Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computer Communications 29 (2006) 2084–2095

computer communications

www.elsevier.com/locate/comcom

An adaptive bluetooth packet selection and scheduling scheme in interference environments

Chen-Han Shih, Kuochen Wang [*,1], Hung-Cheng Shih

Department of Computer and Information Science, National Chiao Tung University, Hsinchu 300, Taiwan

Received 27 April 2005; received in revised form 23 December 2005; accepted 17 January 2006
Available online 17 February 2006

Abstract

Bluetooth is a new technology for Wireless Personal Area Networks (WPANs). It intends to eliminate the need of wires and connectors between a variety of devices, like PCs and their peripherals, walkmans and their earphones, etc. Bluetooth provides robust and secure wireless radio communication of both data and voice, even when the devices are not within line-of-sight. Bluetooth employs the 2.4 GHz ISM band, sharing the same band with the Wireless LAN (WLAN) implementing the IEEE 802.11 series standard. While WLANs and WPANs are complementary rather than competing technologies, the likelihood of mutual interference may occur unexpectedly, which may impact the performance of either severely. In this paper, we propose a Bluetooth *channel state dependent data segmentation and reassembly* (CSD-SAR) scheme and a *queue state dependent priority* (QSD-PR) scheduling policy. The CSD-SAR maintains a receiving frequency table to predict channel conditions and selects the best packet type and packet size to transmit data. In this way, it not only masks bad frequencies without delaying transmission but also leads to the best performance with high link utilization in error-prone environments. In addition, the QSD-PR also uses the receiving frequency table to avoid bad frequencies and gives a selected master–slave pair, which has more queued data to send between each other, a higher priority to eliminate the wastage of slots. The conventional scheduling policy, *Round Robin* (RR), yields poor performance with the *time division duplex* based MAC protocol and results in slot wastage and may not ensure fairness. Simulation results show that our proposed scheme achieves better link utilization and higher throughput with bounded delay compared to the RR scheme in error-free and error-prone environments. Our scheme can also eliminate interference to other wireless networks that share the same spectrum, such as WLANs, by avoiding selecting channels occupied by other networks.

Keywords: Bluetooth; Adaptive packet selection; Channel state dependent packet scheduling; Interference environment

1. Introduction

With the need for new mobility arises, the devices of Wireless Personal Area Networks (WPANs) and Wireless Local Area Networks (WLANs) will increase in a rapid pace. The WPAN category is led by a short-range radio technology called Bluetooth [1–4], which was designed primarily for cable replacement applications. The WLAN category has several technologies competing for dominance, like IEEE 802.11a/b/g [5], HomeRF [6], HiperLAN/2 [7], etc. Bluetooth and existing Wireless LANs (without loss of generality, only IEEE 802.11b is discussed in this paper owing to its popularity) have a number of distinctive features. Bluetooth uses the Frequency Hopping Spread Spectrum (FHSS) scheme and hops over 79 1-MHz-wide channels by 1600 times per second while IEEE 802.11b uses the Direct Sequence Spread Spectrum (DSSS) scheme and occupies one 22-MHz-wide static channel across the acceptable 83.5 MHz of the 2.4 GHz ISM band. Bluetooth was designed for personal area networking that transmits

[*] Corresponding author. Tel.: +886 3 5712121x56613; fax: +886 3 5721490.
E-mail address: kwang@cis.nctu.edu.tw (K. Wang).

[1] This work was supported by the National Science Council under Grant NSC90-2213-E-009-162 and the NCTU EECS-MediaTek Research Center under Grant Q583.

0140-3664/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2006.01.004

at power level of about 1 mW and IEEE 802.11b was designed for wireless local area networking with power level from 30 to 100 mW.

Both WPANs and WLANs share the same 2.4 GHz unlicensed frequency band and provide complimentary wireless solutions for connectivity. This complimentary nature of the services could enhance the use of both protocols at the same physical location and provide an incentive for their adoption. Recently, the issue of designing coexistence mechanisms between WLANs and WPANs has received much attention because both may suffer strong interference from each other [8,9]. Some interference reduction techniques such as power control adjustments [10], channel state dependent error avoidance schemes [1,11,12], collaborative schemes, and adaptive frequency hopping [2,13] were proposed. A scheduling algorithm was proposed in [11] that used a Frequency Usage Table to distribute channels to devices and ensures fairness of access among users by means of max–min fairness criteria. In [12], a Link State History based scheme was proposed to achieve high accuracy in identifying the good and bad periods of the channels. However, the packet selection scheme in Bluetooth also has a significant effect on data scheduling and network performance. It controls the distribution of packet types and packet sizes that may result in different probabilities of packet loss. For this reason, we propose a channel state dependent packet selection scheme and a simple priority scheduling policy that takes queue states and channel conditions into account to maximize link utilization while ensuring a high throughput and low packet error rate in interference environments. In addition, the simulation models in [11,12] are restricted to the link layer and are not optimized for transport layer sessions. We adopted a simulation model to include not only the core Bluetooth protocol layers but also TCP/IP.

This paper is organized as follows. Section 2 gives general insights on the Bluetooth technology. Section 3 presents our packet selection and scheduling scheme. Section 4 shows simulation scenarios and simulation results of our proposed scheme and the performance is then evaluated. Finally, concluding remarks are presented in Section 5.

2. Overview of the Bluetooth system

Bluetooth was designed with the objective of small size, low power consumption, and low cost. Bluetooth has a range of 10 m and provides a nominal data rate of 1 Mbit/s for wireless communications in a small area network. Two or more Bluetooth devices communicating on the same channel form a piconet [14], where one device operates as a master (generally means the unit that establishes the piconet) and the others act as the slaves. Up to seven slaves can be active in the piconet and the master is always responsible for defining and synchronizing the frequency hop pattern of the piconet.

2.1. Medium access control in Bluetooth

As shown in Fig. 1, the Time Division Duplex (TDD) scheme is used in the Bluetooth for resolving contention over wireless links. The master device controls data transmission through a polling procedure: periodically polls slave devices for information and only after receiving such a poll a slave is allowed to transmit. Thus, it is the master that determines which slave is scheduled when and how often. The channel is divided into time slots, each 625 μs in length. The master is required to always start transmission on an even numbered slot while a specific slave on an odd numbered slot. The time slots, where each slot corresponds to an RF hop frequency, are numbered according to the Bluetooth clock of the piconet master. It should be noted that the Bluetooth clock has no relation to the time of day. Since transmission and reception take place at different time slots, transmission and reception also take place at different hop carriers. In order to support asymmetric links, devices have the option of transmitting a single packet lasting as much as five slots. The center frequency used for each packet does not change until that packet has ended, regardless of the number of slots the packet occupies and depends on the frequency at the time when the master begins sending the packet.

2.2. Packet-based communications

The Bluetooth system uses packet-based transmission: the information stream is fragmented into packets. Packets can reserve one, three or five consecutive time slots for transmission. The standard packet format is shown in Fig. 2. Each packet has the same format, starting with an access code, followed by a packet header, and ending with the payload. The access code (72-bits) is used for synchronization and identifying packets in a piconet. The packet header consists of 18 bits and is encoded with a rate 1/3

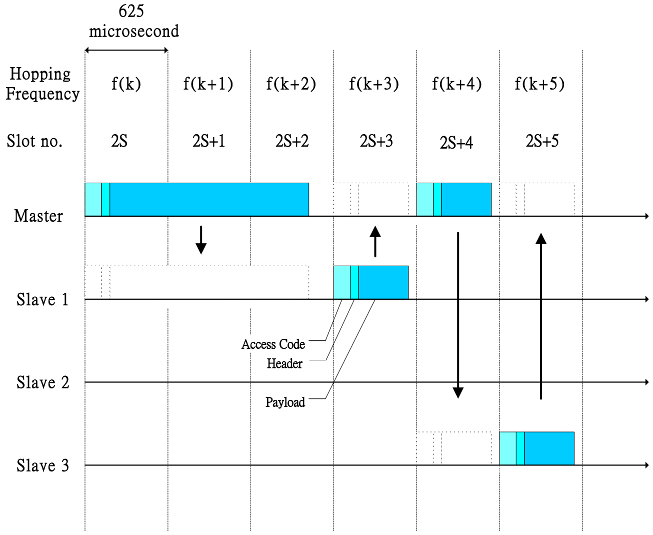


Fig. 1. Bluetooth channel structure.

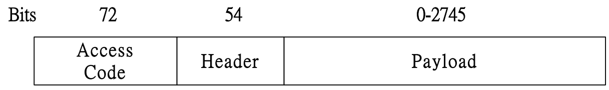


Fig. 2. Bluetooth standard packet format.

Forward Error Correction (FEC) resulting in a 54-bit header. All packets sent in the same piconet are preceded by the same channel access code.

Bluetooth links support both synchronous services such as voice traffic and asynchronous services such as bursty data traffic. There are two types of physical links that can be established between the master and a slave [4]: the *Synchronous Connection-Oriented* (SCO) link and *Asynchronous Connectionless* (ACL) link. The SCO link is designed to support real-time isochronous applications. It is a point-to-point link between the master and a specific slave. The link is established by reservation of duplex slots at regular intervals without being polled. The ACL link is used to exchange data in non-time-critical applications. It is a point-to-multipoint link between the master and all slaves on the piconet and can use all of the remaining slots on the channel not used for the SCO link. The traffic over the ACL link is scheduled by the master with the polling mechanism.

In this paper, the packet selection and scheduling only applies at the ACL link. Unlike SCO packets with a fixed payload length 240 bits and no payload header, the ACL packets have three segments in the payload: a payload header, a payload body, and possibly a CRC code, as shown in the Fig. 3. The payload header specifies the logical channel (2-bit L_CH indication), controls the flow on the logical channels (1-bit FLOW indication), and has a payload length indicator (5-bit LENGTH indication for single time-slot packets, 9-bit LENGTH indication for multi-slot packets) [4].

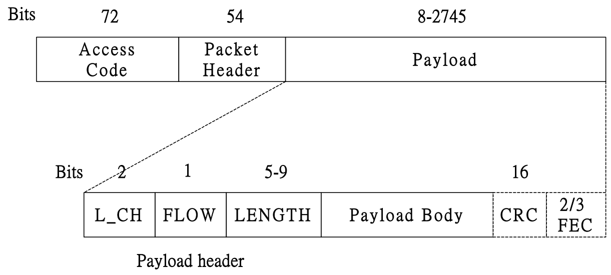


Fig. 3. ACL packet format.

The ACL packets can be classified into two categories. (1) *DH packets* stand for Data-High rate packets and do not incorporate FEC code, and (2) *DM packets* stand for Data-Medium rate packets and are protected with 2/3 FEC code to resist interference. That is, unlike DM packets, DH packets are not protected by the FEC code. The only error recovery used by DH packets is error detection through a 16-bit CRC combined with the Automatic Repeat Request (ARQ) scheme. The packet types of ACL packets are summarized in Table 1 [4]. The user payload represents the packet payload excluding FEC, CRC, and payload header.

2.3. Bluetooth protocol stack

Fig. 4 shows the Bluetooth protocol stack [15]. The Bluetooth Baseband enables adjacent Bluetooth units to form a piconet. Bluetooth provides two different kinds of physical links (SCO link and ACL link) with their corresponding baseband packets. Note that one of the basic limitations of the Bluetooth Baseband protocol is that the packets that make up its transport service are size-limited. The Bluetooth Logical Link Control and Adaptation Protocol (L2CAP) layer adapts upper layer protocols over the Baseband and provides Segmentation and Reassembly

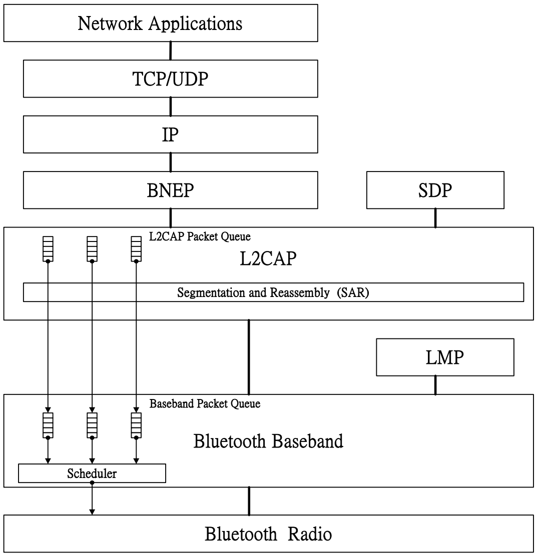


Fig. 4. Bluetooth protocol stack.

Table 1
Summary of ACL packets

Type	Payload header (bytes)	User payload (bytes)	FEC	CRC	Symmetric maximum rate (kbps)	Asymmetric maximum rate (kbps)	
						Forward	Reverse
DM1	1	0–17	2/3	Yes	108.8	108.8	108.8
DH1	1	0–27	No	Yes	172.8	172.8	172.8
DM3	3	0–121	2/3	Yes	258.1	387.2	54.4
DH3	3	0–183	No	Yes	390.4	585.6	86.4
DM5	5	0–224	2/3	Yes	286.7	477.8	36.3
DH5	5	0–339	No	Yes	433.9	723.2	57.6

(SAR) operations to improve efficiency by supporting a maximum transmission unit (MTU) size larger than the largest baseband packet. The L2CAP permits higher-level protocols and applications to transmit and receive L2CAP data packets up to 64 kB in length. This reduces overhead by spreading the network and transport packets used by higher layer protocols over several baseband packets. The primary data buffers in Bluetooth are at the L2CAP and at the Bluetooth Baseband. When the L2CAP fragments L2CAP packets into baseband packets, there is a separate ACL buffer for each slave at the master, and the scheduler decides which packet to send and how often.

The Link Manager Protocol (LMP) in Fig. 4 is responsible for link-setup between Bluetooth devices. Furthermore, it controls the power modes and the connection states of a Bluetooth unit in a piconet. Discovery services are a crucial part of the Bluetooth framework. Using the Service Discovery Protocol (SDP), device information, services and the characteristics of the services can be queries and after that, a connection between two or more Bluetooth devices can be established [15].

Note that the Bluetooth Network Encapsulation Protocol (BNEP) [16] can encapsulate packets from various networking protocols, which are transported directly over the Bluetooth L2CAP protocol. The BNEP is used primarily in the Bluetooth Personal Area Networking Profile [17] to provide networking capabilities for Bluetooth devices.

3. Proposed packet selection and scheduling scheme

3.1. Basic idea

Using different packet types with different lengths and error protection properties results in different packet error rates in the same channel status. In an error-free environment, the DH5 packet would give the best performance [18] since it carries the most information bits per unit time. However, as the bit error rate increases, the resulting network performance will depend on the degree of forward error correction (FEC) and packet length [19–21]. The packet error rate (PER) of different ACL DH packet types can be expressed in terms of the bit error rate (BER) (assume the event of a bit error is independent of others):

$$\mathrm{PER(X)} = 1 - (1 - \mathrm{BER})^m, \tag{1}$$

where BER is the current bit error rate and m is the number of payload bits in packet type X, $m = 240$ for DH1, $m = 1496$ for DH3, and $m = 2744$ for DH5.

The payload of DM packets is protected by a (15, 10) Hamming code, which is capable of correcting one bit error per 15 bits code block. Similarly, we can also estimate the PER of DM packets from the BER as follows:

$$\mathrm{PER(X)} = 1 - ((1 - \mathrm{BER})^{15} + 15 \times \mathrm{BER} \times (1 - \mathrm{BER})^{14})^M \tag{2}$$

where $M = 16$ for DM1, $M = 100$ for DM3, and $M = 183$ for DM5.

Fig. 5 plots the PER of different ACL packet types (include DH and DM packets) with respect to the uniform BER based on Eqs. (1) and (2). However, as shown in the figure, when the BER increases from 10^{-4} to 10^{-3}, the PER of DH packets increases rapidly while the PER of DM packets still increases slowly. Thus, on one hand we can transmit DH packets when the BER is lower than a threshold value, BER^H, and on the other hand, transmit DM packets when the BER is lower than a threshold value, BER^M. Note that we can mark a channel's state as Better, Good, or Bad, by the BER, the PER [12] or received signal strength indication (RSSI) [13], etc. In addition, these two thresholds are not fixed, which can be adjusted dynamically.

Since different Bluetooth devices in a piconet have different interference levels due to location-dependent errors and the bit error rates seen by different frequencies in the hopping spectrum are significantly different from each other [22]. The master maintains a receiving frequency table as shown in Fig. 6, which is a $(n+1) \times h$ matrix M, where $n+1$ represents the master plus n slaves and h represents the number of operating RF channels. At receiving frequency k, the channel state of the master, M_{0K}, or the slave

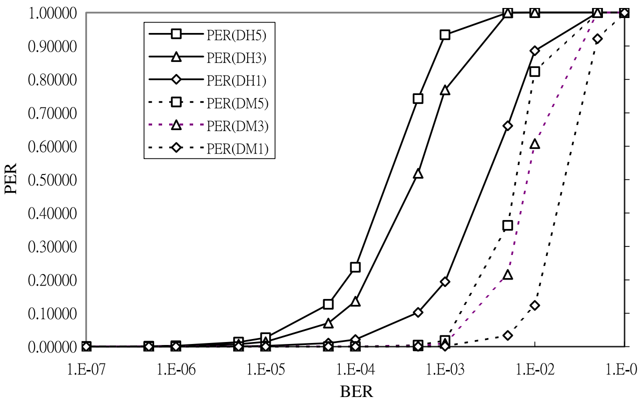
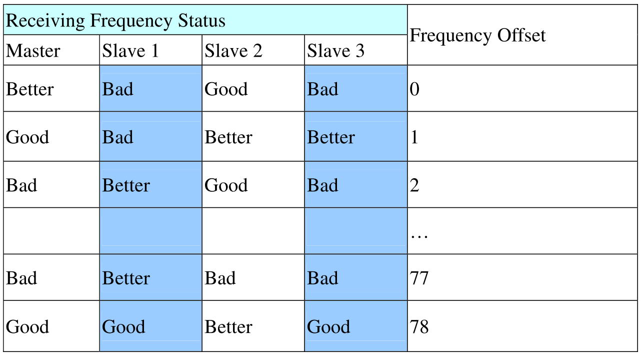


Fig. 5. Packet error rate of different packet types vs. bit error rate.



Receiving Frequency Status				Frequency Offset
Master	Slave 1	Slave 2	Slave 3	
Better	Bad	Good	Bad	0
Good	Bad	Better	Better	1
Bad	Better	Good	Bad	2
				…
Bad	Better	Bad	Bad	77
Good	Good	Better	Good	78

Fig. 6. Receiving frequency table.

i, M_{ik} (located at the column $i + 1$ and row k of the matrix M), is classified according to the BER measured in each channel and is marked as Bad, Good, or Better. Note that the slaves should send its link status to the master at a regular interval to update the receiving frequency table. It is not enough to consider each master–slave connection as an independent channel in the interference environments especially when the Bluetooth uses the Frequency Hopping (FH) scheme. Thus, we define an element of the matrix M to be one channel [12].

The Bluetooth specification did not specify any scheduling policy that the master should adopt for medium access control. The Round Robin (RR) scheduling is the simplest strategy for scheduling in Bluetooth. However, it not only leads to low link utilization and low throughput, but also is unsuitable for traffic sources with different data rates. Thus, we proposes a *Queue State Dependent Priority* (QSD-PR) policy to schedule packets based on the queue backlogs at the master queue and the slave queue to provide higher link utilization and hence higher throughput, and lower end-to-end delay. Note that the QSD-PR also takes the channel conditions into account and avoids bad channels by using the receiving frequency table at the master.

As shown in Fig. 7, when applying channel state dependent packet scheduling, a slot goes to waste primarily from two situations: (1) the master or slave having no data to send and (2) delay transmission in channel state dependent packet scheduling. Based on this observation, we also propose a *Channel State Dependent Segmentation and Reassembly* (CSD-SAR) scheme to maximize the link utilization by using multi-slot packets to mask bad frequencies. In an interference-limited environment, using small size packets and incorporating FEC protection will cause Bluetooth devices to generate more packets and thus result in more interference to the IEEE 802.11b. Oppositely, in a range-limited environment, using large size packets without incorporating FEC protection will result in a high packet error rate. Therefore, we will transmit packets as large size as possible on Better or Good channels and avoid transmitting on Bad channels according to the receiving frequency table. The detailed packet selection and packet scheduling scheme will be illustrated in the next two sections.

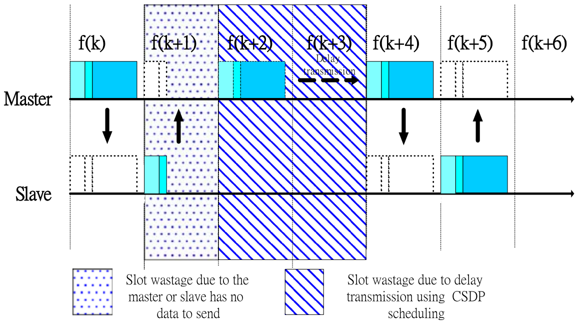

f(k)
f(k+1)
f(k+2)
f(k+3)
f(k+4)
f(k+5)
f(k+6)
Delay transmission
Master
Slave
Slot wastage due to the master or slave has no data to send
Slot wastage due to delay transmission using CSDP scheduling

Fig. 7. Slot wastage scenario in Bluetooth.

3.2. Packet selection scheme

There is a degree of flexibility in the choice of packet type: incorporating FEC or not. We select an appropriate packet type DH or DM to a specific slave according to the ratio of the total number of Good and Better channels between the master and a specific slave to the total number of frequencies, i.e., based on the location-dependent channel conditions for a specific slave. Frequently switching between protected (DM) and unprotected (DH) packets is inefficient due to message-passing overheads [23]. Therefore, according to the channel condition of the slave, we need to decide to use either DH or DM packets during each period of communication. The multi-slot packet uses less time to transmit the same amount of data that will result in higher throughput, lower end-to-end-delay and hence higher link utilization in either error-free or error-prone environments [19–21,24]. Thus, after determining the DM or DH packet type to send, we select the packet size as large as possible.

The adaptive packet selection scheme is shown in Fig. 8. Regardless of the Bad channels of a specific slave, if the ratio of Better channels to Good channels exceeds a threshold, $Ratio^{\text{threshold}}$, we will send DH packets in the Better channels, and select the packet size based on the remaining data size to fragment. However, if the ratio of Better channels to Good channels below a threshold, $Ratio^{\text{threshold}}$, we will use the Better channels and Good channels to transmit DM packets and select the packet size as large as possible, such as DM5 packets. That is, we will use the largest packet type to transmit on the Better or Good channels to mask

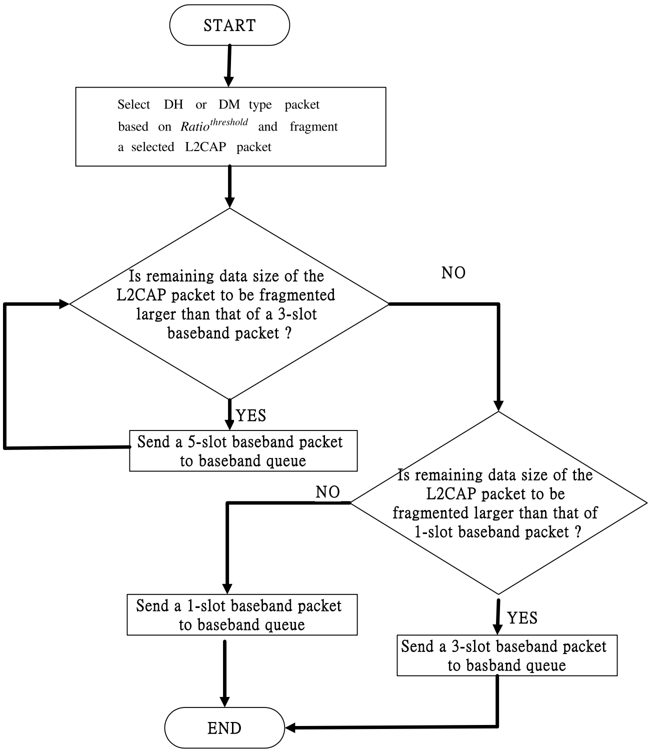

START
Select DH or DM type packet based on Ratio^threshold and fragment a selected L2CAP packet
Is remaining data size of the L2CAP packet to be fragmented larger than that of a 3-slot baseband packet ?
NO
YES
Send a 5-slot baseband packet to baseband queue
Is remaining data size of the L2CAP packet to be fragmented larger than that of 1-slot baseband packet ?
NO
Send a 1-slot baseband packet to baseband queue
YES
Send a 3-slot baseband packet to basband queue
END

Fig. 8. Adaptive packet selection scheme.

bad frequencies. In this way, we can resolve the wastage of slots due to delay transmission, as illustrated in Fig. 7. For example, with the L2CAP packet size of 500 bytes, if we decide to transmit DM packets, we will fragment the L2CAP packet into two DM5 packet of 224 bytes each and one DM3 packet of 52 bytes.

3.3. Scheduling policy

We will give each slave a priority based on the sum of queue backlogs (the number of data packets at the master and the slave queues), Q_{backlog}, and the waiting time T_{wait} since the slave has been scheduled to send packets previously. Thus, we give each slave a priority as follows:

$$P = \gamma\left(\frac{Q_{\text{backlog}}}{Q_{\max}}\right) + (1-\gamma)\left(\frac{T_{\text{wait}}}{T_{\max}}\right), \quad \gamma \leqslant 1 \tag{3}$$

where $Q_{\max}$ is the sum of maximum queue sizes at the master and a specific slave, and $T_{\max}$ is the maximum time that a specific slave can wait, which is negotiated during the master–slave connection setup based on QoS requirements. In this priority scheduling policy, we give a slave that has data to receive/send from/to the master a higher priority, and the lowest priority was given to a slave that has no queue backlogs at the master and the slave. $T_{\max}$ is specific to each slave to guarantee a bounded delay. Based on the priority scheme, each time we can select the next slave queue from which a packet should be sent.

The scheduling policy is shown in Fig. 9. The master selects a slave i to transmit a packet based on the priority from the set of slaves that can receive on current frequency k. Since the present Bluetooth architecture does not support packet reordering, we check the Head of Line (HOL) packet size at the queue corresponding to slave i. If the packet size is five time slots, we assume the slave will respond on frequency k_5, and we need to check the channel state of the receiving frequency of the master, M_{0k_5}, in the receiving frequency table. We will send this packet to slave i if either the packet is a DH packet and the channel state is Better or the packet is a DM packet and the channel condition is not in the Bad state. If the HOL packet size is three time slots or one time slot, we also check the channel state of the receiving frequency, that is similar to the procedure described above.

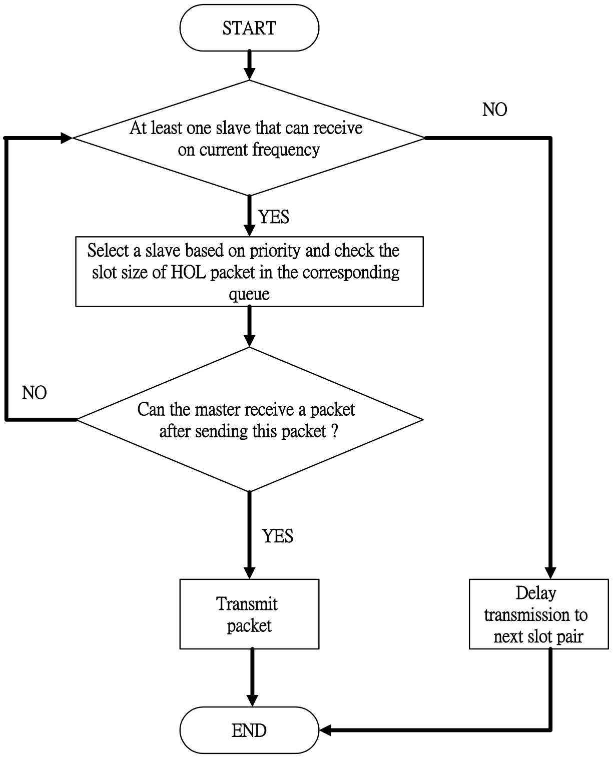

START
At least one slave that can receive on current frequency
NO
YES
Select a slave based on priority and check the slot size of HOL packet in the corresponding queue
NO
Can the master receive a packet after sending this packet ?
YES
Transmit packet
Delay transmission to next slot pair
END

Fig. 9. Packet scheduling policy.

Finally, if the HOL packet at the queue for a specific slave cannot be sent because of channel conditions, we will select another slave queue based on the priority to send its packet. If all slaves are unsuitable to send their HOL packets, we will delay the transmission to the next slot pair. Note that the delay rule is only implemented at the master side.

In the coexistence scenario of IEEE 802.11b and Bluetooth, the primary reason for packet drop is due to the interference between them, not the random bit errors caused by noise or the distance between devices. In the case of persistent errors that occupy certain static frequencies for much large duration, it may range from minutes to even several hours and cause severe interference to each other when Bluetooth hops over these infected frequencies. The design goal of our packet selection and scheduling scheme is to generate fewer packets by using large packet size and schedules the packets in a way to avoid bad frequencies. Thus, the coexistence problems can be solved and the impact of interference can be reduced.

3.4. Compared to BIAS [11]

There are several existing channel state dependent packet scheduling schemes [11,25,26] to improve performance in error-prone environments. Unlike [25,26], which were applied to Wireless LANs, [11] considered the hopping nature of the Bluetooth devices and distributed channels to devices in order to ensure fairness of access among users by means of max–min fairness criteria. It assumes an essential unit is a (master/slave) slot pair and distributes the bandwidth unused by the interference-prone sessions to other error-free connections [11]. Consequently, only the DM1 packet is used in the simulation environment. However, when the downstream (master-to-slave) traffic is not equal to the upstream (slave-to-master) traffic, this scheme will cause the wastage of slots. The objective of the proposed scheme is to maximize link utilization and throughput while that of the BIAS is to ensure fairness. Since the objectives of these two schemes are different, we only compare these two schemes, qualitatively, as shown in Table 2. Since the BIAS only considers the DM1 packet, it will results in low link utilization and low throughput.

Table 2
The proposed scheme in comparison with BIAS

	BIAS [11]	Proposed scheme
ACL link	DM1	All ACL packets
Fairness	Short term max–min fairness Unit: slot pair	Bounded delay
Restriction	Best when master's data rate is equal to each slave's	No
Link utilization	Low	High
Throughput	Low	High
Simulation model	MAC, PHY	TCP/IP, MAC, PHY

In the next section, we will compare our scheme with the RR only.

4. Evaluation and discussion

4.1. Simulation setup

This section explains the details of simulation setup: topology, traffic sources, and the error characteristics of RF channels. We used ns-2 [27] and a Bluetooth extension [28] to simulate our proposed scheme. In the simulation, every element in the receiving frequency table is considered separately as a channel. A channel is considered as either clear or interference affected. If a channel is clear, we assume the channel is in the Better state and the packets sent on this channel to a specific slave will not be corrupted. That is, the packets are always received successfully. An interference-affected channel may be in the Good or Bad state that can be modeled as a two-state Markov channel. According to Fig. 5, we assume the PER is not the same for all Bluetooth data packets. We calculate the PER of each packet type based on Eqs. (1) and (2) when BER is 10^{-3}: 0.934 for DH5, 0.769 for DH3, and 0.194 for DH1, and 0.018 for DM5, 0.010 for DM3, and 0.001 for DM1. Also, we assume that in the Bad state it is not always destructive, and we set the PER is 0.95 for all packets. We compare the performance of different packet selection and scheduling schemes by increasing the number of interference-affected channels gradually.

One study in [29] has proved that the finite state Markovian model can be used to effectively characterize the bursty bit error behavior of wireless links. Previous work on CSDP scheduling [25,26] used a two-state (Good and Bad) Markov process [30,31] to model the wireless link, as shown in Fig. 10. In the good state the BER, P_G, is low and in the bad state the BER, P_B, is high. Transitions between the two states occur according to the corresponding state transition probability of μ_G for transferring from the good state to the bad state and μ_B for transferring from the bad state to the good state.

The parameters of the two-state Markov model that were used in the simulation are to illustrate the behavior of the transport sessions when packets are subject to burst loss. We assume that the time spent in the Good and Bad periods are exponentially distributed, with different mean values, that is, different rates of state transition μ_G and μ_B. According to the properties of exponentially distributed random variables, the average time between state transitions can be expressed by $X_G = 1/\mu_G$ and $X_B = 1/\mu_B$ [24]. We set the parameter values $X_G = 500$ ms, $X_B = 500$ ms in scenario 1, and compare the performance improvements using our proposed scheme with different values of X_G and X_B in scenario 2.

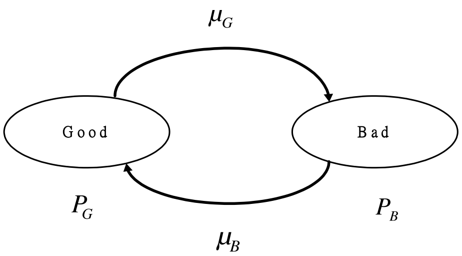


Fig. 10. Two-state Markov model.

The performance metrics that we used include *packet loss*, *end-to-end delay*, *link utilization*, and *transport layer throughput*. The packet loss is the probability that a packet is discarded at the MAC layer due to interference. It is expressed as the number of packets lost divided by the total number of packets sent during the simulation time. The end-to-end delay measures the elapsed time from the packet that is queued in the buffer until it is successfully received at the destination slave. The delay is measured at the L2CAP layer. Link utilization quantifies the percentage of total slots that are successfully used for transmission. That is, we did not take retransmission packets and NULL packets into account. The NULL packet has no payload and occupies one time slot [4]. Transport layer throughput is an indication of how much data that the user can receive per second.

The network topology used in the simulation is shown in Fig. 11. It includes one Bluetooth piconet that contains one master and three slaves. Note that the number of slaves was increased from three to seven in some simulations. In the simulation, data packets flows consist only from the master to the slaves and on the reverse direction only NULL packets for acknowledgements are returned. The traffic model used in the simulation has tried to capture a variety of traffic sources.

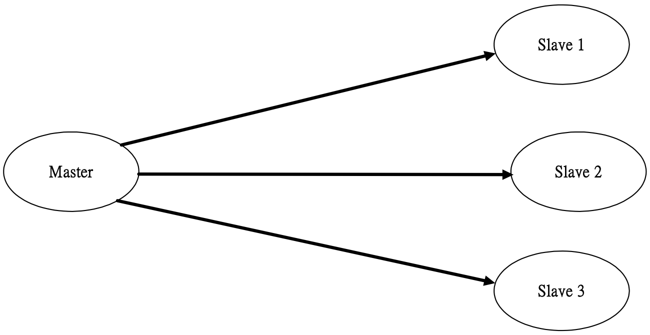


Fig. 11. Network topology used in the simulation.

Note that in this traffic model, we looked into the effect of all sources rather than a single one and hence we used the performance metrics described above to reflect the overall data performance rather than the performance of a single slave in a piconet [24].

4.2. Simulation results

We compare our scheme with RR under light offered load (scenario 1) and heavy offered load (scenario 2).

4.2.1. Scenario 1: Light offered load

In order to understand the packet loss and end-to-end delay, throughput and link utilization in interference environments, we setup simulation scenario 1 (light offered load) which is listed in Table 3. There are three data flows in this scenario: two 100 kbps Constant Bit Rate (CBR) flow and one exponential distributed data flow (exponential traffic). The data flows were all run over the transport layer and the UDP packet size is 500 bytes. The two CBR flows are guaranteed flows. Note that the exponential distributed data flow generates traffic according to an exponential *On*/*Off* distribution. Packets are sent at a fixed rate during *On* periods (Burst time), and no packets are sent during *Off* periods (Idle time). Both *On* and *Off* periods are taken from an exponential distribution [25].

Fig. 12 shows the packet loss when applying the Round Robin and our QSD-PR scheduling with different SAR schemes, respectively. The SAR schemes include the Random-SAR [32], which select data packet sizes (i.e., 1, 3 or 5) randomly and our proposed CSD-SAR scheme. Note that R-SAR (DH) stands for Random-SAR with DH type packets and R-SAR (DM) stands for Random-SAR with DM type packets. We set $r = 0.5$ for QSD-PR and $Ratio^{\text{threshold}} = 1$ for CSD-SAR in scenario 1. When the number of interference-affected channels increases, we can see that using the RR scheduling policy with R-SAR (DH) and R-SAR (DM) result in higher packet loss. The DM packets that incorporate FEC code can effectively reduce the percentages of packet loss. But the percentage of packet loss using RR still increases in proportion to the number of interference-affected channels. By taking channel conditions into consideration and using the receiving frequency table to avoid Bad channels, the percentage of packet loss almost kept at zero when applying our QSD-PR scheduling policy with either R-SAR or CSD-SAR scheme.

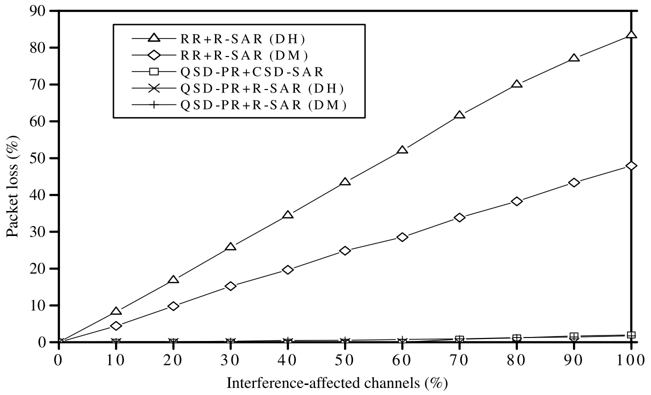


Fig. 12. Packet loss vs. interference-affected channels (%) in scenario 1.

Without loss of generality, we only analyze the simulation results for slave 1. The slave 1's end-to-end delay (experienced by the UDP packets) is shown in Fig. 13. Note that the interference-affected channels were distributed to the three slaves uniformly. When all the channels were clear and the QSD-PR and RR were applied the same R-SAR scheme, we can see that the end-to-end delay of QSD-PR is lower than that of RR. This is because the three data flows did not have equal data rates and resulted in the wastage of slots in the RR. We can also see that the QSD-PR with CSD-SAR or R-SAR (DM) scheme can achieve very low end-to-end delay even when all the channels were interference affected because the QSD-PR can use the Good periods of the channels (according to the receiving frequency table) to transmit DM packets and guarantee low end-to-end delay.

On the contrary, the end-to-end delay in the RR scheduling increases in a rapid pace when the number of interference-affected channels is greater than 50% of the total channels. Also, the end-to-end delay in the QSD-PR with

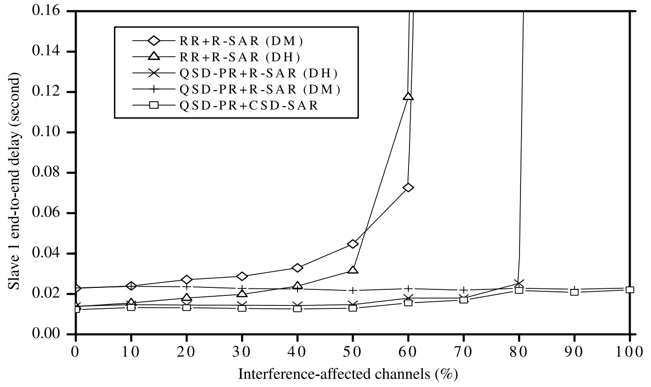


Fig. 13. End-to-end delay vs. interference-affected channels (%) in scenario 1.

Table 3
Properties of the data flows used in scenario 1

Slave no.	Property					
	Traffic type	Data rate (kbps)	Packet size (bytes)	Transport layer	Burst time (ms)	Idle time (ms)
1	CBR	100	500	UDP		
2	CBR	100	500	UDP		
3	Exponential Traffic	64	500	UDP	500	500

R-SAR (DH) scheme also increases rapidly when the number of interference-affected channels is greater than 80% of the total channels. When the number of interference-affected channels increases, the end-to-end delay increasing rapidly in the RR is due to frequent retransmission of packets. However, the end-to-end delay increasing rapidly in the QSD-PR with RSAR (DH) is due to no enough Better channels to transmit DH packets and results in frequent delayed transmission of packets.

Fig. 14 shows the link utilization vs. interference-affected channels. First, we can see that the link utilization is about 31% of the total slots when using DH packets to transmit data. However, the link utilization can achieve up to 48% of the total slots when using DM packets to transmit data. Since the DM packet incorporates (15, 10) Hamming code, the DM packet only carries about 2/3 of the data compared to the DH packet. That is, using the DM packet will need to generate more packets in order to transmit the same amount of data. Note that when the number of interference-affected channels increases up to 70% of the total channels, the ratio of Better channels to Good channels is below the threshold $Ratio^{\text{threshold}}$. Thus, CSD-SAR uses DM packets to fragment the transport layer packets instead of DH packets to resist the interference. This increases the link utilization from 30% to about 50% of the total slots.

Fig. 14 also shows that using the RR scheduling that did not take channel state information into consideration has lower link utilization when the percentage of interference-affected channels increases. The link utilization of the QSD-PR with R-SAR (DH) scheme also decreases when the percentage of interference-affected channels increases up to 90% of the total channels. The decrease of link utilization in the RR was primarily due to frequent packet retransmission while in the QSD-PR it was primarily due to delayed transmission to avoid Bad channels.

Fig. 15 shows the throughput of the piconet by using different packet selection and scheduling schemes. The throughput shows how much available bandwidth that is actually being used. We can see that the throughput kept constant at about 230 kbps (close to the offered data rate) regardless of the number of interference-affected channels of the total channels when applying the QSD-PR with CSD-SAR scheme. In contrast, the RR with RSAR scheme would cause the allocated slots not able to satisfy the required data rates of the three flows.

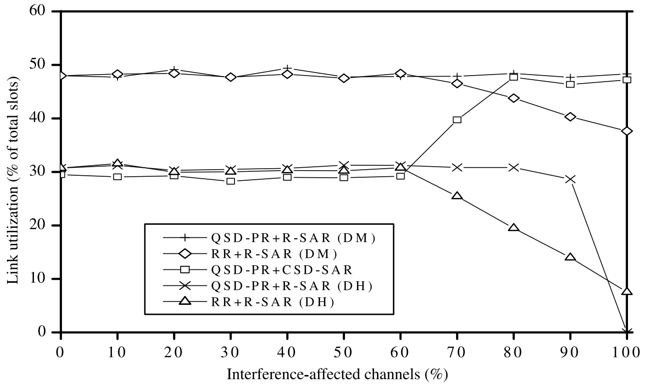


Fig. 14. Link utilization vs. interference-affected channels (%) in scenario 1.

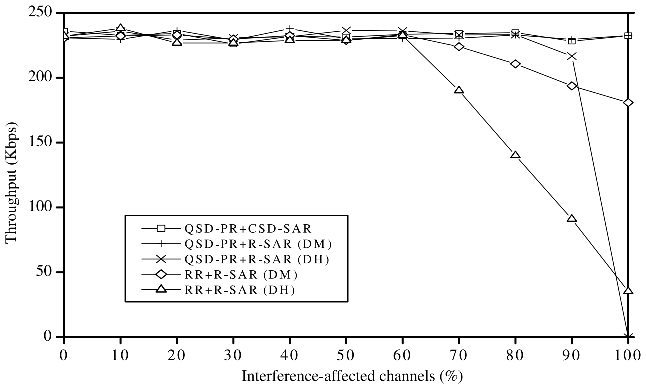


Fig. 15. Throughput vs. interference-affected channels (%) in scenario 1.

In summary, the performance obtained from scenario 1 show that a good packet selection scheme is important when applying a channel state dependent packet scheduling policy. When applying QSD-PR, the performance of CSD-SAR is outstanding compared to that of R-SAR. The CSD-SAR can select the best packet type and packet size according to channel conditions and guarantee higher throughput and lower end-to-end delay. In the next scenario, we applied a heavy offered load to the simulation network, and we will use the CSD-SAR scheme instead of the R-SAR scheme while using QSD-PR.

4.2.2. Scenario 2: heavy offered load

There are three data flows in scenario 2: two FTP flows and one exponential distributed data flow (exponential traffic). The FTP flow simulates bulk data transfer and will occupy as much as bandwidth as possible (ABR). The specifications of scenario 2 are listed in Table 4. In this scenario, the offered load was close to 100% of the total capacity and the link utilization was almost full all the time. That is, the throughput degradation is more susceptible by increasing the number of interference-affected channels. We also set $r = 0.5$ for QSD-PR and compare the link utilization and throughput with different $Ratio^{\text{threshold}}$ in scenario 2.

Fig. 16 shows that the link utilization when the percentage of interference-affected channels increased from 0% to 100%. When the percentage of interference-affected channels increased from 0% to about 40%, the QSD-PR with CSD-SAR could still maintain high link utilization and had high throughput. This is because the QSD-PR with CSD-SAR uses multi-slot packets to mask interference-affected channels. However, when the percentage of interference-affected channels increased over 40% of the total channels, the link utilization decreased gradually because of delayed transmission. In Fig. 17, it shows that by applying our QSD-PR with CSD-SAR scheme, we can achieve higher throughput in either error-free or error-prone

Table 4
Properties of the data flows used in scenario 2

Slave no.	Property					
	Traffic type	Data rate	Transport layer	Packet size	Burst time (ms)	Idle time (ms)
1	FTP	ABR	TCP	500 bytes, 40 bytes ACK		
2	FTP	ABR	TCP	500 bytes, 40 bytes ACK		
3∼7	Exponential traffic	64 kbps	UDP	500 bytes	500	500 ms

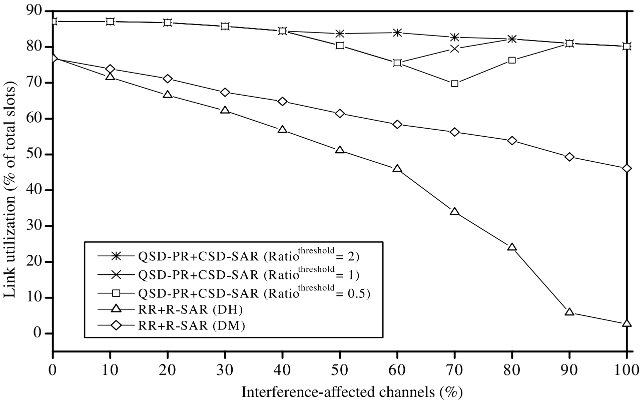


Fig. 16. Link utilization vs. interference-affected channels (%) in scenario 2.

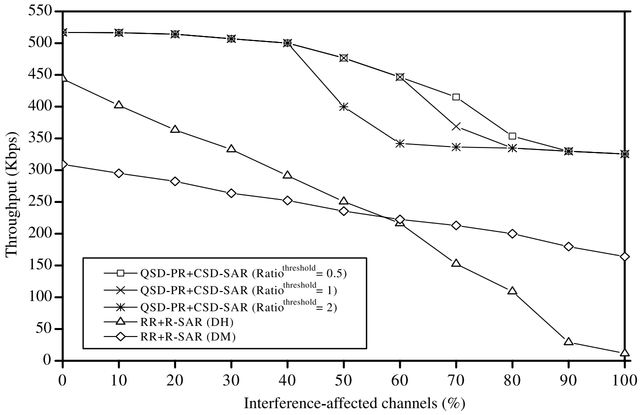


Fig. 17. Throughput vs. interference-affected channels (%) in scenario 2.

environments. Note that the threshold value $Ratio^{threshold}$ in CSD-SAR can be optimized and further enhance the throughput.

In scenario 2, the performance of our proposed scheme is significantly better than the RR with R-SAR scheme because we used the receiving frequency table to avoid bad channels and the transport layer packets were fragmented as large as possible to mask bad frequencies by using multi-slot packets. In addition, using the DM and DH packets based on channel conditions can efficiently reduce the packet error rate and guarantee high throughput in error-prone environments. Simulation results have shown that the QSD-PR with CSD-SAR scheme can adapt to error-prone environments under high load.

According to the results from Fig. 17, we set $Ratio^{threshold}$ *= 0.5 and gave the mean state residency time* X_G and X_B with different values and observed the throughput improvements using the proposed scheme. In Fig. 18, it shows that our QSD-PR with CSD-SAR scheme can offer throughput improvements as high as 195% compared to the RR with R-SAR (DH) scheme (when the percentage of interference-affected channels increases from 0% to 70%). In addition, the QSD-PR with CSD-SAR allows the Bluetooth system to remain usable even when all the channels are interference affected. Note that when channels are more error prone, the more improvement can be obtained by using our proposed scheme.

Finally, we investigated the effect of number of slaves on throughput improvements by increasing the number of

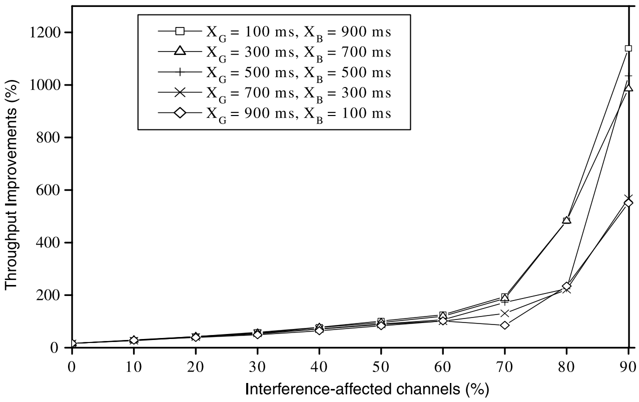


Fig. 18. Throughput improvements vs. different X_G and X_B.

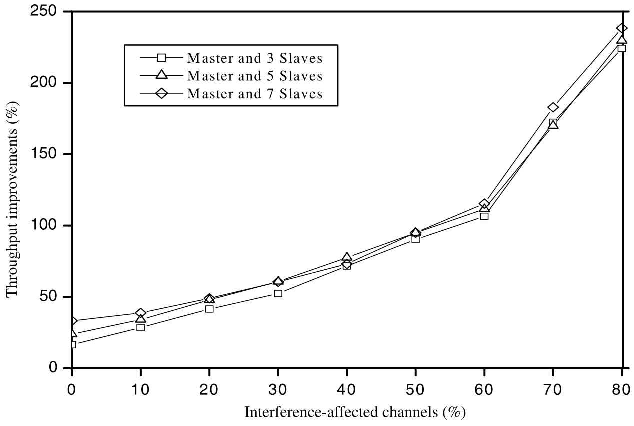


Fig. 19. Throughput improvements with various numbers of slaves in a piconet.

slaves from 3 slaves up to 7. The traffic type of slaves 3 through 7 is the exponential traffic that is specified in Table 4. A piconet has a limit on the maximum number of active slaves. In Fig. 19, by increasing the number of slaves in the piconet, we can see that the more throughput improvements can also be obtained by using our proposed scheme. This is because each slave in the piconet has a different data input rate and the RR scheduling scheme will waste more baseband slots by polling sources with low data input rates. It results in lower link utilization and thus lower throughput using the RR.

5. Conclusions

The market is rapidly moving toward resolving the coexistence concerns surrounding the IEEE 802.11b and Bluetooth [33]. Our proposed approach and other approaches have addressed the issue before it ever affects the end-user. As a result, market forecast for Bluetooth and IEEE 802.11b will remain strong, and the need for effective, multi-standard, coexistence solutions will only increase as wireless devices proliferate and simultaneous operation usage models become pervasive [33].

Simulation results have shown that our packet selection and scheduling scheme based on the channel state and queue state can have higher link utilization and higher throughput compared to the Round Robin packet scheduling scheme in an interference environment. In addition, the scheduling policy that delays transmission to avoid bad frequencies occupied by other devices will alleviate the impact of interference on the other systems significantly [10]. Note that our scheme can also be adapted and used in other centrally controlled TDD wireless systems, such as IEEE 802.15.1.

References

[1] L. Wang, Y.K. Kwok, W.C. Lau, Vincent K.N. Lau, Efficient packet scheduling using channel adaptive fair queueing in distributed mobile computing systems, Mobile Networks and Applications 9 (2004) 297–309.

[2] N. Golmie, Bluetooth dynamic scheduling and interference mitigation, Mobile Networks and Applications 9 (2004) 21–31.

[3] The Bluetooth Web Site <http://www.bluetooth.com./>.

[4] Bluetooth Special Interest Group, Specification of the Bluetooth System, Version 1.1, February 2001.

[5] S. Kapp, 802.11: leaving the wire behind, IEEE Internet Computing 6 (2002) 82–85.

[6] K.J. Negus, A.P. Stephens, J. Lansford, HomeRF: wireless networking for the connected home, IEEE Personal Communications 7 (2000) 20–27.

[7] ETSI, Broadband Radio Access Networks (BRAN): HIPERLAN type 2 technical specification, 2000.

[8] J. Lansford, A. Stephens, R. Nevo, Wi-Fi (802.11b) and Bluetooth: enabling coexistence, IEEE Network 15 (2001) 20–27.

[9] Javier del Prado, Sunghyun Choi, Experimental study on coexistence of 802.11b with alien devices, in: Proceedings of the IEEE 54th Vehicular Technology Conference 2, October 2001, pp. 977–981.

[10] N. Golmie, N. Chevrollier, Techniques to improve Bluetooth performance in interference environments, in: Proceedings of the IEEE Military Communications Conference 1, October 2001, pp. 581–585.

[11] N. Golmie, N. Chevrollier, I. EIBakkouri, Interference aware Bluetooth packet scheduling, in: Proceedings of the IEEE Global Telecommunications Conference 5, November 2001, pp. 2857–2863.

[12] S. Deb, M. Kapoor, A. Sarkar, Error avoidance in wireless networks using link state history, in: Proceedings of the IEEE INFOCOM 2, April 2001, pp. 785–795.

[13] Y.-H. You et al., Adaptive frequency hopping scheme for interference-limited WPAN applications, Electronic Letters 37 (2001) 976–978.

[14] J.C. Haartsen, The Bluetooth radio system, IEEE Personal Communications 7 (2000) 28–36.

[15] Bluetooth Special Interest Group, Bluetooth protocol architecture, White paper, 1999.

[16] Bluetooth Special Interest Group, Bluetooth Networking Encapsulation Protocol (BNEP) Specification, Specification of the Bluetooth System, Version 0.95, 2001.

[17] Bluetooth Special Interest Group, Bluetooth Personal Area Networking Profiles, Specification of the Bluetooth System, Version 0.95, 2001.

[18] S. Zurbes, Considerations on link and system throughput of Bluetooth networks, in: Proceedings of the 11th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications 2, September 2000, pp. 1315–1319.

[19] J. Kim, Y. Lim et al., An adaptive segmentation scheme for the Bluetooth-based wireless channel, in: Proceedings of the 10th International Conference on Computer Communications and Networks, October 2001, pp. 440–445.

[20] M.C. Valenti, M. Robert, J.H. Reed, On the throughput of Bluetooth data transmission, in: Proceedings of the IEEE Wireless Communications and Networking Conference 1, March 2002, pp. 119–123.

[21] T.Y. Chui, F. Thaler, W.G. Scanlon, Bit error rate related load constraints for Bluetooth baseband packets, Electronic Letters 38 (2002) 137–138.

[22] J. Zander, J. Malmgren, Adaptive frequency hopping in HF communications, IEEE Proceedings – Communications 142 (1995) 99–105.

[23] J. Bray, C.F. Sturman, Bluetooth: Connect Without Cables, Prentice Hall PTR, Eaglewood, 2001, pp. 348–350.

[24] A. Das, A. Ghose, A. Razdan, H. Saran, R. Shoery, Enhancing performance of asynchronous data traffic over the Bluetooth wireless ad-hoc network, in: Proceedings of the IEEE INFOCOM 1, April 2001, pp. 591–600.

[25] P. Bhagwat et al., Enhance throughput over wireless LANs using channel state dependent packet scheduling, in: Proceedings of the IEEE INFOCOM 3, March 1996, pp. 1133–1140.

[26] C. Fragouli, V. Sivaraman, M. Srivastava, Controlled multimedia wireless link sharing via enhanced class based queuing with channel state dependent queuing, in: Proceedings of the IEEE INFOCOM 2, March 1998, pp. 572–580.

[27] ns Notes and Documentation, the VINT Projects, UC Berkeley, LBL, USI/ISI and Xerox PARC, <http://www.isi.edu/nsnam/ns/>.

[28] Bluehoc simulator, a Bluetooth extension to NS, <http://www-124.ibm.com/developerworks/opensource/bluehoc/>.

[29] H. Wang, N. Moayeri, Finite state Markov channel – a useful model for radio communications, IEEE Transaction on Vehicular Technology 44 (1995) 163–171.

[30] E.O. Elliott, Estimates of error rate for codes on burst-noise channels, Bell System Technical Journal 42 (1963) 1977–1997.

[31] E.N. Gilbert, Capacity of burst-noise channel, Bell System Technical Journal 39 (1960) 1253–1266.

[32] M. Kalia, D. Bansal, R. Shorey, Data Scheduling and SAR for Bluetooth MAC, in: Proceedings of the IEEE Vehicular Technology Conference 2, May 2000, pp. 716–720.

[33] Mobilian Corp., Wi-Fi and Bluetooth: An examination of coexistence approaches, <http://www.mobilian.com/documents/WinHEC_whitepaper.pdf/>.

Chen-Han Shih received the B.S. degree in civil engineering and the M.S. degree in computer and information science from National Chiao Tung University, Taiwan, in 2000 and 2002, respectively. He is currently an engineer in the Network and Multimedia Institute, Wireless Network Technology Center, Institute for Information Industry. He has been working on WCDMA protocol testing for several years. His research interests include mobile communication and protocol testing.

Kuochen Wang received the B.S. degree in control engineering from National Chiao Tung University, Taiwan, in 1978, and the M.S. and Ph.D. degrees in electrical engineering from the University of Arizona in 1986 and 1991, respectively. He is currently a Professor in the Department of Computer Science, National Chiao Tung University. From 1980 to 1984, he worked on network management, and design and implementation of the Toll Trunk Information System at the Directorate General of Telecommunications in Taiwan. He served in the army as a second lieutenant communication platoon leader from 1978 to 1980. His research interests include wireless networks, mobile computing, and power management for mobile handheld devices.

Hung-Cheng Shih received the B.S. degree and the M.S. degree from the Department of Computer and Information Science, National Chiao Tung University, Taiwan, in 1997 and 2002, respectively. He is currently a Ph.D. student in the same department. His research interests include mobile computing, wireless Internet, 3G telecommunication systems, and power management.