Exact Probabilistic Analysis of the Limited Scheduling Algorithm for Symmetrical Bluetooth Piconets

Gil Zussman[1], Uri Yechiali[2], and Adrian Segall[1]

[1] Department of Electrical Engineering
Technion – Israel Institute of Technology, Haifa 32000, Israel
{gilz@tx, segall@ee}.technion.ac.il
http://www.comnet.technion.ac.il/{~gilz, segall}
[2] Department of Statistics and Operations Research
School of Mathematical Sciences
Tel Aviv University, Tel Aviv 69978, Israel
uriy@post.tau.ac.il
http://www.math.tau.ac.il/~uriy

Abstract. Efficient communication in Bluetooth scatternets requires design of intra and inter-piconet scheduling algorithms, and therefore numerous algorithms have been proposed. However, due to inherent complexities of the Bluetooth Medium Access Control (MAC), the performance of such algorithms has been analyzed mostly via simulation. We show that a piconet operated according to the limited (pure round robin) scheduling algorithm is equivalent to a *1-limited polling system* and utilize methods developed for the analysis of such systems to analyze this algorithm. We present *exact* analytic results regarding symmetrical piconets with bi-directional traffic. Our results differ from the corresponding analytical results presented as exact in [12],[13],[14],[16],[19], and [21]. We show that [14] actually presents *approximate* results, as it ignores important dependencies incorporated in the piconet operation model. Consequently, [14] *underestimates* the intra-piconet delay, in some cases by more than 50%. We also indicate that for similar reasons the analyses presented in [12],[13],[15],[16],[17],[18],[19],[20], and [21] seem to provide only approximate results.

Keywords: Bluetooth, Scheduling, Polling, Queueing, Limited, Time Division Duplex, Personal Area Network (PAN)

1 Introduction

Bluetooth is a Personal Area Network (PAN) technology, which enables portable devices to connect and communicate wirelessly via short-range ad-hoc networks [2],[3]. The basic network topology (referred to as a piconet) is a collection of slave devices operating together with one master. A multihop ad-hoc network of piconets in which some of the devices are present in more than one piconet is referred to as a scatternet (see for example Fig. 1). A device that is a member of more than one piconet (re-

[*] This research was supported by a grant from the Ministry of Science, Israel.

IFIP TC6 PWC 2003, LNCS 2775, pp. 276-290, Sep. 2003.

ferred to as a bridge) must schedule its presence in all the piconets in which it is a member (it cannot be present in more than one piconet simultaneously).

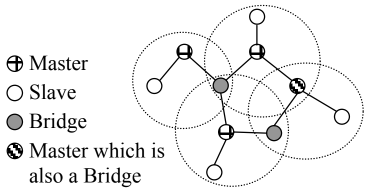


Fig. 1. An example of a Bluetooth scatternet

In the Bluetooth specifications [2], the capacity allocation by the master to each link in its piconet is left open. The master schedules the traffic within a *piconet* by means of polling and determines how the bandwidth capacity is to be distributed among the slaves. Efficient *scatternet* operation requires determining the link capacities that should be allocated in each piconet, such that the network performance will be optimized [27],[28]. The required link capacities should be allocated by *inter-piconet* scheduling algorithms. These algorithms schedule the presence of the bridges in different piconets. Numerous heuristic *intra* and *inter-piconet* scheduling algorithms have been proposed (e.g. [4],[5],[6],[7],[22] and references therein).

Analytical performance evaluation of intra and inter-piconet scheduling algorithms has great importance, since it may provide insight on their design and optimization. However, as mentioned in [5], due to the special characteristics of the Bluetooth Medium Access Control (MAC), the operation model of most scheduling regimes differs from those of classical polling models. Accordingly, in the past most of the proposed scheduling algorithms have been evaluated via simulation.

In this paper we focus on the *limited (pure round robin) scheduling algorithm*. We show that when all packets are 1-slot long[1], the piconet can be analyzed as a *TDMA* (Time Division Multiple Accesses) system. Then, we show that when all packet sizes are used the piconet is equivalent to a *1-limited polling* system[2]. The problem of computing exact mean delays in general 1-limited polling systems has not been resolved yet [8], but we derive exact analytic results for a *symmetrical* piconet with *bi-directional* traffic. We note that due to the equivalence to a polling system, approximate results can be obtained for more complex systems by utilizing the vast amount of research dedicated to 1-limited polling (see [8] for a review).

Recent papers [12],[13],[14],[16],[17],[18],[19],[20],[21] have *claimed to provide exact analytic results* regarding the performance of various intra and inter-piconet scheduling regimes. The analyses there are based on the theory of M/G/1 queue with vacations (introduced in [9], see also [25]). Since our exact results differ from these results, we show that the closed form solutions exhibited in [14] for the *limited* (pure round robin) scheduling algorithm are actually approximate solutions, as they are based on unsatisfied assumptions leading to inaccurate probability generating func-

[1] In Bluetooth piconets, the channel is slotted and the packets lengths are 1, 3, and 5 slots [2].

[2] A polling system consists of several queues served by a single server according to a set of rules (polling scheme) [1, p. 195],[8],[24],[26].

tions. In particular, we argue that important dependencies incorporated in the piconet operation model are ignored when applying the results of the M/G/1 queue with vacations to the analyzed system. We also indicate that the analyses in [12],[13],[15],[16],[17],[18],[19],[20], and [21] are based on similar models, and therefore seem to provide only approximate results. Finally, we present numerical results that demonstrate the difference between the results in [14] and our exact results.

We note that Miorandi et al. [10] have presented an *approximate* analysis of the limited scheduling algorithm for a piconet with *asymmetrical* traffic. Their findings support our observation that the analysis of the limited algorithm in [14] can serve only as an approximation.

Due to space constraints, we do not elaborate on piconets with *unidirectional traffic* and on the *exhaustive scheduling algorithm*. However, we note that in [29] we have shown that a piconet with *unidirectional* traffic operated according to the *exhaustive* scheduling algorithm can be modeled as an *exhaustive polling system* and derived exact analytic and numerical results regarding intra-piconet delays. It should be noted that those results also apply to piconets with asymmetrical traffic. We have also shown that a piconet with *unidirectional* traffic operated according to the *limited* scheduling algorithm can be modeled as a *1-limited polling system*. Then, we have shown that in a piconet with only uplink traffic in which all arrival rates are statistically equal, the mean delays for the *limited* and *exhaustive* regimes are *equal*. This observation has been extended for any arbitrary Time-Division-Duplex system, operated in a similar manner to a Bluetooth piconet, in which the packets are not necessarily 1, 3, and 5 slots long (as required by the Bluetooth specifications [2]). Finally, we have outlined the complexities in analyzing a piconet with *bi-directional* traffic operated according to the *exhaustive scheduling algorithm*.

To the best of our knowledge, the results presented in this paper and in [29] are the only available *correct exact* analytic results regarding the performance of Bluetooth scheduling algorithms.

The rest of the paper is organized as follows. Section 2 gives a brief introduction to the Bluetooth technology, while Section 3 presents the model. In Section 4 we analyze the limited scheduling algorithm and discuss its analysis in [12],[13],[14],[15],[16],[17],[18],[19],[20], and [21]. In Section 5 we present numerical results and in Section 6 we summarize the main results and discuss future research directions.

2 Bluetooth Technology

In a *piconet* one unit acts as a *master* and the others act as *slaves* (a master can have up to 7 slaves). Bluetooth channels use a Frequency-Hop/Time-Division-Duplex (FH/TDD) scheme in which the time is divided into 625-μsec intervals called *slots*. The master-to-slave transmission starts in even-numbered slots, while the slave-to-master transmission starts in odd-numbered slots. Masters and slaves are allowed to send 1, 3 or 5-slot *packets*, which are transmitted in consecutive slots. Packets can carry synchronous information (voice link) or asynchronous information (data link).[1]

[1] We note that we concentrate on networks in which only data links are used.

Information can only be exchanged between a master and a slave, i.e. there is no direct communication between slaves.

A slave is allowed to start transmission in a given slot if the master has addressed it in the preceding slot. The master addresses a slave by sending a data packet or a 1-slot *POLL packet* (if it has no data to transmit). The slave must respond by sending a data packet or a 1-slot *NULL packet* (if it has nothing to send). We refer to the master-to-slave communication as *downlink* and to the slave-to-master communication as *uplink*. An example of the TDD scheme in a piconet with n slaves is given in Fig. 2.

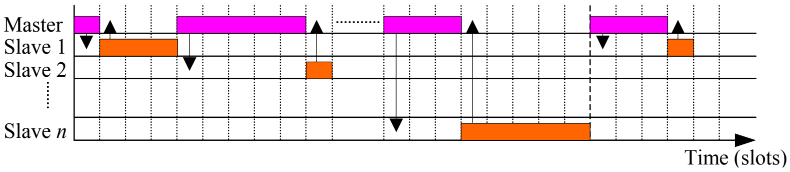


Fig. 2. An example of the Time Division Duplex (TDD) scheme in a Bluetooth piconet

The master schedules the traffic within a *piconet* according to an *intra-piconet* scheduling algorithm. Various intra-piconet scheduling algorithms have been recently proposed. For example:

- *Limited Round Robin* (*Pure Round Robin* [5]) – The master communicates with the slaves according to a fixed cyclic order. At most a single packet is sent in each direction (downlink and uplink) every time a master-slave queue pair is served.
- *Exhaustive Round Robin* – The master communicates with the slaves according to a fixed cyclic order. The master does not switch to the next master-slave queue pair until both the downlink (master-to-slave) and the uplink (slave-to-master) queues are empty.

In this paper, we focus on the limited algorithm.

In a *scatternet*, a unit (referred to as a *bridge*) can participate in two or more piconets, on a time-sharing basis, and even change its role when moving from one piconet to another. Namely, a bridge can be a slave of a few masters or a master in one piconet and a slave in another piconet. Fig. 1 above illustrates an example of a scatternet including bridges from these two types. The presence of a bridge in different piconets has to be controlled by an *inter-piconet* scheduling algorithm.

3 The Model

To facilitate our claims we describe the piconet model presented in [14] and use similar notation.

The number of nodes is denoted by m (accordingly, the number of slaves is $m-1$). We assume that each node has an infinite buffer. It is assumed that the traffic into each node is a compound Poisson process generating bursts (batches) of packets according to a Poisson arrival process with rate λ (bursts/slot). The probability generating function (PGF) of the burst (batch) size (number of packets in a burst) is denoted by $G_b(x)$. Its mean and second factorial moment are denoted by $\overline{B}$ and $\overline{B}^{(2)}$. We will show that the results presented in [14] are inaccurate even for the simplest case in which the traffic is non-bursty i.e. the burst size is always 1. To that end, in the rest of the paper we assume that $G_b(x) = x$.

The probabilities of a packet length being 1, 3, or 5 slots are p_1, p_3, and p_5, respectively. Accordingly, the PGF of the packet length is $G_p(x) = p_1x + p_3x^3 + p_5x^5$ and the mean is denoted by $\overline{L} = p_1 + 3p_3 + 5p_5$.

In [14], it is assumed that all packets within a burst have the same destination node. Furthermore, a burst generated at a given node is intended to one of the other $m - 1$ nodes with probability $1 / (m - 1)$. As a node, the master generates traffic intended for the slaves and in addition routes packets between the slaves. Under these assumptions, the burst arrival rate to each uplink (slave-to-master) queue is $\lambda_u = \lambda$ and the burst arrival rate to each downlink (master-to-slave) queue is $\lambda_d = \lambda$ (i.e. the model is symmetrical). Notice that the arrival process to the uplink queues is Poisson whereas the arrival process to the downlink queues, being dependent on the scheduling regime, is, in general, not Poisson.

Simplifying the above, we assume that the master is the destination of all packets generated at the slaves (i.e. the master *does not* route packets between slaves). On the other hand, we assume that packets are generated at every downlink queue according to a Poisson arrival process with rate λ. Due to the assumption regarding the Poisson arrival process, the analysis of this scenario is simpler than the analysis of the scenario described above in which the master *does* route packets. Therefore, the results regarding the access delay obtained in [14] should also hold for this scenario. However, we show that this is not the case.

Finally, three related performance indicators are defined:

- *Access delay* – The time a packet has to wait in the uplink queue before it is served (denoted by W_a).
- *Queueing delay at the master* – The time a packet has to wait at a downlink queue before it is served (denoted by W_m).
- *End to end delay* – The total time a packet spends in the master's and the slave's queues not including the service times (denoted by $W_e = W_a + W_m$).

The mean values of the delay functions described above are denoted by $\overline{W}_a$, $\overline{W}_m$ and $\overline{W}_e$.

4 Analysis of the Limited (Pure Round Robin) Regime

In this section we show that a piconet operating according to the limited regime and in which all packets are 1 slot long is equivalent to a TDMA system. Then, we consider a "standard" piconet with packet sizes of 1, 3, and 5 slots, and formulate it as a 1-limited *polling* system. We obtain exact analytical results which are *different* from those derived in [14] and indicate that [14] applies the results of the M/G/1 queue with vacations *without* considering the dependencies between the queues.

4.1 Analysis as a TDMA system

Consider a piconet operated in the limited regime in which *all packets are 1 slot long* (i.e. $p_1 = 1$). In such a piconet, a single slot is allocated to each downlink and uplink in every cycle. Therefore, the piconet can be analyzed as a *TDMA system* [1, p. 194] with a cycle length of $2(m-1)$ slots. Every slot in the cycle is allocated to one of the $2(m-1)$ downlinks and uplinks. The computation of the delay in a TDMA system is based on the analogy with the M/D/1 queue with deterministic service time and vacation length both equal to $2(m-1)$ [1, p. 194]. Accordingly, applying [1] eq. (3.58), where the number of queues is $2(m-1)$ and the total arrival rate is $2(m-1)\lambda$, we obtain the mean access delay and queueing delay at the master (in slots):

$$\overline{W}_a = \overline{W}_m = \frac{m-1}{1-2(m-1)\lambda}. \tag{1}$$

For this simple scheduling regime, the result regarding the access delay obtained in eq. (5) in [14] coincides with our result (1).

The model presented in Section 3 is symmetrical with respect to the slaves (the arrival rates to all master and slaves queues are equal). However, in a TDMA system, the queuing behavior of one user is independent of the queuing behavior of other users. Thus, analytic results can be obtained for an asymmetrical piconet. We denote the arrival rate to slave i by λ_u^i and the arrival rate to the master of packets intended for slave i by λ_d^i. We denote the access delay of packets in the uplink queue of slave i by W_a^i and the queueing delay at the master of packets intended to slave i by W_m^i. In this (deterministic) case every link can be independently analyzed as an M/D/1 queue. Thus, the mean access delay (in slots) is again derived from eq. (3.58) in [1]:

$$\overline{W}_a^i = \left(\frac{2(m-1)\lambda_u^i}{2\left(1-2(m-1)\lambda_u^i\right)}\right)2(m-1) + \frac{1}{2}\cdot 2(m-1) = \frac{m-1}{1-2(m-1)\lambda_u^i}. \tag{2}$$

A similar equation describes the mean queueing delay at the master of packets intended to slave i ($\overline{W}_m^i$). In this case $\overline{W}_m^i$ replaces $\overline{W}_a^i$ and λ_d^i replaces λ_u^i.

4.2 Analysis as a 1-limited Polling system

Since in most Bluetooth applications the length of the packets varies, we now consider a "standard" piconet operated in the limited regime with *1,3, and 5-slot packets*. We show that such a piconet can be modeled as a *1-limited polling system*[1] with $2(m-1)$ queues and present a closed form expression for the mean access delay in a piconet with symmetrical traffic.

In a piconet operated according to the limited scheduling regime, even if the master has nothing to send to a specific slave, one slot is used during the downlink communication (by the POLL packet). Similarly, even if the slave has nothing to send,

[1] In a 1-limited polling system, at each visit of the server to a queue only the *first* packet in the queue is served. The server incurs a *switchover* time when it shifts from one queue to another [1, p. 201],[8],[24].

one slot must be used during the uplink communication (by the NULL packet). In order to model the piconet as a 1-limited *polling* system we utilize the fact that data packets are at least one slot long. Thus, when data packets are sent at least one slot must be used.

We define an equivalent *1-limited gated* polling[1] system with the following characteristics: (a) There are $2(m - 1)$ customers and a single server, (b) the server serves the customers in a fixed cyclic order, (c) the server incurs a *switchover* time of 1 slot when it shifts from one customer to another, (d) when the server serves a customer, at most a single packet is served. If at the *beginning* of the switchover the queue is empty, the server completes the switchover and immediately switches to the next customer, and (e) the packet service times are 0, 2, and 4 slots.

This polling system is equivalent to a Bluetooth piconet operated according to the limited scheduling algorithm. Namely, for the same arrival process and packet length distribution (i.e. $p_0 = p_1$, $p_2 = p_3$, and $p_4 = p_5$), the delay (time until the *end* of service) in both systems is equal. This property is demonstrated in Fig. 3, which illustrates an example of the operation of a piconet composed of a master and two slaves and of the equivalent polling system. When the master starts transmitting to the first slave it has a 3-slot data packet in the downlink queue. In the equivalent system, this packet is represented by a 1 slot of switchover and 2 slots of data. The first slave has nothing to send, and therefore it responds with a NULL packet. A 1 slot of switchover represents this packet in the equivalent system. Then, the master sends a 1-slot data packet to the second slave. It is represented in the equivalent system by a 1 slot of switchover and *0 slots of data*. The rest of the transmissions (including a POLL packet) can be seen in the figure.

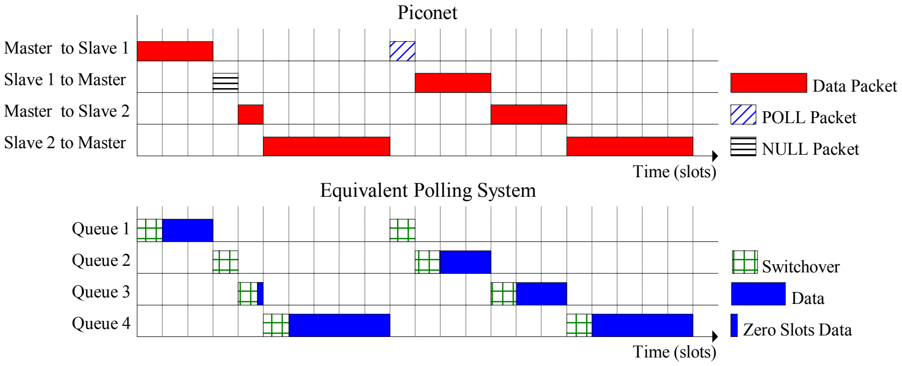


Fig. 3. An example of the operation of a piconet and of the equivalent polling system

In order to obtain the access delay (W_a) in a piconet, one has to deduct the Bluetooth packet length (L) from the delay (time until the *end* of service) in the equivalent polling system. Alternatively, if one obtains the waiting time (the time until the service *starts*) in the equivalent polling system, a single slot has to be deducted in order to obtain the access delay in a Bluetooth piconet. This results from the fact that when

[1] The system is referred to as the *limited gated* polling system, since only a message that is found in the *beginning* of the switchover time is served.

Bluetooth data packets are sent, some of the data is actually sent during the "switch-over" time, as it is defined in the equivalent polling system.

We now focus on symmetrical systems in which the arrival rates to all queues are equal. By applying the model for a symmetrical *limited gated* polling system described in [1, p. 201] we can obtain the mean waiting time of a packet in a queue. The service time of a k-slot (k = 1,3,5) data packet is defined as k – 1 slots. Moreover, the waiting time in [1] is defined as the time a packet waits until its service starts. Thus, in order to obtain the mean access delay, one has to deduct 1 slot from the expression for the waiting time in [1], eq. (3.77). Accordingly, we apply [1] eq. (3.77), where the number of queues is $2(m-1)$, the total arrival rate is $2(m-1)\lambda$, the switchover time is one slot with zero variance, the traffic intensity is $\rho = 2(m-1)\lambda(\overline{L}-1)$, and the second moment of the service time (denoted in [1] as $\overline{X^2}$) is $4p_3 + 16p_5$. Deducting 1 time unit (i.e. 1 slot), we obtain the mean access delay and the mean queueing delay at the master (in slots):

$$\overline{W}_a = \overline{W}_m = \frac{1+(m-1)\{1+2\lambda(p_3+6p_5-1)\}}{1-2(m-1)\lambda\overline{L}} - 1\,. \tag{3}$$

Notice that in this system it must hold that $2(m-1)\lambda\overline{L} < 1$. We shall refer to $2(m-1)\lambda\overline{L}$ as the load in the system.

As a special case, consider a piconet in which only 1-slot packets are used (i.e. $p_1 = 1$, $p_3 = 0$, and $p_5 = 0$). For such a piconet, eq. (3) reduces to eq. (1), which represents the delay in a piconet with 1-slot packets. Moreover, the result given by (3) was verified by two independent simulation models based on OPNET (for more details regarding the design of the simulation models, see [6][1] and [10]). For example, Fig. 4 compares the exact mean access delay ($\overline{W}_a$) (computed according to (3)) to the average access delay computed by simulation[2] in a piconet with 4 slaves in which the probabilities of 1, 3, and 5-slot packets are equal. For each load value, the results have been computed after 230,000 slots using the model presented in [6] or after 48,000 to 2,400,000 slots (depending on the load) using the model presented in [10].

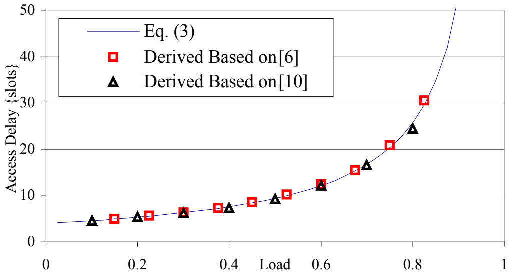


Fig. 4. The mean access delay (computed according to (3)) and the average access delay values computed by simulation in a piconet with 4 slaves in which $p_1 = p_3 = p_5 = 1/3$

[1] In [6] the delay is defined as the time until the whole packet is received by the destination.

[2] The simulation results computed by the model presented in [10] have been obtained by Carlo Caimi from the University of Padova.

Finally, we note that the equivalence between a piconet operated according to limited algorithm and a 1-limited polling system can be exploited in order to obtain exact and approximate results for more complex systems. Namely, using the equivalence property and methods for analyzing 1-limited polling systems (see for example [8]), one may analyze piconets with bursty arrival process ($G_b(x) \neq x$), asymmetrical traffic, and intra-piconet routing. For example, Miorandi and Zanella [11] have recently used this property in order to analyze piconets with asymmetrical traffic and to obtain approximate results which outperform the results in [10] and [14].

4.3 Examination of the Analysis as an M/G/1 Queue with Vacations [14]

The theory of M/G/1 queue with vacations[1] [9],[25] is used in [14] in order to analyze a piconet operated in the limited regime in which the packets are 1, 3, and 5 slots long. We now briefly describe that analysis and point out that the direct use of the model of M/G/1 queue with vacations to analyze a 1-limited polling system, *without taking into consideration the dependencies between the queues*, leads to inaccurate results and may serve only as an approximation. We assume, for simplicity, that the traffic is non-bursty ($G_b(x) = x$) and that packets are generated at every master-to-slave (downlink) and slave-to-master (uplink) queue according to a *Poisson* arrival process with arrival rate λ.

The *piconet service cycle time* X_c is defined in [14] as the time (number of slots) required for the master to serve all the slaves once. The PGF, the mean, and the second moment of the cycle time are denoted by $G_{X_c}(x)$, $\overline{X}_c$, and $\overline{X_c^2}$. It is claimed that since the model is symmetric, it is sufficient to consider a *single master-slave channel* (accordingly, we refer to the considered slave as slave *i*). It is mentioned that the probability that a downlink queue is not empty is $P_d = \lambda \overline{X}_c$ and that the probability that an uplink queue is not empty is $P_u = \lambda \overline{X}_c$. In [14], eq. (1) and (2), the PGFs of the durations of the downlink and uplink communications are presented as:

$$G_d(x) = \left(P_d p_1 + (1-P_d)\right)x + P_d p_3 x^3 + P_d p_5 x^5 \,, \tag{4}$$

$$G_u(x) = \left(P_u p_1 + (1-P_u)\right)x + P_u p_3 x^3 + P_u p_5 x^5 \,, \tag{5}$$

while the PGF of the cycle time is presented as:

$$G_{X_c}(x) = \left(G_d(x) G_u(x)\right)^{m-1} . \tag{6}$$

The model of M/G/1 queue with vacations is used to analyze the system. Accordingly, the service time of a single uplink queue (i.e. of the queue at slave *i*) in the vacation model is equal to the cycle time (X_c). A vacation starts when the master polls slave *i* and finds its uplink queue to be empty. As we understand, the vacation is composed of the NULL packet returned by slave *i*, the service of the other $m-2$ up-

[1] According to the model of M/G/1 queue with multiple vacations, the server begins a vacation every time the system becomes empty. If the server returns from a vacation to find the system not empty, it starts working immediately and continues until the system becomes empty again. If the server returns from a vacation to find an empty system, it begins another vacation immediately [25].

link and downlink queues, and the transmission of a packet to slave *i* in the downlink queue. V_l denotes the duration of the vacation period and its PGF, mean, and second moment are denoted by $G_{V_l}(x)$, $\overline{V}_l$, and $\overline{V_l^2}$. The PGF of the duration of the vacation period is provided in [14], eq. (3):

$$G_{V_l}(x) = xG_d(x)\left(G_d(x)G_u(x)\right)^{m-2}. \tag{7}$$

Finally, the access delay is derived from the waiting time in a batch arrival system $M^{[x]}/G/1$ with multiple vacations ([25, p. 143], eq. (3.21a)). Since we assume that the arrival process is non-bursty (i.e. $G_b(x) = x,\ \overline{B} = 1,\ \overline{B}^{(2)} = 0$), eq. (5) in [14] reduces to [25, p. 123], eq. (2.14a):

$$\overline{W}_a = \frac{\lambda \overline{X_c^2}}{2\left(1 - \lambda \overline{X}_c\right)} + \frac{\overline{V_l^2}}{2\overline{V}_l}. \tag{8}$$

We now describe a few problems in the model presented above. For the clarity of presentation, the description of each problem ignores the existence of other problems.

1. *The cycle length described in eq. (6) is inappropriate for use as a service time in a vacation model.* Consider the instant when the server returns to slave *i* from a vacation or when it completes “service” (i.e. the master completes a cycle) there, and at least one packet is found in the uplink queue. In such a case, the server will not take a vacation and a service period will start. According to [14], the PGF of the service time is defined as the PGF of the cycle time (X_c) presented in (6). It is composed of the duration of the communication in the uplink queue of slave *i* and the durations of the communication in the rest of the uplink and downlink queues. When the service starts, there is obviously a data packet in the queue of slave *i*. Hence, the PGF of the duration of the uplink communication of slave *i* is $G_p(x)$ (defined in Section 3) and not $G_u(x)$ (presented in eq. (5)). Therefore, one of the terms in eq. (6) should be replaced. Consequently, eq. (6) cannot be used as a service time in a vacation model[1].
2. *The uplink and downlink communication periods composing a cycle are not independent.* The PGF of the cycle length presented in (6) is a multiple of the PGFs of the uplink ($G_u(x)$) and the downlink ($G_d(x)$) communication periods. Thus, an underlying assumption in the derivation of $G_{X_c}(x)$ is that all the downlink and uplink communication periods are independent. However, the length of a communication period depends on the existence of a packet in the corresponding queue (if the queue is empty, the length is 1 slot). The existence of a packet depends on the lengths of the preceding uplink and downlink communication periods, since long periods increase the probability of a packet arrival. Hence, the lengths of the periods composing a cycle do not seem to be independent, and therefore the derivation of the cycle length in (6), where independence is assumed, is inaccurate[2].
3. *The cycle time depends on the length of the vacation or the cycle that precedes it.* The vacation model described by Takagi [25] is used in order to compute the mean access delay, presented in eq. (8). One of the important assumptions made in

[1] This difference is critical for small piconets where only a few components compose $G_{X_c}(x)$.

[2] It seems that the mean cycle length derived from combining (4), (5), and (6) is correct. However, for the vacation model the second moment of the cycle length is also required.

[25, p. 111] is that: "*Messages arrive in the system according to a Poisson process of fixed rate and have service times with independent and identical distribution. These service times are independent of the arrival process, and each service time is independent of the sequence of vacation periods that precede that service time.*" However, the service time in the model described in [14] is taken as the cycle time (X_c) and is composed of uplink and downlink communication in $m-1$ queues. The length of each of the communication periods depends on the existence of packets in the corresponding queue. If the cycle follows a long vacation or cycle, the probabilities that the queues are not empty will increase, and thereby the probability of a long cycle will increase. On the other hand, if the cycle follows a short vacation or cycle, the probability of a short cycle will increase. Thus, the requirement for independent service times, and for independence between vacation lengths and service times, does not seem to hold. Therefore, eq. (8) can provide only an approximation for the delay.

In Section 4.1, we have analyzed a piconet operated in the limited regime in which *all packets are 1 slot long*. In such a piconet the duration of a communication period is deterministic (1 slot), and therefore, there are no dependencies between the cycle and vacation lengths as well as within a cycle. Thus, as we have mentioned, for this simple regime, the result regarding the access delay obtained in eq. (5) in [14] coincides with our result (1). However, due to problems 1, 2, and 3, described above, in a "standard" piconet (in which the packets are 1, 3, and 5 slot long) the results presented in [14] differ considerably from our results.

Recall, that we assume that the master is the destination of all packets generated at the slaves (i.e. the master *does not* route packets). On the other hand, in [14] it is assumed that the master routes some of the traffic between the slaves. Accordingly, the mean value of the *end-to-end delay* is defined as the sum of the access delay and the queueing delay at the master. A method for computing the mean queueing delay at the master is described at the end of Section 2 in [14]. For non-bursty traffic, the delay computed according to that method is equal to the access delay described in (8) (i.e. $\overline{W}_a = \overline{W}_m$). Thus, it seems that an underlying assumption is that the arrival process of packets to the master from the slaves can be treated as Poisson.

This assumption probably follows the analysis of *polling systems with probabilistic routing* (e.g. [23]). The analysis of such systems is based on an important assumption that the *service times of a packet in different queues are independent*. However, this is not the case in a piconet (for example, a 1-slot packet sent from the slave to the master cannot become a 3-slot packet when it is forwarded to another slave). Thus, the computation of the end-to-end delay is inaccurate not only because of the inaccuracies in the computation of the access delay but also due to the assumptions made regarding the arrival process of packets which require routing.

We note that Miorandi et al. [10] present an *approximate* analysis of the limited scheduling algorithm by using the tool of probabilistic routing and assuming that the various resulting flows are independent. They state that "*the assumption of independent flows, although providing good results at low traffic load, leads to substantial mismatch with the simulation results as the system gets close to stability limit*".

Finally, we note that assumptions, similar to the ones indicated in this section, also appear in [12],[13],[15],[16],[17],[18],[19],[20], and [21]. For example,

- In [12] the limited scheduling algorithm is analyzed in a similar methodology to the analysis described in [14]. For instance, eq. (5) in [12] presents the mean waiting time. However, the calculation of this waiting time ignores the dependency between the service time and the vacation length preceding it (see item 3 above).
  Moreover, according to eq. (7) in [12] the arrival rate must be lower than $\lambda_0 = (2(m-1)(\bar{L}-1))^{-1}$. If the arrival rate approaches this value, the cycle length defined in [12] approaches *infinity*. However, since the longest possible cycle is composed of only 5-slot packets, the maximal cycle length is $2(m-1)\cdot 5$ slots.
  Although both [12] and [14] deal with the limited scheduling regime, the vacation length described in eq. (4) in [12] differs from the vacation length described in eq. (3) in [14]. In both cases the use of the vacation model leads to approximate results disregarding the exact vacation length.
- The analysis of the limited scheduling algorithm in [13] and [16] is *very similar to* the analysis in [14]. The only difference is that in [13] and [16] it is assumed that the master does not generate traffic and it only routes packets between the slaves.
- In [15],[19], and [21] the performance of scatternets composed of two piconets connected through a bridge is analyzed. The performance of the scatternets is analyzed for exhaustive and limited intra-piconet scheduling algorithms. The limited algorithm is analyzed in a similar manner to the analysis in [14].
- The analysis of the *exhaustive regime* in [14] is also based on the theory of M/G/1 queue with vacations. In [29] we show that the PGF of the time to exhaust the queues derived according to [14] differs from the correct PGF. Moreover, we argue that due to the reasons discussed above, the use of the model of M/G/1 queue with vacations in order to analyze the exhaustive regime leads to approximate results. The same remarks apply to works [17],[18],[19],[20], and [21].

5 Numerical Results

In this section we present exact numerical results computed according to the analysis in Section 4.2. Then, we demonstrate the difference between our results and those presented in [14]. It turns out that in some cases the results obtained according to [14] *underestimate* the mean access delay by more than 50%. We guess that the noticeable difference between analytical and simulation results in high arrival rates, indicated in [12], is due to the inaccuracy of the analytical results there.

Fig. 5 illustrates the exact mean access delay (computed according to (3)) in piconets with various numbers of slaves in which the probabilities of 1, 3, and 5-slot packets are equal ($p_1 = p_3 = p_5 = 1/3$). The figure presents the delay (in slots) as a function of the load in the system (defined in Section 4.2 as $2(m-1)\lambda\bar{L}$).

Fig. 6 compares the mean access delay computed according to [14] to the mean access delay computed according to our model (i.e. according to (3)) when all packets are 5 slots long ($p_5 = 1$). The delay is depicted as a function of the system load in a piconet with 2 slaves (m=3) and in a piconet with 4 slaves (m=5). The figure demonstrates that the results obtained in [14] significantly underestimate the access delay.

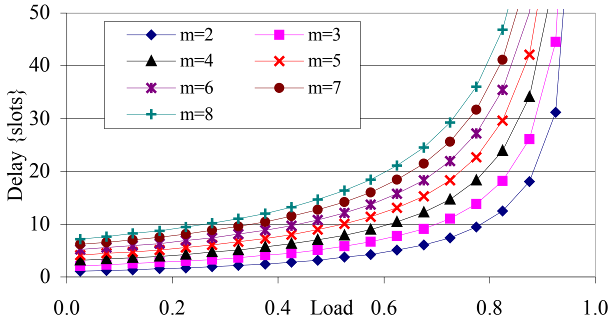


Fig. 5. The exact mean access delay (computed by (3)) in piconets in which $p_1 = p_3 = p_5 = 1/3$

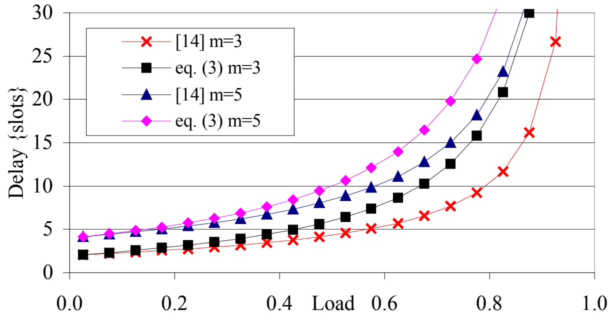
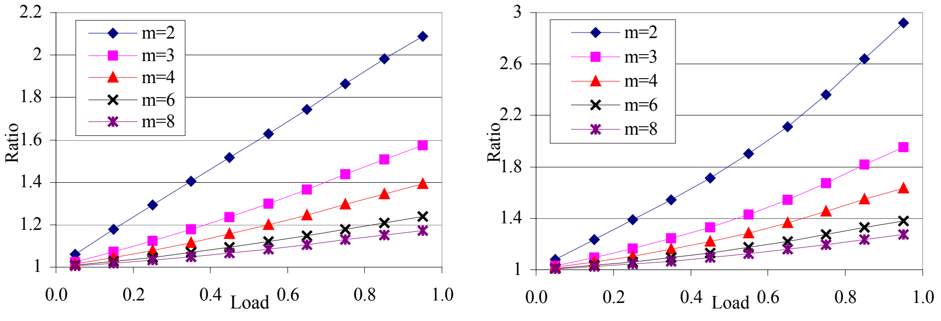


Fig. 6. The mean access delay derived according to [14] and the exact mean access delay (computed by (3)) in piconets with 2 and 4 slaves in which all packets are 5 slots long ($p_5 = 1$)

Fig. 7-A presents the ratio of the exact mean access delay to the mean access delay computed according to [14], in piconets with various numbers of slaves in which the probabilities of 1, 3, and 5-slot packets are equal ($p_1 = p_3 = p_5 = 1/3$). Fig. 7-B presents the same ratio in piconets in which all packets are 5 slots long ($p_5 = 1$).



A

B

Fig. 7. The ratio of the exact mean access delay (obtained by (3)) to the mean access delay derived according to [14] in piconets in which (A) $p_1 = p_3 = p_5 = 1/3$ and (B) $p_5 = 1$

6 Conclusions

This work presents an analytical study of the limited (pure round robin) scheduling algorithm for Bluetooth piconets, and examines the analytical study of this algorithm in [14]. We have modeled a piconet in which all packets are 1 slot long operated according to the limited scheduling algorithm as a TDMA system. Then, we showed that a piconet operated according to the limited scheduling algorithm is equivalent to a 1-limited polling system, and derived exact analytic results for symmetrical systems. These results differ from those obtained in [14] which can actually be viewed as approximate results. We have argued that [12],[13],[15],[16],[17],[18],[19],[20], and [21] seem also to present approximate results. Finally, we have provided numerical examples that illustrate the difference between the exact results and those presented in [14].

Future study will focus on utilizing the equivalence between a piconet and a 1-limited polling system along with the vast amount of research on 1-limited polling in order to obtain good approximate results for asymmetrical piconets with complex traffic patterns. One of the first attempts in that direction has been recently made in [11] where approximate results that outperform those in [10] and [14] have been presented. Moreover, due to the inherent complexities in obtaining the PGF of the time to exhaust the queues at the master and a given slave in the gated and exhaustive regimes (presented in [29]), it seems that there is no closed form expression for the delay under such regimes. Thus, a major future research goal is to obtain a good (at least approximate) analysis of such regimes.

Acknowledgments

We thank Nir Naaman and Daniele Miorandi for helpful discussions, and Ronen Kofman and Carlo Caimi for obtaining simulation results. We also thank the anonymous reviewers for their helpful comments.

References

1. Bertsekas, D.P., Gallager, R.: Data Networks. Prentice-Hall Inc., New Jersey (1992)
2. Bluetooth SIG: Specification of the Bluetooth System – Version 1.1. (Feb. 2001)
3. Bray, J., Sturman, C.: Bluetooth 1.1 Connect without Cables. Prentice-Hall Inc., New Jersey (2001)
4. Bruno, R., Conti, M., and Gregori, E.: Bluetooth: Architecture, Protocols and Scheduling Algorithms. Cluster Computing, Vol. 5 (Apr. 2002) 117–131
5. Capone, A., Gerla, M., Kapoor, R.: Efficient Polling Schemes for Bluetooth Picocells. Proc. IEEE ICC'01 (June 2001)
6. Har-Shai, L., Kofman, R., Zussman, G., Segall, A.: Inter-Piconet Scheduling in Bluetooth Scatternets. Proc. OPNETWORK 2002 (Aug. 2002)
7. Johansson, N., Korner, U., Johansson, P.: Performance Evaluation of Scheduling Algorithms for Bluetooth. Proc. IFIP TC6 Int. Conf. on Broadband Comm. (Nov. 1999)

8. Levy, H., Sidi, M.: Polling Systems: Application, Modeling, and Optimization. IEEE Trans. on Comm., Vol. 38 (Oct. 1990) 1750–1760
9. Levy, Y., Yechiali, U.: Utilization of the Idle Time in an M/G/1 Queueing System. Management Science, Vol. 22 (Oct. 1975) 202–211
10. Miorandi, D., Caimi, C., Zanella, A.: Performance Characterization of a Bluetooth Piconet with Multi-Slot Packets. Proc. WiOpt'03 (Mar. 2003)
11. Miorandi, D., Zanella, A.: An Approximate Delay Analysis for Limited-1 Polling in a Bluetooth Piconet. Working Paper (May 2003)
12. Misic, J., Misic, V.B.: Queueing Analysis of Bluetooth Piconet Traffic. Proc. SPECTS'02 (July 2002)
13. Misic, J., Misic, V.B.: On Delay Times in a Bluetooth Piconet: The Impact of Different Scheduling Policies. Proc. EurAsia ICT 2002, Lecture Notes in Computer Science, Vol. 2510 (eds: Shafazand M.H., Tjoa A M.). Springer-Verlag (Oct. 2002) 891–898
14. Misic, J., Misic, V.B.: Modeling Bluetooth Piconet Performance. IEEE Comm. Letters, Vol. 7 (Jan. 2003) 18–20
15. Misic, J., Misic, V.B.: Bridges of Bluetooth County: Topologies, Scheduling, and Performance. IEEE J. on Selected Areas in Comm., Vol. 21 (Feb. 2003) 240–258
16. Misic, V.B., Misic, J.: On Bluetooth Piconet Traffic Performance", Proc. IEEE PIMRC'02 (Sep. 2002)
17. Misic, V.B., Misic, J.: Minimizing End-to-End Delays in Bluetooth Scatternet with a Slave/Slave Bridge. Proc. ICCCN'02 (Oct. 2002)
18. Misic, V.B., Misic, J.: Queueing Analysis of Delays in Bluetooth Slave/Slave Bridge. Proc. IEEE/ACM MASCOTS'02 (Oct. 2002)
19. Misic, V.B., Misic, J.: Bluetooth Scatternet with a Master/Slave Bridge: a Queueing Theoretic Analysis. Proc. IEEE GLOBECOM'02 (Nov. 2002)
20. Misic, V.B., Misic, J.: Performance of Bluetooth Bridges in Scatternets with Exhaustive Service Scheduling. Proc. HICSS-36 (Jan. 2003)
21. Misic, V.B., Misic, J.: Performance of Bluetooth Slave/Slave Bridge. Telecommunication Systems, Vol. 22 (Jan. – Apr. 2003) 221–239
22. Racz, A., Miklos, G., Kubinszky, F., Valko, A.: A Pseudo Random Coordinated Scheduling Algorithm for Bluetooth Scatternets. Proc. ACM MOBIHOC'01 (Oct. 2001)
23. Sidi, M., Levy, H., Fuhrmann, S.: A Queueing Network with a Single Cyclically Roving Server. Queueing Systems, Vol. 11 (1992) 121–144
24. Takagi H.: Analysis of Polling Systems. MIT Press (1986)
25. Takagi, H.: Queueing Analysis: a Foundation of Performance Evaluation, Vol. 1 – Vacation and Priority Systems. North-Holland (1991)
26. Yechiali, U.: Analysis and Control of Polling Systems. In: Performance Evaluation of Computer and Communication Systems (eds: Donatiello L., Nelson R.). Springer-Verlag (1993) 630–650
27. Zussman, G., Segall, A.: Capacity Assignment in Bluetooth Scatternets – Analysis and Algorithms. Proc. IFIP-TC6 Networking 2002, Lecture Notes in Computer Science, Vol. 2345 (eds: Gregori E. et al.). Springer-Verlag (May 2002) 411–422
28. Zussman, G., Segall, A.: Capacity Assignment in Bluetooth Scatternets – Optimal and Heuristic Algorithms. To appear in: ACM/Kluwer Mobile Networks and Applications (MONET), Vol. 9, No. 1 (Feb. 2004)
29. Zussman, G., Yechiali, U., Segall, A.: On the Exact Analysis of Bluetooth Scheduling Algorithms. Submitted for Publication (available from http://www.comnet.technion.ac.il/~gilz /pub_files/BT_ex_li.pdf) (May 2003)