

# PERFORMANCE EVALUATION OF SCHEDULING ALGORITHMS FOR BLUETOOTH

Niklas Johansson, Ulf Körner  
*Department of Communication Systems*  
*Lund University*  
*Box 118, S-221 00 Lund, Sweden*  
niklasj@tts.lth.se, ulfk@tts.lth.se

Per Johansson  
*Ericsson Radio Systems AB*  
*S-126 25 Stockholm, Sweden*  
Per.Johansson@ericsson.com

**Abstract** During the last couple of years, much attention has been brought to research and development of mobile ad hoc networks. In an ad hoc network, a collection of peer wireless mobile users within range of each other may dynamically form a temporary network without the use of any existing network infrastructure or centralized server, as opposed to cellular systems.

This paper analyses the performance of a wireless ad-hoc network concept called Bluetooth which was presented in February 1998 by its five promoters - Ericsson, Nokia, IBM, Toshiba and Intel. We discuss a modified exhaustive scheduler, proposed by the authors, and show its applicability under various operating conditions. A number of scenarios are analyzed and we also address the importance of multi-slot packets to increase throughput and to keep the delays low.

**Keywords:** Ad hoc networking, Bluetooth, performance, polling, wireless networks

---

This work was partially funded by the Foundation for Strategic Research - Personal Computing and Communication

D. H. K. Tsang et al. (eds.), *Broadband Communications*  
© Springer Science+Business Media New York 2000

## 1. INTRODUCTION

Battery powered, untethered computers, Personal Digital Assistants (PDAs), cameras, multimedia terminals, etc. are likely to become a pervasive part of our computing and communication infrastructure. An *ad hoc network* is the cooperative engagement of a collection of these mobile terminals. In such a network, each mobile node operates not only as a host but also as a router, forwarding packets for other mobile nodes in the network that may not be within direct wireless transmission range of each other. Traditionally, ad-hoc packet radio networks have mainly concerned military applications, where a decentralized network configuration is an advantage or even a necessity. For the commercial sector, equipment for wireless mobile computing and communication has not been available at a price affordable for any larger market. However, as capacity of common mobile computers is steadily increasing, the need for untethered networking is expected to do likewise.

This paper analyses the performance of a wireless ad-hoc network concept called Bluetooth which was presented in February 1998 by its five promoters - Ericsson, Nokia, IBM, Toshiba and Intel. These five companies have formed a special interest group, the Bluetooth SIG. The purpose of the consortium is to establish a de facto standard for the air interface and the software that controls it, thereby ensuring interoperability between devices of different manufacturers. Today, more than 800 companies have joined the SIG. The name Bluetooth was taken from Harald Blåtand, a Danish Viking king from the early Middle Age.

Today the media access scheme used in Bluetooth is based on polling, i.e. one of the communicating devices within a Bluetooth picocell acts as a master and polls the other devices. The basic feature of a polling system is the action of centralized intelligence in polling each node on the network following some rules in order to provide access to the channel. As each node is polled, the connected entity may use the full data rate to transmit its backlog for a time depending on how the channels capacity has to be time-shared among the entities, i.e. the multiplexing technique. In between polls, the connected entities accumulate workload, but do not transmit until polled.

In this paper, the original Bluetooth media access principle, i.e. a strict round robin polling algorithm is challenged by a round robin exhaustive polling algorithm and also by a new polling algorithm proposed by the authors. The objective was to analyze some key performance metrics, like delays and throughput, for the above mentioned algorithms under various operating conditions. A number of scenarios are analyzed and we especially stress the importance of multi-slot transmissions to

increase throughput and keeping the delays low. The first and second papers on Bluetooth performance was presented at the IFIP 4th Workshop on Personal Wireless Communications in Copenhagen, April 1999, [1] and at ICC'99 in Vancouver, June 1999, [2].

## 2. SYSTEM DESCRIPTION

A few years ago it was recognized within Ericsson Mobile Communications that a truly low cost, low power radio based cable replacement, or wireless link, was feasible. Hence, the original intension of Bluetooth was to eliminate the cables between cordless phones, laptops, wireless headsets etc. As the project progressed, it became clear that such an ubiquitous link could provide the basis for small portable devices to communicate together in an ad hoc fashion. Today, Bluetooth is a true ad-hoc wireless network intended for both synchronous traffic, i.e. voice, and asynchronous traffic, i.e. IP-based data traffic.

In Bluetooth all units are peer units with identical hard and software interface. Two or more units sharing the same channel form a piconet, where one unit acts as a master (any unit can become a master), controlling traffic on the piconet, and the other units act as slaves. The communication channel operates in the unlicensed ISM band (Industrial-Scientific-Medical band) at 2.45GHz (bandwidth 80 MHz). A frequency hop transceiver is applied to combat interference and fading. The system offers a gross bit rate of 1 Mbps per piconet and allows a mix of voice and data communication channels. The system uses a slotted Time-Division Duplex (TDD) scheme for full duplex transmission, where each slot is *0.625 ms* long (two slots form one frame). Time slots may carry both synchronous information (Synchronous Connection Oriented, SCO, voice links) and dynamically allocated asynchronous information (Asynchronous Connection-less, ACL, data links). A master unit controls the traffic on the channel by allocating capacity for SCO links, i.e. reserving slots for this link, and by using a polling scheme for ACL links. In this paper we especially focus on the efficiency of three different polling algorithms, the strict round robin polling algorithm, the exhaustive round robin polling algorithm and a polling algorithm proposed by the authors. Up to 8 active devices, but to that many inactive devices, may form a Bluetooth piconet. Several piconets can be linked together, thus forming a scatternet in which each piconet is identified by a unique hopping sequence, and as long as not too many, each provides a capacity of almost 1Mbps.

In each slot, a packet can be exchanged between the master and one of the slaves. Each packet begins with a 72 bit access code, which is unique

for each piconet and is derived from the master identity. Recipients in the piconet compare the incoming packet access code with the access code of its own piconet. If the two do not match, they simply discard the received packet. The access code also contains information used for synchronization and compensation for offset. A header trails the access code. The header contains control information such as a three bit MAC-address, flow control, a type field, an ARQ field and a HEC field. Payload may or may not trail the header. The length of the payload may vary from 0 to 2745 bits.

In order to increase throughput under high data rates, so called multi-slot packets may be used. These packets may cover three or five slots. A three slot packet may contain 6 times as much information as a single slot packet and a five slot packet 12 times as much. In this paper we, among others, analyze how performance is increased by using multi-slot packets instead of single-slot packets, but also how the three mentioned polling algorithms behave under different loads.

Data packets are protected by an ARQ scheme in which lost data packets are automatically retransmitted in the next slot. This results in a fast ARQ scheme - delay due to errors is just one slot in duration. Voice traffic is protected by a robust voice encoding scheme and thus retransmission of voice packets never takes place. For a more detailed description of the system, the reader is referred to [3].

### 3. MODEL DESCRIPTION

We have built up a rather comprehensive model of an ad-hoc wireless network that in most respects maps Bluetooth. It is our main purpose to grasp especially the polling mechanism used for the asynchronous links (ACL links).

The simulation studies were made for one piconet comprising of seven units, i.e. one master and six slaves. The traffic is assumed to be symmetric, i.e. one slave sends the same amount of data to all the other slaves. The master does not send any traffic to the slaves but just forwards traffic from one slave to another.

When we allow multi-slot packets, these packets are formed and sent according to the following scheme: a queue with 0 to 2 1-slot packets sends a 1-slot packet, a queue with 3 to 7 1-slot packets sends a 3-slot packet and a queue with more than 7 one slot packets sends a 5-slot packet.

Transmitted packets, i.e. packets on OSI layer 2, may be lost due to bit errors and this is modeled with a constant packet loss probability. All lost packets are being retransmitted according to the Bluetooth ARQ

scheme. In our simulations the packet loss probability is set to  $10^{-4}$  for all types of packet.

### 3.1 THE QUEUING MODEL

The arrival process of IP-packets is assumed to be bursty. To capture the discrete events of packet arrivals as well as the burstiness, arrivals are modeled as a discrete time Interrupted Bernoulli Process (IBP) (see chapter 3.2, The Arrival Process). The arriving IP-packets are being fragmented in order to fit the length of Bluetooth layer 2 packets. The different queues are of infinite size, a valid assumption for this preliminary study.

### 3.2 THE ARRIVAL PROCESS

The arrival process used in this paper is based on an IP trace generated by a single user connected to a 10Mbps Ethernet LAN. Measurements were performed at the Department of Communication Systems, Lund University. A Linux station running the program *tcpdump* was used to observe the web traffic to the station. All IP packets to the station during a period of approximately 2 hours were captured and the packet sizes were logged to a file together with a time-stamp. The measurements have an accuracy of  $1 \mu\text{s}$ . It is assumed that the behavior of the wireless IP-traffic, which we try to emulate in this paper, shows similar characteristics as the LAN IP-traffic.

**IP-Packet Generator.** The number of users in the modeled network is relatively small and they are assumed to transfer web-like traffic. This combination is expected to result in bursty traffic streams in the network, and hence, Poissonian assumptions are not valid. To be able to make a performance analysis we have to use some other arrival model that reflects these bursty arrivals.

In recent literature it is illustrated that e.g. LAN packet traffic has a packet arrival process that displays variability over a variety of time-scales and causes queuing problems at relatively low loads [5] [6]. This has spurred research in the area of traffic models inherently more suited for modeling second order self similar behavior, e.g. Fractional Brownian Motions (FBM) [7] and chaotic maps [8]. For these models, however, the tools for analyzing queuing behavior are still in their embryonic state. Additionally most of the novels are focused on capturing the first- and second order characteristics of counts that, in general, are known to be insufficient when attempting to predict queuing behavior [9].

Since the knowledge of traffic patterns in ad-hoc networks is limited and those traffic patterns also shift from application to application, we have chosen to model our packet generator as an Interrupted Bernoulli Process (IBP). That is, for a geometrically distributed period (ON state) arrivals occur according to a Bernoulli process. This period is followed by another period (OFF state), also geometrically distributed, during which no arrivals occur (The IBP is a special case of the Markovian Arrival Process (MAP) in discrete time). This is a very straightforward model that provides a bursty, but controlled, traffic characteristic, sufficient for this preliminary analysis.

In an IBP the probability for sending a packet during a slot (Note that the time scale of the IBP is almost 50 times shorter than that of the modeled piconet) is set to 1 in the ON state and to 0 in the OFF state. If  $p$  is the transition probability from the OFF to the ON state and  $q$  is the transition probability in the opposite direction, it can be shown that the probability generating function of the inter arrival times is given by,

$$P(z) = \frac{(1 - q - z(1 - p - q))z}{1 - (1 - p)z} \quad (1)$$

thus we get the load,  $\rho$ , on the piconet channel as,

$$\rho = \frac{N \cdot N_B \cdot \frac{S_B}{S_{IBP}}}{1 + q/p} \quad (2)$$

where,

- $N$ =Number of active units.
- $N_B$ =E[Number of Bluetooth packets per IP packet].
- $S_B$ =Bluetooth slot length.
- $S_{IBP}$ =IBP slot length.

The above equation holds for the case when only single slot packets are used. If multi-slot packets are allowed, a term compensating for this has to be multiplied to the right hand side of equation (2). This term is hard to predict since the fraction of the different packets is not known, but with the aid of simulation one can measure the occurrence of the different packet types and calculate the actual load.

The squared coefficient of variation,  $C^2$ , for the IP-packet inter arrival times were used as a measure of the burstiness in the simulations. From

equation (1) we get the squared coefficient of variation as,

$$C^2 = \frac{q(2-p-q)}{(p+q)^2} \quad (3)$$

Now, by varying  $p$  and  $q$  we can alter the load while keeping the squared coefficient of variation  $C^2$  constant.

The IBP only generates the instances when IP packets are generated. When a packet is generated the length of this packet is drawn from essentially a two point distribution, retrieved from the LAN-trace. The IP packets are further fragmented into Bluetooth layer 2 packets.

## 4. SCHEDULING

To be able to provide Quality of Service (QoS) capabilities that support bandwidth allocation and latency control, some form of intelligent scheduling algorithm is needed. The decision process, based on polling, should be managed by the master alone and handled by a slave scheduler located in the master entity. The decision should be based on the aggregate state of the slave queue and the master output queue for that particular slave.

There are generally two ways to guarantee the negotiated QoS. One is to poll the slaves in a round robin fashion, the other is to associate a local time-stamp to each packet and serve the packet with the least time-stamp. One example of a scheduler using local time-stamps is Weighted Fair Queuing (WFQ) [10]. It has the advantage that it may guarantee a minimum rate of service with arbitrarily fine granularity, however, it is computationally complex and require storage of time-stamps for each packet. Round robin schedulers can guarantee a minimum rate of service and require minimal computation to service each packet, however, they have finite bandwidth granularity and it may therefore be more difficult to modify the bandwidth allotment. Since the schedulers are supposed to run in "thin clients", minimal computation complexity is very important and this is the reason why we decided to focus on round robin schedulers.

In this paper we have focused on three different scheduling algorithms, namely: round robin, exhaustive and a modified exhaustive scheduling algorithm herein called Fair Exhaustive Polling (FEP).

### 4.1 STRICT ROUND ROBIN POLLING

In the round robin service system [11], in some literature also known as limited service system [12], the master polls the slaves in consecutive order. Each slave is allowed to send just one packet per service cycle. With this round robin scheduler the master can assign different

bandwidths to different units by allowing different packet types to the users (see chapter 2, System Description). This has the effect that users with higher demands on bandwidth get a fair share of the excess bandwidth, which is a desirable feature since we then can guarantee max-min fairness [10].

## 4.2 EXHAUSTIVE POLLING

In the exhaustive service system, the master continuous to poll the addressed slave until the slave queue and the master output queue for the specific slave is emptied. This means, that not only those packets present in the queue at the beginning of the service cycle but also those packets generated during the service cycle are served before the master moves on to the next slave. This is of course not an optimal scheduling algorithm since this procedure will favor slaves generating packets at maximum rate. Therefore, an additional parameter has been added, which limits the service time to some predetermined maximum time.

With this scheduler the master may assign different bandwidths to different units by allowing different packet types, i.e. 1, 3 or 5-slot packets, and different maximum poll time for each individual slave. Also for this scheduler one can guarantee the max-min fairness, at least over time scales larger than the maximum sojourn time [13].

## 4.3 FAIR EXHAUSTIVE POLLING (FEP)

The fair exhaustive polling service system can be viewed as a combination of the two earlier mentioned polling algorithms. The main idea behind FEP is to poll slaves that probably has nothing to send as seldom as possible.

The slaves belong to one of two complementary states, the *active state* and the *inactive state*. A polling cycle starts with the master moving all slaves to the active state, and then begins one of possibly several polling sub cycles. In a polling sub cycle all *active* slaves are polled once in a round robin fashion. As mentioned above, the master performs the task of packet scheduling for both the downlink and uplink flows, however, the master has only limited knowledge of the arrival processes at the slaves. This means that the scheduling, of the uplink flows, has to be based on the feedback it gets when polling the slaves. Based on this feedback and the current state of the master output queues, slaves are moved between the different states. A slaves is moved from the active state to the inactive state when both of the following conditions are fulfilled:

- The slave has no information to send



- The master has no information to send to the specific slave

A slave is moved to the active state when the master has information to send to it.

This is an iterative process that continues until the active state is emptied (the exhaustive part of the algorithm) and when it is, a new polling cycle starts.

Also this algorithm will favor slaves generating packets at maximum rate and therefore an additional parameter has been added, which limits the polling interval of any slave to some predetermined maximum time. For the algorithm this means that a slave, whose maximum polling interval timer has expired, is moved to the active state and therefore will be polled in the next polling sub cycle.

FEP behaves, asymptotically, as an exhaustive scheduler at low loads and as a round robin scheduler at high loads. This is a very attractive feature since the mean delay for the exhaustive scheduler is less than or equal to the mean delay for the round robin scheduler for low traffic, whereas the opposite holds for heavy traffic [12]. We have, through simulation, been able to show that the FEP scheduler experiences lower delays than both the exhaustive- and the round robin scheduler at all loads.

With this scheduler the master may assign different bandwidths to different units by allowing different packet types, i.e. 1, 3 or 5-slot packets, and different maximum poll intervals for each individual slave. Also for this scheduler one should be able to guarantee the max-min fairness, at least over time scales larger than the maximum sojourn time.

## 5. SIMULATION RESULTS

In [1] and [2] we have analyzed a number of different situations under the scenario where slaves send web-like traffic to other slaves. There we analyzed symmetric as well as non symmetric load situations for both bursty and non bursty traffic characteristics. We also analyzed the transient behavior of the Bluetooth network when one of the slaves send a very dense burst for a short period.

In this paper the aim of our simulations is twofold. On one hand a case using only single slot packets is compared with one that allows multi slot packets, i.e. 3 or 5 slot packets as well. A 3 slot packet may hold up to 6 times the payload of a 1 slot packet and a 5 slot packet up to 12 times. Furthermore, which is the main focus of this article, the three scheduling algorithms mentioned in the previous chapter are compared. The strict round robin was initially chosen as the media access protocol for Bluetooth. Here we compare performance metrics of the strict round

robin with the classical exhaustive polling algorithm and what we refer to as the Fair Exhaustive Polling algorithm, (FEP).

It is well known that the round robin scheduler outperforms the exhaustive scheduler under high symmetric loads and vice versa under low loads. Our algorithm, the FEP, more or less works as the exhaustive under low loads and as the round robin under high loads and thus shows to be a good alternative for a Bluetooth network especially and for most polling based networks in general.

In order to evaluate these algorithms, we have formed a worst case scenario in that we feed the Bluetooth network with traffic that really stresses the network. The traffic is modeled first at an IP level with a rather high squared coefficient of variation equal to more than 60. (For the reader we again mention that the squared coefficient of variation for the arrival process is equal to the ratio of the variance of the inter arrivals between subsequent IP packets and that of the squared mean.). To that comes the fact that IP-packets normally are much longer in bits than that of a Bluetooth-packet. Thus, at most IP-packet arrivals, a number of Bluetooth packets are generated. We experience extremely long delays but this is so just for us to compare the algorithms under high loads.

In figure 1, where we allow multi slot packets, the average delays are plotted against the total arrival rate in bits to the entire Bluetooth piconet. We observe here that the exhaustive algorithm outperforms that of round robin, while our algorithm, the FEP, outperforms that of the others at high as well as low loads. We also observe from figure 2 that the same behavior also holds for single slot packets though the delays are even longer. It is also worth noting that the round robin policy performs better than the exhaustive policy at high loads and vice versa for low loads. Figure 3 and 4 show the squared coefficient of variations for the end to end delays for the two cases discussed above. Here also our FEP outperforms the two others by a large factor.

## 6. CONCLUSIONS AND FURTHER WORK

In this paper we have analyzed and compared the behavior of three different polling algorithms used as medium access schemes for an ad hoc wireless network called Bluetooth. When Bluetooth was announced, in late spring 1998, the polling scheme proposed was based on a strict round robin principle. This requirement has now been relaxed and we have in this paper been able to demonstrate the strength of a proposed scheduler, herein called the Fair Exhaustive Polling (FEP) scheduler.

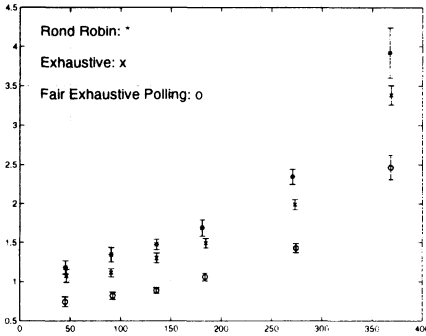


Figure 1 Average delay [s] against bitrate [kbps] for the three scheduling algorithms, multi-slot packets allowed.

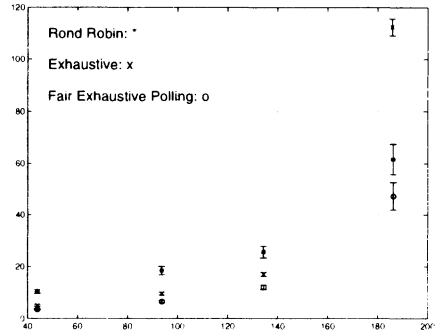


Figure 2 Average delay [s] against bitrate [kbps] for the three scheduling algorithms, only single-slot packets allowed.

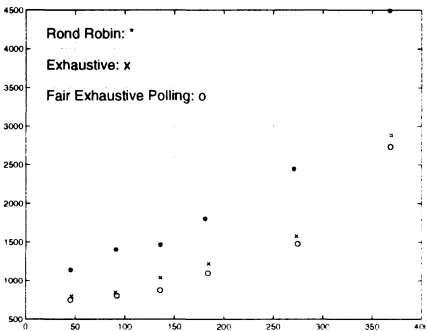


Figure 3 Squared coefficient of variation against bitrate [kbps] for the three scheduling algorithms, multi-slot packets allowed.

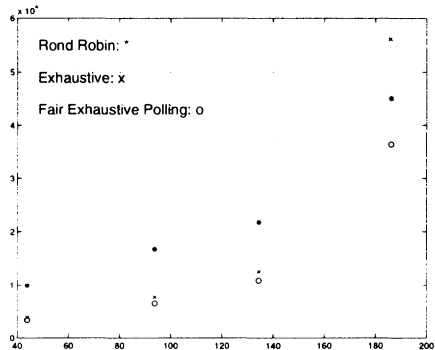


Figure 4 Squared coefficient of variation against bitrate [kbps] for the three scheduling algorithms, only single-slot packets allowed.

We have also clearly demonstrated the increase in performance when allowing packets to be sent in so called multi-slots. Not at least when traffic tends to be very bursty, the use of multi-slots has proven to be efficient. This is of course natural as the transmission overhead dramatically decreases.

We will in coming papers refine the FEP scheduler to also include inter-piconet capabilities. We will also address the question of under which circumstances new piconets should be established for devices in an existing piconet when traffic patterns change dynamically.

## References

- [1] Niklas Johansson, Ulf Körner and Per Johansson. Wireless Ad-hoc Networking with Bluetooth. *4th Workshop on Personal Wireless Communications*, Copenhagen, 1999.
- [2] Per Johansson et al. Short Range Radio Based Ad-hoc Networking: Performance and Properties. *ICC'99*, Vancouver, 1999.
- [3] Jaap Haartsen. Bluetooth - The universal radio interface for ad hoc, wireless connectivity. *Ericsson Review*, no 3, 1998
- [4] The Bluetooth Special Interest Group, Documentation available at <http://www.bluetooth.com/>
- [5] Ashok Erramilli, Onuttom Narayan and Walter Willinger. Experimental Queueing Analysis with Long-Range Dependent Packet traffic. *IEEE/ACM Transactions on Networking*, 4(2):209-223, April 1996.
- [6] Will E. Leland, Murad S. Taqqu, Walter Willinger and Daniel V. Wilson. On the Self-Similar Nature of Ethernet Traffic (Extended Version). *IEEE/ACM Transactions on Networking*, 2:1-15, February 1994.
- [7] I. Norros. A storage model with self similar input. *Queueing Syst.*, vol. 16, pp. 387-396, 1994.
- [8] A.Erramilli, R.P. Singh and P. Pruthi. An application of deterministic chaotic maps to model packet traffic. *Queueing Syst.*, vol. 20, pp. 171-206, 1995.
- [9] Kerry W. Fendick and Ward Whitt. Measurements and Approximations to Describe the Offered Traffic and Predict the Average Workload in a Single-Server Queue. *Proceedings of the IEEE*, 77(1):171-194, January 1989.
- [10] S. Keshav. An Engineering Approach to Computer Networking. *part of the Addison-Wesley Professional Computing Series*, 1998. ISBN 0-201-63442-2.
- [11] Hideaki Takagi. Exact analysis of round-robin scheduling of services. *IBM J. Res. Dev.* 31, 4 (July), 484-488.
- [12] Hideaki Takagi. Queueing Analysis of Polling Models. *ACM Computing Surveys*, Vol. 20, No. 1, March 1988.
- [13] Uri Yechiali. Optimal Dynamic Control of Polling Systems. *ITC-13*, Copenhagen, 1991.