Throughput Optimal Packet Size Selection (TOPS) for Bluetooth Networks

Saswati Sarkar, Farooq Anjum and R. Jain
Dept of EE UPenn, Telcordia Tech, NTT DoCoMo

July 14, 2002

Abstract

Bluetooth is a low power, low cost short range wireless communication system operating in the 2.4GHz ISM band. 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 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. For such a system, we show that we can exert indirect control over the transmission conditions by choosing the packet size transmitted over each frequency as a function of the channel conditions. Our goal then is to provide a packet size selection algorithm that can maximize the throughput in a Bluetooth piconet in the presence of lossy wireless channels.

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 then 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 [10, 14, 20]. 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. Bluetooth has also been proposed for use in a scatternet mode, where an ad-hoc network of Bluetooth devices allow packets to be forwarded across multiple Bluetooth hops. Industry analysts have estimated that 13 million bluetooth devices were shipped in 2001, meeting earlier forecasts, and predict that worldwide by 2005 there will be over 780 million new Bluetooth devices shipped [9].

We focus on the use of Bluetooth as a cable replacement technology. 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. For concreteness in this paper we consider the situation where Bluetooth is used as a cable replacement for connectionless data (such as UDP packets), with high throughput rather than low packet error-rate being the goal.

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 [12, 15]. 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.

Our contributions in this paper are as follows. We first develop a mathematical model of packet transmission in a Bluetooth piconet with a single slave. We then use this model to show that the decision about the packet length to be transmitted for a given frequency is a threshold based process; this is given the objective of maximizing the system throughput. Given the three packet lengths that we consider, this means that there are two optimum values of the thresholds. For BER values below the lower threshold, the largest packet sizes must be transmitted and for BER values greater than the higher threshold the smallest packet sizes must be used. 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 taking the help of standard optimization techniques since it is not clear that these techniques are applicable given the functions involved. We then show that this strategy leads to significantly more throughput as compared to other baseline strategies. Our simulations indicate that over 50 percent improvements in throughput can be obtained over simple strategies. Subsequently we relax these assumptions and show through simulations for a single master-slave pair that the algorithm still works well, and significantly better than simple competing strategies. We then extend our results to multiple active slaves in a piconet.

The paper is organized as follows. In the following section we provide some background on Bluetooth as well as a brief survey of related work. In sec. 3 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. 4 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.

2 Bluetooth

As explained previously, Bluetooth is a low power, low cost short range wireless communication system operating in the 2.4GHz ISM band [20, 14, 10]. 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. We now summarize some details of Bluetooth link and MAC level communication that have bearing on the study in this paper.

Piconets: Every Bluetooth device has a unique device address, and functions either in the slave mode or in the master mode during communication. A set of communicating Bluetooth devices form a group called a piconet, which has (exactly) one device operating as a master and upto seven devices actively functioning as slaves at any given time. Each of the slaves has an address allocated by the master called the active member address. A master can communicate with any slave in its piconet. Slaves on the other hand can communicate directly only with the master.

Multiple Access: The devices in a piconet use frequency hopping to communicate. The unique Bluetooth device address of the master and its clock determine the frequency-hopping pattern used within a piconet. When slaves connect to a piconet, they obtain the device address and the clock of the master, and thus are synchronized to the master. Time is divided into equal-sized slots of length 625 microseconds, and TDMA is used to mediate access to the piconet amongst slaves. TDD is used between the master and each slave. More specifically, the master is responsible for determining which slave is going to transmit and when. The master always transmits in the even-numbered slots and a slave transmits in the odd-numbered slots; a slave can transmit only if the master has addressed it in the previous even numbered slot. In order to keep up the alternation between even and odd slots, packets must occupy an odd number of slots; hence each packet can span 1,3 or 5 slots and is transmitted at a single frequency.

After each packet is transmitted the devices retune their radio to a different frequency resulting in hopping from frequency to frequency. Channel hops are produced at a rate of one hop per slot except during multislot packets such as DH3 and DH5. There is no hop during a multislot packet, although the frequency sequence continues unhindered after the packet is transmitted. The sequence involves all 79 (or 23) hop frequencies spread equally.

Bluetooth links: Two types of links are allowed. Synchronous Connection Oriented (SCO) links support symmetrical, circuit-switched connections and are expected to be used for voice traffic. Asynchronous Connectionless (ACL) links are defined for bursty data transmissions. The master of the piconet controls the allocation of the ACL link bandwidth to each slave in the piconet. While ACL links can have packets with and without error correction mechanisms, we consider only ACL links with packets without error correction mechanisms. This is because error correction consumes part of the expensive bandwidth; in addition we would like to see the performance possible without using error correction mechanisms. (Note though, that the approach used in this paper is easily extensible to the scenario considering packets with error correction as explained in the next section). Hence, we consider three types of packets namely DH5, DH3 and DH1. As explained earlier, DH5 packets contain 2870 bits of user data and span 5 time slots; DH3 packets contain 1622 bits of user data and span 3 time slots; DH1 packets contain 366 bits of user data and can be transmitted in a single time slot.

We skip details regarding other topics such as the formation of a piconet, scatternets etc. Interested reader can refer to [10, 14, 20].

Much of the work on Bluetooth so far has focused on piconets [4, 5, 11, 8] though lately the focus is shifting towards scatternets [7, 6, 18, 17, 3, 23]. In [18, 23] the authors provide an asynchronous distributed protocol for constructing scatternets. This work is oriented towards setting up a fully connected ad-hoc network rapidly. The authors in [17] argue for extensive cross-layer optimizations rather than use the traditional multiple-layer approach when considering Bluetooth scatternets. In [3] the authors propose an innovative routing protocol for use over scatternets.

In [4] the authors propose mechanisms to mitigate interference between 802.11 and Bluetooth networks. For a Bluetooth piconet, these mechanisms are based on avoiding the frequencies used by the 802.11 network. No attempt is made at determining optimal baseband packet sizes at all. The concept of adaptive frequency hopping has been proposed in the IEEE 802.15

group [21]. This proposal advocates dynamically changing the frequency hopping sequence in order to avoid or minimize interference. The focus is only on providing primitives in the Bluetooth stack to make this happen without regards to the actual algorithms to be used. In addition this also requires modification to the existing Bluetooth devices. We are different in that we propose to use the existing frequency hopping sequence more efficiently. Further our algorithms can also be implemented in conjunction with [21] to provide a higher efficiency.

3 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. The model needs to be general enough to represent the various nuances of the problem, and at the same time simple enough to cater to a packet length decision process which can be readily implemented in a resource constrained wireless device. 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 piconets with multiple slaves.

3.1 Piconets with one slave

First we 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 still assumed to occur at the L2CAP layer.

Let the probability of successful transmission of each bit in a given packet be p (BSR*). 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

*BSR = 1 - BER, where BER is the bit error rate.

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 would also like to remark that the mathematical model that we have depends on this fact.

We assume that at the beginning of a slot, the master knows the BSR governing the 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 later incorporate a simple estimation procedure geared towards both stationary as well as slowly time varying channels for gradually learning the channel conditions and investigate the performance of the algorithm we propose in such a scenario. We would also like to remark that the assumption regarding knowledge of channel conditions has been made elsewhere as well[13, 16]. 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. We would like to remark that we can extend our model to cater to two way communications with a lossy reverse channel but do not do so in this paper for lack of space. 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[22]. 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 [22],

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

Thus the throughput η_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 exists 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 $F(p)$.*

Proof of theorem 1: We first prove that the optimum packet length $l^*(p)$ is a nondecreasing function of the success rate p and independent of F(p). Combining this along with the fact that only three packet lengths namely 366 bits, 1622 bits and 2870 bits are allowed[†], it follows that the optimum packet length decision process is a threshold based decision process.

The proof is by contradiction. We consider the discrete version of equation 1 for simplicity. Therefore, we consider a discrete random variable p. Let $n(p)$ denote the optimal packet length (in terms of bits) corresponding to a success rate p. Further, the duration of a packet is denoted as $\bar{n}(p)$. Then the optimum throughput, denoted by A, is given by

$$A = \frac{\sum_p n(p)p^{n(p)}dF(p)}{\sum_p \bar{n}(p)dF(p)} \tag{2}$$

Let the optimum packet length $n(p)$ be a strictly decreasing function of p. Hence, there exist p_1, p_2 such that $n(p_1) > n(p_2)$ while $p_1 < p_2$. Note that this implies that $\bar{n}(p_1) > \bar{n}(p_2)$ Let

$$\begin{aligned}
\alpha &= \sum_{p \neq p_1, p_2} n(p)p^{n(p)}dF(p) \\
\beta &= \sum_{p \neq p_1, p_2} \bar{n}(p)dF(p) \\
A &= \frac{n(p_1)p_1^{n(p_1)}dF(p_1) + n(p_2)p_2^{n(p_2)}dF(p_2) + \alpha}{\bar{n}(p_1)dF(p_1) + \bar{n}(p_2)dF(p_2) + \beta} \\
B &= \frac{n(p_1)p_1^{n(p_1)}dF(p_1) + n(p_1)p_2^{n(p_1)}dF(p_2) + \alpha}{\bar{n}(p_1)dF(p_1) + \bar{n}(p_1)dF(p_2) + \beta} \\
C &= \frac{n(p_2)p_1^{n(p_2)}dF(p_1) + n(p_2)p_2^{n(p_2)}dF(p_2) + \alpha}{\bar{n}(p_2)dF(p_1) + \bar{n}(p_2)dF(p_2) + \beta}
\end{aligned}$$

Here B, C are throughputs obtained with suboptimal decisions. The decision processes in terms of the packet length for B, C are the same as for A for all error rates other than p_1, p_2,; B is the throughput obtained if the packet length is $n(p_1)$ at both p_1, p_2, and C is the throughput obtained if the packet length is $n(p_2)$ at both p_1, p_2.

Since $n(p_1) > n(p_2)$, the denominator of A is larger than that of C. Since $A > C$, the numerator of A must also be larger than that of C. It follows that

$$\begin{aligned}
n(p_1)p_1^{n(p_1)} &> n(p_2)p_1^{n(p_2)} && (3) \\
p_1^{n(p_1)-n(p_2)} &> \frac{n(p_2)}{n(p_1)} && (4) \\
p_2^{n(p_1)-n(p_2)} &> p_1^{n(p_1)-n(p_2)} \text{ (since } p_2 > p_1, n(p_1) > n(p_2)) \\
&> \frac{n(p_2)}{n(p_1)} && (5)
\end{aligned}$$

Now using (4) and (5) we show that

$$n(p_1)p_2^{n(p_1)} - n(p_2)p_2^{n(p_2)} > n(p_1)p_1^{n(p_1)} - n(p_2)p_1^{n(p_2)} \tag{6}$$

[†]We ignore the medium rate DM packets

The argument is as follows:

$$\begin{aligned}
n(p_1)p_2^{n(p_1)} - n(p_2)p_2^{n(p_2)} &= n(p_1)p_2^{n(p_2)}\left(p_2^{n(p_1)-n(p_2)} - \frac{n(p_2)}{n(p_1)}\right)\\
&> n(p_1)p_1^{n(p_2)}\left(p_1^{n(p_1)-n(p_2)} - \frac{n(p_2)}{n(p_1)}\right)\\
\text{(using } p_2 > p_1 \text{ and equation (5))}&\\
&= n(p_1)p_1^{n(p_1)} - n(p_2)p_1^{n(p_2)}
\end{aligned}$$

$$\begin{aligned}
B &= \frac{n(p_1)p_1^{n(p_1)}dF(p_1) + n(p_2)p_2^{n(p_2)}dF(p_2) + \alpha + n(p_1)p_2^{n(p_1)}dF(p_2) - n(p_2)p_2^{n(p_2)}dF(p_2)}{\bar{n}(p_1)dF(p_1) + \bar{n}(p_2)dF(p_2) + \beta + (\bar{n}(p_1) - \bar{n}(p_2))\, dF(p_2)}\\
&= A\frac{1 + dF(p_2)\frac{n(p_1)p_2^{n(p_1)} - n(p_2)p_2^{n(p_2)}}{n(p_1)p_1^{n(p_1)}dF(p_1) + n(p_2)p_2^{n(p_2)}dF(p_2) + \alpha}}{1 + dF(p_2)\frac{(\bar{n}(p_1) - \bar{n}(p_2))}{\bar{n}(p_1)dF(p_1) + \bar{n}(p_2)dF(p_2) + \beta}}
\end{aligned}$$

$$\text{Since } B < A,\ \frac{n(p_1)p_2^{n(p_1)} - n(p_2)p_2^{n(p_2)}}{n(p_1)p_1^{n(p_1)}dF(p_1) + n(p_2)p_2^{n(p_2)}dF(p_2) + \alpha} < \frac{\bar{n}(p_1) - \bar{n}(p_2)}{\bar{n}(p_1)dF(p_1) + \bar{n}(p_2)dF(p_2) + \beta} \tag{7}$$

$$\begin{aligned}
C &= \frac{n(p_2)p_1^{n(p_2)}dF(p_1) + n(p_1)p_1^{n(p_1)}dF(p_1) - n(p_1)p_1^{n(p_1)}dF(p_1) + n(p_2)p_2^{n(p_2)}dF(p_2) + \alpha}{\bar{n}(p_1)dF(p_1) + \bar{n}(p_2)dF(p_2) + \beta - (\bar{n}(p_1) - \bar{n}(p_2))\, dF(p_1)}\\
&= A\frac{1 - dF(p_1)\frac{n(p_1)p_1^{n(p_1)} - n(p_2)p_1^{n(p_2)}}{n(p_1)p_1^{n(p_1)}dF(p_1) + n(p_2)p_2^{n(p_2)}dF(p_2) + \alpha}}{1 - dF(p_1)\frac{\bar{n}(p_1) - \bar{n}(p_2)}{\bar{n}(p_1)dF(p_1) + \bar{n}(p_2)dF(p_2) + \beta}}\\
&> A\frac{1 - dF(p_1)\frac{n(p_1)p_2^{n(p_1)} - n(p_2)p_2^{n(p_2)}}{n(p_1)p_1^{n(p_1)}dF(p_1) + n(p_2)p_2^{n(p_2)}dF(p_2) + \alpha}}{1 - dF(p_1)\frac{\bar{n}(p_1) - \bar{n}(p_2)}{\bar{n}(p_1)dF(p_1) + \bar{n}(p_2)dF(p_2) + \beta}} \text{ (from (6))}\\
&> A \text{ (from (7))}
\end{aligned}$$

This contradicts the fact that $A > C$. This implies that the optimal packet length is a nondecreasing function of the success rate 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 follows:

$$\begin{aligned} A &= \int_0^a 366x^{366} dF(x) + \int_a^b 1622x^{1622} \\ & \quad dF(x) + \int_b^1 2870x^{2870} dF(x) \\ B &= \int_0^a 625 dF(x) + \int_a^b 1875 dF(x) + \\ & \quad \int_b^1 3125 dF(x) \\ f(a,b) &= \frac{A}{B} \end{aligned}$$

From (1) and Theorem 1,

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

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 $0 \le a \le p_2$. 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.

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).

It is not obvious whether the iterates converge to the optimal thresholds, or if there is convergence at all. However, the following theorem shows that TOTS indeed converges to the optimal thresholds.

Theorem 2 *TOTS converges to the optimal thresholds.*

We prove theorem 2 in two steps. We first show that the iterates of the TOTS algorithm converge. This is done in the form of a lemma given next. We then show that the values that the iterates converge to are indeed the optimal values. Thus, we have

Lemma 1 $p_i^{k+1} \ge p_i^k$ *for* $i = 1, 2$ *and* $k \ge 0$.

Proof of Lemma 1: Let the inequality be violated at the $k+1$th iteration for the first time for some i. First consider the case $i = 1$, i.e., $p_1^{k+1} < p_1^k$ and $p_1^m \ge p_1^{m-1}$, $m \ge 1$

Let $f(p_1^k, p_2^k) = x/y$. Since $p_1^{k+1} \ne p_1^k$, we have $p_2^{k+1} = p_2^k$. Thus, $f(p_1^{k+1}, p_2^{k+1}) = \frac{x+\alpha}{y+\beta}$, where $\alpha = \int_{p_1^{k+1}}^{p_1^k} (1622x^{1622} - 366x^{366})\, dF(x)$ and $\beta = \int_{p_1^{k+1}}^{p_1^k} 1250 dF(x)$. Since $p_1^{k+1} < p_1^k$, $f(p_1^{k+1}, p_2^{k+1}) > f(p_1^k, p_2^k)$. Thus

$$
\begin{aligned}
\frac{x+\alpha}{y+\beta} &> \frac{x}{y} \\
\text{Hence, } \frac{x}{y} &< \frac{\alpha}{\beta} \qquad (8)
\end{aligned}
$$

Since $\beta > 0$, it follows from (8) that $\alpha > 0$.

Let p_1^k be reached in lth iteration for the very first time, i.e., $l = \arg\min_{j \ge 0}(p_1^j = p_1^k)$. Clearly $l \le k$. Also $l > 0$ as $p_1^0 = 0 \le p_1^{k+1} < p_1^k$. Thus $p_1^l = \arg\max_{0 \le p \le p_2^l} f(p, p_2^l)$. Let $f(p_1^l, p_2^l) = u/v$. Since $f(p_1^l, p_2^l) \le f(p_1^k, p_2^k)$, $u/v \le x/y < \alpha/\beta$. The last inequality follows from (8). It follows that $\frac{u+\alpha}{v+\beta} > \frac{u}{v}$. Thus $f(p_1^{k+1}, p_2^l) > f(p_1^l, p_2^l)$ and $0 \le p_1^{k+1} < p_1^k = p_1^l \le p_2^l$. This contradicts that fact that $p_1^l = \arg\max_{0 \le p \le p_2^l} f(p, p_2^l)$.

The case for $i = 2$ can be argued similarly. Let p_2^k be reached at l ($l \le k$). It can be shown that $l > 0$. It follows that $p_2^l = \arg\max_{p_1^{l-1} \le p \le 1} f(p_1^{l-1}, p)$. Note that $1 \ge p_2^{k+1} \ge p_1^k \ge p_1^{l-1}$. The first inequality follows since $p_2^{k+1} = \arg\max_{p_1^k \le p \le 1} f(p_1^k, p)$. It can also be shown that

$f(p_1^{l-1}, p_2^{k+1}) > f(p_1^{l-1}, p_2^l)$ which contradicts that fact that $p_2^l = \arg\max_{p_1^{l-1} \le p \le 1} f(p_1^{l-1}, p)$. The result follows. $\square$

It follows that $p_i^k \to p_i^c$ as $k \to \infty$. We next have to show that the iterates converge to the optimal values. Let (p_1^*, p_2^*) be the global maxima of the function $f(a, b)$. We will show that $p_i^c = p_i^*$, $i = 1, 2$. For simplicity we assume that the convergence happens in a finite number of steps. The entire proof can be extended to the case when this assumption is relaxed, by taking properties of the limit. We also assume that if $0 < p_2^* < 1$, then $p_1^* < p_2^*$. Note that if $p_2^* = 0$, then $p_1^* = 0$ and the optimum points are attained at start itself (once the algorithm reaches the optimum, the iterates do not change any longer). So, we will assume that $p_2^* > 0$.

We first make a few observation in the form of lemmas. We will make use of these lemmas later.

Lemma 2

$$\int_{min(p_1^*, p_1^c)}^{\theta} \left(1622x^{1622} - 366x^{366}\right) dF(x) \ge 0 \tag{9}$$

where $min(p_1^*, p_1^c) \le \theta \le max(p_1^*, p_1^c)$

Proof of Lemma 2: We first show that $(1622x^{1256} - 366) \ge 0$ at $x = p_1^*, p_1^c$ The observation for $x = p_1^*$ follows because the optimal threshold for change of packet length from 366 to 1622 is at p_1^*. Note that increase in packet length always increases the denominator of throughput $f(a, b)$. For optimality, the numerator must also increase, otherwise the throughput decreases by increase in packet length. The numerator increases if $1622x^{1256} - 366 \ge 0$. The observation for $x = p_1^c$ follows because otherwise again the throughput increases if the packet length is not increased to 1622 at p_1^c and thus subsequent iteration will increase the value of the first threshold, and the iterate for this will not converge at p_1^c.

Now we show that $(1622x^{1622} - 366x^{366})$ is nondecreasing in the range $\theta_1 \le x \le \theta$ where $\theta_1 = min(p_1^*, p_1^c)$. Note that $(1622x^{1622} - 366x^{366}) = x^{366}(1622x^{1256} - 366)$. Both the terms are nondecreasing for $x \ge 0$. The product is nondecreasing if both terms are nonnegative in the given range. More precisely we need to show that $(1622x^{1256} - 366) \ge 0$ for $\theta_1 \le x \le \theta$. Since $(1622x^{1256} - 366)$ is nondecreasing, it is sufficient to show that $(1622x^{1256} - 366) \ge 0$ at $x = \theta_1$. This follows from the observation made previously. $\square$

Lemma 3

$$\int_{\theta_3}^{\theta} \left(2870x^{2870} - 1622x^{1622}\right) dF(x) \tag{10}$$

where $\theta_3 = min(p_2^*, p_2^c) \le \theta \le max(p_2^*, p_2^c)$

The proof of this lemma is similar to the earlier proof and is hence skipped. Thus, we next prove that the iterates of TOTS converge to the optimal. We prove this in the form of a lemma. Note that the proof the next lemma along with lemma 1 provides the proof of theorem 2.

Lemma 4 $p_i^c = p_i^*$, $i = 1, 2$.

Proof of Theorem 4: The proof is by contradiction. We have, based on the fact that convergence happens in finite number of steps,

$$p_2^c = \arg\max_{p_1^c \leq p \leq 1} f(p_1^c, p) \quad (11)$$

$$p_1^c = \arg\max_{0 \leq p \leq p_2^c} f(p, p_2^c) \quad (12)$$

Let $f(p_1^c, p_2^c) = \frac{x}{y}$. We assume that $(p_1^c, p_2^c) \neq (p_1^*, p_2^*)$. Given that $f(p_1^*, p_2^*)$ is a global optimum, we have $f(p_1^c, p_2^c) < f(p_1^*, p_2^*)$.

We divide the proof into the following cases

- Case 1: $p_1^* < p_1^c < p_2^* < p_2^c$.
- Case 2: $p_1^* < p_2^* < p_1^c < p_2^c$.
- Case 3: $p_1^* < p_1^c < p_2^c < p_2^*$.
- Case 4: $p_1^c < p_1^* < p_2^* < p_2^c$.
- Case 5: $p_1^c < p_1^* < p_2^c < p_2^*$.
- Case 6: $p_1^c < p_2^c < p_1^* < p_2^*$.

While, we have shown only strict inequalities above, we will also consider appropriate equality conditions when looking at each case. Note that the above cases cover all the possible scenarios.

Case 1

First assume that $p_1^* < p_1^c \leq p_2^* < p_2^c$.

$$\alpha_1 = \int_{p_1^*}^{p_1^c} \left(1622x^{1622} - 366x^{366}\right) dF(x)$$

$$\beta_1 = \int_{p_1^*}^{p_1^c} 1250 dF(x)$$

$$\alpha_2 = \int_{p_2^*}^{p_2^c} \left(2870x^{2870} - 1622x^{1622}\right) dF(x)$$

$$\beta_2 = \int_{p_2^*}^{p_2^c} 1250 dF(x)$$

Note that $f(p_1^*, p_2^*) = \frac{x+\alpha_1+\alpha_2}{y+\beta_1+\beta_2}$. Since $f(p_1^c, p_2^c) < f(p_1^*, p_2^*)$, $\frac{x+\alpha_1+\alpha_2}{y+\beta_1+\beta_2} > \frac{x}{y}$. Note that $x \geq 0, y > 0$. Thus $\frac{\alpha_1+\alpha_2}{\beta_1+\beta_2} > \frac{x}{y}$. Since $\beta_1 > 0, \beta_2 > 0$, and $\frac{x+\alpha_1+\alpha_2}{y+\beta_1+\beta_2} > \frac{x}{y}$, $\max(\alpha_1, \alpha_2) > 0$. Thus if $\frac{\alpha_m}{\beta_m} = \max(\frac{\alpha_1}{\beta_1}, \frac{\alpha_2}{\beta_2})$, then $\alpha_m > 0$ and hence $\frac{\alpha_m}{\beta_m} > 0$. Note that $\frac{\alpha_m}{\beta_m} \geq \frac{\alpha_1+\alpha_2}{\beta_1+\beta_2}$. It follows that $\frac{\alpha_m}{\beta_m} > \frac{x}{y}$. It follows that $\frac{x+\alpha_m}{y+\beta_m} > \frac{x}{y}$. Wlog let $m = 1$. Then $f(p_1^*, p_2^c) > f(p_1^c, p_2^c)$. Also, $p_1^* < p_2^c$. These contradict (12). Similarly, if $m = 2$ then (11) is contradicted. Thus it is not possible that $p_1^* < p_1^c \leq p_2^* < p_2^c$. It can be shown similarly that $p_1^* \leq p_1^c \leq p_2^* < p_2^c$ and $p_1^* < p_1^c \leq p_2^* \leq p_2^c$ are not possible (assume $\alpha_1 = \beta_1 = 0$ for the first case and $\alpha_2 = \beta_2 = 0$ for the second case and show that $f(p_1^c, p_2^*) > f(p_1^c, p_2^c)$ respectively).

Case 2

Now we consider the case $p_1^* < p_2^* < p_1^c < p_2^c$.

$$\begin{aligned}
\alpha_1 &= \int_{p_1^*}^{p_2^*} \left(1622x^{1622} - 366x^{366}\right) dF(x) \\
\beta_1 &= \int_{p_1^*}^{p_2^*} 1250 dF(x) \\
\alpha_2 &= \int_{p_2^*}^{p_1^c} \left(1622x^{1622} - 366x^{366}\right) dF(x) \\
\beta_2 &= \int_{p_2^*}^{p_1^c} 1250 dF(x) \\
\alpha_3 &= \int_{p_2^*}^{p_1^c} \left(2870x^{2870} - 1622x^{1622}\right) dF(x) \\
\beta_3 &= \int_{p_2^*}^{p_1^c} 1250 dF(x) \\
\alpha_4 &= \int_{p_1^c}^{p_2^c} \left(2870x^{2870} - 1622x^{1622}\right) dF(x) \\
\beta_4 &= \int_{p_1^c}^{p_2^c} 1250 dF(x)
\end{aligned}$$

Note that $f(p_1^*, p_2^*) = \frac{x+\sum_{i=1}^{4} \alpha_i}{y+\sum_{i=1}^{4} \beta_i}$. Let $\frac{\alpha_m}{\beta_m} = \max_i \frac{\alpha_i}{\beta_i}$. We have $\alpha_i \geq 0$. This follows from Lemma 2 and 3. Further, it is also obvious that $\beta_i \geq 0$. Hence, $\frac{x+\alpha_m}{y+\beta_m} > \frac{x}{y}$.

We next show that $m \in \{2, 4\}$. Specifically, we will show that $\frac{\alpha_1}{\beta_1} \leq \frac{\alpha_2}{\beta_2}$ and $\frac{\alpha_3}{\beta_3} \leq \frac{\alpha_4}{\beta_4}$. From the proof of lemma 2, we know that $1622x^{1622} - 366x^{366}$ is non-decreasing in the range $p_1^* \leq x \leq p_2^*$. Hence, $\alpha_1 \leq \left(1622(p_2^*)^{1622} - 366(p_2^*)^{366}\right) \int_{p_1^*}^{p_2^*} dF(x)$. Thus $\frac{\alpha_1}{\beta_1} \leq \left(1622(p_2^*)^{1622} - 366(p_2^*)^{366}\right)/1250$. Further,it also follows from lemma 2 that $1622x^{1622} - 366x^{366}$ is non-decreasing in the range $p_2^* \leq x \leq p_1^c$. Therefore, $\frac{\alpha_2}{\beta_2} \geq \left(1622(p_2^*)^{1622} - 366(p_2^*)^{366}\right)/1250$. It thus follows that $\frac{\alpha_1}{\beta_1} \leq \frac{\alpha_2}{\beta_2}$. It can be similarly shown that $\frac{\alpha_3}{\beta_3} \leq \frac{\alpha_4}{\beta_4}$. It follows that $m \in \{2, 4\}$.

Consider, when $m = 2$. If $m = 2$ then $f(p_2^*, p_2^c) = \frac{x+\alpha_m}{y+\beta_m}$. In this case, we have $f(p_2^*, p_2^c) > f(p_1^c, p_2^c)$ and $p_2^* < p_2^c$. This contradicts (12). Now consider when $m = 4$ Then $f(p_1^c, p_1^c) = \frac{x+\alpha_m}{y+\beta_m}$. Therefore, $f(p_1^c, p_1^c) > f(p_1^c, p_2^c)$. This contradicts (11).

Now we consider the case where the inequalities are not strict. Note that $p_1^* < p_2^*$ by assumption that the optimal thresholds are unequal. First consider the case where $p_1^* < p_2^* = p_1^c = p_2^c$. This case violates (12) as $f(p_1^*, p_2^c) = f(p_1^*, p_2^*) > f(p_1^c, p_2^c)$ and $p_1^* < p_2^c$. Thus this is not possible.

Now consider $p_1^* < p_2^* = p_1^c < p_2^c$. This implies $\alpha_2 = \alpha_3 = \beta_2 = \beta_3 = 0$ In this case it is possible to show that either $f(p_1^*, p_2^c) > f(p_1^c, p_2^c)$ and $p_1^* < p_2^c$, or $f(p_1^c, p_1^c) > f(p_1^c, p_2^c)$. The first violates (12) and the second (11).

Now consider $p_1^* < p_2^* < p_1^c = p_2^c$. We then have $\alpha_4 = \beta_4 = 0$ Note that $f(p_1^*, p_2^*) = \frac{x+\sum_{i=1}^{3} \alpha_i}{y+\sum_{i=1}^{3} \beta_i}$. Let $\frac{\alpha_2}{\beta_2} = \max_{1 \leq i \leq 3} \frac{\alpha_i}{\beta_i}$. It is possible to show that $\frac{x+\alpha_2}{y+\beta_2} > \frac{x}{y}$. It follows that $f(p_2^*, p_2^c) > f(p_1^c, p_2^c)$ and $p_2^* < p_2^c$. This contradicts (12). Let $\frac{\alpha_2}{\beta_2} < \max_{1 \leq i \leq 3} \frac{\alpha_i}{\beta_i}$. Note that $\frac{\alpha_1}{\beta_1} \leq \frac{\alpha_2}{\beta_2}$. Thus $\frac{\alpha_3}{\beta_3} =$

$\max_{1\leq i\leq 3}\frac{\alpha_i}{\beta_i}$. Thus $\frac{\alpha_1}{\beta_1}\leq\frac{\alpha_2}{\beta_2}<\frac{\alpha_3}{\beta_3}$. If $\frac{\alpha_1}{\beta_1}>\frac{x}{y}$, then again $\frac{x+\alpha_2}{y+\beta_2}>\frac{x}{y}$. It follows that $f(p_2^*,p_2^c)>f(p_1^c,p_2^c)$ and $p_2^*<p_2^c$. This contradicts (12). Now let $\frac{\alpha_1}{\beta_1}\leq\frac{x}{y}$. Also, $\frac{\alpha_1}{\beta_1}=\min_{1\leq i\leq 3}\frac{\alpha_i}{\beta_i}$. It follows that $\frac{x+\sum_{i=2}^{3}\alpha_i}{y+\sum_{i=2}^{3}\beta_i}>\frac{x+\sum_{i=1}^{3}\alpha_i}{y+\sum_{i=1}^{3}\beta_i}$. Thus $f(p_2^*,p_2^*)>f(p_1^*,p_2^*)$. This contradicts the optimality of p_1^*,p_2^*.

Case 4

Now consider $p_1^c<p_1^*<p_2^*<p_2^c$.

$$\begin{aligned}
\alpha_1 &= \int_{p_2^*}^{p_2^c}\left(2870x^{2870}-1622x^{1622}\right)dF(x)\\
\beta_1 &= \int_{p_2^*}^{p_2^c}1250dF(x)\\
\alpha_2 &= \int_{p_1^c}^{p_1^*}\left(1622x^{1622}-366x^{366}\right)dF(x)\\
\beta_2 &= \int_{p_1^c}^{p_1^*}1250dF(x)
\end{aligned}$$

Note that $f(p_1^*,p_2^*)=\frac{x+\alpha_1-\alpha_2}{y+\beta_1-\beta_2}$. We also have $\beta_i,i=1,2\geq 0$ and $\alpha_i,i=1,2$. The latter follows from lemmas 2 and 3. Since $f(p_1^c,p_2^c)<f(p_1^*,p_2^*)$,

$$\frac{x+\alpha_1-\alpha_2}{y+\beta_1-\beta_2}>\frac{x}{y} \tag{13}$$

If $\frac{\alpha_1}{\beta_1}>\frac{x}{y}$ since $\alpha_1>0,\beta_1>0$, $\frac{x+\alpha_1}{y+\beta_1}>\frac{x}{y}$. Then $f(p_1^c,p_2^*)>f(p_1^c,p_2^c)$. This contradicts (11).

$$\text{Thus }\frac{\alpha_1}{\beta_1}\leq\frac{x}{y} \tag{14}$$

Also $\frac{x-\alpha_2}{y-\beta_2}\leq\frac{x}{y}$, else $f(p_1^*,p_2^c)>f(p_1^c,p_2^c)$, which contradicts (12). Thus, since $\beta_2\geq 0$, $\alpha_2\geq 0$

$$\text{We have }\frac{\alpha_2}{\beta_2}\geq\frac{x}{y} \tag{15}$$

$$\text{From (14),(15) }\frac{\alpha_1}{\beta_1}\leq\frac{\alpha_2}{\beta_2} \tag{16}$$

First assume that $\beta_1-\beta_2\geq 0$. From (13) $\alpha_1-\alpha_2\geq 0$ as well. From (13),

$$\frac{\alpha_1-\alpha_2}{\beta_1-\beta_2}>\frac{x}{y} \tag{17}$$

$$\begin{aligned}
\frac{\alpha_1}{\beta_1} &< \frac{\alpha_1-\alpha_2}{\beta_1-\beta_2}\text{ (from (14),(17))}\\
\text{Thus }\frac{\alpha_1}{\beta_1} &> \frac{\alpha_2}{\beta_2}
\end{aligned}$$

The above inequality contradicts (16).

Now assume that $\beta_1-\beta_2<0$. Let $\alpha_1-\alpha_2>0$. It follows that $\frac{\alpha_1}{\beta_1}>\frac{\alpha_2}{\beta_2}$. This contradicts (16). It follows that $\alpha_1-\alpha_2\leq 0$. From (13), $\frac{x-(\alpha_2-\alpha_1)}{y-(\beta_2-\beta_1)}>\frac{x}{y}$. Since $\beta_1-\beta_2<0$ and $\alpha_1-\alpha_2\leq 0$, it follows that $\frac{\alpha_2-\alpha_1}{\beta_2-\beta_1} < \frac{x}{y}$. Using (15), $\frac{\alpha_2-\alpha_1}{\beta_2-\beta_1} < \frac{\alpha_2}{\beta_2}$. It follows that $\frac{\alpha_1}{\beta_1} > \frac{\alpha_2}{\beta_2}$. This contradicts (16). Thus it is not possible that $p_1^c < p_1^* < p_2^* < p_2^c$. It is also easily shown that the cases $p_1^c \leq p_1^* < p_2^* < p_2^c$ and $p_1^c < p_1^* < p_2^* \leq p_2^c$ are also not possible.

Case 5

Now consider $p_1^c < p_1^* \leq p_2^c < p_2^*$.

$$
\begin{aligned}
\alpha_1 &= \int_{p_1^c}^{p_1^*} \left(1622x^{1622} - 366x^{366}\right) dF(x) \\
\beta_1 &= \int_{p_1^c}^{p_1^*} 1250 dF(x) \\
\alpha_2 &= \int_{p_2^c}^{p_2^*} \left(2870x^{2870} - 1622x^{1622}\right) dF(x) \\
\beta_2 &= \int_{p_2^*}^{p_2^c} 1250 dF(x)
\end{aligned}
$$

Note that $f(p_1^*, p_2^*) = \frac{x-\alpha_1-\alpha_2}{y-\beta_1-\beta_2}$. Since $f(p_1^c, p_2^c) < f(p_1^*, p_2^*)$, $\frac{x-\alpha_1-\alpha_2}{y-\beta_1-\beta_2} > \frac{x}{y}$. We know that $\beta_i \geq 0$, $i = 1, 2$. We also have $\alpha_i \geq 0$, $i = 1, 2$ from lemmas 2 and 3. It follows that $\frac{\alpha_2+\alpha_1}{\beta_2+\beta_1} < \frac{x}{y}$. Let $\frac{\alpha_1}{\beta_1} \leq \frac{\alpha_2}{\beta_2}$. Thus $\frac{\alpha_1}{\beta_1} < \frac{x}{y}$. Thus $\frac{x-\alpha_1}{y-\beta_1} > \frac{x}{y}$. Thus $f(p_1^*, p_2^c) > f(p_1^c, p_2^c)$. This contradicts (12). Let $\frac{\alpha_1}{\beta_1} > \frac{\alpha_2}{\beta_2}$. It can be similarly shown that $f(p_1^c, p_2^*) > f(p_1^c, p_2^c)$. This contradicts (11). Thus the case $p_1^c < p_1^* \leq p_2^c < p_2^*$ is not possible. Similarly it is possible to show that the cases $p_1^c \leq p_1^* \leq p_2^c < p_2^*$ and $p_1^c < p_1^* \leq p_2^c \leq p_2^*$ are also not possible.

Case 6

Now we consider the case $p_1^c < p_2^c < p_1^* < p_2^*$.

$$
\begin{aligned}
\alpha_1 &= \int_{p_1^c}^{p_2^c} \left(1622x^{1622} - 366x^{366}\right) dF(x) \\
\beta_1 &= \int_{p_1^c}^{p_2^c} 1250 dF(x) \\
\alpha_2 &= \int_{p_2^c}^{p_1^*} \left(1622x^{1622} - 366x^{366}\right) dF(x) \\
\beta_2 &= \int_{p_2^c}^{p_1^*} 1250 dF(x) \\
\alpha_3 &= \int_{p_2^c}^{p_1^*} \left(2870x^{2870} - 1622x^{1622}\right) dF(x) \\
\beta_3 &= \int_{p_2^c}^{p_1^*} 1250 dF(x) \\
\alpha_4 &= \int_{p_1^*}^{p_2^*} \left(2870x^{2870} - 1622x^{1622}\right) dF(x) \\
\beta_4 &= \int_{p_1^*}^{p_2^*} 1250 dF(x)
\end{aligned}
$$

Note that $f(p_1^*, p_2^*) = \frac{x-\sum_{i=1}^{4}\alpha_i}{y-\sum_{i=1}^{4}\beta_i}$. Let $\frac{\alpha_m}{\beta_m} = \min_i \frac{\alpha_i}{\beta_i}$. As earlier, from lemmas 2 and 3, we have $\alpha_i \geq 0$ for all i. Clearly $\beta_i \geq 0$ for all i. It is possible to show that $\frac{x-\alpha_m}{y-\beta_m} > \frac{x}{y}$. If $m = 1$ then

$f(p_2^c, p_2^c) = \frac{x-\alpha_m}{y-\beta_m}$. If $m = 3$ then $f(p_1^c, p_1^*) = \frac{x-\alpha_m}{y-\beta_m}$. In the first case $f(p_2^c, p_2^c) > f(p_1^c, p_2^c)$. This contradicts (12). In the second case $f(p_1^c, p_1^*) > f(p_1^c, p_2^c)$. This contradicts (11). Now we show that $m \in \{1, 3\}$.

We will show that $\frac{\alpha_1}{\beta_1} \leq \frac{\alpha_2}{\beta_2}$ and $\frac{\alpha_3}{\beta_3} \leq \frac{\alpha_4}{\beta_4}$. The proof is similar to that for the case $p_1^* < p_2^* < p_1^c < p_2^c$.

Now we consider the case where the inequalities are not strict. Note that $p_1^* < p_2^*$ by assumption that the optimal thresholds are unequal. First consider the case where $p_1^c = p_2^c = p_1^* < p_2^*$. This case violates (11) as $f(p_1^c, p_2^*) = f(p_1^*, p_2^*) > f(p_1^c, p_2^c)$ and $p_1^c < p_2^*$. Thus this is not possible.

Now consider $p_1^c < p_2^c = p_1^* < p_2^*$. In this case it is possible to show that either $f(p_1^c, p_2^*) > f(p_1^c, p_2^c)$ and $p_1^c < p_2^*$, or $f(p_2^c, p_2^c) > f(p_1^c, p_2^c)$. The first violates (11) and the second (12).

Now consider $p_1^c = p_2^c < p_1^* < p_2^*$. Note that $f(p_1^*, p_2^*) = \frac{x-\sum_{i=2}^4 \alpha_i}{y-\sum_{i=2}^4 \beta_i}$. Let $\frac{\alpha_3}{\beta_3} = \min_{2 \leq i \leq 4} \frac{\alpha_i}{\beta_i}$. It is possible to show that $\frac{x-\alpha_3}{y-\beta_3} > \frac{x}{y}$. It follows that $f(p_1^c, p_1^*) > f(p_1^c, p_2^c)$ and $p_1^c < p_1^*$. This contradicts (11). Let $\frac{\alpha_3}{\beta_3} > \min_{2 \leq i \leq 4} \frac{\alpha_i}{\beta_i}$. Note that $\frac{\alpha_4}{\beta_4} \geq \frac{\alpha_3}{\beta_3}$ (can be shown as in the case for $p_1^* < p_2^* < p_1^c < p_2^c$). Thus $\frac{\alpha_2}{\beta_2} = \min_{2 \leq i \leq 4} \frac{\alpha_i}{\beta_i}$. Thus $\frac{\alpha_4}{\beta_4} \geq \frac{\alpha_3}{\beta_3} > \frac{\alpha_2}{\beta_2}$. If $\frac{\alpha_4}{\beta_4} < \frac{x}{y}$, then again $\frac{x-\alpha_3}{y-\beta_3} > \frac{x}{y}$. It follows that $f(p_1^c, p_1^*) > f(p_1^c, p_2^c)$ and $p_1^c < p_1^*$. This contradicts (11). Now let $\frac{\alpha_4}{\beta_4} \geq \frac{x}{y}$. Also, $\frac{\alpha_4}{\beta_4} = \max_{2 \leq i \leq 4} \frac{\alpha_i}{\beta_i}$. It follows that $\frac{x-\sum_{i=2}^3 \alpha_i}{y-\sum_{i=2}^3 \beta_i} > \frac{x-\sum_{i=2}^4 \alpha_i}{y+\sum_{i=2}^4 \beta_i}$. Thus $f(p_1^*, p_1^*) > f(p_1^*, p_2^*)$. This contradicts the optimality of p_1^*, p_2^*.

Case 3

Finally the argument for the case $p_1^* < p_1^c \leq p_2^c < p_2^*$ (with and without strict inequalities) is similar to that for $p_1^c < p_1^* < p_2^* < p_2^c$ and is not repeated for brevity.

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 and hence we do not see any problem with this. In addition, we would also like to remark here that certain Bluetooth chipsets might not support DH3 or DH5 slots. The algorithm given above can be easily adapted to cater to such scenarios.

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 $\int_0^1 l(p) \left(\sum_{i=k}^{l(p)} \begin{pmatrix} l(p) \\ i \end{pmatrix} p^i (1-p)^{l(p)-i} \right) dF(p)$.

3.2 Multiple devices

We next consider a piconet with k slaves $1 < k \leq 7$. 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. This happens because of location dependent errors in wireless settings. For example, a particular slave may be in vicinity of a microwave which operates at the current frequency of transmission, while other slaves may be away from the microwave. 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 $F(p_1, p_2 \cdots, p_k)$. Note that the joint distribution allows for correlation between the BSRs perceived by different slaves. For example, there may be correlation between the channel conditions of slaves in proximity. Again if a microwave operates, then all slaves close to the microwave will experience poor channel conditions for the frequency of operation of the microwave. The joint distribution function can represent such scenarios. 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[22] as in the single slave case. Let a policy π 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, η_π is given by

$$\eta_\pi = \frac{\int_0^1 \cdots \int_0^1 l(p_1, \ldots p_k) p_{s(p_1,\ldots,p_k)}^{l(p_1,\ldots p_k)} dF(p_1, \ldots, p_k)}{\int_0^1 l(p_1, \ldots p_k) dF(p_1, \ldots, p_k)} \tag{18}$$

The objective is to design the optimal strategy π^* which maximizes the throughput η_{π^*}. 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. The strategy can be summarized as follows:

- Choose the slave with the highest BSR
- Determine the packet length as per the optimal strategy of the selected slave.

We can show similarly as in the case of a single slave 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.

4 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. We first describe the simulation model. A single piconet consisting of a master and different slaves is created. We do not consider the piconet formation phase in our simulations but assume that the master and the slaves of the piconet have knowledge of each other as well as a knowledge of the frequency hopping sequence of the piconet.

We consider that the master has infinite UDP traffic to send to the slave(s). Receipt of each packet is acknowledged by the slave in the form of NULL packets as provided for in the Bluetooth specification. 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 Rayleigh distribution can be described by a single parameter P as can be see from the probability density function given below; higher the value of this parameter, the worse is the channel going to be. The parameter of the Rayleigh distribution is plotted on the x-axis in all the figures shown.

$$f(x) = \frac{x}{P^2} e^{\frac{-x^2}{2P^2}} \tag{19}$$

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 following three.

- Assumption 1: The retransmissions can be of length DH1, DH3 or DH5 regardless of the length of the original packet (which is not convenient in practice.)
- Assumption 2: The reverse channel is lossless (i.e., ACKs are not lost)
- Assumption 3: The BER for a particular frequency is known.

We next discuss the effects of relaxing these assumptions and use simulations to evaluate the efficacy of the optimal algorithm with and without the relaxed assumtions.

4.1 Relaxations

Relaxation 1: Same size retransmissions. When a packet transmitted by the master is lost, the simplest action is for the master to send exactly the same sized packet as sent earlier. However, channel conditions may have changed, and under the optimal algorithm the master should perhaps send a different sized packet. In practice this complicates the implementation of the Bluetooth software stack at the master and may be undesirable. Further, in case the reverse link is also lossy, this will cause significant complications at both the master and the slave, as we will describe below. Thus our simulations consider the more realistic case where retransmissions are constrained to be the same size as the original packet.

Relaxation 2: Lossy reverse channels. Clearly the assumption of lossless reverse channels, is untenable in practice, and so our simulations consider the more realistic case where ACKs can be lost.

In addition, consider packet retransmissions because of the loss of an ACK. In this case the standard Bluetooth 1-bit ARQ scheme constrains packet retransmissions to be of the same length as the original packet because of the following reason. Suppose the master sends a DH1 packet based on current channel conditions and the ACK from the slave is lost. Now, when the master has to retransmit the packet, the channel conditions may have improved, and under the optimal algorithm it should send a DH5 packet. The master now has two choices. It can indicate to the slave that this packet is a retransmission (by not toggling the SEQN bit in the packet header) but in that case the slave will infer that the packet is a duplicate and discard it. Alternatively, it can indicate that this packet does contain new data and is not an exact retransmission (and hence toggle the SEQN bit). In that case the slave must somehow infer that the first part of the packet has already been received and the rest of it must be saved; this requires protocol as well as software modifications at the slave in order to be handled correctly.

Relaxation 3: Channel BER estimation. The assumption used in the analytical model that the BER is known for every frequency and to every slave is clearly impractical. However, it is possible to estimate the BER, and there has been a large body of work on channel estimation

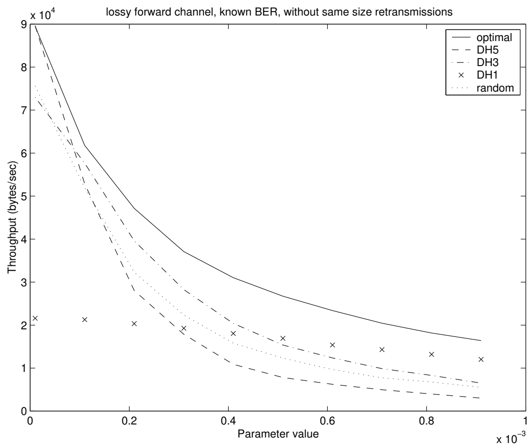

Figure 1: *Performance for a single slave piconet under ideal conditions*

techniques for cellular and other wireless systems, driven by research in smart antennas etc. Our goal in this paper is not to develop new lower-layer techniques for channel estimation, but to evaluate how well the optimal algorithm performs when channel estimation is used. To this end, we use a simple packet-level channel estimation method. We assume that the master knows the distribution governing the packet losses and needs to determine only the parameters of the distribution. The master does this by transmitting DH1 packets initially to the slave and then by observing the fraction of DH1 packet losses that take place when sending to a given slave for each given frequency. The fraction of packets lost on every frequency is then used by the master to determine the parameter of the distribution. In our experiments, the master uses information about DH1 packets transmitted for 3200 slots. The trials then start once the channel is estimated and the optimal thresholds calculated. Note that better estimation schemes can definitely be expected to improve the performance of the algorithms proposed as well as can be used to remove the assumption that the packet loss distribution is known. In addition, the estimation can be done on-line continuously. This should be done in case of slowly time varying channels. We do not though investigate this approach in this paper.

We next consider the performance of the optimal algorithm considering piconets with single slaves.

4.2 Single slave

We first describe the benchmark algorithms used for comparison with the optimal algorithm under ideal conditions as well as with the three relaxations given above. Note that these benchmark algorithms are oblivious to the channel conditions.

- **DHx**: The master only transmits DHx (x=1,3,5) packets to the slave.
- **Random**: At each opportunity to transmit, the master randomly sends a DH1, DH3 or DH5 packet.

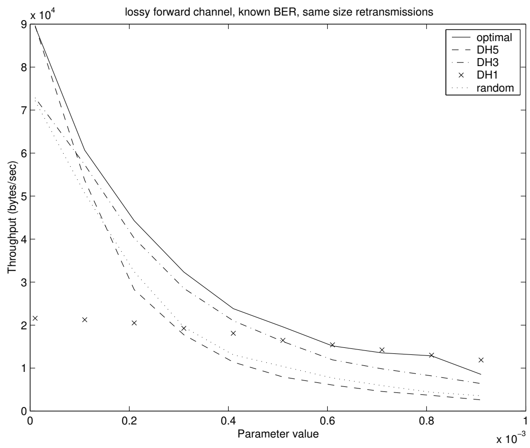

Figure 2: *Performance for a single slave piconet with relaxation 1*

We present the results of the average system throughput versus the channel error rate (as expressed by the Rayleigh distribution parameter). Each plot compares the optimal algorithm with the four oblivious strategies. We present, first, the results when ideal conditions hold (assumptions 1, 2 and 3); we subsequently present results as the relaxations are applied.

In Figure 1, we plot throughput versus channel error rate (with channel quality decreasing towards the right as explained earlier) when the ideal conditions hold. Note that the maximum possible throughput for one way communications between the master and the slave is 90514 bps in this case. As can be seen from the figure, the optimal policy consistently offers a significant improvement in throughput compared to all the oblivious strategies. Also note that different fixed-length policies (DH5, DH3, etc.) offer better throughput in different channel loss regimes. For example, DH5 packets offer the best throughput of the various oblivious strategies when channel conditions are good while DH1 packets offer the best throughput for very lossy channels. The optimal policy on the other hand offers the best throughput in all regimes by smoothly adapting to perform better than any fixed length policy at all times. Also, the random policy behaves slightly worse than DH5 for good channels, since it includes some shorter-length packets, and slightly better than DH5 for poor channels, for the same reason.

We next consider relaxation 1, where the retransmitted packets have to be of the same size as the original packets. The optimal policy has to be slightly modified here. The master uses the optimal packet lengths only when a new packet is to be transmitted; packet retransmission is done using the size of the original packet. (For the random policy the master also checks whether the packet to be sent is a retransmission and is modified similarly.) We show the performance for such a scenario in Figure 2. Note that this relaxation affects only the optimal and the random policies. As expected relaxation 1 will worsen the performance of the optimal strategy since the master is not able to use the optimal packet length in every slot. Most importantly, we see that the modified optimal policy still performs the best in all conditions except when the channel is very poor, at which point DH1 is slightly better.

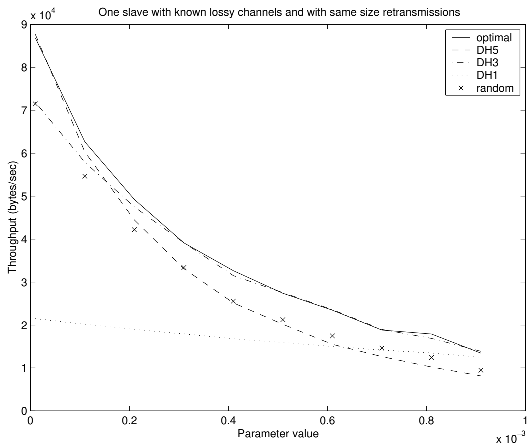

Figure 3: *Performance for a single slave piconet with relaxation 1 and 2*

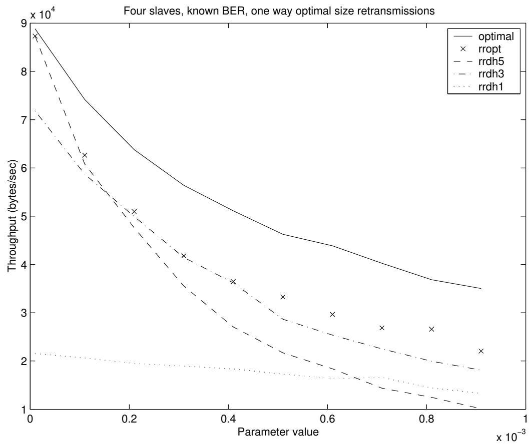

Figure 4: *Performance for a four-slave piconet under ideal conditions*

Figure 3 is for conditions identical to Figure 2 but with a lossy reverse channel. We see that the optimal algorithm is now only slightly better than the best oblivious strategy for any given channel quality. However, unlike the oblivious stategies it adapts smoothly to channel conditions thereby maintaining the best throughput in all channel condition regimes, and hence is preferable.

We have carried out simulation experiments for other combinations of the policies, and the results are qualitatively similar. We have also considered the effect of relaxation 3, where the master estimates the channel conditions; as expected, the performance of the optimal algorithm deteriorates further, although once again, unlike the oblivious strategies it is able to smoothly adapt to channel conditions and hence is preferable. We omit further simulation results to focus on the multiple slave scenarios, and show the effects of channel estimation for those scenarios.

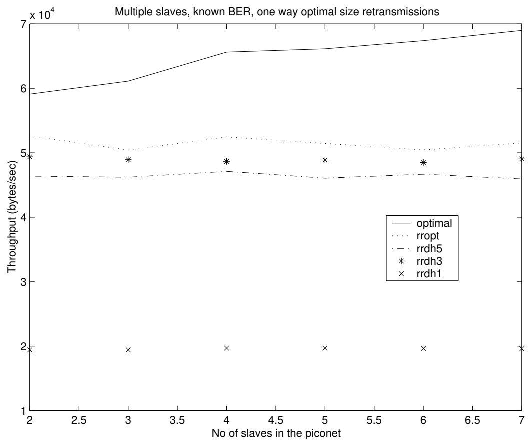

Figure 5: *Performance as a function of the number of slaves in the piconet under ideal conditions*

4.3 Multiple slaves

When there are multiple slaves in a piconet the master has an additional degree of freedom: deciding which slave it will transmit packets to at the next opportunity based on the BER to that slave. Thus the master can choose a scheduling strategy in addition to the packet length.

The oblivious algorithms, used as benchmarks, can be combined with scheduling policies in multiple ways. Here we will investigate the following algorithms as benchmarks. We do not consider the random policy further as its performance is significantly worse than the optimal.

- **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 but chooses the optimal packet length.

We first consider a piconet with 4 slaves, where the optimal algorithm is operating under ideal conditions (Figure 4). The rrdhx policies simply send fixed-length packets in a round robin fashion.

We see that in these conditions the optimal policy can have significantly better throughput over all channel regimes, and comparing it with rropt we see that a large part of its gain is due to intelligent scheduling based on channel BER. Both rropt and optimal adapt to channel conditions smoothly and outperform the oblivious strategies. Similar results are obtained with a different number of slaves in the piconet.

In Figure 5 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

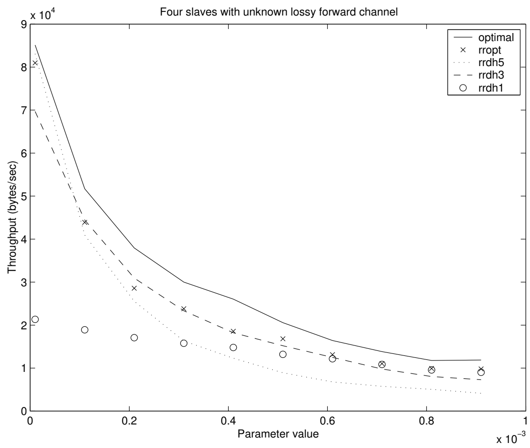

Figure 6: *Performance of a four slave piconet with relaxation 3 and different channel conditions for the various slaves*

the x-axis unlike other figures. In this case we assume a rayleigh channel with a parameter of 2.1x10^{-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 5.9x10^4 bytes per second whereas the maximum total throughput attained with seven slaves is 6.89x10^4 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 6 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. This can be understood by considering a slave which has a consistently bad channel as compared to the channels of the other slaves. The optimal strategy will never schedule the slave. 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.

Note that with multiple slaves, the additional degree of freedom due to scheduling means

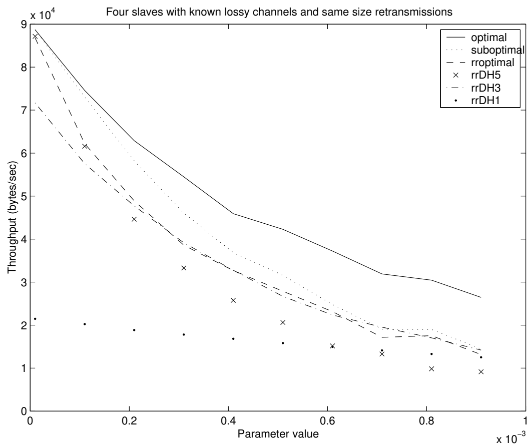

Figure 7: *Performance for a four-slave piconet with relaxation 1 and 2*

that Relaxation 1 has two interpretations. That is, when same size retransmissions are modeled, there are two choices for the optimal algorithm:

- **optimal:** If a packet is lost, at the next opportunity to send, the master selects the slave with lowest BER for transmission, and stores the original packet to be retransmitted until the original slave is selected again. Retransmitted packets always have the same size as the original packets.

- **suboptimal:** If a packet is lost, the master does not modify its scheduling decision, i.e., it retransmits to the same slave until successful. Further, the size of the retransmitted packet is the same as the size of the original packet.

In Figure 7 we consider the effect of relaxation 1 and relaxation 2 together. Thus although it is assumed that the channel BER is known, we assume that retransmissions to a given slave are of the same size as the original packet, and that the reverse channel is lossy.

We see that the 3 variants of the optimal algorithm (optimal, suboptimal, and rropt) perform better than the fixed-length strategies (except for rropt which falls slightly below rrdh3 for poor channels.) Further, as expected, optimal performs better than suboptimal.

Finally, in Figure 8 we consider a four-slave piconet with all three Relaxations applied. This is the most realistic of the simulated scenarios. We see that even for this case the optimal algorithm outperforms all other policies. However, at poor channel quality the fixed-length policy rrdh1 performs better than suboptimal as well as rropt. This is because at poor channel quality retransmissions occur more frequently in addition to the effects caused by estimation.

5 Conclusion

Bluetooth has been the subject of much attention lately. We focus on the use of Bluetooth as a cable replacement technology. Bluetooth uses frequency hopping and operates in the unlicensed

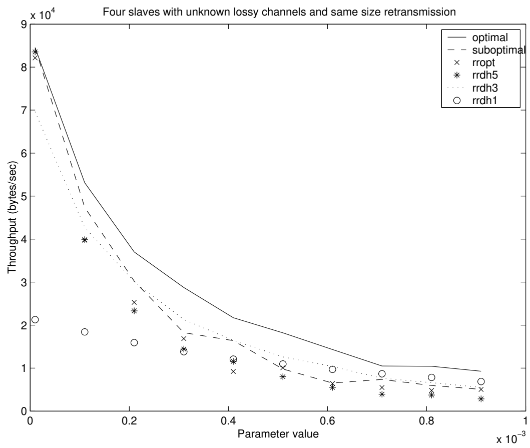

Figure 8: *Performance for a four slave piconet under realistic conditions*

2.4GHz ISM band, which is also used by IEEE 802.11 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. Thus, in this work, 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.

Our contributions in this paper are as follows. We first develop a mathematical model of packet transmission in a frequency hopping system like a Bluetooth piconet. We then use this model to show that given the objective of maximizing the throughput of the system, the decision regarding 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 as explained earlier. We then show that this strategy leads to significantly more throughput as compared to other baseline strategies. Our simulations indicate that over 50 percent improvements in throughput can be obtained over simple strategies. Subsequently we relax these assumptions and show through simulations for a single master-slave pair that the algorithm still works well, and significantly better than simple competing strategies. We then extend our results to multiple active slaves in a piconet.

Plans for future work include implementing these strategies. In addition, we also plan to consider the effects of the proposed algorithms considering traffic scenarios involving a mix of TCP and UDP traffic.

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] P. Bhagwat and A. Segal. A routing vector method for routing in bluetooth scatternets. 2000.

[4] C. Chiasserini and R. Rao. Co-existence mechanisms for interference mitigation between ieee 802.11 wlans and bluetooth. New York, June 2002. IEEE Infocom 2002.

[5] A. Das, A. Ghose, A. Razdan, H. Saran, and R. Shorey. Enhancing performance of asynchronous data traffic over bluetooth wireless ad hoc network. Infocom 01, Anchorage, Alaska, 2001.

[6] Baatz et al. Bluetooth scatternets: An enhanced adaptive scheduling scheme. New York, June 2002. IEEE Infocom.

[7] M. A. Marsan et al. Optimizing the topology of bluetooth wireless personal area networks. New York, June 2002. IEEE Infocom 2002.

[8] Zurbes et al. Radio network performance of bluetooth. New Orleans, June 2000. IEEE ICC 2000.

[9] 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.

[10] Bray J. and C.F. Sturman. Bluetooth: Connect with cables. *Prentice Hall, NJ*, 2001.

[11] M. Kalia, D. Bansal, and R. Shorey. Mac scheduling and sar policies for bluetooth: A master driven tdd pico-cellular wireless system. pages 384–388. IEEE MoMuc 99, 1999.

[12] O.E. Kelly and F. Babich. Protocol optimization using discrete channel statistic. IEEE ICUPC, Florence Italy, Oct 1998.

[13] S. Lu, V. Bharghavan, and R. Srikant. Fair scheduling in wireless packet networks. *IEEE/ACM Transactions on Networking*, 7(4):473–489, Aug 1999.

[14] B. Miller and C. Bisdikian. Bluetooth revealed. *Prentice Hall, NJ*, 2001.

[15] 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.

[16] T. Eugene Ng, I. Stoica, and H. Zhang. Packet fair queueing algorithms for wireless networks with location-dependent errors. IEEE Infocom, 1998.

[17] B. Raman, P. Bhagwat, and S. Seshan. Arguments for cross-layer optimizations in bluetooth scatternets. 2000.

[18] T. Salonidis, P. Bhagwat, L. Tassiulas, and R. LaMaire. Proximity awareness and ad hoc network establishment in bluetooth. Anchorage, Alaska, 2001. IEEE Infocom 01.

[19] 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.

[20] Bluetooth SIG. Bluetooth specification. http://www.bluetooth.com, 2000.

[21] B. Treister, K. C. Chen, and A. Batra. Adaptive frequency hopping. *IEEE P802.15-TG2-366r1, IEEE P802.15, Wireless Personal Area Networks*, April 2002.

[22] R. Wolff. *Stochastic Modeling and the Theory of Queues.* Prentice-Hall, 1989.

[23] G. Zaruba, I. Chlamtac, and S. Basagni. Bluetrees–scatternet formation to enable bluetooth based ad hoc networks. IEEE ICC, 2001.