International Journal of Wireless Networks and Communications.
ISSN 0975-6507 Volume 4, Number 1 (2012), pp. 11-18

http://www.irphouse.com

# Efficiency Parameter (Latency) for Wireless Network to Increase QOS

**Ramesh K. and Ishwar Baidari**

*Department of PG Studies & Research in Computer Science*
*Karnataka University Dharwad - 580003, India*
*E-mail: rvkkud@gmail.com, ishwar_sp@yahoo.co.in*

**Abstract**

Bluetooth is a personal wireless communication technology [2] and is being applied in many scenarios. Current existing scheduling scheme only provide best-effort service for all master-slave connection. It is very challenging to provide Efficiency parameter (Latency) support for different connection due to the feature of Master Driven TDD (Time Division Duplex). This paper addresses the issue to how to enhance Efficiency parameter (Latency) support in a Bluetooth piconet. I propose an scheduling algorithm which can provide Latency as Efficiency parameter for different connection based on priorities[1]. Considering the feature of Master Driven TDD, we define token counters to estimate traffic of real-time slaves. To increase bandwidth utilization, a backoff 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 existing schemes.

**Keywords:** Bluetooth, piconet, Master, Slave, Efficiency Parameter, Scheduling Algorithm, Latency

## Introduction

Bluetooth is a system for providing short-range, small size, low-power and low-cost connectivity operating in the ISM (Industrial Scientific Medicine) band at 2.4GHz [1]. 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, toys and virtually any other digital consumer devices. To date Bluetooth has been seen as a promising candidate for ad-hoc wireless networking and wireless personal area network (WPAN).It has many new potential applications such as Internet Bridge, ultimate headset and so on.

Bluetooth supports both voice and data traffic which is treated differently. Voice is provided a guaranteed service over SCO connections, which occupy fixed slots. Data are provided a best-effort service over ACL connections. Bluetooth can support various applications, such as ftp, telnet, audio and video applications. Because these applications have various Efficiency Parameters requirements (Such as Bandwidth), it is very important to provide different Efficiency Parameters for them, especially for multimedia applications. However, current Bluetooth specification doesn't address how to meet these different Efficiency Parameters requirements and current implementations only provide best-effort service to all applications. To address this problem, It is proposed a new Priority-Based Scheduling Algorithm to enhance latency as Efficiency Parameter for Wireless Network using Bluetooth, To categorize all slaves into two classes and assign different priorities to them. To schedule slaves efficiently and define token counters to estimate traffic of real-time slaves and apply a binary exponential backoff mechanism for best-effort slaves to decrease the frequency of polling idle slaves. Compared with RR scheme, our scheme achieves better performance. This paper is organized as follows. Section 2 introduces some related work. In section 3 we descript our priority-based Scheduling Algorithm. Section 4 shows simulation results and section 5 concludes this paper.

## Related Works

Bluetooth is a Master Driven TDD system [1] as shown in Figure1.In a Bluetooth piconet, the channel access is controlled by the master and a slave can send a packet only after it has received a packet from the master. Current Bluetooth implementations adopt a round-robin scheme for scheduling for ACL connections, which leads to low bandwidth utilization when one or more slaves have no data to transmit.

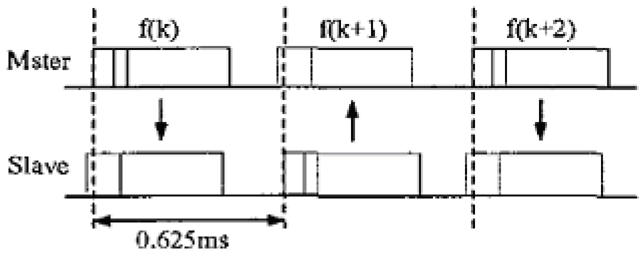



**Figure 1:** Master Driven TDD system

To alleviate this problem, some Scheduling Algorithms have been proposed. Alia ET al. proposed two polices that utilized information about the size of the Head-of-the-Line (HOL) packet at the master and slave queues to schedule the TDD slots. As ET al. also proposed several schemes that scheduled slaves based on their queue lengths. Through 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 Efficiency Parameters issue. Another Algorithm, named Efficient Double-Cycle (EDC), used a truncated binary exponential backoff mechanism to dynamically adapt the polling frequency to the traffic conditions thereby limited the channel bandwidth wastage caused by the polling of idle slaves.

This Algorithm didn't need extra information but it still didn't solve the issue of Efficiency Parameters completely. Subsequently, Chawla ET al. proposed a Efficiency Parameters based scheme, called Voice over ACL, to schedule voice with Earliest Due Deadline (EDD) policy over ACL link while ensuring that the Efficiency Parameters for voice was still met. Although this scheme did address the issue of Efficiency Parameters it only distinguished voice from data and couldn't be extended to the case of multi-class traffic. Moreover, with the EDD policy, the master still needed to know the arrival time of packets at slaves.

## A Priority-Based Scheduling Scheme

Current Bluetooth can provide voice support over SCO connections. In this section, the purpose of a priority based Scheduling Algorithm is to transmit multimedia data (including voice) over ACL connections with Efficiency Parameters provision.

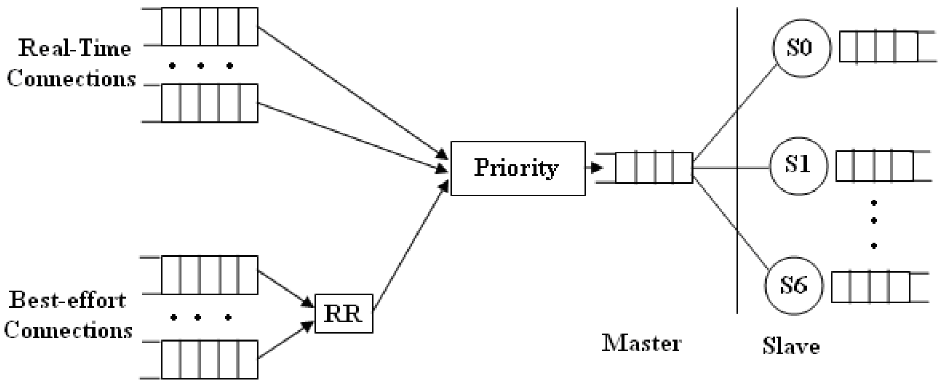



**Figure 3:** Priority based Scheduling Scheme

Figure.3. shows the structure of the scheme in a Bluetooth. The Master maintains a queue for each slave, which maintains it's own Queue. To provide different Efficiency Parameters for different Slaves, to categorize the master-Slave connections in to two classes: One is real time connection whose packet should delivered as soon possible, To meet their Latency requirements. The other one is best-effort connection that has no Efficiency Parameters requirement. The connections that have no data to transmit in both directions belong to the second class. Each real-time connection is assigned a unique priority based on its Latency requirement.

If two connections have the same Latency requirement, earlier one will get a higher priority. All the best-effort connection is given the same priority, the lowest one. To schedule the real-time slaves strictly based on their priorities and best-effort slaves in round-robin manner.

Due to the feature of Master Driven TDD, slave can transmit a packet only after it has received a polling packet from the master.

1. The master cannot know whether a slave has data to transmit unless it sends polling packet to the slave, which leads to that traditional time-stamp based algorithms, such as EDD and WFQ series, cannot be used.
2. When one or more slaves have data to transmit, to poling them will decrease Bandwidth utilization.

To cope with the first issue, try to estimate the traffic of real-time slaves. First, for each real-time slave Si, to define the following parameters: average bit rate Ri, maximum tolerable Latency Di, packet length Li, token counter Ci (similar to polling counter), and token counter generation interval Ti. Then the traffic of Si can be estimated by increasing Ci by 1 per Ti seconds, where Ti=Li /Ri seconds, to assign a unique priority to Si according to Di and schedule real-time slaves strictly based on their priorities.

Combining strict priority policy and token counters, to provide reasonable Latency performance for real-time slaves as well as avoid frequent polling of them. To deal with the second issue, employ a binary exponential backoff mechanism for best-effort slaves to control their polling frequencies. A polling window Wk and a polling interval Ik are defined for each best-effort slave Sk.then, the value of Wk is set to 1 by default and updated with a binary exponential backoff mechanism while Ik is set according to Wk.

The same algorithm has been implemented to show efficient parameter as delay by Yunxin Liu and Qian Zhang.

Scheduling Algorithm Precedes As Follows

1. Schedule the real-time slave with the highest priority, Si. If the master has any packet for Si, send them to Si .If Si a returned NULL packet After the master sent out all the packets, Ci to 0;Otherwise,keep Polling Si until it returns a NULL packet and then set Ci to 0.
2. Schedule slaves with lower priorities with the same policy used in 1.
3. A slave with a lower priority is scheduled only when all slaves with higher priorities have no packet to transmit.
4. A slave is considered to have no packet transmit only if its token Counter is 0 and the master has no packet to send.
5. When all real-time slaves have no packet to transmit, the best-effort Slaves are scheduled in round robin manner.
6. When Scheduling best-effort slaves, if the master send a NULL packet is returned.Wk is doubled unless a maximum value Wmax is reached; Otherwise Wk is set to 1.Then Ik takes the value of Wk.
7. At the beginning of each cycle when the best-effort slaves are scheduled, the polling interval of each slave is decreased by 1 unless a Value 0 is reached. Only those slaves whose polling interval is 0 can transmit packets.

## Simulation

The Simulation in my project is to demonstrate the performance of Priority based Scheduling algorithm applied for Real-Time slaves is better than the existing Round Robin algorithm applied for Best-Effort slaves in Bluetooth network. As depicted in Figure 3, the Simulation configuration in a Bluetooth piconet composed of a Master and seven Slaves. The compared results of Average Latency are tabulated in table 1-table 2

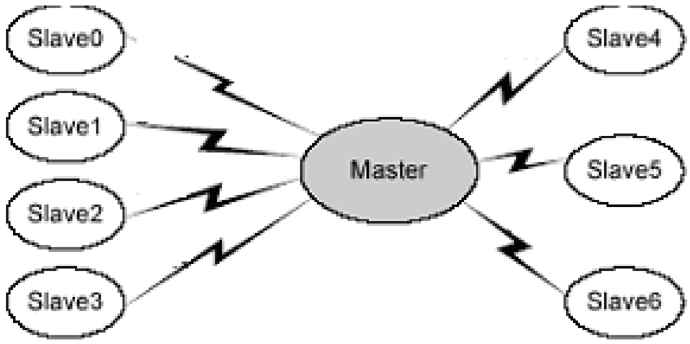



**Figure 3:** Simulation configuration

## Type-style and fonts

**Table 1:** Average Latency (ms) of Real-Time Connection

| Slave Num | 0 (50bytes) | 1 (75bytes) | 2 (100bytes) |
|---|---|---|---|
| Our Algorithm | 522.2 | 634.4 | 928.0 |

**Table 2:** Average Latency (ms) of Best-Effort Connection

| Slave Num | 0 (50bytes) | 1 (75bytes) | 2 (100bytes) |
|---|---|---|---|
| RR | 948.5 | 1296.1 | 1567.5 |

The result of table1-table4 shows Our Algorithm performed better than the existing Round Robin. Because RR deals with all connections equally and provides the same services.

## Results

The Figure 4 shows result of Average Latency Vs Packet Sent which clearly shows real time shows better performance than best effort

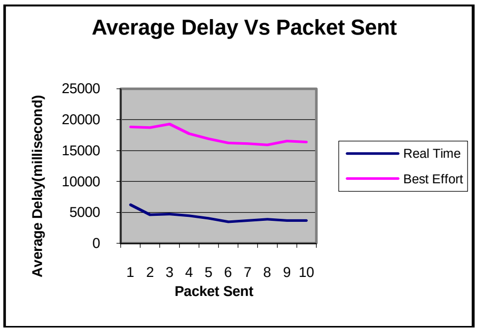



**Figure 4:** Average Latency Vs Packet Sent

The Figure 5 shows result of Average Latency Vs Packet Receive which clearly shows real time shows better performance than best effort

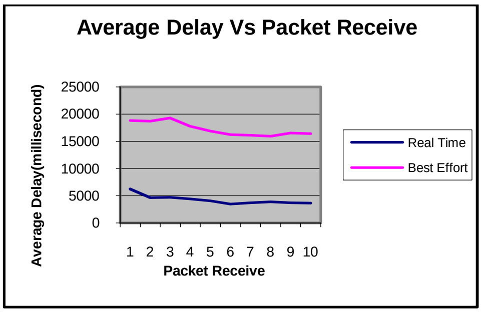



**Figure 5:** Average Latency Vs Packet Receive

The Figure 6 shows result of Average Latency (millisecond) Vs Packet Sent for multiple clients. For different packet Sent Average Latency of Each Real Time is less than the Each Best Effort, which clearly indicates real time shows better performance than best effort.

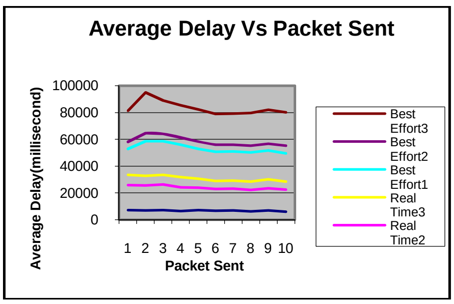
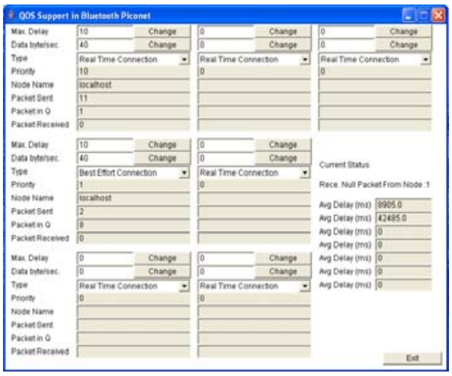


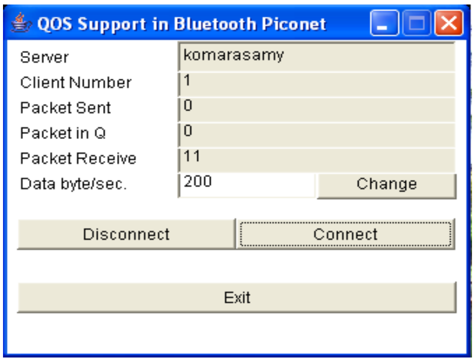

**Figure 6:** Average Latency Vs Packet Sent for Multiple clients

## Implementation Output

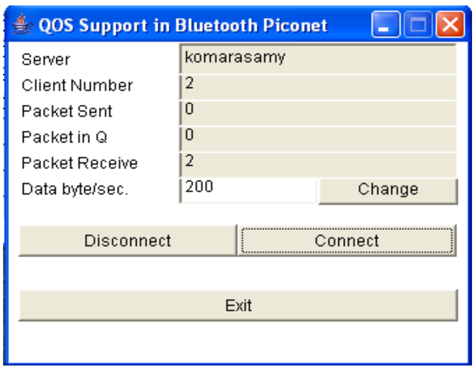
### Server side - QoS Support in Bluetooth Piconet



### Client 1-Qos Support In Bluetooth Piconet




### Client 2-Qos Support In Bluetooth Piconet



## References

[1] Priority Based MAC Scheduling Algorithm for Enhancing QOS for Blue Tooth support by Yunxin Liu
[2] Bluetooth special interest group, "SPECIFICATION OF THE BLUETOOTH SYSTEM", volume 1,URL– http://www.bluetooth.com,Feb.2001
[3] M.Kalia, D. Bansal, and R.Shorey," Data Scheduling and SAR for Bluetooth MAC", IEEE VTC-Spring 2000, volume 2, May 2000.
[4] R Bruno, M Conti and E Gregori, "Wireless Access to Internet Via Bluetooth Performance Evaluation of the EDC Scheduling Algorithm", Proc. ACM. WMI'01,pp 43-49, July 2001.
[5] S Chawla, H Saraan and M Sing,"Efficiency parameter Based Scheduling for incorporating Variable Rate Coded Voice in Bluetooth". IEEEICC'2001, Vol 4, June 2001.
[6] J.C.R Bennett and H Zhang, "Worst-case Fair Queuing", IEEE INFOCOM'96., pp 120-128,1996
[7] A Demers, S.Keshav, and S.Shenker, "Analysis and simulation of a fair queuing Algorithm". Proc. ACM SigComm'89, Vol. 1, pp 1-12, 1990
[8] Bluetooth Security Architecture white paper URL http://www.bluetooth.com/developer/white paper.asp
[9] D.Zhao, X Shen and J.W.Mask, "Improved Efficiency parameter performance bounds for a Wireless ATM Network", APPCC/OECC' 99, Vol .1,pp 183-187,1999
[10] Bluetooth Special Interest Group, Specification of the Bluetooth Syste Version 1.1, Feb.2001.
[11] J.Bray & C.Sturman, Bluetooth 1.1 connect without cables, Prentice Hall, 2001.
[12] A. Capone, M. Gerla and R Kapoor, "Efficient Polling Schemes for Bluetooth Picocells", Proc. IEEEICC'01, June 2001.