CCIT Report #414 February 2003

Bluetooth Time Division Duplex - Exact Analysis as a Polling System

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

[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. Bluetooth enables wireless communication via ad-hoc networks. The basic network topology (*piconet*) is a collection of slaves controlled by a master. A *scatternet* is a multihop network of piconets. Efficient piconet and scatternet communication requires design of intra and inter-piconet scheduling algorithms. Thus, numerous scheduling algorithms have been recently proposed. Analytical performance evaluation of such algorithms has great importance, since it may provide insight on their design and optimization. However, due to inherent complexities of the Bluetooth Medium Access Control (MAC), the performance of these scheduling algorithms has been analyzed mostly via simulation. Recently, Misic and Misic [18],[19],[20],[22],[23],[24],[25],[26] have claimed to provide exact analytic results regarding intra and inter-piconet scheduling algorithms which are based on the theory of M/G/1 queue with vacations. In this paper, we present alternative analytic results which are based on the theory of polling systems. Using these results we show that the results presented in [20] are *incorrect,* as they are based on unsatisfied assumptions leading to inaccurate probability generating functions. Thus, in some cases [20] *underestimates* the intra-piconet delay by more than 50% and in other cases it *overestimates* the delay by more than 50%. We also indicate that for similar reasons the results presented in [18],[19],[21],[22],[23],[24],[25], and [26] seem to be incorrect.

Keywords: Bluetooth, Scheduling, Polling, Queueing, Limited, Exhaustive, Personal Area Network (PAN), Piconet, Scatternet

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 [4],[5]. The basic Bluetooth 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 Figure 1). A device that is a member of more than one

This research was supported by a grant from the Ministry of Science, Israel and by Intel.

piconet (referred to as *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).

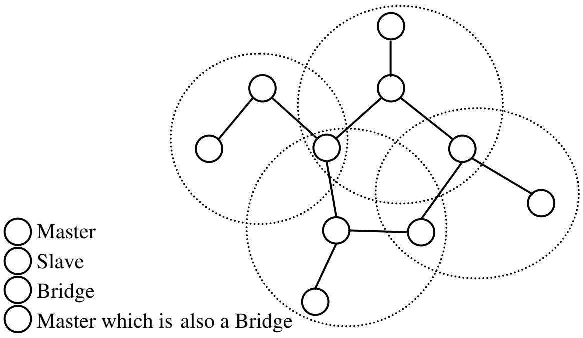

Figure 1. An example of a Bluetooth scatternet

In the Bluetooth specifications [4], 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 [33],[34]. 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 (see for example [1],[6],[7],[10],[11],[12],[13],[27] 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. We will show that for a few intra-piconet scheduling regimes, a piconet can be analytically modeled as a polling system[1]. However, as mentioned in [7], 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.

Recently, Misic and Misic [18],[19],[20],[22],[23],[24],[25],[26] have *claimed to provide exact analytic results* regarding the performance of various intra and inter-piconet scheduling regimes. Their analysis is based on the theory of M/G/1 queue with vacations (see [9],[15],[29], and [31]) and to the

[1] A polling system consists of several queues served by a single server according to a set of rules (polling scheme) [3, p. 200],[14],[30],[32].

best of our knowledge is the only available analytical modeling attempt regarding the performance of Bluetooth scheduling algorithms. In particular, in [20] closed form solutions for the *limited* (pure round robin) and *exhaustive* scheduling algorithms are exhibited. In this paper, we present alternative analytic results and show that the main results provided by Misic and Misic in [20] are incorrect, as they are based on unsatisfied assumptions leading to incorrect probability generating functions. We also indicate that for similar reasons the results presented in [18],[19],[21],[22],[23],[24],[25], and [26] seem to be incorrect. To the best of our knowledge, the results presented in this paper are the only *correct* exact analytic results regarding the performance of Bluetooth scheduling algorithms. In addition, we are not aware of any attempt to derive exact results regarding the performance of piconets with asymmetrical traffic by a queueing theoretic approach, besides our present work.

We model a piconet operated according to the *limited scheduling algorithm* as a 1-limited polling system and derive exact analytic results. Accordingly, it is shown that the analysis of the limited algorithm presented in [20] is incorrect. The oversight in [20] is that it ignores important dependencies incorporated in the piconet operation model when applying the results of the M/G/1 queue with vacations to the analyzed system. Then, we compare numerical results obtained according to [20] with numerical results computed by our model and show that [20] underestimates the delay in the system.

Further, we study the *exhaustive scheduling algorithm* and show that a piconet with unidirectional traffic can be modeled as an exhaustive polling system. Then, we show that due to special characteristics of the Bluetooth MAC, it seems that there is no closed form expression for the probability generating function of the time (in slots) to exhaust the queues at the master and a given slave. Thus, this probability generating function is significantly different from the function derived in [20]. In addition, we argue that the M/G/1 queue with vacations model cannot be used directly to analyze the exhaustive algorithm. We also provide numerical results that demonstrate the difference between the results derived in [20] and the exact results. Finally, since the analysis of inter-piconet scheduling algorithms in [21],[23],[24],[25], and [26] is mostly based on the analysis of the exhaustive regime, it is shown that similar errors appear in the analysis of these algorithms.

We note that Miorandi et al. [17] have recently presented an *approximate* analysis of the limited scheduling algorithm for a piconet with *asymmetrical* traffic. As opposed to [20], which models the piconet as an M|G|1 queue with vacations, Miorandi et al. [17] derive results based on a renewal process and *indicate that their analysis provides approximate results*. Their analysis supports our observation that the analysis of the limited algorithm in [20] does not provide exact results (for more details see Section 4.3).

The rest of the paper is organized as follows. Section 2 gives a brief introduction to the Bluetooth technology and describes related work, while Section 3 presents the model. In Sections 4 and 5, we analyze, respectively, the limited and the exhaustive scheduling algorithms and discuss the analysis of these algorithms in [18],[19],[20],[21],[22],[23],[24],[25], and [26]. Finally, in Section 6, we summarize the main results and discuss future research directions.

2 Background

2.1 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] 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* (in case it has nothing to send). We shall 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 Figure 2.

[1] In this paper we concentrate on networks in which only data links are used.

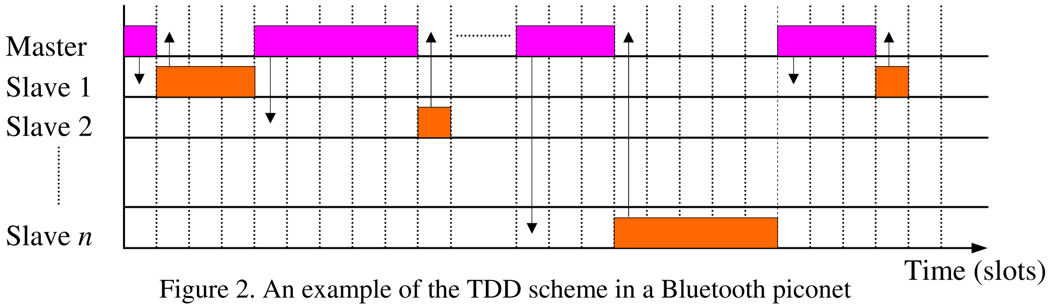

Figure 2. An example of the TDD scheme in a Bluetooth piconet

The master schedules the traffic within a *piconet* according to an *intra-piconet* scheduling algorithm. In this paper, we shall focus on two well-known intra-piconet scheduling regimes, which were analyzed in [20]:

– *Limited Round Robin* (*Pure Round Robin* [7]) – 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.

Connected piconets in the same geographic area form a *scatternet*. In a scatternet, a unit can participate in two or more piconets, on a time-sharing basis, and even change its role when moving from one piconet to another (we refer to such a unit as a *bridge*). A bridge can be a slave of a few masters (referred to as slave/slave bridge) or a master in one piconet and a slave in another piconet (referred to as master/slave bridge). Notice that a unit cannot be a master in more than one piconet. Figure 1 above illustrates an example of a scatternet including a master/slave bridge and a slave/slave bridge. The presence of a bridge in different piconets has to be controlled by an *inter-piconet* scheduling algorithm. One of the roles of this algorithm is to establish “rendezvous points” in which the bridge can switch between piconets.

2.2 Related Work

Due to the special characteristics of Bluetooth networks, many theoretical and practical questions regarding their performance have been raised. Two main issues that received relatively much attention are scatternet formation and scheduling. In this section we briefly review previous work related to these issues.

The Bluetooth specification does not indicate any method for *scatternet formation*. Thus, several scatternet formation algorithms have been recently proposed (see [16] for a through review of these algorithms). For instance, Basagni and Petrioli [2] propose the BlueMesh algorithm that constructs a scatternet topology with multiple paths between any pair of nodes. A different approach is taken by Chiasserini et al. [8] who formulate the topology construction problem as a centralized optimization problem and describe procedures for insertion and removal of nodes from the scatternet. Another algorithm which is based on the assumption that each node knows the location of itself and its neighbors is presented by Li et al. [16].

Once the scatternet is constructed, the link capacities that should be allocated in each piconet, such that the network performance will be optimized, have to be determined. Accordingly, in [33] and [34] the scatternet capacity assignment problem is formulated and optimal and heuristic algorithms for its solution are proposed. The solution of the capacity assignment problem is a desirable input to intra and inter-piconet scheduling algorithms.

Few of the first simple *intra-piconet* scheduling algorithms were proposed and evaluated via simulation in [12]. Since then, numerous intra-piconet scheduling algorithms tailored to optimize various performance metrics (e.g. delay, throughput, and energy consumption) and designed for different traffic patterns (e.g. TCP traffic) have been proposed. For example, Capone et al. [7] and Har-Shai et al. [11] study by simulation the delay in various ideal and practical intra-piconet scheduling algorithms. In [7] and [10] the performance of intra-piconet scheduling algorithms when the transport layer is TCP is also studied. Finally, Bruno et al. [6] propose an algorithm that dynamically adapts the polling frequency to the traffic patterns.

Miorandi et al. [17] present an *approximate* analysis of the limited (pure round robin) scheduling algorithm in a piconet with *asymmetrical* traffic that is based on a renewal process. Among other reasons, the results are approximate, since the tool of probabilistic routing [14] is employed while it is assumed that the various resulting flows are independent.

The scheduling problem is much more complicated in a scatternet. Usually an *inter-piconet* scheduling algorithm establishes "rendezvous points" in which bridges can switch between piconets. For example, Baatz et al. [1] and Johansson et al. [13] describe inter-piconet scheduling algorithms which use the Bluetooth low power mode "sniff" to establish recurring rendezvous points. On the other hand, Har-Shai et al. [11] describe an inter-piconet scheduling algorithm tailored for small-scale scatternets. This algorithm uses the low power mode "hold" to establish rendezvous points adjusted to the traffic patterns. Finally, Racz et al. [27] describe an intra and inter-piconet scheduling algorithm in which the nodes assign rendezvous points with their peers, such that the sequence of these points follows a pseudo random process.

3 The Model

In this section we present a model for a Bluetooth piconet which is based on the model described in [20]. We use a 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 $\overline{B}^{(2)}$. It is assumed that the distribution of the burst size is geometric. We will show that the results presented in [20] are incorrect 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 $\overline{L} = p_1 + 3p_3 + 5p_5$. It is assumed that the

In [20], 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 with respect to the slave devices). 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.

In [20], 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 time from the moment a packet arrives to the uplink queue at the slave, to the moment it is received at another slave (denoted by W_e)[1].

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

In Sections 4 (limited regime) and 5 (exhaustive regime), we present analytic models for the computation of the delay in a piconet. Simplifying the model in [20], 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. 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 [20] should also hold for this scenario. However, we show that these results do not hold even for the simplified scenario. Furthermore, it is argued that the results obtained in [20] for the more complicated model, which includes intra-piconet routing by the master, suffer from the assumption that the arrival of traffic to the downlink queues is Poisson, independent of the uplink traffic.

The model in [20] is symmetrical such that the arrival rates to all downlink and uplink queues are the same. However, in Sections 4 and 5 we present few results for asymmetrical piconets. In these

[1] Since W_e is defined in [20] as the sum of W_a and W_m, it is actually only the total time a packet spends in the master's and the slave's queues not including the service times.

sections, we consider three different cases in which packets are generated at every uplink and downlink queue according to a Poisson arrival process:

- *Symmetrical system* – The arrival rate to every downlink and uplink queue is λ.
- *Half-Symmetrical system* – The arrival rate to all the downlink queues is the same (denoted by λ_d) and the arrival rate to all the uplink queues is the same (denoted by λ_u), but $\lambda_d \neq \lambda_u$.
- *Asymmetrical system* – The arrival rates to each of the downlink and uplink queues are not necessarily equal. We denote the arrival rate to slave *i* as λ_u^i and the arrival rate to the master of packets intended for slave *i* by λ_d^i. The mean access delay of packets in the uplink queue of slave *i* is denoted by $\overline{W}_a^i$ and the mean queueing delay at the master of packets intended to slave *i* is denoted by $\overline{W}_m^i$.

4 Limited (Pure Round Robin) Regime

In this section we focus on the limited round robin scheduling regime. First, we show that a piconet operating according to this regime and in which all packets are 1 slot long is equivalent to a TDMA (Time Division Multiple Access) system[1]. 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 [20] and indicate that [20] applies the results of the M/G/1 queue with vacations *without* considering the dependencies between the queues. Finally, we shall compare numerical results obtained according to the analysis in [20] to numerical results following our analysis and show that in some cases [20] *underestimates* the access delay by more than 50%.

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* [3, 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

---

[1] Although in [20] it is assumed that the system is symmetrical, we also derive analytic results for an asymmetrical system.

vacation length both equal to $2(m-1)$ [3, p.194]. Accordingly, applying [3] 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 ($\overline{W}_a$) and the mean queueing delay at the master ($\overline{W}_m$) (in slots):

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

Notice that under the assumptions that the master does not route packets and the arrival process to the master is Poisson, the mean access delay (queueing delay at the slave) is equal to the mean queueing delay at the master.

For this simple scheduling regime, the result regarding the access delay obtained in eq. (5) in [20] coincides with our result (1).

The model presented in [20] is symmetrical with respect to the slaves (i.e. the arrival rates to all master and slaves queues are the same). 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 easily obtained for an asymmetrical piconet. Since in this (deterministic) case every link can be independently analyzed as an M|D|1 queue, the mean access delay (in slots) of packets in the uplink queue of slave i ($\overline{W}_a^i$) is a function of the arrival rate to slave i (λ_u^i). It is again derived from eq. (3.58) in [3]:

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

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.

---

[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 [3, p. 201],[14],[30].

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, one slot must be used during the uplink communication (by the NULL packet). Thus, in order to model the piconet as a 1-limited *polling* system, we define the switchover time to each of the queues as 1 slot. Accordingly, when data packets are sent, some of the data is actually sent during the "switchover" time. Therefore, the service time of a 1-slot data packet is defined as 0 slots, the service time of a 3-slot packet is 2 slots, and the service time of a 5-slot packets is 4 slots.

We focus on symmetrical systems in which the arrival rates to all queues are equal[1]. By applying the model for a symmetrical limited *gated* polling system[2] described in [3, p. 201] we can obtain the mean waiting time of a packet in a queue. The waiting time in [3] is defined as the time a packet waits until its service starts (i.e. the time until the end of the switchover that precedes it). In our model, the first slot of a data packet is considered as the switchover time. Thus, in order to obtain the mean waiting time in a piconet, i.e. the mean access delay, one has to deduct 1 slot from the expression for the waiting time in [3], eq. (3.77). Accordingly, applying [3] 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 $\rho = 2(m-1)\lambda(\bar{L}-1)$, and the second moment of the service time (denoted in [3] as $\overline{X^2}$) is $4p_3 + 16p_5$, and deducting 1 time unit (i.e. 1 slot), we obtain the mean access delay ($\overline{W}_a$) and the mean queueing delay at the master ($\overline{W}_m$) (in slots):

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

Notice that in this system it must hold that $2(m-1)\lambda\bar{L} < 1$. Thus, the maximal allowable arrival rate is given by $\lambda_{\max} = (2(m-1)\bar{L})^{-1}$. We shall refer to $2(m-1)\lambda\bar{L}$ as the load in the limited system.

---

[1] The problem of computing exact mean delays in general 1-limited polling systems has not been resolved yet [14].

[2] 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.

The result presented in (3) was verified by two independent simulation models based on OPNET (for more details regarding the simulation models, see [11] and [17]). Moreover, in a system in which only 1-slot packets are used (i.e. $p_1 = 1$, $p_3 = 0$, and $p_5 = 0$), the result presented in (3) reduces to (1).

4.3 Examination of the Analysis as an M/G/1 Queue with Vacations [20]

The theory of M/G/1 queue with vacations is used in [20] 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 their analysis and point out that their mistake is the direct use of the model of M/G/1 queue with vacations [15],[31] to analyze a 1-limited polling system, *without taking into consideration the dependencies between the queues*. We note that if their analysis was correct, the analysis of symmetrical 1-limited polling systems would have been much simpler than those presented in [3] and [30]. For simplicity, we assume that the traffic is non-bursty ($G_b(x) = x$) and that packets are generated at every master-to-slave (downlink) queue according to a Poisson arrival process with arrival rate λ.

The *piconet service cycle time* X_c is defined in [20] 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 [20], 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 in [20][1]. According to this analysis, 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 $m-2$ uplink 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 [20], 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 ([29, 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 [20] reduces to [29, 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 crucial problems in the model presented above. For the clarity of presentation, the description of each isolated problem ignores the existence of other problems.

1. *The cycle length described in (6) is inappropriate for use as a service time in a vacation model.* Consider the case in which the server returns to slave i from a vacation or completes "service" there (i.e. the master completes a cycle) and at least one packet is found in the uplink queue of slave i. In such a case, the server will not take a vacation and a service period will start. According to [20], 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 $G_u(x)$

[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 [31].

(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 ($G_{X_c}(x)$) 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 (in case 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) is incorrect[2].

   We note that Miorandi et al. [17], who present an *approximate* analysis of the limited algorithm, mention that "*The cycle period may be expressed as the sum of the times spent for data exchange on the (i,j)-th link, which due to our assumptions, are assumed independent*". This independence assumption is one of the main reasons that the results presented in [17] are approximate. However, this assumption is not made in [20] and the results there are claimed to be *exact*.

3. *The cycle time depends on the length of the vacation or the cycle that precedes it*. In [20] the vacation model described by Takagi [31] is used in order to compute the mean access delay, presented in eq. (8). One of the important assumptions made in [29, 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 [20] 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

---

[1] This problem becomes critical for small piconets in which $G_{X_c}(x)$ is composed of only a few components.

[2] We note that 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.

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 service times, which are independent of the vacation lengths, does not seem to hold in the model presented in [20]. Therefore, the use of the delay presented in eq. (8) is problematic.

In Section 4.1, we have analyzed a piconet operated in the 1-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 results regarding the access delay obtained in eq. (5) in [20] coincides with our result (1). However, due to the problems described above, in a "standard" piconet (in which the packets are 1,3, and 5 slot long) the results presented in [20] 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 [20] 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 ($\overline{W}_e = \overline{W}_a + \overline{W}_m$). A method for computing the mean queueing delay at the master ($\overline{W}_m$) is described at the end of Section 2 in [20]. For non-bursty traffic, $\overline{W}_m$ computed according to that method is equal to $\overline{W}_a$ described in (8) (i.e. $\overline{W}_a = \overline{W}_m$). Thus, it seems that an underlying assumption in [20] 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. [28]). 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 from the master to another slave). We note that a similar observation has also been made by Miorandi et al. [17], who apply the model of probabilistic routing. According to their model, the traffic from the master to a particular slave is computed as the sum of the traffic generated in the whole piconet for that particular slave, and the various resulting flows are *assumed to be independent*. They mention that this assumption results in an *approximate* analysis. Moreover, they note, "*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*".

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.

Finally, we note that inappropriate assumptions, similar to the ones indicated in this section, also appear in [18],[19],[21], and [22]. For example,

– In [18] the limited scheduling algorithm is analyzed in a similar methodology to the analysis described in [20]. For instance, eq. (5) in [18] 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 [18] the maximal allowable arrival rate is given by $\lambda_0 = (2(m-1)(\bar{L}-1))^{-1}$. If the arrival rate approaches this value, the cycle length defined in [18] approaches infinity. Since the longest possible cycle is composed of only 5-slot packets, the maximal cycle length is $10(m-1)$ slots, and therefore, the result in [18] is unreasonable. Recall that we have shown that the maximal allowable arrival rate is given by $\lambda_{\max} = (2(m-1)\bar{L})^{-1}$. If the arrival rate approaches this value, the cycle length defined in [18] approaches $2(m-1)\bar{L}$.

  Oddly, although both [18] and [20] deal with the limited scheduling regime, the vacation length described in eq. (4) in [18] differs from the vacation length described in eq. (3) in [20]. In both cases the use of the vacation model is problematic disregarding the exact vacation length.

– The analysis of the limited scheduling algorithm in [19] and [22] is *very similar to* the analysis in [20]. The only difference is that in [19] and [22] it is assumed that the master does not generate traffic and its only role is to route packets between the slaves.

– In [21] the performance of scatternets composed of two piconets connected through a master/slave bridge and a slave/slave bridge is analyzed. In both cases, the master exchanges packets with the bridge according to an exhaustive regime. The performance of the two scatternets is analyzed for exhaustive and limited intra-piconet scheduling algorithms. The analysis of the intra-piconet limited scheduling algorithm is very similar to the analysis in [20].

4.4 Numerical Results

In this section we present exact numerical results computed according to the analysis in Section 4.2. Then, in order to demonstrate the difference between the results presented in [20] and the exact results, we calculate numerical results according to the analysis in [20] and compare them to results computed by our eq. (3). We show that in some cases the results obtained according to [20] *significantly underestimate* the mean access delay. Notice that in [18], Misic and Misic mention that "*The difference between analytical and simulation results is quite small, although it tends to increase slightly with packet arrival rate λ. The difference becomes noticeable at high arrival rates, where mean waiting time is above 10 to 15* (slots)". It seems that the inconsistency between their simulation results and their analytic results is due to the inaccuracy of the analytic results. We note that in this section we present results computed for symmetrical piconets (i.e. the arrival rates to all the downlink and the uplink queues are equal).

In order to obtain $\overline{W}_a$ according to [20], we have used (4) – (8) and computed $\overline{X}_c = G'_{X_c}(1)$, $\overline{X_c^2} = G''_{X_c}(1) + G'_{X_c}(1)$, $\overline{V}_l = G'_{V_l}(1)$, $\overline{V_l^2} = G''_{V_l}(1) + G'_{V_l}(1)$.

Figure 3 illustrates the exact mean access delay ($\overline{W}_a$) (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 $2(m-1)\lambda\overline{L}$).

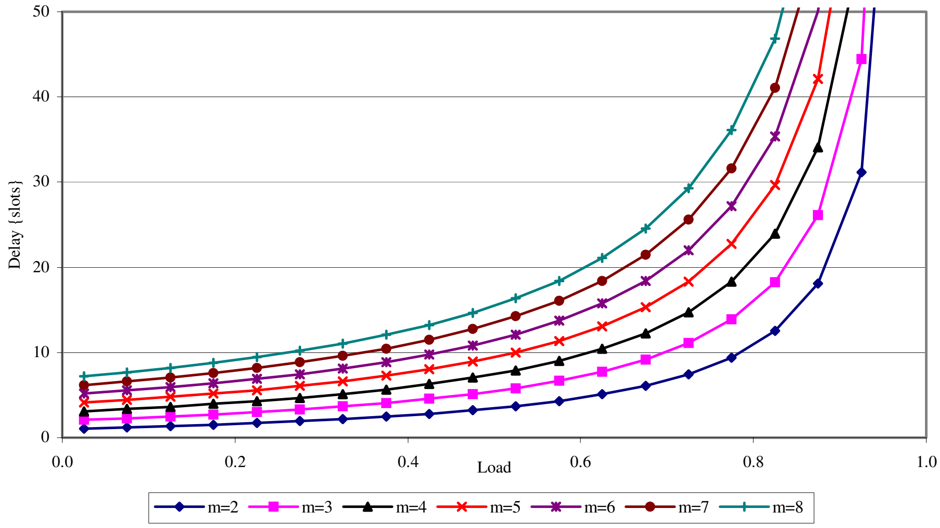

Figure 3. The exact mean access delay (computed according to (3)) in piconets operated according to the limited scheduling algorithm in which $p_1 = p_3 = p_5 = 1/3$

Figure 4 compares the mean access delay ($\overline{W}_a$) computed according to [20] to the mean access delay computed according to our model (i.e. according to (3)) when all packets are 5 slots long ($p_5 = 1$, $\overline{L} = 5$). The delay (in slots) 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 [20] significantly underestimate the access delay.

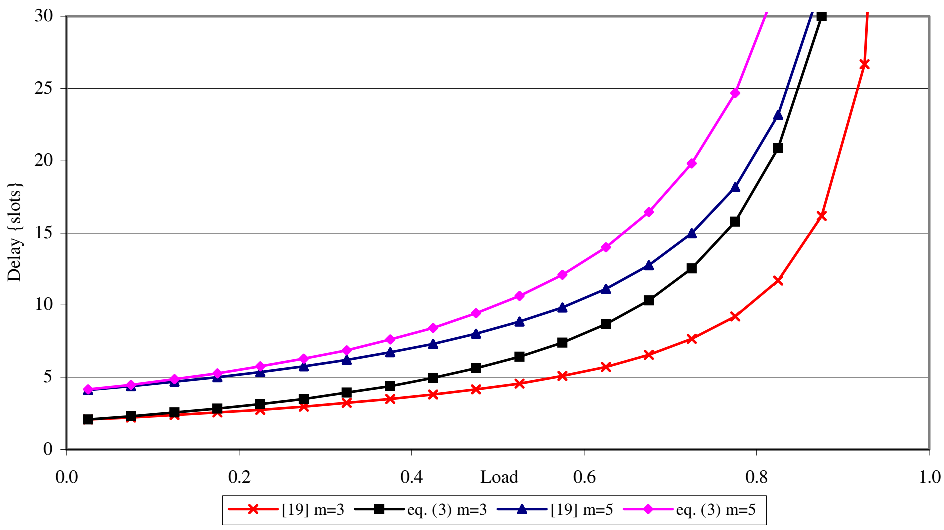

Figure 4. The mean access delay derived according to [20] and the exact mean access delay (computed according to (3)) in piconets with 2 and 4 slaves in which all packets are 5 slots long

As we have mentioned, the exact results derived in [18] are slightly different from the results derived in [20]. Thus, we have computed the mean access delay according to [18] and compared it to our results. In all cases we have checked, the mean delay obtained according to [18] is *lower* than the exact mean delay. Moreover, in Section 4.3 we have mentioned that the cycle length (X_c) is inappropriate for use as a service time in a vacation model. In order to evaluate the effect of this problem (and ignoring all other issues discussed in Section 4.3), we have redefined the PGF of the service time as:

$$G_S(x) = G_p(x)G_d(x)\left(G_d(x)G_u(x)\right)^{m-2}, \tag{9}$$

and computed the mean access delay using this "corrected" model.

Figure 5 compares the mean access delay ($\overline{W}_a$) computed according to [20] to the mean access delay computed according to our model, in a piconet with 4 slaves in which probabilities of 1, 3, and 5-slot packets are equal ($p_1 = p_3 = p_5 = 1/3$, $\overline{L} = 3$). It is seen that the delay values are higher than the values obtained in [20]. The figure also presents the access delay computed according to [18]. Although the results obtained according to [18] differ from the results obtained according to [20], they also underestimate the delay. In addition, the figure presents the access delay computed using the "corrected" service time as presented in (9). This "correction" slightly increases the delay. Yet, due to the other problems, discussed in Section 4.3, the computed delay is still lower than the exact delay.

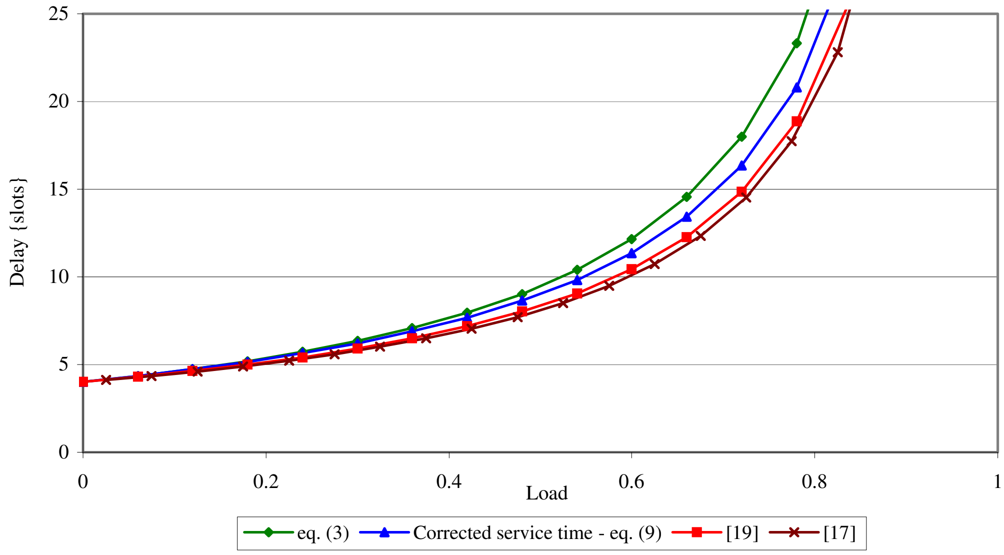

Figure 5. The exact mean access delay (obtained according to (3)), the mean access delay derived using the "corrected" service time (9), the mean access delay derived according to [20], and the mean access delay obtained according to [18] in a piconet with 4 slaves in which $p_1 = p_3 = p_5 = 1/3$

Figure 6 presents the ratio of the exact mean access delay to the mean access delay computed according to [20], in piconets with various numbers of slaves in which the probabilities of 1, 3, and 5-slot packets are equal ($p_1 = p_3 = p_5 = 1/3$). Figure 7 presents the same ratio in piconets in which all packets are 5 slots long ($p_5 = 1$). It can be seen that the results obtained in [20] *significantly* underestimate the access delay in a piconet operated in the limited regime.

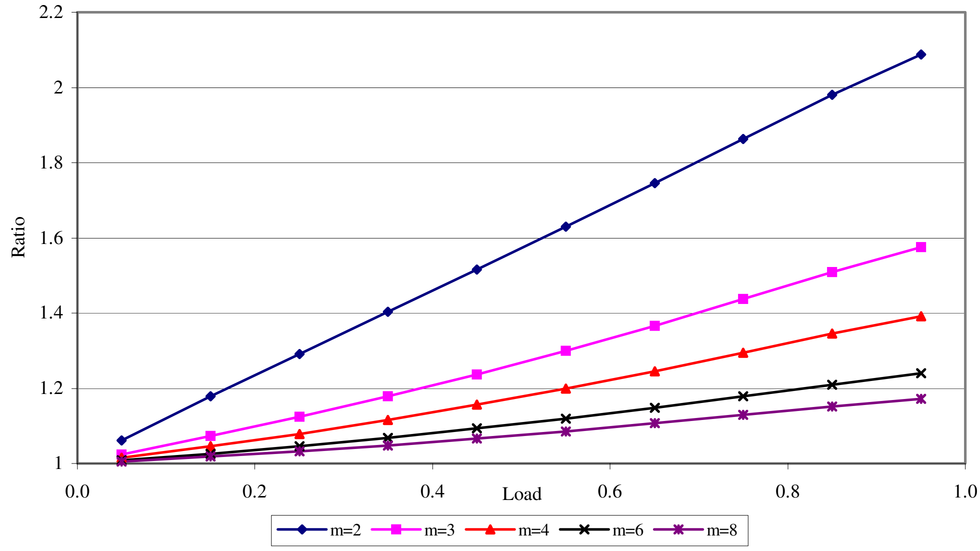

Figure 6. The ratio of the exact mean access delay (obtained according to (3)) to the mean access delay derived according to [20] in piconets in which $p_1 = p_3 = p_5 = 1/3$

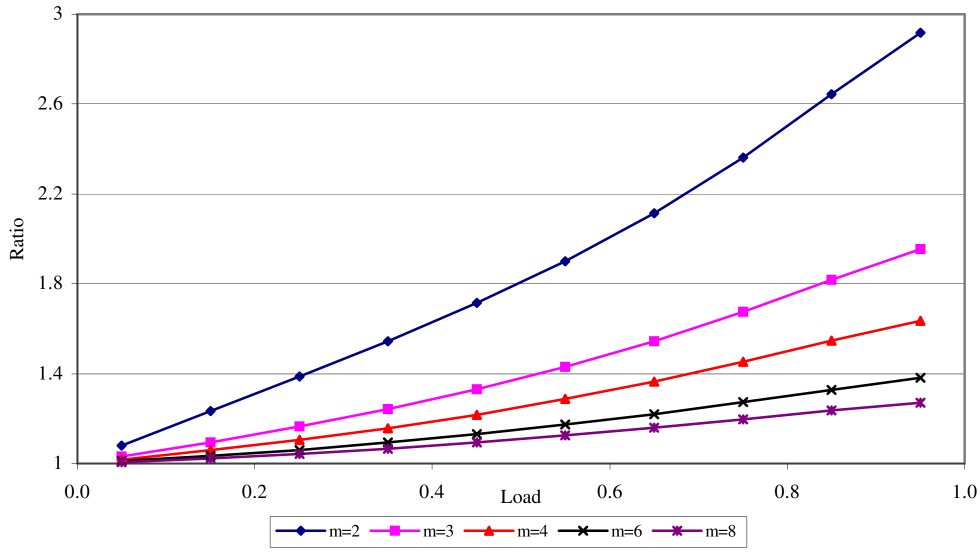

Figure 7. The ratio of the exact mean access delay (obtained according to (3)) to the mean access delay derived according to [20] in piconets in which all packets are 5 slots long ($p_5 = 1$)

5 Exhaustive Regime

In this section, we focus on the exhaustive round robin scheduling regime[1]. It is shown that a piconet with only unidirectional (e.g. slave-to-master) traffic can be modeled as an exhaustive *polling* system. Then, we consider a piconet with bi-directional traffic and outline the complexities in deriving the PGF

[1] In the exhaustive intra-piconet scheduling regime, the master communicates with the slaves according to a fixed cyclic order. It switches from one master-slave queue pair to the next only when the downlink and the uplink queues are empty.

of the time (number of slots) to exhaust the queues at the master and a given slave. These complexities mainly result from the special characteristics of the Bluetooth TDD mechanism and the use of POLL and NULL packets. We show that the PGF of the time to exhaust the queues derived according to [20] differs from the correct PGF and that the use of the model of M/G/1 queue with vacations in order to analyze the exhaustive regime is inappropriate. Finally, we provide numerical results that demonstrate the difference between the exact results and those presented in [20].

5.1 A Piconet with Unidirectional Traffic

Consider a piconet operated in an *exhaustive round robin* regime with unidirectional traffic (i.e. with only slaves-to-master or only master-to-slaves traffic). Recall that the model presented in [20] is symmetrical with respect to the master and the slaves. Namely, the arrival rates to the uplink and the downlink queues are equal. However, in this section, we consider piconets in which these arrival rates *differ*. First, we consider a *half-symmetrical* piconet (i.e. a piconet in which $\lambda_d \neq \lambda_u$). Then, we show that an *asymmetrical* piconet (in which the arrival rates to the uplink queues are not necessarily equal) can be analyzed in a similar manner.

We now concentrate on a piconet with only uplink traffic. When the master communicates with a particular slave it sends POLL packets, since it has nothing to send. The slave replies with data packets until its queue is empty. Then, it replies with a NULL packet which signals the end of the exhaustive communication with that particular slave[1].

At first glance, it seems that this regime can be modeled as an exhaustive *polling* system[2] in a straightforward manner. Namely, the service time can be defined as the packet length plus 1 slot (for the preceding POLL packet) and the switchover time can be defined as 2 slots (the POLL-NULL exchange). However, if a data packet arrives to an empty uplink queue during the transmission of the POLL packet, the slave starts transmitting it in the next slot. Consequently, in this straightforward model a packet that arrives during the first half of the total 2-slot switchover time actually cancels the switchover.

---

1 The termination of the master-slave exchange with a POLL-NULL exchange complies with the assumptions made in [20].

2 In an exhaustive *polling* system, at each visit of the server to a queue, *all* the packets in that queue are served. The server incurs a *switchover* time when it shifts from one queue to another [3, p. 198],[14],[30],[32].

Thus, an alternative modeling is required in order to model the piconet as an exhaustive polling system. To this end, we define the service time of a k-slot data packet (k = 1,3,5) as k+1 slots which are composed of the k slots of data, augmented by the *following* POLL packet. Thus, the service time of a 1-slot packet is defined as 2 slots, for 3-slot packet it is 4 slots, and for 5-slot packet it is 6 slots. The switchover time is defined as 2 slots, composed of the NULL packet ending the exchange with a particular slave and the POLL packet starting the exchange with the next slave.

For a *half-symmetrical* piconet ($\lambda_d = 0$ and $\lambda_u = \lambda > 0$) we apply the model for a symmetrical exhaustive polling system described in [3, p. 198]. Accordingly, using eq. (3.69) in [3], where the number of queues is $(m-1)$, the arrival rate is $(m-1)\lambda$, the switchover time is two slots with zero variance, the traffic intensity is $\rho = (m-1)\lambda(\overline{L}+1)$, and the second moment of the service time is $\overline{X^2} = 4p_1 + 16p_3 + 36p_5$, we obtain the mean access delay ($\overline{W}_a$) (in slots):

$$\overline{W}_a = \frac{(m-1)\{1+4\lambda(p_3+3p_5)\}}{1-(m-1)\lambda(\overline{L}+1)}. \tag{10}$$

Notice that in this system it must hold that $(m-1)\lambda(\overline{L}+1) < 1$. Thus, the maximal allowable arrival rate is given by $\lambda_{\max} = ((m-1)(\overline{L}+1))^{-1}$. We shall refer to $(m-1)\lambda(\overline{L}+1)$ as the load in the *unidirectional exhaustive* system.

The result presented in eq. (10) was verified by a simulation model based on OPNET (see [11]). Moreover, in a system with a single slave (m = 2) there is no difference between the exhaustive and the limited scheduling algorithms. The mean access delay in a piconet with unidirectional traffic of 1-slot packets (i.e. $p_1 = 1$) operated according to the limited regime was derived in Section 4.1 and is given by (2). For a single slave piconet with unidirectional traffic of 1-slot packets, the above result given by eq. (10) coincides with (2).

A half-symmetrical piconet with only *downlink* traffic (i.e. the arrival rate to each downlink queue is $\lambda_d = \lambda$ and to each uplink queue is $\lambda_u = 0$) can be modeled as an exhaustive polling system, in a similar manner. However, the operation model of such a piconet should be precisely defined. For example, since the traffic flows only from the master to the slaves so that the master has complete information on the status of its downlink queues, there is no reason to send a POLL packet in order to end a master-slave exchange. On the other hand, in case all queues are empty, the master should transmit POLL packets until a packet arrives to one of its downlink queues.

We now consider an *asymmetrical* piconet with only uplink traffic (i.e. the arrival rate to each downlink queue is $\lambda_d^i = 0$, the arrival rate to each uplink queue i is $\lambda_u^i > 0$, and the arrival rates to the uplink queues are not necessarily equal). Such a piconet can be analyzed in a similar manner to a half-symmetrical piconet with only uplink traffic. Namely, it can be modeled as an asymmetrical exhaustive *polling* system composed of $m - 1$ queues, with 2-slot switchover time and with service time of k+1 slots for a k-slot data packet (k = 1,3,5).

Accordingly, the PGF of the service times in each uplink queue can be computed in a similar way to the models for exhaustive polling systems described in [14],[30] and [32]. Then, the mean access delay in each uplink queue ($\overline{W}_a^i$) can be obtained by solving $(m - 1)^3$ equations. Since $m \leq 8$, the computational complexity is negligible. We note that results can be obtained even for the case in which the probabilities of a packet length being 1, 3, or 5 slots vary in different uplink queues.

Finally, we note that in Section 5.4, we shall numerically compare the results derived in this section to the results derived in [20] and show that [20] significantly underestimates the access delay in case of unidirectional traffic.

5.2 Analysis of a Single Channel

Analyzing the performance of scheduling regimes such as the exhaustive, gated, and globally gated [32] in a piconet with bi-directional traffic requires obtaining the PGF of the exchange time of a single master-slave queue pair (channel). This analysis is significantly complicated by the TDD mechanism and the use of POLL and NULL packets by the master and the slaves. In order to demonstrate the difficulties in analyzing the exhaustive regime, we analyze in this section a less complicated case, namely a single master-slave channel in a piconet, operated in the *gated regime*, and point out the obstacles in analyzing the TDD mechanism.

In the gated regime, only the packets that are found in the uplink and downlink queues when the master starts serving the master-slave queue pair are transmitted. If the number of downlink packets exceeds the number of uplink packets, the slave sends NULL packets as a response to some data packets. On the other hand, if the number of uplink packets exceeds the number of downlink packets, the master sends some POLL packets in order to allow the slave to reply with data packets. We note that in order for our analysis to comply with the assumptions made in [20], we assume that at the end of the master-slave exchange, the slave has to respond with a NULL packet to a POLL packet[1].

Figure 8 illustrates the operation of a gated regime in a piconet composed of a master and two slaves. When the master starts transmitting the packet intended to slave 1, there are 3 data packets (of sizes 3, 1, and 5 slots) in the downlink queue and a single data packet in the uplink queue, and therefore the slave replies with 2 NULL packets to the second and third data packets. Following the last data packet, the master sends a POLL packet which signals the end of the master-slave exchange. This packet is replied by a NULL packet. When the master starts transmitting a packet intended to slave 2, there is a single data packet (of size 3) in the downlink queue and 3 data packets (of sizes 3, 3, and 1) in the uplink queue, and therefore the master sends POLL packets in order to allow the slave to reply with its data packets.

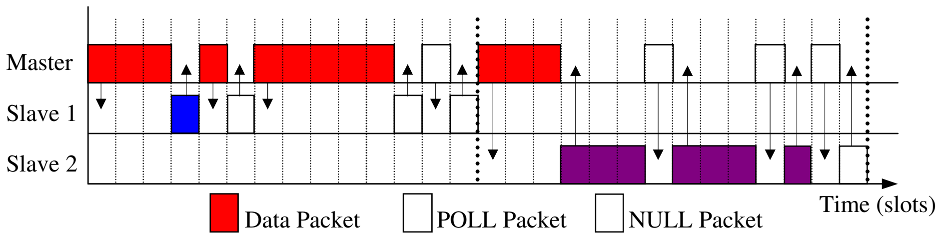

Figure 8. An example of the operation of the gated regime in a piconet composed of two slaves

Let X_G denote the time (number of slots) required for the exchange duration of a single master-slave channel in the gated regime. Namely, it is the number of slots required to serve all the packets which are present in both downlink and uplink queues at the instance when the master starts serving the queue pair plus 2 slots (required for the last POLL-NULL exchange). The PGF and the mean of X_G are denoted by $X_G(x)$ and $\overline{X_G}$, respectively. For simplicity, we assume that *all packets are 1 slot long* (i.e. $p_1 = 1$) and that packets have been accumulated in both queues for some T slots before the gated service starts. We define U and D as the number of packets accumulated in the uplink and downlink queues, respectively, during T slots ($U,D \sim Poisson(\lambda T)$).

[1] In some cases this POLL-NULL exchange is redundant.

Thus, when all packets are 1 slot long, X_G equals twice the maximum of U and D plus 2 slots. Namely, it is a function of the *maximum* of two Poisson random variables. Accordingly, the PGF of the time to serve a single master-slave channel is given by:

$$X_G(x) = E\left(x^{2\max(U,D)+2}\right) = x^2 \sum_{m=0}^{\infty} x^{2m} \mathrm{Prob}(\max(U,D)=m) \ , \tag{11}$$

where

$$\mathrm{Prob}(\max(U,D)=m) = 2e^{-\lambda T}\,\frac{(\lambda T)^m}{m!}\left(\sum_{j=0}^{m-1} e^{-\lambda T}\,\frac{(\lambda T)^j}{j!}\right) + \left(e^{-\lambda T}\,\frac{(\lambda T)^m}{m!}\right)^2 . \tag{12}$$

Unfortunately, it appears that in view of eq. (12) there is no closed form expression for (11). We note that according to (11), the mean time to serve a single master-slave channel is given by:

$$\overline{X_G} = E(2\max(U,D)+2) = 2E(\max(U,D)) + 2 \ . \tag{13}$$

5.3 Examination of the Analysis in [20]

Misic and Misic [20] analyze the performance of the *exhaustive* regime in a piconet with bi-directional traffic. Their analysis is based on 2 stages: (1) the derivation of the PGF of the time to exhaust a single master-slave queue pair, and (2) modeling the piconet as an M/G/1 queue with vacations. We now briefly describe their analysis and show that the derived PGF of the time to exhaust a single queue pair is incorrect. We will not elaborate on their 2nd stage inaccuracies, since they are similar to those discussed for the limited regime in Section 4.3.

Recall that in an *exhaustive* regime, the master does not switch to the next master-slave queue pair until both the downlink and the uplink queues are empty. Although the regime analyzed in [20] is referred to as *exhaustive*, it has some of the characteristics of the *gated* regime. According to [20] the master generates traffic and routes traffic generated by the slaves (see Section 3). Thus, in this exhaustive regime, when the master serves a particular master-slave channel, the packets present in the downlink queue at the beginning of an exchange will be transmitted. Moreover, the packets generated by the master during the exchange, which are intended for the particular slave, will also be transmitted. Accordingly, the arrival rate to the downlink queue *during the exchange period* is λ / (m – 1), whereas the overall mean arrival rate to the downlink queue is λ. Therefore, the analysis of the exhaustive regime is more complex than the analysis of the gated regime, which has its own difficulties, as presented in the previous section.

Denote the time (number of slots) to empty both channel queues for a particular slave in the exhaustive regime as X_{ms}. Its PGF and mean are denoted by $X_{ms}(x)$ and $\overline{X_{ms}}$. In [20], eq. (7), the PGF of X_{ms} is presented as:

$$X_{ms}(x) = e^{(\lambda_u+\lambda_d)X_c(x)(G_b(G_P(x))-1)} e^{|\lambda_u-\lambda_d|X_c(x)(G_b(x)-1)} x^2 , \tag{14}$$

where $X_c(x)$ denotes the PGF of the cycle length and is presented as:

$$X_c(x) = \left(X_{ms}(x)\right)^{m-1} . \tag{15}$$

Recall that we assume that the traffic is non-bursty ($G_b(x) = x$). Moreover, for simplicity, assume that all packets are 1 slot long (i.e. $p_1 = 1$ and $G_p(x) = x$). Under these assumptions, and letting q_j be the probability that the cycle length is j slots ($X_c(x) = \sum_{j=2}^{\infty} q_j x^j$), (14) reduces to:

$$X_{ms}(x) = e^{(\lambda_u+\lambda_d)\left(\sum_{j=2}^{\infty} q_j x^j\right)(x-1)} e^{|\lambda_u-\lambda_d|\left(\sum_{j=2}^{\infty} q_j x^j\right)(x-1)} x^2 , \tag{16}$$

If we assume that the arrival rates are symmetrical ($\lambda_u = \lambda_d = \lambda$), then by differentiating (16) the mean time to serve both queues is:

$$\overline{X_{ms}} = 2\lambda + 2 . \tag{17}$$

This result is *independent* of the number of slaves or the cycle length, and therefore *seems unreasonable*. Thus, we believe that Misic and Misic [20] meant that the PGF of X_{ms} is a function of the mean cycle length ($\overline{X}_c$) and not of the PGF of the cycle length ($X_c(x)$)[1]. Namely:

$$X_{ms}(x) = e^{(\lambda_u+\lambda_d)\overline{X}_c(G_b(G_P(x))-1)} e^{|\lambda_u-\lambda_d|\overline{X}_c(G_b(x)-1)} x^2 . \tag{18}$$

We now show that even this corrected equation is mistaken. Assuming that the traffic is non-bursty, all packets are 1 slot long, and the cycle length is T, eq. (18) reduces to:

$$X_{ms}(x) = e^{(\lambda_u+\lambda_d)T(x-1)} e^{|\lambda_u-\lambda_d|T(x-1)} x^2 . \tag{19}$$

---

[1] We note that this formulation is consistent with eq. (1) in [23], which presents the PGF of the time to exhaust a master-bridge queue pair as a function of the duration in which packets accumulate in the master and the bridge and not as a function of the PGF of this duration. Moreover, it is easy to see that numerical results derived according to (14) regarding the cycle length and the time to exhaust a queue pair do not make sense.

As we understand, (19) is composed of 3 independent components. The first component is the PGF of the number of packets arriving according to a Poisson arrival process with arrival rate ($\lambda_u + \lambda_d$) during T slots. We suspect that this component was intended to represent the transmission time of uplink and downlink data packets. The second component is the PGF of the number of packets arriving during T slots, according to a Poisson arrival process with arrival rate $|\lambda_u - \lambda_d|$. We suspect that this component was intended to represent the POLL or NULL packets sent by the device that has the smaller number of data packets. The last component (x^2) represents the POLL-NULL transmission at the end of the exchange.

As argued in Section 5.2, it appears that there is no closed form expression for the PGF of the number of POLL or NULL packets sent by the device which has the smaller number of data packets (i.e. there is no closed form for $\max(U,D) - \min(U,D)$). Thus, it does not seem that this number follows the Poisson distribution as implied by (19). This can be demonstrated by assuming that the arrival rates are symmetrical ($\lambda_u = \lambda_d = \lambda$). Even in this case a few POLL or NULL packets will be sent during every master-slave channel service. However, (19) reduces to:

$$X_{ms}(x) = e^{2\lambda T(x-1)} x^2 , \tag{20}$$

and the mean time to serve both queues reduces to:

$$\overline{X_{ms}} = 2\lambda T + 2 . \tag{21}$$

Thus, *according to (20) and (21), in such a case, no POLL or NULL packets are sent during a master-slave channel exchange* (except for the last 2 packets). This is of course incorrect. Numerical results that demonstrate the difference between (21) and the results derived in Section 5.2 are presented in Section 5.4.

Although [20] claims to analyze the exhaustive regime, it seems that it does not take into consideration all possible complicated scenarios. For example, it is not clear whether the analysis considers the scenario in which at the beginning of the master-slave exchange the master has more data packets than the slave and during the slave transmission of NULL packets several data packets arrive at the slave's uplink queue. In such a scenario the slave will send a few NULL packets and the master will send a few POLL packets.

Similarly to the analysis of the limited regime, the rest of the analysis of the exhaustive regime in [20] directly uses the model of M/G/1 queue with vacations, without taking into consideration the dependencies between the queues. Thus, even if the PGF of the time to exhaust both queues derived in [20] (i.e. (18)) was correct[1], the rest of the analysis is incorrect. Since the dependencies between the queues have been discussed in detail in Section 4.3, we shall not elaborate on this issue. However, in Section 5.4, we compare numerical results obtained according to [20] for the case in which (18) seems to hold, with exact results obtained according to the analysis in Section 5.1. It will be shown that the use of the vacation model leads to incorrect results.

Finally, we note that inappropriate assumptions, similar to the ones indicated in this section, also appear in [18],[19],[21],[22],[23],[24],[25] and [26]. For example,

- In [18] the exhaustive scheduling algorithm is analyzed in a somewhat different methodology than the analysis described in [20]. However, the model of M/G/1 queue with vacations is directly used and results in problematic dependencies as discussed in Section 4.3.
- The analysis of the exhaustive scheduling algorithm in [19] and [22] is *very similar to* the analysis in [20].
- In [23] a scatternet composed of two piconets connected by a bridge which is a slave of the two masters is analyzed. The intra-piconet scheduling algorithm is exhaustive and its analysis is similar to the analysis in [20] (for example, eq. (3) in [23] is identical to eq. (7) in [20]). Furthermore, each master exchanges packets with the bridge according to an exhaustive regime. The PGF of the length of this exchange is derived in [23], eq. (1) and it does not take into consideration the complexities discussed in Section 5.2. Numerical results based on the analysis in [23] are presented in [24].
- In [26] the performance of piconets connected through a master/slave bridge and a slave/slave bridge is analyzed. In both cases, the intra-piconet scheduling algorithm is exhaustive (analyzed in a similar manner to [20]) and the master exchanges packets with the bridge according to an exhaustive regime (analyzed in a similar manner to [23]).

---

[1] It seems that the PGF of the time to exhaust both queues derived in [20] (i.e.(18)) holds only for unidirectional flows (e.g. $\lambda_u = \lambda$, $\lambda_d = 0$).

– In [21] the performance of scatternets composed of two piconets connected through a master/slave bridge and a slave/slave bridge is analyzed. In both cases, the master exchanges packets with the bridge according to an exhaustive regime. The performance of the two different scatternets is analyzed for exhaustive and limited intra-piconet scheduling algorithms. Although the intra-piconet exhaustive scheduling algorithm is analyzed in a somewhat different methodology than the analysis described in [20], the model of M/G/1 queue with vacations is still directly used. In Section 5.4, we will present numerical results that demonstrate the difference between the exact results and the results obtained according to [21].

5.4 Numerical Results

In this section we present exact numerical results computed according to the analysis in Section 5.1. Then, in order to demonstrate the difference between the results obtained in [20] and the exact results, obtained in Sections 4.2, 5.1, and 5.2, we compare numerical results calculated according to the analysis in [20] with numerical results based on our analysis. We show that in some cases [20] significantly underestimates the time to empty a master-slave queue pair. Moreover, we show that in very simple scenarios, the mean access delay obtained according to [20] considerably differs from the exact mean access delay.

In order to obtain $\overline{W}_a$ according to [20], we have used eq. (7) - (10) in [20], and computed $\overline{X}_c = X'_c(1)$, $\overline{V}_l = G'_{V_l}(1)$, $\overline{V_l^2} = G''_{V_l}(1) + G'_{V_l}(1)$.[1]

Figure 9 illustrates the exact mean access delay ($\overline{W}_a$), computed according to (10), in a *half-symmetrical* piconet with only uplink traffic (i.e. $\lambda_u = \lambda$, $\lambda_d = 0$). The figure presents the delay (in slots) as a function of the load in the unidirectional exhaustive system (defined in Section 5.1 as $(m-1)\lambda(\overline{L}+1)$) in piconets with various numbers of slaves in which the probabilities of 1, 3, and 5-slot packets are equal ($p_1 = p_3 = p_5 = 1/3$).

---

[1] As mentioned in Section 5.3, we assume that the PGFs in eq. (7) and (8) in [20] are functions of the mean cycle length ($\overline{X}_c$) and not of the PGF of the cycle length ($X_c(x)$)

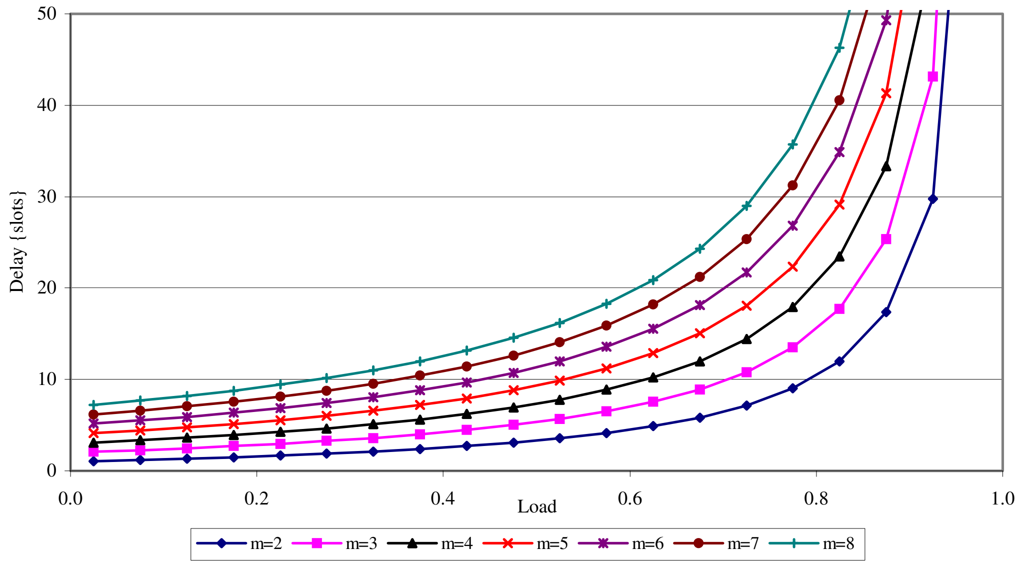

Figure 9. The exact mean access delay (computed according to (10)) in half-symmetrical piconets with only uplink traffic, operated according to the exhaustive scheduling algorithm, in which $p_1 = p_3 = p_5 = 1/3$

Under the assumptions made in Sections 5.2 and 5.3, in a symmetrical piconet the mean time to empty both queues according to [20] (i.e. $2\lambda T + 2$) differs from the mean time obtained in Section 5.2 (i.e. $2E(\max(U,D) + 2)$. In order to demonstrate this difference, we have obtained by simulation *average values* of $\max(U,D)$ for a cycle of length T. Figure 10 illustrates the ratio of the average value of $\max(U,D)$ to λT for different values of λT. Moreover, Figure 11 presents the ratio of the average time to exhaust a master-slave channel, obtained via simulation according to our analysis in Section 5.2, to the mean time to exhaust a channel, obtained according to [20]. The figure presents the ratio when packets are all 1, all 3, or all 5 slots long. The ratio is presented as a function of the mean number of packets arriving during a cycle (λT). It can be seen that the results obtained in [20] underestimate the time to empty both queues. We note that in every simulation and for every value of λT, we have obtained 300,000 different values of U and D in order to obtain the average value of $\max(U,D)$ and the average time to exhaust a channel.

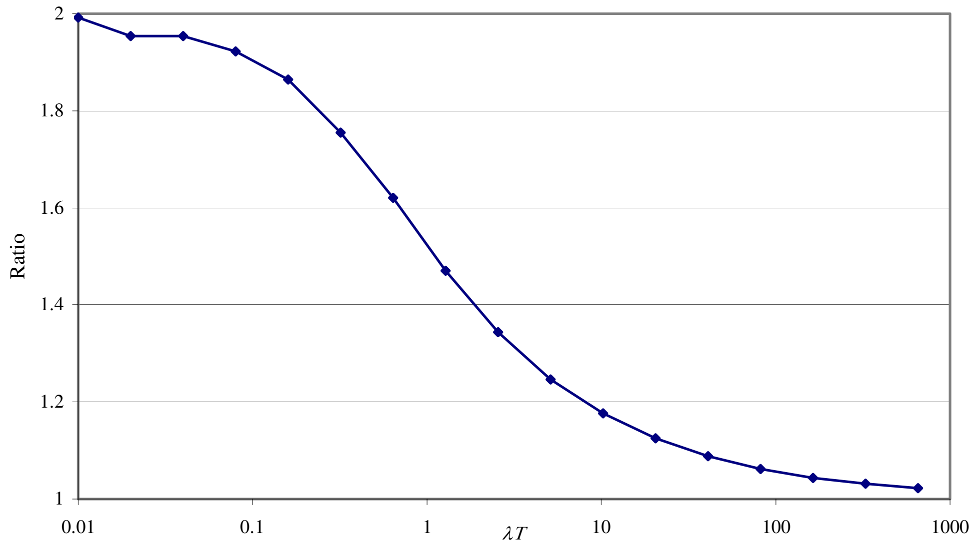

Figure 10. The ratio of the average value of max(U,D) to λT for different values of λT

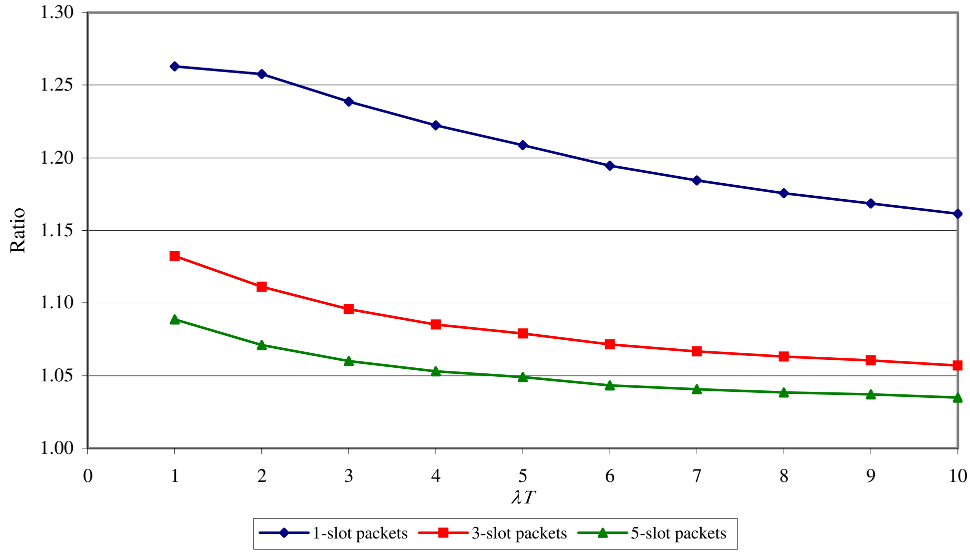

Figure 11. The ratio of the average time to exhaust a master-slave queue pair obtained via simulation according to our analysis in Section 5.2 to the mean time to exhaust a queue pair obtained according to [20] in piconets in which packets are all 1, all 3, or all 5 slots long

As mentioned before, when the piconet is composed of a single slave (m = 2) there is no difference between the limited regime and the exhaustive regime. Figure 12 presents the ratio of the mean access delay ($\overline{W}_a$) computed according to the analysis of the exhaustive regime in [20] to the mean access delay computed according to our limited model (i.e. according to (3)) in a symmetrical piconet with a single slave. The figure presents the ratio as a function of the load in the limited system (defined in Section 4.2 as $2(m-1)\lambda\overline{L}$) for 3 cases: (1) when all packets are 1 slot long (p_1 = 1, $\overline{L}=1$), (2) when the probabilities of 1, 3, and 5-slot packets are equal ($p_1 = p_3 = p_5$ = 1/3, $\overline{L}=3$), and (3) when all packets are 5 slots long (p_5 = 1, $\bar{L} = 5$). We note that even if one uses the results presented in [20] for a piconet with a single slave, the calculated mean access delay in the exhaustive regime is usually much higher than the calculated mean access delay in the limited regime. For example, when all the packets are 5 slots long and the arrival rates is λ = 0.0825 (packets/slot), the ratio between calculated mean access delay in the exhaustive and the limited regimes (both according to [20]) is 3.03.

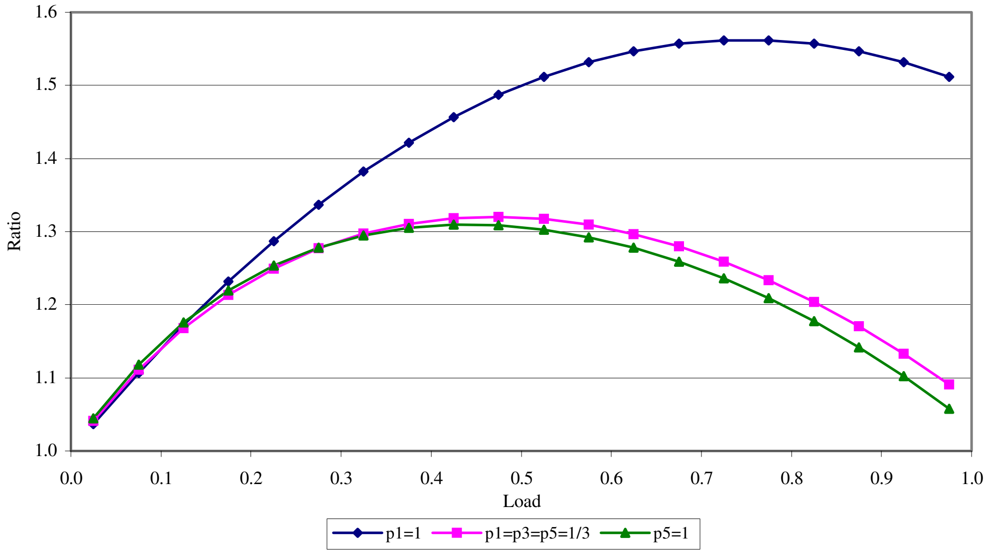

Figure 12. The ratio of the mean access delay derived according to the analysis of the exhaustive regime in [20] to the exact mean access delay (derived according to (3)) in piconets with one slave in which (1) all packets are 1 slot long, (2) the probabilities of 1, 3, and 5-slot packets are equal ($p_1 = p_3 = p_5 = 1/3$), and (3) all packets are 5 slots long

It seems that the PGF of the time to exhaust both queues derived in [20] (i.e.(18)) holds only for unidirectional flows (e.g. $\lambda_u = \lambda$, $\lambda_d = 0$). Figure 13, compares the mean access delay ($\overline{W}_a$) computed according to the analysis of the exhaustive regime in [20] when λ_d = 0 to the mean access delay computed according to our analysis of a unidirectional model (i.e. according to (10)).[1] The figure presents the delay (in slots) as a function of the load in the unidirectional exhaustive system (defined in Section 5.1 as $(m-1)\lambda(\bar{L}+1)$) in a piconet with 2 slaves ($m=3$). It can be seen that the results obtained in [20] underestimate the access delay.

[1] Eq. (10) in [20], which presents the access delay (as a function of λ), does not explicitly distinguish between the uplink and downlink flows (it assumes that $\lambda_u = \lambda_d$). Therefore, we have validated that it is identical to eq. (9) in [23] which explicitly separates the uplink and downlink flows (λ_{u1} and λ_{d1}) and used it to derive the access delay when $\lambda_d = 0$.

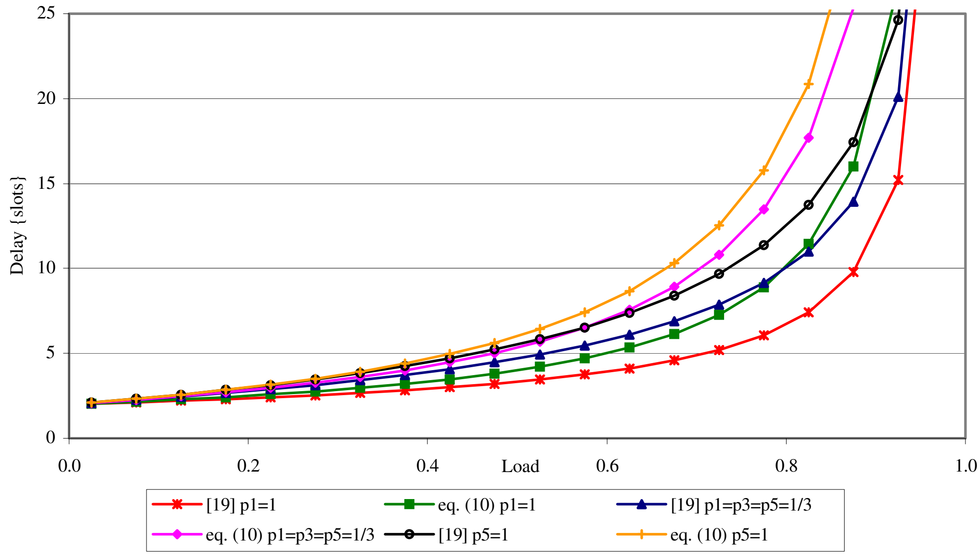

Figure 13. The mean access delay derived according to [20] and the exact mean access delay (derived according to (10)) in piconets with only uplink traffic operated in the exhaustive regime. The piconets are composed of 2 slaves and the packets are 1 slot long, 1, 3, and 5 slots long with equal probability ($p_1 = p_3 = p_5 = 1/3$), and 5 slots long

Finally, we note that in [21] the intra-piconet exhaustive scheduling algorithm is analyzed in a somewhat different methodology than in [20]. Thus, Figure 14 presents the ratio of the exact mean access delay to the mean access delay computed according to [21], in piconets with only unlink traffic ($\lambda_u = \lambda$, $\lambda_d = 0$) in which the probabilities of 1, 3, and 5-slot packets are equal.[1]

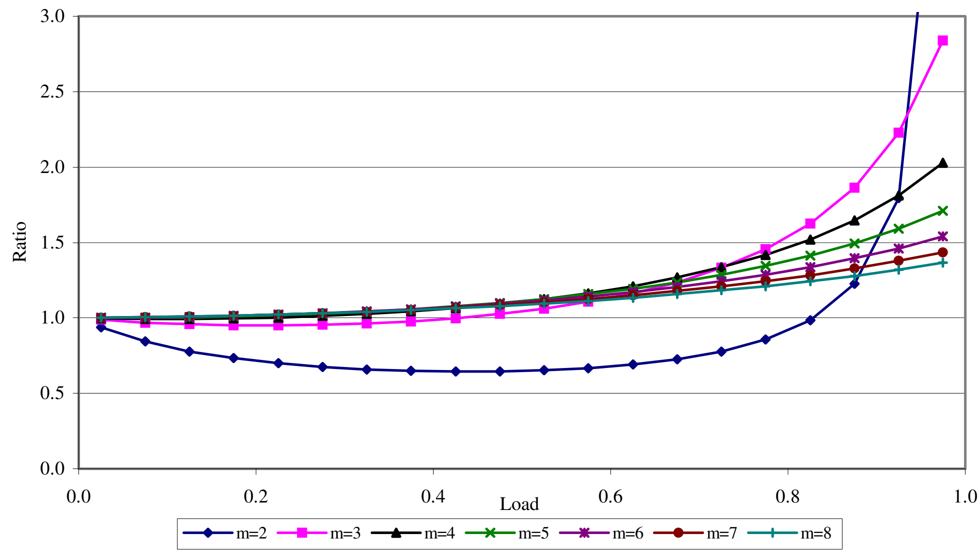

Figure 14. The ratio of the exact mean access delay (obtained according to (10)) to the mean access delay derived according to [21] in piconets with only uplink traffic operated in the exhaustive regime in which the probabilities of 1, 3, and 5-slot packets are equal ($p_1 = p_3 = p_5 = 1/3$)

---

[1] In order to compute results regarding *only* intra-piconet exhaustive scheduling according to [21], we have used eq. (22)-(27),(29),(31)-(33) in [21] and assumed that $G_r(x)$ (defined in eq. (21) in [21]) equals 1. Moreover, we have assumed that $\lambda_{u1} = \lambda$ and $\lambda_{d1} = 0$.

6 Conclusions

This work presents an analytical study of some versions of the limited (pure round robin) and the exhaustive scheduling algorithms in Bluetooth piconets, and examines the analytical study of these algorithms in [20]. We have modeled a piconet operated according to the *limited scheduling algorithm* as a 1-limited polling system and derived exact analytic results, which differ from the results obtained in [20]. Further, we have modeled the *exhaustive scheduling algorithm* in a piconet with unidirectional traffic as an exhaustive polling system. Then, the complexity of analyzing the gated scheduling regime in a piconet with bi-directional traffic has been described. Using these results, it has been shown that the results presented in [20] regarding the exhaustive scheduling algorithm are incorrect. Finally, we have provided numerical examples that illustrate the considerable difference between the exact results and those presented in [20] and argued that the results presented in [18],[19],[21],[22],[23],[24],[25], and [26] seem also to be incorrect.

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 Section 5.2), 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 would like to thank Nir Naaman and Daniele Miorandi for helpful discussions. We would like to thank Ronen Kofman and Carlo Caimi for obtaining simulation results.

References

[1] S. Baatz, M. Frank, C. Kühl, P. Martini, and C. Scholz, "Bluetooth Scatternets: An Enhanced Adaptive Scheduling Scheme", *Proc. IEEE INFOCOM'02*, June 2002.

[2] S. Basagni and C. Petrioli, "Degree-Constrained Multihop Scatternet Formation for Bluetooth Networks", *Proc. IEEE GLOBECOM'02*, Nov. 2002.

[3] D. P. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall Inc., New Jersey, 1992.

[4] Bluetooth Special Interest Group, *Specification of the Bluetooth System – Version 1.1*, Feb. 2001.

[5] J. Bray and C. Sturman, *Bluetooth 1.1 connect without cables*, Prentice Hall, 2001.

[6] R. Bruno, M. Conti, and E. Gregori, "Bluetooth: Architecture, Protocols and Scheduling Algorithms", *Cluster Computing*, Vol. 5, pp. 117-131, Apr. 2002.

[7] A. Capone, M. Gerla, and R. Kapoor, "Efficient Polling Schemes for Bluetooth Picocells", *Proc. IEEE ICC'01*, June 2001.

[8] C. F. Chiasserini, M. A. Marsan, E. Baralis, and P. Garza, "Towards Feasible Distributed Topology Formation Algorithms for Bluetooth-based WPANs", *Proc. HICSS-36*, Jan. 2003.

[9] R. B. Copper, " Queues Served in a Cyclic Order: Waiting Times", *Bell Systems Tech. J.*, Vol. 49, pp. 399-413, Mar. 1970.

[10] A. Das, A. Ghose, A. Razdan, H. Saran, and R. Shorey, "Enhancing Performance of Asynchronous Data Traffic over the Bluetooth Wireless Ad-hoc Network", *Proc. IEEE INFOCOM'01*, Apr. 2001.

[11] L. Har-Shai, R. Kofman, G. Zussman, and A. Segall, "Inter-Piconet Scheduling in Bluetooth Scatternets", *Proc. OPNETWORK 2002*, Aug. 2002.

[12] N. Johansson, U. Korner, and P. Johansson, "Performance Evaluation of Scheduling Algorithms for Bluetooth", *Proc. IFIP TC6 Int. Conf. on Broadband Communications*, Nov. 1999.

[13] P. Johansson, R. Kapoor, M. Kazantzidis, and M. Gerla, "Rendezvous Scheduling in Bluetooth Scatternets", *Proc. IEEE ICC'02*, Apr. 2002.

[14] H. Levy and M. Sidi, "Polling Systems: Application, Modeling, and Optimization", *IEEE Trans. on Comm.*, Vol. 38, pp. 1750-1760, Oct. 1990.

[15] Y. Levy and U. Yechiali, "Utilization of the Idle Time in an M/G/1 Queueing System", *Management Science*, Vol. 22, pp. 202-211, Oct. 1975.

[16] X. Y. Li, I. Stojmenovic, and Y. Wang, "Partial Delaunay Triangulation and Degree Limited Localized Bluetooth Multihop Scatternet Formation", *Submitted for publication* (available from: http://www.site.uottawa.ca/~ivan/wireless.html), 2002.

[17] D. Miorandi, C. Caimi, and A. Zanella, "Performance Characterization of a Bluetooth Piconet with Multi-Slot Packets", *To appear in Proc. WiOpt'03*, Mar. 2003.

[18] J. Misic and V.B. Misic, "Queueing Analysis of Bluetooth Piconet Traffic", *Proc. SPECTS'02*, July 2002.

[19] J. Misic and V.B. Misic, "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: M. H. Shafazand and A M. Tjoa)*, Springer-Verlag, pp. 891-898, Oct. 2002.

[20] J. Misic and V.B. Misic, "Modeling Bluetooth Piconet Performance", *IEEE Communications Letters*, Vol. 7, pp. 18-20, Jan. 2003.

[21] J. Misic and V.B. Misic, "Bridges of Bluetooth County: Topologies, Scheduling, and Performance", *IEEE Journal on Selected Areas in Communications*, Vol. 21, pp. 240-258, Feb. 2003.

[22] V. B. Misic and J. Misic, "On Bluetooth Piconet Traffic Performance", *Proc. IEEE PIMRC'02*, Sep. 2002.

[23] V. B. Misic and J. Misic, "Minimizing End-to-End Delays in Bluetooth Scatternet with a Slave/Slave Bridge", *Proc. ICCCN'02*, Oct. 2002.

[24] V. B. Misic and J. Misic, "Queueing Analysis of Delays in Bluetooth Slave/Slave Bridge", *Proc. IEEE/ACM MASCOTS'02*, Oct. 2002.

[25] V. B. Misic and J. Misic, "Bluetooth Scatternet with a Master/Slave Bridge: a Queueing Theoretic Analysis", *Proc. IEEE GLOBECOM'02*, Nov. 2002.

[26] V. B. Misic and J. Misic, "Performance of Bluetooth Bridges in Scatternets with Exhaustive Service Scheduling", *Proc. HICSS-36*, Jan. 2003.

[27] A. Racz, G. Miklos, F. Kubinszky, and A. Valko, "A Pseudo Random Coordinated Scheduling Algorithm for Bluetooth Scatternets", *Proc. ACM MOBIHOC'01*, Oct. 2001.

[28] M. Sidi, H. Levy, and S. Fuhrmann, "A Queueing Network with a Single Cyclically Roving Server", Queueing Systems Theory and Applications, Vol. 11, pp. 121-144, 1992.

[29] C. E. Skinner, "A Priority Queueing System with Server-Walking Time", *Operations Research*, Vol. 15, No. 2, pp. 278-285, 1967.

[30] H. Takagi, *Analysis of Polling Systems*, MIT Press, 1986.

[31] H. Takagi, *Queueing Analysis: a Foundation of Performance Evaluation, Vol. 1 – Vacation and Priority Systems*, North-Holland, 1991.

[32] U. Yechiali, "Analysis and Control of Polling Systems", in *Performance Evaluation of Computer and Communication Systems (eds: L. Donatiello and R. Nelson)*, Springer-Verlag, pp. 630-650, 1993.

[33] G. Zussman and A. Segall, "Capacity Assignment in Bluetooth Scatternets – Analysis and Algorithms", *Proc. IFIP-TC6 Networking 2002*, *Lecture Notes in Computer Science, Vol. 2345 (eds: E. Gregori et al.)*, Springer-Verlag, pp. 411-422, May 2002.

[34] G. Zussman and A. Segall, "Capacity Assignment in Bluetooth Scatternets – Optimal and Heuristic Algorithms", *To appear in ACM/Kluwer Mobile Networks and Applications (MONET)*, Special Issue on Advances in Research of WPAN and Bluetooth Enabled Networks, 2003.