International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:3, No:2, 2009

Enhancing Performance of Bluetooth Piconets Using Priority Scheduling and Exponential Back-Off Mechanism

Dharmendra Chourishi "Maitraya", and Sridevi Seshadri

***Abstract*—**Bluetooth is a personal wireless communication technology and is being applied in many scenarios. It is an emerging standard for short range, low cost, low power wireless access technology. Current existing MAC (Medium Access Control) scheduling schemes only provide best-effort service for all master-slave connections. It is very challenging to provide QoS (Quality of Service) support for different connections due to the feature of Master Driven TDD (Time Division Duplex). However, there is no solution available to support both delay and bandwidth guarantees required by real time applications. This paper addresses the issue of how to enhance QoS support in a Bluetooth piconet. The Bluetooth specification proposes a Round Robin scheduler as possible solution for scheduling the transmissions in a Bluetooth Piconet. We propose an algorithm which will reduce the bandwidth waste and enhance the efficiency of network. We define token counters to estimate traffic of real-time slaves. To increase bandwidth utilization, a back-off mechanism is then presented for best-effort slaves to decrease the frequency of polling idle slaves. Simulation results demonstrate that our scheme achieves better performance over the Round Robin scheduling.

***Keywords*—**Piconet, Medium Access Control, Polling algorithm, Scheduling, QoS, Time Division Duplex (TDD).

I. Introduction

BLUETOOTH is a wireless technology that allows communication devices and accessories to interconnect using a short-range, low-power, inexpensive radio. Bluetooth was developed initially as a replacement for short-range cable linking portable consumer electronic products, but it can also be adapted for printers, keyboards etc. To date Bluetooth has expanded on wireless LANs. [1].

The smallest Bluetooth unit is called a piconet, which consists of one master node and many slave nodes (up to seven active slaves). All the nodes in a same piconet should follow same frequency hopping pattern. Multiple piconets can also exist in the same area and can be connected via a bridge node, forming a scatternet.

In a Bluetooth system, full-duplex transmission is supported using a master driven TDD (Time Division Duplex) scheme to divide the channel into 625 μs time slots. The time slots are alternatively switched between the master and the slaves. The master sends a poll or a data packet to a slave using the even numbered time slots. The slave sends a packet to the master in the immediate odd numbered slot. Thus, the MAC scheduling in Bluetooth is controlled by the master.

Authors are with Shinas College of Technology, I. T. Department, Sultanate of Oman (e-mails: dharm13@yahoo.com, sri_gow@yahoo.com).

As shown in Fig. 1 Bluetooth system supports two types of data communication channels between the master and the slave: a Synchronous Connection Oriented (SCO) link and Asynchronous Connection Less (ACL) link. An SCO connection supports a circuit-oriented service with a constant bandwidth using a fixed and periodic allocation of slots. An SCO connection is suitable for delay-sensitive multimedia traffic like voice traffic, whereas an ACL connection supports a packet-oriented service between the master and slave. The ACL connection is suitable for various applications such as ftp, telnet, audio and video applications. Because these applications have various QOS requirements (such as delay and bandwidth), it is very important to provide different QOS for them. However, current Bluetooth specification doesn't address how to meet these different QOS requirements, and current implementations only provide best-effort service to all applications.

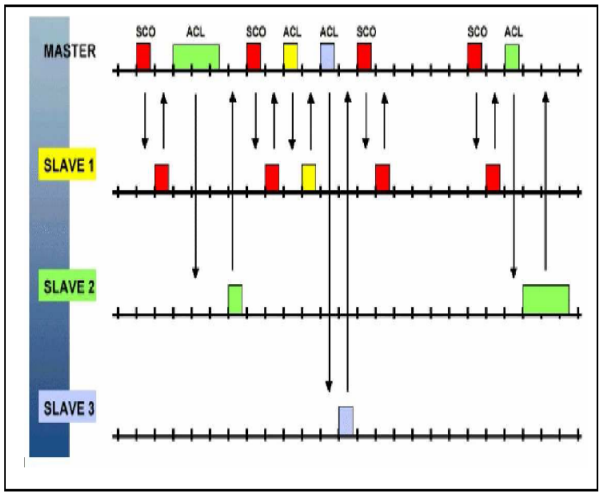

Fig. 1 Master Driven TDD Scheme

The Round Robin (RR) scheme is a default MAC scheduling algorithm for Bluetooth that uses a fixed cyclic order. The POLL packet does not have any information and just gives the polled slave the privilege of transmitting packet in the next slot. If the polled slave does not have any data to transmit, it replies to the master by sending a NULL packet which also does not have any information. As a result,

numerous slots will be wasted with POLL or NULL packet exchanges in the case of no data to transmit.

To address these problems, in this paper, we propose a priority based MAC scheduling algorithm to improve QoS in the Bluetooth Piconet. We divide our slaves into real time and best effort slaves and assign different priorities to them. To improve the QoS, we define token counters to estimate the traffic of the real time slaves and we apply a exponential backoff mechanism for the best effort slaves to decrease the frequency of polling idle slaves.

The paper is organized as follows. In section 2 we present some of the previous work on Bluetooth MAC scheduling algorithms. In section 3 we explain the propose scheduling algorithm and in section 4 we compared the performance of our algorithm with the traditional RR algorithm using simulation. Concluding remarks are summarized in section 5.

II. Bluetooth MAC Scheduling Algorithms

In Round Robin (RR) scheme, every slave has the same opportunity to send one data packet even when they have no packet to transmit. Once the master polls a slave, the next time slot is then assigned to the slave without considering whether the slave has data to transmit or not. Several Bluetooth MAC scheduling algorithms have already been proposed to improve the system performance.

In [2], a Master- Slave Queue-State-Dependent Packet scheduling algorithm is proposed. In that algorithm a free bit in the Bluetooth payload header is used by a slave to inform the master of the next available data. Based on this feedback bit, the master classifies all the master-slave pairs into one of four states, and a higher priority is assigned to a pair that utilizes the slots more efficiently than the other pairs. In [3], HOL-Priority Policy (HOL-PP) is proposed. This algorithm uses similar priority policy with [2]. The master schedules on the basis of Head Of Line(HOL) packet size at the master and slave queue. In [4], the authors proposed several schemes that scheduled slaves based on their queue lengths. Though these policies solved the problem of bandwidth wastage to some extent, they needed to know extra information about the queues at slaves which is not available in the current Bluetooth specification. Moreover, they did not address the QoS issue.

III. Priority Scheduling and Exponential Back-Off Mechanism

In our proposed priority based scheduling mechanism, the master maintains queues with different priority levels. We classify our slaves into real time slaves and best effort slaves. When the master receives packets from the real time slaves it should deliver them as soon as possible to meet their maximum delay requirements. But the packets from the best effort slaves can be delayed, since they don't have any QoS requirement. For each real time slave we assigned a unique priority based on its maximum tolerable delay. If two real time slaves have the same delay requirement, then the master will assign higher priority to the slave which generates the request earlier. All the best effort slaves are given the same priority, the lowest one. In our scheduling algorithm, the real time slaves are scheduled according to their priorities and the best effort slaves are scheduled in round-robin manner.

In Master Driven TDD approach, a slave can transmit a packet only after receiving a polling packet from the master. Because of this, the master doesn't know whether a slave has data to transmit or not unless it sends a polling packet to the slave. To overcome this disadvantage, we estimated the traffic of real time slaves. When the slaves have no data to transmit, polling them will decrease the utilization of bandwidth. To reduce unnecessary polling of idle slaves, we used the exponential backoff mechanism to reduce the polling frequency.

TABLE I
Notations used in this Paper

S_i	Slave i
R_i	Average bit rate of S_i
MD_i	Maximum accepted delay of S_i
L_i	Packet Length of S_i
C_i	Token Counter of S_i
T_i	Token counter generation Interval of S_i
P_i	Priority of S_i
PI_i	Polling Interval of S_i
W_i	Polling Window of S_i
W_{MAX}	Maximum window size

In our proposed algorithm, we assign a unique priority P_i to slave S_i according to MD_i . The slaves with minimum value of MD_i is given the maximum priority. This ensures reasonable delay performance for the slaves. The real time data from the slaves are scheduled according to the priority. Token counter C_i is used to avoid frequent polling of the slaves. The value of C_i is increased by 1 per T_i seconds, where

$$T_i = L_i / R_i$$

We used the exponential backoff algorithm to determine the amount of time an inactive slave is removed from the polling cycle. For each best effort slave S_k , the value of W_k is set to 1 by default and updated with a binary exponential backoff mechanism. The value of I_k is set according to W_k.

In this section we provide a pseudo-code description of our algorithm.

At the beginning of each cycle, the master do the following tasks.

Scheduling of real time slaves

1. Schedule the real-time slave S_i with the highest priority P_i .
2. If there is data packet to S_i
   then send data packet to S_i
   else
   send POLL to S_i
3. If (S_i →M subscript= = NULL)
   then C_i =0;

else
Send POLL to S_i and repeat step 3.

4. Repeat steps through 2 to 3 for the slaves with lower priorities.

In the above algorithm, a slave with a lower priority is scheduled only when all the slaves with the higher priorities have no packets to transmit. A slave is considered to have no packets to transmit only if its token counter is zero and the master has no packet to send. When all the real time slaves have no packets to transmit, the best-effort slaves are scheduled in the round robin manner as follows.

Scheduling of best effort slaves

1. If ($S_k \rightarrow$M payload = = NULL)
   then $W_k = \min(W_{MAX}, (2* W_k))$;
   else $W_k = 1$;
   $PI_k = W_k$;

At the beginning of each cycle when the best effort slaves are scheduled, the polling interval of each slave is decreased by 1. Only those slaves with polling interval with 0 can send the packets.

IV. Simulation Results

In this section we compare the performance of our MAC scheduling algorithm which is based on the priority of the packets and the token counter with the traditional Round Robin algorithm.

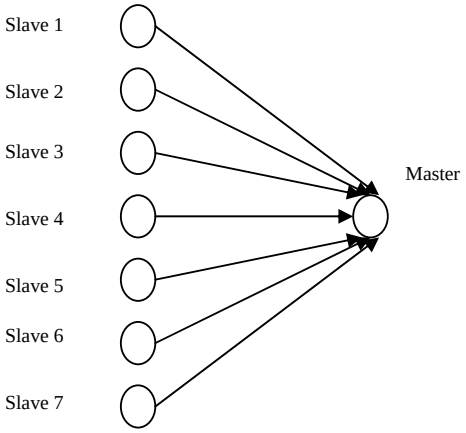

Fig. 2 Simulation Scenario

As shown in Fig. 2, the network model simulated is a single piconet with one master and seven slaves, where the master had a corresponding queue for each slave. The traffic mode considered was ACL. All traffic were generated by the NS2 [5]. Out of the seven slaves, five slaves generates real time data and the remaining two slaves generates best effort data. Slaves 1,2 and 3 generates Constant Bit Rate (CBR) data and slaves 4 and 5 generates Variable Bit Rate (VBR) data. The remaining two slaves 6 and 7 generates best effort data. The data generated by the slaves is transmitted as DH5 packets. DH5 packets have user payload as 0-339 bytes. We assigned priorities to the real time slaves 1 through 5 based on their maximum acceptable delay as shown in the Table II. The best effort slaves 6 and 7 are given the priority 1.

TABLE II
Simulation Parameters

Slave	Flow Type	Start time	Proposed Scheme
1	CBR	0s	6
2	CBR	10s	5
3	CBR	20s	4
4	VBR	30s	3
5	VBR	40s	2
6	FTP	50s	1
7	FTP	60s	1

TABLE III
Average Delays (ms) of Real- Time Slaves

Slave	1	2	3	4	5
RR	92.1	109.0	45.5	48.6	54.66
Proposed Scheme	8.4	9.1	12.3	19.4	21.44

TABLE IV
Maximum Delays (ms) of Real-Time Slaves

Slave	1	2	3	4	5
RR	154.47	154.40	154.74	157.57	163.6
Proposed Scheme	34.65	45.45	54.12	57.54	59.4

Table III represents the average delay experienced by the real time slaves and Table IV represents the maximum delay. These tables show that our proposed scheme gives better performance for real time slaves when compared to RR scheme. Since the RR treats all the slaves equally, real time slaves which need bandwidth according to the maximum tolerable delay are affected much. Figs. 3 through 7 gives the throughput comparison of our scheme with RR for the real time slaves. Our proposed algorithm gives better performance to real time slaves because we used token counters and allotted priority to them according to their maximum tolerable delay. In our scheme we used exponential backoff mechanism for best effort slaves to avoid the polling of idle slaves. And that bandwidth is used for real time slaves to increase their throughput.

Figs. 8 and 9 show the throughput of the best effort slaves 6 and 7. The bandwidth available after used by real time slaves is shared equally by all the best effort slaves.

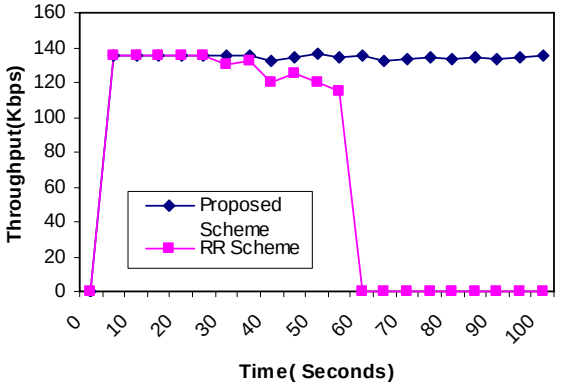

Fig. 3 Throughput of Slave 1

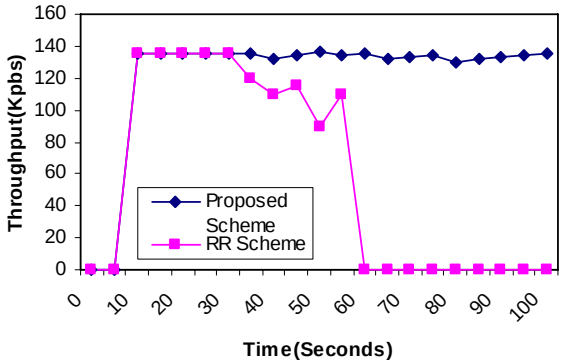

Fig. 4 Throughput of Slave 2

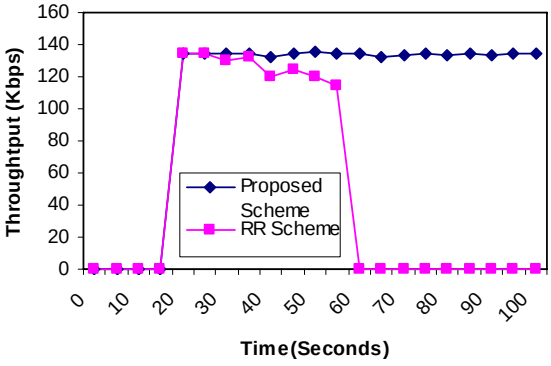

Fig. 5 Throughput of Slave 3

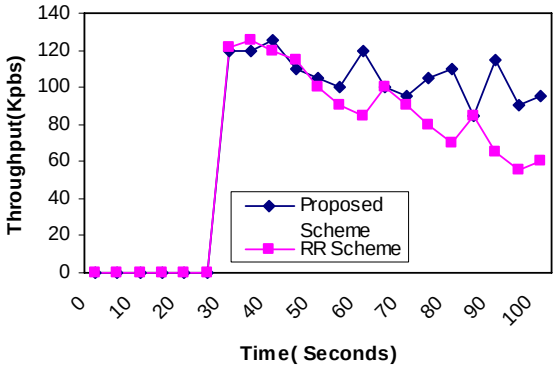

Fig. 6 Throughput of Slave 4

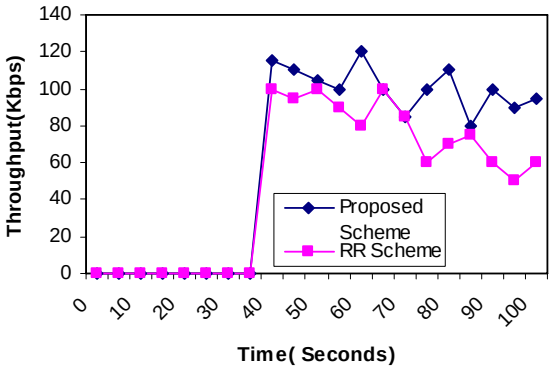

Fig. 7 Throughput of Slave 5

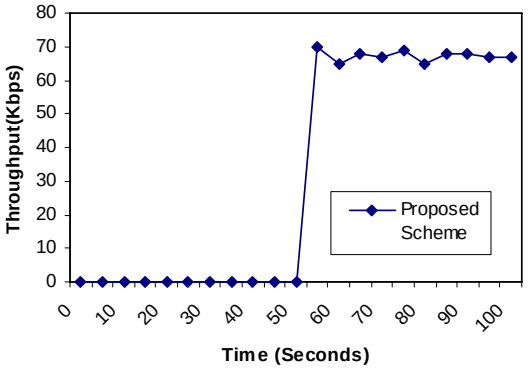

Fig. 8 Throughput of Slave 6

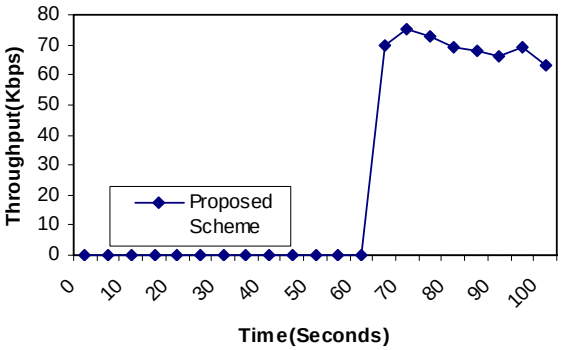

Fig. 9 Throughput of Slave 7

V. Conclusion

In this paper, we proposed a priority-based MAC scheduling algorithm for data exchange within a Bluetooth piconet. We evaluated its throughput and delay performances. We showed how to schedule real-time slaves efficiently and provide better QoS performance by using token counters to estimate their traffic. And also we demonstrated how to decrease the channel bandwidth wastage caused by polling idle slaves by applying a exponential backoff mechanism for polling intervals of best-effort slaves. Simulation results demonstrated that in a Master Driven TDD Bluetooth piconet, our proposed approach achieved significantly better performance over the RR scheme.

References

[1] Bluetooth Special Interest Group. http://www.bluetooth.com/
[2] M. Kalia, D. Bansal, R. Shorey, "Data Scheduling and SAR for Bluetooth MAC",in Proc.of IEEE Vehicular Technology Conference,2000,pp.716-720
[3] M. Kalia, D. Bansal, R. Shorey, "MAC Scheduling and SAR Policies for Bluetooth: A Master Driven TDD Pico-Cellular Wireless System", in Proc.of IEEE International Workshop on Mobile Multimedia Communications,1999,pp.384-388
[4] A. Das, A. Ghose, et al., "Enhancing Performance of Asynchronous Data Traffic over the Bluetooth Wireless Ad-hoc Network", IEEE INFOCOM'01,Vol.1,pp 591-600, April 2001.
[5] S. McCanne and S. Floyd, "NS-Network Simulator",