

On the Analysis of the Bluetooth Time Division Duplex Mechanism

Gil Zussman[†], Adrian Segall^{*} and Uri Yechiali[§]

Abstract

Efficient communication in Bluetooth networks requires design of intra and inter-piconet scheduling algorithms, and therefore numerous algorithms have been proposed. However, due to complexities of the Bluetooth MAC, the performance of these algorithms has been analyzed mostly via simulation. We present analytic results regarding the exhaustive, gated, and limited (pure round robin) scheduling algorithms in piconets with bidirectional and unidirectional traffic. We show that a piconet operated according to the limited scheduling algorithm is equivalent to a 1-limited polling system and present exact results regarding symmetric piconets with bidirectional traffic. Then, the difficulties in analyzing the performance of the exhaustive and gated algorithms in a piconet with bidirectional traffic are demonstrated. In addition, we present exact analytic results for piconets with unidirectional traffic. We show that, surprisingly, in symmetrical piconets with only uplink traffic, the mean waiting time is the same for the exhaustive and limited algorithms. This observation results from the differences between piconets and traditional polling systems and can be extended for Time-Division-Duplex systems with arbitrary packet lengths. Furthermore, we show that the mean waiting time in a piconet with only uplink traffic is significantly higher than its corresponding value in a piconet with only downlink traffic. Finally, we numerically compare the exact results to approximate results, presented in the past.

Index Terms

Personal Area Networks, Time Division Duplex, Medium Access Control, Scheduling, Polling, Queueing.

I. INTRODUCTION

Bluetooth is a Personal Area Network (PAN) technology, which enables devices to connect and communicate wirelessly via short-range ad-hoc networks [4]. 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*. A device that is a member of more than one piconet (referred to as a *bridge*) must schedule its presence in all the piconets in which it is a member.

The master uses *intra-piconet* scheduling algorithms to schedule the traffic within a piconet. *Inter-piconet* scheduling algorithms are used to schedule the presence of the bridges in different piconets. Numerous intra and inter-piconet scheduling algorithms have been proposed (e.g. [2],[6],[8],[10],[11],[28]).

Preliminary and partial versions of this paper have been presented in the 1st IEEE Conference on Sensor and Ad Hoc Communications and Networks (*SECON'04*) [31] and in the 8th IFIP-TC6 Conference on Personal Wireless Communications (*PWC'03*) [30].

[†]Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139, gilz@mit.edu.

^{*}Dept. of Electrical Engineering, Technion–Israel Institute of Technology, Haifa 32000, Israel, segall@ee.technion.ac.il.

[§]Dept. of Statistics and Operations Research, School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel, uriy@post.tau.ac.il.

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 [6], due to the special characteristics of the Bluetooth Medium Access Control (MAC) which is based on Time-Division-Duplex (TDD), the performance of these algorithms has been analyzed mostly via simulation. *In this paper we present overlooked connections between Bluetooth piconets and polling systems¹. We show that these connections can be directly utilized in order to obtain analytic results regarding the performance of the algorithms.* We present results regarding 3 different traffic patterns: bidirectional traffic, unidirectional uplink traffic, and unidirectional downlink traffic.

First, we focus on the *limited (pure round robin) scheduling* algorithm in a piconet with *bidirectional* traffic and show that a piconet operated according to this algorithm is equivalent to a *1-limited polling system*. The problem of computing exact mean delays in general 1-limited polling systems has not been resolved yet [13], but we derive exact analytic results regarding intra-piconet waiting times in a symmetric piconet with (the most commonly assumed) Poisson arrival process. However, considering the exhaustive and gated scheduling algorithms, we argue that when the traffic is bidirectional, it seems that there is no closed form expression for the Probability Generating Function of the time (in slots) to exhaust the two related queues, at a given slave and at the master. Therefore, it seems that there is no closed form expression for the waiting time in a piconet operated according to these algorithms.

We show that a piconet with *unidirectional uplink* traffic operated according to the *exhaustive* scheduling algorithm is equivalent to a specific *exhaustive polling system* and derive exact results regarding intra-piconet waiting times. We also show that a piconet with unidirectional uplink traffic operated according to the *limited (pure round robin) scheduling* algorithm can be modeled as a *1-limited polling system*.

Following this analysis, a surprising result is obtained: the mean waiting times for the limited and exhaustive algorithms are equal in a piconet with only uplink traffic, where all arrival rates are statistically equal. Moreover, when such a piconet is operated according to the gated algorithm, the mean waiting time is higher than in the exhaustive or limited algorithms.²

We extend these observations by considering a general *per-packet polling system*, which is a general-

¹A polling system consists of several queues served by a single server according to a set of rules (polling scheme) [3, p. 195],[13],[26].

²These observations may not necessarily hold when the arrival rates are *not* statistically equal.

ization of a piconet with only uplink traffic. In such a system the server (master) incurs a polling overhead for each packet and some overhead is incurred when a station announces to the server that it is empty. This system differs from the traditional polling system in which a switchover period is encountered only when the server (master) shifts from one queue to another. We show that the observations regarding the piconet also hold for the per-packet polling system. An important specific case of the per-packet polling system is an arbitrary Time-Division-Duplex system, where the packets are not necessarily 1, 3, and 5 slots long (as required in Bluetooth [4]). Such a system is important, since the TDD mechanism will be used by other technologies, such as 3.5G and 4G cellular systems [9].

Furthermore, we show that a piconet with only *unidirectional downlink* traffic operated according to the exhaustive scheduling algorithm is equivalent to an exhaustive polling system with *zero-switchover* periods. A similar equivalence holds for the gated and the limited algorithms. It is shown that the mean waiting time in a piconet with only uplink traffic is significantly higher than in a piconet with only downlink traffic. Finally, although the analysis of the gated and exhaustive algorithms in piconets with bidirectional traffic seems infeasible, we show that the case in which some of the traffic is only in the downlink and some of it is only in the uplink can be analyzed.

We note that various aspects related to the performance evaluation of Bluetooth scheduling algorithms have been recently studied. For example, [22] presents results regarding admission control in Bluetooth piconets, the performance of TCP over Bluetooth, and the performance of inter-piconet scheduling algorithms. In addition, [16],[20],[21], and [23] recently presented approximate results regarding the performance of various intra and inter-piconet scheduling regimes. In Section VIII we numerically compare the exact results to those presented in [20] and [21].

The main contributions of this paper are two-fold: (i) exact results regarding the delay in Bluetooth piconets are derived, and (ii) counterintuitive observation regarding the delay in per-packet polling systems is provided. To the best of our knowledge, the results presented in this paper are the only available *exact* analytic results regarding the delay in piconets.³ The results regarding the limited algorithm, which have been presented in a preliminary version [30], have been recently extended by Miorandi and Zanella for

³We note that analytical results regarding the throughput of Bluetooth piconets have been presented in [24].

an asymmetric arrival process [18] and for fading channels [17]. Similarly, we argue that the rest of the results presented in this paper, regarding specific scenarios (e.g. Poisson arrival process), can be easily extended to different scenarios (e.g. batch arrivals) by utilizing the vast amount of research regarding polling systems.

This paper is organized as follows. Section II gives a brief introduction to the Bluetooth technology, while Section III presents the model. In Section IV we analyze the scheduling algorithms in a piconet with bidirectional traffic. Section V analyzes the algorithms in piconets and general per-packet polling systems with unidirectional uplink traffic. In Section VI we analyze piconets with unidirectional downlink traffic. Section VII briefly discusses the case in which the traffic is either in the uplink or in the downlink. Section VIII presents numerical results and Section IX summarizes the main results.

II. 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- μ sec 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).⁴ Information can only be exchanged between a master and a slave.

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, if it has no data to send, a 1-slot *POLL packet*. The slave must respond by sending a data packet or, if it has nothing to send, a 1-slot *NULL packet*. 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. 1.

The master schedules the traffic in a piconet according to an intra-piconet scheduling algorithm. We focus on the following algorithms in which the master communicates with the slaves according to a *fixed cyclic order*: (i) the *Limited (Pure) Round Robin* algorithm in which at most a single packet is sent in each

⁴We concentrate on networks in which only data links are used.

direction (downlink or uplink) whenever a master-slave queue pair is served, (ii) the *Exhaustive Round Robin* algorithm in which the master does not switch to the next master-slave queue pair until both the downlink and the uplink queues are empty, and (iii) the *Gated Round Robin* algorithm in which only the packets that are found in the uplink and downlink queues when the master starts serving the queue pair are transmitted.

III. THE MODEL

The number of slaves is denoted by N and we assume that each node has an infinite buffer. We assume that the packets are generated at the uplink and downlink queues according to independent Poisson arrival processes. We denote the arrival rate to the uplink queue at slave i by λ_u^i and the downlink arrival rate into the master of packets intended for slave i by λ_d^i . We note that in general, arrival processes in real networks are not Poisson but rather compound Poisson (i.e. *batches* of packets arrive according to a Poisson arrival process). Therefore, throughout the paper we briefly point out the additional steps that have to be taken in order to analyze systems with a compound Poisson arrival process.

We mostly consider 3 different traffic patterns: bidirectional traffic (i.e. there are links on which the traffic is bidirectional), unidirectional uplink traffic, and unidirectional downlink traffic. We will also consider the special case in which there is no bidirectional traffic on any link but on each link the traffic is either solely downlink or solely uplink. For each traffic pattern, we consider one of the following packet generation scenarios:

- *Symmetric piconet* - The arrival rate to every *active* queue is λ (packets/slot). We refer to active queues since, in piconets with unidirectional downlink (uplink) traffic, the uplink (downlink) queues are inactive.
- *Asymmetric piconet* - The arrival rates to the queues are not necessarily the same (λ_u^i and λ_d^i can also be 0 for some queues).

We assume that the master is the final destination of all packets generated at the slaves. The probabilities of a packet length being 1, 3, or 5 slots are p_1 , p_3 , and p_5 , respectively.⁵ The mean and second moment

⁵Although we assume that the packet lengths are randomly selected, in practice, these lengths depend on the Segmentation and Reassembly (SAR) of higher layer packets (see the discussion in [8]). The SAR policy can also affect the arrival process (i.e. in practice, it is likely that batches of packets will arrive simultaneously).

of the packet length are denoted by \bar{L} and \bar{L}^2 . The *waiting time* is the time a packet waits in the uplink or the downlink queue before it is served. The mean waiting times (in slots) in the uplink queues under the limited, exhaustive, and gated regimes are denoted by \bar{W}_u^{Lim} , \bar{W}_u^{Ex} , and \bar{W}_u^{G} , respectively. Similarly, the mean waiting times in the downlink queues under the different regimes are denoted by \bar{W}_d^{Lim} , \bar{W}_d^{Ex} , and \bar{W}_d^{G} . When we present results for an asymmetric piconet operated according to the limited algorithm, the mean waiting time in the uplink queue of slave i is denoted by $\bar{W}_u^{\text{Lim}}(i)$ and the mean waiting time at the master of packets intended to slave i is denoted by $\bar{W}_d^{\text{Lim}}(i)$.

Some of the scheduling algorithms proposed in the past (e.g. [8]) assume that the master has some information regarding the status of the slaves' queues. However, obtaining such information requires changing the Bluetooth specifications [4] or using a proprietary algorithm in all the devices participating in a piconet. Thus, we assume that the master does not have any information about the state of the uplink queues. This assumption complies with the assumptions made in several previous analyses of intra-piconet scheduling algorithms (e.g. [6],[16],[19],[20],[21],[23]).

We note that whenever we refer to results regarding general (non-Bluetooth) symmetric polling systems we follow the notation of Takagi [26]. Namely, the mean and second moment of the packet service times are denoted by b and $b^{(2)}$. The mean and variance of the switchover times are denoted by r and δ^2 .

IV. BIDIRECTIONAL TRAFFIC

In this section we consider the case in which the traffic on the links is bidirectional. The case in which the traffic on each link is either solely downlink or solely uplink will be treated in Section VII.

A. Analysis of the Limited Algorithm

We first discuss a piconet in which *all packets are 1 slot long*. We then analyze the case in which *all packet types are present*. The former piconet type is discussed mainly because it is a special case of the latter. In a symmetric piconet, in which all packets are 1 slot long (i.e. $p_1 = 1$), a single slot is allocated to each downlink and uplink in every cycle. Therefore, the piconet can be modeled as a TDMA system [3, p. 194] with a cycle length of $2N$ slots. Every slot in the cycle is allocated to one of the $2N$ downlinks

and uplinks. The computation of the delay in a TDMA system is based on the analogy with the M/D/1 with vacations queue in which the deterministic service times and vacation lengths are equal to $2N$ [3, p. 194]. Accordingly, applying [3] eq. (3.58), where the number of queues is $2N$ and the total arrival rate is $2N\lambda$, we obtain the mean waiting time in the uplink and downlink queues (in slots):

$$\overline{W}_u^{\text{Lim}} = \overline{W}_d^{\text{Lim}} = \frac{N}{1 - 2N\lambda}. \quad (1)$$

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 asymmetric piconet. In this case every link can be independently analyzed as an M/D/1 queue. Therefore, the waiting time in the uplink queue of slave i (in slots) is again derived from eq. (3.58) in [3]:

$$\overline{W}_u^{\text{Lim}}(i) = \frac{2N\lambda_u^i}{2(1 - 2N\lambda_u^i)}2N + \frac{1}{2}2N = \frac{N}{1 - 2N\lambda_u^i}. \quad (2)$$

A similar equation describes the mean waiting time of packets arriving at the master and intended to slave i ($\overline{W}_d^{\text{Lim}}(i)$). In this case $\overline{W}_d^{\text{Lim}}(i)$ replaces $\overline{W}_u^{\text{Lim}}(i)$ and λ_d^i replaces λ_u^i .

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*⁶ with $2N$ queues.

In a piconet operated according to the limited scheduling algorithm, 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). 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 system*⁷ with the following characteristics: (i) there are $2N$ customers and a single server, (ii) the server serves the customers in a fixed cyclic order, (iii) the server incurs a switchover time of 1 slot when it shifts from one customer to another, (iv) when the server

⁶In a 1-limited polling system, at each visit of the server to a queue only the first packet (if any) in the queue is served. The server incurs a switchover time when it shifts from one queue to another [3, p. 201],[13],[26].

⁷It 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.

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 (v) 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. the probability of the service time in the polling system being $(k - 1)$ slots long is p_k), the delay (time until the end of service) in both systems is equal. This property is demonstrated in Fig. 2, 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. One 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.

In order to obtain the waiting time 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 waiting time in a Bluetooth piconet. This results from the fact that when Bluetooth data packets are sent, some of the data is actually sent during the “switchover” time, as it is defined in the equivalent polling system.

We now focus on symmetric systems in which the arrival rates to all queues are equal. By applying the model for a symmetric limited gated polling system described in [3, 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 [3] is defined as the time a packet waits until its service starts. Thus, in order to obtain the mean waiting time in a piconet, one has to deduct 1 slot from the expression for the waiting time in [3], eq. (3.77). Accordingly, we apply [3] eq. (3.77), where the number of queues is $2N$, the total arrival rate is $2N\lambda$, the switchover time is one slot with zero variance, the traffic intensity

is $\rho = 2N\lambda(\bar{L} - 1)$, and the second moment of the service time (denoted in [3] as $\overline{X^2}$) is $4p_3 + 16p_5$. Deducting 1 time unit (i.e. 1 slot), we obtain the mean waiting time in the uplink and downlink queues:

$$\overline{W}_u^{\text{Lim}} = \overline{W}_d^{\text{Lim}} = \frac{1 + N \left[1 + 2\lambda(p_3 + 6p_5 - 1) \right]}{1 - 2N\lambda\bar{L}} - 1. \quad (3)$$

Notice that in this system it must hold that $2N\lambda\bar{L} < 1$. We shall refer to $2N\lambda\bar{L}$ as the load in the bidirectional limited system.

As a special case, consider a piconet in which only 1-slot packets are used (i.e. $p_3 = p_5 = 0$). For such a piconet, (3) reduces to (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 [10] and [16]). For example, Fig. 3 compares the exact mean waiting time (\overline{W}_u) (computed according to (3)) to the average waiting time computed by simulation⁸ 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 [10] or after 48,000 to 2,400,000 slots (depending on the load) using the model presented in [16].

B. The Gated and Exhaustive Algorithms

Analyzing the performance of scheduling regimes such as the gated and exhaustive in a piconet with bidirectional traffic requires obtaining the PGF of the exchange time (in slots) 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 algorithm, we discuss a less complicated case, namely a single master-slave channel in a piconet, operated according to the *gated algorithm*.

In the gated algorithm, 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

⁸The simulation results computed by the model presented in [16] have been obtained by Carlo Caimi from the University of Padova.

sends some POLL packets in order to allow the slave to reply with data packets. We assume that at the end of the master-slave exchange, the slave has to respond with a NULL packet to a POLL packet.

Let X_G denote the total time (number of slots) required for the exchange duration of a single master-slave channel in the gated algorithm. The PGF and the mean of X_G are denoted by $X_G(x)$ and $\overline{X_G}$. For simplicity, we assume that *all packets are 1 slot long* ($p_1 = 1$) and that packets have 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 ($U, D \sim \text{Poisson}(\lambda T)$).

Thus, given that $p_1 = 1$, X_G equals twice the maximum of U and D plus 2 slots (the last POLL-NULL exchange). 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: $X_G(x) = x^2 \sum_{m=0}^{\infty} x^{2m} \text{Prob}(\max[U, D] = m)$, where

$$\text{Prob}(\max[U, D] = m) = 2 e^{-\lambda T} \frac{(\lambda T)^m}{m!} \left(\sum_{j=0}^{m-1} e^{-\lambda T} \frac{(\lambda T)^j}{j!} \right) + \left(e^{-\lambda T} \frac{(\lambda T)^m}{m!} \right)^2. \quad (4)$$

Unfortunately, it appears that in view of (4) there is no closed form expression for $X_G(x)$ and consequently, it seems that there is no closed form expression for the waiting time in a piconet with bidirectional traffic operated according to the gated algorithm. It is clear that the exact analysis of the exhaustive algorithm is more involved.

The mean time to serve a single master-slave channel is given by: $\overline{X_G} = 2E(\max[U, D]) + 2$. In order to bound the value of $\overline{X_G}$, we observe that for a given $\lambda > 0$ and $U, D \sim \text{Poisson}(\lambda)$: $1 < E(\max[U, D])/\lambda < 2$. To illustrate the behavior of this ratio, we have randomly generated 300,000 different values of U and D (for 17 various values of λ) and computed the average value of $\max[U, D]$ and its ratio to λ . The results are depicted in Fig. 4.

Using these results, we shall now provide a simple explanation for an observation made via simulation in [6] and [10]. According to [6] and [10], in piconets with bidirectional traffic and high loads, the waiting time under the limited algorithm is lower than under the exhaustive algorithm. Consider a symmetric piconet operated according to the exhaustive or gated algorithm, with only 1-slot packets. According to the above analysis, for an arrival rate of λ (packets/slot), a node will have to transmit on average

$E(\max[U, D])$ packets per slot (where $U, D \sim \text{Poisson}(\lambda)$). Thus, the arrival rate λ should be set such that $2NE(\max[U, D]) < 1$. Fig. 4 exhibits the fact that $E(\max[U, D])$ can approach 2λ . Hence, a necessary condition for stability is $\lambda < 1/(\alpha N)$, where $2 < \alpha < 4$. On the other hand, according to (1), when the same piconet is operated according to the limited algorithm, a necessary condition for stability is $\lambda < 1/(2N)$. When λ approaches the stability limit, the waiting time approaches infinity. Thus, in a piconet using the exhaustive or gated algorithm, the waiting time approaches infinity for lower values of λ than in a piconet using the limited algorithm. Therefore, for high values of load the waiting time in the limited piconet will be lower than in the exhaustive or gated piconet.

V. UPLINK TRAFFIC

A. Analysis of the Exhaustive Algorithm

We first analyze a *symmetric* piconet with only uplink traffic (i.e. a piconet in which, for all i : $\lambda_u^i = \lambda > 0$ and $\lambda_d^i = 0$). We will then argue that an *asymmetric* piconet (in which the arrival rates to the uplink queues are not necessarily equal) can be analyzed in a similar manner. Since $\lambda_d^i = 0 \forall i$, when the master communicates with a particular slave it sends only POLL packets (one for each data packet). The slave replies with data packets until its queue is empty. Then, it sends a NULL packet which signals the end of the exhaustive communication with that particular slave⁹.

In order to model the piconet as an exhaustive polling system, we define the service time of a k -slot data packet as $(k+1)$ slots which are composed of the k slots of data, augmented by the *following* POLL packet. 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 *symmetric* piconet, we apply the model for a symmetric discrete-time exhaustive polling system described in [26, p. 68]. Accordingly, we apply eq. (3.63b) in [26], where the number of queues is N , the arrival process is Poisson with intensity λ , the switchover time is two slots ($r = 2$) with zero variance ($\delta^2 = 0$), the mean service time is $b = \bar{L} + 1$, and the second moment of the service time is

⁹The termination of the master-slave exchange with a POLL-NULL exchange results from the fact that the master has no information about the slaves' queues and complies with the assumptions made in previous analyses of the exhaustive algorithm (e.g. [19],[20],[21]). Below, we will briefly discuss an alternative system in which this POLL-NULL exchange is not required.

$b^{(2)} = 4p_1 + 16p_3 + 36p_5$. By adding 0.5 slot, we obtain the mean waiting time¹⁰:

$$\overline{W}_u^{\text{Ex}} = \frac{N \left[1 + 4\lambda(p_3 + 3p_5) \right]}{1 - N\lambda(\overline{L} + 1)}. \quad (5)$$

We shall refer to $N\lambda(\overline{L} + 1)$ as the load in the uplink exhaustive system.

In a piconet with a single slave ($N = 1$) there is no difference between the exhaustive and the limited scheduling algorithms. As a special case, consider a piconet with unidirectional traffic of 1-slot packets (i.e. $p_1 = 1$) operated according to the limited algorithm. Its mean waiting time is given in (2). It readily follows that for such a piconet ($\lambda_u^i = \lambda$, $N = 1$, and $p_1 = 1$) (5) and (2) coincide.

The result presented in (5) was also verified by a simulation model based on OPNET (described in [10]). For example, Fig. 5 compares the exact mean waiting time to the computed (by simulation) average waiting time, in piconets with 7 slaves in which (i) all packets are 1-slot long and (ii) $p_1 = 0.2$, $p_3 = 0.6$, and $p_5 = 0.2$. For each load value, the results have been computed after 230,000 slots.

An *asymmetric* piconet with only *uplink traffic* (i.e. $\lambda_d^i = 0 \forall i$, and $\lambda_u^i > 0 \forall i$, not all necessarily equal) can be analyzed in a similar manner to a symmetric piconet. Namely, it can be modeled as an asymmetric exhaustive *polling* system composed of N queues, with 2-slot switchover time and with service time of $(k + 1)$ slots for a k -slot data packet. Accordingly, the mean waiting time in each uplink queue can be obtained by any of the methods for analyzing exhaustive polling systems described in [26] and [27]. Since some of these methods require solving $O(N^2)$ equations and since $N \leq 7$, 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 emphasize that once a piconet has been modeled as a polling system, a piconet with batch arrivals can be analyzed by *directly applying* the methods for the analysis of exhaustive polling systems with a compound Poisson arrival process (see for example [26]). Moreover, results can be obtained regarding even more general arrival processes (such as correlated arrivals to the different queues [14]) by directly applying the methods for analyzing exhaustive polling systems with such arrival processes.

¹⁰We add 0.5 slot, since we are interested in the waiting period from the time of arrival, whereas in [26] the waiting time is counted from the end of the slot in which a packet arrives. We note that using eq. (3.69) in [3, p. 200] with the same parameters does not require adding 0.5 slot and yields the same result.

We note that one could devise a more efficient mechanism in which the last POLL-NULL exchange can be avoided. For example, an appropriate message signalling that the slave's queue is empty can be piggybacked in the trailer of last data packet sent by a slave. Of course, implementing such a mechanism may require slight modifications to the Bluetooth specifications. Yet, it would be interesting to understand the piconet performance under such a mechanism. Such a modification will reduce the number of POLL-NULL exchanges and will probably reduce the waiting time incurred under the exhaustive regime. However, even under the modified regime, a POLL-NULL exchange will take place when the master contacts a slave with an empty queue. On the contrary, when it contacts a slave with a non-empty queue, only extended service (that includes a poll packet for each data packet) will take place.

Such an improved exhaustive scheduling algorithm can be viewed as an exhaustive polling system with zero-switchover periods in which, once the server switches into an empty queue, a penalty (e.g. setup time) of two slots is paid. This polling system has some similarities to State-Dependent (SD) polling systems in which the server performs setup at a polled queue only if it is *not* empty. Most of the previous work regarding SD polling systems provided approximate analysis (e.g. [1]), whereas recent work provided some exact results (e.g. [7],[25]). Since the considered polling system differs from SD polling systems, analyzing the performance of such a general (not necessarily Bluetooth) system is an interesting future research direction.

B. Analysis of the Gated Algorithm

A piconet with only *uplink* traffic operated according to the *gated* algorithm is similar to a piconet operated according to the exhaustive algorithm. The main difference is that a slave replies to the master *only* with the data packets that were present in the uplink queue at the moment it received the first POLL packet from the master. In order to signal the end of the gated communication, the slave sends a NULL packet. Since we assume that the master and the slave do not exchange queue status information, the last POLL-NULL exchange is required. Yet, similarly to the discussion above, we note that by slightly modifying the protocol this exchange could be avoided. This will probably result in reduced waiting times.

The gated algorithm can be modeled as a gated *polling* system in a similar manner to the modeling of the exhaustive algorithm. Namely, we define the service time of a k -slot data packet as $(k + 1)$ slots,

and the switchover time as 2 slots. For a *symmetric* piconet, we apply the model for a symmetric gated polling system described in [26, p. 104] and use eq. (5.23) in [26] to obtain the mean waiting time:

$$\overline{W}_u^G = \frac{N \left[1 + 4\lambda(p_3 + 3p_5) \right]}{1 - N\lambda(\overline{L} + 1)} + \frac{2N\lambda(\overline{L} + 1)}{1 - N\lambda(\overline{L} + 1)}. \quad (6)$$

An *asymmetric* piconet with only *uplink traffic* can be similarly analyzed by one of the methods described in [26] and [27].

C. Analysis of the Limited Algorithm

We now show that a piconet with only *uplink traffic* operated according to the *limited* (pure round robin) scheduling algorithm can be modeled as a 1-limited polling system. In such a piconet the master continuously sends POLL packets to the slaves. Even if the slave has nothing to send, one slot must be used during the uplink communication (by the NULL packet).

At first glance, it seems that such a piconet can be modeled as a *1-limited polling system with N queues* in a similar manner to the modeling of a piconet with bidirectional traffic as a *1-limited polling system with 2N queues* (see Section IV-A). Namely, the switchover time can be defined as 2 slots that are composed of the POLL packet and the first slot of the data packet or of the POLL-NULL exchange. As a result, the service time of a k -slot packet should be defined as $(k - 1)$ slots. According to this modeling a packet that arrives to an empty uplink queue during the first half of the switchover time (i.e. during the transmission of the POLL packet) will be served immediately after the switchover time. On the other hand, a packet that arrives to an empty uplink queue during the second half of the switchover time (i.e. during the transmission of the NULL packet) will not be served immediately (as opposed to the situation in a 1-limited polling system). Thus, an alternative modeling is required.

We define the beginning of the switchover to a queue as the instance in which the *preceding* slave starts transmitting the last slot of a data packet or a NULL packet. A switchover ends when the master completes the transmission of the POLL packet *intended to the slave* (if at the end of the switchover the queue is empty, the switchover to the next queue is immediately started). We define the switchover time to each of the queues as 2 slots. If the preceding slave sends a 3 or 5-slot data packet, the 2 switchover

slots are composed of the last slot of the packet and the following POLL packet. Alternatively, if the preceding slave sends a 1-slot data packet or a NULL packet, the 2-slot switchover time is composed of the packet sent by the preceding slave and the following POLL packet.

Consequently, when data packets are sent, some of the data is actually sent during the “switchover” to the next queue. Therefore, the service time of a k -slot data packet is defined as $(k - 1)$ slots. Fig. 6 illustrates an example of the operation of a piconet and of the equivalent polling system.

We now focus on symmetric systems with only uplink traffic (i.e. $\lambda_u^i = \lambda > 0 \forall i$ and $\lambda_d^i = 0 \forall i$) and apply the model for a symmetric discrete-time 1-limited polling system described in [26, p. 140]. We use [26] eq. (6.60), where the switchover time is two slots ($r = 2$) with zero variance ($\delta^2 = 0$), the mean service time is $b = \bar{L} - 1$, and the second moment of the service time is $b^{(2)} = 4p_3 + 16p_5$. By adding 0.5 slot, we obtain the mean waiting time (in slots):

$$\bar{W}_u^{\text{Lim}} = \frac{N \left[1 + 4\lambda(p_3 + 3p_5) \right]}{1 - N\lambda(\bar{L} + 1)}. \quad (7)$$

As a special case, consider a symmetric piconet with unidirectional uplink traffic of 1-slot packets (i.e. $p_1 = 1$) operated according to the limited algorithm. Its mean waiting time has been derived in Section IV-A and it is given by (2). It readily follows that for such a piconet ($\lambda_u^i = \lambda$ and $p_1 = 1$), (7) coincides with (2). Moreover, the result presented in (7) was also verified by a simulation model based on OPNET.

An *asymmetric* piconet with unidirectional uplink traffic can be modeled as a 1-limited polling system with N queues in a similar manner. Since there are no closed form results for the latter case, approximation methods reviewed in [27] can be used. Moreover, a piconet with batch arrivals can be analyzed by applying the methods for the analysis of 1-limited polling systems with a compound Poisson arrival process [27].

D. Equality of Mean Waiting Times

Eq. (5) and (7) lead to the following.

Corollary 1: The mean waiting time in a symmetric piconet with only uplink traffic is the same for the exhaustive and for the limited scheduling algorithms.

It is well known [26],[27] that in the classical symmetric polling systems, where switchover time is incurred *whenever* the server moves from one channel to the next, the mean waiting time in the exhaustive

regime is *smaller* than its counterpart in the 1-limited regime. When the piconet is operated according to the exhaustive algorithm, switchover time is incurred at the end of a slave-master session. On the other hand, in the limited algorithm, when two adjacent slaves' queues are non-empty, no *real* switchover time is incurred. Real switchover times are paid only when a slave has nothing to transmit. Thus, the current limited procedure is more efficient than the classical one and leads to the above waiting time equality.

The mean waiting time in a piconet using the gated regime (6) is higher than the corresponding value in piconets using the exhaustive and limited algorithms. Again, this observation differs from the situation in classical polling systems. Usually, one can use the gated algorithm in order to provide some fairness to the different queues, while maintaining relatively low delay. It seems that in a symmetric piconet with unidirectional traffic, the limited algorithm provides both the desirable fairness and the lowest delay.

As mentioned in sections V-A and V-B, by slightly modifying the Bluetooth specifications one could devise more efficient exhaustive and gated scheduling algorithms. Such algorithms will probably result in reduced waiting times. The exact derivation of these waiting times is a subject for further research.

E. Generalization to Polling Systems

The result presented in Corollary 1 is counterintuitive to the known results regarding the performance of *traditional* polling systems. In addition, as mentioned in Section IV-B, [6] and [10] observed via simulation that in piconets with bidirectional traffic and high loads, the limited algorithm outperforms the exhaustive algorithm. In this section we briefly explore the differences between polling systems that generalize Bluetooth piconets and the traditional polling systems. We focus on the generalization of piconets with only uplink traffic. However, the discussion also sheds light on piconets with bidirectional traffic.

A traditional polling system can be defined as a *per-station polling system* in which the server serves the stations in a cyclic manner and a switchover period is encountered only when the server shifts from one station to another. In contrast, we define a *per-packet polling system*, which is a generalization of a piconet with only uplink traffic. In such a system the server pays a *polling overhead* for *each* packet (we will denote this overhead by r_p). In addition, some *announcement overhead* is also incurred when a station announces to the server that it has a packet to transmit or that it is empty (we will denote this overhead by

r_a). For different scheduling regimes, it is possible to define specific per-station polling systems that are equivalent to the per-packet polling systems. This can be done by using similar methodologies to those used in sections V-A, V-B, and V-C.

Using such equivalences, we now extend the result presented in Corollary 1 regarding symmetric Bluetooth piconets to symmetric (non-Bluetooth) per-packet polling systems. We denote the mean and second moment of the packet service time in the symmetric per-packet polling system by b_{pp} and $b_{pp}^{(2)}$. We assume that the two overheads mentioned above (r_p and r_a) are constants. The service time in the 1-limited per-station system (corresponding to the 1-limited per-packet system) is the same as in the 1-limited per-packet system (i.e. $b = b_{pp}$ and $b^{(2)} = b_{pp}^{(2)}$). The switchover time in the 1-limited per-station system is $r = r_p + r_a$. However, in the exhaustive per-station system (corresponding to the exhaustive per-packet system), the service time is always $r = r_p + r_a$ time units longer than in the 1-limited per-station system. This results from the fact that the per-station exhaustive service time includes the polling and announcement overheads besides the real per-packet service time. Hence, in the exhaustive per-station system, the mean and second moment of the service time are $b + r$ and $b^{(2)} + 2rb + r^2$. The switchover time in this system is also r .

Observation 1: The mean waiting times in symmetric 1-limited and exhaustive per-packet polling systems, which have the same service time distribution and the same constant overheads (r_p and r_a), are equal.

Proof: Consider a symmetric per-packet polling system with b_{pp} , $b_{pp}^{(2)}$, r_p , and r_a (defined above). We apply [26, eq. (6.19)] with service time b , second moment of service time $b^{(2)}$, and switchover time r to get the waiting time in the 1-limited per-station polling system, corresponding to the 1-limited per-packet system. We apply [26, eq. (4.33b)] with service time $b + r$, second moment of service time $b^{(2)} + 2rb + r^2$, and switchover time r to get the waiting time in the exhaustive per-station polling system, corresponding to the exhaustive per-packet system. The obtained waiting times are equal. ■

Similarly to the piconet case, it can also be shown that in a symmetric per-packet polling system, the 1-limited regime outperforms the gated regime (in terms of mean waiting times). According to Observation 1, the result about the equality of the mean waiting times in Bluetooth piconets operated according to the

exhaustive and the limited algorithms (i.e. Corollary 1) is a special case of a result that holds in symmetric per-packet polling systems. A more general special case is of systems operated according to the TDD mechanism in which the packets are not necessarily 1, 3, and 5 slots long (as required in Bluetooth [4]).

We emphasize that the equality of the waiting times under the exhaustive and limited regimes, as well as the fact that the limited regime outperforms the gated regime, *hold only for symmetric traffic*. These observations do not necessarily hold when the traffic is asymmetric. Since in asymmetric systems the service levels to the different stations may differ, the scheduling regimes in asymmetric per-packet polling systems cannot be directly compared through mean waiting times. For such an asymmetric case, Levy et al. [15] presented a sample path comparison that evaluates the efficiency of the different per-station polling systems based on the total amount of work found in the system in any time. They showed that the exhaustive regime dominates the gated which dominates the 1-limited.

We now consider per-packet polling systems. In these systems there are arrival processes for which the amount of work in the system under the 1-limited regime is lower than that under the gated regime, whereas there are arrival processes for which the amount of work under the gated regime is lower than that under the 1-limited regime. Similar relations exist between the exhaustive and 1-limited regimes. Hence, unlike in the traditional polling systems, the gated and exhaustive regimes do not strictly dominate the 1-limited regime. On the other hand, the 1-limited regime does not strictly dominate the others. In general, the proof methodology of [15] cannot be directly applied to identify dominance relations, since the corresponding per-station polling systems differ from each other in their service times. Yet, the results of [15] directly imply that the exhaustive per-packet system dominates the gated per-packet system. To conclude, the difference in the dominance relations between per-packet and (traditional) per-station polling systems stems from the operational differences between these systems.

VI. DOWNLINK TRAFFIC

A. Analysis of the Exhaustive Algorithm

Consider a piconet with only *downlink* traffic operated according to the exhaustive algorithm. In such a piconet traffic flows only from the master to the slaves and the master has complete information on the status of its downlink queues. Thus, there is no reason to send a POLL packet in order to end a

master-slave exchange. Yet, in case all queues are empty, the master regularly transmits POLL packets (and receives NULL packets) until a data packet arrives to one of its downlink queues.

We define the operation model of the piconet as follows. The master serves the downlink queues in a fixed cyclic order. When serving queue i , the master sends all data packets present in the queue and the slave replies with a NULL packet for each data packet. When the master empties queue i , it immediately switches, in a cyclic manner to the next non-empty downlink queue. In case all queues are empty, the master sends a POLL packet to one of the slaves which replies with a NULL packet. If by the end of the NULL packet at least one of the queues becomes non-empty, the master randomly selects one of the N queues and proceeds from there in a cyclic manner until it finds a non-empty queue which is immediately served.

Fig. 7 illustrates an example of the operation of such a piconet. In this example, when the master empties the downlink queue of packets intended to slave 1, the queue of slave 2 is empty. When it empties the queue of slave 3, all the queues are empty, and therefore, it sends POLL packets until at least one packet arrives. In the described scenario, packets arrive to queues 1 and 4 during the transmission of the NULL packet by slave 2. The master randomly selects queue 4, serves it, and continues to queue 1 in a cyclic manner.

The piconet can be modeled as a *discrete time* polling system with zero-switchover periods [12]. In order to obtain results for a discrete time exhaustive polling system with zero-switchover periods, Levy and Kleinrock [12] define the AZSOP (*Almost Zero Switchover Period*) polling system. In that system it is assumed that the switchover period is nonzero with probability p ($p > 0$) and zero with probability $1 - p$. Then, the system is analyzed as an exhaustive polling system with mean switchover times defined as p and the variances of switchover time defined as $p(1 - p)$. It is shown that when $p \rightarrow 0$, the waiting time in the AZSOP system approaches the waiting time in the zero-switchover period system.

We note that the *continuous time* polling system with zero-switchover periods [26, p. 142] *does not comply* with the operation model of a piconet, due to the following reason. In the continuous time model it is assumed that if a packet arrives while the server is idle, its service starts immediately. On the other hand, in a piconet, if a packet arrives while a POLL or a NULL packet is sent, it could be served only

after the transmission of the NULL packet.

In order to model the piconet as an AZSOP polling system, *we define a single slot in the AZSOP system as two slots in a Bluetooth piconet*. To this end, we define the service time of a k -slot data packet in a Bluetooth piconet as $(k + 1)/2$ slots in the AZSOP system, which are composed of the k slots of data, augmented by the *following* NULL packet. Thus, in the corresponding AZSOP polling system, the service time of a 1-slot Bluetooth packet is defined as 1 slot, for 3-slot Bluetooth packet it is 2 slots, and for 5-slot Bluetooth packet it is 3 slots. The switchover time in the AZSOP system is defined as 1 slot, composed of POLL and NULL packets. As mentioned above, the length of this period is 1 slot with probability p .

We now focus on symmetric systems (i.e. $\lambda_d^i = \lambda > 0 \forall i$ and $\lambda_u = 0 \forall i$). By applying the model for a discrete-time exhaustive polling system described in [26] and using the methodology described in [12], we can obtain the mean waiting time. Accordingly, we apply [26] eq. (3.63b), where the arrival process is Poisson with intensity 2λ , the switchover time is $r = p$, the variance of the switchover time is $\delta^2 = p(1 - p)$, the mean service time is $b = (\bar{L} + 1)/2$, and the second moment of the service time is $b^{(2)} = p_1 + 4p_3 + 9p_5$. Letting $p \rightarrow 0$, adding 0.5 slot (since in [26] the waiting time is counted from the end of the slot), and multiplying by 2 (since the obtained result is the number of slots in the AZSOP system and we are interested in waiting time measured in Bluetooth slots), we obtain the mean waiting time (in Bluetooth slots):

$$\overline{W}_d^{\text{Ex}} = \frac{1 + 4N\lambda(p_3 + 3p_5)}{1 - N\lambda(\bar{L} + 1)}. \quad (8)$$

A similar approach can be used for the analysis of *asymmetric* piconets with only downlink traffic (i.e. $\lambda_u^i = 0 \forall i$, and $\lambda_d^i > 0 \forall i$, not all necessarily equal). That is, it can be modeled as an asymmetric AZSOP polling system operated according to the exhaustive regime and composed of N queues, with 1-slot switchover time and with service time of $(k + 1)/2$ slots for a k -slot data packet. Accordingly, a relatively good approximation of the waiting times in each downlink queue can be computed by solving $O(N^3)$ equations as described in [12, Section 3.6].

B. Gated and Limited Algorithms

Symmetric piconets with only downlink traffic that are operated according to the gated and limited algorithms can be modeled as AZSOP polling systems similarly to the modeling of exhaustive algorithm. By applying the models for discrete-time gated and 1-limited polling systems described in [26] and using the methodology described in [12], we obtain the mean waiting times for the two schemes. It turns out that all 3 mean waiting times, for the exhaustive, gated, and limited, are equal and given by (8).

C. Comparison

The fact that in a symmetric piconet with only downlink traffic, the mean waiting time is the same for all algorithms is expected. Such a result was obtained in [12] for a symmetric discrete-time polling system with fixed service times and zero switchover times. Similarly, it is well known [12],[26] that the mean waiting time in *symmetric* continuous time polling systems with zero switchover time is equal to the mean waiting time in an M/G/1 system with the combined inputs of all queues, regardless of the polling regime (exhaustive, gated, or 1-limited). Yet, it is interesting to compare the results obtained for systems with only downlink traffic (i.e. (8)) to the results for systems with only uplink traffic (i.e. (5), (6), and (7)). It can be seen that

$$\overline{W}_d^{\text{Lim}} = \overline{W}_d^{\text{G}} = \overline{W}_d^{\text{Ex}} = \overline{W}_u^{\text{Ex}} - \frac{N-1}{1 - N\lambda(\overline{L} + 1)}, \quad (9)$$

where it has been shown in Section V that

$$\overline{W}_u^{\text{Ex}} = \overline{W}_u^{\text{Lim}} = \overline{W}_u^{\text{G}} - \frac{2N\lambda(\overline{L} + 1)}{1 - N\lambda(\overline{L} + 1)}. \quad (10)$$

Moreover, for the special case in which the traffic is composed of only 1-slot packets (i.e. $p_1 = 1$), there is a significant difference between the values in only downlink and only uplink piconets. Namely, $\overline{W}_u^{\text{Lim}} = \overline{W}_u^{\text{Ex}} = N\overline{W}_d^{\text{Lim}} = N\overline{W}_d^{\text{Ex}} = N\overline{W}_d^{\text{G}}$. The difference between the waiting times results from the fact that in a system with only downlink traffic the master has complete information regarding the queues. Therefore, in such a system there is no need to waste POLL and NULL packets on communicating with slaves having empty queues.

The above results can be useful for developing topology construction algorithms for piconets and for scatternets (see [28] for a review of these algorithms), that are able to utilize information regarding the traffic statistics. When the traffic is mostly unidirectional, allowing the node that generates most of the traffic to be the master would significantly decrease the delay.

VII. UPLINK AND DOWNLINK TRAFFIC

In section IV we discussed the case in which the traffic on the links is bidirectional. In this section we briefly discuss the case in which the traffic on each link is either downlink or uplink traffic. In such a case the system can be analyzed using a combination of the methods described above. We will demonstrate the analysis of such a system operated according to the exhaustive algorithm. The gated and limited algorithms can be analyzed in a similar manner.

The piconet operates as follows. The master serves the queues in a fixed cyclic order. When serving an uplink traffic link, it sends POLL packets and receives data packets until the service is terminated by a POLL-NULL exchange. When serving a downlink traffic link, data packets are sent and NULL packets are received as long as the queue is not empty. If a downlink queue is empty, the master *immediately* switches to the next non-empty downlink queue or to the next uplink queue (i.e. no time is spent on an empty downlink queue).

This system can be modeled as an exhaustive polling system, where the service time of a k -slot data packet is defined as $k + 1$ slots (composed of the data packet and the following POLL or NULL packet). The switchover times between different queue types are as follows:

- Uplink queues – two slots which are composed of the NULL packet ending the exchange and the POLL packet starting the next exchange.
- Downlink queues – zero slots.
- An uplink queue and a downlink queue – a single slot which is the NULL packet ending the exchange with the uplink queue.
- A downlink queue and an uplink queue – a single slot which is the POLL packet starting the exchange with the uplink queue.

Since the system consists of at least one uplink queue, not all the switchover times are zero. Therefore,

results can be obtained by any of the methods for analyzing asymmetric exhaustive polling systems (see Section V-A).

VIII. NUMERICAL RESULTS

Approximate results regarding the performance of various intra and inter-piconet scheduling algorithms have been presented in [16],[20],[21], and [23]. For example, the analysis of the *limited* and the *exhaustive* algorithms in [20] is based on modeling the piconet as an M/G/1 queue with vacations. In addition, the analysis of the exhaustive algorithm in [20] is based on the derivation of the PGF of the time to exhaust a single master-slave queue pair. The results obtained in [20] are approximate, since the analysis does not fully take into account the complexities discussed in Section IV-B and since the application of results from the M/G/1 queue with vacations model to the piconet system neglects statistical dependencies that exist in the piconet operation model (for more details see [29] and [30]). In this section we compare our exact numerical results to numerical results computed according to [20] and [21]. We note that in [18] the approximate results of [16] have already been compared to our exact results.

The model presented in Section III is a specific case of the piconet model presented in [20]. Thus, Fig. 8, compares the mean waiting time computed according to the analysis of the limited regime in [20] to the mean waiting time computed according to our model (i.e. according to (3)) when all packets are 5 slots long. The waiting time (in slots) is depicted as a function of the load in the bidirectional limited system ($2N\lambda\bar{L}$) in symmetric piconets with 2 and 4 slaves. Similarly, Fig. 9, compares the mean waiting time computed according to the analysis of the exhaustive regime in [20] when $\lambda_d^i = 0 \forall i$ to the mean waiting time computed according to our analysis (i.e. according to (5)). The figure presents the waiting time as a function of the load in the uplink exhaustive system ($N\lambda(\bar{L} + 1)$) in symmetric piconets with 2 slaves.

In [21] the intra-piconet exhaustive scheduling algorithm is analyzed in a somewhat different methodology than the analysis described in [20]. In Fig. 10 we compare the exact mean waiting time to the mean waiting time computed according to [21], in symmetric piconets with only uplink traffic ($\lambda_u^i = \lambda, \lambda_d^i = 0 \forall i$) in which the probabilities of 1, 3, and 5-slot packets are equal.

It is seen that in all cases shown, the results presented in [20] underestimate the mean waiting time

	<i>bidirectional</i>	<i>Unidirectional Uplink</i>	<i>Unidirectional Downlink</i>	<i>Unidirectional Downlink or Uplink</i>
Exhaustive	—	asymmetric	symmetric (exact) and asymmetric (approximate)	asymmetric
Gated	—	asymmetric	symmetric (exact) and asymmetric (approximate)	asymmetric
Limited	symmetric	symmetric	symmetric	symmetric

TABLE I

THE TYPE OF ARRIVAL PROCESS (SYMMETRIC OR ASYMMETRIC) FOR WHICH EXACT RESULTS CAN BE OBTAINED UNDER DIFFERENT TRAFFIC PATTERNS AND SCHEDULING ALGORITHMS (THE CASE IN WHICH GOOD APPROXIMATE RESULTS CAN BE OBTAINED IS ALSO INDICATED).

while those in [21] either underestimate or overestimate the mean waiting time. Thus, we conjecture that for complicated scenarios, deriving approximate results, which are based on the relationship between Bluetooth piconets and polling systems, will yield significantly better approximations than those that are based on M/G/1 queue with vacations.

IX. CONCLUSIONS

This work reveals overlooked connections between *Bluetooth piconets* and *polling systems*. This connection enables us to obtain exact analytical results regarding the performance of Bluetooth scheduling algorithms. The obtained results are summarized in Table I. First, we have analyzed piconets with bidirectional traffic. We have shown that a piconet operated according to the limited scheduling algorithm is equivalent to a 1-limited polling system, and derived exact analytic results for symmetric systems. The complications in analyzing the gated scheduling algorithm in piconets with bidirectional traffic have been described, indicating that the corresponding analysis of the exhaustive regime is even more complex.

Then, we have analyzed piconets with unidirectional uplink traffic. We obtained exact results for the symmetric limited, gated, and exhaustive regimes, and showed that exact results can also be obtained for asymmetric piconets operated according to the gated and exhaustive algorithms. We have shown that in *symmetric* piconets with only uplink traffic, the mean waiting times are the same for the limited and exhaustive algorithms. In order to facilitate the explanation of this phenomenon, we have defined per-packet polling systems and generalized the results for these systems.

Furthermore, we have shown that a piconet with unidirectional downlink traffic is equivalent to a

polling system with zero-switchover times. The mean waiting times in such a piconet can be significantly lower than in piconets with only uplink traffic. We also discussed piconets in which the traffic on the links is either unidirectional uplink or unidirectional downlink. Finally, we compared numerical results to approximate results derived in the past.

The presented analysis can be extended in various directions (e.g. batch arrivals, asymmetric arrival processes, retransmissions) by applying various results regarding the performance of polling systems (e.g. the extensions in [17] and [18] to our preliminary results regarding the limited algorithm [30]). In addition, a similar approach can be used to analyze the performance of other centrally controlled wireless networks [5].

The exact results presented in this paper can be utilized in order to validate and evaluate simulation models and approximate analytic models. They also provide a few important insights regarding the design and performance of Bluetooth piconets and scatternets. For example, since the mean waiting times are equal for the exhaustive and limited algorithms, it seems that when the traffic is mostly unidirectional and symmetric, the limited algorithm, which provides some degree of fairness, is preferable. Moreover, topology construction algorithms, that have some information regarding the traffic statistics, can exploit the observation that when the traffic is mostly unidirectional, allowing the node that generates most of the traffic to be the master would significantly decrease the delay. Finally, the effect of the packet length distribution on the waiting time has been revealed (the effect on the piconet throughput has already been studied in [24]).

Due to the inherent complexities in analyzing the gated and exhaustive algorithms operating in a piconet, a future research direction is to obtain a good (at least approximate) analysis of such regimes. In addition, this work raises interesting questions regarding the performance of State-Dependent polling systems in which setup time is required in empty queues. Finally, it has been shown that due to the TDD mechanism, algorithms that tend to optimize the performance of polling systems are not necessarily optimal for piconets and for per-packet polling systems. Therefore, a future research direction is the development of scheduling algorithms that will optimize the performance of the latter systems.

ACKNOWLEDGMENTS

This research (No. 148/03) was supported by THE ISRAEL SCIENCE FOUNDATION, by a grant from the Ministry of Science, Israel, and by a Marie Curie International Fellowship within the 6th European Community Framework Programme.

We thank Daniele Miorandi and Hanoch Levy for helpful discussions and comments. We thank the reviewers for helpful comments. The term per-packet polling system was suggested by an anonymous reviewer.

REFERENCES

- [1] E. Altman, H. Blanc, A. Khamisy, and U. Yechiali, "Gated-type polling systems with walking and switch-in times", *Stochastic Models*, Vol. 10, pp. 741-763, 1994.
- [2] S. Baatz, M. Frank, C. Kühl, P. Martini, and C. Scholz, "Bluetooth scatternets: an enhanced adaptive scheduling scheme", *Proc. IEEE INFOCOM'02*, June 2002.
- [3] D. P. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall Inc., 1992.
- [4] Bluetooth Special Interest Group, *Specification of the Bluetooth System – Version 2.0*, Nov. 2004.
- [5] E. Biton, D. Sade, D. Shklarsky, M. Zussman, and G. Zussman, "Challenge: CeTV and Ca-Fi - Cellular and Wi-Fi over CATV", *Proc. ACM MOBICOM'05*, Sep. 2005.
- [6] A. Capone, M. Gerla, and R. Kapoor, "Efficient polling schemes for Bluetooth picocells", *Proc. IEEE ICC'01*, June 2001.
- [7] R. B. Cooper, S.-C. Niu, and M. M. Srinivasan, "Setups in polling models: does it make sense to set up if no work is waiting?", *J. Appl. Probab.*, Vol. 36, pp. 585-592, 1999.
- [8] 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.
- [9] R. Esmailzadeh, M. Nakagawa, and A. Jones, "TDD-CDMA for the 4th generation of wireless communications", *IEEE Wireless Comm.*, Vol. 10, pp. 8–15, Aug. 2003.
- [10] L. Har-Shai, R. Kofman, A. Segall, and G. Zussman, "Load adaptive inter-piconet scheduling in small-scale Bluetooth scatternets", *IEEE Comm.*, Vol. 42, pp. 136–142, July 2004.
- [11] R. Kapoor, A. Zanella, and M. Gerla, "A fair and traffic dependent scheduling algorithm for Bluetooth scatternets", *ACM/Kluwer MONET*, Vol. 9, pp. 9–20, Feb. 2004.
- [12] H. Levy and L. Kleinrock, "Polling systems with zero switch-over periods: a general method for analyzing the expected delay", *Performance Evaluation*, Vol. 13, pp. 97–107, Oct. 1991.
- [13] H. Levy and M. Sidi, "Polling systems: applications, modeling and optimization", *IEEE Trans. on Comm.*, Vol. 38, pp. 1750-1760, Oct. 1990.
- [14] H. Levy and M. Sidi, "Polling systems with simultaneous arrivals", *IEEE Trans. on Comm.*, Vol. 39, pp. 823–827, June 1991.
- [15] H. Levy, M. Sidi, and O. Boxma, "Dominance relations in polling systems", *QUESTA*, Vol. 6, pp. 155–171, Apr. 1990.
- [16] D. Miorandi, C. Caimi, and A. Zanella, "Performance characterization of a Bluetooth piconet with multi-slot packets", *Proc. WiOpt'03*, Mar. 2003.
- [17] D. Miorandi and A. Zanella, "Achievable rate regions for Bluetooth piconets in fading channels", *Proc. IEEE VTC'04 Spring*, May 2004.
- [18] D. Miorandi, A. Zanella, and S. Merlin, "Mathematical analysis of the packet delay statistics in Bluetooth piconets under round robin polling regime", *Med. J. on Comp. Netw.*, Vol. 2, pp. 48-55, Jan. 2006.
- [19] D. Miorandi, A. Zanella, and G. L. Pierobon, "Performance evaluation of Bluetooth polling schemes: an analytical approach", *ACM/Kluwer MONET*, Vol. 9, pp. 63–72, Feb. 2004.
- [20] J. Mistic and V. B. Mistic, "Modeling Bluetooth piconet performance", *IEEE Comm. Letters*, Vol. 7, pp. 18-20, Jan. 2003.
- [21] J. Mistic and V. B. Mistic, "Bridges of Bluetooth county: topologies, scheduling, and performance", *IEEE J. on Selected Areas in Comm.*, Vol. 21, pp. 240–258, Feb. 2003.
- [22] J. Mistic and V. B. Mistic, *Performance modeling and analysis of Bluetooth networks: polling, scheduling, and traffic control*, Auerbach Publications, 2006.
- [23] V. B. Mistic and J. Mistic, "Performance of Bluetooth slave/slave bridge", *Telecommunications Systems*, Vol. 22, pp. 221–239, Jan.–Apr. 2003.
- [24] S. Sarkar, F. Anjum, and R. Jain, "Throughput optimal packet size selection (TOPS) for Bluetooth networks", *Proc. ITC-18*, Sep. 2003.
- [25] M. M. Srinivasan and M. P. Singh, "An efficient exact procedure for the analysis of the state-dependent polling model", *IIE Trans.*, Vol. 38, pp. 153–158, Feb. 2006.
- [26] H. Takagi, *Analysis of polling systems*, MIT Press, 1986.

- [27] H. Takagi, "Queueing analysis of polling models: progress in 1990-1994", In *Frontiers in Queueing* (ed.: J. H. Dshalalow), CRC Press, pp. 119–146, 1997.
- [28] R. M. Whitaker, L. Hodge, and I. Chlamtac, "Bluetooth scatternet formation: a survey", *Ad Hoc Networks*, Vol. 3, pp. 403-450, July 2005.
- [29] G. Zussman, A. Segall, and U. Yechiali, "Bluetooth time division duplex – exact analysis as a polling system", *CCIT Report No. 414, Dept. of Electrical Engineering, Technion*, Available at http://www.mit.edu/~gilz/pub_files/ccit_414.pdf, Feb. 2003.
- [30] G. Zussman, U. Yechiali, and A. Segall, "Exact probabilistic analysis of the limited scheduling algorithm for symmetric Bluetooth piconets", *Proc. IFIP-TC6 PWC'03, LNCS Vol. 2775* (eds.: M. Conti et al.), pp. 276–290, Springer, Sep. 2003.
- [31] G. Zussman, A. Segall, and U. Yechiali, "Bluetooth time division duplex – analysis as a polling system", *Proc. IEEE SECON'04*, pp. 547–556, Oct. 2004.

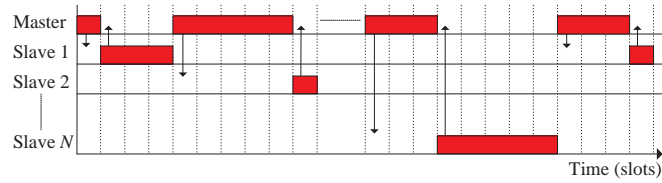


Fig. 1. An example of the TDD scheme in a Bluetooth piconet.

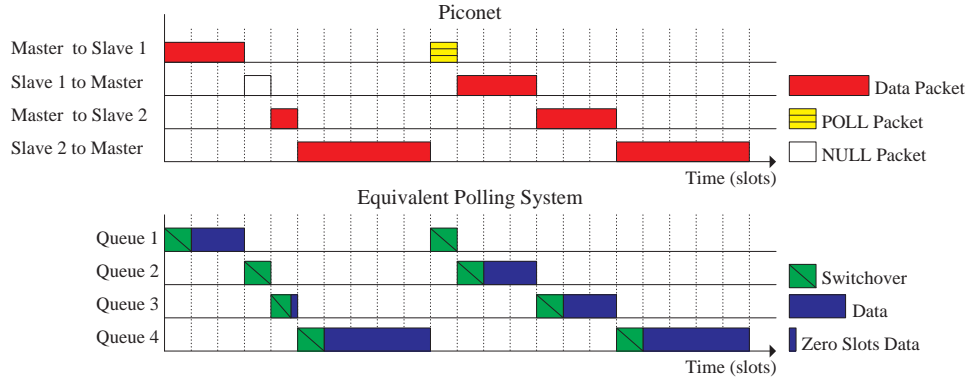


Fig. 2. An example of a piconet operated according to the limited algorithm and of the equivalent polling system.

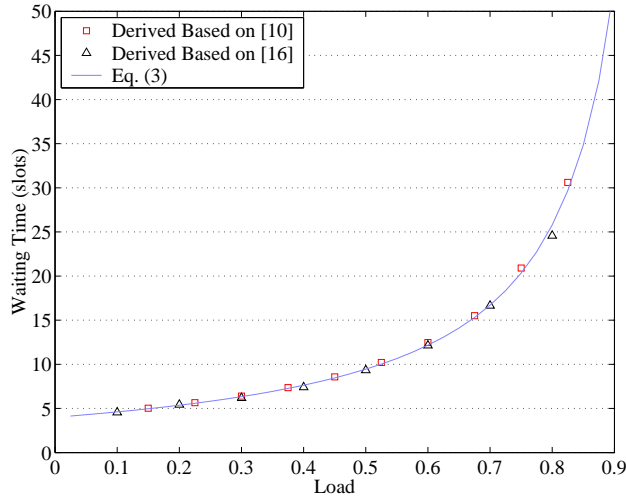


Fig. 3. The mean waiting time (calculated according to (3)) and the average waiting time values computed by simulation in a piconet with 4 slaves in which $p_1 = p_3 = p_5 = 1/3$.

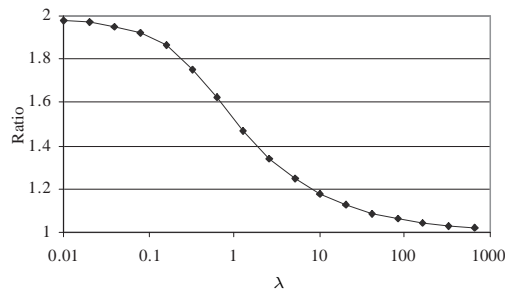


Fig. 4. The ratio of the average value of $\max [U, D]$ to λ .

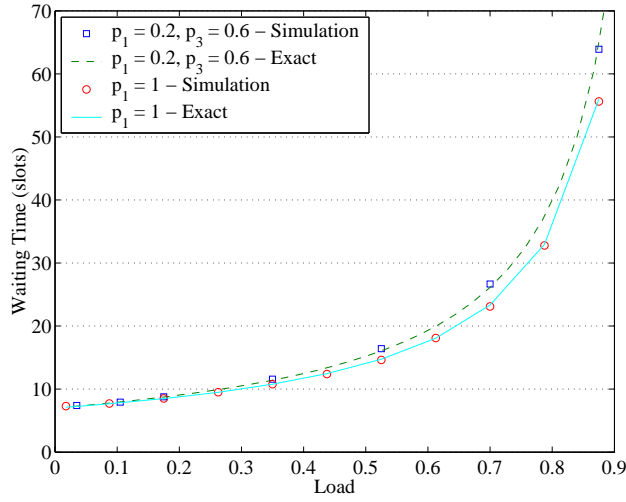


Fig. 5. The exact mean waiting time (calculated according to (5)) and the average waiting time values, computed by simulation, in piconets composed of 7 slaves and with only uplink traffic, operated according to the exhaustive algorithm.

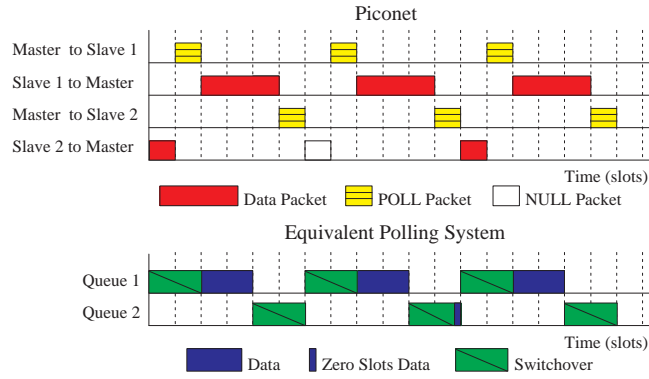


Fig. 6. An example of a piconet with only uplink traffic operated according to the limited algorithm and of the equivalent polling system.

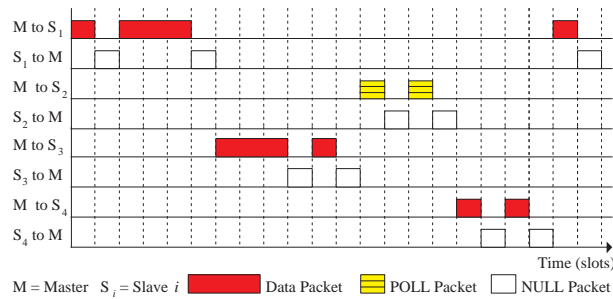


Fig. 7. An example of the operation of the exhaustive algorithm in a piconet with only downlink traffic.

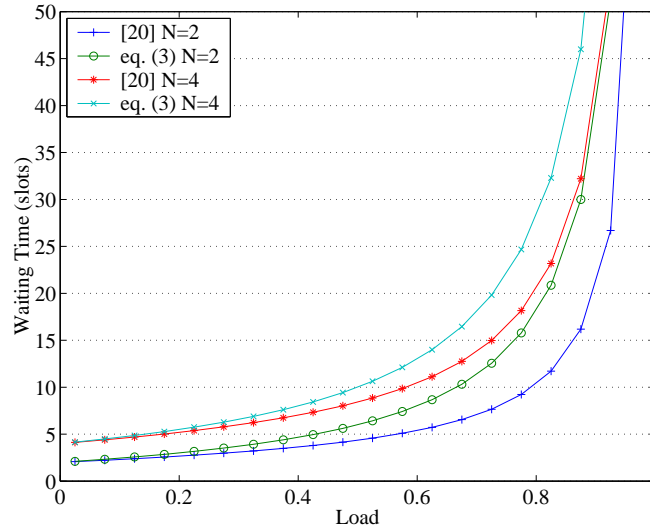


Fig. 8. The mean waiting time derived according to [20] and the exact mean waiting time (computed according to (3)) in symmetric piconets operated according to the limited algorithm in which all packets are 5 slots long ($p_5 = 1$).

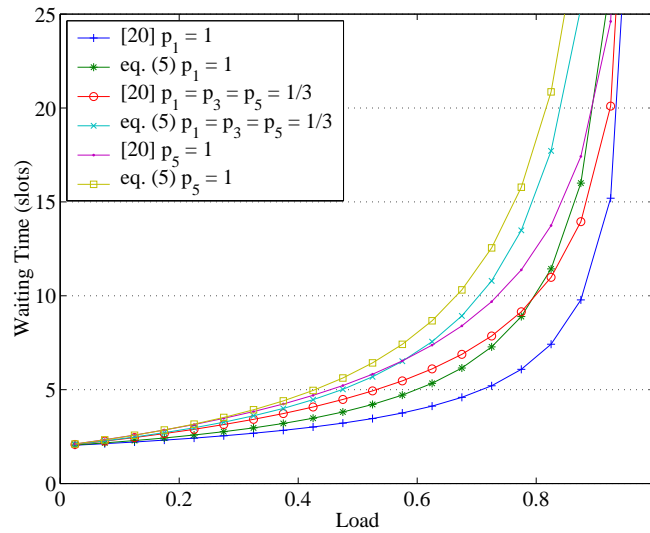


Fig. 9. The mean waiting time derived according to [20] and the exact mean waiting time (derived according to (5)) in symmetric piconets with only uplink traffic, composed of 2 slaves, and operated according to the exhaustive algorithm.

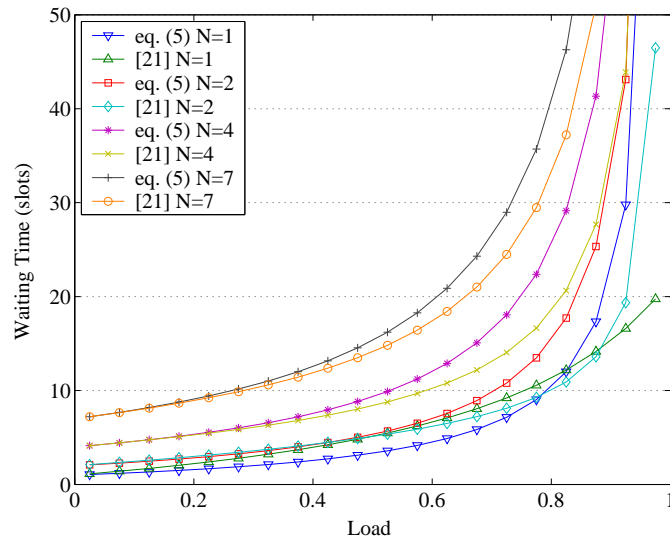


Fig. 10. The mean waiting time derived according to [21] and the exact mean waiting time (derived according to (5)) in symmetric piconets with only uplink traffic, operated according to the exhaustive algorithm, in which $p_1 = p_3 = p_5 = 1/3$.