Performance Analysis of Single Bluetooth Piconet in Error-Prone Environments

Soo Young Shin, Hong Seong Park, Dong-Sung Kim, and Wook Hyun Kwon

Abstract: **This paper analyzes the performance of a Bluetooth piconet in error-prone environments. A statistical characterization of a waiting time, an end-to-end delay, and a goodput are derived analytically in terms of the arrival rates, the number of slaves, and the packet error rate (PER). For simplicity, half-symmetric piconet is assumed in this analysis. Both exhaustive and limited scheduling are considered. The analytic results are validated by simulations.**

Index Terms: **Bluetooth, end-to-end delay, error-prone environments, goodput, packet error rate (PER), performance, piconet, waiting time.**

I. INTRODUCTION

A Bluetooth piconet consists of one master and up to seven active slaves [1]. In an applied environment, devices in Bluetooth piconets will sometimes receive erroneous packets, both the payload packets and the acknowledgement (ACK) packets. Therefore, for practical Bluetooth applications, it is necessary to investigate the performance of the Bluetooth piconet in environments where such errors occur. Because channel access in a Bluetooth piconet is controlled by means of a polling scheme, scheduling policy is one of the important factors that affect the performance of a polling system. The effect of scheduling policies on performance within a single Bluetooth piconet is analyzed in [2]–[4]. However, the analysis is based on simulation results only without considering the packet errors.

Some researches have analyzed the waiting time of a Bluetooth piconet [5]–[9]. The waiting time of a Bluetooth piconet is analyzed using an M/G/1 queue with vacations in [5]. In [6], the analysis of the limited scheduling algorithm for a piconet with asymmetrical traffic is analyzed based on polling system theory. The exact analysis of the waiting time for the limited scheduling algorithm for a symmetric Bluetooth is performed in [7]. In [8], the waiting time for the symmetric limited, gated, and exhaustive scheduling are obtained. The packet delay is analyzed with asymmetric traffic in a piconet [9].

These studies have analyzed performance using exhaustive, limited, and/or gated scheduling algorithms. However, these researches do not take packet errors into consideration either. When the packet errors occur, retransmissions are required for the successful data exchange. Those retransmissions make the service time vary. Therefore, the packet errors should be taken into consideration.

Some researches have examined the packet error rate (PER) [10]–[14]. In [10], the throughputs of various packet types are analyzed using simulation, taking into account errors in ACK packets. In [11], the PER and the throughput for various packet types are analyzed in the presence of fading and inter-piconet interference. In [12], the throughput of a piconet is provided analytically as a function of the number of packets in the presence of inter-piconet interference, although errors in ACK packets are not considered. The PER and the transfer delay in the presence of interference between Bluetooth and IEEE 802.11 WLAN are obtained using simulation in [13], [14].

Even though, these studies perform a mathematical analysis or a simulation of the effect of the PER on the throughput, but they do not consider the waiting time and the end-to-end delay. To the best of authors' knowledge, the analysis of the waiting time and the end-to-end delay of Bluetooth network considering the packet error has not been reported on yet in the technical literature.

In this paper, an average waiting time, an end-to-end delay and a goodput of a Bluetooth piconet in an error-prone environment are described in terms of the arrival rate, the number of slaves, and the PER. This paper considers the error control mechanisms such as the forward error correction (FEC) and the automatic repeat request (ARQ) to cope with packet error. For analytic simplicity, a piconet with unidirectional uplink traffic, i.e., half-symmetrical piconet, is considered with the exhaustive scheduling and the limited scheduling regimes.

The paper is organized as follows: The Bluetooth piconet system model is defined in Section II, the performance measures are derived analytically in Section III, numerical results are provided in Section IV, and the conclusions are presented in Section V.

II. BLUETOOTH PICONET SYSTEM MODEL

The Bluetooth piconet uses a frequency hopping with time division duplex (FH/TDD) for the channel access mechanism as shown in Fig. 1.

In Bluetooth piconet, master controls the channel access by two methods: Sending data and polling. The master transmits data packet to the desired slave at even time slot, $f(2k)$, and the desired slave can transmit data packet if there is data to send at odd time slot, $f(2k+1)$. Otherwise the desired slave only responds with ACK packet. Polling scheme by a POLL packet is used when there are packets for slave to send to the master and the master has no packets to send.

Fig. 2 shows a packet structure of Bluetooth. For simplicity, only asynchronous connectionless link (ACL) with DHx

Manuscript received August 31, 2005; approved for publication by Luciano Lenzini, Division II Editor, July 4, 2007.

S. Y. Shin and W. H. Kwon are with the School of Electrical Engineering and Computer Science, Seoul National University, Seoul, Korea, email: {wdragon, whkwon}@cisl.snu.ac.kr.

H. S. Park is with the department of Electrical and Computer Engineering, Kangwon National University, Kangwondo, Korea, email: hspark@kangwon.ac.kr.

D.-S. Kim is with the School of Electronic Engineering, Kumoh National Institute of Technology, Korea. email: dskim@kumoh.ac.kr.

1229-2370/07/$10.00 © 2007 KICS

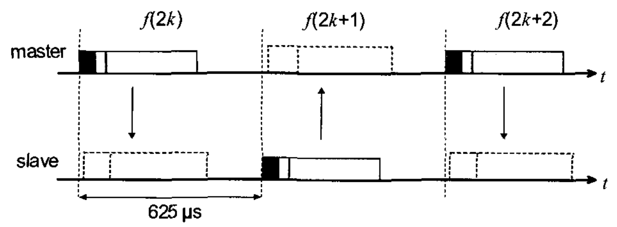

Fig. 1. Time division duplex of the Bluetooth piconet.

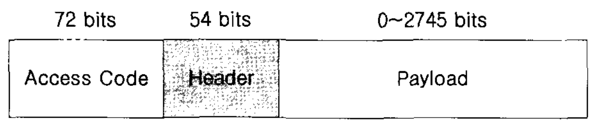

Fig. 2. Bluetooth packet structure.

packets is considered in this paper, where x means the length of time slots (x = 1, 3, 5). A Bluetooth packet consists of three fields: the access code, the header, and the payload. Denote P_{ac}, P_H, and $P_{P,DHx}$ be the error probability in the access code, the packet header, and the payload of a DHx packet, respectively. In the Bluetooth systems, data transmission consists of sending a data packet, which in general contains all the access code, header and payload, and receiving an ACK packet, which contains an access code and a header with or without a payload.

Assume that b is the bit error rate (BER) over a memoryless channel. The access code uses (64, 30) BCH coding that can correct at most 9 errors [15]. Then, the access code error probability can be shown as

$$P_{ac} = 1 - \left[\sum_{x=0}^{9} \left(\begin{array}{c} 64 \\ x \end{array}\right) b^x (1-b)^{64-x}\right]. \quad (1)$$

The header uses the 1/3 rate FEC, in which each bit is repeated three times, thus one error within each three consecutive bits can be corrected. The header error probability is shown as

$$P_H = 1 - \left[\sum_{x=0}^{1} \left(\begin{array}{c} 3 \\ x \end{array}\right) b^x (1-b)^{3-x}\right]^{18}. \quad (2)$$

The payload error probability of the DHx packet is given by:

$$P_{P,DHx} = 1 - (1-b)^{L(\mathrm{DH}x)} \quad (3)$$

where $L(DHx)$ is the length of DHx packet payload.

A piconet model is shown as Fig. 3. Denote N be the number of slaves in a piconet, where $N \leq 7$. Each slave has an infinite buffer. A half-symmetrical piconet is assumed, i.e., the arrival rate to all slave queues is the same with $\lambda_{ui} = \lambda$ and the arrival rate to master is assumed to be 0, i.e., $\lambda_d = 0$ as shown in Fig. 3. The arrival process is assumed to be Poisson. The master is the destination of all packet generated at the slaves. The packet length can be 1, 3, or 5 slots with probability p_1, p_3, or p_5, respectively. In this scenario, the probability of a successful data packet transmission is

$$P_{S,DHx} = \{(1-P_{ac})(1-P_H)\}^2(1-P_{P,DHx}) \quad (4)$$

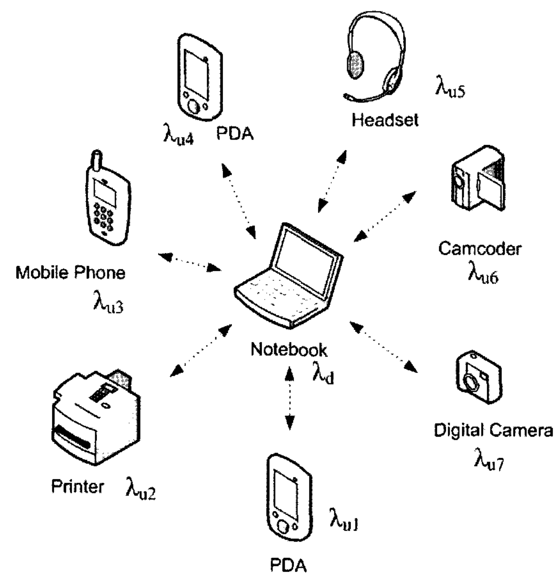

Fig. 3. Bluetooth piconet.

which means that the POLL packet by the master and the DHx packet by a desired slave are successfully transmitted.

III. PERFORMANCE ANALYSIS

In this paper, an average waiting time, an end-to-end delay, and a goodput are used for measuring the performance of a piconet.

- Waiting time—the waiting time is the time a packet waits in the slave queue before it is served.
- End-to-end delay—the end-to-end delay is the time elapsed from the moment that a packet is arrived at a sending queue to the moment that the packet is arrived correctly to receiver.
- Goodput—the goodput is the the throughput of successfully transmitted packets.

A. Exhaustive Scheduling

As stated earlier, half-symmetrical piconet, i.e., a piconet in which $\lambda_u = \lambda > 0$ and $\lambda_d = 0$ is assumed. Since $\lambda_d = 0$, when the master communicates with a slave, it sends only POLL packets. In the exhaustive scheduling, the slave replies with data packets until its queue is empty. If the queue is empty, the slave responds with NULL packet. In the exhaustive scheduling, the service time of DHx packet is defined as $(x+1)$ time slots as [8]. In other words, to serve DHx packet, $(x+1)$ time slots are required because of x time slots for data and 1 time slot for POLL packet. Hence, the service time of a DH1 slot packet is defined as 2 time slots, where one slot time is $T_S = 625\mu s$. For a DH3 slot and a DH5 slots packet, the service time are 4 time slots and 6 time slots, respectively. The switch over time is defined as $2T_S$, i.e., the NULL packet ending the exchange with a slave and the POLL packet starting the exchange with the next slave. Denote the mean and the second moment of the packet service time to be $\overline{X}$ and $\overline{X^2}$. The mean and variance of the switchover time are defined as $\overline{V}$ and σ_V^2. Fig. 4 shows an example of the piconet operation with exhaustive scheduling

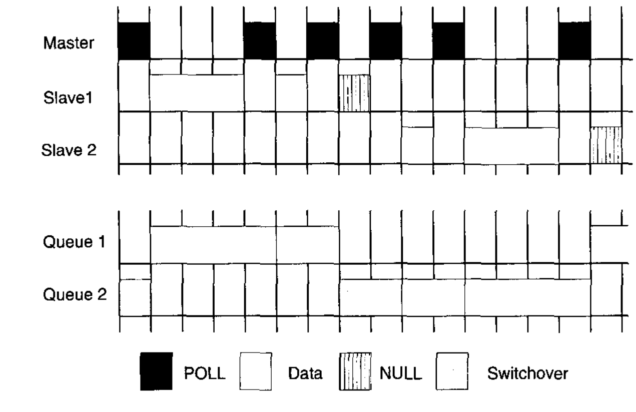

Fig. 4. A piconet with exhaustive scheduling algorithm and the equivalent polling system.

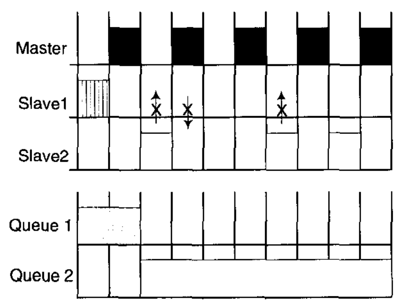

Fig. 5. DH1 packet service time example with packet errors.

algorithm and its the equivalent polling system.

However, if the packet error occurs during the packet exchanges, both the service time and the switchover time will vary according to the PER. Let's consider transmissions of the DH1 packet in error-prone environments shown as Fig. 5. Note that an arrow with 'X' mark means unsuccessful packet transmission.

At the first and the second slot, the switchover is accomplished. Slave 2 tries to send a DH1 packet but it is transmitted incorrectly. Master polls again Slave 2, but at this time, the POLL packet is corrupted. Because there is no response from the desired slave, i.e., Slave 2, Master polls again, and Slave 2 sends the DH1 packet, which is also corrupted. Finally, Master transmits the POLL packet and Slave 2 transmits the DH1 packet successfully. Then, Master sends another POLL packet to Slave 2. The DH1 packet can be transmitted successfully at the 10th time slot. Therefore, as illustrated in Fig. 5, the service time is $8T_S$[1].

Assume that a DH1 packet is successfully transmitted after the kth retransmission. Then, the service time distribution of the DH1 packet can be expressed as

$$\Pr\{X_{DH1} = 2(k+1)T_S\} = (1 - P_{S,DH1})^k P_{S,DH1} \quad (5)$$

where the X_{DHx} is a random variable for the service time of the DHx packet and $k = 0, 1, \cdots$. By simple manipulation,

[1]In the exhaustive scheduling, it is assumed that the succeeding POLL packet from a master will be successful when the data packet from a slave is successful. This assumption could be applied without loss of generality because it occurs scarcely that the succeeding POLL packet from a master fails after the data packet from a slave is successful.

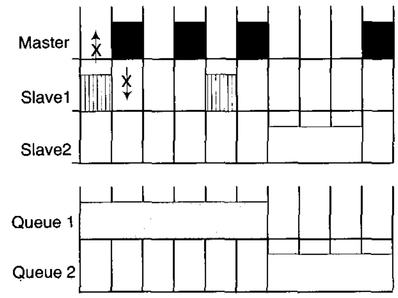

Fig. 6. An example of switchover time with packet errors.

the mean and the second moment of the delay suffered by DH1 packet are obtained as

$$\overline{X_{DH1}} = \frac{2T_S}{P_{S,DH1}} \quad (6)$$

and

$$\overline{X_{DH1}^2} = \frac{4T_S^2(2 - P_{S,DH1})}{P_{S,DH1}^2}. \quad (7)$$

Using the same method, the mean and the second moment of the delay suffered by the DH3 and DH5 packet can be obtained easily.

$$\begin{aligned} \overline{X_{DH3}} &= \frac{4T_S}{P_{S,DH3}} \\ \overline{X_{DH3}^2} &= \frac{16T_S^2(2 - P_{S,DH3})}{P_{S,DH3}^2} \quad (8) \\ P_{S,DH3}^2\overline{X_{DH5}} &= \frac{6T_S}{P_{S,DH5}} \\ \overline{X_{DH5}^2} &= \frac{36T_S^2(2 - P_{S,DH5})}{P_{S,DH5}^2}. \quad (9) \end{aligned}$$

Fig. 6 illustrates the switchover time which is varying with packet errors. In Fig. 6, the first NULL packet is received incorrectly at the master. Then, the master retransmits the POLL packet, but it is received incorrectly at the desired slave. After T_S, the master retransmits the POLL packet because there is no response from the slave. At this time, the slave receives the POLL packet well, and transmits the NULL packet. The master receives the NULL packet well and transmits the POLL packet to the next desired slave. In this example, the switchover time is 6 time slots, i.e., 6 T_S. Since both the POLL and the NULL packet are composed of an access-code and a header only, the probability of the successful POLL or NULL packet transmission is $(1 - P_{ac})(1 - P_H)$. Therefore, the probability of the successful switchover is $\{(1 - P_{ac})(1 - P_H)\}^2$ due to the exchange of one POLL packet and one NULL packet.

Assume that j retransmissions are required for a successful switchover. The switchover time distribution is given by

$$\Pr\{V = 2(j+1)T_S\} = ((1 - P_{ac})(1 - P_H))^2 \left\{1 - ((1 - P_{ac})(1 - P_H))^2\right\}^j \quad (10)$$

where $j = 0, 1, \cdots$. Then, the mean switchover time is obtained as

$$\overline{V} = \frac{2T_S}{\{(1-P_{ac})(1-P_H)\}^2} \tag{11}$$

The variance of the switchover time can be obtained easily as

$$\sigma_V^2 = \frac{4T_S^2\left\{1-((1-P_{ac})(1-P_H))^2\right\}}{((1-P_{ac})(1-P_H))^4}. \tag{12}$$

As stated in [8], the waiting time could be obtained using (3.69) in [16]. The mean service time and the second moment of the service time of one DH packet, which are denoted as $\overline{X_{DH}}$ and $\overline{X_{DH}^2}$, are derived as

$$\begin{aligned}\overline{X_{DH}} &= p_1\overline{X_{DH1}} + p_3\overline{X_{DH3}} + p_5\overline{X_{DH5}}\\ \overline{X_{DH}^2} &= p_1\overline{X_{DH1}^2} + p_3\overline{X_{DH3}^2} + p_5\overline{X_{DH5}^2}.\end{aligned} \tag{13}$$

The system load, ρ, is $N\lambda\overline{X_{DH}}$ and the average waiting time of the exhaustive scheduling can be expressed as

$$W_{u,ex} = \frac{N\lambda\overline{X_{DH}^2}}{2(1-\rho)} + \frac{(N-\rho)\overline{V}}{2(1-\rho)} + \frac{\sigma_V^2}{2\overline{V}}. \tag{14}$$

Note that (14) is identical to (1) in [8] when the BER is zero.

An end-to-end delay of the exhaustive scheduling, $D_{e2e,ex}$, is defined as sum of the waiting time and the service time. Hence, the average end-to-end delay can be expressed as

$$D_{e2e,ex} = W_{u,ex} + \overline{X_{DH}} - T_S. \tag{15}$$

A Goodput is defined as the the throughput of successfully transmitted packets and is interpreted as the bandwidth usage for successful packet transmission excluding the packet overhead such as access code and header. The goodput of a piconet, G_{ex}, can be easily obtained from the packet arrival rate, the number of slaves, and the service time as

$$G_{ex} = \lambda N\overline{L} \tag{16}$$

where $\overline{L}$ is the average payload length, i.e., $\overline{L} = p_1L(DH1) + p_3L(DH3) + p_5L(DH5)$.

B. Limited Scheduling

In the limited scheduling, i.e., pure round robin, at most a single packet is sent in each direction (downlink and uplink) whenever a master-slave queue pair is served. In the half-symmetrical piconet, i.e., a piconet in which $\lambda_u = \lambda > 0$ and $\lambda_d = 0$, the master communicates with a slave with POLL packets only. The polled slave replies with a data packet or a NULL packet and then the master polls another slave. A piconet with only uplink traffic operated according to the limited (pure round robin) scheduling algorithm can be modeled as a 1-limited polling system [8]. The switchover time to a queue is defined as the time elapsed from the instance in which the preceding slave starts transmitting the last time slot (even for the NULL packet) to the instance that the master completes the transmission of the POLL packet intended to the slave. In error-free environments, the switchover time is $2T_S$.

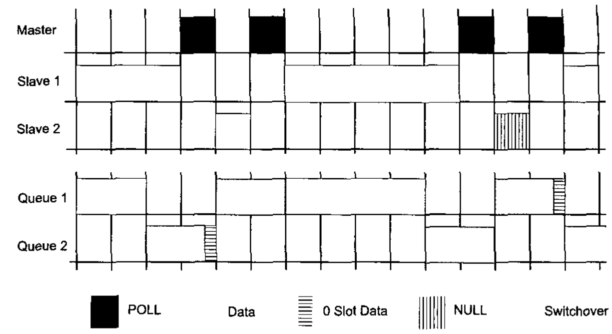

Fig. 7. A piconet with limited scheduling algorithm and the equivalent polling system.

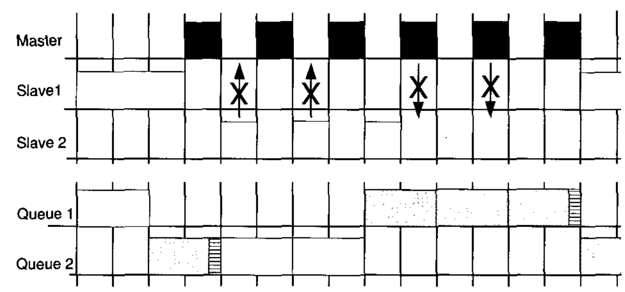

Fig. 8. Packet service time and switchover time example in limited scheduling with packet errors.

Fig. 7 shows an example of the piconet operation with the limited scheduling algorithm and its the equivalent polling system based on [8]. When data packets are sent, some of the data is actually sent during the switchover to the next queue as illustrated at 4th slot of the equivalent polling system in Fig. 7. In the limited scheduling, the service time of a DHx packet is defined as $(x-1)$ time slots without packet errors, where x is 1, 3, or 5.

However, if there are packet errors, both the service time and the switchover time will vary due to the retransmissions time. Let's consider transmissions of the DH1 packet in the error-prone environments first as illustrated in Fig. 8. To transmit a DH1 packet, the preceding switchover must be successful. In other words, the POLL packet transmitted by the master is successfully received at the slave as shown in the 4th time slot in Fig. 8. At the 5th time slot, the Slave 2 tries to send DH1 packet but it is corrupted. Master polls again Slave 2, but Slave 2 fails to transmit successfully at the 7th slot[2]. Finally, the DH1 packet from Slave 2 is successfully transmitted at the 9th slot. Therefore, in the example, the service time of a DH1 packet is $4T_S$, compared to $0T_S$ in the error-free environments in Fig. 7.

Because the successful switchover is prerequisite for the data transmission, in other words, POLL packet is received successfully, the service time of the DH1 packet can be 0 with the probability of $\alpha_{DH1} = (1-P_{ac})(1-P_H)(1-P_{P,DH1})$, which means only a DH1 packet is successfully transmitted in the first transmission. Assume that a DH1 packet is successfully transmitted after the kth retransmission. Then, the service time distribution

[2]There are two options for the packet retransmissions such as immediate and not-immediate ARQ in the limited scheduling [18]. In this paper, the immediate ARQ is assumed for the time critical applications.

of the DH1 packet can be expressed as

$$\Pr\{X_{DH1} = 2kT_S\} = \begin{cases} \alpha_{DH1}, & k = 0 \\ \beta\left(1 - P_{S,DH1}\right)^{k-1}, & k = 1, 2, \cdots \end{cases} \quad (17)$$

where $\beta = (1 - \alpha_{DH1}) P_{S,DH1}$. By simple manipulation, the mean and the second moment of the delay suffered by a DH1 packet in the limited scheduling are obtained as

$$\overline{X_{DH1}} = (1 - \alpha_{DH1}) \frac{2T_S}{P_{S,DH1}} \quad (18)$$

and

$$\overline{X^2_{DH1}} = (1 - \alpha_{DH1}) \frac{T_S^2 (2 - P_{S,DH1})}{P^2_{S,DH1}}. \quad (19)$$

The mean service time and the second moment of the delay suffered by a DH3 and a DH5 packets are easily obtained as

$$\begin{aligned} \overline{X_{DH3}} &= 2T_S + (1 - \alpha_{DH3}) \tfrac{4T_S}{P_{S,DH3}} \\ \overline{X^2_{DH3}} &= 4T_S^2 + (1 - \alpha_{DH3}) \tfrac{32T_S^2}{P^2_{S,DH3}} \end{aligned} \quad (20)$$

and

$$\begin{aligned} \overline{X_{DH5}} &= 6T_S + (1 - \alpha_{DH5}) \tfrac{6T_S}{P_{S,DH5}} \\ \overline{X^2_{DH5}} &= 16T_S^2 + (1 - \alpha_{DH5}) \tfrac{T_S^2(72+12P_{S,DH5})}{P^2_{S,DH5}} \end{aligned} \quad (21)$$

where $\alpha_{DHx} = (1 - P_{ac})(1 - P_H)(1 - P_{P,DHx})$.

In the limited scheduling piconet, the switchover will be taken place only when the previous data packet, transmitted by previous slave, is successfully received at the master. Otherwise, the time required for the packet retransmission is counted as the service time of the previous packet. Since the previous data packet is received successfully, if the POLL packet for the next slave is error-free with probability of $(1 - P_{ac})(1 - P_H)$, the switchover time is $2T_S$. That means the switchover is finished without retransmissions. However, if the POLL packet could be erroneous like the 10th time slot in Fig. 8, Slave 1 cannot respond. Then, the master polls Slave 1 again, but it fails again. Finally, the master polls Slave 2 successfully at the 14th time slot, which means the switchover is performed successfully. Here, as illustrated in Fig. 8, the switchover time is $6T_S$. Note that switchover contains a 0 Slot Data because the polled slave, Slave 1, transmits a DH1 packet.

Assume that j retransmissions are required for a successful switchover. The switchover time distribution is given by

$$\begin{aligned} &\Pr\{V = 2(j+1)T_S\} \\ &= (1 - (1 - P_{ac})(1 - P_H))^j (1 - P_{ac})(1 - P_H) \end{aligned} \quad (22)$$

where $j = 0, 1, \cdots$. The mean switchover time and the variance are obtained as

$$\overline{V} = \frac{2T_S}{(1 - P_{ac})(1 - P_H)} \quad (23)$$

$$\sigma_V^2 = \frac{4T_S^2 \{1 - (1 - P_{ac})(1 - P_H)\}}{((1 - P_{ac})(1 - P_H))^2} \quad (24)$$

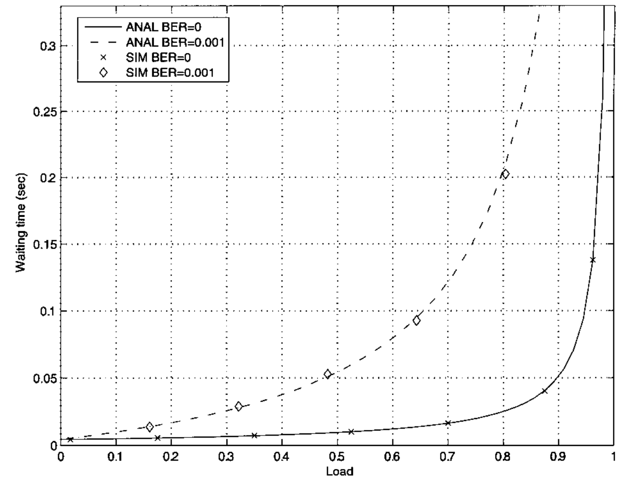

Fig. 9. Average waiting time values vs. load in a piconet composed of 7 slaves and with only uplink traffic, operated according to the exhaustive algorithm with $p_1 = p_3 = p_5 = 1/3$.

For the half-symmetric piconet with uplink traffics only, the model for a symmetrical discrete time 1-limited polling system is used in [17]. Using the mean service time, $\overline{X_{DH}} = p_1\overline{X_{DH1}} + p_3\overline{X_{DH3}} + p_5\overline{X_{DH5}}$ and the second moment of the service time, $\overline{X^2_{DH}} = p_1\overline{X^2_{DH1}} + p_3\overline{X^2_{DH3}} + p_5\overline{X^2_{DH5}}$ of one DH packet, the system load, $\rho = N\lambda(\overline{X_{DH}} + T_S)$, the average waiting time of the limited scheduling, $W_{u,lm}$, can be expressed as

$$W_{u,lm} = \frac{\sigma_V^2}{2\overline{V}} + \frac{N\left(\lambda\overline{X^2_{DH}} + \overline{V}\left(1 + \lambda\overline{X_{DH}}\right) + \lambda\sigma_V^2\right)}{2\left(1 - N\lambda\left(\overline{V} + \overline{X_{DH}}\right)\right)}. \quad (25)$$

Note that (25) is identical to (14) and (4) of [8] in the error-free environments.

An end-to-end delay, $D_{e2e,lm}$ and a goodput, G_{lm} are obtained easily as defined as

$$\begin{aligned} D_{e2e,lm} &= W_{u,lm} + \overline{X_{DH}} + T_S \\ G_{lm} &= \lambda N \overline{L} \end{aligned} \quad (26)$$

where $\overline{L}$ is the average payload length.

IV. NUMERICAL RESULTS

Simulation is performed using OPNET [19]. In the error-prone environments, the bit error rate (BER) is set to 0.1%, which is minimum requirements in the Bluetooth specification [1]. For each load value, the results have been computed after 4 hours simulation.

A. Exhaustive Scheduling

Figs. 9, 10, and 11 compare the analytical results with simulation results of the exhaustive scheduling under the error-free and error-prone environments.

Figs. 9 and 10 show the waiting time and the end-to-end delay as a function of the load in the uplink exhaustive system, ρ, in half-symmetrical piconet with 7 slaves.

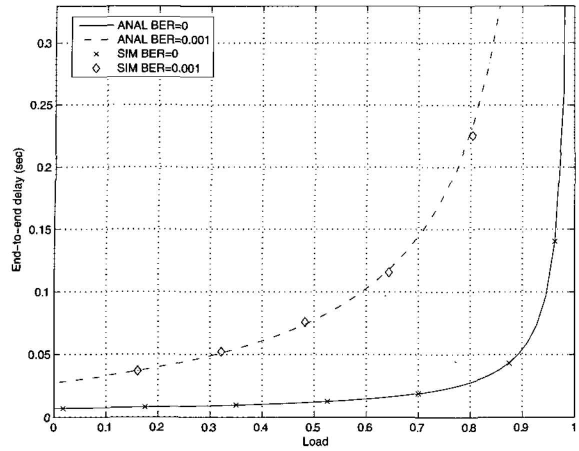

Fig. 10. End-to-end delay values vs. load in a piconet composed of 7 slaves and with only uplink traffic, operated according to the exhaustive algorithm with $p_1 = p_3 = p_5 = 1/3$.

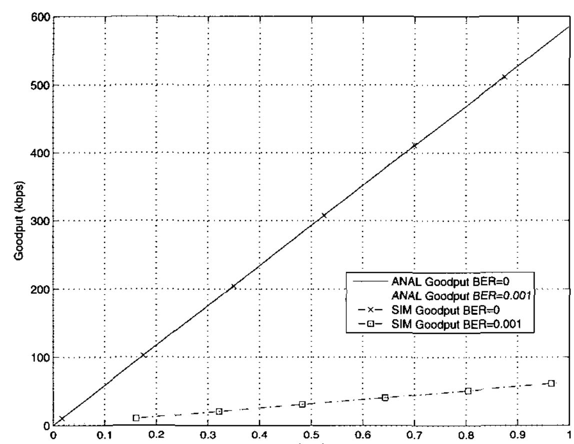

Fig. 11. Goodput vs. load in a piconet composed of 7 slaves and with only uplink traffic, operated according to the exhaustive algorithm with $p_1 = p_3 = p_5 = 1/3$.

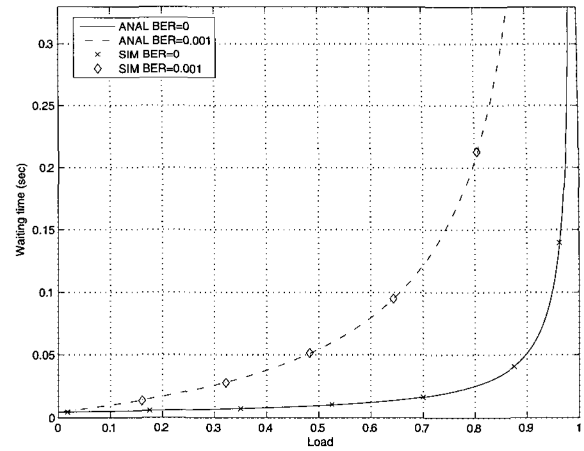

Fig. 12. Average waiting time values vs. load in a piconet composed of 7 slaves and with only uplink traffic, operated according to the limited algorithm with $p_1 = p_3 = p_5 = 1/3$.

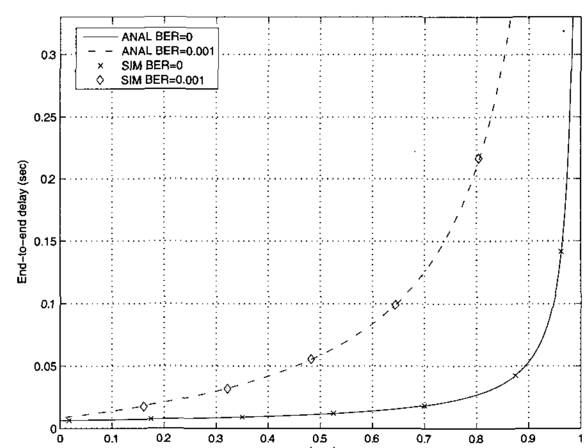

Fig. 13. End-to-end delay values vs. load in a piconet composed of 7 slaves and with only uplink traffic, operated according to the limited algorithm with $p_1 = p_3 = p_5 = 1/3$.

For example, when $\rho = 0.5$, the average waiting time of BER $= 0.1\%$ is about 0.054548 second. The end-to-end delay of the BER $= 0.1\%$ is 0.076898 second in the exhaustive scheduling when $\rho = 0.5$.

Goodputs of a piconet with different system load using the exhaustive scheduling is shown in Fig. 11. For example, when $\rho = 0.5$, the goodput of BER $= 0\%$ is about 292.8 kbps and that of BER $= 0.1\%$ is about 31.9 kbps. The goodput with BER $= 0$ is approximately 9.2 times larger than that with BER $= 0.1\%$.

Figs. 9, 10, and 11 validate the analytical results of the exhaustive scheduling in the error-free and error-prone environments.

B. Limited Scheduling

Figs. 12 and 13 show the waiting time and the end-to-end delay as a function of the load in the uplink limited system, ρ, in half-symmetrical piconets with 7 slaves.

The waiting time of the limited scheduling with BER $= 0\%$ shows the same plotting with Fig. 9 because the mean waiting time of the limited scheduling is identical to that of the exhaustive scheduling in the error-free environments. In the error-prone environments, when $\rho = 0.5$, the average waiting time of BER $= 0.1\%$ of the limited scheduling is about 0.054587 second.

The end-to-end delay of the BER $= 0.1\%$ is 0.076936 second in the limited scheduling scheduling when $\rho = 0.5$. Both the average waiting time and the end-to-end delay are larger in the limited scheduling compared to the exhaustive scheduling when BER $= 0.1\%$.

Figs. 12, and 13 validate the analytical results of the limited scheduling in the error-free and error-prone environments.

V. CONCLUSION

In this paper, the average waiting time, the end-to-end delay, and the goodput of a single Bluetooth piconet were analyzed for the error-prone environments in terms of the arrival rate, the number of slaves, and the PER. This paper considers the error control mechanisms such as the FEC and the ARQ to cope with packet errors. In the half-symmetric piconet with uplink traffics only, both the exhaustive and the limited scheduling are considered for the analysis.

The main contribution of this paper is to provide an analysis of the performance of a Bluetooth piconet in the error-prone environments, which to our knowledge has not been carried out previously. The performance of the error-free case was compared to that of the 0.1% BER case. The effects of different packet types such as DH1, DH3, DH5, and various arrival rates were analyzed. Some simulations validated the correctness of the analysis.

This analysis may be useful in the design and application of Bluetooth piconets.

REFERENCES

[1] SIG Bluetooth, *Specificaton of the Bluetooth System*, 2000.

[2] N. Johansson, U. Korner, and P. Johansson, "Pefromance evaluation of scheduling algorithms for Bluetooth," in *Proc. BC IFIP TC 6 Fifth International Conference on Broadband Communication*, vol. 1, 1999, pp. 139–150.

[3] M. Kalia, D. Bansal, and R. Shorey, "Data scheduling and SAR for Bluetooth MAC," in *Proc. VTC-spring*, vol. 2, May 2000, pp. 716–720.

[4] A. Capone, M. Gerla, and R. Kapoor, "Efficient polling schemes for Bluetooth picocells," in *Proc. ICC*, vol. 7, June 2001, pp. 1990–1994.

[5] J. Misic, V. B. Misic, "Modeling bluetooth piconet traffic performance," *IEEE Commun Lett.*, vol. 7, no. 1, pp.18–20, Jan. 2003.

[6] D. Miorandi, C. Caimi, and A. Zanella, "Performance Characterization of a Bluetooth piconet with multi-slot packets," in *Proc. WiOpt*, Mar. 2003.

[7] G. Zussman, U. Yechiali, and A. Segall, "Exact probabilistic analysis of the limited scheduling algorithm for symmetrical Bluetooth piconets," in *Proc. IFIP PWC*, Sept. 2003, pp. 276–290.

[8] G. Zussman, A. Segall, and U. Yechiali, "Bluetooth time division duplex—analysis as a polling system," in *Proc. IEEE SECON*, Oct. 2004, pp. 547–556.

[9] D. Miorandi, A. Zanella and S. Merlin, "Mathematical analysis of the packet delay statistics in Bluetooth piconets under round robin polling regime," to appear in Med. J. Comp. Netw., 2006.

[10] S. Zürbe, "Considerations on link and system throughput of bluetooth networks," in *Proc. IEEE PIMRC*, vol. 2, 2000, pp. 1315–1319.

[11] N. Golmie, N. Chevrollier, and I. ElBakkouri, "Interference aware Bluetooth packet scheduling," in *Proc. GLOBECOM*, vol. 5, 2001, pp. 2857–2863.

[12] N. Golmie, R. E. VanDyck, and A. Soltanian, "Interference of Bluetooth and IEEE 802.11: Simulation modeling and performance evaluation," in *Proc. ACM MSWiM*, July 2001, pp. 11–18.

[13] A. Zanella, "On the impact of fading and inter-piconet interference on Bluetooth perfomance," in *Proc. IEEE WPMC*, Oct. 2002, pp. 218–222.

[14] C. M. Cordeiro, D. Sadok, and D. P. Agrawal, "Piconet interference modeling and performance evaluation of Bluetooth MAC protocol," in *Proc. GLOBECOM*, vol. 5, Nov. 2001, pp. 2870–2874.

[15] A. Conti, D. Dardari, G. Pasolini, O. Andrisano, "Bluetooth and IEEE 802.11b coexistence: Analytical performance evaluation in fading channels," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 2, pp.259–269, Feb. 2003.

[16] D. P. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall Inc., 1992.

[17] H. Takagi, *Analysis of Polling Systems*, MIT Press, 1986.

[18] D. Miorandi and A. Zanella, "Achievable rate regions for Bluetooth piconets in fading channels," in *Proc. IEEE VTC-spring*, vol. 5, May 2004, pp. 2610–2614.

[19] OPNET simulator. [Online]. Available: http://www.opnet.com

Soo Young Shin was born in 1975. He received his B.S., M.S., and Ph.D degrees in Electrical Engineering and Computer Science from Seoul National University, Korea in 1999, 2001, and 2006, respectively. After short post doc in Seoul Nation University, he joined the FUNLab at Univ. of Washington in July 2006 as a visiting scholar. His research interests include wireless LAN, WPAN, wireless mesh network, sensor networks, coexistence among wireless networks, and cognitive radio networks.

Hong Seong Park was born in Korea in 1961. He received his B.S., M.S. and Ph.D. degrees from Seoul National University, Korea in 1983, 1986, and 1992, respectively. Since 1992, he has been professor in the department of Electrical and Computer Engineering, Kangwon National University, Kangwondo, Korea. His research interests involve the design and analysis of communication networks and mobile/wireless communication, discrete event systems, and network-based control systems.

Dong-Sung Kim was received his Ph.D. degree in Electrical and Computer Engineering from the Seoul National University, Seoul, Korea, in 2003. From 1994 to 2003, he worked as a full-time researcher in ERC-ACI at Seoul National University, Seoul, Korea. From March 2003 to February 2005, he worked as a postdoctoral researcher at the Wireless Network Laboratory in the School of Electrical and Computer Engineering at Cornell University, NY. Since 2004, he has been an assistant professor in the School of Electronic Engineering at Kumoh National Institute of Technology, Gumi, Korea. His current main research interests are wireless industrial network, networked control system, and home automation system.

Wook Hyun Kwon was born in Korea on January 19, 1943. He received his B.S. and M.S. degrees in Electrical Engineering from Seoul National University, Seoul, Korea, in 1966 and 1972, respectively. He received his Ph.D. degree from Brown University, Providence, RI, in 1975. From 1976 to 1977, he was an adjunct Assistant Professor at the University of Iowa, Iowa City. Since 1977, he has been with the School of Electrical Engineering, Seoul National University. From 1981 to 1982, he was a visiting assistant professor at Stanford University, Stanford, CA. Since 1991, he has held the position of director of the Engineering Research Center for Advanced Control and Instrumentation. His main research interests are currently multi-variable robust and predictive controls, statistical signal processing, discrete event systems, and industrial networks.