# Maximizing efficiency in Bluetooth Piconets using Throughput Optimal Packet Size Selection (TOPS)

Saswati Sarkar[a] *, Farooq M. Anjum[b] and Ravi Jain[c]

[a]Dept. of ESE, UPenn, Philadelphia, PA USA

[b]Telcordia Tech Inc, Morristown NJ USA

[c]NTTDoCoMo, SanDiego, CA USA

Bluetooth wireless links use frequency hopping whereby each packet is sent on a single frequency while different packets are sent on different frequencies. Further, there are a limited number of packet sizes. We propose a scheduling algorithm to choose the packet size transmitted over each frequency as a function of the channel conditions, in order to maximize throughput in a Bluetooth piconet.

Our contributions in this paper are as follows. We first develop a renewal theory based mathematical model of packet transmission in a frequency hopping system like a Bluetooth piconet. We use this model to show that given the objective of maximizing the throughput of the system, the decision about the packet length to be transmitted for a given frequency is a threshold based process. We then provide an algorithm that determines these optimal thresholds efficiently for a given system under certain assumptions. We show the optimality of this algorithm without using standard optimization techniques since it is not clear that these techniques would be applicable given the functions involved. Using simulations, we observe that this strategy leads to significantly better throughput as compared to other baseline strategies even if the assumptions made to prove optimality are relaxed. We then extend our results to multiple active slaves in a piconet. In this case, we show that scheduling along with packet size selection can significantly improve the system throughput. To our knowledge, this is the first attempt to introduce advanced probabilistic arguments for optimizing the throughput of Bluetooth transmissions.

## 1. Introduction

There has recently been tremendous interest in applications of Bluetooth wireless technology. Bluetooth is a low-power, low-cost short range wireless communication system [4,7,11]. It enables small portable devices to connect to each other and communicate in an ad-hoc fashion with nominal speeds of up to 1 Mbps. Bluetooth's low range (typically 10m) and low cost make it suitable as a single-hop cable replacement, such as eliminating the wire from a personal portable radio player to the headphones. Industry analysts have estimated that 13 million bluetooth devices were shipped in 2001, meeting earlier forecasts, and predict that

*The work of S. Sarkar was supported in part by NSF grant ANI01-06984

worldwide by 2005 there will be over 780 million new Bluetooth devices shipped [3].

Bluetooth uses frequency hopping and operates in the unlicensed 2.4GHz ISM band, which is also used by IEEE 802.11 radios as well as other devices such as microwave ovens, baby monitors etc. Thus, it can be expected that the frequencies in this band will be subjected to interference from other sources in addition to being subjected to the vagaries of wireless links. In this paper we concentrate on providing solutions for enabling efficient communication between Bluetooth devices in the face of such interference. Rather than mitigating these effects by increased error correction, improved power control, or other lower-layer techniques, we consider how scheduling packet transmissions and modifying the length of packet transmissions in response to the current channel conditions can be used to improve system throughput.

The basic idea of the paper is as follows. A Bluetooth cell (or *piconet*) consists of a *master* device and up to seven active *slaves*, with all communication taking place only between the master and the slaves. Time is divided into slots and the master controls which slave can communicate and in which time slot. Bluetooth uses 79 different frequencies during its frequency hopping sequence. The master's fixed Bluetooth address determines the hopping sequence used within the piconet; thus each time slot in a given piconet has a particular frequency used in that slot. At any given time each frequency can have a different quality, in terms of Bit Error Rate (BER), associated with it. If the BER for a particular frequency can be estimated, the master can choose the length of the packet sent over that frequency. Intuitively, shorter packets are sent over frequencies with high BER so as to reduce the probability that they are lost. Further, the BER to each slave may be different and time-varying, depending on its distance from the master and other factors. If the BER to a particular slave can be estimated, the master can choose how to schedule communication with that slave at a given time. Thus slaves which have a high BER at the moment may be temporarily skipped in favor of slaves with a low BER, thus reducing packet loss and increasing throughput.

Bluetooth packets are restricted to be of three lengths, namely 1, 3 or 5 time slots. For connectionless data we consider what are called in Bluetooth DH packets. DH1 packets of length 366 bits are sent using 1 time slot, DH3 packets of 1622 bits using 3 time slots, and DH5 packets of 2870 bits using 5 time slots. Once a time slot is chosen at the start of a packet transmission, the entire packet is sent at the frequency of the first time slot irrespective of the length of the packet, and a new frequency is used only for the next packet. Given the fixed overhead of packet headers a longer packet is more efficient in terms of user data throughput. In addition to the overhead considerations, the effect of choosing the packet length in response to the BER can be very signicant in terms of the throughput attained on account of the characteristics of the frequency channel; we use a simple example to illustrate this.

**Example.** Consider a Bluetooth device A transmitting to a device B, using only two frequency channels, namely $f_1$ and $f_2$, with associated BER of $10^{-3}$ and $10^{-4}$ respectively. Let the frequency sequence alternate between $f_1$ and $f_2$. Consider the effect of the following three transmission strategies: A transmits all packets as DH5 packets; it transmits DH5 packets on frequency $f_1$ and DH1 on $f_2$; or it transmits DH1 on $f_1$ and DH5 on $f_2$. It can be shown that a DH5 packet sent on $f_1$ experiences a Packet Error Rate (PER) of 0.94 while a DH5 packet sent on $f_2$ experiences a PER of 0.249; the corresponding values for a DH1 packet are 0.306 and 0.036. Therefore, the throughput with the first strategy is 36657.2 bytes per sec, with the second is 9273.6 bytes per sec and 54665.8 bytes per sec with the third strategy.

The example indicates that by judiciously adapting the packet length to channel conditions, we can increase the throughput of the system significantly. Hence, given the precious nature of bandwidth in a wireless network, we aim to provide algorithms that can choose the packet lengths so as to achieve the maximal bandwidth for the given channel conditions.

The problem of choosing packet lengths in lossy channels has been investigated widely for systems with only one frequency of transmission [5,8]. The aim in previous work has largely been to trade off the overhead of packet headers with the PER, i.e., that small packets have more overhead than longer ones but have an increased probability of successful transmission. Our work differs on account of the focus and exploitation of the nature of frequency hopping systems such as Bluetooth. Thus, we consider the case whereby the entire packet is sent on a single frequency with different packets being sent on different frequencies. Hence, we can mitigate the effects of channel conditions by selecting the packet length to be transmitted over a frequency as a function of the frequency channel conditions. As illustrated in the example earlier, by deciding to send larger packets over good frequency channels and small packets over bad frequency channels, we can increase the fraction of time the system experiences good transmission conditions and thereby maximize the bandwidth attained by the system. Frequency hopping in Bluetooth means that the BER for each frequency and each time slot has to be considered, but this also means that the sender has more choices. In addition, packet lengths cannot be chosen arbitrarily but are restricted to specified values. Finally, the master-slave relationship in Bluetooth piconets means that the master can schedule which slaves receive or transmit information, what packets lengths are used by the master and which time slots are used for a slave thus allowing finer-grained control over the system throughput.

The paper is organized as follows. In the following section we develop a mathematical model of the problem and using this model propose an optimal algorithm considering both a single slave piconet and a multiple slave piconet. In sec. 3 we present our simulation model and results for the single slave case, as well as for the multiple slave case. We end with a brief discussion and conclusions. We omit proofs of all theorems in this paper due to lack of space. These are available in a detailed version of this paper at [10].

## 2. System Model and Analysis

Our goal is to provide a packet size selection algorithm that can maximize the throughput in a Bluetooth piconet. For simplicity, we consider one way data transfer from the master to the slave. For this purpose we first develop a mathematical model for capturing the throughput optimization problem in the Bluetooth scenario. Subsequently, we exploit the model to generate an optimal solution to the problem of interest. Even though we consider the specific case of Bluetooth in this paper, the model we develop applies to any frequency hopping system where a single packet must be transmitted on a single frequency. We initially consider a piconet with a single slave only. Later we extend this model to consider a piconet with multiple slaves.

## 2.1. Piconets with one slave

We will first explain the model and subsequently justify it. Consider a Bluetooth piconet with a single slave. The master of the piconet is responsible for transmitting packets to the slave using frequency hopping. The master can choose from three different packet lengths namely 366 bits, 1622 bits and 2870 bits. These packets occupy one, three or five bluetooth slots; each slot of length 625 microseconds. We assume that there are multiple sessions (at least three) from a master to every slave with an infinite supply of data for each session. Each session uses a single baseband packet size i.e. DH5, DH3 and DH1. This is justified since our objective is to determine the maximum throughput possible. Thus this translates to an unlimited supply of each packet type. Segmentation is assumed to occur at the L2CAP layer. Let the probability of successful transmission of each bit in a given packet be $p$ (BSR$^{\dagger}$). We assume that $p$ is a random variable with cumulative distribution function given by $F(p)$. All bits of a packet are assumed to have the same BSR while the BSR associated with different packets are mutually independent. A packet is corrupted if even a single bit is in error.

Now we justify the model. A packet is transmitted on a single frequency irrespective of its length. The BSR associated with a bit depends strongly on the frequency as transmission conditions can be poor in one frequency while they may be good in another. If the frequencies are random variables, then the BSR is also random. In Bluetooth, frequencies in the frequency hopping sequence are generated by a pseudorandom number generator seeded by the masters address and clock. Thus the frequencies constitute a "pseudo-random variable" and this motivates the above model. However we would like to mention that since the frequencies are generated by a PN sequence, all the frequencies can be known ahead of time. If the BSRs associated with these frequencies are known ahead as well, then the entire packet length sequence can be determined using a deterministic optimization. However, the frequency sequences are usually long and contain several different frequencies. Thus the optimization will be computationally intensive. Besides the storage of the results and the table lookup will also require substantial space. More importantly, BSR is not a deterministic function of frequency since the transmission conditions vary with time for the same frequency. Thus the BSR is not known ahead of time even if the frequency sequence is. Hence the previous knowledge of the frequency sequence may not be useful. This motivates our model which assumes that the frequencies are random variables, and subsequently the associated BSRs are also random variables. We do not assume any specific characteristic of the cumulative distributed function of the random variable $p$ (BSR).

A packet is transmitted on a single frequency independent of its length. The length of the packet is at most 5 slots (3.125 ms). Thus transmission conditions associated with a frequency do not change significantly in a short duration; this is especially so when considering wireless devices that are not mobile. This motivates the assumption of the same BSR for all bits in a packet. We assume that the BSRs for different packets are mutually independent. This assumption is motivated by the fact that different packets are transmitted in different frequencies and transmission conditions may be quite different for different frequencies. Note that bursty losses would be realistic in the absence of frequency hopping. However as the frequencies to be used for transmission are generated as a pseudorandom number sequence, assuming independent packet losses is justified.

We assume that at the beginning of a slot, the master knows the BSR governing the

$^{\dagger}$BSR = 1 - BER, where BER is the bit error rate.

bits of the packet to be transmitted in the slot, and uses this knowledge in deciding the packet length. We use this assumption mainly for analytical ease. This assumption is not necessary for the execution of the proposed algorithm; the algorithm can be executed based on estimates of channel conditions also. We would also like to remark that the assumption regarding knowledge of channel conditions has been made elsewhere as well[6,9]. Finally, we also assume that the reverse channel from the slave to the master is non-lossy. This can be justified on the basis of the small size of the acknowledgement packets sent on the reverse channel in case of one way communication from the master to the slave. Our model can be extended to cater to two way communications with a lossy reverse channel. Later, we will relax these assumptions in our simulations. Now we demonstrate how this model can be used to provide a throughput optimal packet length choice algorithm, which is simple to implement.

Given the BSR $p$ for the frequency on which a packet is to be transmitted, the master of the piconet decides the packet length $l(p)$ bits, as a function of the BSR, at the beginning of the packet transmission. The master has three choices for $l(p)$ as discussed before. Since BSRs of different packets are mutually independent, lengths of different packets are also mutually independent. Thus the packet transmission process is a renewal process, with the system renewing itself after every packet transmission[12]. We will now evoke renewal reward theory to characterize the throughput of any arbitrary packet length selection rule $l(p)$. Let the cumulative distribution function corresponding to the BSR be denoted by $F(p)$. Then, the average duration of a single renewal period is the average length of a packet which we denote as $\bar{l}(p)$, $E\left(\bar{l}(p)\right) = \int_0^1 \bar{l}(p)dF(p)$. Note that $\bar{l}(p)$ and $l(p)$ are closely related for Bluetooth with $\bar{l}$ taking the values of .625msec, 1.875msec and 3.125msec for $l$ values of 366, 1622 and 2870 bits respectively. Thus, once the master decides $l$ based on $p$, $\bar{l}$ is also automatically determined based on the choice of $l$. There is a reward associated with every packet transmission. If the packet has $l$ bits, then the reward is $l$ if all the bits of the packet reach the receiver successfully and 0 if at least one bit reaches the receiver in error. Thus the expected reward for a packet is $l(p)p^{l(p)}$. Now, given the cumulative distribution function $F(p)$, the expected reward for any packet is $E\left(l(p)p^{l(p)}\right) = \int_0^1 l(p)p^{l(p)}dF(p)$. Throughput of the system is the number of successful bits transmitted per unit time. Thus the throughput equals the average reward per unit time, $\lim_{t\to\infty} E\left(R(t)\right)/t$, where $R(t)$ is the total reward obtained before time $t$. Using renewal-reward theorem [12],

$$\lim_{t\to\infty} E\left(R(t)\right)/t = \frac{E\{l(p)p^{l(p)}\}}{E\{\bar{l}(p)\}} = \frac{\int_0^1 l(p)p^{l(p)}dF(p)}{\int_0^1 \bar{l}(p)dF(p)}$$

Thus the throughput $\eta_l$ of packet length selection rule $l(p)$ is

$$\eta_l = \frac{\int_0^1 l(p)p^{l(p)}dF(p)}{\int_0^1 \bar{l}(p)dF(p)} \tag{1}$$

We next wish to find a packet length selection rule $l(p)$, that maximizes equation 1. The optimal strategy will clearly depend on the distribution for $p$, $F(p)$. However, we first present a generic property of the optimal strategy which holds irrespective of the distribution.

**Theorem 1** *The optimum packet length $l^*(p)$ is a threshold based decision process for any cumulative distribution function, $F(p)$. Hence, there exist thresholds $p_1^*, p_2^*$ such that the* *optimal packet length, $l^*(p) = 366$ bits (1 slot) for $p < p_1^*$, $l^*(p) = 1622$ bits (3 slots), $p_1^* \leq p < p_2^*$, and $l^*(p) = 2870$ bits (5 slots) $p \geq p_2^*$. The values of the optimal thresholds depend on the distribution $F(p)$.*

We now present the intuition behind the above result. Different packets are transmitted under different frequencies and hence different channel conditions. If the channel conditions are good (high BSR), then large packets must be transmitted. This follows since the frequency and hence the BSR does not change until the packet transmission ends, and thus larger number of bits will be transmitted under better channel condition which improves the throughput. Similarly, for poor channel conditions (low BSR), small size packets are advantageous as this exposes fewer bits to poor channel conditions. At the same time, under similar transmission conditions large packets are more likely to have error in at least one bit than small packets, and thus transmitting excessively long packets may be detrimental for throughput. This advocates for a cautious increase in packet lengths with improving channel conditions. Thus a threshold type decision process agrees with the intuition that packets of smallest size be transmitted for low values of BSR, packets of medium size be transmitted for middle values of BSR and packets of largest size be transmitted for high values of BSR. The significance of this result is that independent of the distribution $F(p)$, a broad nature of the optimal decision strategy is known.

We next discuss the computation of the optimal thresholds, $p_1^*, p_2^*$. Define a function $f(a,b)$ as $f(a,b) = A/B$ where:

$$A = \int_0^a 366x^{366} dF(x) + \int_a^b 1622x^{1622} dF(x) + \int_b^1 2870x^{2870} dF(x)$$
$$B = \int_0^a 625 dF(x) + \int_a^b 1875 dF(x) + \int_b^1 3125 dF(x)$$

From (1) and Theorem 1,

$$(p_1^*, p_2^*) = \arg \max_{0 \leq a \leq b \leq 1} f(a,b)$$
$$\eta_{l^*} = f(p_1^*, p_2^*) \text{ (Optimal throughput)}$$

Thus the threshold determination process involves maximization of function $f(a,b)$ with respect to the arguments $a, b$. The function $f(a,b)$ can be non-concave depending on the distribution $F(p)$. In general non-concave functions can have several global maximas and also local maximas which differ from the global maximas. The usual gradient search based optimization algorithms[1] are not guaranteed to converge or attain the global optima in this case. Thus *the rich body of optimization literature can not be used to design an efficient algorithm in this case.* However, using the structure of $f(a,b)$ we have devised a single-variable optimization based iterative technique which is guaranteed to converge to the global optimas for any distribution function $F(p)$. The basic idea is to iterate in two variables, $p_1, p_2$ until they converge to the optimum thresholds $p_1^*, p_2^*$. The procedure starts with $p_1 = p_2 = 0$. In each iteration, first $f(p_1, b)$ is maximized by varying $b$ in the range $[p_1, 1]$. If the value of $b$ which maximizes $f(p_1, b)$, $b_{\max}$ is different from the current value of $p_2$, then $p_2$ is set to $b_{\max}$, otherwise the next step is to maximize $f(a, p_2)$ with respect to $a$ in the range $0 \leq a \leq p_2$. The procedure terminates when a fixed point is reached. In other words the algorithm terminates when these maximizations do not alter the value of $p_1$ and $p_2$ any longer, i.e.,

$f(p_1, p_2) = \max_{0 \le a \le p_2} f(a, p_2)$ and $f(p_1, b) = \max_{p_1 \le b \le 1} f(p_1, b)$. Thus the strategy is to reach the global maxima via a sequence of maximizations in one variable. We present the algorithm next. This algorithm has been proven to converge to optimal thresholds [10].

**Throughput Optimal Threshold Selection (TOTS) Algorithm**

1. $p_1^0 = p_2^0 = 0$, $k = 1$.
2. $p_2^k = \arg\max_{p_1^{k-1} \le p \le 1} f(p_1^{k-1}, p)$
3. If $p_2^k \ne p_2^{k-1}$, $p_1^k = p_1^{k-1}$. Go to step (5)
4. If $p_2^k = p_2^{k-1}$, then
   - $p_1^k = \arg\max_{0 \le p \le p_2^k} f(p, p_2^k)$
   - Terminate if $p_1^k = p_1^{k-1}$.
5. $k \to k + 1$ Go to step (2).

From an implementation perspective, for known stationary channels the optimal thresholds can be computed off-line as per TOTS, and packet lengths decided each time by comparing the respective BSRs to these thresholds. For unknown, stationary or time varying, channels the optimal thresholds will have to be computed on-line after estimating the channel conditions. Note that the above algorithm is not complex to implement in resource constrained wireless devices.

Finally, we would like to mention that the renewal reward framework can be generalized to accommodate many other attributes. For example, we assumed that a packet is successfully received only if all bits are received without error (thereby ignoring the error correction schemes used in the Bluetooth baseband packet headers). However, if error correction schemes are used (as is done for the headers of the baseband packet types or with other Bluetooth packet types such as DM5, DM3 and DM1), then a packet can be successfully received even if a few bits are in error. In a more general setting a packet is successfully received as long as at least $k$ bits are correct. The renewal reward setting can accommodate this by assuming that a packet of length $l$ is associated with a reward of $l$ bits as long as $k$ bits are correct. This generalization can be accommodated by altering the average reward (numerator) in the throughput expression in (1) to $\int_0^1 l(p) \left( \sum_{i=k}^{l(p)} \binom{l(p)}{i} p^i (1-p)^{l(p)-i} \right) dF(p)$.

## 2.2. Multiple devices

We next consider a piconet with $k$ slaves $1 < k \le 7$. When the master of such a piconet transmits a packet, it has to make two decisions; firstly the master has to choose the slave which it must transmit to and secondly the master has to decide the size of the packet to be transmitted to the chosen slave.

The same frequency is used irrespective of the slave chosen. Even then, the BSR may be different for different slaves. Hence, in any slot we consider separate BSRs for different slaves, $p_1, \ldots, p_k$. These are random variables governed by a joint cumulative distribution function denoted as $F(p_1, p_2 \cdots, p_k)$. As before, we consider that all the bits of a packet experience the same BSR, and a packet is lost if one or more bits are in error. Further the master knows the BSR $p_1, \ldots, p_k$ for all the slaves at the beginning of transmission of a packet, and uses this knowledge to choose the appropriate slave and the packet length. Finally, BSRs for different packets are mutually independent. We justify all these assumptions in a similar manner as earlier.

Since packet lengths depend on the BSRs only, and BSRs for different packets are mutually independent, packet transmission is a renewal process. System throughput can be related to the packet length selection policy using the renewal reward theorem[12] as in the single slave case. Let a policy $\pi$ choose the slaves as per function $s(p_1, \ldots, p_k)$ and the packet length as per $l(p_1, \ldots, p_k)$, where $p_1, \ldots, p_k$ are the BSRs of slaves $1, 2, \ldots, k$. The throughput of this policy, $\eta_\pi$ is given by

$$\eta_\pi = \frac{\int_0^1 \ldots \int_0^1 l(p_1, \ldots p_k) p_{s(p_1, \ldots, p_k)}^{l(p_1, \ldots p_k)} dF(p_1, \ldots, p_k)}{\int_0^1 l(p_1, \ldots p_k) dF(p_1, \ldots, p_k)} \tag{2}$$

The objective is to design the optimal strategy $\pi^*$ which maximizes the throughput $\eta_{\pi^*}$. We describe such a strategy next. The basic idea is to select the slave with the best transmission condition (highest BSR), and then choose the packet length as per the channel conditions of this slave, using the optimal strategy in the single slave case. More specifically, the optimal packet length thresholds for each slave are pre-computed as per the TOTS algorithm, using the cumulative distribution function $F_i(p)$ for each slave $i$ (the individual cumulative distribution functions $F_i(p)$, $i = 1, 2, \ldots, k$ can be obtained from the joint distribution functions $F(p_1, \ldots, p_k)$). Whenever a slave $i$ is selected, i.e., $p_i = \max_j p_j$ the packet lengths are selected as per the optimal threshold driven procedure for slave $i$.

We have shown analytically ([10]) that the above policy attains the maximum possible throughput in the multiple slave case for any joint distribution function $F(p_1, \ldots, p_k)$ and any number of slaves $k$. The intuition behind the optimality result is as follows. Once the slave is chosen, intuitively throughput is optimized if the packet length is selected as per the optimal strategy for the slave. Thus the task is to choose the slave optimally. Intuitively, the choice of the slave with the best channel condition maximizes the probability of successful transmission. In fact, in a different context the authors in [2] showed that throughput is maximized when the transmission is over the channel with the least probability of error. This paper applies to generic wireless channels and not specifically to Bluetooth, and as such does not contain the packet length optimization problem we are focusing on. Nevertheless, the receiver selection strategy turns out to be identical in both.

## 3. Simulation Results

We use a simulation model to investigate the performance advantage of the optimal algorithm as compared to benchmark algorithms. We examine the performance both in ideal conditions as well as by relaxing some of the assumptions used in the analytical model as we explain later. The different benchmark algorithms used for comparison are also explained later. We use Matlab for these simulations. A single piconet consisting of a master and different slaves is created. We do not consider the piconet formation phase in our simulations. In all the simulation results, the throughput shown is the average over 10 trials, where each trial consists of 1600 Bluetooth slots. We consider a Rayleigh error model for the frequency channels in all the figures.

The theoretical model developed in the previous section is used to determine an optimal packet length selection algorithm. The model is based on assumptions of which we relax the assumption that the BER for a particular frequency is known. We call this BER relaxation. We consider the performance of the optimal algorithm considering only piconets with multiple slaves due to lack of space. Here we will investigate the following algorithms as benchmarks to compare with the optimal algorithms.

**Round robin DHx (rrdhx)**: The master transmits only DHx (where $x = 1, 3, 5$) packets to each slave in a round robin fashion without regard to the channel condition for each slave. Thus this strategy is oblivious both in regards to choosing packet length and scheduling.

**Round robin optimal (rropt)**: The master transmits to each slave in a strict round robin fashion, but chooses the packet size to transmit based on the channel conditions to that slave. The master is thus oblivious with respect to scheduling.

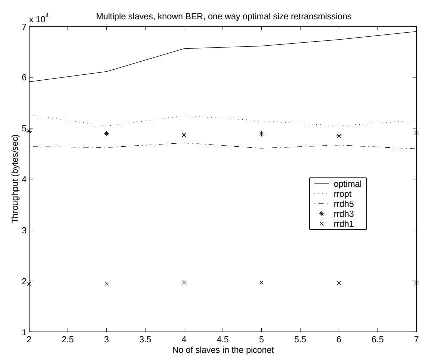


Figure 1. *Performance as a function of the number of slaves under ideal conditions*

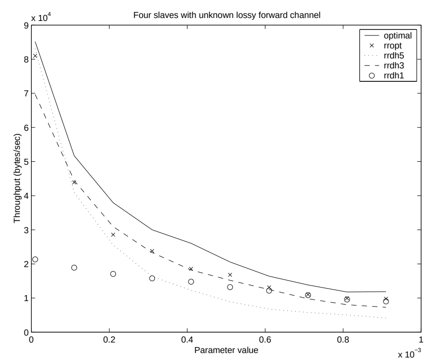


Figure 2. *Performance of a four slave piconet with BER relaxation*

In Figure 1 we show the behavior of the different strategies as a function of the number of slaves in the piconet. Note that in this figure, the number of slaves in the piconet is plotted on the x-axis unlike other figures. In this case we assume a rayleigh channel with a parameter of $2.1\text{x}10^{-4}$. The number of slaves in the piconet is varied from 2 to 7. The optimal algorithm has the best performance of all the strategies considered irrespective of the number of slaves in the piconet. An interesting observation though is that the performance of the optimal increases with an increase in the number of slaves. The other strategies including rropt do not show any improvement in system performance with an increase in the number of slaves. Thus, using the optimal strategy, with two slaves the maximum total throughput attained is $5.9\text{x}10^4$ bytes per second whereas the maximum total throughput attained with seven slaves is $6.89\text{x}10^4$ bytes per second; an increase of about 17 per cent. With more slaves, the optimal strategy has a greater choice of slaves which translates to better system performance.

We next consider relaxation 3, i.e., assume that the channel BER is not known but has to be estimated. The channel BER is estimated as explained earlier. In Figure 2 we show the throughput versus channel quality when the optimal algorithm uses channel estimation. We consider a scenario whereby the different slaves have slightly different channel conditions.

The ith slave has a channel given by Rayleigh distribution with a parameter $iP$ where $P$ is the parameter of the rayleigh distribution as given earlier and is plotted on the x-axis of the figure. We see that the optimal algorithm performs better under all regimes than the oblivious rrdhx strategies that send fixed-length packets. It is obvious that the optimal strategy gives up fairness in exchange for this gain. If channel-based scheduling is not used on account of fairness considerations, rropt achieves the best performance obtained by any fixed-length strategy for a given channel quality, and adapts smoothly to change in channel quality. Adding channel-based scheduling provides an additional increase in throughput across the board. We have considered more variations in simulations including relaxing the assumptions of loss free reverse channels and same size packet retransmissions. These results are not shown for lack of space.

## 4. Conclusion

Bluetooth has been the subject of much attention lately. Bluetooth uses frequency hopping and operates in the unlicensed 2.4GHz ISM band, which is also used by IEEE802.11 radios as well as other devices such as microwave ovens. Thus, it can be expected that the frequencies in this band will be subjected to interference from other sources in addition to being subjected to the vagaries of wireless links. In this paper we concentrate on providing solutions for enabling efficient communication between Bluetooth devices in the face of such interference. We consider a Bluetooth piconet operating in lossy channel conditions and provide algorithms to maximize the throughput in a piconet with one or more slaves by choosing the packet lengths to be transmitted at the different frequencies.

## REFERENCES


1. D. Bertsekas. *Nonlinear Optimization.* Athena Scientific, 1995.
2. P. Bhagwat, P. Bhattacharya, A. Krishna, and S.K. Tripathi. Enhancing throughput over wireless lans using channel state dependent packet scheduling. IEEE Infocom 1996.
3. Cahners In-Stat Group. Bluetooth overtakes 802.11x with 2001 shipments on track. *Report MM01-18BW, http://www.instat.com/abstracts/mm/2001/ mm0118bw_abs.htm*, Nov 2001.
4. Bray J. and C.F. Sturman. Bluetooth: Connect with cables. *Prentice Hall, NJ*, 2001.
5. O.E. Kelly and F. Babich. Protocol optimization using discrete channel statistic. IEEE ICUPC, Florence Italy, Oct 1998.
6. S. Lu, V. Bharghavan, and R. Srikant. Fair scheduling in wireless packet networks. *IEEE/ACM Transactions on Networking*, 7(4):473–489, Aug 1999.
7. B. Miller and C. Bisdikian. Bluetooth revealed. *Prentice Hall, NJ*, 2001.
8. E. Modiano. A simple algorithm for optimizing the packet size in arq protocols based on acknowledgement history. Proc. of 31st CISS, Baltimore USA, Mar 1997.
9. T. Eugene Ng, I. Stoica, and H. Zhang. Packet fair queueing algorithms for wireless networks with location-dependent errors. IEEE Infocom, 1998.
10. S. Saswati, F. Anjum, and R. Jain. *Throughput Optimal Packet Size Selection for Bluetooth Networks.* http://www.seas.upenn.edu/ swati/publication.htm, Aug 2002.
11. Bluetooth SIG. Bluetooth specification. http://www.bluetooth.com, 2000.
12. R. Wolff. *Stochastic Modeling and the Theory of Queues.* Prentice-Hall, 1989.