An Overlapping Communication Protocol Using Improved Time-Slot Leasing for Bluetooth WPANs

Yuh-Shyan Chen[1], Yun-Wei Lin[1], and Chih-Yung Chang[2]

[1] Department of Computer Science and Information Engineering, National Chung Cheng University, Taiwan
[2] Department of Computer Science and Information Engineering, Tamkang University, Taiwan

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 scheme. The key contribution of our improved time-slot leasing scheme additionally offers the overlapping communication capability and 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, wireless communication.

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 equipments for constructing an ubiquitous communication environment. Bluetooth[2] is a low cost, low power, and short range communication technology that operates at 2.4GHz ISM bands.

A master polls slaves by sent polling packets to slaves using round robin (RR) scheme within the piconet. The master communicates with one slave and all other slaves must hold and wait the polling packet, so the transmission of other slaces is arrested. This condition is called the "transmission holding problem." To reduce the "transmission holding problem", one interest issue is how to develop a novel scheme which can effectively solve the "transmission holding" problem under the

X. Jia, J. Wu, and Y. He (Eds.): MSN 2005, LNCS 3794, pp. 1–10, 2005.

fixed-topology situation. To satisify that purpose, Zhang *et al.* develops a time-slot leasing scheme (TSL) [5]. Zhang *et al.*'s time-slot leasing scheme provides a general mechanism to support the direct slave-slave communication, but the master node uses round-robin (RR) mechanism to check slave node intended to send or receive data. Other slave node waits the polling time and gains the transmission holding time. More recently, Cordeiro *et al.* proposed a QoS-driven dyamic slot assignment (DSA) schedule scheme [1] to more efficiently utilize time-slot leasing scheme. With TSL and DSA scheme, the tramnsission holding problem still exists. Effort will be made to effectively reduce the the tramnsission holding problem under the fixed topology structure.

In this paper, we propose an overlapping communication protocol using improved time-slot leasing in the Bluetooth WPANS. The overlapping communication scheme is based on the improved time-slot leasing scheme which modified from the original time-slot leasing scheme, while the original time-slot leasing scheme only provides the slave-to-slave communication capability. The key contribution of our improved time-slot leasing scheme additionally offers the overlapping communication capability and 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 Section 3. The performance analysis is discussed in Section 4. Section 5 concludes this work.

2 Basic Idea

The transmission holding problem is originated from the drawback of the master/slave model. In a piconet, since the slave may transmit packets only if it receives the polling packet from master. As a result, when there are many salves have to transmit data, slaves must hold its transmission until receving the polling packet, as shown in Fig. 1(a). To solve the transmission holding problem, a time-slot leasing (TSL) approach [5] has been proposed. Slaves can directly transmit packets to each others without the master relaying. Using TSL approach, the waiting time of the other holding slaves are reduced. Unfortunately, the effect of transmission holding problem is reduced, but it still exists. To slove 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 time-slot leasing (TSL) scheme [5] and QoS-driven dyamic slot assignment (DSA) scheme [1]. The frequency-hopping spread spectrum is used in the Bluetooth network. Let $C(x)$ denote the used channel for time slot x, and $\overrightarrow{\alpha\beta}$ denote slave node α sends data to slave node β. An example is shown in Fig. 1(b), $\overrightarrow{S_1S_2}$,

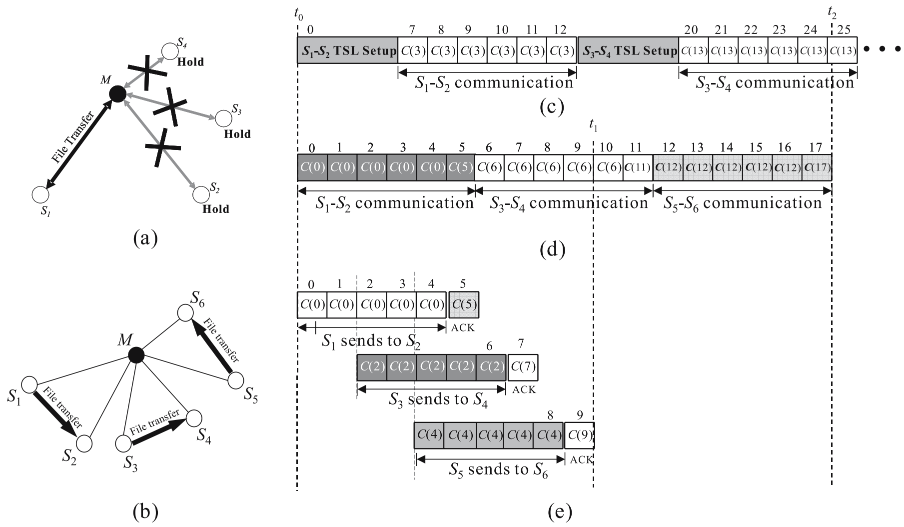


Fig. 1. (a) Transmission holding problem (b) Three communication requests (c) TSL (d) DSA (e) Overlapping communication protocol

$\overrightarrow{S_3S_4}$, and $\overrightarrow{S_5S_6}$ simultaneously occur in a piconet. The transmission holding problem is heavily occurred in master node M. Fig. 1(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. 1(d). However, the time cost will be improved by using our overlapping communication scheme. Fig. 1(e) illustrates that the works of $\overrightarrow{S_1S_2}$, $\overrightarrow{S_3S_4}$, and $\overrightarrow{S_5S_6}$ can be accomplished in time $t_1 - t_0$.

To explain 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. From the sepcification of the Bluetooth system 1.2 [2], the consecutive five time slots keep the same channel if using a DH5 packet. The rule is same for packets DH3, DM3, and DM5. Let channel $C(x)$ denote a Bluetooth 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. Efforts will be 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 Figs. 2(a)(b). The data transmission of $\overleftrightarrow{M_1S_3}$

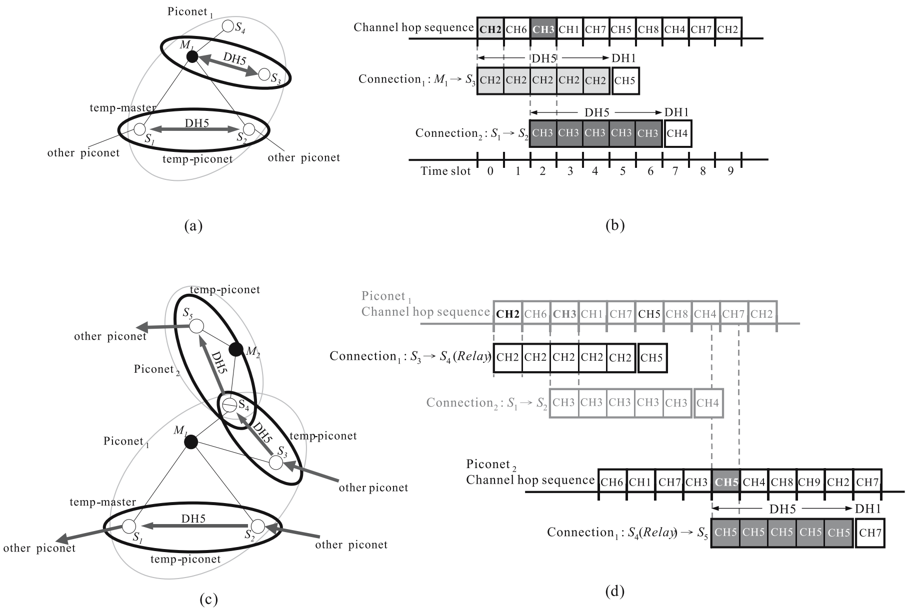


Fig. 2. The concept of overlapping communication scheme

and $\overleftrightarrow{S_1S_2}$ can be overlapped. In addition, Fig. 2(c)(d) illustrates the overlapping condition for performing the intra-piconet overlapping communication protocol.

3 Overlapping Communication Protocol

To significantly overcome the "transmission holding problem", an overlapping communication protocol is presented. The overlapping communication protocol is divided into two phases; (1) *queuing scheduling*, (2) *overlapping time-slot assignment*. The details describe in the following.

3.1 Queuing Scheduling Phase

The queuing scheduling phase is to achieve the overlapping communication schedule. To complete the overlapping communication, master initially forms *data-flow matrix* and *queuing table*. This work can be done in the *BTIM* window as shown in Fig. 3(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* $DM_{m\times m+1} = \begin{bmatrix} D_{10} & ... & D_{1m} \\ \vdots & D_{ij} & \vdots \\ D_{m0} & ... & D_{mm} \end{bmatrix}$, where D_{ij} denote the amount of data transmission from node i to node j, where

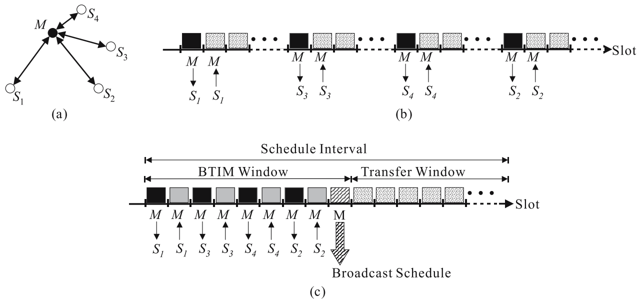
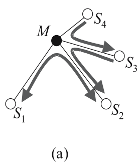


Fig. 3. (a) Master polling (b) Polling and transmission in origin piconet (c) The structure of BTIM



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

(b)

Queuing Table

Queue 1	$\overleftrightarrow{S_1S_2}$	$\overleftrightarrow{S_3S_2}$
Queue 2	$\overleftrightarrow{S_4S_3}$	

(c)

Fig. 4. (a) An example of piconet communication, (b) flow matrix, and (c) n-queue

$1 \leq i \leq m$, $0 \leq j \leq m$, and m is the number of slave nodes in a piconet. Observe that node j is the master node if $j = 0$. Example is shown in Fig. 4(b).

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 $\overleftrightarrow{i_0j}, ..., \overleftrightarrow{i_kj}, \overleftrightarrow{i_{k+1}j}, ...,$ and $\overleftrightarrow{i_mj}$ in a piconet, where all of these links have the same destination node j. Further, let $Q = \{\overleftrightarrow{i_0j}, \cdots, \overleftrightarrow{i_kj}, \overleftrightarrow{i_{k+1}j}, \cdots, \overleftrightarrow{i_mj}\}$, where $pri(\overleftrightarrow{i_kj}) > pri(\overleftrightarrow{i_{k+1}j})$ and $0 \leq k \leq m-1$. For instance, $\{\overleftrightarrow{S_1S_2}, \overleftrightarrow{S_3S_2}\}$.

Assumed that there are n link queues Q_1, $Q_2, \cdots, Q_q, \cdots, Q_n$, where each Q_q has differnt destination node, and $1 \le q \le n$, 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 $Q_q = \{\overleftrightarrow{i_0j_q}, \cdots, \overleftrightarrow{i_kj_q}, \overleftrightarrow{i_{k+1}j_q}, \cdots, \overleftrightarrow{i_mj_q}\}$, we denote $MAX(Q_q) = \underset{k=0}{\overset{m}{MAX}}\ pri(\overleftrightarrow{i_kj_q}) = pri(\overleftrightarrow{i_0j_q})$. All of Q_1, $Q_2, \cdots, Q_q, \cdots, Q_n$ can be combined into a 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 $1 \le q \le n$. Example is shown in Fig. 4(a).

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 $q \le k \le n$. For higher bandwidth utilization, DH5 packet type is the prior choice to transmit data. Overlapping communication scheme uses 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 i to node j, and *Excess* denote the excess part of transmission. For each transmission, master arranges approoriate packet type by a packet type distribution rule which describes as following:

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

The packet amount of DH5 and DH3 is 339 bytes and 183 bytes, respectively. A set of packet type distribution $PD_{ij} = \{pd_{ij_0}, pd_{ij_1}, ..., pd_{ij_s}, ..., pd_{ij_p}\}$ is used to record $p+1$ packet type distribution from node i transmitting to node j, where pd_{ij_s} denotes the t^{th} transmission packet type form i to j and $0 \le s \le p$. 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. 4(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 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 $\overleftrightarrow{ij}$.

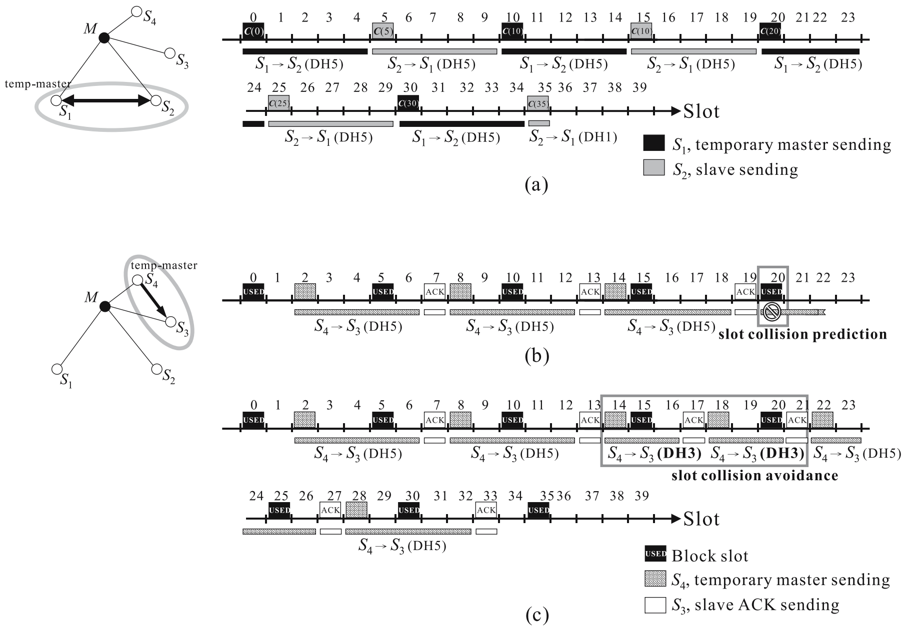


Fig. 5. (a) the block slot calculation, (b) collision occurred, (c) packet type changing

According the packet type distribution, the overlapping time-slot assignment phase assigns suitable time-slot to use improved time-slot leasing 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 $SO_{ij} = \{so_{ij_0}, so_{ij_1}, ..., so_{ij_t}, ..., so_{ij_q}\}$ is defined to record $q+1$ time-slot offset from node i transmitting to node j, and $0 \leq t \leq q$. The so_{ij_t} denotes the t^{th} time-slot offset transmission from i to j. SO_{ij} is accumulated form PD_{ij} and PD_{ji}, which is used to predict the occupied time-slot offset by link $\overleftrightarrow{ij}$. For using improve time-slot leasing, master announces each slave WS_{ij} and SO_{ij}. Slaves wake up at assigned slot WS_{ij} and obey the set of time-slot offset SO_{ij} to transmit data.

About overlapping communication scheme, master assigns the roles, WS_{ij}, and SO_{ij} to each transmission $\overleftrightarrow{ij}$ at BTIM. We consider two roles, temp-master and slave and if $D_{ij} > D_{ji}$ then i is temp-master and j is slave. For a transmission from temp-master i to slave j, master assigns time-slot to link $\overleftrightarrow{ij}$ as follows.

$$SO_{ji} : so_{ji_t} = so_{ij_t} + pd_{ij_s}$$
$$SO_{ij} : so_{ij_{t+1}} = so_{ji_t} + pd_{ji_s},$$

where the so_{ij_0} initially sets to 0. During computing SO_{ij}, if master runs out of the PD_{ij} set, $pd_{ji_{p+t}}$ sets to 1 for ACK responding. Master 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 4(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. 5(a).

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

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

If the equation is ture, the problem of time-slot collision is detected. For example as shown in Fig. 5(b), master checks $(WS_{S_4S_3} + SO_{S_4S_3}) \bigcap BS = \{20\}$. 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-slots is collision, master adapts the following rules to avoid the collision.

S1: /* Avoidance for DH5 packet */
If the t^{th} collision packet type is DH5, master change the packet to two DH3 packets. Master changes the $pd_{ij_t} = \{5\}$ to $pd_{ij_t} = \{3, 3\}$.

S2: /* Avoidance for DH3 packet */
If the t^{th} collision packet type is DH3, master change the packet to one DH5 packet. Master changes the $pd_{ij_t} = \{3\}$ to $pd_{ij_t} = \{5\}$.

S3: /* Avoidance for DH1 packet */
If the t^{th} collision packet type is DH1, master change the packet to one DH3 packet. Master changes the $pd_{ij_t} = \{1\}$ to $pd_{ij_t} = \{3\}$.

After master applying above rules to the collision condition of $\overrightarrow{S_4S_3}$, the transmission process has shown in Fig. 5(c). After the schedule arrangement, master informs all slaves by a schedule-assignment packet. If a device does not do anything, it changes to sleep mode to save energy until next BTIM.

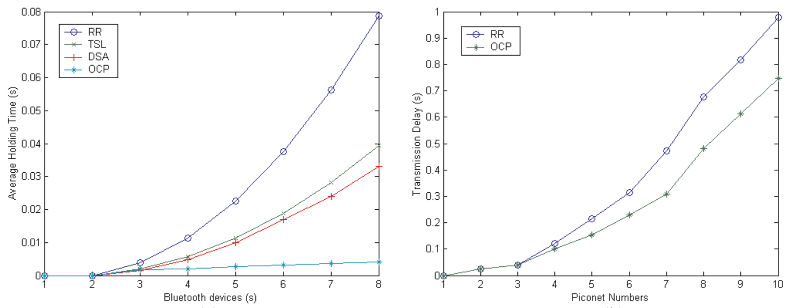
4 Experimental Results

In simulation, we investigate the performance of OCP protocol. To compare with three algorithms, RR, TSL [5] and DSA [1], we have implemnted OCP protocol and others using the Network Simulator (ns-2) [3] and BlueHoc [4]. The performance metrics of the simulation are given below. The simulation parameters are shown in table 1.

- *Average Holding Time*: the time period of packet is arrested by other transmission.
- *Transmission Delay*: the latency from a souurce device to a destination device.
- *Throughput*: the number of data bytes received by all Bluetooth devices per unit time.

Table 1. The detail simulation parameters

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



(a) (b)

Fig. 6. Performance of comparison about (a) average of holding time vs. Bluetooth devices (b) transmission delay vs. piconet numbers

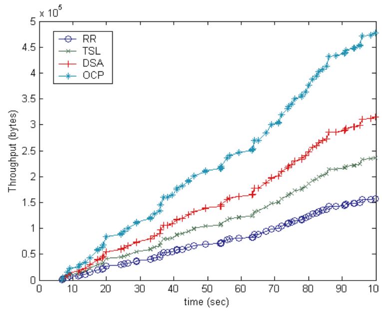
4.1 Performance of Average Holding Time and Transmission Delay

In a piconet, Fig. 6(a) indicates that the holding time of TSL is half of RR approximately. DSA uses TSL more efficiently and decreases the holding time more than TSL. OCP uses the overlapping communication scheme. Therefore, there are several communication pairs transmission simultaneously and the average holding time is almost reduced to two slots.

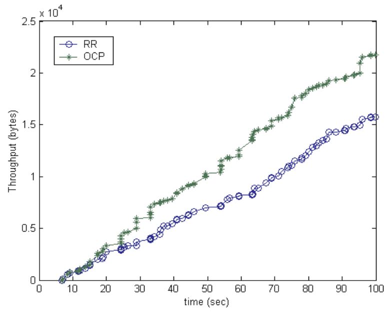
In scatternet communication using RR, relay can transmit or receive data only by master polling. If master routes packets to relay and relay can not transmit packets to another master immediately, the transmission delay time grows. In OCP, a relay can have more high priority to decide the transmission and receive time. As shown in Fig. 6(b), OCP can decrease the transmission delay in scatternet communication.

4.2 Performance of Throughput

In a piconet, RR keeps one slave transmits data to another slave by master relay. TSL adopts the slave-to-slave direct communication without master relay. As shown in Fig. 7(a), by the time passing, the total throughput is more than RR. DSA uses the TSL more efficiently. Therefore, the total throughput is more than TSL. OCP uses the overlapping communication scheme, there are several communication pairs transmitting at the same time, hence the throughput is more than DSA.



(a)



(b)

Fig. 7. Performance of throughput vs. seconds in (a) a piconet (b) a scatternet

In a scatternet, OCP decreases the holding time because the relay has more high priority to decide the transmission time and transmits to another relay directly by overlapping communication scheme. As shown in Fig. 7(b), the total throughput of OCP is more than RR in scatternet communication.

5 Conclusion

In this paper, we have proposed an efficient overlaping communication protocol to address the transmission holding problem. We realize OCP by using improved time-slot leasing. By OCP, devices can communicate simultaneously under the same frequence hopping sequence, and there are less packet delay time and low probability of network congestion. In scatternet, OCP is able to advance the data packet transmission between piconets. OCP has improvement in bandwidth utilization and transmission delay.

References

1. C. Cordeiro, S. Abhynkar, and D. Agrawal. "Design and Implementation of QoS-driven Dyamic Slot Assignment and Piconet Partitioning Algorithms over Bluetooth WPANS". *INFOCOM 2004 Global Telecommunications Conference*, pages 27–38, April 2004.
2. Bluetooth Special Interest Group. *"Specification of the Bluetooth System 1.2"*, volume 1: Core. http://www.bluetooth.com, March 2004.
3. VINT Project. "Network Simulator version 2 (ns2)". Technical report, `http://www.isi.edu/nsnam/ns` , June 2001.
4. IBM research. "BlueHoc, IBM Bluetooth Simulator". Technical report, http://www-124.ibm.com/developerworks/opensource/bluehoc/, February 2001.
5. W. Zhang, H. Zhu, and G. Cao. "Improving Bluetooth Network Performance Through A Time-Slot Leasing Approach". *IEEE Wireless Communications and Networking Conference (WCNC'02)*, pages 592–596, March 2002.