# Data Scheduling and SAR for Bluetooth MAC

Manish Kalia, Deepak Bansal, Rajeev Shorey
IBM India Research Laboratory,
Block 1, Indian Institute of Technology,
Hauz Khas, New Delhi 110016, India.
Email: srajeev@in.ibm.com, Phone: 91-11-6861100, Fax: 91-11-6861555

*Abstract*—**Motivated by the emerging standards for indoor pico-cellular wireless systems, such as the Bluetooth, we propose and study the scheduling policies for Master driven Time Division Duplex (TDD) Wireless Networks. In these networks, the frequency band is divided into time slots, and each end (i.e., Master or Slave) takes turns in using the time slots. In Bluetooth, a Slave transmits packets in the reverse slot only after the Master polls the slave in a forward slot (by sending data to it). The conventional scheduling policies such as Round Robin do not perform well in these systems as they are not suited to the tight coupling of uplink-downlink. We propose new scheduling policies, (i) the Priority scheme, and, (ii) the K-Fairness scheme that utilize the state at the Master and Slaves to schedule the TDD slots effectively. Active slaves are differentiated based upon the binary information (i.e., the presence or absence of packets in a slave queue) about the Master-Slave queue pairs. The priority scheme achieves high throughput as compared to the Packet-by-Packet Generalized Processor Sharing (PGPS) based policies while guaranteeing a minimal service to each active Slave while the K-Fairness policy is characterized by a tight fairness bound as well as high system throughput. We then extend these policies for scheduling variable size data in the presence of voice. Further, since Bluetooth supports variable size data packets (1, 3 or 5 slots) on the same connections, the Segmentation and Reassembly (SAR) can significantly impact scheduling of data packets by varying packet size distribution. We propose an Intelligent SAR policy (ISAR) and compare it with the naive Random-SAR in which the data packet sizes (i.e., 1, 3 or 5) are assigned probabilistically. ISAR adapts MAC packet size at the Master and Slave queues depending on the data arrival rates at both the queues.**

**Keywords**: Time Division Duplex (TDD), Bluetooth, Medium Access Control (MAC), Fairness, Throughput, Scheduling, Round Robin (RR), Discrete Event Simulation (DES), Segmentation and Reassembly (SAR)

## I. INTRODUCTION

As we see an explosion of activities in the area of wireless networks, it is desirable to see a wireless solution that brings all the multiple technologies/standards in different sectors together and at the same time provide a universal and ubiquitous connectivity solution between computing, communication and supporting devices. Bluetooth is an effort to realize this vision [1], [2].

In this paper we propose and study Data Scheduling and Segmentation and Reassembly (SAR) policies for Bluetooth Media Access Control (MAC). Bluetooth is an emerging standard for low-power, low-cost indoor pico-cellular wireless systems. It is a Master driven Time Division Duplex (TDD) system ([4], [1]) in which each pico-cell has a Master-Slave configuration. A Slave transmits packets in the reverse slot only after the Master polls the slave (or transmits a packet to the Slave) in a forward slot. The Master transmits packets to a Slave in even slots while the slave transmits packets to the Master in an odd slot. As a result scheduling occurs in pairs of slots (i.e., the Master-Slave pair). Further, the task of scheduling is vested with the Master. This mode of operation makes multiple access straightforward, since the Master provides a single point of coordination. Thus, Bluetooth is a Master driven TDD standard.

However, Master driven TDD scheduling poses several challenges since the traditional scheduling policies do not perform well over this kind of a MAC [4]. Once a Master polls a Slave, the next slot is reserved for the Slave irrespective of whether the Slave has data to send or not. An efficient scheduling policy would depend upon the (i) state of the queues at the Master and at the Slaves, (ii) the traffic arrival process at these queues, and (iii) the packet length distribution at the Master and the Slave. It is with this in mind that we develop scheduling policies adapted for TDD MAC at the Master node in the Bluetooth system. The parameters of interest that we study are the system throughput, packet delays, fairness and the packet drop probability. It is to be noted that the twin objectives of throughput and fairness can conflict since the throughput might have to be sacrificed for fairness and vice-versa [4], [3]. Further, the time frame of interest over which to provide fairness needs to be selected carefully.

We do not model the wireless link failures since, our aim in this paper, is to understand and propose good scheduling policies for TDD based MAC protocols. It is known that wireless link failures due to Rayleigh fading can be modeled as a two-state Markov Chain [3]. In [8], we have incorporated wireless link failures and studied the performance of our policies in the presence of wireless errors. In [8], we also analyze the utility of having more information about Master-Slave queue pairs rather than just binary information (i.e., presence or absence of backlog at the FIFOs). We show that binary feedback is an effective, simple and easy to implement solution.

Fairness in wireless systems has been extensively studied in [6], [5]. The authors in [3] have described how combining Class Based Queueing and Channel State Dependent Packet Scheduling can enable controlled wireless link sharing. The policies that have been proposed in these papers can be combined with the scheduling policies that we propose in order to get efficient scheduling policies for Master driven TDD systems such as the Bluetooth system.

0-7803-5718-3/00/$10.00 ©2000 IEEE.

Further, we study the effect of voice on data scheduling and how SAR can be made intelligent to improve the performance of data scheduling in the presence of voice.

The paper is organized as follows. In Section II, we study the Bluetooth Master Driven TDD MAC protocol. We then introduce two scheduling policies: Priority and K-Fairness. In Section III, we study the performance of the data traffic in the presence of voice in the Bluetooth system. We propose extensions to the Prority and K-Fairness policy for scheduling data traffic in presence of voice. We also present our Intelligent SAR policy in section III. Section IV describes the simulation and results. Finally, in Section V, we present the conclusion.

## II. MEDIUM ACCESS CONTROL IN BLUETOOTH

Bluetooth uses a TDD slot structure for resolving contention over the wireless links. There is a strict alternation of slots between the Master and the Slaves. This implies that once a forward slot is assigned to a Master, the following reverse slot is assigned to the corresponding Slave. Thus, scheduling occurs in pairs of slots (i.e., the Master-Slave pair). Further, the task of scheduling is vested with the Master. This makes Bluetooth a Master driven standard.

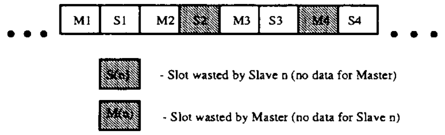

Fig. 1. Slot wastage in TDD

As shown by Figure 1, it can be seen that there could be a wastage of slots in the TDD scheme, since if only one of the Master or the Slave has data to send, a slot gets wasted. Further, the fairness issue is complicated since service may be given to a Master-Slave connection[1] even though it is not backlogged (i.e., Master and Slave do not, at the same time, have data to send). Due to above reasons, policies like round-robin (RR) [4] which have been studied extensively in the literature yield low throughput with TDD based MAC protocols, and may not ensure fairness.

In view of the above problems, new and efficient scheduling policies are required which tackle the additional constraints imposed by the Master driven TDD structure. These policies should be simple to implement (such as the round robin policy) in order to satisfy the low cost objective of the Bluetooth standard [1]. Note that throughout this paper, we assume that if there are N Slaves in a picocell, there are N queues at the Master; for each Slave, there is a corresponding queue at the Master.

### A. *Priority and K-Fairness Scheduling Policies for TDD MAC Layer*

The scheduling policies that we propose can be labelled as *Master-Slave Queue-State-Dependent Packet Scheduling Policies*. In these models, we distinguish the Master-Slave pairs based on the state of the queues at the Master and Slaves. Denote by 1 the state in which a node (Master or Slave) has data to send and by 0 when there is no data to send. Clearly, this leads us to four distinct states of the Master-Slave queue pairs. For example, the Master-Slave pair is in the 1-1 state if both the Master and the Slave of a connection have data to send; if both Master and Slave have no data, this pair is in the 0-0 state.

In these models, we assume that the binary information regarding status of the queue at the Slave is available at the Master. We require one bit to transfer this information (i.e., presence or absence of backlog at a Slave). There are free bits in the Bluetooth payload header that can be used to communicate the backlog information at a Slave to the Master.

In the *priority policy (PP)*, we give a higher priority to the Master-Slave connections in the 1-1 state over the Master-Slave pairs in 0-1 or 1-0 state. The 1-0 and 0-1 Master-Slave pairs are treated as being of equal priority.[2] Master-Slave pairs in 0-0 state are not scheduled. Note that the PP policy achieves a higher throughput than pure round robin policy since the connections in 1-1 state are given a higher number of slots. As a result, there is less wastage of slots since fewer slots are given to 1-0 and 0-1 pairs. At the same time, starvation of 0-1 or 1-0 connections is avoided. The important parameter in this policy is the priority $P$ given to the 1-1 connections over 1-0 or 0-1 (1-0 or 0-1 connections get a priority of 1). We study the effect of varying $P$ on the system performance.

In the *K-Fairness policy (KFP)*, we go beyond optimizing the system throughput and give a strict fairness bound. We perform round robin scheduling among all the Master-Slave connection pairs that are in 1-1, 0-1 or 1-0 states. When the scheduler is at a 0-1 or 1-0 connection, its service is sacrificed to a 1-1 connection provided that the difference between the service received by any two backlogged connections does not exceed K slots. To do this, we maintain counters for each pair of queues that keep track of the excess or deficit service received by a pair. We define *qmax* as the Master-Slave queue pair that has received maximum excess service (service sacrificed to it) from other pairs. Similarly, we define *qmin* as the Master-Slave queue pair that has sacrificed maximum service to other connections. Both *qmax* and *qmin* are defined for the

---

[1] We use the term Master-Slave pair and Master-Slave connection interchangeably throughout this paper

[2] In wireless systems, the 1-0 state may be given a higher priority than the 0-1 state since the Master may need to transmit important data packets to a non-backlogged Slave; in this paper, however, we give equal priority to the 1-0 and 0-1 M-S connections

backlogged queue pairs (1-0, 0-1 or 1-1). If the difference between the service received by *qmax* and *qmin* exceeds a number K, no further service is sacrificed by the pair being scheduled (i.e., a 1-0 or a 0-1 connection) and it gets the service. We calculate *qmax* and *qmin* by maintaining service counters for each pair at the Master. Thus, in the KFP, an absolute fairness (max-min fairness [4]) bound of K is guaranteed. The K-Fairness policy also allows for connections that have lost service to gain later and vice versa. This can happen by a 1-0 connection changing state to 1-1 connection and gaining excess service from the 1-0 or 0-1 connections thus making its service counter positive. The excess service to the 1-1 type connections is given in a round robin manner to ensure fairness. This policy achieves higher throughput as compared to the RR based policies and also maintains fairness bounds. As K (the fairness parameter) becomes 0 in KFP, the policy degenerates to pure RR.

It should be noted that the lag (service sacrificed) and the lead (excess service received) in KFP is different from the lag and lead concept that has been proposed in [5], [6]. In these papers, the authors try to overcome unfairness resulting due to bad wireless links. In KFP, the throughput is increased by sacrificing fairness.

## III. SCHEDULING DATA IN PRESENCE OF VOICE

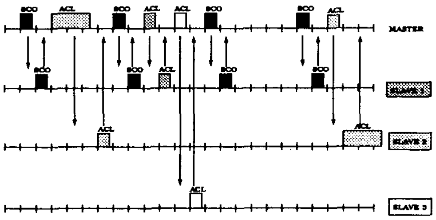

Fig. 2. Variable size data (ACL) in presence of reserved voice channels (SCO)

Bluetooth supports both voice (Synchronous Connection Oriented (SCO) links) and data traffic (Asynchronous Connectionless (ACL) links) [1], [2]. For voice, slots are reserved at regular intervals. The slot reservation for voice traffic imposes additional constraints to the scheduling policies for scheduling data in Bluetooth. Further, the TDD structure requires that data packets be of size 1,3 or 5.

For simplicity and clarity of presentation, we illustrate our policies for the case when 4 data slots are available between consecutive, non contiguous voice slots. This scenario is shown clearly in Figure 2. An important constraint in data scheduling is that a data packet transmission from a Master or Slave cannot span across voice slots.

We extend the PP (HOL-PP) and KFP (HOL-KFP) to use the information about the size of the Head-of-the-line (HOL) packets at the Master and Slave queues which are used to divide Master-Slave pairs into different classes. These classes are based on the slot utilization. In HOL-PP, we give higher priority to classes with high slot utilization. While in HOL-KFP, we transfer service from classes with low slot utilization to classes with high slot utilization. The transfer is bounded in a similar manner as in KFP. In the presence of 4 data slots between consecutive, non contiguous voice slots, the classes formed are {3-1, 1-3, 1-1} (100% slot utilization), {3-0, 0-3} (75% slot utilization) and {1-0, 0-1} (50% slot utilization).

As discussed in the previous paragraph, the data packet size distribution significantly effects the slot utilization. The distribution of data packet sizes is determined by Bluetooth SAR policy. The naive SAR policy (Random-SAR) is to assign the data packet sizes (i.e., 1, 3 or 5) probabilistically. We propose a new SAR policy, the *Intelligent SAR* (ISAR). In ISAR, we adapt MAC packet size at the Master and Slave queues depending on the data arrival rates at both the queues. Initially all queues have a packet of size one (i.e, one slot length). If both the Master and Slave queues have the same (i.e., high or low) arrival rates, the packet size remains equal to one. In case any one queue (Master or Slave) has a high arrival rate, while the other has a low arrival rate, packet sizes are changed. ISAR results in larger size packets at the queue with a high arrival rate and smaller size packets at the queue with a low arrival rate. This reduces slot wastage. In ISAR, the SAR unit at the Master needs to know about the arrival rate at the Slave queue and vice versa. This can be done through the exchange of one bit information between the Slave and the Master indicating high or low arrival rate.

## IV. SIMULATION RESULTS

Discrete Event Simulations have been performed for the various MAC scheduling policies proposed in this paper. We simulate a single piconet consisting of five active Slaves and a Master. For each Slave, there is a corresponding queue at the Master. The TDD slot length in Bluetooth is equal to 625 $\mu$secs [1]. In the simulation, we do not model the voice arrival process explicitly. Instead, in Bluetooth, since voice has a higher priority than data, voice is modelled as slot reservation at regular intervals.

The data arrival process at the Master and Slave queues is assumed to be either (i) Poisson (MP) or (ii) a two-state Markov Modulated Poisson Process (MMPP) [7]. For the MMPP process, the transition probability from one state to another is equal to 0.01 and the probability of remaining in a state is 0.99. The service time of a data packet depends upon the packet length. A packet of size one occupies a single slot.

Various simulation parameters are shown in Tables I & II. The arrival rate is in units of packets per TDD slot.

0-7803-5718-3/00/$10.00 ©2000 IEEE.

| Queue Type | M1 | S1 | M2 | S2 | M3 | S3 |
|---|---|---|---|---|---|---|
| Arrival Process | MP | MP | MP | MP | MP | MP |
| Arrival Rate | 0.2 | 0.2 | 0.39 | 0.01 | 0.01 | 0.39 |

TABLE I

VARIOUS SIMULATION PARAMETERS

| Queue Type | M4 | S4 | M5 | S5 |
|---|---|---|---|---|
| Arrival Process | MMPP | MMPP | MMPP | MMPP |
| Arrival Rate | 0.39,0.01 | 0.39,0.01 | 0.39,0.01 | 0.39,0.01 |

TABLE II

VARIOUS SIMULATION PARAMETERS

From Table I & Table II, we see that the Master-Slave (M-S) pair 1 has a high arrival rate (equal to 0.2) at both the Master and the Slave queues, and thus will be in the 1-1 state most of the time. M-S Pairs 2 and 3 always have very high arrival rate at one queue and low arrival rate at the other queue and thus will be in 1-0 or 0-1 state most of the time. Pairs with MMPP traffic (4 and 5) at times have a high arrival rate in both the queues or, at times they have a high arrival rate in one queue and a low arrival rate in the other queue, or, low arrival rates in both the queues.

The buffer size at the Slaves as well as Master queues is taken to be 20 packets. Discrete Event Simulation were run for 5000 TDD slots and the statistics were collected after the first 100 slots to remove any initialization bias in the simulations.

| Scheduling Policy | RR | PP | KFP |
|---|---|---|---|
| Throughput (% of slots used) | 76.3 | 86.9 | 89.1 |
| Avg. Delay (in units of slots) | 164.1 | 127.5 | 125.3 |

TABLE III

PERFORMANCE COMPARISON OF VARIOUS SCHEDULING POLICIES

| Scheduling Policy | RR | HOL-PP | HOL-KFP |
|---|---|---|---|
| System Throughput (% of slots used) | 54.08 | 54.938 | 58.5714 |
| Avg. Packet Delay (in units of slots) | 686.83 | 688.09 | 563.99 |
| Total Packet Drops | 3525 | 3387 | 3354 |

TABLE IV

PERFORMANCE COMPARISON OF VARIOUS SCHEDULING POLICIES WITH RANDOM-DISTRIBUTION

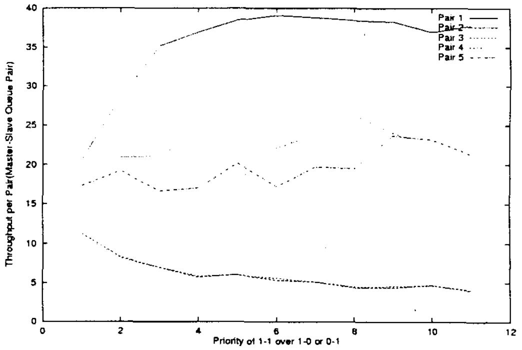

Fig. 3. Throughput of different hosts vs. parameter P in PP

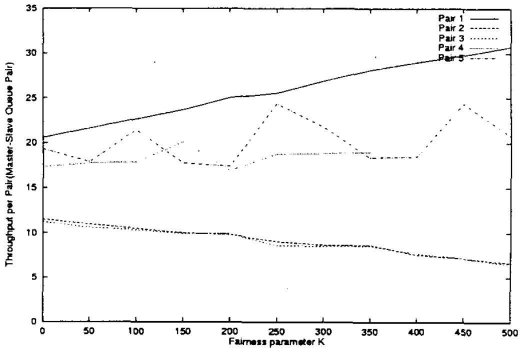

Fig. 4. Throughput of different hosts vs. parameter K in KFP

| P | 2 | 4 | 8 | 10 |
|---|---|---|---|---|
| HOL-PP | 54.3 | 56.7 | 59.2 | 59.1 |
| K | 50 | 100 | 200 | 250 |
| HOL-KFP | 56.2 | 57.5 | 59.6 | 61.9 |

TABLE V

SYSTEM THROUGHPUT (PERCENTAGE OF TOTAL SLOTS) IN HOL-PP AND HOL-KFP

In Table III, we compare the system througput achieved by using the three policies discussed: RR, PP and KFP in the absence of voice. The results for PP and KFP in Table III have been shown for P equal to 4 and K equal to 500. Note that for the above values of P and K, both PP and KFP give equal fairness. As the table shows, KFP performs the best among the three scheduling policies. PP performs better than RR. In Figures 3 and 4, we compare the fairness achieved in PP and KFP. Fairness is measured as the difference in throughput of different connections (Master-Slave pairs). In PP, for small values of P, the unfairness rises quadratically with an increase in pa-

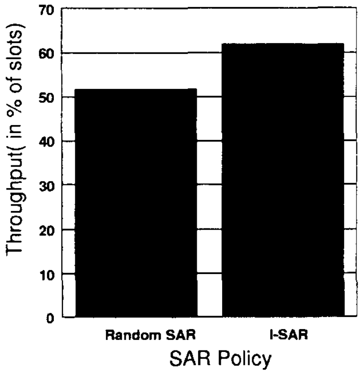

Fig. 5. Total System Throughput in ISAR and Random SAR

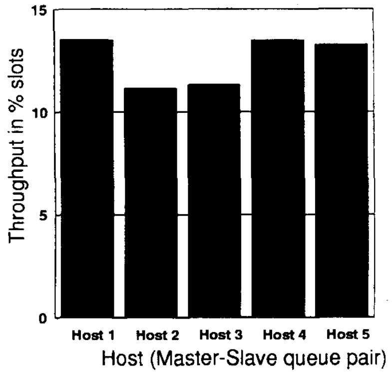

Fig. 6. Throughput of different hosts in ISAR

rameter P (for $P = 2$, unfairness increases by 4). As P is increased, this quadratic increase in unfairness decreases because the throughput achieved by the connection receiving highest throughput starts becoming input limited (and thus, the throughput curve flattens as shown in figure 3). In KFP, the unfairness is bounded and increases linearly with increasing K. Thus, KFP gives better throughput than PP with more fairness. Further, fairness bounds help one to make QoS guarantees in KFP. The results for HOL-PP and HOL-KFP are presented in IV. In V, we investigate the performance of HOL-PP and HOL-KFP as a function of P and K respectively.

In Figure 5, we see the performance improvement of ISAR as compared to the Random SAR. In Figure 6, we see the fairness achieved by ISAR. The throughput of all Master-Slave pairs is close to each other.

## V. CONCLUSION

In this paper we have proposed new and simple scheduling policies for the Bluetooth MAC. The algorithms utilize the state at the Master and the Slaves in a Piconet to schedule the TDD slots effectively. We extend the basic policies for scheduling variable size data packets in the presence of voice traffic. We have studied various SAR policies in Bluetooth with the aim of achieving high system throughput and fairness. An important problem that remains to be studied is how SAR should be modified/developed in the light of the applications for which Bluetooth will be used.

## REFERENCES

[1] Bluetooth Special Interest Group. http://www.bluetooth.com/

[2] http://www.bluetooth.net/

[3] C. Fragouli, V. Sivaraman, M. B. Srivastava, "Controlled Multimedia Wireless Link Sharing via Enhanced Class-Based Queueing with Channel-State-Dependent Packet Scheduling", IEEE INFOCOM'98, San Francisco, CA, USA, March, 1998.

[4] S. Keshav, *An Engineering Approach to Computer Networking: ATM Networks, the Internet, and the Telephone Network*, Addison-Wesley, 1999.

[5] S. Lu, V. Bharghavan and R. Srikant, "Fair Scheduling in Wireless Packet Networks", ACM SIGCOMM'97, August, 1997.

[6] S. Lu, T. Nandagopal and V. Bharghavan, "A Wireless Fair Service Algorithm for Packet Cellular Networks", ACM MOBICOM'98, Dallas, Texas, 1998.

[7] R. W. Wolff, *Stochastic Modelling and the Theory of Queues*, Prentice Hall Inc, Englewood Cliffs, NJ, 1989.

[8] 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 on Mobile Multimedia Communications (MoMuC'99), San Diego, California, November, 1999.