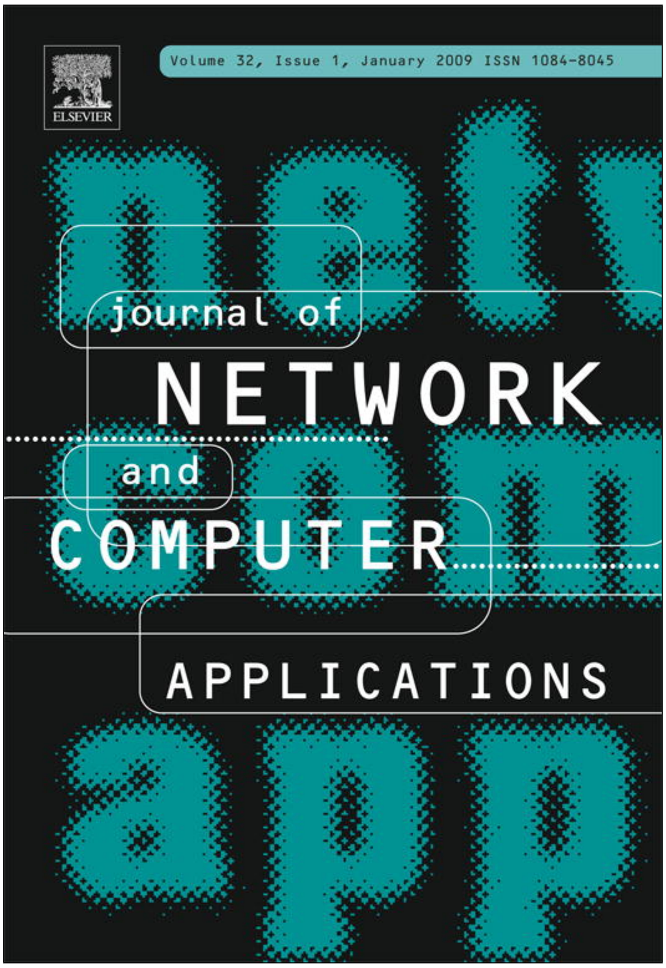

ELSEVIER
Volume 32, Issue 1, January 2009 ISSN 1084-8045
journal of
NETWORK
and
COMPUTER
APPLICATIONS

Available online at www.sciencedirect.com

ScienceDirect

Journal of Network and
Computer Applications 32 (2009) 273–292

Journal of NETWORK and COMPUTER APPLICATIONS

www.elsevier.com/locate/jnca

An overlapping communication protocol using improved time-slot leasing for Bluetooth WPANs

Yuh-Shyan Chen[a,*], Yun-Wei Lin[b], Chih-Yung Chang[c]

[a] *Department of Computer Science and Information Engineering, National Taipei University, Taiwan*
[b] *Department of Computer Science and Information Engineering, National Chung Cheng University, Taiwan*
[c] *Department of Computer Science and Information Engineering, Tamkang University, Taiwan*

Received 3 May 2007; received in revised form 30 January 2008; accepted 22 February 2008

Abstract

In this paper, we propose an overlapping communication protocol using improved time-slot leasing in the Bluetooth WPANS. One or many slave–master–slave communications usually exist in a piconet of the Bluetooth network. A fatal communication bottleneck is incurred in the master node if many slave–master–slave communications are required at the same time. To alleviate the problem, an overlapping communication scheme is presented to allow slave node directly and simultaneously communicates with another slave node to replace with the original slave–master–slave communication works in a piconet. This overlapping communication scheme is based on the improved time-slot leasing (TSL) scheme which modified from the original TSL, while the original TSL scheme only provides the slave-to-slave communication capability. The key contribution of our improved TSL scheme is to offer the overlapping communication capability. With the overlapping communication scheme, we developed an overlapping communication protocol in a Bluetooth WPANs. Finally, simulation results demonstrate that our developed communication protocol achieves the performance improvements on bandwidth utilization, transmission delay time, network congestion, and energy consumption.

Keywords: Bluetooth; Time-slot leasing; WPAN; Communication protocol; Wireless communication; Mobile computing

[*] Corresponding author. Tel.: +886 2 8674 6686; fax: +886 2 2674 4448.
E-mail address: yschen@mail.ntpu.edu.tw (Y.-S. Chen).

1084-8045/$ - see front matter © 2008 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jnca.2008.02.007

1. Introduction

The advances of computer technology and the population of wireless equipment have promoted the quality of our daily life. The trend of recent communication technology is to make good use of wireless equipment for constructing an ubiquitous communication environment. Bluetooth (The Bluetooth^{TM} wireless technology white paper in white paper; Bluetooth Special Interest Group, 2004; McDermott-Wells, 2005b; Rashid and Yusoff, 2006) is a low cost, low power, and short-range communication technology that operates at 2.4 GHz ISM bands. Those features may applicable for many applications, such as the novel technology, wireless sensor networks (WSNs) (Dai et al., 2005; Cai et al., 2006) and short-range wireless personal area networks (WPANs). Energy consumption is a key issue for WSNs and WPANs. The features of low power, low cost are suitable for the physical layer of WSNs and WPANs. Bluetooth is a possible solution for WSNs and WPANs for the applications with the requirements of the higher data rate.

A Bluetooth scatternet is constructed by many piconets. In a piconet, a master node is responsible to manage at most seven active slave nodes and hundred parked slave nodes (McDermott-Wells, 2005a; Riley and Zhang, 2005). A host can participate two or more piconets simultaneously and alternatively play role of slave node in various piconets. The slave node that participates two or more piconets is defined as relay node. A relay node is used to deliver messages among piconets so that the resources or services will not be restricted due to the maximum number of active members in a piconet. The packet transmission among piconets can be achieved by their common *relay*s (Chang and Yu, 2005; Bray and Sturman, 2001; Proulx et al., 2006; Amin and Bhuyan, 2006). When a master node invites a Bluetooth device as its slave node, it must switch to *inquiry* state and then changes to *paging* state after receiving the 48-bit Bluetooth address and clock of slave node. After paging state, a piconet is constructed. A master node polls slave nodes by sent polling packets to slave nodes using round robin (RR) scheme within the piconet. The master node communicates with one slave node and all other slave nodes must hold and wait the polling packet, so the transmission of other slave nodes is arrested. This condition is called the ‘‘transmission holding problem’’. In addition, the master node has to relay the packets from one slave node to the other slave node (Cordeiro et al., 2006), which is incurred double bandwidth and energy consumption, and the slave–slave communication occupies approximately 75% of all connections (Lee et al., 2007).

To reduce the ‘‘transmission holding problem’’, the efficient scatternet topology formation and establishment schemes are investigated. Law et al. develop a new scatternet formation protocol (Law et al., 2000) by optimizing the number of piconets and minimizing the number of Bluetooth device in each piconet. The transmission holding time is slightly decreased because that piconet polling cycle time is reduced. Amin and Bhuyan (2006) develop a Bluetree scatternet formation scheme to construct a bluetree. It reduces the transmission holding probability by using the bluetree scheduling result. A proximity-awareness and fast connection establishment scheme is proposed by Salonidis et al. (2000) to reduce transmission holding problem. Unfortunately, ‘‘transmission holding’’ problem still exists if utilizing the existing scatternet topology formation and establishment schemes.

Many other novel schemes are concurrently investigated, which aims to completely reduce the ‘‘transmission holding’’ problem. Capone et al. (2001) presented an efficient polling scheme to reduce the master and slave queue lengths. With the smaller length of the master and slave queues, the transmission holding time can be decreased. Kalia et al. (1999) investigated an

MAC scheduling scheme to schedule the time division duplex (TDD) time-slots effectively. The transmission holding time can be reduced which compared to the RR based scheduling scheme. Kalia et al. (2000) proposed a capacity increasing scheme by utilizing the park mode to provide more transmission time for the active slave node. The "transmission holding" problem is still existed, because the "transmission holding" problem is originated from the drawback of the master/slave model (Miller and Bisdikian, 2001). Therefore, Zhang et al. initially proposed the dynamic role management scheme (2001) to address the transmission holding problem by using master–slave switching (MSS) mechanism. The MSS mechanism allows slave node directly communicate with other slave node by performing the role switching procedure. A similar result, a piconet partition (PP) scheme, is proposed by Zhang et al. (2001) to partition the original piconet into a set of piconets to effectively solve the "transmission holding" problem. Therefore, the role switching and topology partition schemes can completely solve the "transmission holding" problem, but it produces heavily unnecessary packets due to its dynamic scatternet topologies.

One interest issue is how to develop a novel scheme which can effectively solve the "transmission holding" problem under the fixed-topology situation. To satisfy that purpose, Zhang et al. develops a time-slot leasing (TSL) scheme (Zhang et al., 2002). Zhang et al.'s TSL scheme provides a general mechanism to support the direct slave–slave communication, but the master node uses RR mechanism to check slave node intended to send or receive data. Other slave nodes wait the polling time and gains the transmission holding time. More recently, Cordeiro et al. proposed a QoS-driven dynamic time-slot assignment (DSA) schedule scheme (Cordeiro et al., 2004) to more efficiently utilize TSL scheme. With TSL and DSA scheme, the transmission holding problem still exists. Effort will be made to effectively reduce the transmission holding problem under the fixed topology structure.

In this paper, we propose an overlapping communication protocol using improved TSL in the Bluetooth WPANs. One or many slave–master–slave communications usually exist in a piconet of the Bluetooth network. A fatal communication bottleneck is incurred in the master node if many slave–master–slave communications are required at the same time. To alleviate the problem, an overlapping communication scheme is presented to allow slave node directly and simultaneously communicates with another slave node to replace with the original slave–master–slave communication works in a piconet. This overlapping communication scheme is based on the improved TSL scheme which modified from the original TSL scheme, while the original TSL scheme only provides the slave-to-slave communication capability. The key contribution of our improved TSL scheme additionally offers the overlapping communication capability. With the overlapping communication scheme, we developed an overlapping communication protocol in a Bluetooth WPANs. Finally, simulation results demonstrate that our developed communication protocol achieves the performance improvements on bandwidth utilization, transmission delay time, network congestion, and energy consumption.

This paper is organized as follows. Section 2 describes the basic idea of our new scheme. The new communication protocol is presented in Sections 3 and 4. The performance analysis is discussed in Section 5. Section 6 concludes this work.

2. Basic idea

Bluetooth is the technique for low power, low cost, and short-range communication for WPANs. It operates in the unlicensed industrial–scientific–medical (ISM) band and employs a frequency-hopping spread-spectrum (FHSS) technique to avoid system interference. The frequency-hopping sequence is used by all Bluetooth devices in a piconet. Bluetooth devices must switch to the same channel to send and receive data. In a piconet, Bluetooth device cannot easily communicate with each other, even if these piconets in the vicinity, because these master nodes in each piconet are different. The master node manages the communication time by setting the different type of transmission packet. Besides, the master node schedules the communication order by sending a polling packet to each slave node through a RR mechanism. In this way, if a slave node does not receive the polling packet, all transmitted packets must be hold until receiving the polling packet from the master node. This is the "transmission holding" problem. Efforts will be made in this work to develop a novel overlapping communication protocol to completely solve the "transmission holding" problem without the adjustment of the scatternet topology.

The transmission holding problem is originated from the drawback of the master/slave model. In a piconet, since the slave node may transmit packets only if it receives the polling packet from master node. As a result, when there are many salves have to transmit data, slave nodes must hold its transmission until receiving the polling packet, as shown in Fig. 2(a). To solve the transmission holding problem, a TSL approach (Zhang et al., 2002) has been proposed. The seven phases of the TSL procedure are given in Fig. 1(a). After the procedure executed, as shown in Fig. 1(b), a temp-piconet is established. Therefore, slave nodes can directly transmit packets to each other without the master mode relaying. Using TSL approach, the waiting time of the other holding slave nodes are reduced. Unfortunately, the effect of transmission holding problem is reduced, but it still exists. To solve the transmission holding problem completely and overcome the drawback of RR scheme, an overlapping communication scheme is investigated in this work to offer the overlapping communication capability for multi-pair of devices within a piconet. With the overlapping communication scheme, Bluetooth device can simultaneously and directly communicate with each other. The performance of communication will be improved.

In the following, we describe the main contribution of our scheme, compared to TSL scheme (Zhang et al., 2002) and QoS-driven dynamic time-slot assignment (DSA) scheme (Cordeiro et al., 2004). The FHSS is used in the Bluetooth network. Let $C(x)$ denote the used channel for time-slot x, and $\overrightarrow{\alpha\beta}$ denote slave node S_α sends data to slave node S_β. An

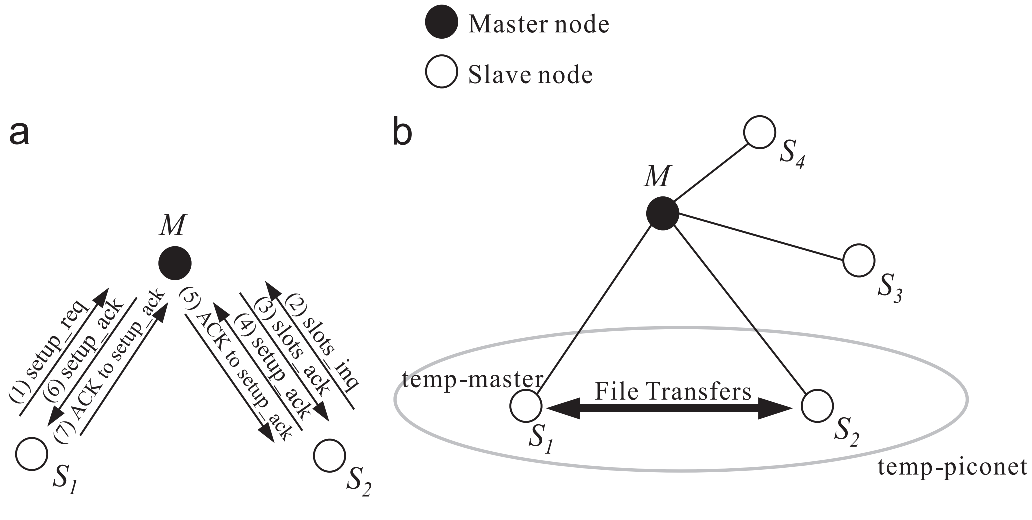


Fig. 1. The procedure of time-slot leasing.

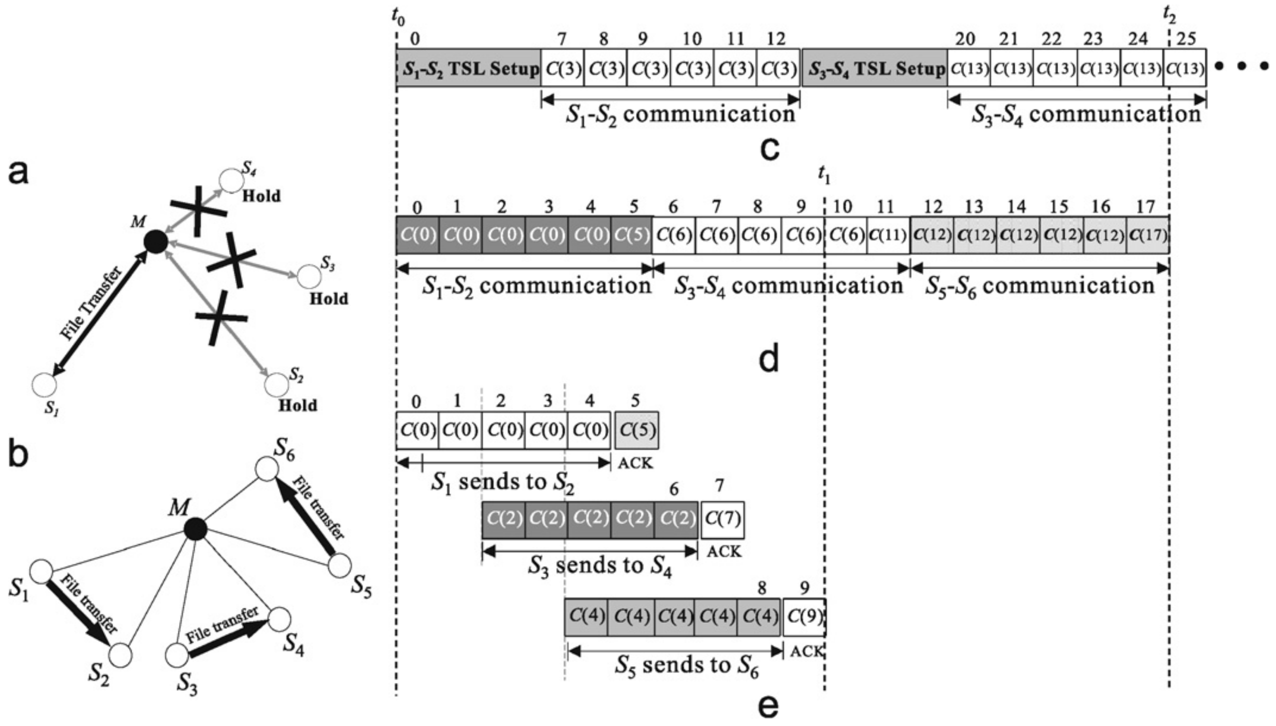

a
S_4
Hold
M
S_3
Hold
File Transfer
S_2
Hold
S_1
b
S_6
M
File transfer
S_1
S_5
File transfer
File transfer
S_4
S_2
S_3
t_0
t_2
0 7 8 9 10 11 12 20 21 22 23 24 25
S_1-S_2 TSL Setup
C(3) C(3) C(3) C(3) C(3) C(3)
S_3-S_4 TSL Setup
C(13) C(13) C(13) C(13) C(13) C(13)
S_1-S_2 communication
S_3-S_4 communication
c
t_1
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
C(0) C(0) C(0) C(0) C(0) C(5) C(6) C(6) C(6) C(6) C(6) C(11) C(12) C(12) C(12) C(12) C(12) C(17)
S_1-S_2 communication
S_3-S_4 communication
S_5-S_6 communication
d
0 1 2 3 4 5
C(0) C(0) C(0) C(0) C(0) C(5)
ACK
S_1 sends to S_2
6 7
C(2) C(2) C(2) C(2) C(2) C(7)
ACK
S_3 sends to S_4
8 9
C(4) C(4) C(4) C(4) C(4) C(9)
ACK
S_5 sends to S_6
e

Fig. 2. (a) Transmission holding problem, (b) three communication requests, (c) TSL, (d) DSA, and (e) overlapping communication protocol.

example is shown in Fig. 2(b), $\overrightarrow{S_1S_2}$, $\overrightarrow{S_3S_4}$, and $\overrightarrow{S_5S_6}$simultaneously occur in a piconet. The transmission holding problem is heavily occurred in master node M. Fig. 2(c) shows that the time cost is more than $t_2 - t_0$ by using TSL scheme for the $\overrightarrow{S_1S_2}$, $\overrightarrow{S_3S_4}$, and $\overrightarrow{S_5S_6}$. With the same works of $\overrightarrow{S_1S_2}$, $\overrightarrow{S_3S_4}$, and $\overrightarrow{S_5S_6}$, time cost is obviously reduced to $t_2 - t_0$ by using the DSA scheme as illustrated in Fig. 2(d). However, the time cost will be improved by using our overlapping communication scheme. Fig. 2(e) illustrates that the works of $\overrightarrow{S_1S_2}$, $\overrightarrow{S_3S_4}$,and $\overrightarrow{S_5S_6}$ can be accomplished in time $t_1 - t_0$.

Packets with different packet types are listed in the Table 1. The key idea of our overlapping communication is to use the long packet, packets with packet type DH5 or DM5. In the following, we describe the reason. In general, the bandwidth utilization of DM5 (DH5) is higher than DM3 (DH3) and DM1 (DH1). The similar idea of using the long packet is used in S-MAC protocol for WSN (Ye et al., 2006) to improve the bandwidth utilization but slightly increase the fairness problem. With utilizing Bluetooth's frequency-hopping property, our overlapping scheme not only improve the bandwidth utilization but also reduce the fairness problem.

To illustrate the frequency hopping technology, every time-slot during the transmission adopts the different channel, we let channel $FH(x)$ denote the frequency used at time-slot x. Fig. 3(a) shows that if a DH1 packet is used, then the frequency hopping sequence which can be represented as $FH(0..7) = 2, 6, 3, 1, 7, 5, 8,$ and 4. From the specification of the Bluetooth system 1.2 (Bluetooth Special Interest Group, 2004), the consecutive five time slots keep the same channel if using a DH5 packet. The rule is same for packets DH3, DM3, and DM5. An example is given in Fig. 3(b). Let channel $C(x)$ denote a Bluetooth

Table 1
Bluetooth ACL data packet

Type	Used payload (bytes)	Bandwidth utilization (bytes/slot)
DH1	27	27.0
DH3	183	61.0
DH5	339	67.8
DM1	17	17.0
DM3	121	40.3
DM5	224	44.8

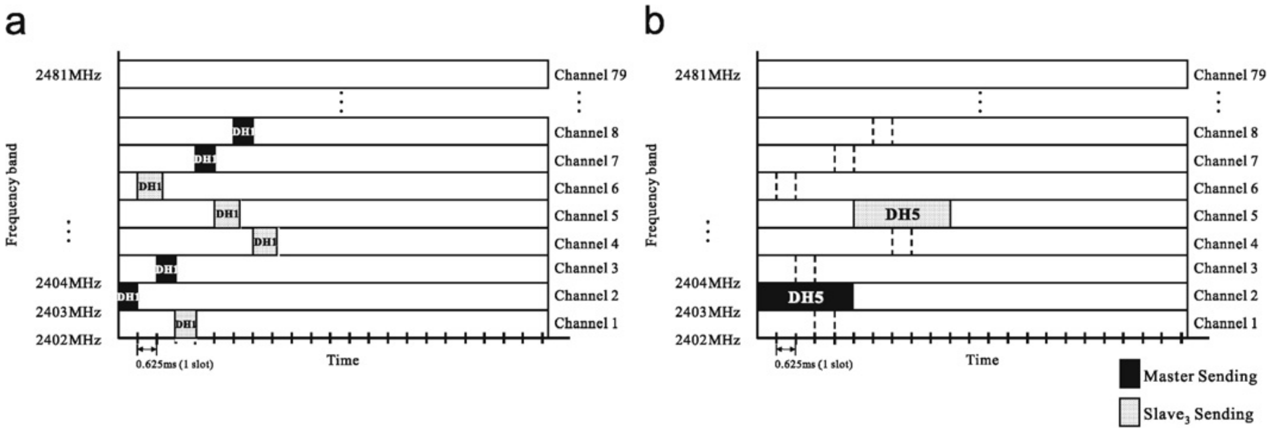


Fig. 3. (a) Transmission using packet DH1 and (b) transmission using packet DH5.

device sends a packet at time-slot x using the channel $C(x)$. If a device sends a DH1 packet at time-slot x, then we have the result of $C(x) = FH(x)$, $C(x+1) = FH(x+1)$, $C(x+2) = FH(x+2)$, $C(x+3) = FH(x+3)$, $C(x+4) = FH(x+4)$, where $C(x)\neq C(x+1)\neq C(x+2)\neq C(x+3)\neq C(x+4)$. But if a device sends a DH5 packet at time-slot x, then five connective time slots use the same channel, $C(x) = C(x+1) = C(x+2) = C(x+3) = C(x+4) = FH(x)$. Observe that channels $F(x+1)$, $F(x+2)$, $F(x+3)$, and $F(x+4)$ in the original frequency hopping sequence are free. Example is given in Fig. 3(b), channels 2 and 5 are used, but channels 6, 3, 1, 7 are free. It is possibly to utilize these free time slots to offer an overlapping communication protocol. Efforts are made to significantly improve the throughput in a scatternet by using our new overlapping communication scheme. This work is achieved by developing intra-piconet and inter-piconet overlapping protocols, which are presented in the following sections. The intra-piconet overlapping communication protocol is shown in Fig. 4(a) and (b). The data transmission of $\overleftrightarrow{M_1S_3}$ and $\overleftrightarrow{S_1S_2}$ can be overlapped. In addition, Fig. 4(c) and (d) illustrates the overlapping condition for performing the intra-piconet overlapping communication protocol.

3. Intra-piconet overlapping routing protocol

To significantly overcome the "transmission holding problem", an intra-piconet overlapping communication protocol is presented in a piconet. The intra-piconet

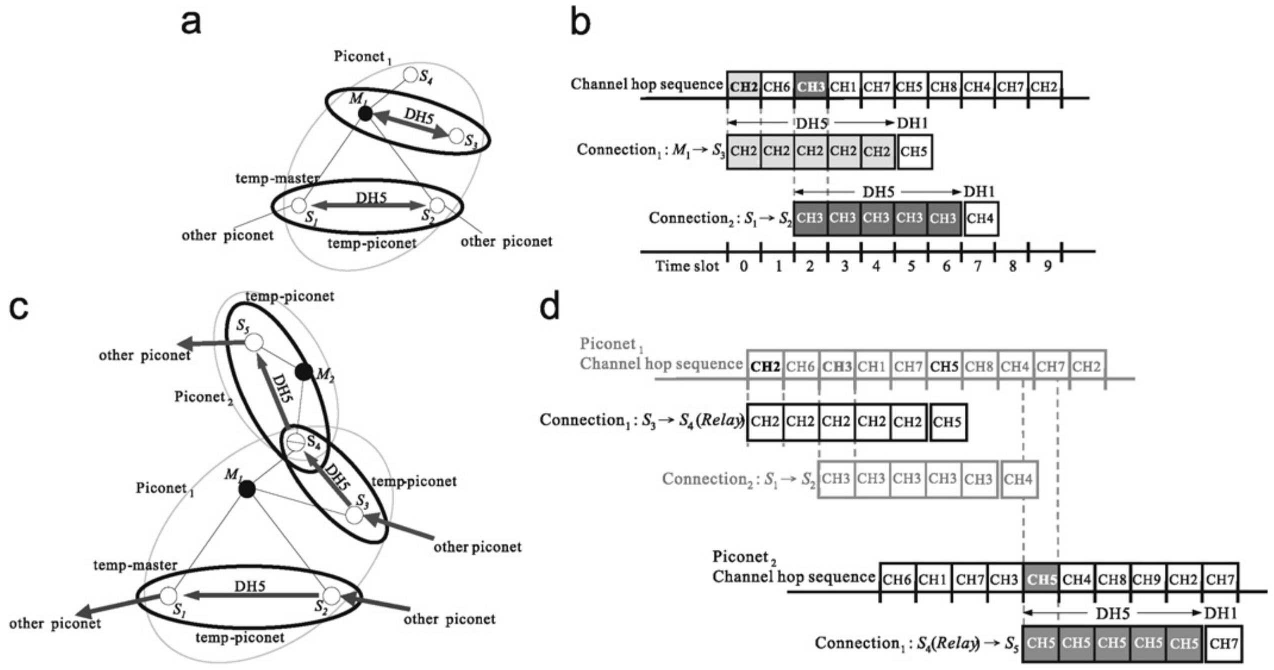

a
Piconet$_1$
S_4
M_1
DH5
S_3
temp-master
DH5
S_1
S_2
other piconet
temp-piconet
other piconet
b
Channel hop sequence CH2 CH6 CH3 CH1 CH7 CH5 CH8 CH4 CH7 CH2
DH5 DH1
Connection$_1$: $M_1 \rightarrow S_3$ CH2 CH2 CH2 CH2 CH2 CH5
DH5 DH1
Connection$_2$: $S_1 \rightarrow S_2$ CH3 CH3 CH3 CH3 CH3 CH4
Time slot 0 1 2 3 4 5 6 7 8 9
c
temp-piconet
S_5
other piconet
M_2
DH5
Piconet$_2$
S_4
DH5
temp-piconet
M_1
Piconet$_1$
S_3
other piconet
temp-master
DH5
S_1
S_2
other piconet
other piconet
temp-piconet
d
Piconet$_1$
Channel hop sequence CH2 CH6 CH3 CH1 CH7 CH5 CH8 CH4 CH7 CH2
Connection$_1$: $S_3 \rightarrow S_4$ (Relay) CH2 CH2 CH2 CH2 CH2 CH5
Connection$_2$: $S_1 \rightarrow S_2$ CH3 CH3 CH3 CH3 CH3 CH4
Piconet$_2$
Channel hop sequence CH6 CH1 CH7 CH3 CH5 CH4 CH8 CH9 CH2 CH7
DH5 DH1
Connection$_1$: S_4(Relay) $\rightarrow S_5$ CH5 CH5 CH5 CH5 CH5 CH7

Fig. 4. The concept of overlapping communication scheme.

overlapping communication protocol is divided into two phases: (1) *queuing scheduling* and (2) *overlapping time-slot assignment*. Initially, the queuing scheduling phase forms two components, *data-flow matrix* and *queuing table*. The data-flow matrix collects the information of the transmitted data amount from all slave nodes. The queuing table determines the transmission order for all salves nodes based on the data-flow matrix. The overlapping time-slot assignment phase utilizes the queuing table to schedule the overlapping time-slot assignment. Finally, the execution phase broadcasts the scheduling result to all slave nodes and to perform the data transmissions based on the overlapping result.

3.1. Queuing scheduling phase

The queuing scheduling phase is to achieve the overlapping communication schedule. To complete the overlapping communication, master node initially forms *data-flow matrix* and *queuing table*. This work can be done in the *BTIM* window as shown in Fig. 5(c). During a schedule interval, each source node (Bluetooth device) in a piconet just can transmit data to one destination node. The amount of data transmission of all pair of source-destination nodes is kept in the master node, and can be stored in a *data flow matrix*

$$DM_{m\times m+1} = \begin{bmatrix} D_{10} & \dots & D_{1m} \\ \vdots & D_{ij} & \vdots \\ D_{m0} & \dots & D_{mm} \end{bmatrix},$$

where D_{ij} denotes the amount of data transmits from node N_i to node N_j, where $1 \leqslant i \leqslant m$, $0 \leqslant j \leqslant m$, and m is the number of slave nodes in a piconet. Observe that node N_j is the

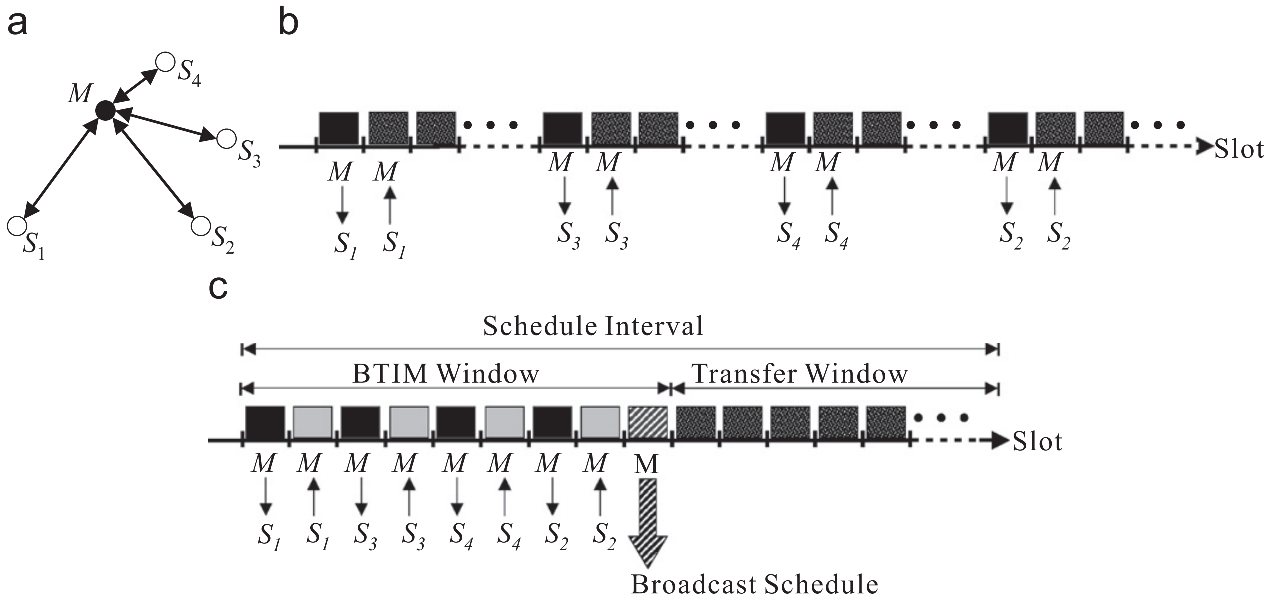
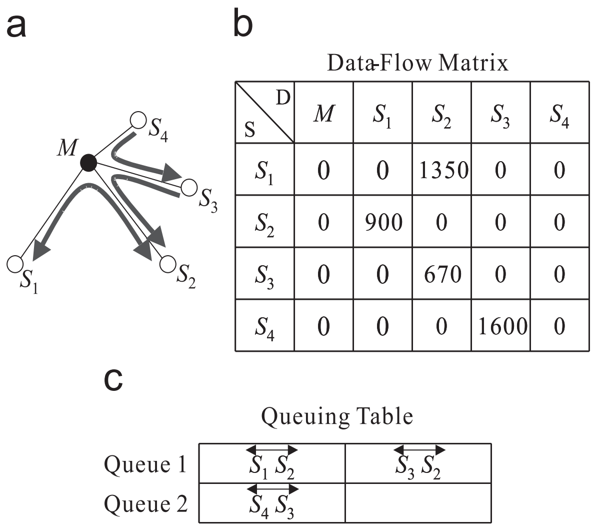


Fig. 5. (a) Master polling, (b) polling and transmission in origin piconet, and (c) the structure of BTIM.



Data-Flow Matrix

S \ D	M	S_1	S_2	S_3	S_4
S_1	0	0	1350	0	0
S_2	0	900	0	0	0
S_3	0	0	670	0	0
S_4	0	0	0	1600	0

Queuing Table

Queue 1	$\overleftrightarrow{S_1 S_2}$	$\overleftrightarrow{S_3 S_2}$
Queue 2	$\overleftrightarrow{S_4 S_3}$	

Fig. 6. (a) An example of piconet communication, (b) flow matrix, and (c) *n*-queue.

master node if $j = 0$. For example as shown in Fig. 6(a),

$$DM_{4\times 5} = \begin{bmatrix} 0 & 0 & 1350 & 0 & 0 \\ 0 & 900 & 0 & 0 & 0 \\ 0 & 0 & 670 & 0 & 0 \\ 0 & 0 & 0 & 1600 & 0 \end{bmatrix}$$

is shown in Fig. 6(b). Effort will be made to fully utilize the information of the *data flow matrix*

$$DM_{m\times m+1} = \begin{bmatrix} D_{10} & \dots & D_{1m} \\ \vdots & D_{ij} & \vdots \\ D_{m0} & \dots & D_{mm} \end{bmatrix}$$

to produce the overlapping communication schedule.

Master node further calculates a *queuing sequence* based on data-flow matrix $DM_{m\times m+1}$. Our overlapping communication scheme is to utilize the long packet, since the high utilization of the long packet can significantly increase the chance of overlapping communication.

Before describing the overlapping time-slot assignment operation, we define the following notations. First, a link with a greater number of data has the higher priority for the data transmission. Therefore, we define priority function $pri(\overleftrightarrow{ij})$ of link $\overleftrightarrow{ij}$ as

$$pri(\overleftrightarrow{ij}) = \frac{D_{ij} + D_{ji}}{|D_{ij} - D_{ji}|}.$$

A link queue Q is defined to record $m+1$ pairs of source-to-destination links $\overleftrightarrow{i_0j}, \ldots, \overleftrightarrow{i_kj}, \overleftrightarrow{i_{k+1}j}, \ldots,$ and $\overleftrightarrow{i_mj}$ in a piconet, where all of these links have the same destination node N_j. Further, let $Q = \{\overleftrightarrow{i_0j}, \ldots, \overleftrightarrow{i_kj}, \overleftrightarrow{i_{k+1}j}, \ldots, \overleftrightarrow{i_mj}\}$, where $pri(\overleftrightarrow{i_kj}) > pri(\overleftrightarrow{i_{k+1}j})$ and $0 \leqslant k \leqslant m-1$. For instance, $\{\overleftrightarrow{S_1S_2}, \overleftrightarrow{S_3S_2}\}$.

Assumed that there are n link queues $Q_1, Q_2, \ldots, Q_q, \ldots, Q_n$, where each Q_q has different destination node, and $1 \leqslant q \leqslant n$, where n is real number. For instance, $Q_1 = \{\overleftrightarrow{S_1S_2}, \overleftrightarrow{S_3S_2}\}$ and $Q_2 = \{\overleftrightarrow{S_4S_3}\}$. Given $Q_q = \{\overleftrightarrow{i_0j_q}, \ldots, \overleftrightarrow{i_kj_q}, \overleftrightarrow{i_{k+1}j_q}, \ldots, \overleftrightarrow{i_mj_q}\}$, we denote $MAX(Q_q) = MAX_{k=0}^{m}\, pri(\overleftrightarrow{i_kj_q}) = pri(\overleftrightarrow{i_0j_q})$. All of $Q_1, Q_2, \ldots, Q_q, \ldots, Q_n$ can be combined into a

$$\text{queuing sequence} = \begin{pmatrix} Q_1 \\ \vdots \\ Q_q \\ Q_{q+1} \\ \vdots \\ Q_n \end{pmatrix},$$

where $MAX(Q_q) > MAX(Q_{q+1})$, and $1 \leqslant q \leqslant n$. For example as shown in Fig. 6(a), the queuing sequence is

$$\begin{pmatrix} \{\overleftrightarrow{S_1S_2}, \overleftrightarrow{S_3S_2}\} \\ \{\overleftrightarrow{S_4S_3}\} \end{pmatrix},$$

where $pri(\overleftrightarrow{S_1S_2}) = \frac{2250}{450}$, $pri(\overleftrightarrow{S_3S_2}) = \frac{670}{670}$, $pri(\overleftrightarrow{S_4S_3}) = \frac{1600}{1600}$, $MAX(Q_1) = pri(\overleftrightarrow{S_1S_2}) = \frac{2250}{450}$, and $MAX(Q_2) = pri(\overleftrightarrow{S_4S_3}) = \frac{1600}{1600}$, as illustrated in Fig. 6(c).

3.2. Overlapping time-slot assignment phase

The queuing scheduling phase determines the appropriate transmission order which records in a queuing sequence. This queuing sequence is used in the overlapping time-slot assignment phase to assign suitable time slots for each transmission. In the overlapping time-slot assignment phase, there are two conditions occurred, collision-free and collision detection. In the following, we introduce collision-free and collision detection.

In the collision-free condition, the overlapping time-slot assignment phase assign suitable time slots to each transmission $\overleftrightarrow{ij}$ according to the order in the queue sequence. Q_q processes before Q_{q+1} in queue sequence. Each Q_q starts to process at time-slot $2k$, and $q \leqslant k \leqslant n$. For higher bandwidth utilization, DH5 packet type is the prior choice to transmit data. From the specification of the Bluetooth system 1.2 (Bluetooth Special Interest Group, 2004), the consecutive five time slots keep the same channel if master node uses a DH5 packet. Hence, four free channels refers to four free time slots in DH5 packets and no free channel refers to any free time slot in DH1 packets. Therefore, DH1 is helpless for overlapping communication scheme. But master node only uses DH3 and DH5 packet to fragment transmitted data, total amount of packet size could not fit total amount of transmitted data without using DH1. Some excess bytes might deliver. Overlapping communication scheme calculates appropriate packet type to minimize the excess bytes and assigns packets to free time slots to transmit data using improved TSL. Transmission using the appropriate packet type improves bandwidth and decrease the energy consumption. We let n_DH5 and n_DH3 denote the amount of DH5 and DH3 packet type, respectively, D_{ij} denote the amount of data transmission from node N_i to node N_j, and *Excess* denote the excess part of transmission. For each transmission, the master node arranges appropriate packet type by a packet type distribution rule as follows:

$$n\_DH5 * 339 + n\_DH3 * 183 = D_{ij} + min(Excess).$$

The packet amount of DH5 and DH3 is 339 and 183 bytes, respectively. A set of packet type distribution $PD_{ij} = \{pd_{ij_0}, pd_{ij_1}, \ldots, pd_{ij_s}, \ldots, pd_{ij_p}\}$ is used to record $p+1$ packet type distribution from node N_i transmitting to node N_j, where pd_{ij_s} denotes the tth transmission packet type from node N_i to node N_j and $0 \leqslant s \leqslant p$. The value of p is $n\_DH5 + n\_DH3$, where n_DH5 and n_DH3 are related to pd_{ij_s}. For example in Fig. 6(b), $D_{S_1S_2}$ is 1350 bytes composed of four DH5 packets ($n\_DH5 = 4$), $PD_{S_1S_2} = \{5, 5, 5, 5\}$ and $D_{S_2S_1}$ is 900 composed of three DH5 packets bytes ($n\_DH5 = 3$), $PD_{S_2S_1} = \{5, 5, 5\}$. Master node gets a PD_{ij} for link $\overleftrightarrow{ij}$ by packet type distribution rule. Set PD_{ij} is used to predict the occupied time slots by link $\overleftrightarrow{ij}$.

According to the packet type distribution, the overlapping time-slot assignment phase assigns suitable time slot to use improved TSL for data transmission. A time-slot WS_{ij} which is defined as the i's wake up time slot between link $\overleftrightarrow{ij}$. A set of time-slot offset $SO_{ij} = \{so_{ij_0}, so_{ij_1}, \ldots, so_{ij_t}, \ldots, so_{ij_q}\}$ is defined to record $q+1$ time-slot offset from node N_i transmitting to node N_j, and $0 \leqslant t \leqslant q$. The so_{ij_t} denotes the tth time-slot offset transmission from i to j. SO_{ij} is accumulated from PD_{ij} and PD_{ji}, which is used to predict the occupied time-slot offset by link $\overleftrightarrow{ij}$. For using improve TSL, master node announces each slave node WS_{ij} and SO_{ij}. Slave nodes wake up at assigned time slots WS_{ij} and obey the set of time-slot offset SO_{ij} to transmit data.

Using improved TSL, a temp-piconet is established. Two roles are assigned in this temp-piconet, one is temp-master node, the other are original slave nodes. One of slave nodes in the temp-piconet turned into temp-master node after improved TSL execution. About overlapping communication scheme, master node assigns the roles,

WS_{ij}, and SO_{ij} to each transmission $\overleftrightarrow{ij}$ at BTIM. We consider two roles, temp-master node and slave node and if $D_{ij} > D_{ji}$ then N_i is temp-master node and N_j is slave node. For a transmission from temp-master node N_i to slave node N_j, master node assigns time slots to link $\overleftrightarrow{ij}$ as follows:

$$\begin{cases} SO_{ji}\colon so_{ji_t} = so_{ij_t} + pd_{ij_s}, \\ SO_{ij}\colon so_{ij_{t+1}} = so_{ji_t} + pd_{ji_s} \end{cases}.$$

The so_{ij_0} initially sets to 0. During computing SO_{ij}, if master node runs out of the PD_{ij} set, $pd_{ji_{p+t}}$ sets to 1 for ACK responding. Master node assigns the nearest free time slot as WS_{ij} and $WS_{ji} = WS_{ij} + pd_{ij_1}$. $WS_{ij} + SO_{ij}$ is the real time slot. For example about $\overleftrightarrow{S_1S_2}$ in Fig. 6(c), $D_{S_1S_2}$ is 1350 bytes, $SO_{S_1S_2} = \{0, 10, 20, 30\}$, $D_{S_2S_1}$ is 900 bytes, $SO_{S_2S_1} = \{5, 15, 25, 35\}$, $WS_{S_1S_2} = 0$, $WS_{S_2S_1} = 5$, and the all transmission process of $\overleftrightarrow{S_1S_2}$ as shown in Fig. 7(a).

In the collision detection condition, some occupied time slots have been assigned again to transmission possibly. Before the data transmission, the master node should detect this condition first. A set of used time-slot $US = \{us_0, us_1, \ldots, us_k\}$ is defined to record $k+1$ used time slots in a piconet. The master node adds $WS_{ij} + SO_{ij}$ into US after assigned time slots for N_i. If the master node assigns time slots including in US to slave nodes, we called *time-slot collision*. Master node uses the following equation to check the collision status:

$$(WS_{ij} + SO_{ij}) \cap US \neq \{\phi\}.$$

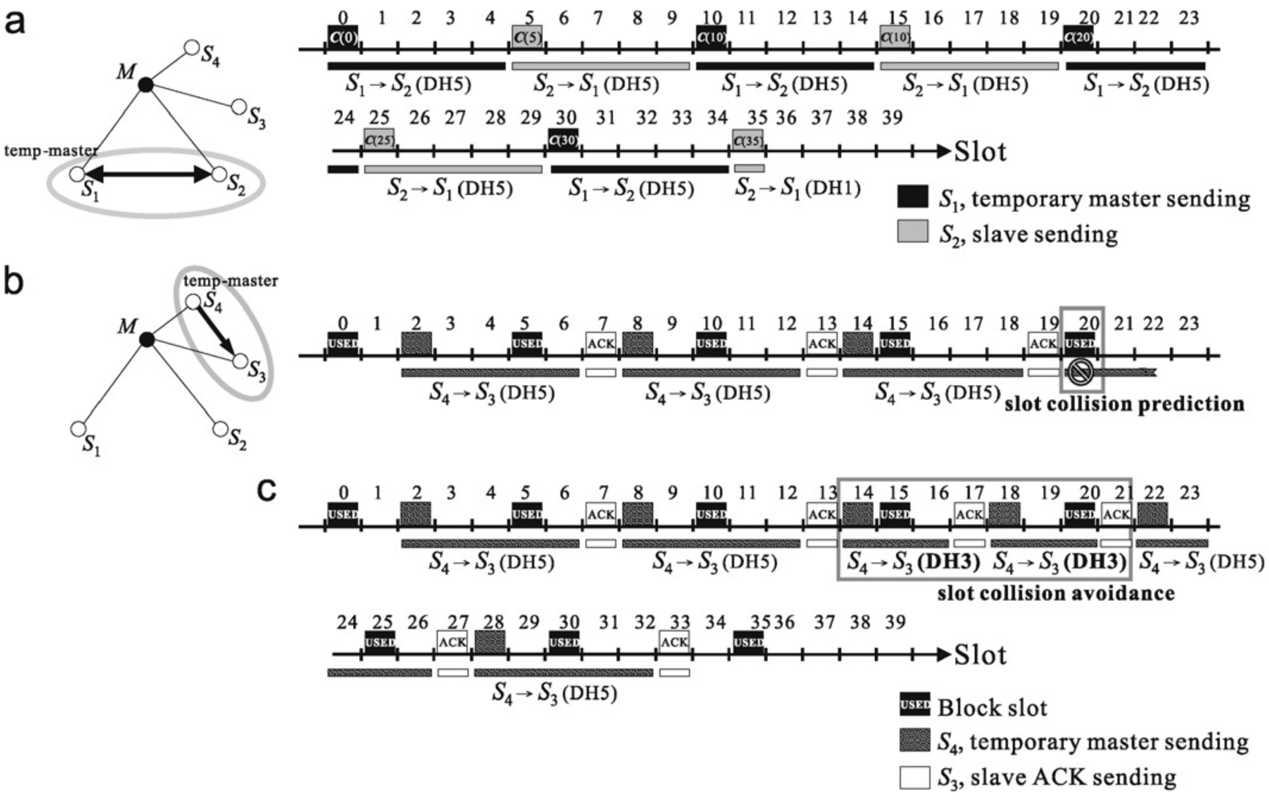


Fig. 7. (a) The block time-slot calculation, (b) collision occurred, and (c) packet type changing.

If the equation is true, the problem of time-slot collision is detected. For example as shown in Fig. 7(a), $\overleftrightarrow{S_1S_2}$ uses the actual time slots recorded in $US = \{0, 5, 10, 15, 20, 25, 30, 35\}$. As shown in Fig. 7(b), master node assigns the time slot for $\overrightarrow{S_4S_3}$ with $SO_{S_4S_3} = \{0, 6, 12, 18, 24\}$ and $WS_{S_4S_3} = 2$. Master node checks $(WS_{S_4S_3} + SO_{S_4S_3}) \cap US = \{20\}$. The result set is non-empty. Collision is detected in assigned time slots for $\overrightarrow{S_4S_3}$.

In the following, we describe how to solve the time-slot collision problem. When the time slot is collision, master node adapts the following rules to avoid the collision.

(S1) /* Avoidance for DH5 packet */
If the tth collision packet type is DH5, master node changes the packet to two DH3 packets.
Master node changes the $pd_{ij_t} = \{5\}$ to $pd_{ij_t} = \{3, 3\}$.

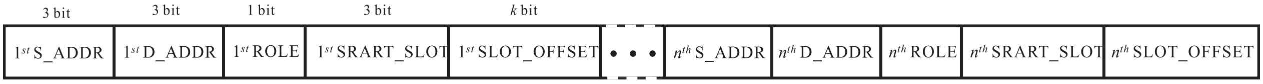
(S2) /* Avoidance for DH3 packet */
If the tth collision packet type is DH3, master node changes the packet to one DH5 packet.
Master node changes the $pd_{ij_t} = \{3\}$ to $pd_{ij_t} = \{5\}$.

(S3) /* Avoidance for DH1 packet */
If the tth collision packet type is DH1, master node changes the packet to one DH3 packet.
Master node changes the $pd_{ij_t} = \{1\}$ to $pd_{ij_t} = \{3\}$.

After the master node adapting above rules to $\overrightarrow{S_4S_3}$, the transmission process has shown in Fig. 7(c). After calculated the transmission time-slot assignment, we record the time-slot assignment in a packet called time-slot-assignment-schedule (SAS). The detailed information of SAS packet field has shown in Table 2. The master node gathers those all communication information into an SAS packet as shown in Fig. 8 and broadcasts the SAS packet to all slave nodes sequentially to announce the transmission schedule. Slave nodes obey the schedule to wake up to transmit/receive. If a device does not do anything, it changes to sleep mode to save energy until next BTIM. During transfer windows, devices are only active in transmitting/receiving time slots, other time slots stay in sleep mode to save energy.

Table 2
The information about each field in the SAS packet

Packet field	Field description
S_ADDR	The BD_ADDR of the source device (3-bits)
D_ADDR	The BD_ADDR of the destination device (3-bits)
ROLE	The role of device (temporary master or slave) (1-bits)
WAKE_SLOT	The wake up time slot (3-bits)
SLOT_OFFSET	The offset time-slot assignment (k-bits)



3 bit	3 bit	1 bit	3 bit	k bit						
1^{st} S_ADDR	1^{st} D_ADDR	1^{st} ROLE	1^{st} SRART_SLOT	1^{st} SLOT_OFFSET	• • •	n^{th} S_ADDR	n^{th} D_ADDR	n^{th} ROLE	n^{th} SRART_SLOT	n^{th} SLOT_OFFSET

Fig. 8. The SAS packet format.

4. Inter-piconet overlapping routing protocol

This section presents the inter-piconet overlapping routing protocol as follows. Relay node operation is a high challenge design issue in a scatternet communication. The role of a relay node is a bridge to transmit and forward packets between different master nodes. In general, the schedule of relay node is negotiated by master nodes. Relay node switches to a different piconet to forward and transmit packets. Master node must keep packets when its corresponding relay node temporarily switches to a different piconet. The packet delay time is surely increased. Some research results (Proulx et al., 2006; Chang and Yu, 2005) had been proposed for the relay node scheduling. Proulx et al. (2006) proposed a relay node scheduling using sniff mode. Salonidis et al. (2000) constructed a long-lived scatternet in Bluetooth networks. These two relay-node scheduling protocols are based on the negotiation scheme between master nodes. Observe that relay node scheduling is difficult to be maintained because master node do not easily to have schedule information from other master nodes. For this reason, our scheme try to arrange relay node schedule by relay node itself. Relay node manages the available time slots in each piconet according to routing information between piconets. The scatternet formation algorithm proposed by Yu et al. (2007) which is used to construct our scatternets. Relay node has low degree by Yu et al.'s algorithm (2007). Different scatternet formations do not affect the intra-piconet routing protocol, but it affects the inter-piconet routing protocol due to the degree of relay node. If the degree of relay node is large, relay node needs more time to switch between different piconets and takes more service time. Our scatternet formation algorithm is modified the Yu et al.'s algorithm (2007) to further control the degree of relay node to improve the performance of inter-piconet routing protocol. Relay node acquires routing information by joining the three specific time slots in the end of BTIM in each piconet. Let ST_1, ST_2, and ST_3 denote the first, second, and third specific time slots, respectively. The detail relay node operations is given as follows:

(S1) In ST_1, a master node sends the routing information

$$DM_{m\times 1} = \begin{bmatrix} D_{0R} \\ \vdots \\ D_{iR} \\ \vdots \\ D_{mR} \end{bmatrix}$$

to a relay node, where D_{iR} denotes the amount of data which is transmitted from N_i to the relay node, $0 \leqslant i \leqslant m$, and m is the number of slave nodes in a piconet. Observe that node N_i is the master node if $i = 0$. Relay node has two routing information, $DM_{m\times 1}$ and $DM_{1\times m}$. $DM_{1\times m} = [D_{R0} \cdots D_{Rj} \cdots D_{Rm}]$ is collected previously from other piconets, where D_{Rj} denotes the amount of data which is transmitted from the relay node to N_j, where $0 \leqslant j \leqslant m$, and m is the number of slave nodes in a piconet. Observe that N_j is a master node if $j = 0$. Recall the queuing scheduling phase and overlapping time-slot assignment phase as mentioned in Section 3, the relay node calculates time-slot sets WS_{iR}, SO_{iR}, WS_{Ri}, and SO_{Ri} for receiving data, and WS_{Rj}, SO_{Rj}, WS_{jR}, and SO_{jR} for transmitting data. The relay node maintains a counter C_i

for piconet P_i, where $0 \leqslant C_i \leqslant C_{max}$. To avoid starvation, let C_{max} be a constant. If the relay node transmits completely for piconet P_i in a transfer window, then $C_i = C_i - 1$, otherwise $C_i = C_i + 1$. The relay node assigns tw_i transfer time slots to piconet P_i based on the counter C_i. Let TW denote the windows size of relay node's transfer window. The relay node assigns tw_i transfer time slots to each piconet P_i by

$$tw_i = TW \times \frac{C_i}{\sum_{i=1}^{TP} C_i},$$

where TP denotes the number of piconet, which the relay node have been joined. The size of the time-slot sets for the piconet P_i satisfies

$$\begin{aligned}&\text{WindowSize}(\{WS_{iR} + SO_{iR}\} \cup \{WS_{Ri} + SO_{Ri}\}\\&\quad \cup \{WS_{Rj} + SO_{Rj}\} \cup \{WS_{jR} + SO_{jR}\}) \leqslant tw_i.\end{aligned}$$

(S2) In ST_2, a relay node sends the calculated results WS_{iR}, SO_{iR}, WS_{jR}, and SO_{jR} to the master node. The master node adds $WS_{iR} + SO_{iR}$ and $WS_{jR} + SO_{jR}$ into the US set.

(S3) In ST_3, a master node broadcasts the schedule to all slave nodes, and slave nodes obey the time-slot schedule for communication.

Fig. 10 gives an example that a relay node R joins the specific three time slots of BTIM in $Piconet_1$ and $Piconet_2$. The R exchanges communication information with M_1 and M_2, and receives the schedules from M_1 and M_2. By our scheme, a relay node transmits/receives data as shown in Fig. 9(b). Comparing to existing Bluetooth protocol as shown in Fig. 9(a), our scheme actually reduces the delay time and improves scatternet throughput (see also Figs. 9 and 10).

5. Experimental results

This section investigates the performance of OCP protocol. To make a fair comparison, we have implemented OCP protocol and RR, TSL (Zhang et al., 2002) and DSA (Cordeiro

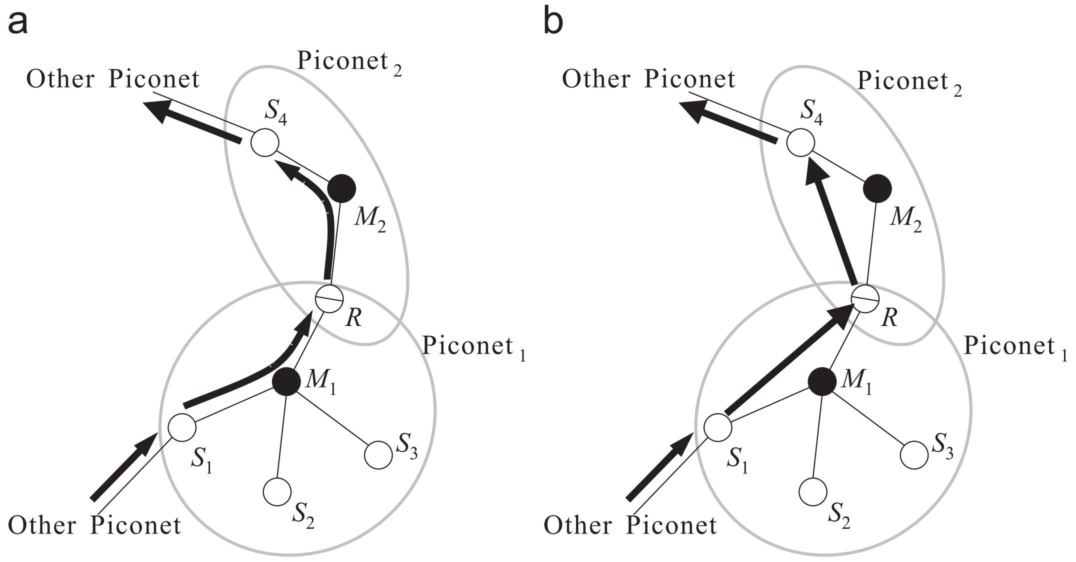


Fig. 9. An example of scatternet communication (a) by standard protocol and (b) by OCP.

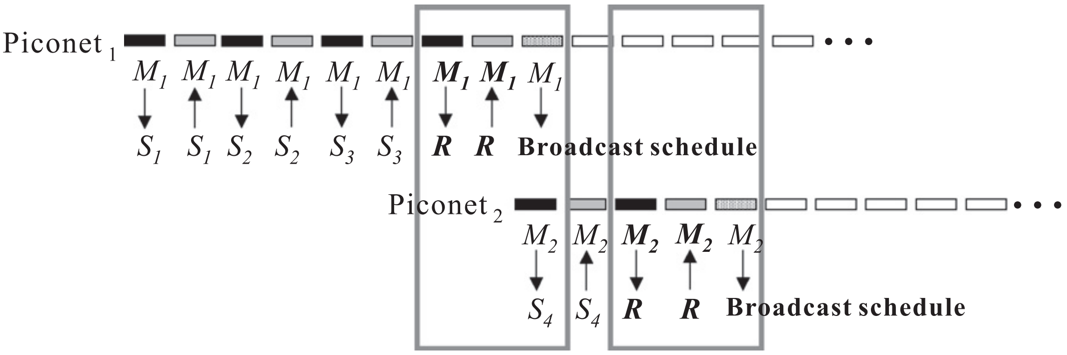

Piconet 1
M1 M1 M1 M1 M1 M1 M1 M1 M1
S1 S1 S2 S2 S3 S3 R R Broadcast schedule
Piconet 2
M2 M2 M2 M2 M2
S4 S4 R R Broadcast schedule

Fig. 10. Relay communication under scatternet.

et al., 2004) using the Network Simulator (ns-2) (VINT Project, 2007) and BlueHoc (IBM Research, 2007). We investigate the performance results which is affected by giving different parameters. The simulation is conducted in a square 2D space ($100 \times 100\,\mathrm{m}^2$) by randomly deploying N nodes ($2 \leqslant N \leqslant 70$). The radius of transmitter range of each node is 10 M. The simulated topology is constructed by randomly deployed. The performance metrics of the simulation are given below (see Table 3):

- *Transmission delay*: The latency from a source node to a destination node.
- *Average holding time*: The time period of the data transmission of the slave node which is blocked by other transmissions.
- *Throughput*: The number of data bytes received by a Bluetooth device per unit time.
- *Control overhead*: The total number of control packets under a fixed time interval.

5.1. Transmission delay

We discuss the performance of transmission delay as follows. In a piconet, as shown in Fig. 11(a) and (b), the master node adopts RR to poll slave nodes and slave nodes communicate to another slave nodes via master node. But TSL, DSA, OCP use the slave-to-slave direct communication, hence the transmission delay time from a source node to a destination node is only the half of that of RR. Besides, the key method of DSA and OCP are the TSL approach, all of DSA, OCP, and TSL have the same transmission delay time.

Observe that our scatternet communication uses the RR, so relay node transmits or receives packets by the polling mechanism of the master node. If a master node transmits packets to a relay node, but the relay node cannot immediately transmit packets to another master node, then the transmission delay time grows. In OCP, a relay node has high priority to determine which time slot should be transmitted or received. A relay node calculates its free time slots to determine the suitable time slots for transmissions and reception by OCP. When a relay node chosen the time slots, the relay node adopts the overlapping communication scheme to transmit and receive packets. All transmissions of other slave nodes do not arrest by the transmission of this relay node in a piconet. As shown in Fig. 12(a) and (b), OCP decreases the transmission delay in the scatternet communication.

Only one pair of devices communicated with each other at the same time for RR, TSL, and DSA in a piconet. When one pair of devices is communicated, all other devices hold packets and wait for the next communication. In RR, packets are relayed by the master

Table 3
The detail simulation parameters

Parameters	Value
Number of device	$2 \leqslant N \leqslant 70$
Network region	100 m × 100 m
Radio propagation range	10 m
Mobility	No
Schedule interval	64 time slots
Packet type	DH1 or DH3 or DH5

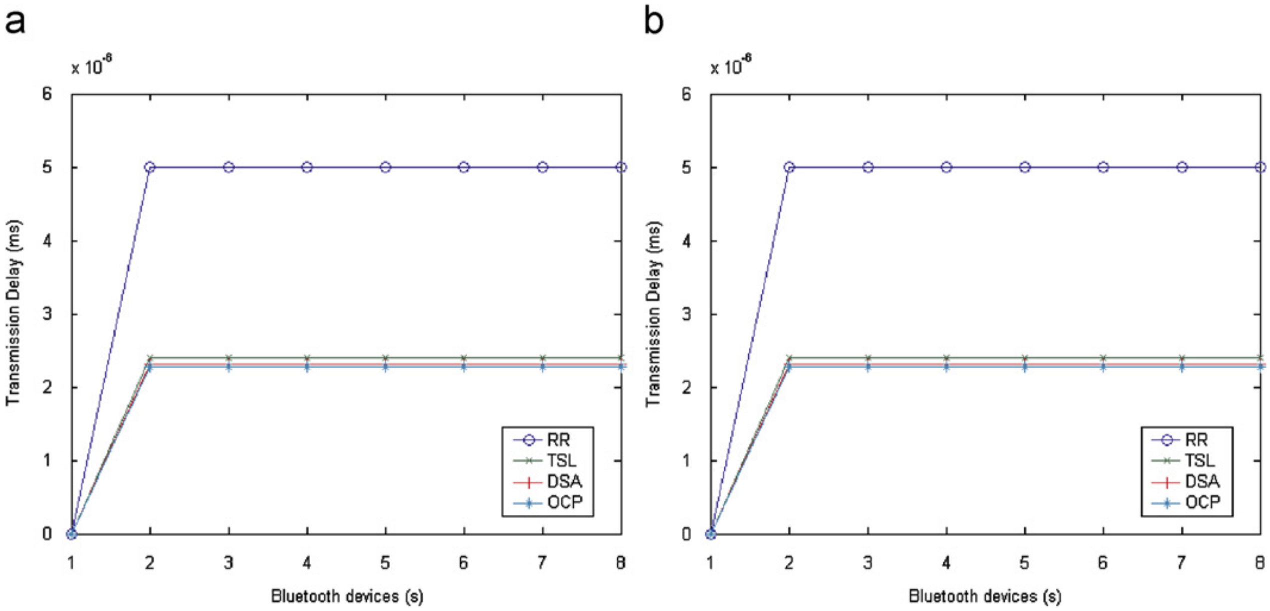


Fig. 11. Performance of transmission delay vs. Bluetooth devices in a piconet under (a) HTTP and (b) FTP.

node, and TSL sets up a slave-to-slave direct communication. The average holding time of RR is evaluated as follows. A piconet has k slave nodes. Each slave node spends U_i for upstream (a slave node to a master node) and D_i for downstream (a master node to a slave node), where $1 \leqslant i \leqslant m$.

$$\text{The average holding time of RR} = \frac{\sum_{n=1}^{m}\sum_{i=1}^{n-1}(D_i + U_i)}{m}.$$

5.2. Average holding time

This subsection describes the performance of average holding time. Fig. 13(a) and (b) shows that the average holding time of TSL is approximately half of that of RR. This is because that DSA collects several communication requests from slave nodes to assign the transmission period. Slave nodes obey the communication assignments using TSL. DSA adopts the more efficient slave-to-slave communication than TSL to significantly decrease the holding time. Observe that our OCP further uses the overlapping communication scheme. The master node collects communication requests from salve nodes to compute the transmission sequence. After the master node broadcasting the transmission sequence, slave nodes obey the scheduled time slots to transmit data packets by the slave-to-slave

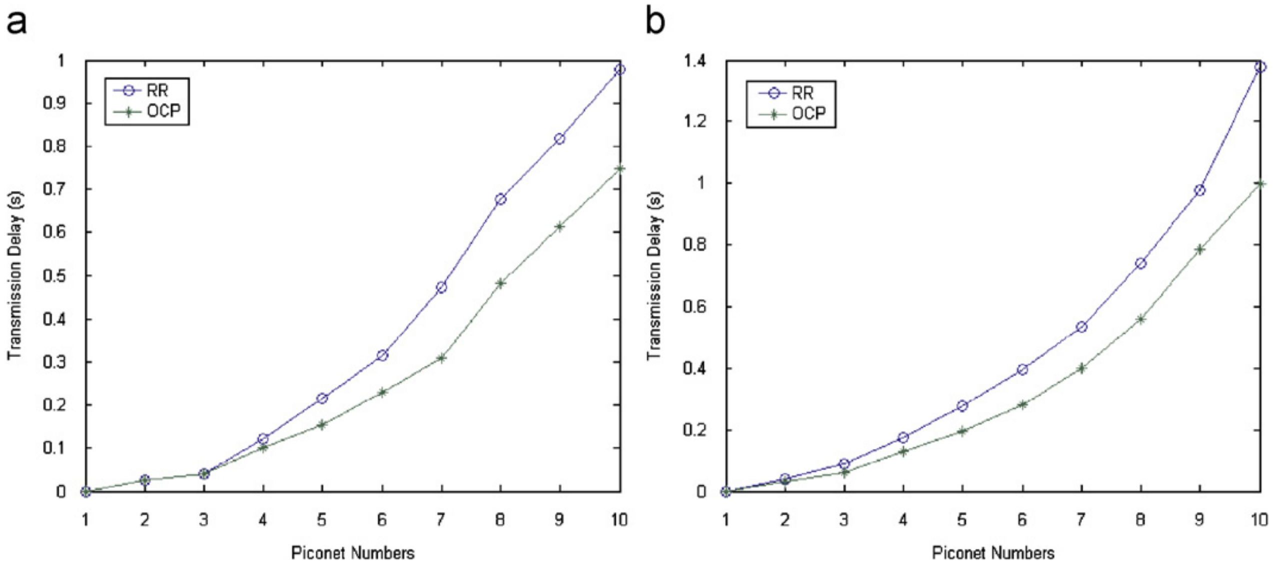


Fig. 12. Performance of transmission delay vs. piconet numbers in a scatternet under (a) HTTP and (b) FTP.

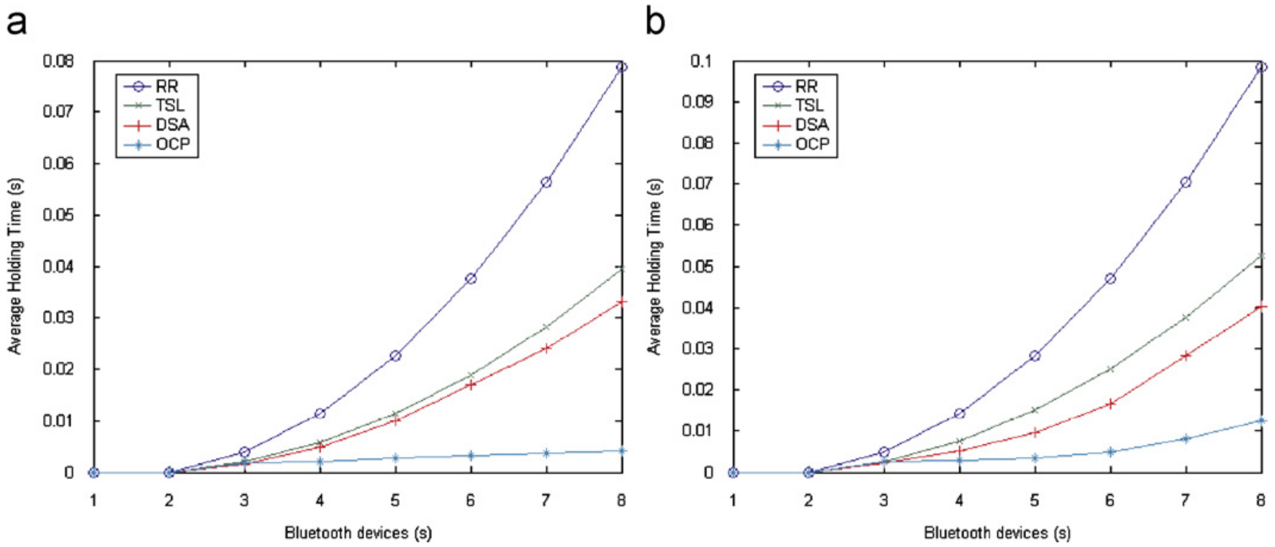


Fig. 13. Performance of average holding time vs. Bluetooth devices under (a) HTTP and (b) FTP.

direct communication. One or more communication pairs occur simultaneously and thus the holding time is significantly improved. Fig. 13(a) and (b) gives the performance results.

5.3. Throughput

We discuss the performance of throughput as follows. Given a piconet, RR keeps one transmission from one slave node to other slave node by a master-node relaying communication pattern. But, TSL adopts the slave-to-slave direct communication without master-node relaying. As illustrated in Fig. 14(a) and (b), the total throughput is more than that of RR. DSA collects several communication requests from slave nodes at one cycle and determine slave nodes to set up the slave-to-slave direct communication at a specific time period using TSL. DSA efficiently adopts the TSL. Therefore, the total throughput of DSA is more than that of TSL. OCP further adopts the overlapping communication scheme, several communication pairs occur at the same time, hence the throughput is more than that of DSA. Basically, the average throughput of OCP is about one-fourth of that of RR, as shown in Fig. 14(a) and (b). For the scatternet communication, OCP decreases the holding time because the relay node has high priority to choose the suitable time slots to transmit packet to another relay node by the overlapping communication scheme. As shown in Fig. 15(a) and (b), the total throughput of OCP is more than that of RR.

5.4. Control overhead

This subsection describes the performance of control overhead. As shown in Fig. 16(a) and (b), during each transmission cycle of RR, the master node polls a slave node and the

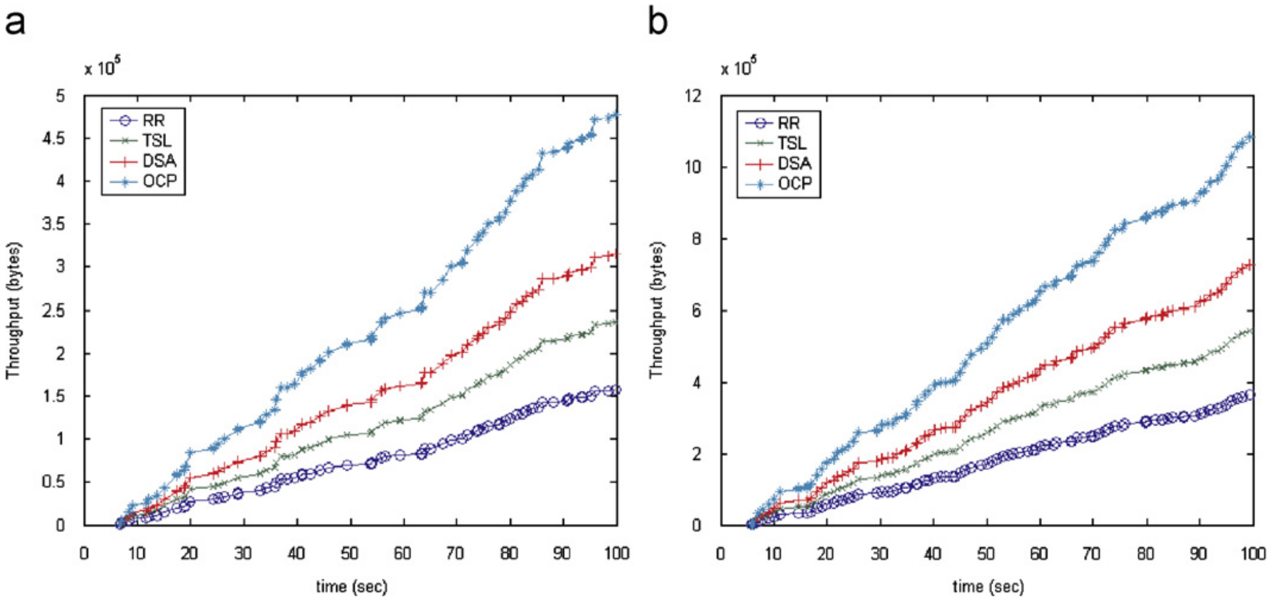


Fig. 14. Performance of throughput vs. seconds in a piconet under (a) HTTP and (b) FTP.

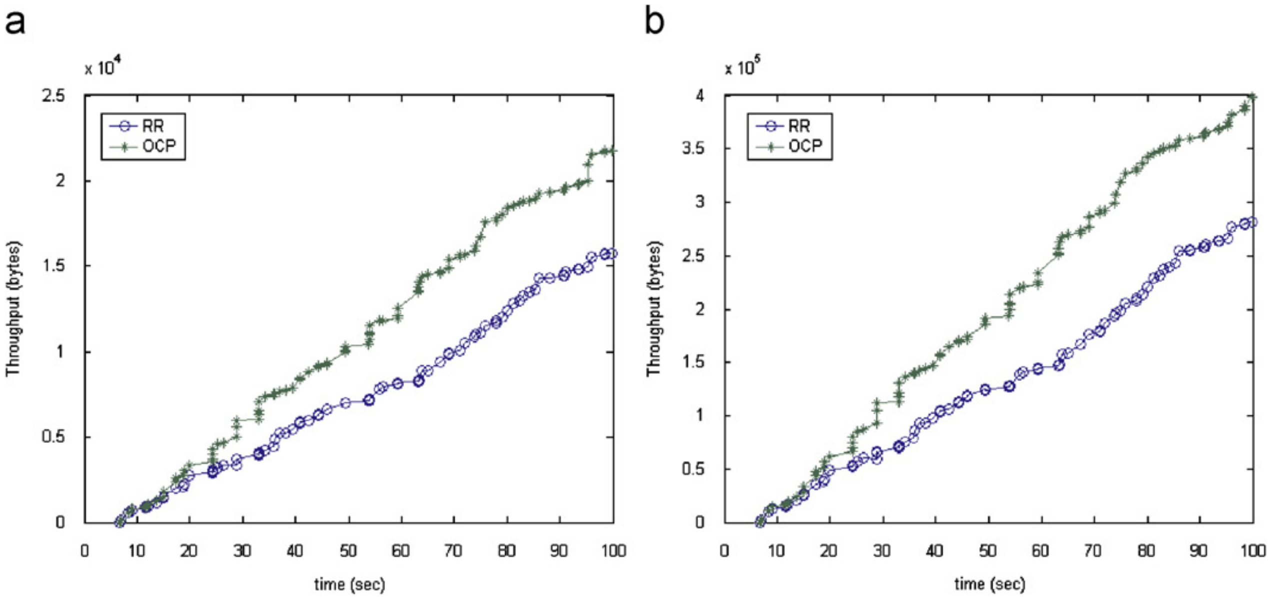


Fig. 15. Performance of throughput vs. seconds in a scatternet under (a) HTTP and (b) FTP.

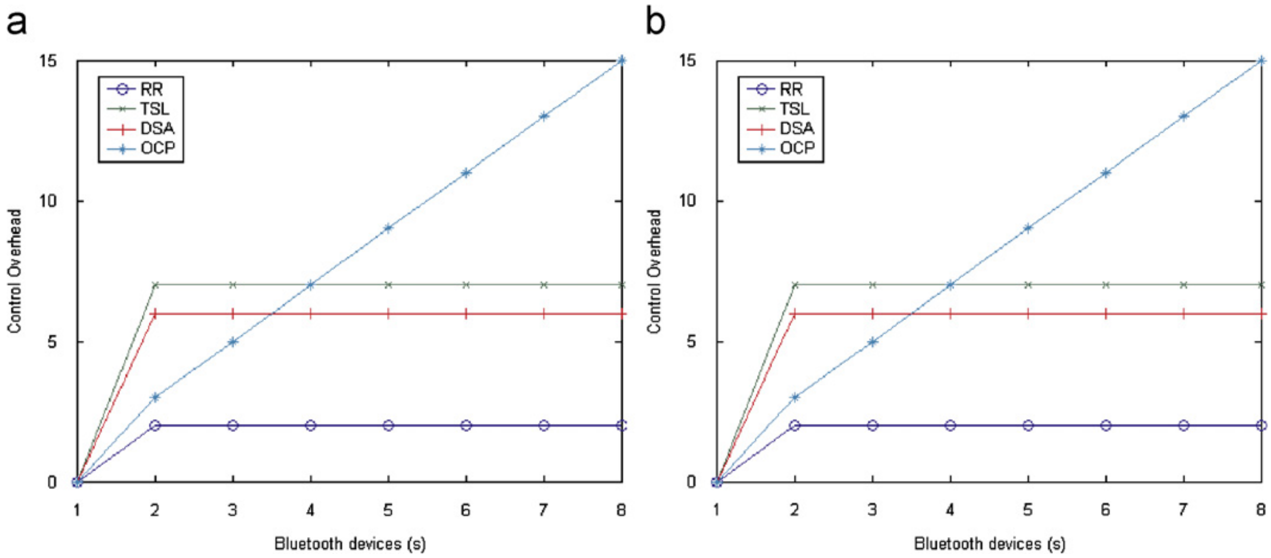


Fig. 16. Performance of control overhead vs. Bluetooth devices under (a) HTTP and (b) FTP.

slave node responses ACK. message to the master node, the control overhead is two time slots. To set up the slave-to-slave direct communication, TSL needs seven time slots to transmit necessary information when two slave nodes needed to be communicated, the control overhead is seven time slots. In DSA, the master node uses six time slots to collect slave node communication requests to assign the time period of transmissions, the control overhead is six time slots. In our OCP, the master node polls each slave node to collect communication requests to broadcast the transmission sequence, the control overhead is depended on the number of slave nodes. If a piconet with n slave nodes, the control overhead is about $2n+1$ time slots.

6. Conclusion

In this paper, we propose an efficient overlapping communication protocol to overcome the transmission holding problem and improve the performance of transmission in Bluetooth WPANs. Multiple devices can communicate simultaneously by using our proposed OCP under the same frequency hopping sequence. The OCP utilizes the improved time-slot leasing to offer the overlapping communication. With OCP, the packet delay time and the network-congestion occurrence are slight. Furthermore, OCP is very useful to improve the throughput between piconets. By simulation results, OCP is illustrated to achieve the performance improvement of bandwidth utilization and transmission delay.

References

Amin MS, Bhuyan FA. An empirical study of Bluetooth scatternet formation protocol. In: IEEE international conference on mechatronics and automation, June 2006. p. 1424–9.

Bluetooth Special Interest Group. Specification of the Bluetooth system 1.2, vol. 1, Core, March 2004 ⟨http://www.bluetooth.com⟩.

Bray J, Sturman C. Bluetooth: connect without cables. Englewood Cliffs: Prentice-Hall; 2001.

Cai W, Jin X, Zhang Y, Chen K, Tang J. Research on reliability model of large-scale wireless sensor networks. In: International conference on wireless communications, networking and mobile computing, September 2006. p. 1–4.

Capone A, Gerla M, Kapoor R. Efficient polling schemes for Bluetooth picocells. IEEE Int Conf Commun 2001;7:11–4.

Chang HY, Yu CW. Constructing long-lived scatternets in Bluetooth networks. In: Third international conference on information technology and applications, vol. 2, July 2005. p. 283–8.

Cordeiro C, Abhynkar S, Agrawal D. Design and implementation of QoS-driven dynamic slot assignment and piconet partitioning algorithms over Bluetooth WPANS. INFOCOM, April 2004. p. 27–38.

Cordeiro C, Abhyankar S, Agrawal DP. Scalable and QoS-aware dynamic slot assignment and piconet partitioning to enhance the performance of Bluetooth ad hoc networks. IEEE Trans Mobile Comput 2006; 1313–30.

Dai S, Jing X, Li L. Research and analysis on routing protocols for wireless sensor networks. In: International conference on communications, circuits and systems, vol. 1, May 2005. p. 407–11.

IBM Research. BlueHoc, IBM Bluetooth simulator. Technical Report, February 2007 ⟨http://www-124.ibm.com/developerworks/opensource/bluehoc/⟩.

Kalia M, Bansal D, Shorey R. MAC scheduling and SAR policies for Bluetooth: a master driven tdd pico-cellular wireless system. In: IEEE international workshop on mobile multimedia communications, November 1999. p. 15–7.

Kalia M, Garg S, Shorey R. Efficient policies for increasing capacity in Bluetooth: an indoor pico-cellular wireless system. In: IEEE 51st VTC, vol. 2, May 2000. p. 907–91.

Law C, Mehta AK, Siu KY. Performance of a new Bluetooth scatternet formation protocol. In: ACM symposium on mobile ad hoc networking and computing, October 2000. p. 183–92.

Lee S, Park B, Latchman H. Enhanced limited round robin with a priority policy for Bluetooth. In: The 9th international conference on advanced communication technology, vol. 1, February 2007. p. 182–5.

McDermott-Wells P. Bluetooth scatternet models. In: IEEE potentials, vol. 23(5), January 2005a. p. 36–9.

McDermott-Wells P. What is Bluetooth? In: IEEE potentials, vol. 23(5), January 2005b. p. 33–5.

Miller B, Bisdikian C. Bluetooth revealed. Englewood Cliffs, NJ: Prentice-Hall; 2001.

Proulx J, Clifford R, Sorensen S, Sorensen S, Archibald J. Development and evaluation of a Bluetooth EKG monitoring sensor. In: 19th IEEE symposium on international computer-based medical systems, June 2006. p. 507–11.

Rashid RA, Yusoff R. Bluetooth performance analysis in personal area network. In: International RF and microwave conference, September 2006. p. 393–7.

Riley GF, Zhang X. Energy-aware on-demand scatternet formation and routing for Bluetooth-based wireless sensor networks. IEEE Commun Mag 2005;43(7):126–33.

Salonidis T, Bhagwat P, Tassiulas L. Proximity awareness and fast connection establishment in Bluetooth. In: MobiHOC, 2000. p. 141–2.

The Bluetooth$^{\text{TM}}$ wireless technology white paper in white paper. Product Plans, Early Access Program.

VINT Project. Network simulator version 2 (ns2). Technical Report, June 2007 ⟨http://www.isi.edu/nsnam/ns⟩.

Ye W, Heidemann J, Estrin D. Duty cycle criterion for target packet delay of S-MAC protocol. IEEE TENCON, November 2006. p. 1–4.

Yu GJ, Chang CY, Shih KP, Lee SC. Relay reduction and disjoint route construction for scatternet over Bluetooth radio systems. J Network Comput Appl 2007;30:728–49.

Zhang W, Zhu H,. Cao G. On improving the performance of Bluetooth networks through dynamic role management. Technical Report, CSE-01-018, Pennsylvania State University, May 2001 ⟨http://www.cse.psu.edu/gcao/paper/bluetooth.ps⟩.

Zhang W, Zhu H, Cao G. Improving Bluetooth network performance through a time-slot leasing approach. IEEE wireless communications and networking conference, March 2002. p. 592–6.