Differentiated Priority Scheduling and Adaptive Segmentation for Bluetooth Piconets

Jyothsna Kalvala, Gergely Záruba
The University of Texas at Arlington
k_jyothsna@yahoo.com, zaruba@uta.edu

Abstract

Bluetooth is a wireless communication technology, aimed at supporting connectivity among close proximity mobile devices. Bluetooth enables the design of low-power, low-cost, and small-size radios. Bluetooth's MAC is a polling based protocol, where a central Bluetooth unit (master) determines channel access to all other nodes (slaves) in the network (piconet). One of the open research problems in Bluetooth is the design of efficient scheduling protocols. This paper proposes a polling policy that aims to achieve increased system throughput and reduced packet delays while providing reasonable fairness among all traffic flows, even in the presence of asymmetric traffic rates and unpredictable channel error conditions. Simulation results confirm that our proposed policy achieves higher throughput, lower packet delays with reasonable fairness among all the connections, compared to previous work in the literature.

1. Introduction

Bluetooth (BT) is a wireless technology with the original aim to eliminate cables between devices. BT is built on a fast frequency hopping (1600 hops/sec) physical layer operating in the 2.4 GHz frequency ISM band. The potential of this technology opened a wide range of applications; the three most popular usage scenarios being replacement for cables that are used to connect devices, universal bridging to connect data networks, and ad-hoc networking to provide a mechanism to form small personal area networks (PANs).

The smallest network unit formed among BT devices is called a piconet, which comprises of a master node and one or more slave nodes. During the process of the piconet establishment, the device that initiates a link connection with another device within its range takes the role of the master while the latter takes the role of a slave. Eventually, other slave devices may join the master and thus increase the size of the piconet. Exchange of information takes place only between the master and a slave (i.e., there is no direct slave-slave communication).

In BT, a virtual channel is defined by a random hopping sequence, determined by the master of the piconet (slaves are hop synchronized to that of the master and transmit data on the same frequency that the master hops to). All the stations within the piconet share the radio channel in a time division duplex basis, where the uplink and downlink transmissions between the master and each slave are alternated. The master can only start transmissions in an even-numbered slot while slaves may transmit data in the following odd-numbered slot if and only if they have been addressed in the previous slot. Thus, access to the medium is controlled by the master of the piconet, which schedules time slots among all the slaves within the piconet.

Bluetooth allows two types of virtual data communication links namely, Synchronous Connection Oriented (SCO) links for voice and Asynchronous Connection Less links (ACL) for data. Voice traffic (SCO links) is allocated reserved time slots while data traffic (ACL links) from different connections are scheduled based on a polling access scheme that is controlled by the master of the piconet. Figure 1 shows the TDD slots that are shared among the SCO voice traffic and the ACL data traffic from different master-slave connections.

The ACL data are classified based on the data rates that they carry, namely high data rate packets (also called DH packets) and medium data rate packets (also called DM packets). The packets are also classified into three types based on their length and these include the 1-slot, 3-slots and 5-slots long packets. The combination of these two classifications gives rise to six packet types which have been summarized in Table 1. The 2/3 FEC error-correction scheme is provided for the entire payload for DM packets, which helps to reduce the number of retransmissions that may occur due to wireless errors. However, in a reasonably error-free environment, FEC scheme creates unnecessary overhead thus reducing the throughput. DH packets do not use the 2/3FEC scheme in the payload. Both DM and DH packets employ an unnumbered ARQ (Automatic Repeat Request) scheme that allows data to be transmitted repeatedly from the sender to the receiver until an acknowledgement is received in the next scheduled slot.

The limitation of the BT polling scheme is that once a master polls a slave, the next slot is reserved for the slave irrespective of whether it has data to transmit or not, resulting in a pair wise scheduling of slots (i.e., the master-slave pairs). For such a system, the default Round

0-7695-2507-5/06/$20.00 (C) 2006 IEEE

Robin scheduler, suggested in the BT specification [2], is not suitable for the BT Piconet as it performs very poorly in the presence of asymmetric and heterogeneous traffic conditions.

Another issue that affects the performance is the presence of errors that are typically present on wireless channels. Errors may come as interference from other users using the same frequency band or as impairments such as multipath fading and shadowing from objects. Therefore an error-adaptive scheduling scheme is required that should allow the master to decide whether to use the more reliable, 2/3 FEC encoded DM packets, or the high data rate DH packets based on the channel error conditions.

The above two issues boil down to designing a scheme that employs an efficient scheduling scheme, which can predict the availability of data at the master and slave, thereby preventing wastage of slots and an adaptive packet selection scheme that can adapt the data transmission according to channel conditions, by choosing the correct packet types. This paper focuses on scheduling ACL data traffic in the presence of asymmetric and heterogeneous traffic and various wireless channel error conditions.

The rest of the paper is organized as follows: Section 2 introduces applying previous work in piconet scheduling. Section 3 introduces our proposed piconet scheduling scheme. Section 4 describes the simulation model used; while Section 5 presents the simulation results. Finally, Section 6 concludes the paper.

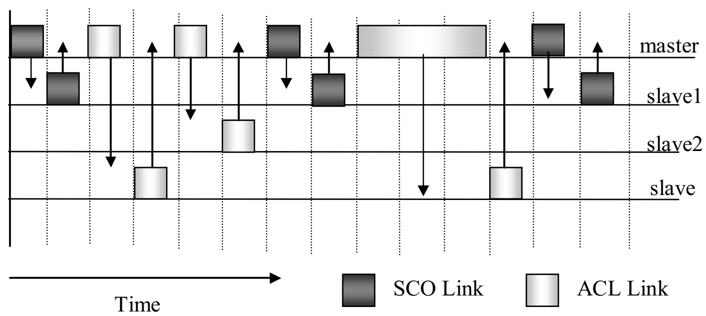

Figure 1. ACL and SCO Links.

Table 1. Bluetooth data packet types

Packet Type	Maximum payload [bytes]	FEC encoding	Maximum data rate [kbps]
DM1	17	Yes	108.8
DH1	27	No	172.8
DM3	121	Yes	387.2
DH3	183	No	585.6
DM5	224	Yes	477.8
DH5	339	No	732.2

2. Related Work

Previous work has been focused on designing efficient MAC scheduling and SAR policies in Bluetooth, which is motivated by the need to achieve maximum slot usage, minimum average packet delays, fairness among all the scheduled connections and adaptive usage of different packet types that are available as per the BT standard. The Pure Round Robin (PRR) scheme was the suggested default-scheduling algorithm for the Bluetooth MAC [2], which defines a fixed cyclic order in which every slave gets a chance to transmit one data packet, irrespective of whether they have data to transmit or not. Thus the limitation of this algorithm is that all the master-slave pairs at polled at the same rate, even under asymmetric traffic (data arrives at different rates at different master-to-slave and slave-to-master queues) conditions, eventually wasting slots with connections not having any backlog.

In [5] and [6], the authors suggest a master-slave queue-state dependent packet scheduling policy that assigns a priority to every master-slave pair based on their queue status. An unused bit in the payload header is used which is set to indicate the presence next available data in the queues and vice versa. A higher priority is assigned to the pair that utilizes slots more efficiently than the pairs that do not and is hence polled more often than the latter. The K Fairness policy [5] (KFP) is one such policy, where a counter is maintained for every master-slave pair and when a lower priority sacrifices its service to a higher priority pair, the counter is decreased by one for the former and increased by one for the latter. This sacrifice of service is allowed only until the difference between the maximum and minimum counter values lies within a threshold K. Once this threshold value is reached, no more service-sacrifice is performed and Round Robin scheduling is resumed.

However, the limitation of this algorithm is that when the threshold is attained, services are no longer sacrificed to higher priority pairs, and the Round Robin scheme is resumed once again, thus leading to degraded performance after the threshold is reached. Thus the performance of this algorithm is limited by the maximum and minimum counter values. In order to overcome this limitation, the authors in [11] suggest the Differentiated K Fairness Policy (Diff_KFP) \which compares the counter values for the two pairs that are involved in transferring and receiving the sacrificed service, instead of comparing the highest and lowest counter values among all the pairs. If the difference is less than K, then the sacrifice is performed, otherwise the lower priority pair is polled as per schedule and its counter value is increased by one. The authors suggest that QoS can be incorporated in this algorithm by incrementing the counter of the lower priority pair by a step-size, when the threshold is reached. However, the current work does not focus on QoS and hence the step-size is set to one. This policy attains higher throughput and decreased packet delays compared to the K-fairness policy. However, the limitation of the KFP and the Diff_KFP schemes is that they do not consider the presence of errors on the wireless channel, which affect the performance of the system substantially.

In order to provide error protection, this scheme has been modified to an error adaptive version, according to which the service is transferred by a master-slave pair that encounters errors to any other pair such that the difference between the maximum and minimum counter values among the pairs is less than the threshold value K. The limitation of this scheme is that there is no predictive analysis about the occurrences of errors on the channel. The service is simply transferred to any other pair that satisfies the constraint for K value. However, there is no guarantee that this pair does not have any errors on its channel. Additionally, on encountering an error at one connection, the service is transferred to another connection, without even checking if the second connection has data to transmit. This will result in inefficient slot usage. Further, the scheme does not make use of error correction schemes provided by the Bluetooth standard, like the FEC and ARQ schemes to handle the error conditions. These may be more appropriate solutions to compensate for channel errors.

The segmentation and reassembly scheme incorporated in the Bluetooth MAC allows the data packets to span over 1, 3 or 5 slots in length. The manner in which the data is segmented into these packet types, greatly affects the efficiency of the system. This is because each packet type has its own performance characteristics in terms of data rates and error correction schemes and hence the SAR policy plays a vital role in achieving high utilization efficiency.

Kalia et al. have suggested SAR policies for Bluetooth in [5], which include the Random SAR policy, Batch SAR policy, Intelligent SAR policy and SAR with partial reordering. The Random and Batch SAR policies are static policies and do not adapt to the changing traffic rates or error rates on the channel. The Intelligent SAR policy introduces the idea of adaptive segmentation by dynamically segmenting the datagram packets into different multi-slot baseband packets, based on the varying data rates at the master and slave ends. According to this scheme, initially, all the queues have a baseband packet size equal to one-slot length. If the data rates at both the master and slave ends have the same arrival rate (high or low), the slot size is maintained at one, so that both ends are served at an equal rate. However, if one of the ends has a high data rate while the other end has a low data rate, large MAC packets are used for the side with a high data rate and small packets are used for the side with low rates. In order for the SAR at the master to know the data rate at the slave's end and vice versa, this policy suggests the use of a single reserved bit in the data packet to convey this information. The bit is set to 1, when the rates are very high and set to 0 when they are very low.

The limitation of this work is that the segmentation scheme does not consider the presence of error conditions and the packet type produced based on data rates may not be suitable for transmission under channel error conditions. For instance, in a case where the data rates are very high and a large MAC packet is used as suggested by the algorithm, the packet will most likely be hit by an error, under bad error conditions, and this will lead to multiple retransmissions. This would increase the overall delay and eventually waste time slots in retransmissions. Thus, efficient slot utilization is not achieved and this leads to degradation in system performance.

In [4], SAR – Best Fit and SAR – Optimum Slot Utilization have been suggested, which also do not consider the presence of error conditions on the wireless channel and hence are not very appropriate segmentation schemes under such conditions. The error adaptive segmentation scheme has been suggested in [8], according to which the datagram packets are segmented dynamically based on the error conditions on the channel, such that minimum number of retransmissions takes place. By doing this for every pair, an overall decrease in packet delays and increase in network throughput is achieved. This scheme is based on the fact that a large packet has low overheads and is very advantageous to use when the error rates on the channel are very low. On the other hand, small packets have a low packet error rate and so are advantageous to use when the channel error rates are very high. Therefore, there exists a threshold where the smaller packet error rate of a small packet outweighs the benefit of efficiency of a big packet. Based on this principle a finite state machine has been derived that suggests the most suitable packet type to be used based on the packet error rates (Figure 3). This model assumes a uniform bit error model that allows one to interpolate the packet error rates of different packet sizes so that the best packet type can be determined. Moreover, only high data rate packets

(DH5, DH3 and DH1) have been used in this model. However, it has been suggested that the model may be extended to incorporate medium rate packets as well.

The decision about the packet type to be used depending on the channel error conditions is made based on the calculated packet error rate. The DH packets have are said to be unsuccessfully transmitted if even a single bit error occurs. If PER(X) represents the packet error rate of the DHx (x = 1, 3 or 5) packet and BER represents the channel bit error rate, then,

PER(X) = 1-(1-BER)bps(X)

If N represents the number of transmissions of a packet before a successful transmission,

$$N = \frac{1}{1 - PER}$$

Therefore the effective bandwidth R of a packet type DHx can be represented as,

$$R = \frac{K}{(DN)(625x10^{-6})} bps$$

where K represents the number of data bits in the packet, D represents the number of slots occupied by the packet type DHx. K/(D*625) represents the efficiency of the packet type used for transmission (Table 2).

R = (1-PER) x efficiency x bandwidth

From the equations above it can be inferred that the effective bandwidth of a packet type DHx can be calculated based on the bit error rate (Figure 2).

This adaptive segmentation scheme has not been suggested to be employed with a scheduling scheme in the presence of real-life traffic conditions. Therefore we propose that a combination scheme that includes the Differentiated K fairness scheduling policy and the adaptive segmentation scheme would achieves higher system efficiency and lower average packet delays compared to each of these schemes employed individually in the presence of asymmetric traffic and channel-error conditions. In order for the adaptive segmentation scheme to work dynamically with the scheduling scheme in the presence of changing traffic and error conditions we suggest a few modifications to each of these schemes.

Table 2. Efficiency of DH packets

Packet type	Efficiency [%]
DH1	0.17
DH3	0.59
DH5	0.72

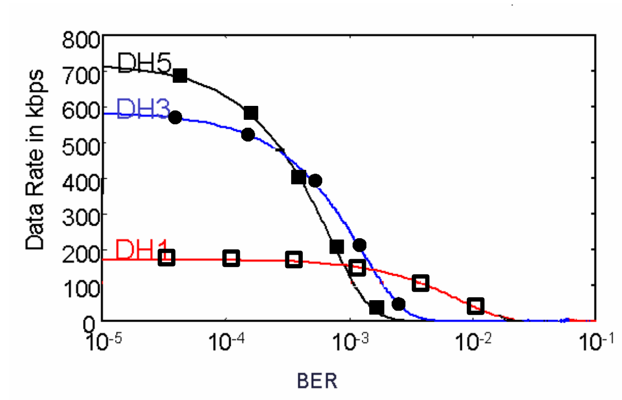

Figure 2. Effective Bandwidth of packets vs. BER

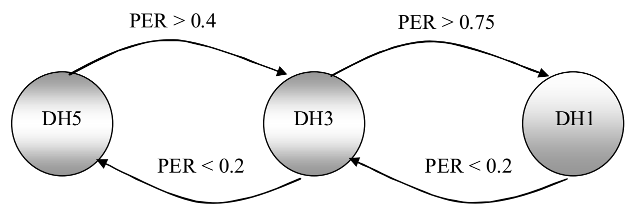

Figure 3. FSA for adaptive SAR

3. Proposed Piconet Polling Protocol

The Diff_KFP scheme schedules the pair to be polled based on the queue status at the master and slave. We assume that three types of priorities are assigned to the master-slave pairs based on the presence or absence of data at the respective queues. These have been summarized in Table 3.

The 0-0 pairs are polled every threshold number of cycles to check for any change in the respective master and slave queue status. This polling is done even if the threshold value K is not reached. Once the pair to be polled is picked, the adaptive segmentation scheme is employed so that the most appropriate baseband packet is used based on the channel error conditions. Currently, this work assumes the presence of only the high data rate packets, namely, DH1, DH3 and DH5.

Our proposed scheme defines CS_i, as the channel status, which represents the maximum allowable packet length that may be transmitted on the channel between the i^{th} master-slave pair, for the given error conditions. Therefore, CS_i can assume one of the values in {DH1, DH3, DH5}. CS_i is calculated based on the Packet Error Rate (PER), as explained by the FSA shown in Figure 3.

Packet Error Rate is defined as the ratio of number of attempts before a successful transmission to the total number of attempts until a successful transmission.

$$PacketErrorRate(PER) = \frac{Numberoftrialsuntilsuccessfultransmission - 1}{Numberoftrialsuntilsuccessfultransmission}$$

Table 3. Priorities of master-slave queue conditions

Priority	Queue status	Description
1	1-1	Data present at master-to-slave queue and slave-to-master queue
2	1-0	Data present at the master-to-slave queue, no data present from slave-to-master queue
2	0-1	No data present at master-to-slave queue, data present at the slave-to-master queues
3	0-0	No data present at master-to-slave queue, no data present at slave-to-master queue

Since the error conditions change dynamically on a channel, the PER needs to be calculated periodically on every connection so that the value of CS_i can be increased when the channel goes into a good state (low Bit Error Rate) and decreased when it goes into a bad state (high Bit Error Rate). These two processes have been defined as the Stepping Up process and the Stepping Down process respectively. The goal of both the processes is to minimize the number of retransmissions (hence the average packet delay) and maximize the throughput in the piconet under the given error conditions. When the channel goes from a good state to a bad state, the Stepping Down Procedure is implemented where the master observes the increase in PER for the current packet type. Once a threshold PER value reached, the next smaller packet type is chosen for data transmission. This reduces the chances for this packet type to get hit by an error. On the other hand, when the channel goes from a bad state to a good state it is difficult for the master to deduce this situation. This is because when the packet gets successfully transmitted, the master does not know if it is because the current packet type is suitable for the current error conditions or if the channel is going into a good state. This is unlike in the stepping down procedure, where it can be easily deduced when the channel is going from a good state to a bad state as indicated by an increase in PER. Therefore, the master periodically performs the Stepping Up procedure for every master-slave pair and this period is defined by the number of slots passed since the pair was last polled. Afterwards, the master transmits packets continuously to the same slave until the master is able to deduce the channel error conditions for this pair by calculating the PER.

The above algorithm helps calculating CS_i, which represents the largest packet type that may be transmitted on the channel with the least number of retransmissions, for the given error conditions. However, sometimes, the data at the head of the queue may contain much less data than what can be carried as indicated by CS_i. In such a case, the following SAR scheme is employed, which we name as the random segmentation scheme:

1. if HOL_data_packet_size < 27 bytes , use DH1
2. if 27 bytes < HOL_data_packet_size < 183 bytes, use DH3
3. if 183 bytes < HOL_data_packet_size, use DH5.

Therefore, packet type used to transmit the data from the master to the slave, defined as m_packet_length, is min(CS_i, HOL_data_packet_size).

The above algorithm is implemented at the master's end to deduce the suitable packet type for transmitting data from the master to the slave. However, the slave is unable to perform the same calculation while transmitting data from its end to the master. This is because, calculation of PER requires subsequent retransmissions of data packets, until a successful transmission takes place. However according to the BT specification, subsequent retransmissions are not allowed from the slave's side. Therefore, when an error occurs during data transmission from the slave, this baseband packet is retransmitted only when it is polled again. As a result, the slave is not able to deduce the correct packet type based on errors on the channel. Taking this into consideration, we propose that the slave can use the same packet type that was transmitted from the master, defined by m_packet_length. However, as explained in the above algorithm, the HOL packet size may be smaller than the m_packet_length and so the slave deduces the correct packet type as min(m_packet_length., HOL_packet_size (at the slave)).

4. Simulation Model

In order to obtain results for our proposed algorithm and to compare with the previously suggested algorithms in literature, we used a discrete event simulator we have developed in C++. All simulations were performed on a piconet consisting of six slaves and a master. For all measurement points, enough simulations have been run to claim a 95% confidence that the average result shown has less than 5% error.

4.1 Data Traffic Model

Data traffic was generated independently for each M-S pair in each direction according to a Poisson process. The length for the datagram packets was also assumed to be I.I.D. exponentially distributed with average packet size of 500 bytes. The load in the piconet was varied by varying the inter-arrival times for the packets. For example, high load is simulated by reducing the packet inter-arrival time and low load is simulated by increasing the inter-arrival time.

4.2 Correlated Fading Channel Model

Based on the observations reported earlier, we model the Bluetooth RF link as a 2-state Markov Chain [1][7][9][12]. Since every wireless connection between the master and slave is an independent channel with its own time-varying error distribution [10], this scenario was simulated using a 2-state Markov model on each master-slave connection. According to the 2-state Markov model, at any instant, the channel can be in any one of the two states, namely, good state or bad state. The bit error rate (BER) is low in the good state and high in the bad state. However, errors occur uniformly in each of these states. Figure 4. describes the two states in the 2-state Markov model.

The time spent in each of these good and bad error states in exponentially distributed with different rates of state transitions $\mu_G=1/P_G$ and $\mu_B=1/P_B$. In our current simulation, we used values from [3]; $P_G = 435$ ms and $P_B = 55.8$ ms. The BER values were simulated in orders of 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3} and 10^{-2}. The threshold value, K for the Differentiated K-fairness scheduling scheme was assumed to be equal to 300 and the threshold number of slots since last poll was taken to be 100 slots before performing the Step Up process for every master-slave pair.

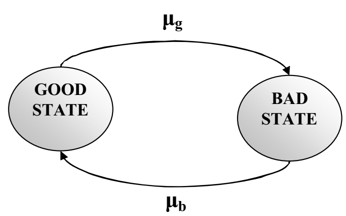

Figure 4. Markov model of wireless channel errors

4.3 Algorithms Implemented

We have implemented and compared five algorithms namely, i) differentiated scheduling with adaptive segmentation scheme; ii) Round Robin scheduling with adaptive segmentation scheme; iii) wireless KFP scheduling with random SAR; iv) KFP scheduling with random SAR and v) Round Robin scheduling with random SAR scheme. Each of these policies was implemented with full compliance to the BT ARQ scheme.

4.4 Performance Metrics

The algorithms have been compared in terms of three metrics namely, utilization, delay and fairness. Utilization was measured by comparing actual throughput to the 1Mbps total channel capacity. Delay has been measured in the form of two different parameters namely: average end-to-end packet delay, also called the average waiting time per packet and average segment delay. Fairness has been calculated in terms of throughput and delay, which is used as a measure to calculate the bandwidth and delay equality between connections. Assuming that there are n individual connections (these connections include master-to-slave and slave-to-master connections, each considered separately), and T_i is the throughput for the i^{th} connection, the fairness in throughput is calculated using the formula in Equation 1.

$$\text{Fairness } f(T_1, T_2 \ldots T_n) = \frac{\left(\sum_{i=0}^{n} T_i\right)^2}{n\sum_{i=0}^{n} (T_i)^2} \qquad (1)$$

If the connections share the bandwidth equally then f=1, else it is less than 1. Fairness in delay is calculated according to Equation 2.

$$\text{Fairness } f(D_1, D_2 \ldots D_n) = \frac{\left(\sum_{i=0}^{n} D_i\right)^2}{n\sum_{i=0}^{n} (D_i)^2} \qquad (2)$$

where D_i represents the average packet delay on every connection, between the master and each slave.

Yet another comparison is performed in terms of the effect of variation in the distribution of load (for a particular load value) on the channel utilization and average end-to-end packet delays. Coefficient of variation of a random variable can be defined as the ratio of standard deviation to the mean of the distribution – Equation 3. Using COV as a load measure we can easily depict heterogeneous traffic. These comparisons were performed for a high load of 0.84Mbps and a low load of 0.192Mbps

$$\text{COV} = \frac{\sum_{i=0}^{n} \lambda_i^2 - \frac{1}{n}\left(\sum_{i=0}^{n} \lambda_i\right)^2}{\sum_{i=0}^{n} \lambda_i / n} = \frac{StandardDeviation}{Mean} \qquad (3)$$

5. Simulation Results

Figure 5 shows the comparison of the five target algorithms in terms of channel utilization with varying system load. It can be seen that our proposed combination scheme outperforms the other four algorithms, while the Round Robin scheduling policy with adaptive segmentation performs next best. This indicates that the error adaptive packet segmentation is one of the most important factors that determine the performance of a system. The wireless adaptive K fairness scheduling scheme with random segmentation on the other hand does not perform very well, which is due to two reasons, i) when the threshold value, K, is reached, the Round Robin scheme is resumed, which is an inefficient scheduling policy under asymmetric traffic conditions; ii) random SAR policy does not suitably handle error conditions further adding to the degradation in performance. According to this scheme error handling is performed by transferring service from an error hit connection to another connection provided the threshold value is not reached. However, this is an inefficient scheme because, in a scenario where most of the connections are in a bad error state, lot of slots would get wasted in trying to transfer the service to a connection that is in good error state (less BER). This only decreases the overall throughput and increases the waiting time of packets. The Differentiated K fairness scheduling with random SAR policy, also shows degraded channel utilization. Though the differentiated scheduling scheme avoids slot wastage by polling the higher priority pairs more often than the lower priority pairs, the random SAR scheme leads to very poor system throughput in the presence of wireless channel conditions. Finally, the Round Robin scheduling with random segmentation shows the most degraded performance since it neither provides a mechanism to avoid polling pairs that do not have data to transmit nor a mechanism to avoid retransmissions due to errors.

Figure 6 shows the average end-to-end packet delay versus a varying channel load. It can be seen that proposed scheme leads to the least average packet delay compared to the other algorithms. The differentiated KFP scheme reduces delay by polling those pairs that have data to be transmitted, unlike the Round Robin scheme which keeps these pairs waiting though they have data to transmit. Furthermore, the adaptive segmentation scheme reduces the number of retransmissions, and consequently minimizes the overall delay in the system.

Figure 7 compares the average segment delay versus varying channel load in the five algorithms. It can be observed that the average segment delay is the lowest for the differentiated K Fairness scheduling with adaptive segmentations scheme. Since the segmentation scheme chooses the most appropriate segment type, depending on the channel error conditions, it reduces the error hit ratio of the segments. Furthermore, the differentiated K Fairness scheme handles those pairs more often that have data to transmit and thus decrease the delay. Extending the logic of this reasoning, the other policies do not perform as well as this scheme.

Fairness in throughput versus the channel load is compared in Figure 8. It can be seen that the differentiated KFP with adaptive segmentation scheme provides reasonable fairness though it does not provide the maximum fairness. This is because factors that enable high throughput oppose those that enable high fairness and hence both the goals are not achievable simultaneously. It can be seen that the Round Robin scheduling scheme provides the highest fairness in terms of throughput at very high loads. This is because all the pairs will have data to transmit and each pair gets an equal opportunity for data transmission by getting polled in a sequence.

Figure 9 shows a comparison of fairness in average packet delay versus the channel load. It can be seen that proposed scheme performs reasonably fair in terms of average packet delay though it does not provide the best fairness index. The reason is similar to what has been explained for fairness in throughput.

The effect of variation in channel load (high channel load = 0.84Mbps) on the system throughput and average packet delay have been compared for the five algorithms in Figure 10 and Figure 11 respectively. From Figure 10, we can deduct that the Round Robin scheduling with random SAR and Round Robin scheduling with adaptive SAR starts to become unstable at high COV values, while the other three schemes perform almost steadily for all COV values. According to Figure 11, the delay performance of Round Robin scheduling with random segmentation becomes increasingly unstable for higher COV values. The other four algorithms on the other hand show a relatively stable performance for all COV values. The figures show that Round Robin scheduling scheme is not a suitable under high variations in traffic conditions.

Figure 12 and Figure 13 plot similar curves to Figure 10 and Figure 11 except they are showing results for low overall channel load (0.192Mbps). These figures second the findings described in the previous paragraph.

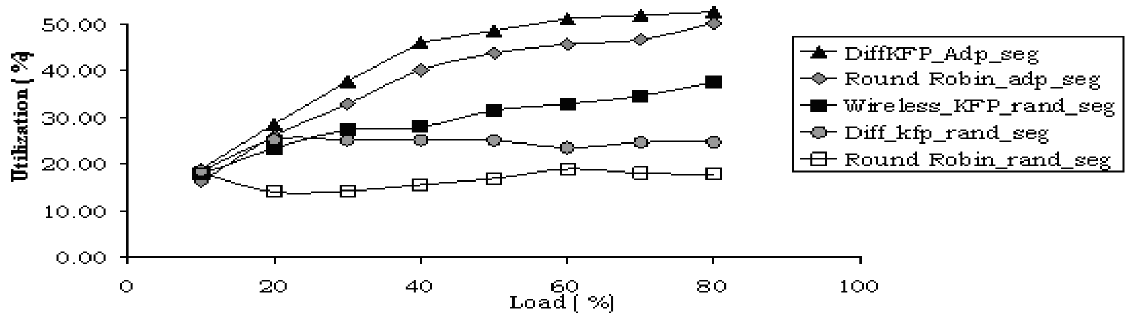

Figure 5. Channel utilization vs. load.

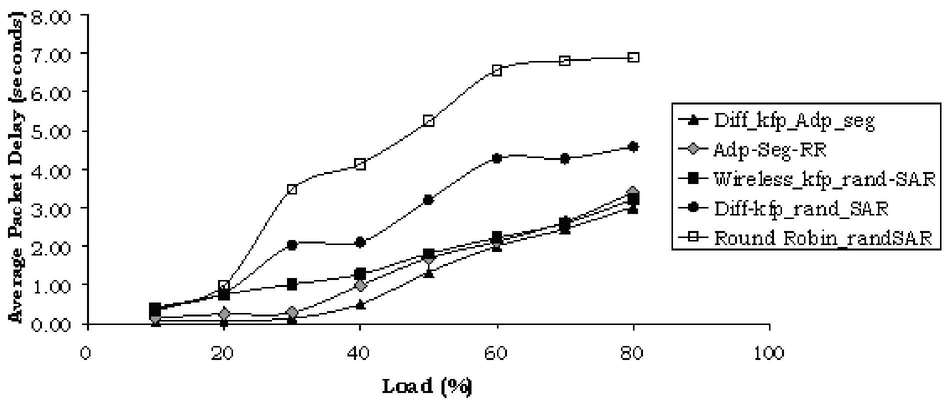

Figure 6. Packet delay vs. load.

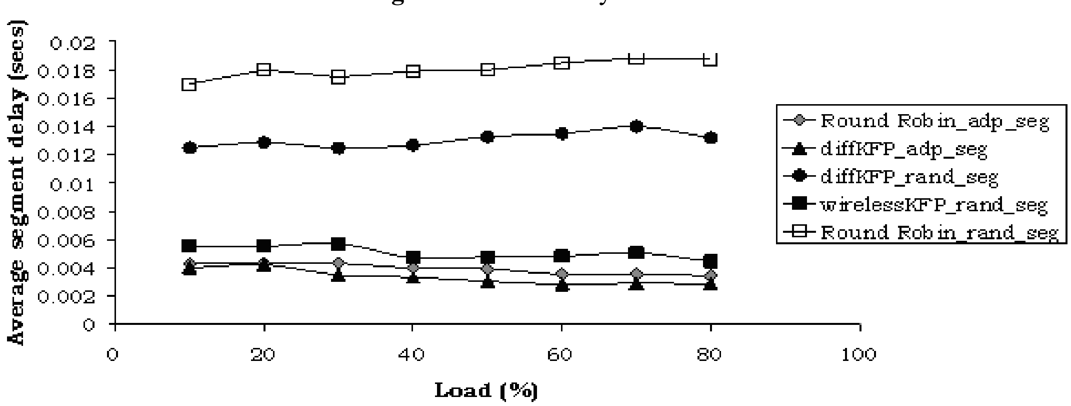

Figure 7. Segment delay vs. load.

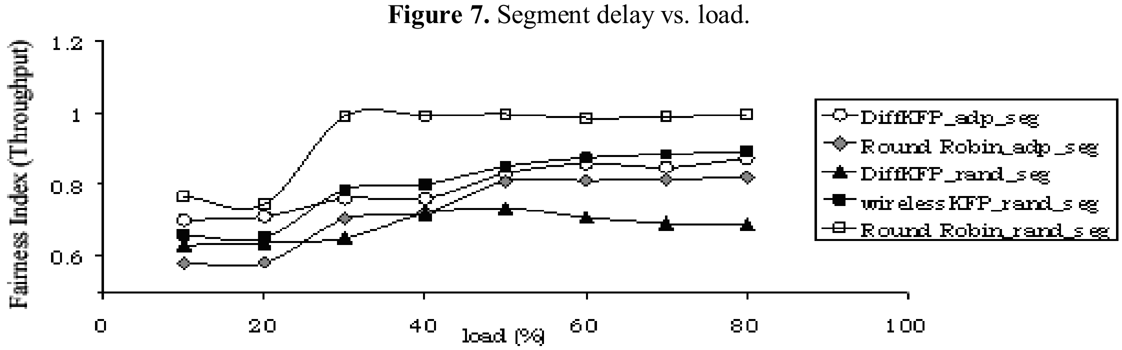

Figure 8. Fairness in throughput vs. load.

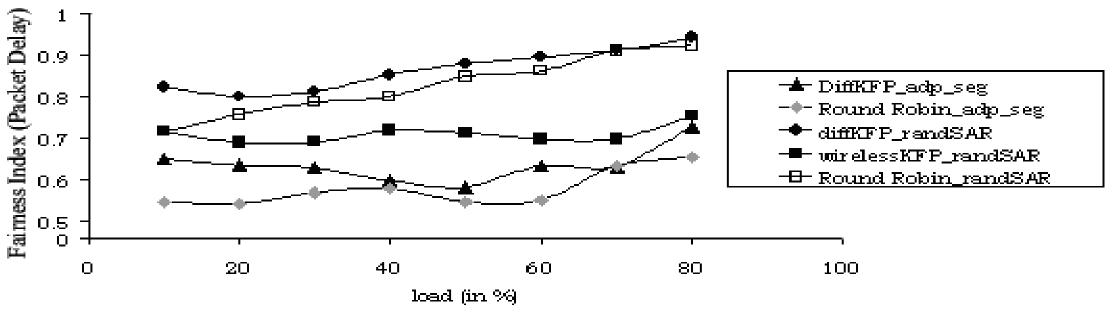

Figure 9. Fairness in packet delay vs. load.

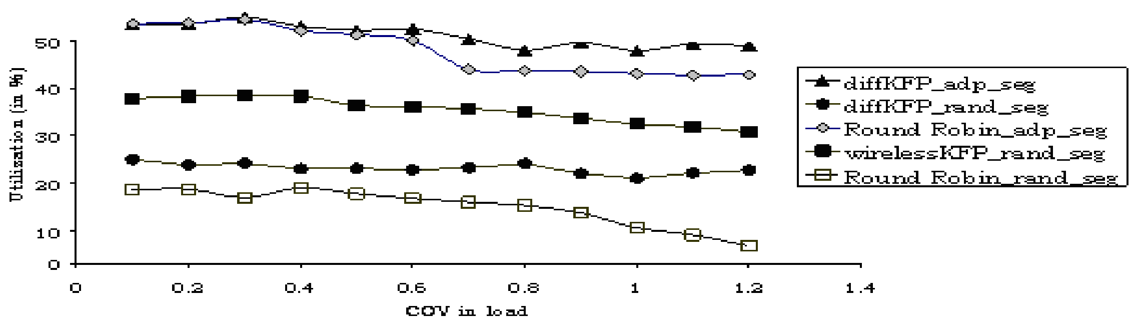

Figure 10. Fairness in throughput vs. COV in load (high load of 0.84Mbps).

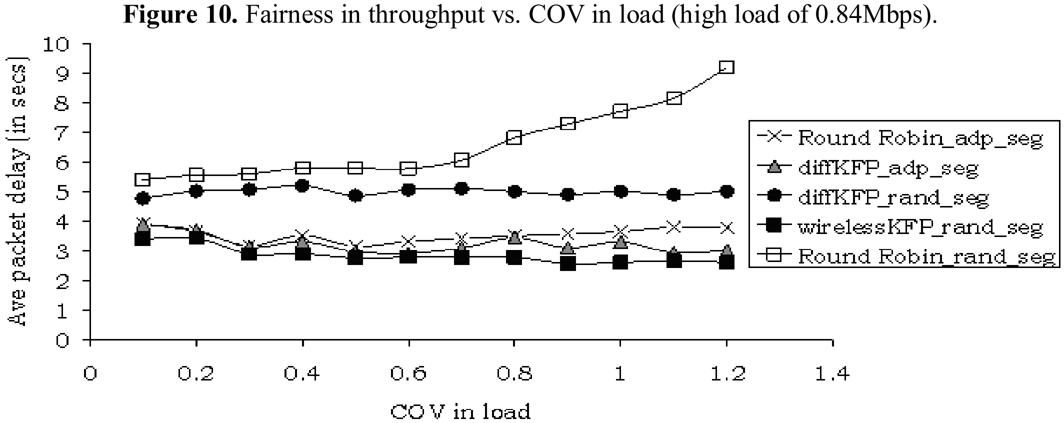

Figure 11. Average packet delay vs. load (high load of 0.84Mbps).

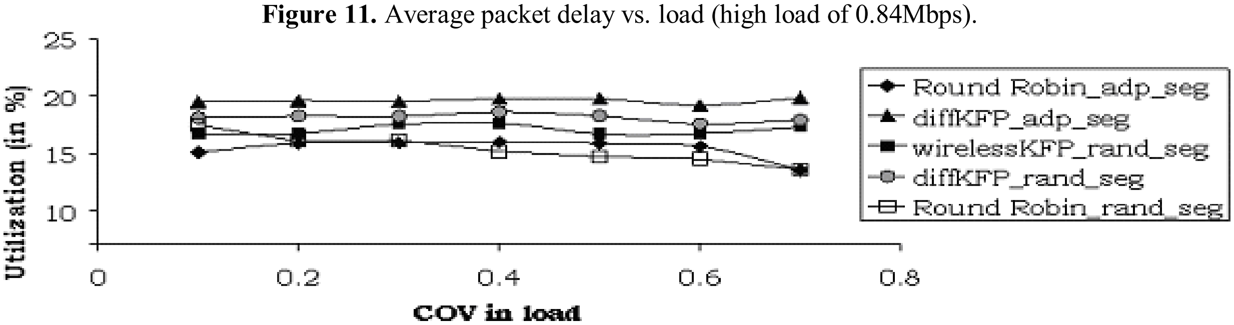

Figure 12. Fairness in throughput vs. COV in load (low load of 0.192Mbps).

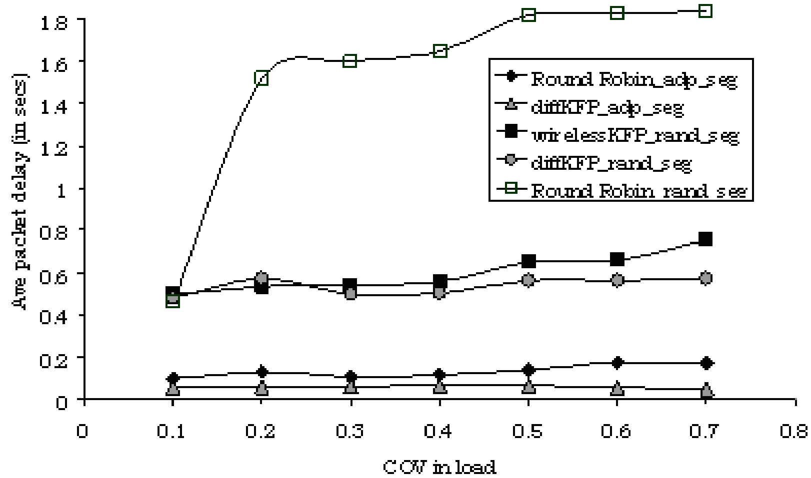

Figure 13. Average packet delay vs. load (low load of 0.192Mbps).

6. Conclusions and Future Work

In this work we have suggested a piconet scheduling scheme for the Bluetooth MAC that aims at increasing throughput, decreasing average end-to-end packet delay and providing reasonable fairness among all the connections, achieving all these in the presence of asymmetric traffic conditions and wireless channel errors. Our proposed algorithm employs a differentiated K fairness scheduling policy combined with an error adaptive segmentation scheme that enables good system performance by minimizing slot wastage. The differentiated K fairness scheme uses the TDD slots effectively by polling those pairs more often that have data to transfer, while the adaptive segmentation scheme does the same by segmenting the data into suitable packet types so that minimum number of retransmissions takes place in the presence of error conditions. Our simulations have shown that the proposed scheme adapts better to the changing traffic conditions and channel error conditions, than previous algorithms reviewed.

In the current work, we have considered only the presence of ACL traffic. More specifically, we have considered only DH packets; the scheme can be extended to include DM packets as well by modifying the FSA used to deduce the packet types. The suggested adaptive segmentation scheme can be extended by choosing the suitable packet type not only based on the deduced packet error rates but also based on QoS requests, data rates arriving at the master and slave ends and the number of slots available between the reserved SCO slots. Our work considers only errors in the payload and does not consider errors in the packet access code and packet header. The finite state scheme can be modified by considering these as well.

References

[1] H.Balakrishnan, V.N. Padmanabhan, S.Seshan and R.H. Katz, “A comparison of mechanism for improving TCP performance over wireless links”, IEEE/ACM Transactions on Networking, vol. 5, no.6, pp.756-769, Dec. 1997.

[2] Bluetooth Special Interest group, “Specification of the Bluetooth System,” 1.1b, vol. 1 and 2, Feb 2001.

[3] A. Das, A. Ghose, R. Shorey, “Adaptive Link-Level Error Recovery Mechanisms in Bluetooth,” Proceedings of IEEE International Conference on Personal Wireless Communication, pp.85-89, 2000.

[4] Das, A. Ghose, R. Shorey, “Enhancing Performance of Asynchronous Data Traffic over the BT Wireless Ad Hoc Network”, IEEE INFOCOM, pp. 591-600, April 2001.

[5] M. Kalia, D.Bansal, and R.Shorey, “Data Scheduling and SAR for Bluetooth MAC,” Proceedings of the IEEE VTC 2001, vol. 2, pp. 716-720, Tokyo, Japan, May 2001.

[6] M. Kalia, D. Bansal, R. Shorey, “MAC Scheduling and SAR Policies for Bluetooth: A master Driven TDD Pico-Cellular Wireless System,” 6th IEEE International Workshop MOMUC, Nov 1999.

[7] R.H. Katz, “A trace Based Approach for modeling wireless channel behavior”, ACM SIGCOMM, France, Aug. 1997.

[8] J. Kim, Y. Kim, J.S. Ma, “An Adaptive segmentation scheme for the Bluetooth based wireless channel,” Proceedings of IEEE IC3N, pp. 440-445, 2001.

[9] A. Kumar, “Comparative Performance Analysis of versions of TCP in Local Network with a Lossy link,” IEEE Trans. on Networking, vol. 6, no.4, pp.485-498, Aug. 1998.

[10] D.Moldkar, “Review on radio propagation into and within buildings”, IEEE Proc-H, vol. 138, no.1, Feb 1991.

[11] J.S. Oh, Y. Kim, Y. Joo, O.Kwon, K.H. Tchah, T. Lee, “Differentiated Fairness Guaranteeing Scheduling Policies for Bluetooth,” Proceedings of IEEE 56th VTC, Sept. 2002

[12] H.Wang and N. Moayeri, “Finite State Markov channel - a useful model for radio communication channels”, IEEE Trans. Vehic. Technology, vol. 44, pp-163-171 Feb. 1995