Available online at www.sciencedirect.com
ScienceDirect
Information Sciences 177 (2007) 3161–3177
INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL
www.elsevier.com/locate/ins

A location-aware multicasting protocol for Bluetooth Location Networks

Chih-Yung Chang [a,*], Kuei-Ping Shih [a], Chung-Hsien Hsu [b], Hung-Chang Chen [a]

[a] Department of Computer Science and Information Engineering, Tamkang University, Taipei, Taiwan
[b] Department of Communication Engineering, National Chiao-Tung University, Hsinchu, Taiwan

Received 12 December 2005; received in revised form 11 December 2006; accepted 15 December 2006

Abstract

Bluetooth Location Network (BLN) is a Bluetooth radio network that is composed of some mobile Bluetooth devices and static Bluetooth units, and is established at the system initialization to form a spontaneous network topology. In a BLN, a multicast service is defined as the periodical delivering of messages from a Service Server to a set of mobile devices which are the multicast members predefined by the Service Server. Several multicast protocols have been proposed for the Ad-Hoc networks, but they create an inefficient multicast tree for the BLN due to the existing differences in the radio characteristics between Ad-Hoc and Bluetooth radio networks. The present paper analyzes these differences and proposes a novel multicasting protocol for constructing an efficient multicast tree in a BLN. The proposed protocol constructs a multicast tree with good features which include the shortest path, a higher degree of path sharing, and fewer forwarding nodes. Simulation results reveal that the proposed multicast protocol outperforms the existing multicast protocols in the BLN.

Keywords: Location aware; Bluetooth; Bluetooth Location Networks; Multicast; Path sharing

1. Introduction

Bluetooth is a low-cost, low-power and short-range communication technology. To avoid the co-channel interference, the radio frequency (RF) module hops over 79 channels at a speed of 1600 times per second. The designs of the short packet and the fast hopping increase the communication reliability. A piconet comprises up to eight active Bluetooth devices and includes one master and up to seven active slaves. The master of a piconet manages the schedule of data transmission of its slaves [2].

To support inter-piconet communication, a scatternet is a network that consists of more than one piconet. A mobile device that participates in two or more piconets is defined as a *bridge*. The S/S (or Slave/Slave) bridge simultaneously participates in more than one piconet and alternatively plays the role of slave in the

* Corresponding author.
E-mail addresses: cychang@mail.tku.edu.tw (C.-Y. Chang), kpshih@mail.tku.edu.tw (K.-P. Shih), chhsu.cm94g@nctu.edu.tw (C.-H. Hsu), ksas@wireless.cs.tku.edu.tw (H.-C. Chen).

0020-0255/$ - see front matter
doi:10.1016/j.ins.2006.12.007

participated piconets. The *M*/*S* (or Master/Slave) bridge is a device that plays a master role in one piconet and a slave role in at least one piconet. A bridge can deliver messages from one piconet to another so that the data transmission service can be provided over a scatternet.

In the literature, a number of papers [7,9,12,19,20,22] have developed routing protocols for 802.11 Ad-Hoc networks based on the principle of on-demand route discovery. Routes to a destination are sought only if the node has data to send to that destination. Packet flooding is extensively used to establish a routing path from the source host to the destination. Most of the routing approaches flood the network with a broadcast query when the route is desired. The node receives and then replies to this query if it is a destination host or else, it simply forwards the broadcast query. By considering all of the possible paths linking the source and the destination, the source host can ascertain the shortest communication path.

Based on the flooding scheme, a Routing Vector Method (RVM) [1] was proposed to construct a routing path for a pair of Bluetooth devices. Similar to the on-demand routing protocols proposed for Ad-hoc networks, RVM floods a search request over the scatternet to find the destination device. Afterwards, the destination device replies to this query and constructs the route. Another routing protocol, called BlueRing [15,17], was developed to construct a ring structure for a scatternet. Although routing on the BlueRing scatternet is stateless, the lengths of the routes are long and thus, result into inefficient communication. BlueMesh [18] was introduced to establish a connected mesh scatternet which is characterized by a small network diameter, disjoint paths between any pair of devices, and easier routing and scheduling. However, the routing is developed on a special topology of the scatternet which is unlike the usual scatternet constructed in most of the previous researches. Hence, existing routing mechanisms developed for the Bluetooth scatternet cannot be applied to provide multicast service in Bluetooth radio networks.

In a Bluetooth scatternet, the multicast service provides data transmission from the source Bluetooth device to multiple multicast member devices which may belong to different piconets. A multicast protocol requires the construction of efficient communication paths over a scatternet from a source to all the destinations. One simple way of constructing a multicast tree is to repeatedly apply the existing routing protocol in order to construct a route from the same source to each destination. However, it will not only create a large amount of control packets but also an inefficient multicast tree. On the other hand, an efficient multicast tree minimizes the end-to-end delay and reduces the power and bandwidth consumption. Thus, the constructed multicast tree should have properties such as having the shortest path, few forwarding nodes and common link sharing.

A number of multicast protocols [3,4,8] have been proposed for wireless Ad-Hoc networks based on 802.11 radio characteristics. Previous studies [3,4,8] proposed a shared tree scheme to reduce the bandwidth consumption. However, the radio characteristics between 802.11 and Bluetooth radios are different. In the 802.11-based Ad-Hoc network, two devices can communicate directly with each other if their distance is smaller than the radio communication range. On the other hand, two Bluetooth devices belonging to different piconets cannot communicate with each other even if their distance is smaller than the radio communication range, since their channel hopping sequences are different. Hence, applying the existing multicast protocols on the Bluetooth scatternet cannot create an efficient multicast tree.

The present paper aims at developing an efficient multicast protocol for providing multicast service in a BLN. With the assumption that the BLN is aware of the location of each mobile device, we proposed an efficient multicast protocol with good features such as the shortest route, a higher degree of path sharing, and fewer forwarding nodes. The remainder of the present paper is organized as follows. Section 2 presents the background which includes the Bluetooth link construction procedure and the Bluetooth Location Networks. The basic concept of the proposed relative coordinates-based multicasting protocol (or RCMP for short) is also proposed. Section 3 describes in detail the proposed RCMP, while Section 4 investigates the performance study of the proposed RCMP. Section 5 concludes the paper.

2. Backgrounds and basic concepts

The multicast protocol proposed in the present paper was developed based on the environment of the Bluetooth Location Networks (BLNs). This section introduces the link construction and the radio characteristics of the Bluetooth networks, and then followed by an introduction to the Bluetooth Location Networks. The network model and the investigated problem are also presented.

2.1. Link construction and scatternet formation

Bluetooth radio networks adopt frequency hopping to avoid possible interferences which exist in the 2.4 GHz ISM band. In a piconet, the master and its slaves adopt the common hopping sequence which can be derived from the information of the Master's 48-bit Bluetooth address and its clock. The link construction protocol defines how the slaves can obtain the master's Bluetooth address and clock information. When a master intends to connect with one or more slave devices, it initially remains in the Inquiry state and tries to obtain the information of Bluetooth address and clock of each slave. A slave device would stay in the Inquiry-Scan state and try to send its Bluetooth address and clock information to the master. Afterwards, the slave device switches to the Page-Scan state using its own Bluetooth address and clock to derive the hopping sequence that will be used in that state. When the master device intends to connect with a slave in a Page Scan state, it changes to the Page state and switches to the channel that was previously derived from the slave's Bluetooth address and clock. In the Page state, the master sends to its slave an FHS packet containing its own Bluetooth address and clock information, plus a 3-bit active member address (or AM_ADDR in short). The master's Bluetooth address and clock would help the slave to derive the hopping sequence, while the active member address is used for the slave's identification. Afterwards, the link is constructed.

A time slot defined as 625 μs is a time period derived by partitioning one second into 1600 fragments. In a piconet, the channel is shared using a slotted time division duplex (TDD) mechanism where the master starts its transmission in the even-numbered time slots (*master-to-slave*), and the slave that receives the packet from the master in the even-numbered slot will be allowed to start its transmission in the odd-numbered time slots (*slave-to-master*). To save on power consumption, Bluetooth specifications define three types of power saving modes namely, Park, Hold, and Sniff modes. According to the master's 48-bit Bluetooth address, a unique hopping sequence is generated for its piconet. In a piconet, the master and its slaves apply the same hopping sequence to select the channel for communication. Since the Bluetooth address of the masters in different piconets are different, different piconets adopt different frequency hopping sequences to prevent co-channel interferences during data transmission.

2.2. Bluetooth Location Networks

González-Castaño and García-Reinoso [6] proposed a Bluetooth Location Network (BLN) to provide Bluetooth mobile devices with a location-aware service. The BLN consists of some mobile Bluetooth devices and static Bluetooth units which are established during system initialization to cover the whole target area. Herein, the *Bluetooth Stations* (*BS*s) refer to the Static Bluetooth units. In a BLN, users carry either a Bluetooth-enabled handheld or any mobile data terminal with a Bluetooth badge. A Service Server (or several Service Servers) in the BLN collects the user's location information in real-time and enables it to send the context-aware information to the users' handheld devices via the BLN. The location information is transmitted automatically in the BLN without the user's participation.

Fig. 1a shows the configuration of a BLN. The target area is partitioned into several equally-sized hexagon zones called *cell*s and a *BS* is arranged in the center of each cell. The distance between any two *BS*s is 10 m to ensure their direct communication in a wireless manner. A ***Service Server*** is a *BS* which acts as an interface between the BLN and the outside network. All *BSs* perform Inquiry and Inquiry Scan procedures alternately and periodically in order to scan their surroundings and to publish their existence individually. If *BS i* receives a Bluetooth address and clock information from *BS j*, and *j* is not its slave, *i* must execute the Page procedure to establish the connection with *j*. As a result, each *BS* can become the master of its six surrounding slave *BS*s and a slave of its six neighboring *BS*s. On the other hand, all mobile badges perform the Inquiry Scan process to publish their existence. It was observed that the mobile badges do not try to establish connections with the *BS*s. They simply answer inquiries with the FHS packet which do not violate the seven-slave constraint of the Bluetooth specification. Then the *BS*s report the sensing information about the mobile badge to the Service Server which, in turn, uses the concepts of overlapped ranges and sensing information to determine the location of each mobile badge.

Fig. 1b shows how the BLN obtains the location information of each mobile device. As a mobile device appears in the BLN, stations BS_2, BS_7, and BS_8 detect its existence and acquire its Bluetooth address and

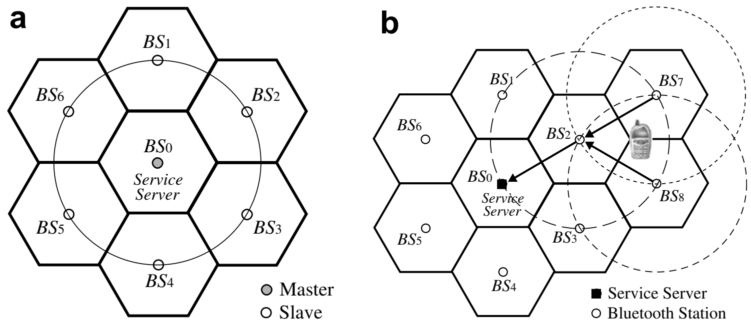

Fig. 1. Bluetooth Location Network architecture: (a) Bluetooth Stations and Service Server in a BLN, (b) example of obtaining location information of a Bluetooth mobile device in BLNs.

clock information. Then the three stations report their location information and the mobile device's Bluetooth address back to the Service Server. Upon receiving the report message, the Service Server measures the overlapped area and estimates the possible location of the mobile device so that it can provide the location-aware services in the BLN.

In a BLN, the present paper investigates how the multicast service can be provided to the multicast members that are mobile devices predefined by the Service Server. Let *H-BS* denote the *BS* that contains at least one mobile device in its radio transmission range. A simple but inefficient way of constructing a multicast tree is to apply the existing unicast protocol repeatedly to construct the shortest path from Service Server to each *H-BS*. Fig. 2 shows the constructed multicast tree where eight *BS*s, namely BS_2, BS_6, BS_7, BS_{12}, BS_{13}, BS_{14}, BS_{15}, and BS_{16}, and eight *H-BS*s, namely BS_1, BS_5, BS_8, BS_{10}, BS_{11}, BS_{17}, BS_{18}, and BS_{19}, participate in the constructed tree.

To reduce the number of forwarding *BS*s, the developed protocol called RCMP, tries to find a forwarding *BS* from the neighboring *BS*(s) which can deliver multicast packets to the largest number of members. The basic concept of the RCMP is described below. Initially, the Service Server chooses the *H-BS* according to the number of mobile devices located in its transmission range. Based on the relative coordinates, the Service Server selects the optimal forwarding *BS*(s) to forward the multicast packets to each *H-BS*. For example, in Fig. 3 the BS_5 is the selected *H-BS* which is responsible for forwarding packets to member m_2 and m_3, and the BS_2 is selected to forward packets to BS_5. Furthermore, the BS_2 is also responsible for forwarding packets to BS_{10} and BS_{11}. The same process, which is used to select the optimal forwarding *BS*(s) by the Service Server,

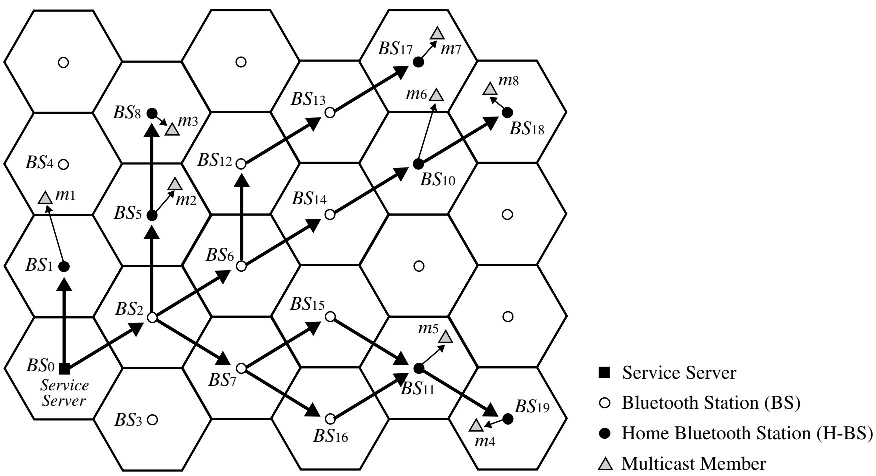

Fig. 2. A constructed multicast tree obtained by executing the existing unicast protocol repeatedly.

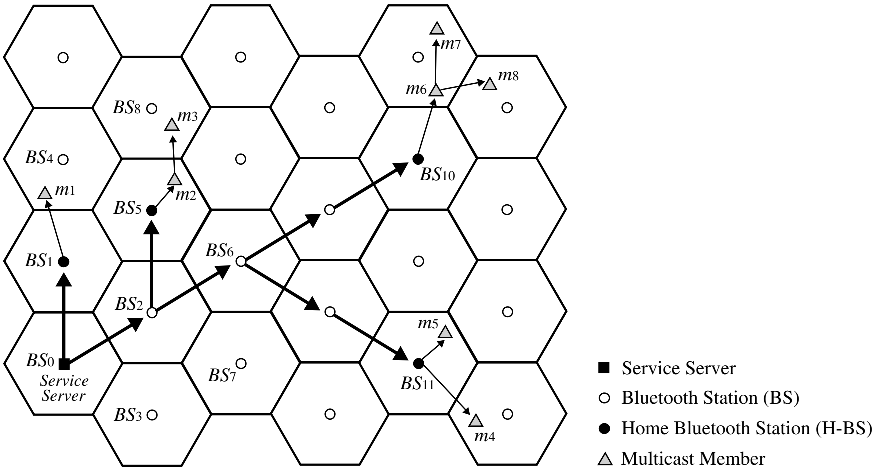

Fig. 3. A constructed multicast tree obtained by applying the proposed multicasting protocol.

will also be applied by each forwarding *BS* in a distributed manner. As a result, the multicast tree with a characteristic path sharing is constructed.

By comparing the multicast tree of Fig. 2 with that of Fig. 3, the RCMP reduces 8 forwarding *BS*s, thus saving on power and bandwidth consumption. The multicast tree which was constructed by applying the proposed RCMP has good features which include the shortest route between each source–destination pair, a higher degree of path sharing, and fewer forwarding nodes which reduce the end-to-end transmission delay and the power and bandwidth consumption. The next Section will formally present the RCMP in detail.

3. Related coordinates-based multicasting protocol

This section introduces the network model and the problem statement of the present paper and afterwards, a multicast tree construction protocol is proposed.

3.1. Network model and problem statement

The BLN architecture supports the location-aware services in environments such as malls and shops where the Service Server is able to transmit some real-time information to specific mobile users. The present paper develops a multicast routing protocol that utilizes the location information of each multicast member in order to construct an efficient multicast tree for the Bluetooth Location Network. We assume that a Service Server and a number of Bluetooth Stations have been deployed so that the target area is fully covered. Some users are equipped with Bluetooth handheld devices and the Service Server is aware of the location of each mobile device. The multicast members are mobile and intend to receive the multicast message from the Service Server.

To provide the multicast service efficiently, an efficient multicast routing protocol is required. In the literature, no multicast routing protocol is proposed for the BLN. The multicast routing protocols [4,10,13,16,21] designed for 802.11-based Ad-Hoc networks have been widely investigated in the past. However, they construct an inefficient multicast tree for Bluetooth radio networks due to the existing differences in the radio characteristics between the Bluetooth and the 802.11-based Ad-Hoc networks. Also found in the literature are some routing protocols such as RVM [1], BlueRing [14,17], and Scatternet-Route [15] which have been proposed for Bluetooth radio networks. However, none of them investigated the multicast routing problem.

The present paper aims at developing an efficient multicasting protocol to establish the multicast forwarding routes without using excess control packets. The constructed multicast tree has several good features which includes the shortest path between each source–destination pair, a higher degree of path sharing, fewer forwarding nodes, and a low control overhead for route maintenance.

3.2. The RCMP mechanism

We now define the following terminologies that will be used in describing the *RCMP*.

- **Candidate Home Bluetooth Station (*CH-BS*)**
  In the BLN, a mobile Bluetooth device may be detected by several *BS*s. The *BS* which detects a multicast member and has the minimal number of hops from it to the Service Server is referred to as the *CH-BS*.
- **Home Bluetooth Station (*H-BS*)**
  Home Bluetooth Station (or *H-BS* in short) is the *BS* which is responsible for forwarding packets to mobile Bluetooth devices (also referred to as multicast members) within its transmission range.
- **Direction (*d*)**
  The packet forwarding from each cell to a neighboring cell has six possible directions d_n where $1 \leqq n \leqq 6$.
- **Multicast routing table**
  A multicast routing table is created on demand and maintained by each of the *BS* which participates in the multicast tree. The multicast table mainly records the three fields namely, the Group ID, the Next Hop *BS* ID, and the Destination *H-BS*s. The Group ID denotes the multicast group. The Next Hop records the neighboring *BS* to which a received multicast data packet should be forwarded. The Destination *H-BS* field records the destination *BS*s which serve as the multicast members.
- **Multicast request packet**
  A multicast request packet will be initiated from the Service Server for constructing a multicast tree. There are five fields in this packet namely, the Group ID, the Sender ID, the Receiver *BS* ID, the Destination *H-BS* ID and Coordinate, and the Responsible Members. The Group ID denotes the multicast group. The Sender ID denotes the *BS* who sends this packet. The Receiver *BS* ID is the neighboring *BS* which should receive this packet. The Destination *H-BS* ID and Coordinate record the destination *H-BS* and its coordinates, respectively. The Responsible Members denote those multicast members that should be served by the corresponding *H-BS*.

Fig. 4a shows the six directions along which each *BS* may forward the received multicast data packets to its neighbors. The BLN has a coordinate system as shown in Fig. 4b. Depending on the coordinates of the *H-BS*, each *BS* can evaluate the direction for packet forwarding so that the minimal number of hops from the current *BS* to the *H-BS* can be achieved.

As shown in Fig. 5, BS_5 is a *CH-BS* since the multicast members m_2 and m_3 exist within its transmission range. The BS_5 also acts as the *H-BS* since it has the minimal number of hops from the Service Server to itself. Here, we assume that the coordinates of the Service Server is $(0,0)$ and the coordinates of BS_1, BS_2, and BS_3 are $(0,2)$, $(1,1)$, and $(1,-1)$, respectively.

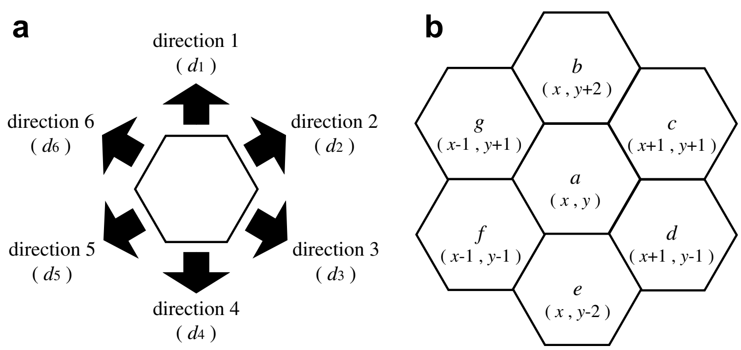

Fig. 4. The possible directions and relative coordinates of each cell: (a) six possible directions for packet forwarding, (b) relative coordinates in BLN.

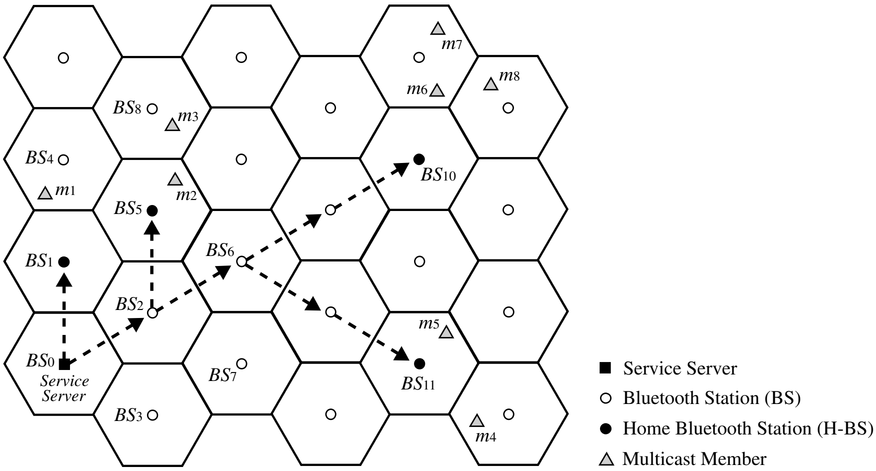

Fig. 5. The multicast tree constructed by applying the MRC phase of RCMP.

3.2.1. Candidate Direction Calculation procedure

This subsection proposes a *Candidate Direction Calculation* (*CDC*) procedure which calculates the optimal packet forwarding direction for each *H-BS*, and describes in detail the RCMP. Upon receiving the multicast request packet, each forwarding *BS* should apply this procedure to calculate which neighboring *BS* will be the next hop receiver according to the *H-BS* indicated in the packet. The "optimal direction" refers to "the route established in this direction with the shortest distance between the current *BS* and the destination *H-BS* node." Fig. 6 illustrates the concepts of this algorithm.

In Fig. 6a, the four shortest routes between nodes *S* and *D* are *S-a-c-f-D*, *S-b-c-f-D*, *S-b-e-f-D*, and *S-b-e-g-D*. All of these routes are 4 hops and can be established by traveling through the gray cells. We can observe that if node *S* intends to transmit the packet to node *D*, the shortest routes can be achieved by sending packets from node *S* in the two directions. The two directions are defined as the "optimal directions". The shortest route between nodes *S* and *D* can be achieved by forwarding the received packet along the optimal directions.

Let the *target area* in a specific direction *d* of a source cell *S* refer to the set of cells which can be reached by traveling the shortest path from cell *S*′, where *S*′ is the neighboring cell of *S* in direction *d*. Fig. 6b and c show the target areas in directions 1 and 2 of node *S*, respectively. In other words, if there is a cell included in the target area in the direction *d* of source *S*, there exists the shortest path from source *S* to this cell and that the transmission direction from source *S* to its neighboring cell follows the direction *d*. Notice that node *D* is

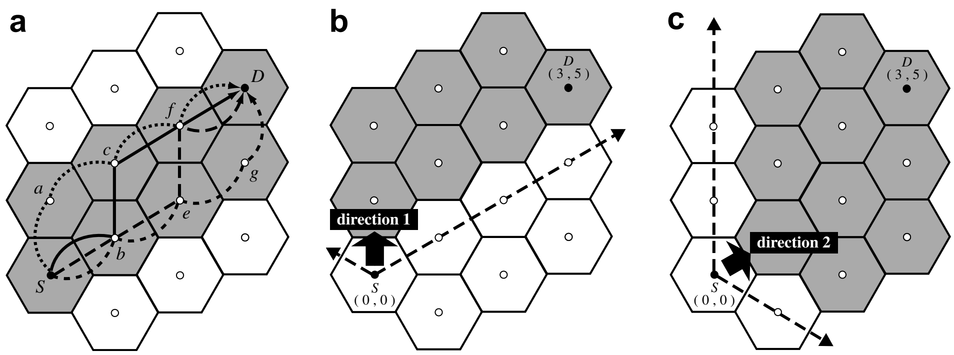

Fig. 6. The concepts of the Candidate Direction Calculation: (a) the shortest routes between *S* and *D*, (b) the target area of direction 1, (c) the target area of direction 2.

located in the target area in both directions 1 and 2 of cell S, as shown in Fig. 6b and c. This means that the shortest route between nodes S and D can be achieved by forwarding the packet from source node S to directions 1 or 2.

The proposed Candidate Direction Calculation (*CDC* in short) procedure introduced in this section is used to calculate the possible directions in which the multicast packets can be forwarded from the current cell to its neighboring cell so that the packets can arrive to each *H-BS* via the shortest path. These directions are called *candidate directions*. The following procedure derives the candidate direction for a given *H-BS*.

Candidate Direction Calculation procedure

Input: The relative coordinates of the *H-BS*
Output: The candidate directions

```
if (x ≠ 0 and y ≠ 0)
{
    if (y ⩾ 2 and y ∈ {|x| + 2k|k ∈ Z^+})
    {
        if (x ⩾ 1 and y ∈ {−x + 2k|k ∈ Z^+})
            return (direction 1 and direction 2);
        else if (x ⩽ −1 and y ∈ {x + 2k|k ∈ Z^+})
            return (direction 1 and direction 6);
        else
            return (direction 1);
    }
    else if (y ⩽ −2 and y ∈ {−|x| − 2k|k ∈ Z^+})
    {
        if (x ⩾ 1 and y ∈ {x − 2k|k ∈ Z^+})
            return (direction 4 and direction 3);
        else if (x ⩽ −1 and y ∈ {−x − 2k|k ∈ Z^+})
            return (direction 4 and direction 5);
        else
            return (direction 4);
    }
    else if (x ⩾ 1)
    {
        if (y ∈ {−x + 2k—k ∈ Z^+} and y ∈ {x − 2k|k ∈ Z^+})
            return (direction 2 and direction 3);
        else if (y ∈ {−x + 2k|k ∈ Z^+})
            return (direction 2);
        else
            return (direction 3);
    }
    else
    {
        if (y ∈ {−x − 2k|k ∈ Z^+} and y ∈ {x + 2k|k ∈ Z^+})
            return (direction 5 and direction 6);
        else if (y ∈ {−x − 2k|k ∈ Z^+})
            return (direction 5);
        else
            return (direction 6);
    }
}
```

Fig. 6b is used as an example to illustrate the *CDC* procedure. As shown in Fig. 6b, the relative coordinates of cell D to cell S is (3,5). According to the calculation rules of the *CDC* procedure, $y = 5 \geqq 2$ and $y = 5 = (x + 2 \times 1) \in \{5, 7, 9, 11, \ldots\}$, which imply that cell D is located in the direction 1's target area of cell S (line 3 of *CDC* procedure). Cell D is also located in the direction 2's target area of cell S because $x = 3 \geqq 1$ and $y = 5 = ((-x) + 2 \times 4) \in \{-1, 1, 3, 5, \ldots\}$ (line 5 of *CDC* procedure). Therefore, the candidate directions for cell S to transmit packets to cell D are directions 1 and 2 (line 6 of *CDC* procedure).

The candidate direction derived in this procedure will be adopted in the Multicast Routes Construction (MRC) phase of the RCMP and is detailed in the next subsection. It is used to construct the shortest routes for each pair of *BS* and *H-BS*.

3.2.2. Multicast Routes Construction phase

The Related Coordinates-based Multicasting Protocol (RCMP) is initiated by the Service Server in order to construct an efficient multicast tree. The first phase of RCMP is the *Multicast Routes Construction* (*MRC*) which is executed by the Service Server and each forwarding *BS*. In the MRC phase, the Service Server initially identifies the *H-BS*s. Then the Service Server executes the *CDC* procedure to select the optimal forwarding direction that can deliver the multicast packets to the largest number of *H-BS*s, and transmits the multicast request packet to the neighboring *BS* in this direction. Upon receiving the multicast request packet, the forwarding *BS* executes the *CDC* procedure according to the received request packet, and then selects the next forwarding *BS* and constructs the routes of the multicast tree in a distributed manner. When the *H-BS* receives the request packet, it executes the *Role Assignment* (*RA*) phase which is the second phase of the RCMP. In the RA phase, *H-BS* establishes the connections with its responsible members in order to complete the construction of the multicast tree. Finally, the RCMP employs the role switching and the time-slot leasing mechanisms to further reduce the transmission delay and power consumption. Thus, the constructed multicast tree has good features which include the shortest route between each source–destination pair, a higher degree of path sharing, fewer forwarding nodes, and smaller transmission delay and power consumption.

In the following paragraphs, the MRC phase is discussed in detail. Note that in the BLN environments, a member may be detected by several *BS*s and the detected information will be sent to the Service Server in a multi-hop manner. Hence, the Service Server will collect all of the members' information which will be used to calculate the location of each member device. When the Service Server intends to provide the multicast service and transmits the data packets to the members, it executes the following MRC operations to initiate the RCMP:

Step 1: ***H-BS selection***. The Service Server selects the *H-BS* from the *CH-BS* for each multicast member based on two criteria: (1) the selected *CH-BS* detects the maximum number of members, and (2) the number of hops from the Service Server to the selected *CH-BS* is minimal. In case that there are more than one *CH-BS* detecting the same number of members, the Service Server will arbitrarily select one from these candidates to play the *H-BS* role. The selected *H-BS* will be the last forwarding BS that will forward the multicast data packets to the members located within its transmission range.

Step 2: ***Candidate Direction Calculation***. The Service Server or current forwarding *BS* computes the relative coordinates of each *H-BS*. The relative coordinates of the $H\text{-}BS_j$ to the Service Server or current BS_i is $R_{ij}(x,y) = C_j(x,y) - C_i(x,y)$, where $C_i(x,y)$ and $C_j(x,y)$ stand for the coordinates of the current BS_i and $H\text{-}BS_j$, respectively. Substituting these relative coordinates as the input parameters of the *CDC procedure*, the Service Server or current *BS* can derive the target area and the candidate direction for each *H-BS*.

Step 3: ***Multicast routing table creation***. The Service Server or current *BS* executes the following operations to create its multicast routing table.

(1) Record the Group ID in the multicast routing table.
(2) Select a direction d_k, such that $|S_{H\text{-}BS}(d_k)| = \max|S_{H\text{-}BS}(d_n)|$, for $1 \leqq n \leqq 6$. The notation $S_{H\text{-}BS}(d_n)$ is a set of *H-BS*s where d_n is the candidate direction of these *H-BS*s.
(3) Record d_k, all *H-BS*s of $S_{H\text{-}BS}(d_k)$ and their coordinates in the multicast routing table.
(4) Remove d_k and all *H-BS*s of $S_{H\text{-}BS}(d_k)$ from the direction set and *H-BS* set in the received multicast request packet.
(5) Repeat operations (1)–(4), until all the *H-BS*s have been recorded in the multicast routing table.

Step 4: ***Request packet construction and transmission***. For each direction d_k, create a new multicast request packet. The Sender ID is the ID of the current *BS* and the Receiver ID is the ID of the neighboring BS in the direction d_k. The set of *H-BS* should be $S_{H\text{-}BS}(d_k)$.

Upon receiving the multicast request packet, the forwarding BS executes Steps 2, 3, and 4 and the *H-BS selection* Step is only implemented by the Service Server. Consider the BLN as shown in Fig. 5. When the Service Server intends to provide the members with the multicast service, the MCRP is initiated by the Service Server to construct a multicast tree. In this example, there are eight member devices in the BLN. The Service Server initially identifies the *H-BS*s of all the members (*H-BS selection* Step) according to the location of the member devices. In this step, BS_1, BS_5, BS_{10}, and BS_{11} will be identified to play the *H-BS* role and they are responsible for forwarding the multicast messages to the sets of mobile devices $\{m_1\}$, $\{m_2, m_3\}$, $\{m_6, m_7, m_8\}$ and $\{m_4, m_5\}$, respectively. In Step 2, the Service Server executes the *CDC* procedure and determines the candidate directions for these *H-BS*s

$$S_{H\text{-}BS}(d_1) = \{H\text{-}BS_1, H\text{-}BS_5\}, S_{H\text{-}BS}(d_2) = \{H\text{-}BS_5, H\text{-}BS_{10}, H\text{-}BS_{11}\}, \quad \text{and} \quad S_{H\text{-}BS}(d_3) = \{H\text{-}BS_{11}\}.$$

Afterwards, in the *Multicast Routing Table creation* Step, the Service Server selects direction d_1 to forward the multicast packets for destination $H\text{-}BS_1$, and also selects direction d_2 to forward the multicast packets for destinations $H\text{-}BS_5$, $H\text{-}BS_{10}$, and $H\text{-}BS_{11}$. Finally, the Service Server records these two directions and the corresponding *H-BS*s in the routing table, and then the multicast tree is constructed in the Service Server. The last step of the MCRP which is executed by the Service Server is the *Request packet construction and transmission* Step. In this step, the Service Server creates two multicast request packets, and one of them is sent to BS_1. This packet, containing $H\text{-}BS_1$ and its responsible members, indicates that the neighbor BS_1 is the *H-BS*. Another packet will be sent to BS_2. This packet, containing the destinations $H\text{-}BS_5$, $H\text{-}BS_{10}$, and $H\text{-}BS_{11}$, indicates that BS_2 should further construct a shared path to the three destinations. The Multicast Routing Table constructed in the Service Server as well as the two initiated multicast request packets are shown in Tables 1–3, respectively. Upon receiving the multicast request packet, all forwarding *BS*s execute the above-mentioned steps and hence, an efficient multicast tree is constructed.

Table 1
The Multicast Routing Table maintained in the Service Server

Group ID	Next hop (BS_n)	Destination *H-BS* ($H\text{-}BS_i$)
1	BS_1	$H\text{-}BS_1(0,2)$
1	BS_2	$H\text{-}BS_5(1,2)$ $H\text{-}BS_{10}(4,4)$ $H\text{-}BS_{11}(4,0)$

Table 2
The multicast request packet sent from the Service Server to BS_1

Group ID	Source ID (BS_m)	Destination ID (BS_n)	Destination *H-BS* ($H\text{-}BS_i$)	Responsible member (m_k)
1	BS_0	BS_1	$H\text{-}BS_1(0,2)$	m_1

Table 3
The multicast request packet sent from Service Server to BS_2

Group ID	Source ID (BS_m)	Destination ID (BS_n)	Destination *H-BS* ($H\text{-}BS_i$)	Responsible member (m_k)
1	BS_0	BS_2	$H\text{-}BS_5(1,2)$ $H\text{-}BS_{10}(4,4)$ $H\text{-}BS_{11}(4,0)$	m_2, m_3 m_6, m_7, m_8 m_4, m_5

Notice that the multicast tree construction process is executed when the Service Server transmits or when the *BS* receives the multicast request packet. Each forwarding *BS* needs only to forward the subsequent data packets to the downstream *BS*s according to the multicast routing table.

3.2.3. Role Assignment phase

The MRC phase of the RCMP aims at constructing an efficient multicast tree from the Service Server to each *H-BS*. In the *Role Assignment* (*RA*) phase, the links between each *H-BS* and its responsible members are further established. These links will form part of the constructed multicast tree.

In the BLN environment, each *BS* not only plays the master role to control six surrounding slave *BS*s, but also plays the slave role which is controlled by the six neighboring master *BS*s. The Bluetooth standard specifies that the maximum number of active slaves connected with the master node is seven. Since each *BS* in the BLN has already six neighboring *BS*s, only one connection remains for each *BS* in order to sense the surrounding mobile Bluetooth devices (members). Thus, the following limitations exist in the BLN.

Limitation 1: **The trade-off between sensing function and member connection.** There is only one available connection for each *BS* which it can use for detecting mobile devices in the BLN. If the BS decides to establish the connection with the member, it would loss the function of detecting mobile devices.

Limitation 2: **The limitation in the number of connected members.** The *BS* cannot establish connections with more than one member since the number of slaves connected to each master cannot exceed seven.

To provide a multicast service under the above-mentioned constraints and reduce the transmission delay and power consumption for multicast services, the present paper adapted the *Role Switching* [2,5] (*RS*) and *Time-Slot Leasing* [5,23,24] (*TSL*) mechanisms in the proposed *RA* phase. The RS mechanism is a technique to exchange the roles of master and slave, while the TSL mechanism for slaves was proposed to temporarily borrow some slots from their master for slave-to-slave communication.

In the RA phase, the main goal is to establish the connections between each *H-BS* and its responsible members. When the *H-BS* receives the multicast request packet, it executes the *RA* phase to complete the multicast tree construction. The following steps detail the operations in the RA phase.

Step 1: ***Main connection selection.*** The *H-BS* executes the page procedure to establish the link with a responsible member *j* (rm_j in short). At this point, *H-BS* and rm_j play the master and slave role, respectively. If there is only one *rm*, the *H-BS* goes to Step 3; otherwise the *H-BS* executes the next step.

Step 2: ***Member-to-member connection request.*** The *H-BS* sends the *mmpeg_req*(S_{rm}(*H-BS*)-{rm_j}) packet to rm_j, where S_{rm}(*H-BS*) denotes a set of *H-BS*'s *rm*s. This packet includes the BD_ADDRs and clock information of all *rm*s except for rm_j, and intends to request the rm_j to connect with other unconnected *rm*s.

Step 3: ***Role switching execution.*** The *H-BS* executes the Role Switching procedure to exchange its role with rm_j. After executing this step, the member rm_j plays the master role while the *H-BS* plays the slave role.

Step 4: ***Member-to-member connection.*** The rm_j attempts to establish connection with each unconnected *rm* when it receives the *mmpeg_req*(S_{rm}(*H-BS*)-{rm_j}) packet. Once the rm_j successfully establishes the connection with a responsible member rm_k, it removes rm_k from set S_{rm}(*H-BS*). Finally, rm_j replies with the *mmpeg_rep*(S_{rm}(*H-BS*)) packet to *H-BS* to report the results of the member-to-member connection.

Step 5: ***Time-slot leasing execution.*** The *H-BS* initiates the TSL process to lease time slots from rm_j. Throughout the duration of leasing the slots, the *H-BS* is the temp-master and the other connected *rm*s are the temp-slaves in the new piconet.

Step 6: ***RA phase completion checking.*** The *H-BS* checks the connection results from the received *mmpeg_rep*(S_{rm}(*H-BS*)) packet. If there is still some unconnected *rm*s, the *H-BS* goes back to Step 1; otherwise the RA phase is completed.

The Step 3 in the RA phase resolves the problem of limitation 1. Since the *H-BS* changes role from master to slave, the number of available connections existing in its piconet is six. Step 5 employs the TSL technology

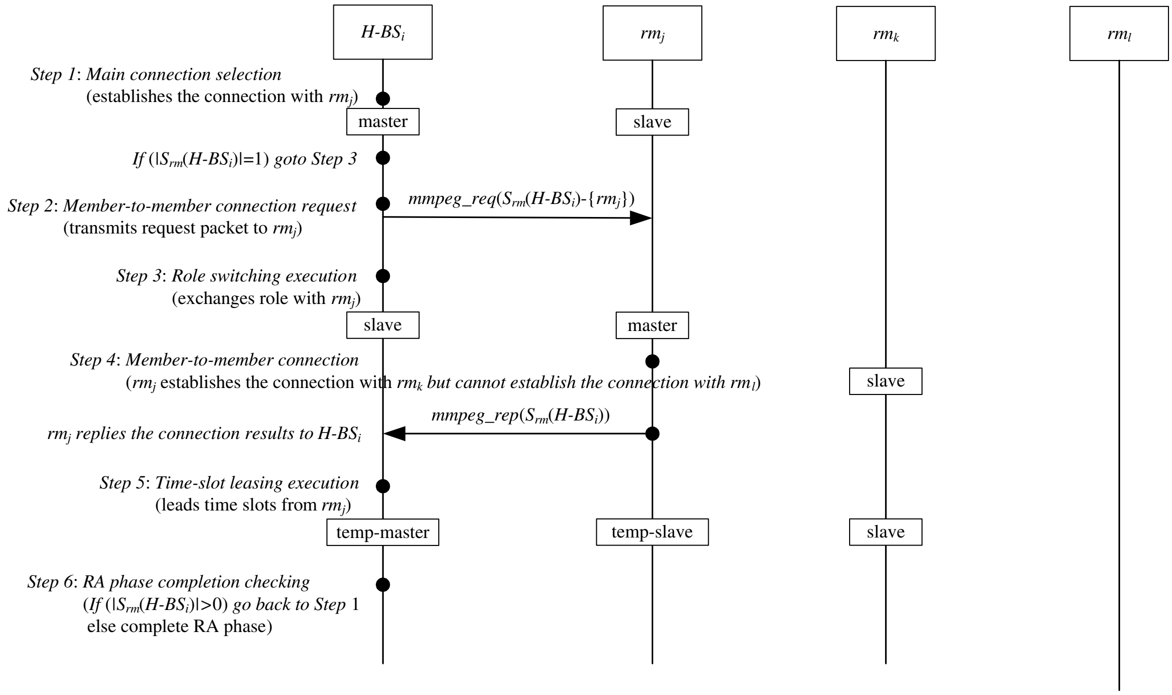

Fig. 7. Message flow in the Role Assignment phase.

to resolve the problem of limitation 2. Fig. 7 shows the message flow of the RA phase. In the slot-leasing duration, the *H-BS* leases slots from the *rm* in order to transmit the multicast data packets to the *rm*s at the same time. Moreover, the TSL technology was employed to change the authority of data transmission but not to change the devices' roles. Hence, the *H-BS* still plays the slave role although it has the authority to transmit the multicast data packets without waiting for the *rm*'s polling. In this way, the data transmission delay can be reduced significantly.

4. Performance study

To evaluate the performance of the proposed RCMP, we implemented it using the BlueHoc simulator [11] which is an ns2 [18]-based simulator released by IBM. Our simulation arranges a Bluetooth Location Network with 114 deployed Bluetooth Stations and 30 mobile multicast members in a $100 \times 100\ m^2$ square area. The radio propagation range of each Bluetooth device is 10 m, and the nominal channel capacity is set at 1 Mbps. All members are randomly generated and move at random in one of the six directions at a speed ranging from 0 to 9 m/s. The simulation time is 100 s. Constant bit rate (CBR) traffic with a rate of 10 data packets per second is used to generate traffic from the Service Server. The DH5 (containing 2870 bits per packet) and DH1 (containing 366 bits per packet) packets are used for data and control packet types, respectively.

In the simulation experiments, four transmission mechanisms namely, Flooding mechanism, Multiple Unicast Routing mechanism, Traditional Multicast Routing mechanism, and the proposed Relative Coordinates-based Multicasting mechanism are compared in terms of the average multicast degree, the control overhead, the average delay, the throughput, and the multicast service cost. In the Flooding mechanism, each multicast packet is broadcasted over the BLN. The Multiple Unicast Routing mechanism applies the Unicast routing Protocol to construct a route from the source to each member individually.

4.1. Average multicast degree

In the wireless channel, the multicast protocol can take advantage of the MAC layer broadcast. A packet transmitted by a Bluetooth Station can be received by all neighbors at the same time if a broadcast operation is utilized. An efficient wireless multicast protocol can make good use of the MAC layer broadcast facility. Thus, it is appropriate to evaluate and compare the efficiency of multicast schemes by measuring the "Average Multicast Degree" (or AMD in short) which is defined as the ratio of the "total number of data packets received at the nodes on the tree" to the "total number of data packets transmitted." If $AMD < 1$, it means that some of the nodes which do not participate in the multicast tree still receive the data packets. Note that the unicast mechanism with a possible retransmission will yield an AMD which is less than or equal to 1.

Fig. 8 compares the average multicast degree of the four schemes. The Flooding mechanism has the lowest degree of multicast because all of the *BS*s participate in the packet forwarding, and the significant redundant transmissions result to a smaller value of AMD. The proposed RCMP outperforms the other three schemes because the RCMP considers path sharing and reduces the number of forwarding nodes in the constructed multicast tree. On the other hand, the Traditional Multicast Routing protocol only considers the shortest routes between the source and each destination without considering the benefit of path sharing. Thus, the number of redundant transmissions which occurred in the traditional multicast routing protocol is larger than those of the RCMP.

4.2. Control overhead

The "Control Overhead" stands for the total number of packets used for constructing and maintaining the routing paths. The on-demand routing protocols used the route request (RREQ) packets and the route reply (RREP) packets to construct the routing paths. The route reconstruction (RREC) packets and their corresponding reconstruction acknowledgment (ACK) packets are used in the maintenance section to handle the route disconnection problems. The "Control Overhead" consists of the four types of packets mentioned above. The costs of route construction and maintenance increase with the control overhead.

Fig. 9 shows the control overhead of the compared schemes. The simulation result shows that the control overhead of the Multiple Unicast is higher than the other three schemes. In the Multiple Unicast scheme, the Service Server constructs the route to each member individually. Hence, the number of flooding of RREQ packets increases with the number of multicast members. The Traditional Multicast scheme floods RREQ packets to construct the routing paths between the Service Server and all destination members. The proposed RCMP does not use the RREQ and RREP packets to construct the routing path. Upon receiving the multicast request, *BS* dynamically determines the direction of the next forwarding nodes and automatically constructs the efficient multicast tree. Therefore, when all the members are static, the control overhead of RCMP is zero. The control overhead of RCMP increases with mobility. This is because RCMP creates the RRES and the corresponding ACK packets which are required to handle the disconnection problems raised by mobility. By comparison, the proposed RCMP outperforms the other three schemes in control overhead.

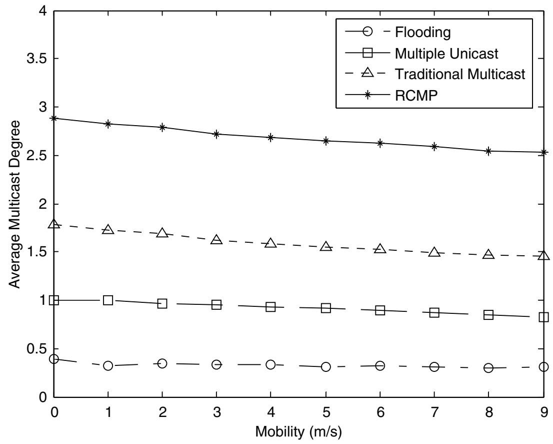

Fig. 8. Comparison of average multicast degrees.

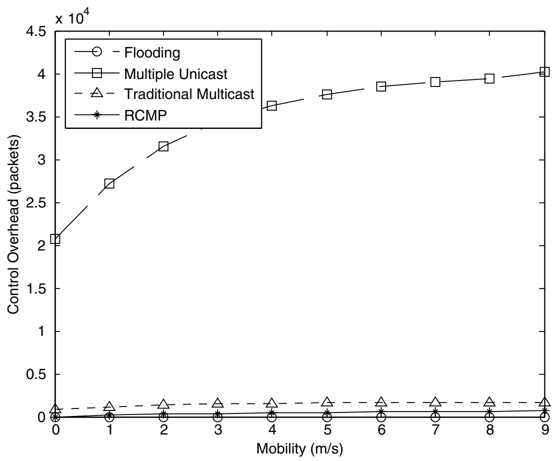

Fig. 9. Comparison of control overheads when considering mobility.

4.3. Average delay

Average delay is measured at each multicast member. Each data packet carries the time stamp set by the Service Server. The total delay is accumulated and averaged over the entire simulation run. The delay consists of the transmission delay and queuing delay. The transmission delay and queuing delay refer to the duration of data packet transmission from sender to receiver and the duration of keeping the data packet in the buffer of a forwarding node, respectively.

Fig. 10 shows the average delay of the compared schemes. As the mobility increases, the average delays of all the schemes increase except for the Flooding scheme. This is because all of the schemes, except for the Flooding scheme, required the repair of the disconnected routes due to mobility. The Multiple Unicast scheme needs a longer period of time to reconnect the routing path as compared to the Traditional Multicast scheme. Hence, the average delay of the Multiple Unicast scheme is higher. Similarly, the average delay of the Traditional Multicast scheme is higher than that of the RCMP.

4.4. Throughput

To evaluate the multicast performance, we measured the throughput at the receivers. The "Throughput" refers to the total number of data packets per second received by all the members excluding the duplicated

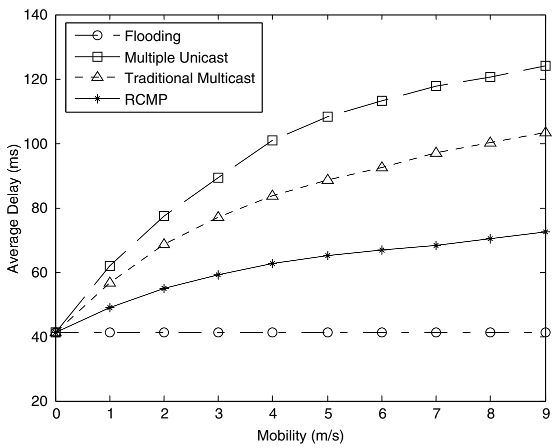

Fig. 10. Comparison of average delay versus mobility curves.

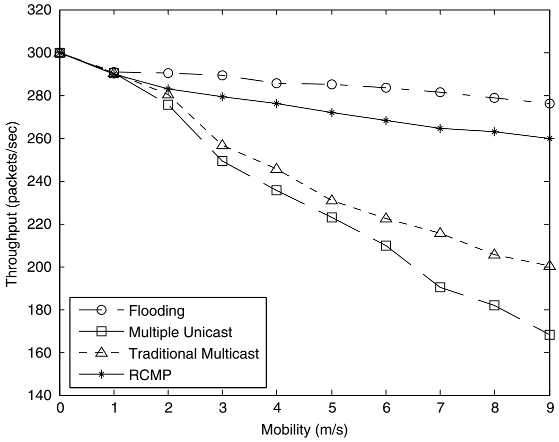

Fig. 11. Comparison of throughput versus mobility curves.

packets. In the network layer, the throughput performance is affected by the temporary loss of multicast packets due to mobility.

Fig. 11 compares the throughput of Flooding, Multiple Unicast, Traditional Multicast and RCMP. For a static network (without considering mobility), all schemes achieve the same throughput. However, the throughputs of all the schemes decrease with mobility. The Multiple Unicast and Traditional Multicast schemes yield lower throughputs due to the impacts of route reconstruction. Both of them need a period of time to repair the disconnected routes, especially in the case of the Multiple Unicast scheme. The Flooding scheme always keeps the highest throughput because it does not need to create specific routes to all the members. The Flooding scheme sends packets via all possible routes and thus, outperforms the other three schemes in terms of the throughput.

4.5. Multicast Service Cost

The ''Multicast Service Cost'' (also denoted by *MSC*) refers to the total number of transmitted data packets during the simulation time. A good multicast tree will deliver the multicast data packets to all the members at a lower service cost. The number of forwarding nodes and their forwarded data packets determine the cost of the multicast service. The energy and bandwidth consumptions increase with the number of forwarding nodes.

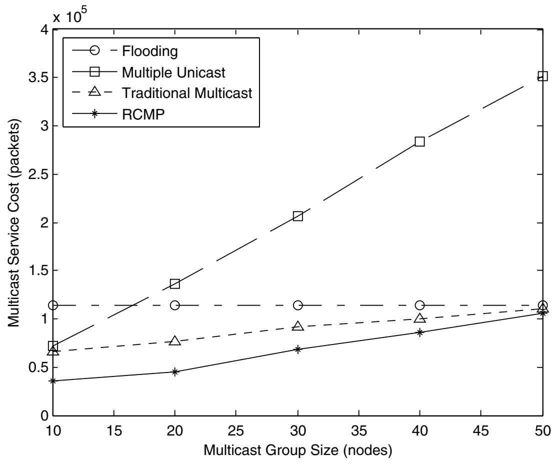

Fig. 12. Comparison of multicast service costs with variant in multicast group size.

The size of the multicast group was varied, ranging from 10 to 50, in order to investigate the relation between *MSC* and the number of members. Fig. 12 shows the comparison of the multicast service costs of the four schemes with variant sizes of multicast group. The Flooding scheme keeps a constant *MSC* even though the group size is varied because all the Bluetooth Stations are forwarding nodes regardless of the number of members. The Multiple Unicast scheme transmits individual data packets to each member. Hence, as the multicast group size increases, the *MSC* of the Multiple Unicast scheme increases significantly. The Traditional Multicast scheme and the proposed RCMP have lower *MSC* than the other two schemes because the shortest path factor was taken into consideration. Since the Traditional Multicast scheme does not consider the path sharing feature, RCMP has a lower *MSC* than the Traditional Multicast scheme.

5. Conclusions

The present paper proposed a novel Relative Coordinates-based Multicasting Protocol (RCMP) for Bluetooth Location Networks. Depending on the location of each mobile device, the Service Server and BS cooperate to construct an efficient multicast tree that shares the common link for reducing the number of forwarding nodes and lowering the power and bandwidth consumptions. The relative coordinate concept is used for each *BS* to calculate the relative direction of the destination BS and then to construct the shortest and the shared path for packets forwarding. Finally, the role switching and time leasing mechanisms are applied to efficiently construct the link connections from the *H-BS* to their responsible multicast members. The multicast tree constructed by the proposed RCMP has several good features which include the shortest route between each source–destination pair, a high degree of path sharing, fewer forwarding nodes, and low control overhead for route maintenance. Four multicast mechanisms namely, Flooding, Multiple Unicast Routing mechanism, Traditional Multicast Routing mechanism, and the proposed RCMP were compared in the simulation study. Simulation results show that the proposed RCMP scheme outperforms the other three mechanisms in terms of the average multicast degree, control overhead, and multicast service cost.

References

[1] P. Bhagwat, A. Segall, A Routing Vector Method (RVM) for routing in Bluetooth scatternets, in: Proceedings of IEEE International Workshop on Mobile Multimedia Communications, MoMuC 1999, 1999, pp. 375–379.

[2] Bluetooth SIG, Specification of the Bluetooth system – wireless connections made easy, Bluetooth Core Specification v1.1, 2001. Available from: <http://www.bluetooth.org/spec>.

[3] C.C. Chiang, M. Gerla, On-demand multicast in mobile wireless networks, in: Proceedings of the 6th International Conference on Network Protocols, ICNP 1998, 2998, pp. 262–270.

[4] C.C. Chiang, M. Gerla, L. Zhang, Forwarding Group Multicast Protocol (FGMP) for multihop, mobile wireless networks, Cluster Computing 1 (2) (1998) 187–196.

[5] C.Y. Chang, K.P. Shih, C.H. Tseng, C.F. Wang, A role switching agent for reducing packet lost phenomenon in bluetooth wireless networks, in: Proceedings of the 8th International Conference on Distributed Multimedia System, DMS 2002, 2002, pp. 473–479.

[6] C. Cordeiro, S. Abhyankar, D.P. Agrawal, A dynamic slot assignment scheme for slave-to-slave and multicast-like communication in bluetooth personal area networks, in: Proceedings of IEEE Global Telecommunications Conference, GLOBECOM 2003, 2003, pp. 4127–4132.

[7] F.J. González-Castaño, J. Garcia-Reinoso, Bluetooth location networks, in: Proceedings of IEEE Global Telecommunications Conference, GLOBECOM 2002, Madrid, 2002, pp. 223–237.

[8] Z.J. Haas, M.R. Pearlman, The Zone Routing Protocol (ZRP) for Ad-Hoc Networks, IETF Internet Draft, 1998.

[9] L. Ji, M.S. Corson, A lightweight adaptive multicast algorithm, in: Proceedings of IEEE Global Telecommunications Conference, GLOBECOM 1998, pp. 1036–1042.

[10] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad hoc networks, Mobile Computing (1996) 153–181.

[11] T. Kunz, E. Cheng, On-demand multicasting in ad-hoc networks: comparing AODV and ODMRP, in: Proceedings of International Conference on Distributed Computing Systems, 2002, pp. 453–454.

[12] A. Kumar, BlueHoc Manual, IBM India Research Lab. Available from: <http://oss.software.ibm.com/bluehoc>.

[13] Y.B. Ko, N.H. Vaidya, Location-aided routing (LAR) in mobile ad hoc networks, in: Proceedings of the 4th ACM/IEEE International Conference on Mobile Computing and Networking, MobiCom 1998, 1998, pp. 66–75.

[14] S.-J. Lee, M. Geria, C.-C. Chiang, On-demand multicast routing protocol, in: Proceedings of IEEE Wireless Communications and Networking Conference, WCNC 1999, 1999, pp. 1298–1302.

[15] T.-Y. Lin, Y.-C. Tseng, K.-M. Chang, Formation, routing, and maintenance protocols for the BlueRing scatternet of bluetooths, in: Proceedings of Hawaii International Conference on System Sciences, HICSS 2003, 2003.

[16] Y. Liu, M.J. Lee, T.N. Saadawi, A bluetooth scatternet-route structure for multihop ad hoc networks, IEEE Journal on Selected Areas in Communications 21 (2) (2003) 229–239.
[17] T.-Y. Lin, Y.-C. Tseng, K.-M. Chang, A new BlueRing scatternet topology for bluetooth with its formation, routing, and maintenance protocols, Wireless Communications and Mobile Computing 3 (44) (2003) 517–537.
[18] M. Medidi, A. Daptardar, A distributed algorithm for mesh scatternet formation in bluetooth networks, in: Proceedings of International Conference on Wireless Networks, 2004, pp. 295–301.
[19] Network Simulator, NS-2. Available from: <http://www.isi.edu/nsnam/ns>.
[20] C.E. Perkins, E.M. Royer, Ad-hoc on-demand distance vector routing, in: Proceedings of IEEE Workshop on Mobile Computing System and Applications, WMCSA 1999, 1999, pp. 90–100.
[21] C.E. Perkins, P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, in: Proceedings of the Conference on Communication Architectures, Protocol, and Applications, 1994, pp. 234–244.
[22] E.M. Royer, C.E. Perkins, Multicast operation of the ad-hoc on-demand distance vector routing protocol, in: Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking, 1999, pp. 207–218.
[23] Y.H. Wang, C.F. Chao, Dynamic backup routes routing protocol for mobile ad hoc networks, Journal of Information Sciences 176 (2006) 161–185.
[24] W. Zhang, H. Zhu, G. Cao, Improving bluetooth network performance through a time-slot leasing approach, in: Proceedings of IEEE Wireless Communication and Networking Conference, WCNC 2002, 2002, pp. 592–596.