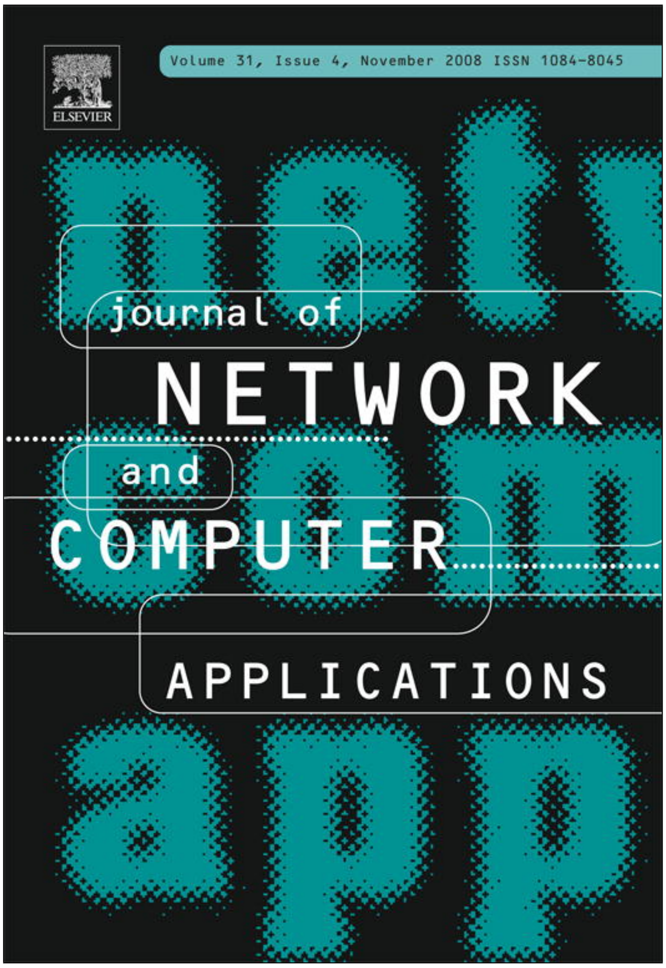
ELSEVIER
Volume 31, Issue 4, November 2008 ISSN 1084-8045
journal of
NETWORK
and
COMPUTER
APPLICATIONS

ELSEVIER

Available online at www.sciencedirect.com
ScienceDirect

Journal of Network and Computer Applications 31 (2008) 535–558

Journal of NETWORK and COMPUTER APPLICATIONS

www.elsevier.com/locate/jnca

Novel route maintenance protocols for the Bluetooth ad hoc network with mobility

Prasan Kumar Sahoo[a,*], Chih-Yung Chang[b], Sheng-Wen Chang[b]

[a]*Department of Information Management, Vanung University, Chungli, Taiwan 32045, ROC*
[b]*Department of Computer Science and Information Engineering, Tamkang University, Taipei, Taiwan, ROC*

Received 31 May 2007; received in revised form 30 November 2007; accepted 4 December 2007

Abstract

Bluetooth technology is specially designed for the wireless personal area networks to replace cable. Several challenges exist in Bluetooth scatternet formation and routing, since nodes can arrive and depart at arbitrary times. In this paper, novel route maintenance algorithms are proposed for the Bluetooth ad hoc networks, where nodes can enter or exit from the piconets time to time. Our protocols guarantee the connectivity among nodes and reconstruct the routes dynamically by considering location information of the nodes. Besides, it is proposed how to reduce the number of hops and to form the shortest route between the source and the destination due to addition of new nodes to a piconet. Performance analysis of our protocols show that they outperform in terms of end to end transmission delay, bandwidth consumption and average hop counts as compared to similar Bluetooth routing protocols that we have considered.

Keywords: Bluetooth; Scatternet; Route maintenance; Mobility management

1. Introduction

Bluetooth is a low-cost, low-power and short range communication technology, which operates in 2.4 GHz ISM band. It provides communication technology among battery-operated portable radio devices, such as personal digital assistant, headsets and notebooks.

*Corresponding author.
E-mail addresses: pksahoo@msa.vnu.edu.tw (P.K. Sahoo), cychang@mail.tku.edu.tw (C.-Y. Chang), swchang@wireless.cs.tku.edu.tw (S.-W. Chang).

1084-8045/$ - see front matter
doi:10.1016/j.jnca.2007.12.001

It supports both voice and data traffic among the devices typically located within range of 10 m and is designed to replace the inter-connection cables. As per Bluetooth specification (http://www.bluetooth.org), a 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/s and time-division duplex is employed for the sequential medium access. The piconets employ different frequency hopping code-division multiple-access techniques to prevent mutual interferences. Hence, multiple piconets can co-exist in a common area and can form a bigger ad hoc network known as scatternet. The relay node, which is referred as a bridge, can be a master in one piconet and slave in another or bridge between two or more piconets. Though, Bluetooth standard specifies about the piconet formation, scatternet formation and routing are still two major issues left open in the current specification. Since, Bluetooth scatternet is considered as a special type of ad hoc network, routing protocols for Bluetooth can be categorized as: table driven and on-demand routing protocols. In the table driven routing protocols (Perkins and Bhagwat, 1994), 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 on-demand routing protocols (Bhagwat and Segall, 1999; Perkins and Royer, 1999), a node first floods a query message to learn the route to the destination before it can send a message. Some drawbacks in on-demand routing protocols are due to the delay incurred by the query phase and flooding of the query signals. A blue-tree scatternet formation algorithm (Sun et al., 2002) is proposed to build a self-routing scatternet to minimize the routing overhead. But, it does not mention how to construct the scatternet, if nodes are not within proximity of each other. Moreover, the number of hops between the source and the destination nodes of this blue-tree based scatternet is high, which incurs more delay time to dispatch the packets. The link formation time of current Bluetooth specification is too long for the mobile devices. Kapoor and Gerla (2003) define a routing scheme for Bluetooth scatternets, which is based on the Zone Routing Protocol and explain how the scheme takes into account the specifics of the Bluetooth MAC layer and also provide simulation results showing the performance of the scheme. The relay reduction and route construction protocol (LORP) (Yu et al., 2007) proposes how to retain the suitable relays and remove other nodes to reduce the routing path length. Mirza and Pollard (2002) present 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 the simulation of this routing protocol for Bluetooth PAN is an on-demand network layer protocol that may be suitable for power-limited, multi-hop, ad hoc mobile devices.

The authors propose a so-called Blueline algorithm (Chang and Chou, 2005) to reduce the time and path length in routing, in which two Bluetooth nodes should communicate directly, if they are within each other's transmission range. In their work, they indicate that the communicating path between two Bluetooth nodes is shorter. In all of the above routing protocols for the Bluetooth scatternet, no work considers the location information of the nodes to shorten the routing path, since it is expensive to provide location information through GPS. However, many users-positioning solutions have been proposed recently, though they are based on the specialized devices not supported by commercially available data terminals (Werb and Lanzl, 1998; Priyantha et al., 2000; Harter et al., 1999;

http://www.ti.com/tiris/default.htm). Such location-aware protocols (Gonzalez-Castano and Garcia-Reinoso, 2003) 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 (Vershney et al., 2000; Darling, 2001), 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 receive store coupons and advertisements. As per the recently developed MagicBeamer (http://www.blueblitz.com), any information or advertisement can be transferred to a mobile phone or PDA with help of Bluetooth technology. It is to be noted that now-a-days the mobile phones and PDA are equipped (Engebretson, 2006; ETRI, 2005) with Radio Frequency Identification (RFID) (Bridgelall, 2003; Sangani, 2004; Shepard, 2004; Stanford, 2003), which is highly useful to m-commerce scenarios. The beta version of the handheld Bluetooth RFID reader, dubbed as IDBlue (http://www.physorg.com) have been developed and launched in the market that enables the coexistence (Shepard) of RFID and Bluetooth in many handheld mobile devices.

Considering the recent technological developments for the m-commerce environments, we assume that location information can be transferred to the Bluetooth enabled handheld devices by several means. For example, LANDMARC (Ni et al., 2003), a location sensing prototype system that uses RFID technology for locating objects inside buildings and it improves the overall accuracy of locating objects by utilizing the concept of reference tags. Besides, the Bluetooth Location Networks (BLN) (Gonzalez-Castano and Garcia-Reinoso, 2002) transmits location information to the service servers without user participation and its base technology is supported by existing commercial handhelds (http://www.nokia.com). Korhonen et al. (2006) have introduced mTag, a distributed event driven architecture for discovering location specific mobile web services. In their work, service discovery is initiated by touching a fixed RFID reader with a mobile passive RFID tag attached to a mobile device such as PDA or phone. It is found that (http://www.patentstorm.us/patents/6717516.html,), devices having RFID tags can be located using dual function fixed devices which are distributed throughout a facility. The devices will identify those units with which they are communicating using a wireless radio data communications protocol and also identify items within the local area using RFID tags on the units. The multimedia information system (Martin and Trummer, 2005) allows the presentation of digital content adjusted to the individual visitor's interests. It supports different display devices from PDA up to projection screens and allows the integration of different localization techniques for location based information presentation.

Other than these routing schemes, the location-aware mobile network is an important research issue in Bluetooth technology. The authors have proposed a route reduction protocol (Chang et al., 2007), which requires the location information of the nodes, and can reduce the number of hops as compared to the works in Yu et al. (2007) and Bhagwat and Segall (1999). But, the reduction in number of hops has not been considered based on the mobility of the nodes. A robust scatternet topology in terms of node mobility for the Bluetooth ad hoc networks is proposed in Song et al. (2004). The authors propose an on-demand routing protocol (Yang and Ruan, 2005) for the Bluetooth scatternets, which can detect the mobility of the devices and establishes routes in a mobile scatternet to cope with both power consumption and device mobility issues. However, these protocols do not consider location information of the nodes and the number of hop counts is not optimum. 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 node is added or left out of the piconet. Hence, we propose the ROute Maintenance Algorithms (ROMA) that consider mobility and location of the nodes and reconstruct the routing path time to time and try to reduce the routing path length.

The rest of the paper is organized as follows. Section 2 discusses some important routing protocols related to our protocol. Our proposed ROMA are described in Section 3. Simulation results and performance analysis of our protocols are discussed in Section 4 and concluding remarks are made in Section 5 of the paper.

2. Overview of routing protocols

This section presents few routing protocols and compares their algorithms in terms of routing path reduction. As shown in Fig. 1, nodes S, S_{41} and D are pure slaves, M_1 and M_4 are pure masters for the piconet P_1 and P_4, respectively. Node B_{12} is a master for the piconet P_2 and a bridge between piconets P_1 and P_2. Node B_{23} is a master for the piconet P_3, as well as a bridge between P_2 and P_3. Node B_{45} is a master for the piconet P_5 as well as a bridge between P_4 and P_5. Node B_{34} is the bridge between piconets P_3 and P_4. In this scatternet, let node S has to route data to the destination device D. We use Fig. 1 to explain the required number of hops between the source and the destination based on the routing algorithms of routing vector method (RVM) (Bhagwat and Segall, 1999), LORP (Yu et al., 2007) and location-aware routing protocol (LARP) (Chang et al., 2007), as described below.

2.1. Routing vector method (RVM)

The RVM (Bhagwat and Segall, 1999) proposes an efficient routing scheme for the Bluetooth scatternet to discover new routes. According to RVM, the source broadcasts the

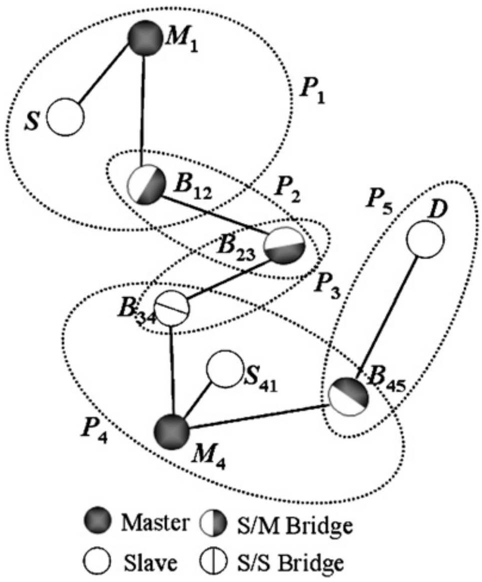

Fig. 1. Example of a scatternet initially formed by several Bluetooth nodes.

SEARCH packets, which accumulate the list of nodes along the routing path from the source to the destination. Ultimately, several broadcast packets may reach at the destination node, who considers the list of nodes of the first *SEARCH* packet as the routing path and unicasts a *REPLY* packet to the source through the same path, as decided during the *SEARCH* process. An example of the routing path formed by the RVM is shown in Fig. 2.

Suppose, a packet is sent from source S of piconet P_1 to the destination node D of piconet P_5. According to RVM, the final routing path could be $S \longrightarrow M_1 \longrightarrow B_{12} \longrightarrow B_{23} \longrightarrow B_{34} \longrightarrow M_4 \longrightarrow B_{45} \longrightarrow D$, which requires seven hops to route the packet from the source to the destination.

2.2. Relay reduction routing protocol (LORP)

An efficient relay reduction and disjoint route construction protocol (LORP) (Yu et al., 2007) for the Bluetooth scatternet is proposed to improve the drawbacks in RVM. As per LORP, the network topology could be adjusted dynamically by reducing the unnecessary relay nodes. According to this protocol, reduction of routing path and hop counts could be achieved by considering the physical distance among the nodes. Thus, two disjoint routes for any pair of source and destination, located in different piconets could be created. As shown in Fig. 3, as per LORP, since nodes S and B_{12} are within communication range (normally 10 m for the Bluetooth) of each other, packets can be routed through S to B_{12}. Similarly, the packet can be routed through B_{12} to B_{34} and also from B_{34} to B_{45}, as they are within the communication range. Thus, the final routing path from the source to the destination could be $S \longrightarrow B_{12} \longrightarrow B_{34} \longrightarrow B_{45} \longrightarrow D$, which requires four hops instead of seven hops, as in RVM. In LORP, if any node is within communication range of thesource, it pages the farthest bridge of the scatternet and constructs the piconet repeatedly. However, LORP does not consider the piconet reconstruction, if any node is added or left from the original scatternet.

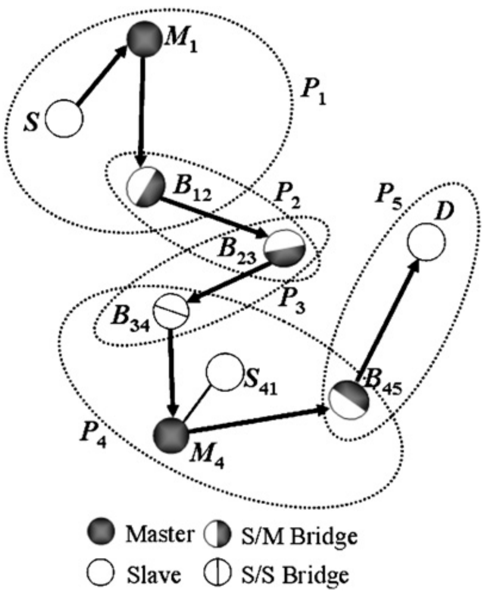

Fig. 2. Routing path formed by the RVM algorithm to route data from S to D.

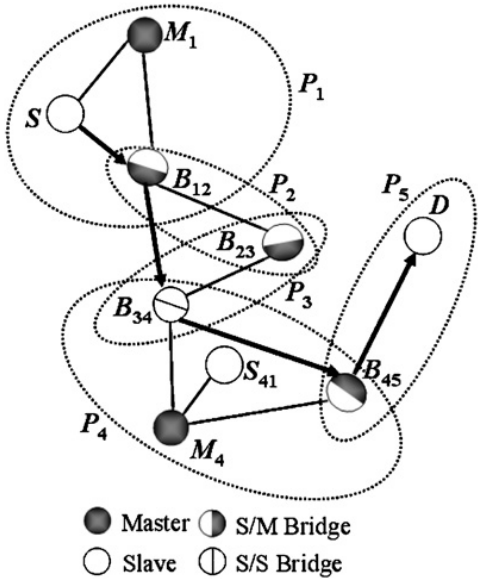

Fig. 3. Routing path formed by the LORP algorithm to route data from S to D.

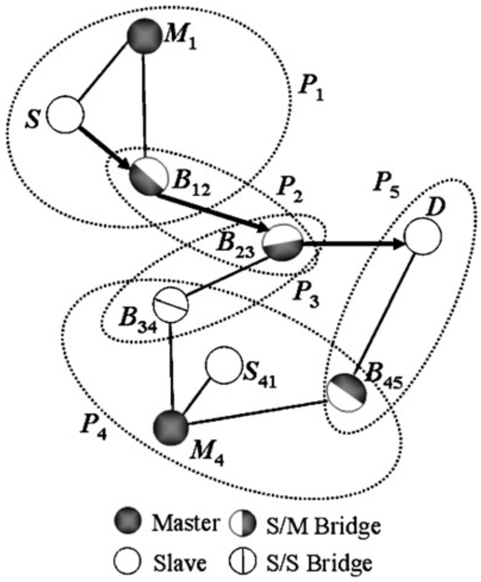

Fig. 4. Routing path formed by LARP algorithm to route data from S to D.

2.3. Location-aware routing protocol (LARP)

A novel LARP (Chang et al., 2007) for the Bluetooth scatternet is proposed, which reduces the hop counts between the source and the destination and reconstructs the routes dynamically using location information of the Bluetooth devices. It is observed that LARP still reduces the number of hops as compared to RVM and LORP and can still construct a shortest routing path, as shown in Fig. 4. As per LARP, before routing a packet, source node goes for the *ROUTE SEARCH* phase and collects the location information of the nodes in the *SEARCH* phase. However, during the *ROUTE REPLY* phase, it selects fewer nodes from the list of *ROUTE SEARCH* phase, based on the shortest distance between the source and the destination. For example, as shown in Fig. 4, since node B_{23} can be

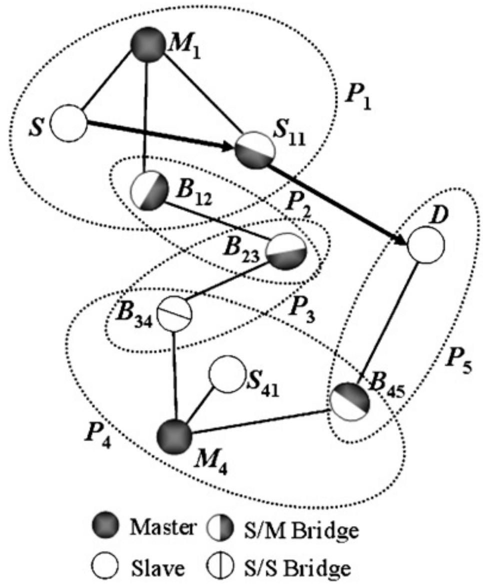

Fig. 5. Routing path formed by ROMA to route data from S to D.

connected to the destination node D based on its location information, the routing path can still be reduced. Thus, the final routing path as per LARP could be $S \longrightarrow B_{12} \longrightarrow B_{23} \longrightarrow D$, which requires only three hops instead of four as in LORP and seven as in RVM.

Though, the routing path in LARP is shorter than LORP and RVM, LARP does not consider mobility of the nodes and the corresponding change in the routing path. It does not say how to maintain the routing path, if nodes are added or left from a piconet. However, it may be possible that the existing routing path will be disturbed and can be changed due to mobility of a node, thereby increasing the routing overheads. Hence, we propose the ROMA that consider the location information and mobility of the nodes and reduce the routing path, if nodes are added to an existing scatternet, as shown in Fig. 1. As mentioned in the related work, the LARP reduces the routing path as compared to RVM and LORP and fails, if any new node enters into the piconet. However, our algorithms can accommodate new nodes to the original scatternet and can still reduce the routing path to $S \longrightarrow S_{11} \longrightarrow D$, as shown in Fig. 5, where S_{11} is the newly added node. The complete algorithms of our route maintenance protocols are given as follows.

3. The ROute Maintenance Algorithms (ROMA)

In our protocols, it is assumed that each device of the scatternet knows its location information through RFID (Ni et al., 2003) and BLN (Gonzalez-Castano and Garcia-Reinoso, 2002) and each node has a unique ID different from its *BD_ADDR*. The source node of one piconet intends to communicate with destination node of another, whose ID is known, but location is unknown. Besides, it is assumed that each master knows the ID, *clock_offset* and location information of its slaves during the scatternet formation phase. The master can get this information about its slaves during connection phase of the piconet. The intermediate nodes can get location information of the source and the destination, thanks to the control packet routed from the source to the destination during the route search phase. Before proceeding to our protocols and algorithms, we introduce here some definitions, which are used in our protocols.

3.1. Definitions

Location (*LOC* (*A*)): Location of any Bluetooth device A is its location in the scatternet, which is expressed in Cartesian co-ordinate system $A(x, y)$. As per our assumptions, BLN (Gonzalez-Castano and Garcia-Reinoso, 2002) transmits location information to the service servers without user's participation.

Routing master (M_i): Any master M of ith piconet is known as a routing master M_i, if any of its slaves or master itself is a member of the initial shortest path between the source and the destination. It is to be noted that initially a shortest path is formed between the source and destination and the route is reconstructed, if any node is added or left out of the piconet. Each routing master stores the route information, including *BD_ADDR*, *clock_offset*, and location information (*LOC*) of the members, who participate in the routing.

As shown in Fig. 4, nodes M_1, B_{12}, B_{23}, and B_{45} are the routing masters of the whole scatternet, since their slaves or themselves are members of the initial shortest routing path between the source and the destination.

Routing piconet (P_i): Any piconet that contains a routing master M_i is known as a routing piconet P_i. As shown in Fig. 4, piconets P_1, P_2, P_3 and P_5 are routing piconets. If any slave joins in any of the routing piconet, *BD_ADDR*, *clock_offset* and *LOC* of that new node is forwarded to the routing master of the corresponding routing piconet.

Signal to noise ratio (*SNR*) *threshold*: The ratio of the received signal to the noise is called the SNR, which should be $\geqslant \rho$, where ρ is a user defined threshold and is fixed for all nodes of the scatternet. According to definition

$$SNR = \frac{received\_power}{interference\_power} \geqslant \rho$$

and ρ is known as the SNR threshold.

Weak node: The node whose SNR value is less than the SNR threshold (ρ) is termed as a weak node. It is to be noted that a weak node must be a receiver. An example of the weak node is given in Fig. 6(b).

Weak link: A link connecting to any node with a weak node is termed as a weak link. It is to be noted that a weak link may be connected to two weak nodes or one weak node with another normal node. In our protocol, since, the sender is not aware of the status of its receiver, a weak node notifies its sender, if it becomes weak one and ultimately the link between the sender and receiver becomes a weak link. An example of the weak link is given in Fig. 6(b).

As shown in Fig. 6(a), let there exists an initial link between nodes M_i and S_i. If node S_i moves towards right, as shown in Fig. 6(b), the distance between nodes M_i and S_i is increased. Since, signal strength received by the receiver S_i may be reduced due to its

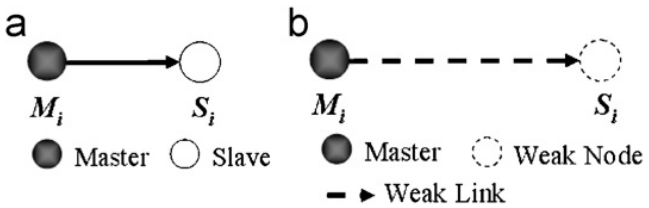

Fig. 6. Illustration of weak node and weak link. (a) Example of a normal link. (b) Example of a weak link and weak node due to mobility of any node S_i.

mobility, SNR value of node S_i may be $<\rho$. In this case, node S_i becomes a weak node and the respective link between M_i and S_i becomes a weak link, which is notified by S_i to its sender M_i.

Member Collection Procedure (*MCP*): In this procedure, a weak node S_i requests its routing masters M_i and M_{i+1} of *i*th and $i+1$th piconets, respectively to return the *BD_ADDR*, *clock_offset* and *LOC* of all devices (including the master) of the routing piconets P_i and P_{i+1}. When a weak node goes for the *MCP*, it forwards a member collection packet to the routing masters to get these information of their slaves.

3.2. Node Add Procedure

In an ad hoc Bluetooth network, it is possible that new nodes may be added to the original scatternet. For example, if a new slave S_{new} joins to any routing piconet P_i, the respective routing master M_i verifies, if S_{new} can reduce the routing path length or not. In order to verify this, the routing master M_i carries out the *Node Add Procedure*, as given in Table 1.

The above algorithm can be explained with an example, as shown in Fig. 7. Let, a new slave S_{11} joins to the routing master M_1 of piconet P_1, as shown in Fig. 7(a). Then, the routing master M_1 initiates the *Node Add Procedure*. Since, M_1 knows *BD_ADDR*, *clock_offset* and *LOC* of nodes S, B_{12}, B_{23} and D, it calculates if S_{11} can connect to S such that the routing path is reduced to S, S_{11}, and D. The routing master M_1 executes the procedure connecting (S, S_{11}) and connecting (S_{11}, D). The routing master M_1 notifies nodes S_{11} and D to enter to page scan state and node S to enter to page state. Then, node S establishes a link with node S_{11} and node S_{11} enters page state. After that, node S_{11} constructs a link with node D and combines the piconets to reduce number of hops and piconets. Thus, nodes S, S_{11} and D construct a piconet, as shown in Fig. 7(b), where node S_{11} plays the role of a master and number of hops is reduced to 2.

Table 1

Algorithm 1: Node Add Procedure
Notations:
1. S_{new}: Newly added node to the piconet;
2. M_i: Routing master of *i*th piconet;
3. P_i: Routing piconet;
4. S_i or S_j: Slave i or j;

Node Add Procedure (S_{new}, Role switch operation)
1. **Step 1:** If: a node S_{new} newly connects to any P_i
2. {
3. M_i, which is connected to S_i calculates:
4. If: S_{new} can reduce routing path, with any one of the slaves $S_1, S_2, \ldots, S_{i-1}, S_i, S_{i+1}, \ldots, S_n$,
5. **Step 2:** If: S_{new} can connect to S_k, for any $k \geqslant i+3$ or can connect to S_j, for any $j \leqslant i-2$
6. Routing path can be reduced;
7. **Step 3:** If: S_{new} can reduce routing path,
8. M_i executes *Connecting Procedure* $(S_j, S_{new}) \| (S_{new}, S_k)$;
9. **Step 4:** S_{new} combines the piconets;
10. }

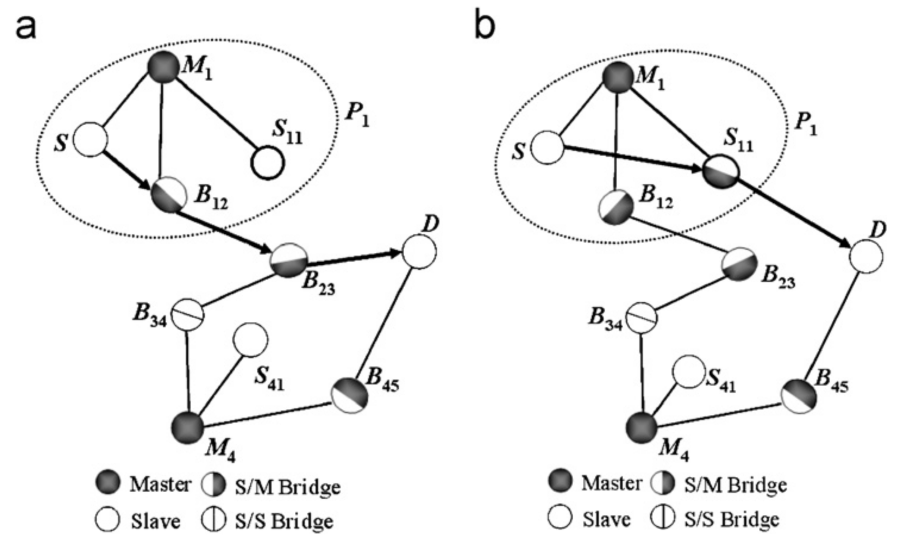

Fig. 7. Illustration of route maintenance due to addition of a new node. (a) Addition of new node S_{11} to the routing master M_1. (b) The new routing path after addition of node S_{11}.

3.3. Node leaving procedure

It is to be noted that a weak link is formed in a scatternet, if a node moves away from its initial position, which is termed as the node leaving procedure. The node leaving procedure comprises three different procedures to maintain the routing path of the scatternet. The first one is the *Node Replacement* Procedure, which finds another node to replace the weak node. The second one is the *Link Replacement* Procedure, which finds another link to replace the weak link and finally the *Subroute Construction Procedure*, which reconstructs the routing path, if a node leaves from the piconet. In our protocol, we suggest that *Node Replacement* Procedure has higher priority over *Link Replacement* Procedure, since *Node Replacement* Procedure maintains a shorter route. Similarly, *Link Replacement* Procedure has higher priority over *Subroute Construction Procedure*, since *Link Replacement* Procedure costs less control overhead than the *Subroute Construction Procedure*.

3.3.1. Node Replacement Procedure

In this procedure, a routing master or weak node intends to select one of the devices to replace the weak link. In the selection process, the first priority is given to the slave over the master and a master is given priority over a bridge. Normally, a slave node is preferred to be selected by the routing master or by the weak node, since it does not raise any additional cost in guard time and its traffic overhead is lower than the master. Details of our node replacement algorithm are described in Table 2. After selecting any node from the piconets based on the algorithm given in Table 2, either the routing master or the weak node goes for the Connecting Procedure, as given in Table 3.

An example is given in Fig. 8 to explain the node replacement algorithm. As shown in Fig. 8(a), let, initially there exists a route in the original scatternet. If node S_{11} moves towards right, a weak link (S, S_{11}) is formed, as shown in Fig. 8(b). Since, node S_{11} becomes a weak node, it informs its routing master M_1 to initiate the node leaving procedure (S_{11}, M_1), after waiting for a random back-off time. If any weak node S_i has only one weak link, routing master M_1 executes the Node Replacement (S_{11}, M_1)

Table 2

Algorithm 2: Node Replacement Procedure

Notations:

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

Node Replacement (S_i, M_i)
{
  If (S_i is connected to one M_i and has one weak link)
  {
    If (M_i selects one of S located in P_i
    such that S can connect to S_{i-1} and S_{i+1})
    {
      M_i executes *Connecting Procedure* (S_{i-1}, S) and (S, S_{i+1});
    }
    Else
      GO to *LINK REPLACEMENT* procedure;
  }
  Else
  {
    If (S_i selects one of devices S located in $P_i \| P_{i+1}$
    such that S can connect to S_{i-1} and S_{i+1})
    {
      S_i executes *Connecting Procedure* (S_{i-1}, S) and (S, S_{i+1});
    }
    GO to LINK REPLACEMENT procedure;
  }
}

Table 3

Algorithm 3: Connecting Procedure

Notations:

1. M_i: *Routing master*;
2. S_i: *Weak node*;
3. d_i: *Pure slave i*;

Connecting (d_1, d_2) **Procedure**
{
  Routing master M_i or weak node S_i notifies devices d_1 and d_2;
  Device d_1 goes to page state;
  Device d_2 goes to page scan state;
  Device d_1 constructs a link with d_2;
}

Procedure. As shown in Fig. 8(b), since, node S_{12} can connect to nodes S and D, routing master M_1 selects S_{12} to replace S_{11} along the route. Finally, the routing master M_1 executes the *Connecting Procedure* to connect (S, S_{12}) and (S_{12}, D). Then, node S_{12} combines the piconets to reduce number of piconets and becomes master of nodes S and D. As shown in the figure, it is observed that the length of the new route is same as the old one.

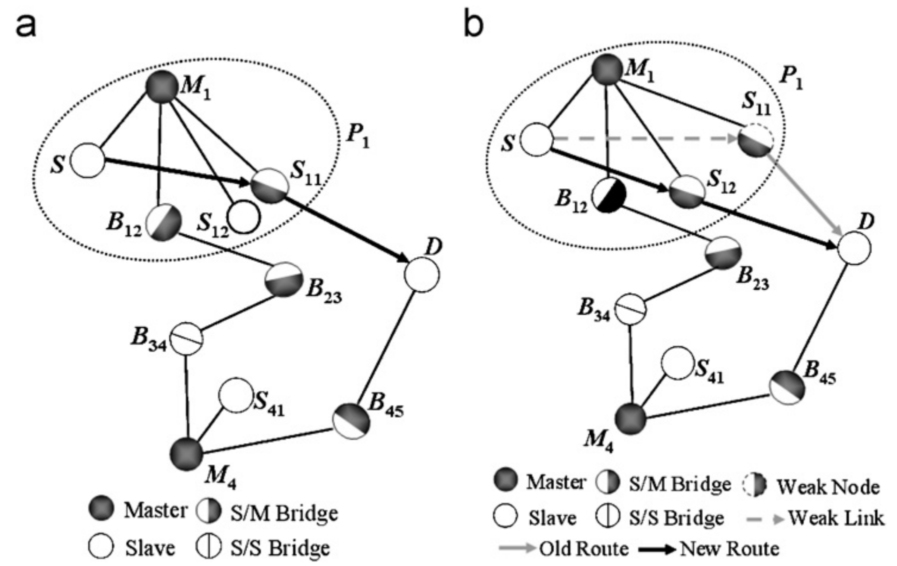

Fig. 8. Example of node replacement due to mobility of a node. (a) Original scatternet without any node replacement. (b) Replacement of node due to mobility of S_{11}.

3.3.2. *Link Replacement Procedure*

The Link Replacement Procedure is executed, if weak link is formed due to mobility of a node. The link replacement can be categorized into three cases, as described below.

Case 1: If a weak node has only one weak link and is connected to one routing master:

In this case a weak node does not execute the MCP. The complete algorithm of this procedure is given in Table 4 and an example as shown in Fig. 9 is given to explain the algorithm. Let, there exists a routing path $S \longrightarrow S_{11} \longrightarrow D$ in the scatternet, as shown in Fig. 9(a). If node S_{11} moves to right, it creates a weak link (S, S_{11}), as shown in Fig. 9(b). Since, node S_{11} becomes the weak node, it informs M_1 to initiate the node leaving procedure (S_{11}, M_1), after waiting for a random backoff time. If any weak node S_i has only one weak link, routing master M_1 executes the Node Replacement (S_{11}, M_1) Procedure. However, Node Replacement Procedure is failed, since, no device can be connected to S and D. In this case, routing master M_1 executes the Link Replacement (S_{11}, M_1) Procedure and finds a slave S_{13}, which can connect to nodes S and S_{11}. Then, routing master M_1 executes the *Connecting Procedure* (S, S_{13}) and (S_{13}, D). Finally, node S_{13} combines the piconets to reduce number of piconets and becomes the master between nodes S and S_{11}. In this case, length of the new routing path is longer and more than one hop as compared to the old one.

Case 2: If a weak node has two weak links and is connected to one routing master:

In this case a weak node executes the MCP. An example of this link replacement is shown in Fig. 10. As shown in Fig. 10(a), initially there exists a routing path $S \longrightarrow S_{11} \longrightarrow D$ in the scatternet. If node S_{11} moves to right in upward direction, it creates weak links (S, S_{11}) and (S_{11}, D), as shown in Fig. 10(b). By measuring the *SNR* value, both nodes S_{11} and D become the weak node and notify to their senders S and S_{11}, respectively. Since, node S_{11} has two weak links, it initiates the node leaving procedure (S_{11}, M_1) after waiting for a random backoff time. The weak node S_{11} executes the Member Collection (M_1, B_{45}) Procedure, since it has two weak links. Weak node S_{11} sends a member collection packet to the routing master M_1 and routing master B_{45} through node D. Let, S_1 and S_2 be the set of devices located in the routing piconet P_1 and P_5, respectively. $S_1 = \{M_1, S, B_{12}, S_{11}, S_{14}\}$

Table 4

Algorithm 4: Link Replacement Procedure
Notation:
1. P_i: *Routing piconet*;
2. M_i: *Routing master*;
3. S_i: *Weak node* (*may be pure slave or bridge*);
4. S: *Any device*;
1. **Link Replacement** (S_i, M_i)
2. $\{S_{left}$: $\{s|s$ is able to connect with both S_{i-1} and $S_i\}$;
3. S_{right}: $\{s|s$ is able to connect with both S_i and $S_{i+1}\}$;
4. If (S_i has one weak link and connects to one M_i)
CASE 1
5. {If($M_i$selects one of devices S located in P_i such that S can connect to S_{i-1} and S_i)
6. {M_i executes *Connecting Procedure* (S_{i-1}, S_i) and (S, S_i);}
7. Else
8. GO to *Subroute Construction Procedure*;}
9. Else
10. {If(S_i has one weak link and is connected to two M_i)
CASE 3
11. {If(S_i selects one of the devices S located in $P_i \| P_{i+1}$ such that S can connect to S_{i-1} and S_{i+1})
12. {S_i executes *Connecting Procedure* (S_{i-1}, S) and (S, S_{i+1});}
13. Else
14. GO to *Subroute Construction Procedure*;}
15.Else
16. {If(S_i has two weak links and connects to one or two M_i)
CASE 2 and 3:
17. {If($\exists S_1 \in S_{left}$ and $S_2 \in S_{right}$ such that S_1 can connect to S_2)
18. {S_i executes *Connecting Procedure* (S_{i-1}, S_1) and (S_1, S_2);
and connecting (S_2, S_{i+1})}
19. Else
20. GO to *Subroute Construction Procedure*;}}}}

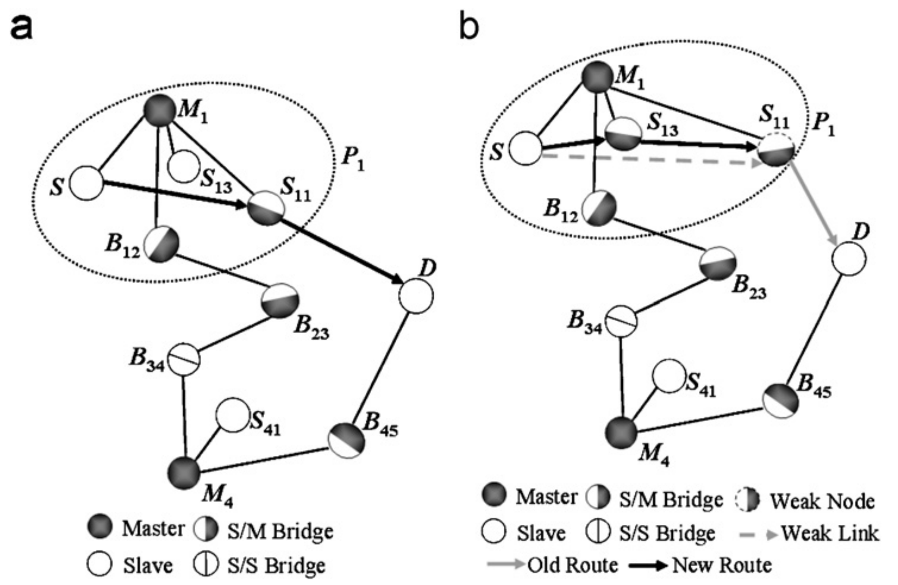

Fig. 9. Example of link replacement due to mobility of a node. (a) Original scatternet without any link replacement. (b) One weak link is formed due to mobility of S_{11}, which is replaced by the bold lines.

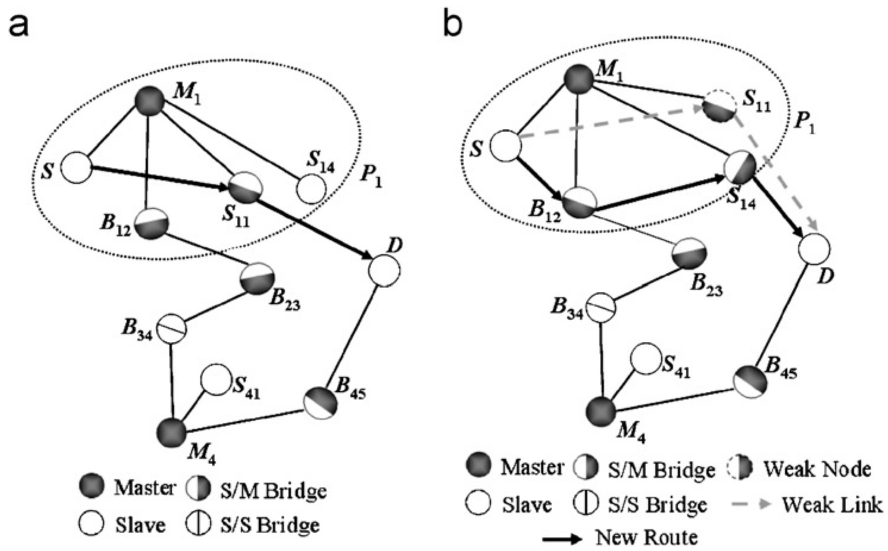

Fig. 10. Formation of two weak links and their replacement. (a) Original scatternet without any link problem. (b) Two weak links are formed due to mobility of S_{11}, which are replaced later.

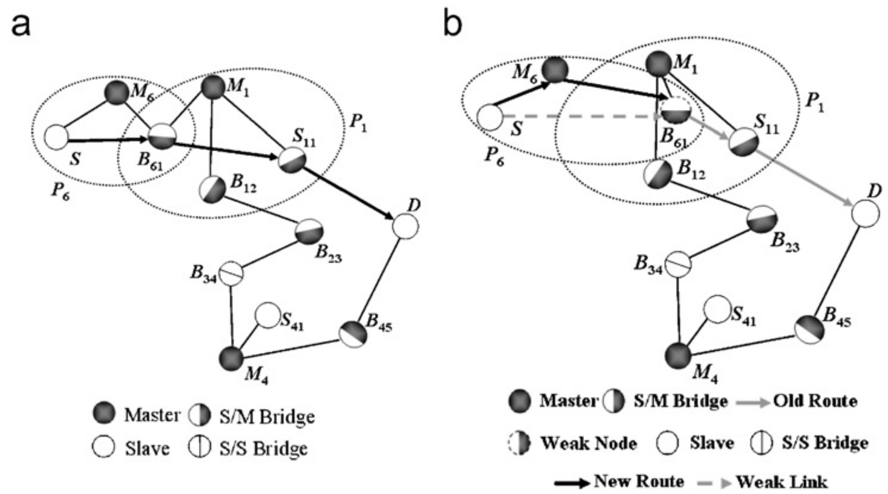

Fig. 11. Formation of weak link, where a weak node is connected to two routing masters. (a) Original scatternet without any link problem. (b) Weak link is formed due to mobility of B_{61}.

and $S_2 = \{B_{45}, D\}$ and hence $S_{left} = \{M_1, B_{12}\}$ and $S_{right} = \{S_{14}\}$. Weak node S_{11} checks, if there exists any device in $S_1 \cup S_2$ such that node replacement or link replacement can be applied. Weak node S_{11} executes the Node Replacement (S_{11}, M_1) Procedure, but it is failed, since no device can connect to nodes S and D. Then, weak node S_{11} executes the Link Replacement (S_{11}, M_1) Procedure and finds B_{12} along S_{left} and S_{14} in S_{right} such that B_{12} can connect to S_{14}. Weak node S_{11} executes the *Connecting Procedure* (S, B_{12}), (B_{12}, S_{14}) and (S_{14}, D) and node B_{12} combines the piconets to reduce number of piconets. Ultimately, B_{12} becomes the master of nodes S and S_{14}. In this case, length of the new route is more by one hop than the old one.

Case 3: If a weak node has one or two weak links, but is connected to two routing masters:

In this case a weak node executes MCP. The link replacement of this case is shown in Fig. 11. As shown in Fig. 11(a), let there exists a routing path $S \longrightarrow B_{61} \longrightarrow S_{11} \longrightarrow D$ in the scatternet. If node B_{61} moves towards right, it generates a weak link (S, B_{61}), as shown in Fig. 11(b). By measuring the *SNR* value, node B_{61} becomes a weak node and notifies to its sender S. In this case, weak node B_{61} connects to two routing masters and therefore executes the MCP (M_6, M_1). Weak node B_{61} sends a member collection packet to the routing masters M_6 and M_1. Let, S_1 and S_2 be the set of devices located in the routing piconets P_1 and P_6, respectively. $S_1 = \{M_6, S, B_{61}\}$ and $S_2 = \{M_1, B_{61}, B_{12}, S_{11}\}$. Weak node B_{61} checks, if there exists any device in $S_1 \cup S_2$ such that node replacement or link replacement can be applied. Weak node B_{61} executes the Node Replacement (B_{61}, M_6) Procedure, but it is failed, since no device can connect to nodes S and S_{11}. Then, weak node B_{61} executes the Link Replacement (B_{61}, M_6) Procedure and finds M_6 can connect to S and B_{61}. Weak node B_{61} executes the *Connecting Procedure* (S, M_6) and (M_6, B_{61}) and node M_6 combines the piconets to reduce the number of piconets. In this case, length of the new route is more by one hop than the old one.

A node, after becoming weak one, notifies its status to its connecting nodes, waits for a random backoff time and then executes the node leaving procedure or notifies its routing master to execute the same. If a relay (bridge) node has two weak links along its forwarding route, the relay node and its receiver, both become weak nodes. In this case, both of them individually execute the node leaving procedure after waiting for a random backoff time. This creates two sub-routing paths or length of the routing path is increased further. In order to avoid this situation, the weak node having two weak links waits less backoff time and first executes the node leaving procedure by itself. In order to repair the two weak links efficiently, the weak node requires information about other nodes located in its routing piconet and the routing piconet connected by its receiver. The weak node sends a member collection packet to its routing master and to the routing master connected by its receiver, before going for the MCP. Upon receiving a member collection packet, its receiver node stops repairing the routing path. The details of the link replacement algorithm are described in Table 4. Finally, algorithm of node leaving procedure can be summarized, as given in Table 5. It is to be noted that the node and link replacement algorithms for all possible cases are combined together and are presented in Table 5.

3.3.3. Subroute Construction Procedure

This procedure is executed by the weak nodes involved in the existing routing path to construct a sub-routing path. Algorithm of this procedure is given in Table 6 and examples are shown in Fig. 12(a) and (b) to explain the procedure. As shown in Fig. 12(a), let there exists a routing path $S \longrightarrow B_{61} \longrightarrow S_{11} \longrightarrow D$ in the scatternet. If node S_{11} moves to right in upward direction, it creates the weak links (B_{61}, S_{11}) and (S_{11}, D), as shown in Fig. 12(b). Since, S_{11} is a receiver with respect to node B_{61} and D is a receiver with respect to S_{11}, both S_{11} and D become the weak nodes and notify to their senders B_{61} and S_{11}, respectively. Since, node S_{11} has two weak links, it initiates the node leaving procedure (S_{11}, M_1) after waiting for a random backoff time. Moreover, weak node S_{11} has two weak links. Hence, it executes the Member Collection (M_1, B_{45}) Procedure. Let, S_1 and S_2 be the set of devices present in the routing piconets P_1 and P_5, respectively, and S_{left}, S_{right} represent set of nodes located along left and right sides of node S_{11} in its routing piconet, respectively. Then, $S_1 = \{M_1, B_{61}, B_{12}, S_{11}\}$, $S_2 = \{B_{45}, D\}$ and $S_{left} = \{M_1, B_{12}\}$ and $S_{right} = \{\Phi\}$. Upon collecting these information, weak node S_{11} executes the *Node Replacement* (S_{11}, M_1) *Procedure* and finds that no device can connect to B_{61} and D. After this, weak node S_{11}

Table 5

Algorithm 5: Node Leaving Procedure
Notations:
1. P_i: *Routing piconet*;
2. M_i: *Routing master*;
3. S_i: *Weak node* (*may be pure slave or bridge*);
4. *S*: *Any device*;
1. **Node Leaving** (S_i, M_i)
2. **CASE 1:** If (S_i connects to one M_i and has one weak link)
//Node Replacement
3. Step 1: M_i executes *Node Replacement* (S_i, M_i) *Procedure*;
4. If (Node Replacement fails)
5. M_i proceeds to Step 2;
//Link Replacement
6. Step 2: M_i executes *Link Replacement* (S_i, M_i) *Procedure*;
If (Link Replacement fails)
7. M_i proceeds to Step 3;
//Executes *Subroute Construction Procedure* to repair the sub-path;
8.Step 3: M_i executes *Subroute Construction Procedure* (S_{i-1}, S_{i+1});
9. **CASE 2:** If (S_i connects to one M_i and has two weak links)
10. Step 1: S_i executes *Member Collection* (M_i, M_{i+1}) *Procedure* and proceeds to Step 2;
//Node Replacement
11. Step 2: S_i executes *Node Replacement* (S_i, M_i) *Procedure*;
12. If (Node Replacement fails)
13. Weak node S_i proceeds to Step 3;
//Link Replacement
14. Step 3: Weak node S_i executes *Link Replacement* (S_i, M_i) *Procedure*;
15. If (Link Replacement fails)
16. S_i proceeds to Step 4;
//Executes *Subroute Construction Procedure* to repair sub-path;
17. Step 4: S_i executes *Subroute Construction Procedure* (S_{i-1}, S_{i+1});
18. **CASE 3:** If (S_i connects to two M_i and having one or two weak links)
19. Step 1: GO to Step 1 of Case 2;

Table 6

Algorithm 6: Subroute Construction Procedure
Notations:
1. P_i: *Routing piconet*;
2. M_i: *Routing master*;
3. S_i: *Weak node* (*pure slave or bridge node*);
4. S_{i-1}: *Source node*;
5. S_{i+1}: *Destination node*;
Subroute Construction (S_{i-1}, S_{i+1}) **Procedure**
1. STEP 1: M_i or S_i notifies S_{i-1};
2. STEP 2: S_{i-1} executes local flooding to find destination node S_{i+1};
3. STEP 3: S_{i+1} transmits the packet containing location of forwarding nodes and itself along the shortest subroute of the original path to S_{i-1};
4. STEP 4: S_{i-1} constructs the subroute with S_{i+1};

executes the *Link Replacement* (S_{11}, M_1) *Procedure* and also finds that no two devices can connect to each other. Weak node S_{11} initiates the *Subroute Construction Procedure* (B_{61}, D) and then the new routing path $B_{61} \longrightarrow B_{12} \longrightarrow B_{23} \longrightarrow D$ is constructed. Node B_{23}

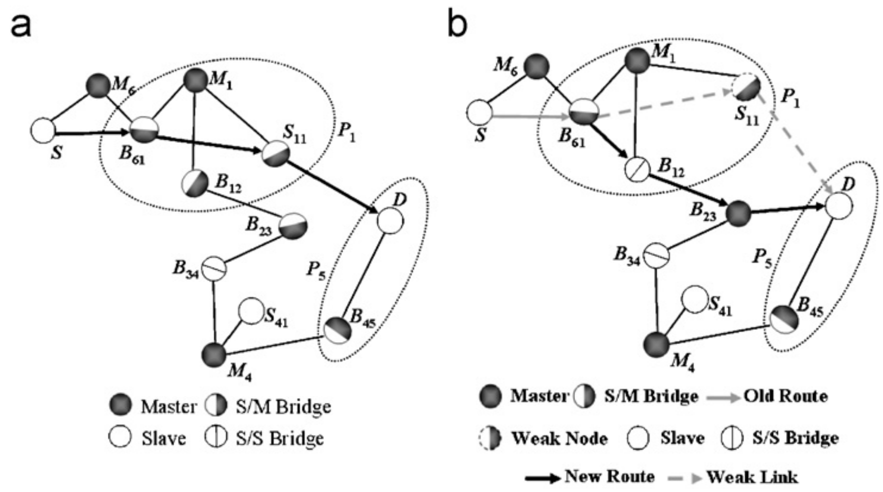

Fig. 12. Example for executing Subroute Construction Procedure before and after mobility of a node. (a) Execution of Subroute Construction Procedure: before movement of node S_{11}. (b) Execution of Subroute Construction Procedure: after movement of node S_{11}.

combines the piconets and becomes the master of nodes B_{12} and D. Thus, node B_{12} becomes a S/S bridge node.

4. Performance analysis

4.1. Simulation setups

In our simulation, initially a connected scatternet is taken with fixed numbers of 100 Bluetooth devices, which are randomly distributed over a squared area of $50 \times 50\,m^2$. The routes are chosen based on the transmission of control packets from the source to the destination and initial routing paths are generated using C++ programming. In our simulation, communication range of each Bluetooth device is fixed up to 10 m and 50 pairs of source and destination are randomly selected to construct the routes using RVM, LORP, LARP and ROMA. The Constant Bit Rate (CBR) model is used to generate the traffic load for each route in the performance evaluation and traffic arrival rate is kept at 100 kbps. The random way point model is considered as the mobility model in our simulation. New routing paths are regenerated by adding or taking away of nodes. The control packets are sent from one node to another and all possible successful paths between the source and the destination are simulated taking mobility into consideration. Thus, the average route construction time and routing path length are estimated for different numbers of mobile nodes. Finally, the performance results of our ROMA is compared with RVM (Bhagwat and Segall, 1999), LORP (Yu et al., 2007) and LARP (Chang et al., 2007), as follows.

4.2. Performance results

It is to be noted that our ROMA consider the mobility of the nodes to maintain the route. Hence, our performance analysis is made to evaluate the average number of hop counts, end-to-end delay, bandwidth consumption and required number of control packets

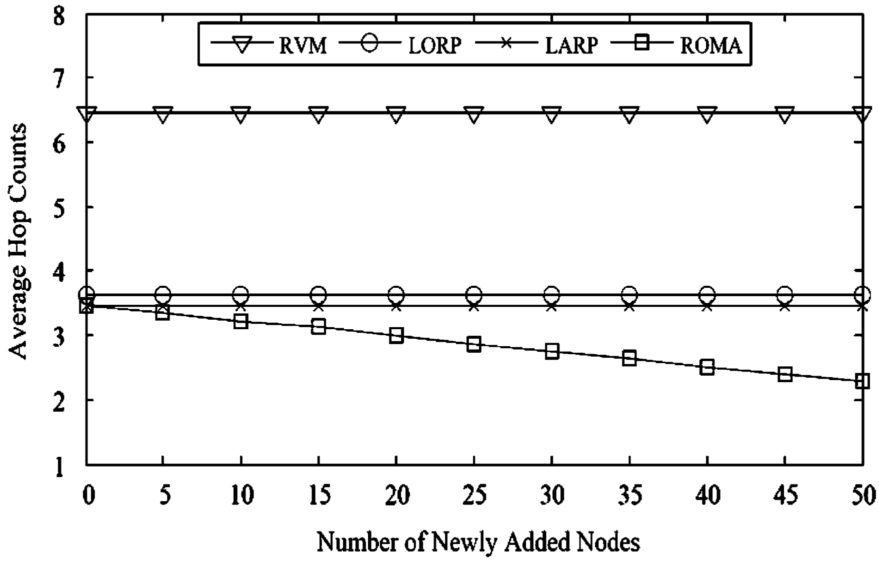

Fig. 13. Average number of hop counts for different number of newly added nodes.

based on the rate of mobile nodes and their average mobility speed. The performance analysis results are given in detail, as follows.

4.2.1. Average hop counts

As shown in Fig. 13, average number of hop counts for different number of newly added nodes are simulated with different routing protocols that we have considered. It is observed that our protocol outperforms in terms of number of hop counts as compared to RVM, LORP and LARP, when more new nodes are added to the existing scatternet. As per our algorithm, we got the most expected results as number of hops are reduced in ROMA due to addition of new nodes. In our simulation, we have analyzed the average number of hop counts for different rate of moving devices as shown in Fig. 14. The average speed of each moving device is considered as 2 m/s in the simulation. From the simulation results, it is observed that the average number of hop counts of our protocol is less than that of the RVM, LORP and LARP. It happens, since the higher rate of moving devices break more links. In case of RVM, LORP and LARP, new and worse routes are found after a link is broken as a result of which average hop counts are raised in these protocols, when number of moving devices is increased. However, ROMA uses the node replacement and link replacement policies to compensate the broken links, by which the existing routing path remains unchanged or increase slightly. The improvement in terms of average number of hop counts is very well evident from Fig. 14. The average number of hop counts for differentaverage speed of the nodes is shown in Fig. 15. It is observed that our protocol gives tremendous improvement in terms of hop counts for different average speed of the nodes. In RVM, LORP and LARP, they initialize their protocols to find a new and worse routing path from the source to the destination if a link of the routing path is broken. Therefore, the average hop counts of RVM, LORP and LARP are increased, when average speed is less than or equal to 3 m/s. However, when average speed is larger than 3 m/s, the scatternet topology is changed and thereby RVM, LORP and LARP may find a better route than the original one. Besides, ROMA executes the node replacement, link replacement or local LARP policies to maintain or increase the route length, if a link of a

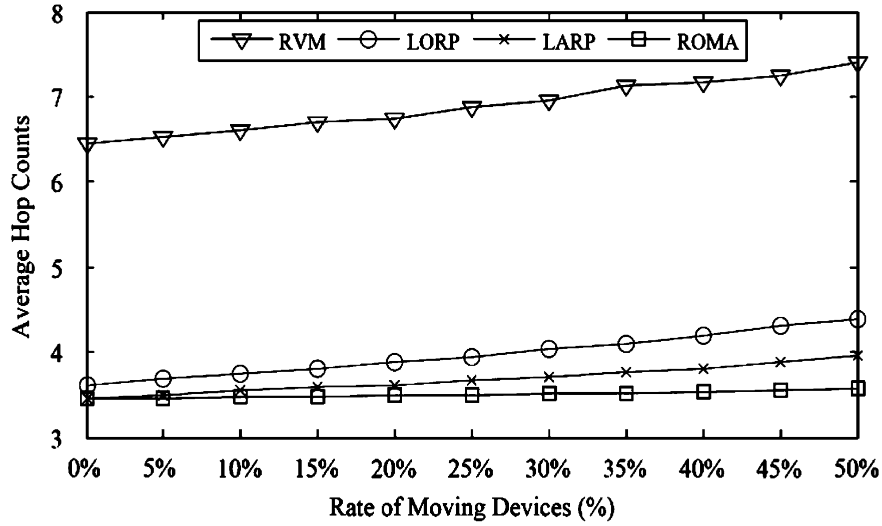

Fig. 14. Average number of hop counts in different protocols for different rate of mobile devices.

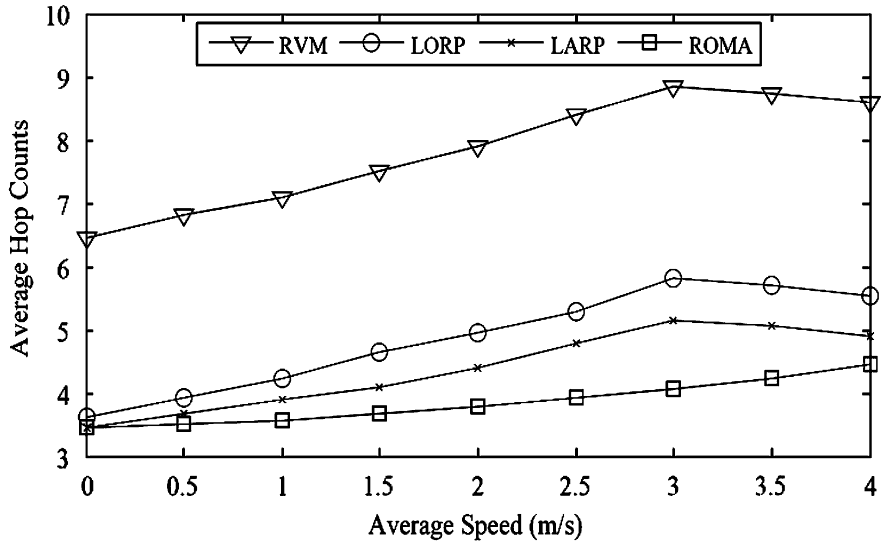

Fig. 15. Average number of hop counts for different average speed of the nodes.

routing path is broken. Therefore, if average speed is larger, the movement of a node easily causes to break two links simultaneously. Hence, ROMA easily executes the link replacement or local LARP policies to increase the hop counts of the routing path. In our simulation, we have analyzed the average number of control packets for different routing protocols such as RVM, LORP, etc.

4.2.2. End-to-end delay

From Fig. 16, it is observed that the average end-to-end delay of our protocol is less than that of the RVM, LORP and LARP. When, number of newly added nodes is less than or equal to 30, the average end-to-end delay of RVM, LORP, LARP and ROMA is increased, since newly added nodes extend the polling time of the masters. However, we find that the average end-to-end delay of ROMA is decreased when number of newly added nodes is more than 30. This is because the shortest routing paths improve the

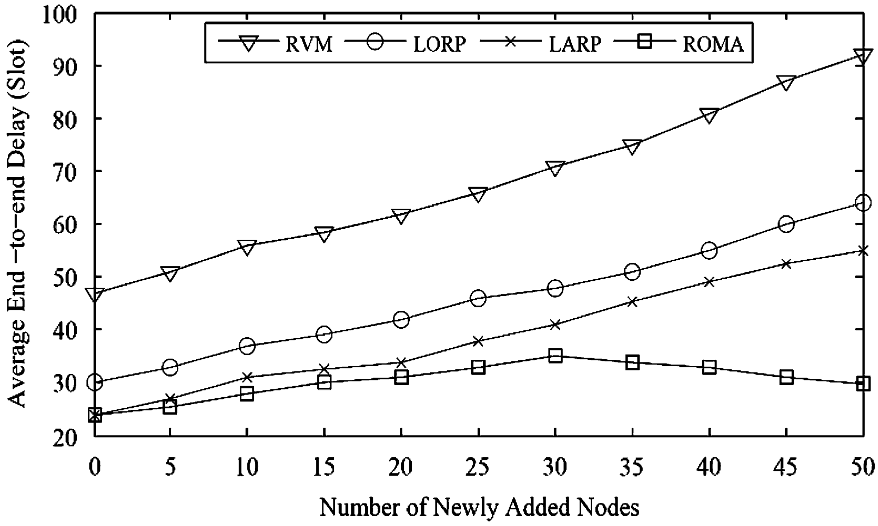

Fig. 16. Average end-to-end delay for different number of newly added nodes.

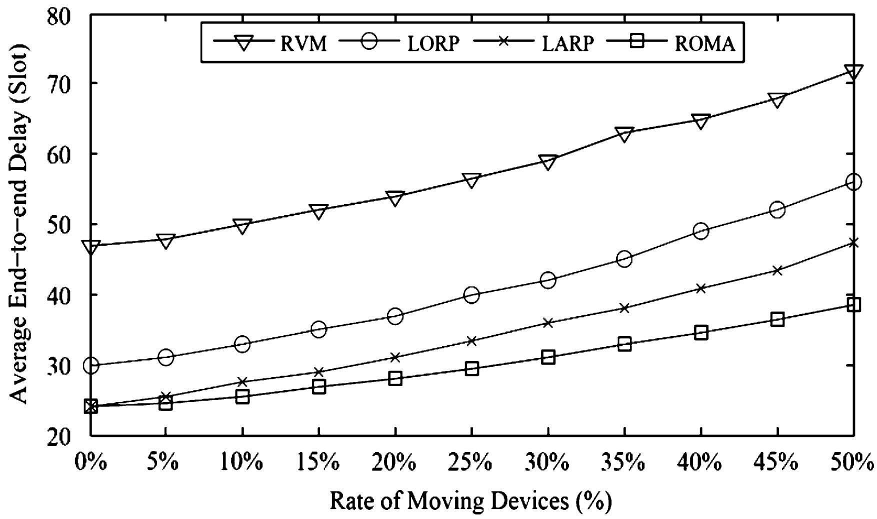

Fig. 17. Average end-to-end delay in different protocols for different rate of mobile devices.

average packet transmission delay. Fig. 17 shows the average end-to-end delay in different protocols for different rate of the moving devices. It is observed that our protocol outperforms over RVM, LORP and LARP. Since, moving devices always affect the routing path by breaking the links, route maintenance increases the packet delay time. When number of moving devices is increased, the existing links are broken more frequently by the devices and thereby the average end-to-end delay of RVM, LORP and LARP is increased, as they do not consider the route maintenance. However, due to route maintenance in ROMA, the average end-to-end delay is better than other protocols as each broken link is reconfigured.

Fig. 18 shows average end-to-end delay of different protocols for different average speed of the mobile nodes. In the simulation, the end to end delay includes the maintenance time that RVM, LORP and LARP construct a new and worse route and that ROMA executes node leaving procedure to repair the broken links. It is observed that the average end-to-end delay of our protocol is less than that of the RVM, LORP and LARP, since

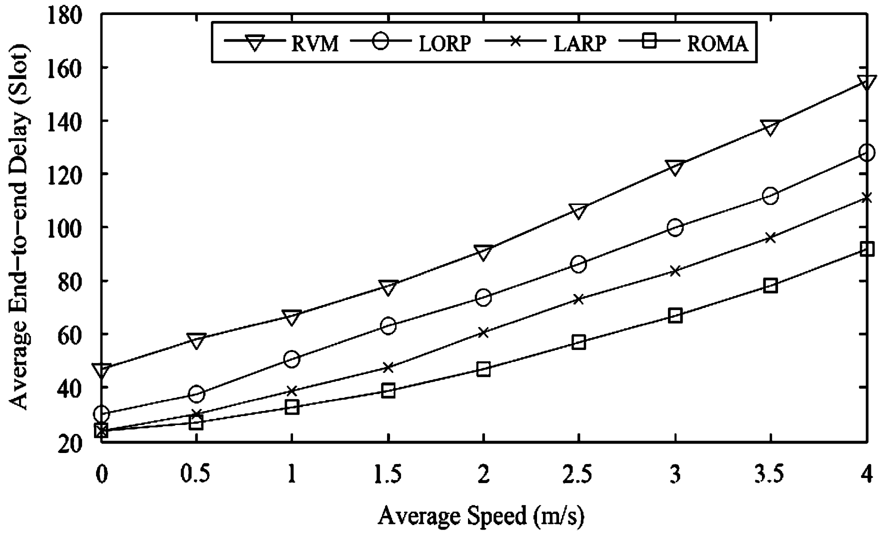

Fig. 18. Average end-to-end delay for different average speed of the nodes.

ROMA locally executes the ROMA to repair the links. Besides, it is to be noted that the higher average speed causes the links broken easily and thereby raises the delay time in executing the route maintenance. Although route length of RVM, LORP and LARP can be shortened for average speed is larger than 3 m/s and scatternet topology is changed, the route maintenance still incurs larger value of delay time.

4.2.3. Bandwidth consumption

In Fig. 19, we have compared the bandwidth consumption ratio for different number of newly added nodes for different routing protocols. It is found that our protocol outperforms over RVM, LORP and LARP. Since, the newly added nodes help to shorten the routing path length, it is obvious that the bandwidth consumption in ROMA is reduced.

4.2.4. Control packets requirement

In Fig. 20, it is observed that our protocol consumes least number of control packets as compared to LARP, LORP and RVM. Since, higher average speed of the nodes improves larger number of the broken links, RVM, LORP and LARP create more control traffic overhead to reestablish a route than ROMA. Besides, since LORP and LARP tries to shorten the route length, their control packets are larger than the control packets of RVM. Moreover, the route length of LARP is shorter than LORP and hence control packets of LARP are less than that of LORP. The average number of control packets that are required for different protocols with different rate of moving devices is shown in Fig. 21. The rate of the moving devices is defined as the number of nodes moved out of the routing piconet from total number of nodes of the scatternet. From the figure, it is observed that our protocol requires least number of control packets as compared to LARP, LORP and RVM, when rate of moving devices is larger than 20%. Since, higher number of mobile devices cause larger number of broken links, RVM, LORP and LARP require more number of control packets to maintain the existing route. However, when rate of moving devices is less than 20%, RVM outperforms to ROMA. Since, RVM uses least control packet to construct the original route and the broken links are fewer, when rate of mobile

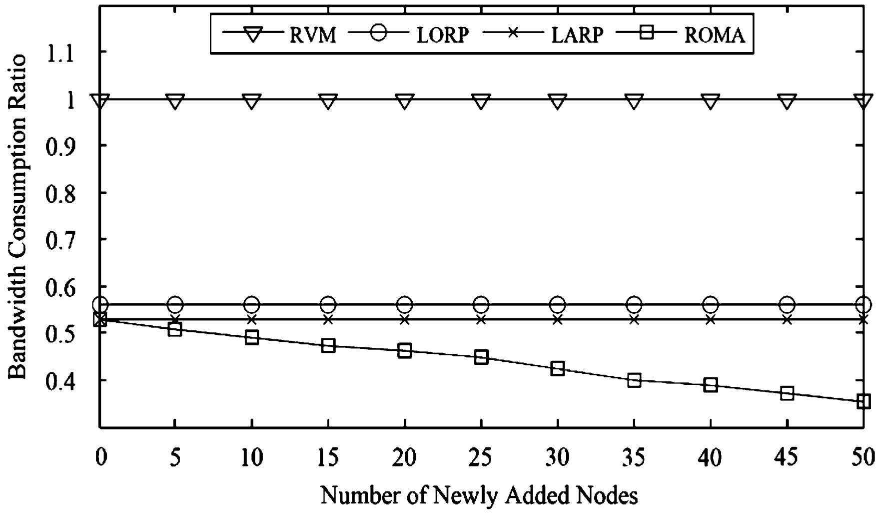

Fig. 19. Bandwidth consumption ratio for different number of newly added nodes.

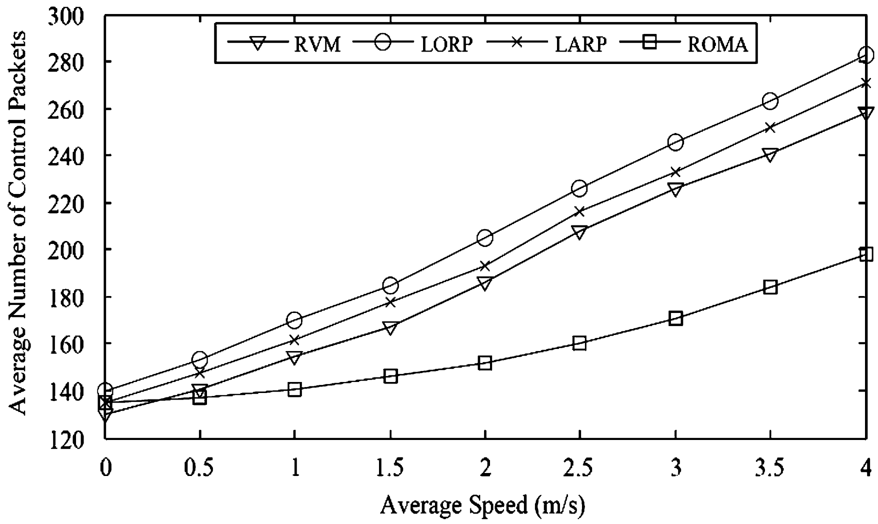

Fig. 20. Required average number of control packets for different average speed of the mobile nodes.

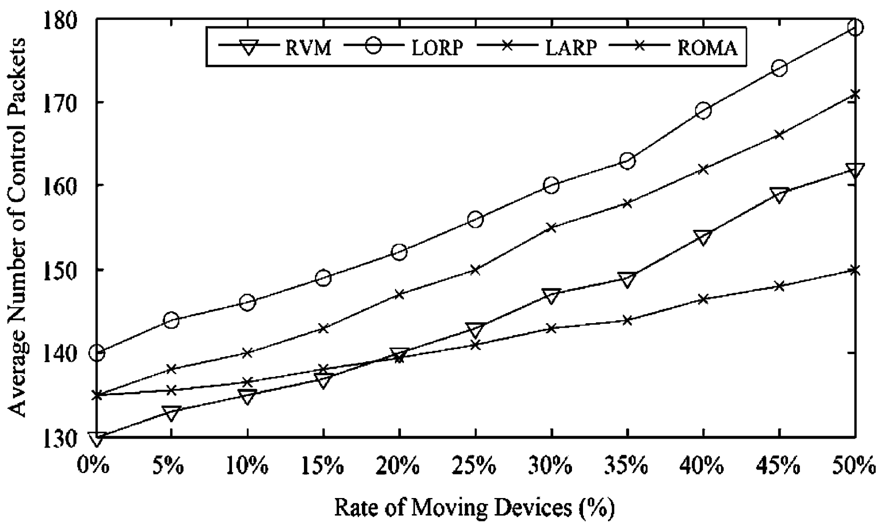

Fig. 21. Average number of control packets for different rate of mobile nodes.

devices is less, the average number of control packet of RVM is less than that of ROMA. Besides, as LORP and LARP require to shorten the routing path, their control packets are largerthan control packets of RVM.

5. Conclusion

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

References

Bhagwat P, Segall A. A routing vector method (RVM) for routing in Bluetooth scatternets. In: Proceedings of the IEEE workshop MOMUC, 1999. p. 375–9.

Bluetooth Nokia ⟨http://www.nokia.com/phones/6210/bluetooth.html⟩.

Bluetooth Specification ⟨http://www.bluetooth.org⟩ 1.0b and 1.1.

Bridgelall R. Enabling mobile commerce through pervasive communications with ubiquitous RF tags. In: Proceedings of the IEEE wireless communications and networking (WCNC), vol. 3, 2003. p. 2041–6.

Chang C-Y, Sahoo PK, Lee S-C. A location-aware routing protocol for the bluetooth scatternet. Wireless Pers Commun 2007;40(1):117–35.

Chang R-S, Chou M-T. Blueline: a distributed Bluetooth scatternet formation and routing algorithm. J Inform Sci Eng 2005;21:479–94.

Darling A. Waiting for the m-commerce explosion. Telecommun Int 2001;3:34–9.

Engebretson J. Opportunities for growth. Wireless Asia, 2006.

First Handheld Bluetooth RFID Reader. Electronic Devices, September 2004.

Gonzalez-Castano FJ, Garcia-Reinoso J. Bluetooth location networks. In: Proceedings of the IEEE Globecom, 2002.

Gonzalez-Castano FJ, Garcia-Reinoso J. Survivable Bluetooth location networks. In: IEEE ICC, vol. 2, 2003. p. 1014–8.

Harter A, Hopper A, Steggles P, Ward A, Webster P. The anatomy of a context-aware application. In: Proceedings of the 5th annual ACM/IEEE international conference on mobile computing and networking, 1999. p. 59–68.

⟨http://www.blueblitz.com/⟩.

⟨http://www.patentstorm.us/patents/6717516.html⟩.

Kapoor R, Gerla M. A zone routing protocol for Bluetooth scatternets. In: Proceedings of the IEEE wireless communications and networking conference (WCNC), 2003. p. 1459–64.

Korhonen J, Ojala T, Klemola M, Vaananen P. mTag—architecture for discovering location specific mobile web services using RFID and its evaluation with two case studies. In: Proceedings of the IEEE advanced international conference on telecommunications and international conference on internet and web applications and services, 2006. p. 191.

Martin J, Trummer C. SCALEX—A personalized multimedia information system for museums and exhibitions. In: Proceedings of the first international conference on automated production of cross media content for multi-channel distribution, 2005.

Mirza AB, Pollard JK. Dynamic source routing in Bluetooth personal area networks. Int J Simulation 2002;5(1–2):58–60.

Mobile RFID Activities in Korea. Electronics and Telecommunications Research Institute (ETRI), Expert Group Meeting on RFID, 2005.

Ni LM, Liu Y, Lau YC, Patil AP. LANDMARC: indoor location sensing using active RFID. In: Proceedings of the first IEEE international conference on pervasive computing and communications, 2003.

Perkins C, Bhagwat P. Destination sequenced distance vector routing for mobile computers. In: Proceedings of the ACM SIGCOMM, 1994. p. 234–44.

Perkins C, Royer E. Ad-hoc on-demand distance vector routing. In: Proceedings of the IEEE workshop on mobile computing systems and applications, 1999. p. 90–100.

Priyantha NB, Chakraborty A, Balakrishnan H. The cricket location-support system. In: Proceedings of the sixth annual ACM international conference on mobile computing and networking, 2000.

Sangani K. RFID sees all. IEE Rev 2004;50(4):22–4.

Shepard S. RFID. McGraw-Hill Networking Professional.

Song O, Lim C, Choi C-H. Mobility management in Bluetooth ad hoc networks. In: Proceedings of the 14th joint conference on communications and information, 2004.

Stanford V. Pervasive computing goes the last hundred feet with RFID systems. In: Proceedings of the IEEE pervasive computing, vol. 2, no. 2, June 2003. p. 9–14.

Sun M, Chang CK, Lai TH. A self-routing topology for Bluetooth scatternets. In: Proceedings of the international symposium on parallel architectures (I-SPAN), 2002.

Texas Instruments TIRIS ⟨http://www.ti.com/tiris/default.htm⟩.

Vershney U, Vetter RJ, Kalakota R. Mobile commerce: a new frontier. Computer 2000;10:32–8.

Werb J, Lanzl C. A positioning system for finding things indoors. IEEE Spectrum 1998;9(35):71–8.

Yang C-H, Ruan J-W. On-demand routing for Bluetooth scatternets subject to device mobility. In: Proceedings of the 19th international conference on advanced information networking and applications, 2005.

Yu G-J, Chang C-Y, Shih K-P, Lee S-C. Relay reduction and route construction for scatternet over Bluetooth radio systems. J Network Comput Appl 2007;30:728–49.