



# Relay reduction and disjoint routes construction for scatternet over Bluetooth radio system

Gwo-Jong Yu<sup>a</sup>, Chih-Yung Chang<sup>b,\*</sup>, Kuei-Ping Shih<sup>b</sup>,  
Shih-Chieh Lee<sup>b</sup>

<sup>a</sup>*Department of Computer and Information Science, Aletheia University, Tamsui, Taipei, Taiwan*

<sup>b</sup>*Department of Computer Science and Information Engineering, Tamkang University, 151 Ying-Chuan Road, Tamsui, Taipei, Taiwan*

Received 3 August 2004; received in revised form 1 November 2005; accepted 16 November 2005

---

## Abstract

Bluetooth is a new technology for low-cost, low-power, and short-range wireless communication. By constructing a piconet, Bluetooth device establishes link and communicates with other device in a master–slave manner. Relay is a Bluetooth device that joins two or more piconets and forwards data from one piconet to another, providing multi-hop (or inter-piconet) communication services. In a Bluetooth scatternet, the number of relays and the degree of each relay are factors that significantly affect the performance of entire network. Unnecessary relays raise the difficulty of scheduling, leading to frequent packet loss. Relay switching among several piconets in turns also creates guard time overhead and increases the transmission delay. This study presents an effective protocol that can dynamically adjust the network topology by reducing the unnecessary relays. An efficient scatternet environment thus can be constructed with characteristics of connected, high bandwidth utilization and low maintenance cost. Additionally, a routing protocol is developed to reduce the path length and generate two disjoint routes for any pair of source and destination devices located in different piconets. Experimental results demonstrate that the proposed protocols perform well in terms of route length, bandwidth consumption, and transmission delay.

© 2005 Elsevier Ltd. All rights reserved.

**Keywords:** Bluetooth; Relay reduction; Route

---

---

\*Corresponding author. Tel.: +886 2 26215656x2749; fax: +886 2 26209749.

E-mail addresses: [yugj@wireless.mcs.au.edu.tw](mailto:yugj@wireless.mcs.au.edu.tw) (G.-J. Yu), [cychang@mail.tku.edu.tw](mailto:cychang@mail.tku.edu.tw), [cychang@cs.tku.edu.tw](mailto:cychang@cs.tku.edu.tw) (C.-Y. Chang), [kpshih@mail.tku.edu.tw](mailto:kpshih@mail.tku.edu.tw) (K.-P. Shih), [691190218@ms91.tku.edu.tw](mailto:691190218@ms91.tku.edu.tw) (S.-C. Lee).

## 1. Introduction

Bluetooth is a low-cost, low-power, short-range communication technology. To avoid co-channel interference during communication, the radio frequency (RF) module hops over 79 channels at a speed of 1600 times/s. The short packet and the fast hopping designs increase the communication reliability. A *piconet* comprises up to eight active Bluetooth devices, including one master and up to seven active slaves. The master of a piconet manages the schedule of data transmission of its slaves (<http://www.bluetooth.org>; Bhagwat, 2001).

A Bluetooth device can join two or more piconets simultaneously and, alternatively, act as slave in various piconets. The master in one piconet can also act as slave role in another piconet. However, a device cannot act as the master role in more than one piconet. Mobile device that joins two or more piconets is defined as *relay*. A relay is responsible to deliver messages from one piconet to another so that the resources or services will not be restricted. Packet transmission among piconets can be achieved by their common relays (Baatz et al.).

When intending to invite a Bluetooth device to be its slave, a master first switches to the inquiry state and obtains slave's Bluetooth address and clock information. The master then changes to the paging state, assigning a 3-bit *Active Member Address* (AM\_Addr) to the dedicated slave on the channel that is derived from information of slave's Bluetooth address and clock. The value of AM\_Addr ranges from 000 to 111. Each slave is assigned a different AM\_Addr for identification. If the AM\_Addr is exhausted, then the master cannot invite more devices as active members. Hence, the AM\_Addr is an important resource to help the master organize the piconet. A relay joining  $n$  piconet occupies  $n$  AM\_Addrs, consuming the AM\_Addr resource and raising the maintenance overhead. This work develops a relay reduction procedure to eliminate unnecessary relays so that the remaining scatternet is connected, and the number of occupied AM\_Addr and the maintenance overhead are minimized.

The performance of a connected scatternet highly relies on the factors including the number of relays and the degree of relay devices. Inter-piconet scheduling has been widely explored (Baatz et al., 2002; Kazantzidis and Gerla, 2002; Har-Shai et al., 2002; Har-Shai et al., 2004; Chang et al., 2004). Sniff and active modes are adopted in (Baatz et al., 2002) and (Kazantzidis and Gerla, 2002) to enable master to schedule its slaves adaptively. Rather than applying the sniff mode, some previous works (Har-Shai et al., 2002; Har-Shai et al., 2004) have utilized the hold mode to help the relay node switch among piconets. Regardless of whether the master schedule adopts the Sniff or Hold mode, experimental results of previous works demonstrate that the difficulty of master scheduling increases with the number of relays. Scatternet containing a large number of relays has various benefits, such as low probability of disconnection, short routing path and fast flooding, but also has drawbacks, including consumption of active member address, heavy packet generation in flooding and difficulty of synchronization among piconets. A relay with large degree will frequently switch among participating piconets, increasing the difficulty of scheduling and the probability of packet loss. Thus, retaining appropriate relays and removing others is critical to a connected scatternet's performance. One objective of this paper is to propose a relay reduction protocol for removing the unnecessary relays.

In addition to relay reduction problem, this study also investigates the route construction problem. Various scatternet formation algorithms have been proposed

(Zaruba et al., 2001; Petrioli and Basagni, 2002; Ramachandran et al., 2000; Salonidis et al., 2001; Tan et al., 2001). By randomly controlling the state of each Bluetooth device, a connected scatternet can be established in a very short time. Scatternets topology have been formed (Ramachandran et al., 2000) using Bernoulli trials to determine the role of each device in the Bluetooth discovery process. This method not only fulfills the scatternet connection requirement, but also minimizes the number of piconets in the scatternet. Protocols (Salonidis et al., 2001; Zaruba et al., 2001; Tan et al., 2001) have been proposed to generate connected scatternets topology and enable each master to manage its neighboring relays to improve communication efficiency. However, two Bluetooth devices that intend to communicate with each other may belong to different piconets, even if they are both located within the communication range. These devices cannot communicate directly, due to their difference in hopping sequences. Constructing an efficient routing path is an essential and important issue for providing devices with communication service over scatternet. A routing protocol is also proposed herein to reduce the path length and create disjoint paths.

In literature, a number of papers (Johnson and Maltz, 1996; Haas and Pearlman, 1998; Ko and Vaidya, 1998; Perkins and Bhagwat, 1994) have developed routing protocol for 802.11 Ad-Hoc networks based on on-demand route discovery philosophy. 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 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, otherwise, simply forwards the broadcast query. By considering all possible paths linking the source and destination, the source host can ascertain the shortest communication path.

Based on the flooding scheme, a Routing Vector Method (RVM) (Bhagwat and Segall, 1999) is proposed to construct a routing path for a pair of Bluetooth devices. Similar to on-demand routing protocols, RVM floods a search request over scatternet to find the destination device. The destination device then replies to this query and constructs the route. The route constructed by RVM highly depends on the network structure of scatternet. In 802.11-based Ad Hoc network, two devices can communicate directly with each other if their distance is smaller than the signal transmission range. However, the scatternet constructed by Bluetooth devices is different from Ad Hoc networks constructed by 802.11. Two Bluetooth devices belonging to different piconets cannot communicate with each other even if their distance is smaller than 10 m, since their hopping sequences are different. In this case, some relay nodes are required to serve for the route, forwarding the data packets from the source to the destination. Thus, constructing a route over the original scatternet is inefficient. Flooding creates long routes, causing significant power and bandwidth consumptions and transmission delay. This study presents a new routing protocol to shorten the routing path and create disjoint paths. Simulation results show that the proposed protocols are performance well in terms of routing length and data traffic overhead.

The remaining parts of this paper are organized as follows. Section 2 gives some examples to illustrate the relation between relay and piconet. A relay reduction protocol that dynamically adjusts the scatternet configuration is proposed in Section 3. Section 4 proposes a routing protocol to create disjoint and short paths. Section 5 investigates the performance improvements of the proposed protocols via experimental results. The conclusion and future work are given in Section 6.

## 2. Backgrounds and basic concepts

Each Bluetooth device in a piconet plays one of three roles, master, slave, or relay. A master device is responsible to schedule the data transmission in a piconet. A device that connects to multiple piconets plays a *relay* role, which can relay packets from one piconet to another.

In a piconet, master assigns a unique Active Member Address (AM\_Addr) to each active slave. Bluetooth technology adopts Time-Division-Duplex (TDD) frequency access technology to divide 1 s into 1600 time slots of 625  $\mu$ s. Under the control of master, the even time slot is reserved for master to transmit packet to its slave. Slave that receives packet from master in even slot has the right to transmit packet to master in the next odd slot (Pei et al., 2000; Ramachandran et al., 2000). A piconet contains at most eight devices. If the number of Bluetooth devices is more than eight, then at least two piconets are required to cover all devices.

According to the master's 48-bit BD\_Addr and clock information, a piconet generates a hopping sequence, which is adopted by all active devices to hop over the 79 channels. A relay that simultaneously participates in multiple piconets switches among piconets and synchronizes with the hopping sequence of each piconet to which it switches. For example, in Fig. 1, relay  $r_1$  simultaneously participates piconets  $P_1$  and  $P_2$ . The packet transmission from piconet  $P_1$  to  $P_2$  is illustrated as follows. In the first time slot (which is assumed to be an even time slot),  $r_1$  synchronizes with piconet  $P_1$  and receives a data packet from master  $m_1$ . Relay  $r_1$  then sends a packet to master  $m_1$  in the next odd slot. Relay  $r_1$  then enters a power saving mode (for example, sniff mode) in piconet  $P_1$  and switches to piconet  $P_2$  at the same time. Relay  $r_1$  then synchronizes with piconet  $P_2$ , waiting for receiving the data from master  $m_2$  in the even slots and then transmitting to master  $m_2$  the data received from master  $m_1$  in the next odd slots.

Although relay is necessary to construct a connected scatternet and forward data from one piconet to another, having too many relays has many disadvantages. One disadvantage is the consumption of AM\_Addr, causing master restricted to invite another device to be its active member. Unnecessary relay will also increase the maintenance overhead of entire scatternet, making synchronization among the participated piconets difficult and causing the packet loss phenomenon. This article proposes a relay reduction protocol for removing the unnecessary relays while maintaining a connected scatternet. An effective configuration of scatternet should have the following features:

- (1) *Connected*: Any Bluetooth device must be connected to a scatternet. Therefore, data on any Bluetooth device can be delivered to any other device.

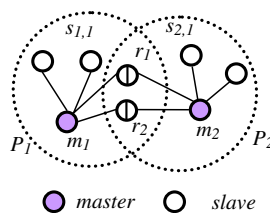


Fig. 1. Scatternet structure before executing relay reduction protocol.

- (2) *Small number of relays*: The advantages of a small number of relays in a connected scatternet include not only low maintenance cost, but also low AM\_Addr consumption. Because that one relay occupies an Active Member Address in each piconet, the relay reduction creates more opportunities for other devices to obtain the Active Member Address and enable them to participate in the piconet.

According to Fig. 1, piconets  $P_1$  and  $P_2$  have four active slaves each. Two relays,  $r_1$  and  $r_2$ , connect piconets  $P_1$  and  $P_2$ . If the link between  $r_2$  and  $m_1$  is broken, the set of relay nodes  $\{r_1, r_2\}$  is reduced to  $\{r_1\}$  and the released Active Member Address that is originally consumed by  $r_2$  can be reused by some other device that intends to participate in piconet  $P_1$ . As well as investigating the relay reduction problem, this paper proposes a routing protocol for constructing an efficient route over a relay-reduced scatternet. Since piconet has most seven active members, two Bluetooth devices that intend to communicate with each other may belong to different piconets. Relay nodes are thus required to serve the route, forwarding packet from one piconet to another. In addition to the relay reduction problem, the route construction problem is another important issue for providing devices with inter-piconet communication services. Based on the flooding scheme, a RVM (Bhagwat and Segall, 1999) is proposed to construct a routing path in a given scatternet. However, the route constructed by RVM is restricted by the original scatternet structure, making the route inefficient. As shown in Fig. 1, assume that  $s_{1,1}$  and  $s_{2,1}$  are source and destination devices, respectively. Applying RVM protocol will create a route *path1*:  $s_{1,1} \rightarrow m_1 \rightarrow r_1 \rightarrow m_2 \rightarrow s_{2,1}$ . However, if the distance between  $s_{1,1}$  and  $s_{2,1}$  is smaller than the signal transmission range, then they can construct a new piconet and create a new route *path2*:  $s_{1,1} \rightarrow s_{2,1}$  to replace *path1*, which consumes more network bandwidth and power of relay nodes than *path2*.

Therefore, in the proposed protocol, source device firstly sends a Route Search Packet to find the destination device. On receiving the Route Search Packet, the destination device replies with a Route Reply Packet, containing the destination's information including 48-bit BD\_Addr and clock offset value. The Route Reply Packet follows that the route the Route Search Packet passes through. Once the source device receives the Route Reply Packet, it can create a route to the destination device according to the information collected by Route Reply Packet, including the 48-bit BD\_Addr and clock offset of each relay. Compared with *path1*, *path2* has advantages including shorter path length and lower power and bandwidth consumptions. Complex conditions may exist between source and destination devices. For example, the distance between source and destination may exceed their signal transmission range. Proper relay should be involved to forward the data from source to destination. Reducing the number of involved relays reduces bandwidth and power consumption and alleviates the transmission delay. The proposed routing protocol will consider factors described above to construct an efficient route from source to destination.

This study considers two issues. First, a dynamic configuration approach is proposed to reduce the number of unnecessary relays and establish an efficient configuration for a given connected scatternet. Second, a routing protocol that creates disjoint routing paths with few hop count is investigated. Data transmission over scatternet can be more efficient. The following terms to explain the protocol details are defined.

**Definition.** Piconet ( $P_i$ )

A piconet consists of a master and at up to seven slaves. The term  $P_i = \{(m_i, s_{i,j}) \mid 1 \leq j \leq 7\}$  denotes a piconet, where  $m_i$  represents the master of piconet  $P_i$ , and  $s_{i,j}$  denotes one of the slaves dominated by  $m_i$ .

For example, Fig. 2 shows a scatternet containing three piconets  $P_1$ ,  $P_2$  and  $P_3$  where  $m_2$  is a master of piconet  $P_2$  and manages slave  $s_{2,1}$  and two relays.

**Definition.** Relay ( $r_c^k$ )

The set of relays participating in set of piconets  $c$  is denoted by  $r_c$ . According to the ascending order of Bluetooth address, the relay belonging to set  $c$  is numbered and denoted by  $r_c^k$  where  $1 \leq k \leq |r_c|$ . The term  $k$  can be omitted if its value is 1.

As described in Fig. 2,  $r_{c=\{2,3\}}$  represents the first relay that connects piconets  $P_2$  and  $P_3$ . Similarly,  $r_{c=\{1,2,3\}}$  represents the first relay that connects piconets  $P_1$ ,  $P_2$  and  $P_3$ .

**Definition.** Neighboring relation

Piconets  $P_i$  and  $P_j$  are said to be neighboring if  $P_i \cap P_j \neq \emptyset$ . Masters  $m_i$  and  $m_j$  are neighboring if their Piconets are neighboring.

The proposed *relay reduction protocol* maintains the connection property of any pair of piconets sharing common relays and removes unnecessary relays. Thus the consumption of Active Member Address can be reduced. The cost for maintaining relays and the probability of packet loss can also be reduced. The proposed relay reduction protocol has the following two properties.

**Connectivity.** For each  $P_i, \exists P_j$  such that  $P_i \cap P_j \neq \emptyset$ , where  $P_i$  and  $P_j \in$  scatternet.

**Least Relay Property.** A configuration with smallest number of relays in a scatternet can be obtained.

The aim of the relay reduction protocol is to reduce the maximum number of relays in a distributed manner while maintaining connectivity. To prevent the scatternet from disconnection while minimizing the number of reserve relays, the investigated protocol preserves the relay with larger degree and removes relays with a smaller degree, enabling the resources of Active Member Address can be explored. For example, as depicted in

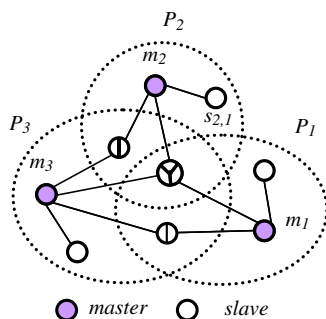


Fig. 2. A simple scatternet environment, where  $r_{c=\{1,3\}}$ ,  $r_{c=\{2,3\}}$ ,  $r_{c=\{1,2,3\}}$  are relays in the scatternet.

Fig. 2, to construct a configuration with the smallest possible number of relays, the protocol first preserves the relay with degree 3, such as  $r_{c=\{1,2,3\}}$ , rather than those relays with degree 2, such as  $r_{c=\{1,3\}}$  and  $r_{c=\{2,3\}}$ . Therefore, relays  $r_{c=\{1,3\}}$  and  $r_{c=\{2,3\}}$  change their role to a pure slave and only serve for  $m_1$  and  $m_2$ , respectively. As regard to  $P_2$  and  $P_3$ ,  $r_{c=\{1,3\}}$  and  $r_{c=\{2,3\}}$  will no longer play the role of relay. One active member address is therefore saved for another device to participate piconets  $P_2$  and  $P_3$ . To apply the proposed reduction protocol, a connection table (CT), should be maintained in each relay as follows.

**Definition.** Connection Table (CT)

Each relay  $r$  maintains a CT with *col* columns and *row* rows. The first column of CT records the relays connected by masters connected to  $r$ . The first row of CT contains masters connected by relays listed in the first column. The values in each entry is determined by

$$CT(r_c^k, m_j) = \begin{cases} 1, & r_c^k \text{ connects } m_j, \\ null, & r_c^k \text{ does not connect } m_j \end{cases}$$

for each  $r_c^k$  in first column and each  $m_j$  in first row.

Table 1 shows the CT maintained by relay  $r_{c=\{1,3\}}$  in Fig. 3. Because relay  $r_{c=\{1,3\}}$  connects to  $m_1$  and  $m_3$ , the first column of CT contains relays connected by  $m_1$  or  $m_3$ , i.e.

Table 1  
The CT stored in relay  $r_{c=\{1,3\}}^1$

| $r_{c=\{1,3\}}^1$   | $m_1$ | $m_2$ | $m_3$ | $m_4$ | $m_5$ |
|---------------------|-------|-------|-------|-------|-------|
| $r_{c=\{1,2\}}^1$   | 1     | 1     |       |       |       |
| $r_{c=\{1,5\}}^1$   | 1     |       |       |       | 1     |
| $r_{c=\{1,3,5\}}^1$ | 1     |       | 1     |       | 1     |
| $r_{c=\{1,3\}}^1$   | 1     |       | 1     |       |       |
| $r_{c=\{1,3,4\}}^1$ | 1     |       | 1     | 1     |       |
| $r_{c=\{1,4\}}^1$   | 1     |       |       | 1     |       |
| $r_{c=\{3,5\}}^1$   |       |       | 1     |       | 1     |

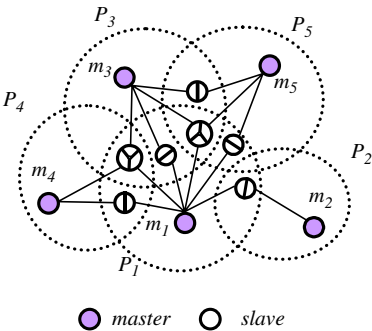


Fig. 3. A scatternet structure before executing the relay reduction protocol.

$r_{c=\{1,2\}}$ ,  $r_{c=\{1,5\}}$ ,  $r_{c=\{1,3,5\}}$ ,  $r_{c=\{1,3\}}$ ,  $r_{c=\{1,3,4\}}$ ,  $r_{c=\{1,4\}}$  and  $r_{c=\{3,5\}}$ . The first row of *CT* records the masters  $m_1$ ,  $m_2$ ,  $m_3$ ,  $m_4$  and  $m_5$  which are served by these relays. The values in row  $r_{c=\{1,3,5\}}$  and columns  $m_1$ ,  $m_3$  and  $m_5$  are 1, indicating that relay  $r_{c=\{1,3,5\}}$  serves for masters  $m_1$ ,  $m_3$  and  $m_5$ . Consider another example in the intersections of row  $r_{c=\{1,3\}}$  and columns  $m_1$  and  $m_3$ . The value 1 implies that relay  $r_{c=\{1,3\}}$  serves for  $m_1$  and  $m_3$ . Each relay can determine whether or not it should play a relay role by checking *CT*, so that the scatternet has the smallest number of relays while remaining connected.

### 3. Dynamic relay reduction protocol

This section proposes a distributed relay reduction protocol to remove the unnecessary relays from a given scatternet. Each relay initially transmits a *Relay Information Collection (RIC)* packet to the masters it connects, requesting each master to collect information about all relays belonging to its piconet. The relay's *CT* can thus be constructed by collecting the reply information of those masters served by this relay. For example, relay  $r_{c=\{1,3\}}$  sends a *RIC* packet to its masters  $m_1$  and  $m_3$  which then forward it to their relays. For instance,  $m_1$  forwards the *RIC* packet to relays  $r_{c=\{1,2\}}$ ,  $r_{c=\{1,5\}}$ ,  $r_{c=\{1,3,5\}}$ ,  $r_{c=\{1,3\}}$ ,  $r_{c=\{1,3,4\}}$  and  $r_{c=\{1,4\}}$ , and each relay replies to master  $m_1$  with the information of all masters that it serves. Thus, on receiving the packet reply from  $m_1$ , relay  $r_{c=\{1,3\}}$  uses the information to construct its *CT*, as shown in Table 1, where the value 1 denotes a link between the corresponding relay and master, while a null value indicate that the relay and the master are not connected. After the *CT* is constructed, each relay examines the *CT* and determines whether to change the role from a relay to a pure slave.

As shown in Table 1,  $r_{c=\{1,3\}}$  connects to masters  $m_1$  and  $m_3$ . According to the *CT*, relay  $r_{c=\{1,3\}}$  checks whether any other relay has a larger degree, and also connects to  $m_1$  and  $m_3$ . If such relay exists, the role change of relay  $r_{c=\{1,3\}}$  from relay to pure slave does not cause the disconnection of  $P_1$  and  $P_3$  and relay  $r_{c=\{1,3\}}$  should determine the piconet in which to participate as a pure slave according to balance consideration. All relays that simultaneously execute the same relay reduction procedure will minimize the number of relays and guarantee that the restructured scatternet is connected.

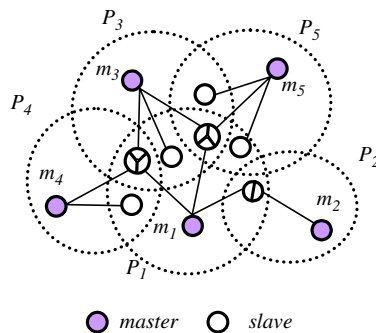


Fig. 4. The resultant scatternet after executing the relay reduction protocol.

Fig. 4 shows the scatternet resulting from applying the proposed relay reduction protocol on the example in Fig. 3. The roles of relays  $r_{c=\{1,5\}}$ ,  $r_{c=\{1,3\}}$ ,  $r_{c=\{1,4\}}$  and  $r_{c=\{3,5\}}$  have been changed to pure slave in the new scatternet. Thus, four Active Member Addresses are saved, while the scatternet maintains connected. Reducing the number of relays not only decreases the guard time cost for relay switching among piconets, but also reduces the probability of packet loss and improves the transmission efficiency. The relay nodes update the *CT* in the following two situations. First, the relay nodes use the received polling messages from masters to check the link validity and update the *CT*. Second, the relay node updates the *CT* table when routes create new links. Therefore, the relay nodes update or refresh their *CTs* dynamically based on the two cases mentioned above.

The proposed relay reduction protocol is given below.

### 3.1. Relay reduction protocol

*Step 1:* In a Scatternet, each relay  $r$  sends the *RIC* message to each of its masters.

*Step 2:* On receiving the *RIC* message, master  $m_i$  collects information of relays belonging to piconet  $P_i$  and the connection information of neighboring masters.

*Step 3:* Once the relay  $r$  receives the information provided by the masters the relay connects to, each relay creates a *CT*, where

$$CT(r_c^k, m_j) = \begin{cases} 1, & r_c^k \text{ connects } m_j \\ \text{null}, & r_c^k \text{ does not connect } m_j \end{cases}$$

for each  $r_c^k$  in first column and each  $m_j$  in first row.

*Step 4:* Let *rows* denote the number of rows in *CT*, *my\_row* denote the row currently handled by  $r$ , and  $e_{i,j}$  denote the value of the  $j$ th entry of row  $i$ . A row  $x$  is said to be dominated by row  $y$  if the following two conditions are satisfied:

(1) For each entry  $e_{x,j} = 1$  in row  $x$ , the corresponding entry  $e_{y,j} = 1$  in row  $y$ .

(2)  $\sum_{1 \leq j \leq \text{col}} e_{x,j} < \sum_{1 \leq j \leq \text{col}} e_{y,j}$ .

Relay  $r$  executes the following procedures according to the *CT*:

for  $i = 1$  to *rows*,

if (*my\_row* is dominated by row  $i$ ).

{

1. Relay  $r$  changes its role from relay to pure slave.

2. Let  $R$  denote the set of piconets in which  $r$  participates and  $P_{\min} = \{P_j | \min |P_j| \text{ for } \forall P_j \in R\}$ , where  $|P_j|$  denotes the number of devices in piconet  $P_j$ . Relay  $r$  plays the slave role in Piconet  $P_{\min}$  and breaks all connections between  $r$  and all masters in piconets  $R - P_{\min}$ .

}

Applying the proposed relay reduction protocol, a scatternet can be dynamically adjusted to an ideal scatternet where all unnecessary relays are removed in a distributed manner. The next section describes a new routing protocol, which constructs a routing disjoint path, which minimizes the number of hops and has QoS features.

#### 4. The routing protocol

Section 3 describes how to reduce the number of relays. This section proposes a routing protocol, based on the reduced scatternet, which constructs disjoint paths with smallest number of hop, and also has QoS services. Previous related research is first discussed. The new routing protocol is then proposed.

Two Bluetooth devices that intend to communicate with each other but belong to different piconets require a routing path. Since every piconet contains at most seven slaves, two devices that intend to communicate with each other may belong to different piconets. For example, in Fig. 1, even if the distance between  $s_{1,1}$  and  $s_{2,1}$  is smaller than the Bluetooth communicative range, they cannot communicate directly with each other due to their different hopping sequences. The following describes two main reasons why slaves  $s_{1,1}$  and  $s_{2,1}$  belonging to different piconets:

- (1) Assume that slave  $s_{1,1}$  is connected to  $m_1$ . During executing the inquiry and inquiry scan operations,  $s_{2,1}$  connects to  $m_2$ , so devices  $s_{1,1}$  and  $s_{2,1}$  fall in different piconets.
- (2) Piconet of  $s_{1,1}$  has already collected seven slaves, so  $s_{2,1}$  has to join another piconet.

Assume that devices  $s$  and  $d$  are source and destination, respectively. An efficient route should be constructed if they belong to different piconets. In the RVM (Bhagwat and Segall, 1999) routing protocol, presented by Bhagwat, the source device first adopts flooding mechanism, broadcasting the route search packet over the scatternet. On receiving the route search packet, the destination device generates a reply packet and sends it to the source following the route through which the search packet has passed. The first routing path is thus constructed. As required, a second route that contains the common relay of the first route as few as possible can be constructed.

The routing path constructed by RVM technique may introduce large transmission latency, because the created route has a large hop count. Furthermore, a long routing path consumes bandwidth and power. For example, Fig. 5(a) shows the route constructed by the RVM routing protocol. Devices  $s$  and  $d$  are source and destination, respectively. The constructed route of devices  $s$  and  $d$  has eight hops, yet the physical distance between  $s$  and  $d$  may be less than two hops as shown in Fig. 5(b). This section proposes a new routing protocol to construct an efficient route in a Bluetooth scatternet.

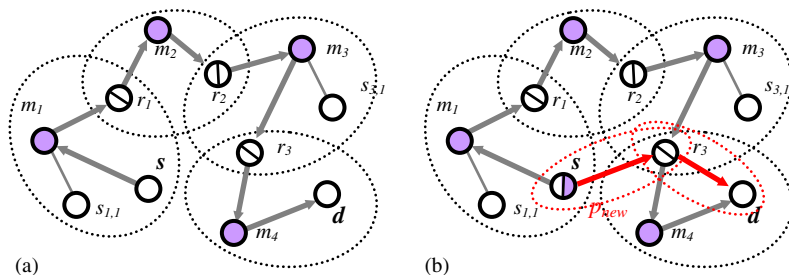


Fig. 5. Path reduction procedure: (a) route before executing path reduction and (b) route after executing path reduction.

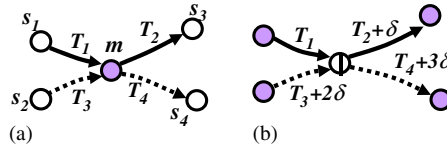


Fig. 6. Two routes share the same Bluetooth node: (a) two routes share a master and (b) two routes share a relay.

Bluetooth divides 1 s of time into 1600 time slots, with each time slot  $625 \mu\text{s}$ . The time slot is the basic time unit in a piconet and the clock of each slave devices should be synchronized with its master. Master can transmit data to one of its slaves in an even time slot. Slave that receives data from master in the even slots has the right to transmit data or null packets to master in the next odd slot. If two routing paths passing through the same piconet, then two relays forward their data to the master. However, only one relay can communicate with master at a time. In Fig. 6(a), the master  $m$  connects four slaves in a piconet. Master  $m$  receives DH5 packet  $a$  from slave  $s_1$  in odd time slot  $T_1$ , and forwards the received packet to slave  $s_3$  in time slot  $T_2 = T_1 + 5$ . In time slot  $T_3$ ,  $s_2$  is scheduled to transmit another DH5 packet  $b$  to master  $m$ , and this packet will be transmitted from master  $m$  to slave  $s_4$  in time slot  $T_4 = T_3 + 5$  (Das et al., 2001; Kalia et al., 1999). Two routing paths sharing the same master cause a bottleneck between the two routes in master. Consequently, the throughput of two routing paths is the same as that of a single routing path. If the shared host is a relay, as shown in Fig. 6(b), then the transmission time of two packets requires  $3\delta$  more, provided that  $\delta$  is the time required for a relay switching from one piconet to another (Kalia et al., 2000). Therefore, if two routing paths share master or relay, then the throughput may be worse than single routing path. By contrast, if two disjoint routing paths transmit packets independently, then the throughput is twice that of a single path. Multiple routes sharing a host node or link leads a poor throughput and should be avoided when designing the routing protocol. A procedure with a path reduction feature is proposed below to create short and disjoint paths.

#### 4.1. Path reduction procedure

This section proposes a routing protocol with features of disjoint and shortest routing path. The example in Fig. 5 is utilized to illustrate the method and properties of the proposed protocol. The complete protocol is then presented.

As shown in Fig. 5(a), when source host  $s$  intends to establish a route to destination host  $d$ , it broadcasts a Route Search Packet based by flooding. The destination host executes the following Path Reduction Procedure as soon as receives the Route Search Packets. First, the source host initiates and broadcasts a route search packet. On receiving the route search packet, host  $d$  selects a route and replies with the Route Reply Packet from destination to source along the reversal direction of the route and appends its Bluetooth Address and Clock offset value between  $d$  and  $m_4$  to the Route Reply Packet, which is then transmitted to relay  $r_3$  via master  $m_4$ . Meanwhile,  $d$  will in turn enter *Hold* mode in the piconet managed by master  $m_4$ , and *Page Scan* mode to construct a new piconet to reduce the route length. When  $r_3$  receives Route Reply Packet, it records the Bluetooth Address and clock offset values of  $d$  in its cache and appends its Bluetooth Address and clock offset value between  $r_3$  and  $m_4$  to Route Reply Packet. Relay  $r_3$  then transmits Route Reply

Packet to master  $m_3$ . Hereafter, relay  $r_3$  in turn enters *Hold* mode in its original piconet and *Page Scan* mode to construct a new piconet to shorten the routing length. Relays  $r_2$  and  $r_1$  will execute the similar procedures as relay  $r_3$  when they receive the Route Reply Packet. Therefore, source host will finally receive the Bluetooth address and clock offset information of devices  $d$ ,  $r_3$ ,  $r_2$  and  $r_1$ .

Source host  $s$  then enters *Hold* mode in the original piconet and tries to play the role of master by entering page mode. Host  $s$  constructs a new piconet with relays  $r_1$ ,  $r_2$ ,  $r_3$ , or  $d$  by examines Bluetooth addresses of hosts capable of constructing new piconets. This example assumed that  $s$  is in *Page* mode and can construct a connection with devices  $r_1$ ,  $r_2$  or  $r_3$ . The connection between  $s$  and  $d$  cannot be constructed since their distance is out of the Bluetooth Radio Range. Source  $s$  then tries to connect with one device, in order of  $r_3$ ,  $r_2$ ,  $r_1$ . Consequently,  $s$  constructs a new piconet  $P_{new}$  with  $r_3$ , as shown in Fig. 5(b).

After the connection is established between  $s$  and  $r_3$ , those relays whose sequence order of Bluetooth address in Route Reply packet less than  $r_3$  are asked to stop executing the *Page Scan* operation and quit the routing service. The route is shortened due to the abandonment of  $r_1$  and  $r_2$ . Then relay  $r_3$  repeatedly applies this procedure to construct a link from  $r_3$  to the destination. Consequently, relay  $r_3$  enters *Page Mode* and tries to construct a new piconet with host  $d$  that currently stays on *Page Scan* mode. A new route  $s \rightarrow r_3 \rightarrow d$  results from these procedures. The new route has a hop count of 2, which is shorter than 7 as found in the original route. The Path Reduction Procedure only requires the Bluetooth Address and Clock information to construct the new link. The time-consuming operations such as Inquiry and Inquiry Scan are omitted. Therefore, the path reduction procedure can be completed very quickly.

Another example in Fig. 3 illustrates that the proposed *Path Reduction Procedure* still constructs a shorter path in a relay reduced topology. Fig. 4 depicts the topology after applying the relay reduction protocol. Some relay nodes like  $r_{c=\{1,5\}}$ ,  $r_{c=\{1,3\}}$ ,  $r_{c=\{1,4\}}$  and  $r_{c=\{3,5\}}$  are converted to pure slave roles. When  $r_{c=\{1,4\}}$  intends to communicate with  $m_1$ , the *Path Reduction Procedure* in the proposed routing protocol shortens the path from  $r_c = \{1,4\} \rightarrow m_4 \rightarrow r_c = \{1,4,3\} \rightarrow m_1$  to  $r_c = \{1,4\} \rightarrow m_1$ . The *Path Reduction Procedure* removes the redundant forwarding nodes of a routing path by connecting two nodes (i.e.  $r_c$  and  $m_1$  in this example) if they are within the communication range. Therefore,  $r_{c=\{1,4\}}$  becomes a master/slave node and connects directly to  $m_1$ . Consequently, the route has a length of 2, which is the same as the length of the route without applying relay reduction protocol. Therefore, executing the relay reduction protocol does not increase the routing length.

This section presents a routing protocol to construct the shortest possible routing path, and thus use bandwidth efficiently. The next section proposes a protocol for creating disjoint routes to either speed up transmission or provide backup routes.

#### 4.2. Creating the disjoint routes

Alternative routes may be created for two major reasons. One reason is for backup. As soon as some link of the first route is broken, the second route can instead transmit data from source to destination. The other reason is to speedup the throughput between two nodes.

An enhanced routing protocol is proposed to construct disjoint routes, both to speed up data transmission and to create backup routes. Consider Fig. 5(b) as an example. After constructing the route by applying route reduction procedure, if source host requests to

create multiple routes, the protocol proposed in this section creates a short disjoint route with a limited control packet overhead. Since the information of the 48-bits Bluetooth address and clock offset of all relays and destination have been collected in the source host, new piconets can be constructed to establish disjoint routes by using the unused relays ( $r_1$ ,  $r_2$ ), thus avoiding large number of packets due to flooding.

When source host  $s$  intends to construct a disjoint path,  $r_1$  and  $r_2$  stay on Page Scan mode. During the second route construction phase,  $r_3$  abandons Page Scan mode since it has already participated in the first route. The destination host  $d$  should stay on Page Scan mode to construct another route. The disjoint route is constructed by a similar approach to the construction of the first route.

To speed up the throughput of a route, two disjoint paths could be utilized to transmit different data at the same time. The precise difference of hop count for the disjoint paths should be controlled so that packets transmitted by these two routes can be received at their destination continuously without any collision. In the two disjoint routes, hop count is the major factor that determines whether packets can be received continuously in destination. To avoid the collision of schedules for packets transmission in destination  $d$ , the length difference of two disjoint routes should be maintained at  $2k$  hops, where  $k$  is an integer.

For example, as shown in Fig. 7, two disjoint routes, Route1:  $s \rightarrow r_3 \rightarrow d$  and Route2:  $s \rightarrow r_1 \rightarrow r_2 \rightarrow s_{3,1} \rightarrow d$ , are constructed. The hop counts of Route1 and Route2 are 2 and 4, respectively. Their hop count difference satisfies the constraint  $2k$ , where  $k = 1$ . Packets could therefore be transmitted in parallel by these two routes. At time  $t_1$ , source host  $s$  transmits packet 1 on link  $s \rightarrow r_3$  of Route1. At time  $t_2$ , source host  $s$  transmits packet 2 on link  $s \rightarrow r_1$  of Route2 and  $r_3$  simultaneously transmits packet 1 to its destination via link  $r_3 \rightarrow d$ . At time  $t_3$ , packet 3 is transmitted on link  $s \rightarrow r_3$  and packet 2 is transmitted on link  $r_1 \rightarrow r_2$  of Route 2. At time  $t_4$ , the destination receives packet 3 from link  $r_3 \rightarrow d$  of Route1 while packet 2 is delivered on link  $r_2 \rightarrow s_{3,1}$  of Route2. Thus, the proposed protocol guarantees that after four time steps, the max hop count of routes, packets can be delivered to destination from two routes in different time steps continuously. The source data thus can be transmitted on these two routes in parallel, increasing the throughput.

In addition to the collision avoidance problem, the following problem should also be investigated. Assume that the first route is  $s \rightarrow r_3 \rightarrow d$  as shown in Fig. 8. Let  $r_2$  represent the current relay that executes the second-route construction procedure and there is no relay between  $r_2$  and  $d$  available to construct the disjoint route. Relay  $r_2$  cannot construct a new piconet with destination since the required distance is outside the communication range.

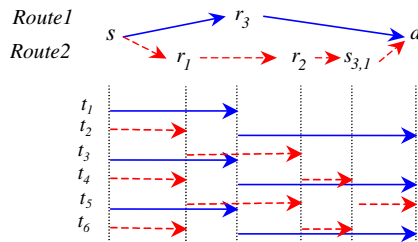


Fig. 7. Scheduling of parallel data transmission by two disjoint routes.

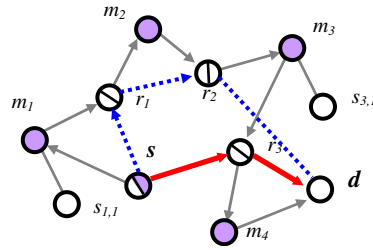


Fig. 8. Problem encountered during the construction of second routes. Relay  $r_2$  is unable to construct a link to  $d$ .

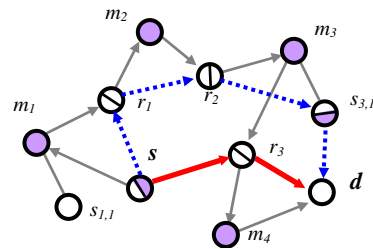


Fig. 9. Construction of two disjoint routes.

To construct the second route,  $r_2$  should request  $m_3$  to ask its slaves with a low traffic load to enter Page Scan mode. The selected slave can thus act as the relay of  $r_2$  and the destination host. As shown in Fig. 9,  $m_3$  selects slave  $s_{3,1}$  to enter Page Scan mode and transfers the Bluetooth Address and Clock offset information of  $s_{3,1}$  to  $r_2$  such that  $r_2$  and  $s_{3,1}$  can construct a new link. Therefore, the following two disjoint routes are formed:

$$\text{Route1} : s \rightarrow r_3 \rightarrow d, \quad \text{Route2} : s \rightarrow r_1 \rightarrow r_2 \rightarrow s_{3,1} \rightarrow d.$$

With such a mechanism, the probability of successfully constructing the second disjoint path is increased and the difference of hop counts is compensated to  $nk$  exactly.

#### 4.3. The routing protocol

This section formally describes the disjoint routing protocol. Given a pair of source and destination devices, the proposed protocol constructs two disjoint routes that can transmit data from source to destination in parallel. The routes constructed by the proposed protocol are shorter than routes constructed by flooding. Objectives of high throughput, low bandwidth and power consumptions, and low transmission delay are thus achieved. The disjoint routing protocol consists of three phases. Phase I constructs the first route and prepares route reduction information. Phase II performs the route reduction to minimize the bandwidth consumption and transmission delay of the constructed route. Phase III constructs the disjoint route either to speed up data transmission or as a backup route. The protocol is described below.

#### 4.3.1. Phase I: construction of first route

1. When the source host  $s$  intends to create a communication path to destination device  $d$ , it broadcasts the Route Search Packet containing the Source\_Host\_ID and Destination\_Host\_ID.
2. On receiving the Route Search Packet, destination host  $d$  creates and transmits a Route Reply Packet to source host  $s$  in reverse order of the path that the Route Search Packet flooded from  $s$  to  $d$ . Here, the path from source  $s$  to  $d$  is assumed to be  $s [\rightarrow m_0] \rightarrow r_0 \rightarrow m_1 \dots \rightarrow r_{n-1} [\rightarrow m_n] \rightarrow d$ , where  $m_0$  indicates that the second host on the route is either a master or a relay.
3. Once relay  $r_i$  receives the Route Reply Packet, it appends its Bluetooth Address and Clock offset to the packet between its own and its master's clock, and then transfers the packet to its master  $m_i$ . Hereafter, relay  $r_i$  changes its state from active to hold mode in the original piconet, enabling it to enter page scan mode to wait for the source host's paging.

Source host  $s$  begins the following Phase II operation as soon as it receives the Route Reply Packet.

#### 4.3.2. Phase II: path reduction procedure

Let  $U$  denote the set of devices on the reduced route. Set  $U$  is initialized as an empty set. Let  $\dot{R} = \{r_0, r_1, \dots, r_{n-1}\}$  denote the set of relay nodes passed through by flooding the Route Search Packet.

1. If source node plays the role of Master, then as soon as it receives the Route Reply Packet, it enters Page state and tries to invite the destination device or relay in set  $\dot{R}$  to be a member of the current piconet. If source node is not a Master, then it is either a slave or relay. In this case, the source host changes to Hold mode in the original piconet, and enters Page state to create a new piconet with destination or relays in set  $\dot{R}$ .
2. Source host takes  $\Delta t$  to connect with the destination or the relay closest to destination on the route (relay  $r_i$  with max value of  $i$ ).
3. If the source host  $s$  successfully connects to the destination host  $d$ , then a reduced new path  $s \rightarrow d$  with single hop is created. Set  $U$  is updated by  $U = \{s, d\}$ .
4. If host  $s$  failed to connect to destination  $d$ , then it connects to the relay  $r_{max}$ , which is the relay with highest index to which host  $s$  is capable connected in set  $\dot{R}$ . Source host  $s$  updates sets  $U$  and  $\dot{R}$  by

$$U = \{s, r_{max}\}, \quad \dot{R} = \dot{R} - \{r_{max}\}.$$

5. Steps 3 and 4 cannot work if destination  $d$  and all relays in set  $\dot{R}$  are more than 10 m from the source  $s$ . In this case, master  $m_0$  asks its idle slaves to enter the Page state to try to connect to destination  $d$  or a relay in set  $\dot{R}$ . If an idle slave, given by  $s_{0,j}$ , connects to destination  $d$ , it sends its Bluetooth address and clock offset information to source  $s$ , which can then construct a new piconet with slave  $s_{0,j}$ . Source host  $s$  then updates set  $U$  by  $U = \{s, s_{0,j}, d\}$ , thus successfully constructing a shortened path  $s \rightarrow s_{0,j} \rightarrow d$ . In case that all idle slaves cannot connect to destination  $d$ , they will try to connect to any

possible relay  $r_{max}$  in set  $\hat{R}$ . If the connection is successful, slave  $s_{0,j}$  will transmit its Bluetooth address and clock offset information to source  $s$ , which can then construct a new piconet with slave  $s_{0,j}$ . Source host  $s$  then updates set  $U$  to  $U = \{s, s_{0,j}, r_{max}\}$  and a reduced subpath  $s \rightarrow s_{0,j} \rightarrow r_{max}$  is thus constructed. Acting as a source host  $s$ , relay  $r_{max}$  once again performs the path reduction procedure enabling it to construct a reduced path to destination  $d$ . The reduced path is thus constructed.

6. Source host  $s$  cannot perform the path reduction procedure if step 5 fails. The source host  $s$  asks  $r_0$  to play its role and to perform the path reduction procedure so that the length of subpath from  $r_0$  to destination  $d$  could be reduced. In this case, source host  $s$  removes from set  $\hat{R}$  and updates set  $U$  by setting  $U = \{s, m_0, r_0\}$ .

#### 4.3.3. Phase III: construction of the disjoint route

Set  $\hat{R} = \hat{R} - U$ . Since set  $U$  collects all devices on the reduced path, set  $\hat{R}$  collects all the devices that appear on the route created by flooding and are not utilized on the reduced path. In Phase III, we utilize the devices in set  $\hat{R}$  for constructing a disjoint route. Let  $U' = \{\emptyset\}$ . Set  $U'$  is used for collecting all the devices on the disjoint route.

1. Adopt sets  $\hat{R}$  and  $U'$  to execute the Phase II procedure to construct the disjoint route and to collect all devices on the disjoint route.
2. To ensure that packets transmitted via the two routes arrive at destination in different time slots, the difference in hop count of these two routes should be  $2k$ , where  $k$  is an integer. That is,  $U$  and  $U'$  should satisfy the following constraint:

$$|U| - |U'| = 2k.$$

If the above constraint is not satisfied, then  $s$  inserts an additional device  $s_{0,j}$  on the second route, as described in Step 5 of Phase II.

This section presents a routing protocol that constructs disjoint routes to enhance the traffic throughput or prevent the constructed route from breaking. The two disjoint routes created by the proposed protocol are much shorter than routes constructed by RVM. The performance of the proposed protocol is examined in the next section.

## 5. Performance study

This section presents the performance investigation of the proposed protocol, in terms of the number of piconets, number of relays, path length, probability of disjoint paths and the number of control packets. The simulation environment is described as follows. The Constant Bit Rate (CBR) Model was adopted to generate the traffic load for each route in the performance simulation, and the traffic arrival rate was 100 kbps. The space size was set to  $20 \times 20$ ,  $40 \times 40$ , or  $80 \times 80$ , while the radio transmission range of a Bluetooth device was set at a constant 10 units. The number of devices varied from 100 to 140, and their locations were randomly determined.

Figs. 10 and 11 are snapshots of relay reduction. Fig. 10 shows the connection status before relay reduction, and Fig. 11 shows the scatternet after relay reduction. Fig. 12 shows the efficiency of relay reduction protocol, revealing that the number of relays can be reduced regardless of the change of environment. Since the relay reduction protocol can

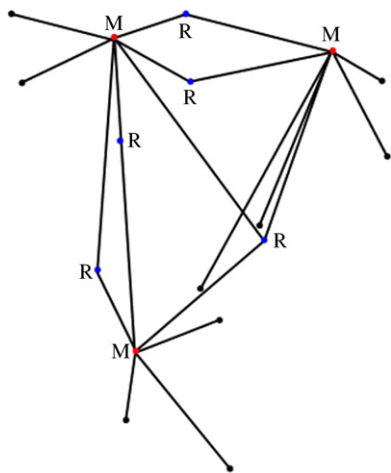


Fig. 10. A snapshot before relay reduction.

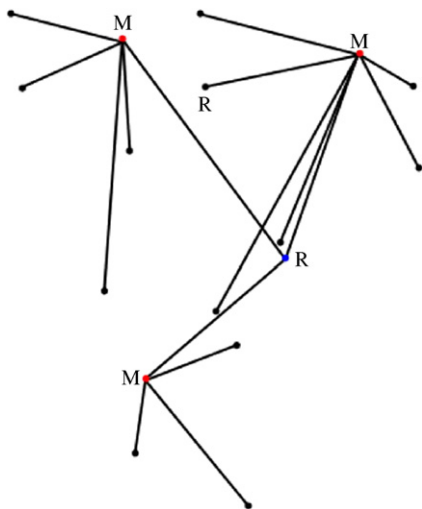


Fig. 11. A snapshot after relay reduction.

reduce guard time for relay switching among the participated piconets and reduce the probability of packet loss, the proposed protocol helps to improve the performance.

Fig. 13 is a snapshot of the execution result. The number of hosts in the scatternet was kept small to observe and compare the behavior of RVM and the proposed protocol. In Fig. 13, the labels ‘M’ and ‘R’ denote ‘Master’ and ‘Relay’ nodes, respectively. The scatternet in Fig. 13 consists of three piconets. A dashed line connecting two nodes denotes a link between a slave and a master. The source and destination hosts, marked by thick and big nodes, are selected from the different piconets. The bold lines denote the routing path where the marked value denotes the link order on the path. As shown in Fig. 13, RVM

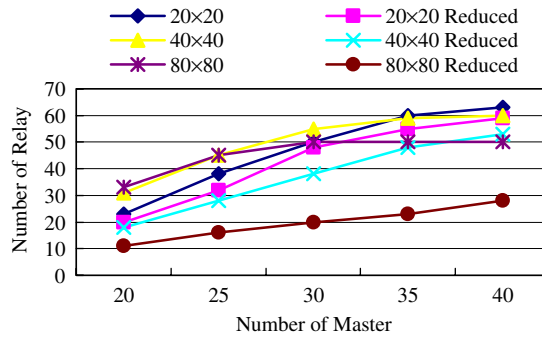


Fig. 12. Comparison of the number relay under any size.

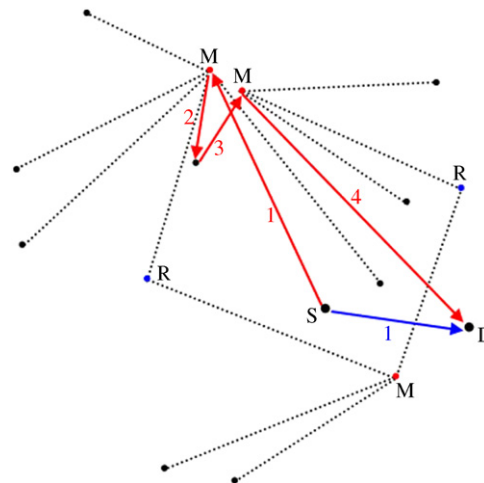


Fig. 13. A snapshot of executing routing protocol.

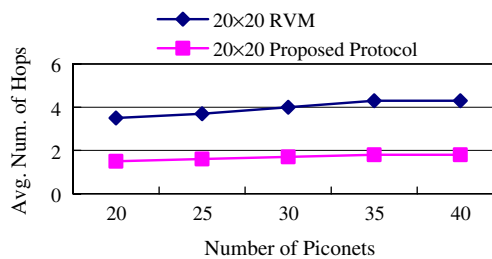


Fig. 14. Comparison in route length with space size 20\*20.

constructed a 4-hop route, while the proposed protocol constructs 1-hop route. The proposed protocol reduces relays on the route, thus minimizing the route length.

A connected scatternet was first randomly constructed with 100 hosts in a given space size. The simulator then randomly selects source and destination hosts from the existing

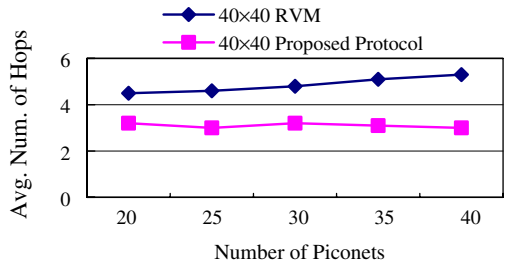


Fig. 15. Comparison in route length with space size 40\*40.

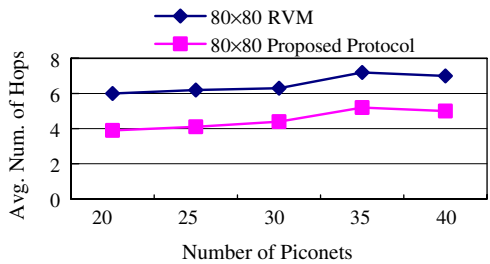


Fig. 16. Comparison in route length with space size 80\*80.

hosts. A search packet is created by source host and is flooded over scatternet for constructing a route, noted by RVM in figures. Hereafter, the route reduction procedure was adopted to create the reduced route, denoted by ‘proposed protocol’ in the figures. Figs. 14–16 compare the number of hop of routes created by RVM and the proposed protocol in space sizes 20\*20, 40\*40 and 80\*80, respectively. The proposed protocol generally creates a shorter path than RVM, because applying the route reduction removes some relays from the route created by RVM. At a space size of 20\*20, the maximum distance between any two devices is  $20\sqrt{2}$  units. Applying the proposed protocol reduces a number of relays on the route created by RVM so that the distance between two neighboring relays on the reduced path is close to 10 units. Thus, the route created by the proposed protocol has a maximum of three hops, regardless of the number of piconets. As shown in Figs. 14 and 15, route created by the proposed protocol has a constant number of hops, in average. However, the number of hops of route created by RVM scheme increases with the number of piconet since the RVM utilizes the flooding scheme to construct the route.

Figs. 17–19 compare RVM with and the proposed protocol, in terms of control packet overhead. Since the proposed protocol performs extra operations to reduce the path length, it created more control traffic overhead than RVM. However, route created by the proposed protocol is expected to have less amount of data packet traffic than that of RVM, since it has fewer relays.

In the following the impact of the network density on the average packet delay and packet lost rate are investigated. As shown in Fig. 20, the proposed protocol outperformed RVM in terms of average packet delay. One reason is that the proposed relay reduction

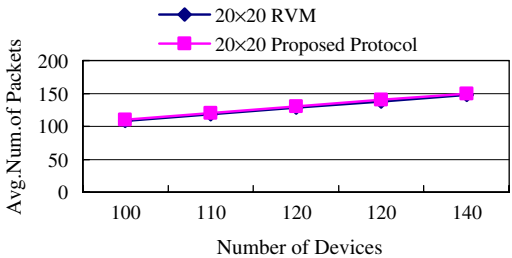


Fig. 17. Comparison in the number of control packet with space size 20\*20.

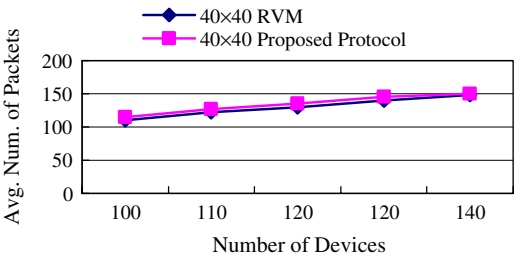


Fig. 18. Comparison in the number of control packet with space size 40\*40.

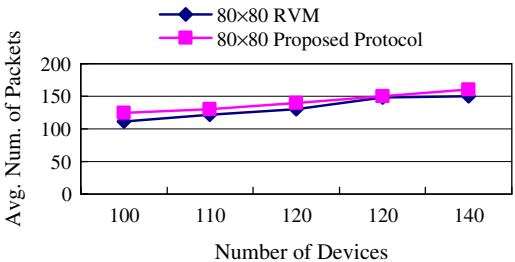


Fig. 19. Comparison in the number of control packet with space size 80\*80.

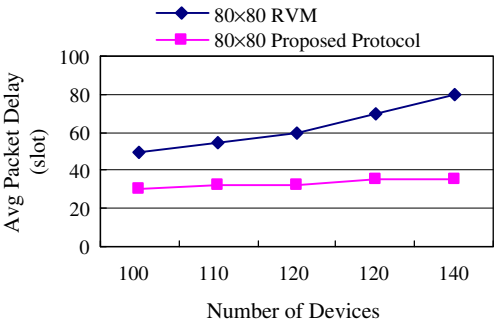


Fig. 20. The impact of network density on throughput.

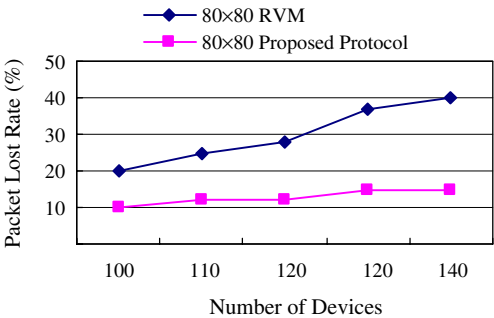


Fig. 21. The impact of network density on packet lost rate.

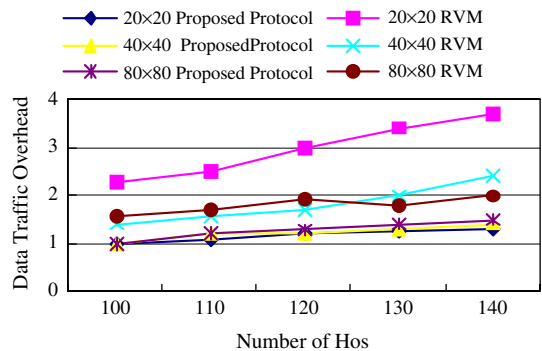


Fig. 22. Comparison of the data packet traffic.

protocol efficiently removes redundant relay nodes, reducing the time required for relay switching among piconets. Another important reason is that the path reduction procedure shortens the route length, thus improving the average packet transmission delay. Fig. 21 depicts that the proposed protocols have smaller packet loss rate than RVM. The proposed relay reduction protocol reduces the number of relays significantly when the number of devices exceeds 120. Packet loss occurs when relays improperly switches among piconets. Therefore, reducing the number of relays improves the packet loss rate. The shortened route of the proposed protocol also contributes to the improvement in packet lost rate. In general, both the proposed relay reduction and path reduction protocols improve the performance in terms of average packet delay and packet loss rate.

Fig. 22 compares the data traffic over routes created by RVM and the proposed protocol. The data traffic overhead of route created by the proposed protocol is normalized to 1, with the space size of 20\*20. Routes constructed by the proposed protocol create a less data traffic than routes constructed by RVM.

6. Conclusions

Bluetooth technology provides low-power, low-cost, interference-resistant wireless communication. To alleviate the flooding overhead, utilize the use of Active Member Address, reduce the difficulty of scheduling and guard time coast in relay, a distributed

protocol is proposed to remove the unnecessary relays in scatternet. Additionally, a routing protocol is proposed herein to reduce the route length and create disjoint routes. The performance study demonstrates that the proposed protocols perform well in terms of routing length and data traffic overhead.

## References

- Baatz S, Frank M, Kuhl C, Martini P, Scholz C. Adaptive scatternet support for Bluetooth using sniff mode, In: Proceedings of the 26th annual conference on local computer networks, LCN 2001. Tampa, Florida, November 2001. p. 112–20.
- Baatz S, Frank M, Kuhl C, Martini P, Scholz C. Bluetooth scatternets: an enhanced adaptive scheduling scheme. In: Proceedings of the IEEE INFOCOM, the annual joint conference of the IEEE computer and communications societies, June 2002, p. 782–90.
- Bhagwat P. Bluetooth technology for short-range wireless apps. *IEEE Internet Comput* 2001;5(3).
- Bhagwat P, Segall A. A Routing vector method (RVM) for routing in Bluetooth scatternets, The sixth IEEE international workshop on mobile multimedia communications (MOMUC'99), November 1999, p. 375–79.
- The Bluetooth Specification, <http://www.bluetooth.org> 1.0b and 1.1
- Chang Y, Park S-C, Lee S. An enhanced throughput and QoS-aware scheduling policy for Bluetooth. In: Proceedings of wireless communications and networking conference, USA, March 2004, p. 1004–7.
- Das A, Ghose A, Razdan A, Saran H, Shorey R. Enhancing performance of asynchronous data traffic over the Bluetooth wireless ad-hoc network. In: Proceedings of IEEE INFOCOM 2001, Alaska, USA, April 2001.
- Haas ZJ, Pearlman MR. The Zone Routing Protocol (ZRP) for Ad-Hoc Networks (internet draft). August 1998.
- Har-Shai L, Kofman R, Zussman G, Segall A. Inter-piconet scheduling in Bluetooth scatternets. In: Proceedings of OPNETWORK, August 2002.
- Har-Shai L, Kofman R, Segall A, Zussman G. Load adaptive inter-piconet scheduling in small-scale Bluetooth scatternets. *IEEE Commun Mag* 2004;42(7):136–42.
- Johnson DB, Maltz DA. Dynamic source routing in Ad Hoc networks. *Mobile Comput* 1996:153–81.
- Kalia M, Bansal D, Shory R. MAC scheduling and SAR policies for Bluetooth: a master driven TDD piconet cellular wireless. In: Proceedings of IEEE international workshop on mobile multimedia communications (MoMuC), San Diego, USA, November 1999.
- Kalia M, Garg S, Shorey R. Scatternet structure and inter-piconet communication in the Bluetooth system. IEEE national conference on communications 2000, New Delhi.
- Kazantzidis M, Gerla M. On the impact of inter-piconet scheduling in Bluetooth scatternets. In: Proceedings of the international conference on internet computing (IC2002), vol. 1, Las Vegas, Nevada, USA, June 2002, p. 37–43.
- Ko YB, Vaidya NH. Location-aided routing (LAR) in mobile Ad Hoc networks. In: Proceedings of the fourth ACM/IEEE international conference on mobile computing and networking (MobiCom'98), October, 1998, p. 66–75.
- Pei Z, Weidong L, Jing W, Youzhen W. Bluetooth—the fastest developing wireless technology. *WCC-ICCT* 2000;2:1657–64.
- Perkins CE, Bhagwat P. Highly Dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In: Proceedings of the conference on communication architectures, protocol, and applications, 1994, p. 234–44.
- Petrioli C, Basagni S. Multihop scatternet formation for Bluetooth networks. *IEEE Vehicular Technol Conf* 2002;1(Spring):424–8.
- Ramachandran L, Kapoor M, Sarkar A, Aggarwal A. Clustering algorithms for wireless Ad Hoc networks. In: Proceedings of the 4th international workshop on discrete algorithms and methods for mobile computing and communications, Boston, MA, USA, August 11, 2000, p. 54–63.
- Salonidis T, Bhagwat P, Tassioulas L. Distributed topology construction of Bluetooth personal area networks. In: LaMaire R, editor. *INFORMCOM* 2001, p 1577–86.
- Tan G, Miu A, Gutttag J, Balakrishnan H. Forming scatternets from Bluetooth personal area networks, MIT Technical Report, MIT-LCS-TR-826, October 2001.
- Zaruba GV, Basagni S, Chlamtac I. Bluetrees—scatternet formation to enable Bluetooth-based ad hoc networks. IEEE international conference on communications (ICC), 2001b, p. 273–7.