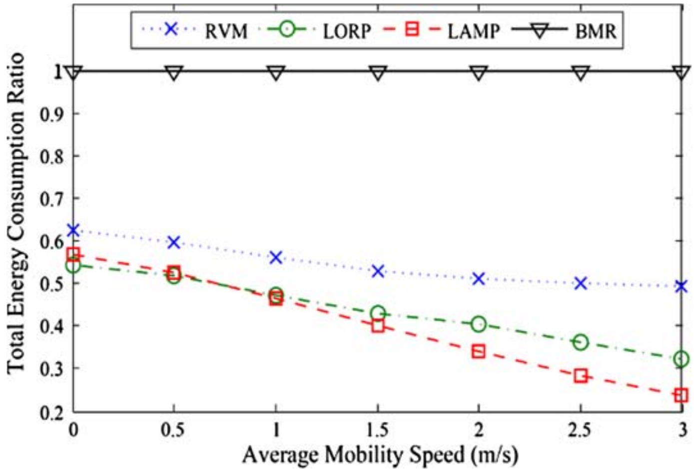

Fig. 23 The total energy consumption ratio in different protocols for the different average mobility speed

Figure 23 depicts the total energy consumption ratio in different protocols for the different average mobility speed. LORP has smallest ratio when the average mobile speed is less 1 m/s, since it consumes less energy for control packets than that of LAMP and has shorter routing path length. However, it is found that the total energy consumption of LAMP obtains smallest ratio if the average mobility speed is equal or larger than 1 m/s. This is because LAMP has quite shorter route length and less control overhead no matter the mobility speed is increased. On the other hand, since BMR has large number of control packets and LORP and LAMP have shorter route length, the total energy consumption ratio of LAMP and LORP significantly reduce with the average mobility speed.

7 Conclusions

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. We also analyze the network bottleneck problems during the construction of route and propose role switch operation to mitigate these problems. 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. http://www.ieee802.org/15/
3. Chang, C. Y., Yu, G. J., Lin, C.-F., & Wu, T.-T. (2002). Relay reduction and route construction for Scatternet over Bluetooth radio systems. In *Proc. IEEE 16th International Conference on Information Networking, II*, 2.1–2.10.
4. Perkins, C., & Bhagwat, P. (1994). Destination sequenced distance vector routing for mobile computers. In *Proc. ACM SIGCOMM*, pp. 234–244, Sep. 1994.

5. Perkins, C., & Royer, E. (1999). Ad-hoc on-demand distance vector routing. In *Proc. IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90–100, Feb. 1999.
6. Giordano, S., Stojmenovic, I., & Blazevic, L. (2003). Position based routing algorithms for ad hoc networks: A taxonomy. *Ad Hoc Wireless Networking*, Kluwer Academic Publishers, 2003.
7. Ali, S. (2004). The feasibility of mobile adhoc routing over Bluetooth. Thesis of MEng Degree in Computer Science, University College London.
8. Sun, M., Chang, C. K., & Lai, T. H. (2002). A self-routing topology for Bluetooth Scatternets. In *Proc. The International Symposium on Parallel Architectures*, Philippines, May, 2002.
9. Choi, C. S., & Choi, C. W. (2002). DSR based Bluetooth Scatternet. In *Proc. International Conference on Circuits/Systems Computers and Communications*, Philippines, May, 2002.
10. Mirza, A. B., & Pollard, J. K. Dynamic Source Routing in Bluetooth Personal Area Networks. http://www.ducati.doc.ntu.ac.uk/uksim/journal/Vol-5/No-1&2/MIRZA.pdf
11. Yang, C.-H., & Ruan, J.-W. (2005). On-demand routing for Bluetooth Scatternets subject to device mobility. In *Proceedings of the 19th International Conference on Advanced Information Networking and Applications*, 2005.
12. Jian, Z., Fei, Z., & Bin, W. (2005). Cluster based routing in Bluetooth ad-hoc network. http://www.ieeexplore.ieee.org/iel5/10362/32965/01544162.pdf, 2005
13. Werb, J., & Lanzl, C. (1998). A positioning system for finding things indoors. *IEEE Spectrum, 9*(35), 71–78.
14. Priyantha, N.B., Chakraborty, A., & Balakrishnan, H. (2000). The Cricket location-support system. In *Proc. Sixth Annual ACM Intl. Conf. on mobile Computing and Networking*, 2000.
15. Texas Instruments TIRIS. http://www.ti.com/tiris/default.htm
16. Gonzalez-Castano, F. J., Garcia-Reinoso, J. (2003). Survivable Bluetooth Location Networks. In *Proc. IEEE ICC, 2*, pp. 1014–1018.
17. Vershney, U., Vetter, R. J., & Kalakota, R. (2000). Mobile commerce: A new frontier. *Computer, 10*, pp. 32–38.
18. Darling, A. (2001). Waiting for the m-commerce explosion. *Telecommunication International, 3*, pp. 34–39.
19. Engebretson, J. (2006). Opportunities for growth. Wireless Asia.
20. Bridgelall, R. (2003). Enabling mobile commerce through pervasive communications with ubiquitous RF tags. In *Proc. IEEE Wireless Communications and Networking, 3*, pp. 2041–2046.
21. Sangani, K. (2004). RFID sees all. *IEEE Review, 50*(4), pp. 22–24.
22. Ni, L. M., Liu, Y., Lau, Y.-C., & Patil, A. P. (2003). LANDMARC: Indoor location sensing using active RFID. In *Proc. First IEEE International Conference on Pervasive Computing and Communications*, 2003.
23. Gonzalez-Castano, F. J., & Garcia-Reinoso, J. (2002). Bluetooth Location Networks. In *Proc. IEEE Globecom, 2002*.
24. Bluetooth Nokia. http://www.nokia.com/phones/6210/bluetooth.html
25. Bhagwat, P., & Segall, A. (1999). A routing vector method (RVM) for routing in Bluetooth Scatternets. In *Proc. IEEE Workshop MOMUC*, pp. 375–379, Nov. 1999.
26. Song, O., Lim, C., & Choi, C.-H. (2004). Mobility Management in Bluetooth ad hoc networks. In *Proc. of 14th Joint Conference on Communications & Information (JCCI)*, 2004.

Author Biographies

Sheng-Wen Chang received the B.S. degree in Computer Science and Information Engineering from Tamkang University, Taiwan, in 2004. Since 2005, he was working toward his Ph.D degree and currently he is a PhD candidate in Department of Computer Science and Information Engineering at Tamkang University. Mr. Chang is a student member of the IEEE Computer Society, Communication Society and IEICE society. He has won lots of scholarships in Taiwan and participated in many Bluetooth and Wireless Sensor Networking projects and published extensively in the wireless networking area. His research interests are WiMAX, wireless sensor networks, Bluetooth radio networks, wireless mesh networks and Ad Hoc wireless networks, concerning both theoretic results and algorithms design.

Prasan Kumar Sahoo received Master of Science in Mathematics from Utkal University, India. He received his M. Tech. degree in Computer Science from Indian Institute of Technology (IIT), Kharagpur, India and received his Ph.D in Mathematics from Utkal University, India in April, 2002. He joined in the Software Research Center, National Central University, Taiwan in 2001 and later joined as an Assistant Professor in the Department of Information Management, Vanung University, Taiwan in 2003. Currently he is working as an Associate Professor in the same department of Vanung University, since 2007. He was the Program Committee Member of *MSEAT'2004, MSEAT'2005*, *WASA'2006*, and *IEEE AHUC'2006* and 2008. His research interests include coverage problems, modeling and performance analysis of wireless sensor network and WiMAX.

Chih-Yung Chang received the Ph.D. degree in Computer Science and Information Engineering from National Central University, Taiwan, in 1995. He joined the faculty of the Department of Computer and Information Science at Aletheia University, Taiwan, as an Assistant Professor in 1997. He was the Chair of the Department of Computer and Information Science, Aletheia University, from August 2000 to July 2002. He is currently Professor of Department of Computer Science and Information Engineering at Tamkang University, Taiwan. Dr Chang served as an Associate Guest Editor of *Journal of Internet Technology* (JIT, 2004), *Journal of Mobile Multimedia* (JMM, 2005), and a member of Editorial Board of *Tamsui Oxford Journal of Mathematical Sciences* (2001–2005). He was an Area Chair of *IEEE AINA'2005*, Vice Chair of *IEEE WisCom* 2005 and *EUC* 2005, Track Chair (Learning Technology in Education Track) of *IEEE ITRE*'2005, Program Co-Chair of *MNSA*'2005, Workshop Co-Chair of *INA*'2005, *MSEAT' 2003*, *MSEAT'2004*, Publication Chair of *MSEA*T'2005, and the Program Committee Member of *USW'2005, WASN'2005*, and the 11th Mobile Computing Workshop. Dr. Chang is a member of the IEEE Computer Society, Communication Society and IEICE society. His current research interests include wireless sensor networks, mobile learning, Bluetooth radio systems, Ad Hoc wireless networks, and mobile computing.