MAC Scheduling and SAR policies for Bluetooth: A Master Driven TDD Pico-Cellular Wireless System

Manish Kalia, Deepak Bansal, Rajeev Shorey
IBM Research Center,
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*—In this paper, we propose and study data scheduling and Segmentation and Reassembly (SAR) policies in Bluetooth. In such systems, the conventional scheduling policies such as Round Robin perform poorly as they are not suited to the tight coupling of uplink-downlink scheduling and result in slot wastage and unfairness. Scheduling in Bluetooth is complex due to (i) reserved slots for voice traffic, and, (ii) variable sized data packets. The reservation of voice slots at regular intervals results in non-contiguous TDD slots available for data. In this paper, we propose two new scheduling policies that utilize information about the size of the Head-of-the-Line (HOL) packet at the Master and Slave queues to schedule the TDD slots effectively. These policies achieve high throughput and greater fairness compared to the Round-Robin based scheduling policies. We then study the Segmentation and Reassembly policies at the Bluetooth MAC. SAR policies have a significant effect on data scheduling as they govern the distribution of packet size. We propose two new SAR policies for Bluetooth that give good performance in terms of throughput, delay and fairness. Finally, we include channel errors in Bluetooth and propose a modified scheduling algorithm that gives good performance.**

I. INTRODUCTION

The physical constraints of the wireless medium often lead naturally to a Master-Slave configuration [3], [1]. In a Master-Slave configuration, one of the stations in a cell is the Master (this could be the fixed Access Point or the Base Station) and the other remote stations are Slaves (e.g., the handheld devices such as palmtop computers, cell phones, pagers). The need for simplicity and low complexity has made Time Division Duplex (TDD) one of the promising candidates for medium access control (MAC) in wireless systems having a Master-Slave configuration. Proposed standards for low-power, low-cost wireless mobile communication systems, such as the Bluetooth [1], have adopted centralized TDD scheduling (i.e., at the Master) as the MAC protocol for scheduling access to the wireless medium. Efficient MAC scheduling and SAR policies need to be designed for such systems.

II. MEDIA 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. The Master can only send packets to a Slave in even slots while the slave can send packets to the Master in an odd slot [1]. *This implies that the scheduling occurs in pairs of slots (i.e., the Master-Slave pair).* Further, the task of scheduling is vested with the Master. Note that from the above description, it is not difficult to infer 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. Bluetooth supports both voice and data traffic [1]. Two types of links are supported between any two members of the piconet (i.e., a cell in Bluetooth) forming a Master-Slave pair. These are the (i) Synchronous Connection Oriented (SCO) links for voice, and the (ii) Asynchronous Connectionless (ACL) links for data. Voice traffic occupies fixed slots that are assigned a priori by the Master. When there is one SCO (i.e., voice) link, we can have four TDD slots available for the data traffic after every two voice slots [1]. Fig. 1 illustrates the Bluetooth slot structure and shows the TDD slots that are being shared by the SCO and ACL links. In Bluetooth only three data packet sizes are allowed : one slot, three slot and five slot length. An important constraint is that a data packet transmission from a Master or Slave cannot span across different voice slots.

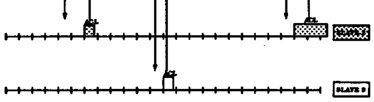

Fig. 1. Variable size data (ACL) in presence of reserved voice channels (SCO)

0-7803-5904-6/99/$10.00 © 1999 IEEE

III. HOL PRIORITY AND HOL K-FAIRNESS SCHEDULING POLICIES

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 size of the Head-of-the-Line (HOL) packets at the Master and Slaves queues. Master-Slave pairs (M-S pair) are referred to on the basis of the size of HOL packet at both queues (e.g., a 3-1 Master-Slave pair has a size three HOL packet at the Master queue and a size one HOL packet at the Slave queue). We study scheduling policies in the presence of one voice channel (i.e., one SCO link). We classify the Master-Slave pairs into three classes. When four consecutive slots for data are available, the Master-Slave pairs which are in the states 1-1, 3-1, 1-3 are best to schedule and form the first class (with no slot wastage). The 3-3 M-S pairs are also added to this class. The second class of M-S pairs is that of 3-0 and 0-3 (with 25% wastage). The third class is that of 1-0 and 0-1 (with 50% wastage). If only 2 consecutive slots for data are available the classification changes. In the first class only the 1-1 pairs remain. In the second class we have 3-1, 1-3, 1-0 and 0-1 (these have at least one 1 size packet). The third class has 3-0 and 0-3 (these have no 1 size packet, hence both slots are wasted).

A. HOL Priority Policy

In this policy we visit all the Master Slave pairs in a weighted Round Robin manner with different Master-Slave pairs having different priorities. All Master-Slave pairs in the same class are given the same priority. In the *HOL priority policy (HOL-PP)*, we give highest priority, P1, to the Master-Slave connections in the first class. The second class gets the second highest priority, P2. The third class is given the least priority, 1. The policy proceeds in the following manner:

1. Visit all backlogged (data packets at either Master or Slave queues) connections in a Round Robin manner.
2. If a connection has a priority P, serve the connection P times. If the connection changes class (as defined above) while in service, the service is stopped and we shift to the next connection.
3. The priorities are decided by the above mentioned class structure.

The important parameters in this policy are the priorities P1 and P2. We study the effect of varying P1 and P2 on the system performance. We take $P1 = 2P$ and $P2 = 3P/2$.

B. HOL K-Fairness Policy

In the *HOL K-Fairness policy (HOL-KFP)*, we go beyond optimizing the system throughput and ensure a strict fairness bound. The algorithm proceeds as follows:

1. Backlogged Master-Slave pairs are visited in Round Robin.
2. A Master-Slave pair belonging to class 1 is given its due service.
3. A Master-Slave pair belonging to class 2 tries to sacrifice its service to a pair belonging to class 1. This transfer of service is only allowed if the constraint $Qmax - Qmin < K$ is valid (Qmax, Qmin and K are as defined below).
4. A Master-Slave pair belonging to class 3 tries to sacrifice its service to a pair belonging to class 1. This transfer of service is only allowed if the constraint $Qmax - Qmin < K$ is valid. If no class 1 connection is available it tries to transfer the service to a class 2 connection.
5. If an M-S pair is not allowed to transfer its service, it is given its due service.
6. To ensure fairness the 1-1, 1-0 and 0-1 M-S pairs are served for 4 consecutive slots (if they have data to send). This is done because the M-S pairs having one or more three size packets also take four consecutive data slots of service.

To control the transfer of service, we maintain counters for each pair of queues that keep track of the excess or deficit service received by a pair. If a M-S pair A sacrifices its service to another pair B, the service counter of A is decremented while that of B is incremented. If a connection receives service when its turn is due in the round robin order, its service counter remains unchanged. 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 backlogged queue pairs. 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 and it gets the service. The HOL K-Fairness policy also allows for connections that have lost service to gain later and vice versa. *The parameter K represents the maximum allowed unfairness (in terms of slots) among any two backlogged Master-Slave pairs.*

IV. SAR IN BLUETOOTH MAC

The Segmentation and Reassembly incorporated in Bluetooth MAC is simple in design. The size of the MAC packets created by the SAR can vary among 1,3 and 5 slot lengths both at the Master and the Slave [1]. Data packets of different sizes can be sent on the same connection. In the present Bluetooth SAR, MAC packets are not reordered. This implies there are no explicit MAC level sequence numbers to detect reordering. We consider two types of packet size distributions for the SAR. In the *Random Distribution*, we assume that the MAC packet size changes randomly between 1, 3 and 5 slot lengths. For each packet the SAR segments, the packet size is chosen uniformly from 1, 3 and 5 with equal probability. We refer to the present Bluetooth SAR with random packet size distribution as Normal SAR. We then consider the *Batch SAR distribution.* In this policy, we assume that the packet size remains constant in a batch of packets. The batch size is a critical parameter in this policy. We keep batch sizes of the order of 20-40 packets (the buffer limit is 20 packets).

V. MODIFIED SAR POLICIES FOR BLUETOOTH MAC

A. Intelligent SAR - ISAR

In this section we present a SAR policy which determines the packet size based on the data traffic distribution pattern at the Master and Slave. We consider three broad categories of data traffic distributions which can typically occur. We can have high data rates from Master to Slave and Slave to Master (Video-conferencing etc.). In such a scenario we should have equal MAC packet sizes at both ends so that both directions are served at equal rate. We can have high data rate at one end (Master or Slave) while a low data rate at other end (e.g., FTP). In such a traffic pattern we have large MAC packets where there is high data rate and smaller MAC packets where there is a low data rate. This results in overall lower delays and packet drops and reduces throughput wastage. Finally, we can have varying data rates at both Master and Slave (Telnet, Chat etc). To change the MAC packet size it is essential for the SAR unit to know the arrival rate at the SAR at the other end. We use a reserved bit in the data packets. This bit is marked 1 if the arrival rate at the SAR is high and 0 if it is low. The SAR is aware of the rate at which data is being handed over to it for segmentation. To determine high or low data arrival rate, we compare the effective data rate in terms of bytes per second being delivered to SAR by the upper layers with the rate at which the queue is being served by the MAC scheduler. This is continuously updated using exponential averaging.

B. SAR with Partial Reordering (PRSAR)

In SAR with partial reordering, we try to achieve performance improvement with a random packet size distribution. In this scheme we look at two types of packets for transmission, the HOL packet and the SHOL (Second head of line or the second packet in the queue). Due to slot restrictions and the size of the HOL packet, we are at times unable to send the HOL packet and a slot is wasted. If the SHOL packet has an appropriate size, it can be sent. The protocol proceeds in two steps:

1. The Master queue chooses to schedule the HOL or the SHOL packet depending on the HOL packet size at the Slave queue.
2. The Slave queue chooses to schedule the HOL or the SHOL packet depending on the size of the packet transmitted by the Master.

The choice of the HOL packet or the SHOL packet is based on the slot restrictions. If four slots are available between reserved voice slots, two three sized packets should not be scheduled. Similarly if two consecutive data slots are available between reserved voice slots then only one size data packets should be scheduled. In this protocol there can be packet reordering. Since the present Bluetooth architecture does not support reordering, we need to improve this protocol. We restrict the packet reordering to at most 1 packet (partial reordering). This allows us to keep just an additional bit in the packet headers indicating the order of the packets (Note that this bit is different from the bit used for ARQ in Bluetooth).

VI. WIRELESS ADAPTED K-FAIRNESS POLICY

Bluetooth is a frequency hopping wireless system [1]. We have chosen two state Markov chain as the error model [5]. In this section we propose an extension to the K-fairness policy which performs better in presence of wireless errors. In the presence of wireless errors, fairness in HOL-KFP can suffer. We propose this policy to ensure better fairness in the presence of channel errors. If the wireless link between a Master and Slave is down it cannot be given service and hence its service is transferred to some other connection(chosen such that the constraint $Qmax - Qmin < K$ is satisfied). We treat this transfer of service on par with a sacrifice of service by a connection of class 2 or 3 to a class 1 connection. On transferring service due to errors in the wireless channel, we update the service counters (the service counters are described in the section HOL-KFP).

VII. ASSUMPTIONS IN THE SIMULATION

Discrete Event Simulations have been performed for the various MAC scheduling and SAR policies proposed in this paper. We simulate a piconet consisting of five 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 μsecs [1]. The data arrival process at the Master and Slave queues is assumed to be one of the following (i) Poisson (MP) or (ii) a two-state Markov Modulated Poisson Process (MMPP) [4]. 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 first and fourth M-S pair have MP arrival process at both Master and Slave queue with an arrival rate of 0.1. The second and third M-S pairs also have MP arrival processes. In second pair Master has a arrival rate 0.19 and Slave 0.01 while it is the opposite in third pair. The fifth pair has MMPP arrivals with rate varying between 0.19 and 0.01. The arrival rate is in units of packets per TDD slot. The buffer size at the Slaves and the Master queues is 20 packets. Discrete Event Simulation was run for 5000 TDD slots and the statistics were collected after the first 100 slots to remove any initialization bias in the simulations, if any. We assume One voice channel (SCO). This results in a reservation of two Voice slots after every four consecutive data slots. As a result the maximum throughput achievable is 66.67%.

VIII. SIMULATION RESULTS

In this section we present a detailed description of the performance of various scheduling and SAR policies that we have studied for the Bluetooth MAC.

A. Performance of HOL Priority and HOL K-Fairness Policy

In Table I, we compare the performance of the HOL Priority Policy (HOL-PP), the HOL K-Fairness Policy (HOL-KFP), and Round Robin (RR) in Random distribution of packet sizes. We see form Table I that RR and HOL-PP perform poorly in Random Distribution. Results for HOL-PP are presented for P equals to 4 while for HOL-KFP are presented for K equals to 240. The HOL-KFP performs the best among the three as it temporarily shifts service from one M-S pair to another, depending upon the size of HOL-packets.

In Table II, we compare the performance of the Priority Policy (HOL-PP), the K-Fairness Policy (HOL-KFP), and Round Robin (RR) in Batch-Distribution (with a batch size of 25 packets). We see from Table II that HOL-PP and HOL-KFP perform very well. Results for HOL-PP are presented for a P equals to 9 while for HOL-KFP are presented for K equals to 240. These values have been chosen as the unfairness is same in the two policies at these values (we measure unfairness by the maximum difference in throughput among any two Master-Slave pairs) and maximum throughput is achieved. The HOL-PP is able to perform better in Batch-Distribution as the priorities of M-S pairs change much less frequently (this is because packet size remains constant over a batch). This allows more service to be given to the connections which have a lesser slot wastage. The HOL-KFP again performs the best among the three. In Table III we study the total system throughput variation with P in HOL-PP and with K in HOL-KFP. In HOL-PP, the system throughput increases exponentially for small values of P and saturates at P equal to 10. In contrast, for HOL-KFP the increase in system throughput is almost linear with respect to parameter K. Thus, the system performance can be accurately adjusted through parameter K. In Fig. 2 and Fig. 3 we plot throughput versus P and throughput versus K for HOL-PP and HOL-KFP respectively for different M-S pairs. We observe that in spite of giving high throughput gains, HOL-PP results in drastic and uncontrolled degradation in performance for some of the connections. HOL-KFP is better as it gives a fairness bound (and hence guarantees QoS to the connections) while achieving a high throughput.

TABLE I

PERFORMANCE COMPARISON OF VARIOUS SCHEDULING POLICIES WITH RANDOM-DISTRIBUTION

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 II

PERFORMANCE COMPARISON OF VARIOUS SCHEDULING POLICIES WITH BATCH-DISTRIBUTION

Scheduling Policy	RR	HOL-PP	HOL-KFP
System Throughput (% of slots used)	54.26	59.59	61.71
Avg. Packet Delay (in units of slots)	646.60	555.70	518.60
Total Packet Drops	3289	2896	2749

TABLE III

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

B. Performance of the SAR Policies

In this section we compare the performance of the two SAR policies - Partially reordered SAR (PRSAR) and Intelligent SAR (ISAR). We compare their performance with Normal SAR.

In Table IV we observe the total system throughput for

TABLE IV

PERFORMANCE COMPARISON OF VARIOUS SAR POLICIES

SAR Policy	Normal SAR	PRSAR	ISAR
Throughput (% of slots used)	51.76	56.34	61.98
Packet Delay (in units of slots)	487.71	467.57	310.408

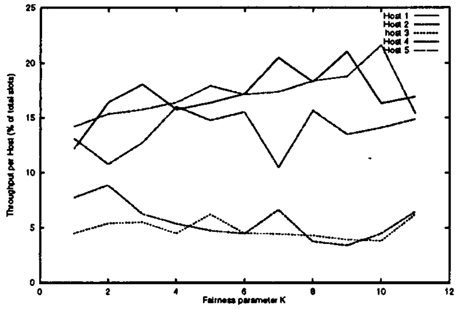

Fig. 2. Throughput of different hosts vs. parameter P in HOL-PP (in Batch Distribution)

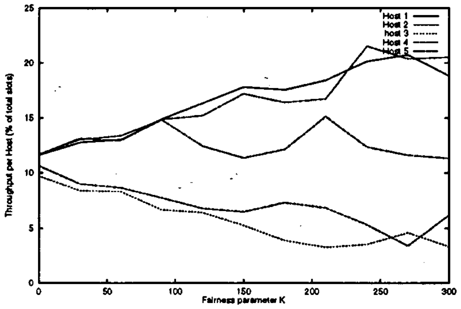

Fig. 3. Throughput of different hosts vs. parameter K in HOL-KFP (in Batch Distribution)

the SAR policies. ISAR policy performs very well. *This is expected as it adapts the packet sizes to the traffic patterns.* The PR-SAR policy performs better than the Normal SAR. This illustrates that by allowing a reordering of even 1 packet we can achieve significantly good results. In Table IV, the average delays are shown for the three SAR policies. The ISAR policy performs the best since it avoids the blocking of the HOL packet by generating an appropriately sized HOL packet. The PR-SAR performs better than the normal SAR. This is primarily because it has higher throughput than the normal SAR.

In Table V we highlight the fairness achieved by the ISAR and PRSAR policies. It can be seen that the ISAR policy is most fair among all policies discussed so far.

C. Wireless Adapted KFP (WAKFP)

In this section we study the performance of wireless adapted HOL-KFP. We compare it with the performance of HOL-KFP in presence of wireless errors. Wireless errors are introduced using two-state Markov chain model [2], [5]. For low channel errors, good to bad state transition probability (GBP) was taken 0.05 while bad to good transition probability (BGP) 0.95. For high channel errors GBP was taken as 0.35 and BGP as 0.75. In the simulation scenario we varied the wireless channel conditions for different Master-Slave pairs. WAKFP performs better than HOL-KFP in terms of total system throughput(see Fig. 4). WAKFP also leads to more fairness than HOL-KFP.

TABLE V

THROUGHPUT (PERCENTAGE OF TOTAL SLOTS) OF DIFFERENT MASTER SLAVE PAIRS IN PRSAR AND ISAR.

M-S pair	1	2	3	4	5
PRSAR	12.61	10.1	8.1	11.81	13.46
ISAR	13.53	11.18	11.36	13.51	13.31

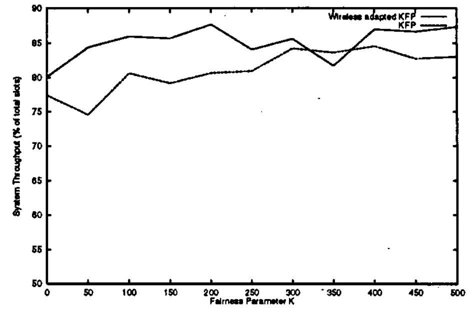

Fig. 4. System throughput for different K values for WAKFP and HOL-KFP

IX. CONCLUSIONS

We have studied efficient MAC scheduling and SAR policies in Bluetooth with an aim of achieving a high channel utilization (throughput) and ensuring almost equal bandwidth to different connections. The scheduling policies that have been proposed in this paper can be appropriately labelled as *Master-Slave Queue-State-Dependent Packet Scheduling Policies.* We have seen that Segmentation and Reassembly in Bluetooth MAC affects system performance significantly. The two new SAR policies, ISAR and PRSAR lead to good performance gains with little overheads.

REFERENCES

[1] Bluetooth special interest group. http://www.bluetooth.com/
[2] 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.
[3] S. Keshav, *An engineering approach to computer networking: ATM networks, the Internet, and the Telephone network*, Addison-Wesley, 1999.
[4] R. W. Wolff, *Stochastic modelling and the theory of queues*, Prentice Hall Inc, Englewood Cliffs, NJ, 1989.
[5] A. Kumar, *Comparative performance analysis of versions of TCP in a local network with a lossy link*, IEEE Transactions on Networking, August'98.