Mobile Networks and Applications 9, 63–72, 2004

Performance Evaluation of Bluetooth Polling Schemes: An Analytical Approach *

DANIELE MIORANDI, ANDREA ZANELLA and GIANFRANCO PIEROBON
Department of Information Engineering, University of Padova, v. Gradenigo 6/B, 35131 Padova, Italy

Abstract. In the recent years, many polling schemes for Bluetooth networks have been proposed and evaluated. To the authors knowledge, however, analysis has been carried out mainly through computer simulations and, up to now, no mathematical treatment of this topic has been presented. In this paper, we propose an analytical framework for performance evaluation of polling algorithms in Bluetooth piconets. The analysis is carried out by resorting to an effective and simple mathematical method, called Equilibrium Point Analysis. The system is modelled as a multidimensional finite Markov chain and performance metrics are evaluated at the equilibrium state. The analysis is focused on three classical polling schemes, namely Pure Round Robin, Gated Round Robin and Exhaustive Round Robin, which are compared in terms of packet delay, channel utilization, and fairness among users. Both analytical and simulation results are presented for three relevant scenarios, in order to validate the accuracy of the analysis proposed.

Keywords: Bluetooth, polling schemes, equilibrium point analysis, performance evaluation

1. Introduction

Originally born as a wireless replacement for cables connecting electronic devices, Bluetooth has been gaining a lot of consideration and attention by the scientific community in the last few years. The ability of Bluetooth devices to form small networks called piconets, opens up a whole new arena for applications where information may be exchanged seamlessly among the devices in the piconet. Typically, such a network, referred to as a WPAN (Wireless Personal Area Network), may consist of a mobile phone, laptop, palmtop, headset, and other electronic devices that a person carries around in his every day life. The WPAN may, from time to time, also include devices that are not carried along with the user, e.g., an access point for Internet connection or sensors located in a room. Moreover, devices from other WPANs may also be interconnected to enable information sharing. The commercial success of WPANs is intimately linked to their ability to support advanced digital services, like audio and video streaming, web browsing, etc. In such a scenario, the performance aspects of the radio technologies involved appear of primary importance. For Bluetooth, in particular, the design of effective polling schemes is an attractive issue, due to its potentially dramatic impact on system performance.

Polling schemes have been extensively studied in the last decades and exact analysis has been performed for many policies [12,14,15]. Nevertheless, the application of such results to the specific case of Bluetooth appears rather difficult because of the Time Division Duplex (TDD) nature of Bluetooth links. This consideration has driven many researchers to investigate the performance achieved by classic polling schemes in Bluetooth piconets, and has fostered the design of new, effective polling strategies [3,5,7,8,18] for Bluetooth. To the knowledge of the authors, however, no theoretical treatment of Bluetooth polling schemes has been presented in the literature.

In this paper, we provide a simple mathematical model for a Bluetooth piconet. Our primary interest is to evaluate, by analytical methods, the performance of exhaustive and semi-exhaustive polling schemes for a Bluetooth piconet. It is, indeed, widely acknowledged that exhaustive disciplines are a valuable choice [13] when aggregated throughput and channel utilization are of primary interest. In particular, we consider three basic disciplines: Pure Round Robin (PRR), Gated Round Robin (GRR) and Exhaustive Round Robin (ERR).

Unfortunately, the exact analysis of such polling strategies for a Bluetooth-like TDD system may result practically untractable. For instance, given a piconet with N active slaves, the delay analysis for the exhaustive polling strategy (using the so-called buffer occupancy method [12]) would require the solution of order of $(2N)^3$ equations. To overcome this drawback, we resort to an approximate method, based on the so-called Equilibrium Point Analysis (EPA).

The EPA was first introduced by Tasaka [16] to study the dynamic behavior of the R-ALOHA protocol, generalizing a concept introduced by Kleinrock and Lam [11] and, independently, by Carleial and Hellman [4]. The EPA method is generally used to obtain an estimation of the performance yielded by Markovian systems. The basic idea is to evaluate the performance of the system in its *equilibrium point*. An equilibrium point may be described as an attractive point in the state space of the system. At the equilibrium, the sum of the stochastic forces that act on the system is, on average, zero. Hence, the system statistically tends to *gravitate* around that point [17]. This intuitive notion of equilibrium can be given a precise mathematical setting by considering a random vector,

* This work was supported by the Department of Information Engineering (DEI) of the University of Padova (Italy).

$\boldsymbol{\beta}(\cdot)$, representing the state of the system, and by setting

$$E\big[\boldsymbol{\beta}(n+1) \mid \boldsymbol{\beta}(n) = \mathbf{b}\big] = \mathbf{b}. \tag{1}$$

This gives rise to a set of equations that can be interpreted as equilibrium curves in the state space: the solution of the system is given by the points where the curves intersect. (Note that, in general, it is possible to have more than a single equilibrium point, as happens, for example, for the ALOHA system [4].)

In order to apply the EPA method to the case of interest, we first describe the system as an appropriate discrete-time homogeneous Markov chain. Then, we define the stability conditions, which, in turn, force the system to present a unique equilibrium point. Hence, we solve the system of equilibrium equations given by (1), for the different polling strategies considered. Finally, we compute the performance metrics, namely packet delay, channel utilization and fairness (which will be defined in the following), at the equilibrium point. It may be worth noticing that, in some papers, stability issues are not taken into account and unstable systems are simulated, giving completely misleading results.

The paper is organized as follows. In section 2, we provide a mathematical model for the system, and introduce the simplifying assumptions that make the problem mathematically tractable. In section 3, we derive the equilibrium equations for the three disciplines considered (PRR, GRR, ERR) and the stability conditions. In section 4, we define the performance metrics and derive their expressions at the equilibrium, for the three polling strategies. In section 5, we present and discuss simulation and analytical results for some case studies. Finally, in section 6, we present our conclusions and point out open issues and future work.

2. System model

The scenario we consider is a single piconet with N active slaves, i.e., N full-duplex links. We model the system as a network of $2N$ queues, two for each link (one for the master-to-slave and one for the slave-to-master transmissions), so that the master node will appear as formed by N (independent) queues. We assume that the processes of arrivals at the various nodes are independent, neglecting the possibility of two slaves communicating through the master. We assume that the packet arrivals processes can be modelled as independent Poisson processes. Although it is widely acknowledged that Markovian models are not completely suited to describe traffic patterns in modern packet switching networks, this is a classical assumption that, on the one hand, lends itself to a simple mathematical treatment and, on the other hand, makes it possible to compare our results with some recent literature, e.g., [8]. At this stage of the work, moreover, we consider only single-slot packets, more precisely $DH1$ packets. The introduction of multi-slot packets in the model would raise many important issues, like the interaction between Segmentation-And-Reassembly (SAR) policy and the polling scheme employed [8], and would require further analysis that we defer to future works.

In the rest of the paper, we adopt the following conventions. Links are enumerated from 1 to N, according to the order they are served in the polling cycle. Each node (or, equivalently, each queue) is uniquely identified by the pair (i, t), where $i \in \{1, \dots, N\}$ indicates which link we are referring to, while t takes on the value M for the master's queue and S for the slave's queue. The intensity of arrivals (in packets/s) at node (i, t) is denoted by $\lambda_{i,t}$. Then, denoting by $T = 0.625$ ms the Bluetooth time slot duration, the normalized traffic offered to queue (i, t) is given by $G_{i,t} = \lambda_{i,t}T$. For the generic lth polling cycle, we define $x_{i,t}(l)$ as the length of the queue (i, t) when the node enters the lth service period, while $y_{i,t}(l)$ indicates the total number of packets sent by node (i, t) during that service period. The time duration of the service period for the link j, that is, the time the link j holds the *token* for, is denoted by $S(j, l)$. Finally, $N_{i,t}(u)$ denotes the number of packets that arrive at node (i, t) during the generic time interval u.

In order to obtain a homogeneous Markovian model, we observe the system in an asynchronous manner, that is, we look at each queue when it is polled. In synthesis, we define the state of the system at step l as the vector

$$\boldsymbol{\beta}(l) = \big[x_{1,M}(l), x_{1,S}(l), x_{2,M}(l), \dots, x_{N,S}(l)\big].$$

At each cycle, the number of packets queued at a given node, when the node gets the token, is given by the number of packets queued at the previous cycle, minus the packets sent during that cycle, plus the new arrivals. Since the generic link i is preceded by $i-1$ and followed by $N-i+1$ other links in the polling cycle, the one-step evolution of the system state can be expressed by the following the set of equations:

$$\begin{aligned} x_{i,t}(l+1) = {} & x_{i,t}(l) - y_{i,t}(l) \\ & + N_{i,t}\left(\sum_{j=1}^{i-1} S(j, l+1) + \sum_{j=i}^{N} S(j, l)\right); \end{aligned} \tag{2}$$

for $i = 1, 2, \dots, N$, $t = S, M$. For almost all schemes of practical interest, the statistics of $S(i, l)$ and $y_{i,t}(l)$ depend just upon $\boldsymbol{\beta}(l)$. Hence, under the assumption of Markovian arrivals, the system can be described by a Markov chain and, at least in theory, it could be solved using the well-known classical techniques [12,14,15].

In practice, an exact analysis is feasible only for the case of Pure Round Robin (see, for instance, [10]). On the contrary, the exhaustive and semi-exhaustive polling strategies applied to the Bluetooth network topology appear much more critic to be studied. The difficulties arise from the lack of closed-form expression for the statistic of the depletion time of a master–slave couple, when the ERR and GRR strategies are considered. Moreover, the expectation of such time can be computed only numerically. Using the buffer occupancy approach [12] to compute the average length of queue (i, t) when polled, we end up with a system of $(2N)^2$ non-linear equations that has to be solved with the aid of a computer program. Furthermore, an exact delay analysis could be carried out by solving a system of $(2N)^3$ equations. Although possible in theory, this approach has shown not to be feasible in practice, due to the extremely high amount of time necessary to solve the system.

On the basis of these considerations, we attempt to solve the problem in an approximate manner, reducing the analysis to a computable one. To this purpose we resort to EPA, a method that, indeed, allows us to reduce to a system of (still nonlinear) $2N$ equations.

3. Equilibrium point equations and stability conditions

Applying (1) to (2), we get a system of $2N$ equations, whose solution gives the equilibrium point. For the generic node (i, t), with $i = 1, 2, \ldots, N$ and $t = S, M$, the equilibrium equation can be written as

$$E\left[b_{i,t} - y_{i,t} + N_{i,t}\left(\sum_{i<j} S(j, l+1) + \sum_{j \geqslant i} S(j, l)\right) \middle| \boldsymbol{\beta}(l) = \mathbf{b}\right] = b_{i,t}, \tag{3}$$

where the vector $\mathbf{b} = [b_{1,M}, b_{1,S}, \ldots, b_{N,S}]$ that satisfies the system is the equilibrium point.

The system does not lend itself to a simple solution. In order to simplify it, we need to introduce some heuristics. In the case of a single equilibrium point, what we do with EPA is somehow equivalent to approximate the stationary probability distribution of an ergodic Markov chain with a unit impulse [17]. Hence, it is reasonable to assume that, in equilibrium, we have $E[S(j, l+1)|\boldsymbol{\beta}(l) = \mathbf{b}] \approx E[S(j, l)|\boldsymbol{\beta}(l) = \mathbf{b}]$. Under this assumption (whose effectiveness will be proven by means of computer simulation), equation (3) becomes:

$$E\left[b_{i,t} - y_{i,t} + N_{i,t}\left(\sum_{j=1}^{N} S(j, l)\right) \middle| \boldsymbol{\beta}(l) = \mathbf{b}\right] = b_{i,t}. \tag{4}$$

According to our assumptions, the process of arrivals at node (i, t) is Poisson with intensity $\lambda_{i,t}$. Thus, exploiting the linearity of expectation, (4) may be expressed as

$$\lambda_{i,t} \cdot \sum_{j=1}^{N} E\big[S(j, l) \mid \boldsymbol{\beta}(l) = \mathbf{b}\big] = E\big[y_{i,t} \mid \boldsymbol{\beta}(l) = \mathbf{b}\big]. \tag{5}$$

Note that, the equation does not contain the *walk time*, i.e., the time elapsed between the end of the service period for a link and the beginning of the service phase of the successive link in the polling cycle. The walk time, however, can be included in the expression of the link service time $S(j, l)$.

The system obtained by (5) for $i = 1, 2, \ldots, N$ and $t = S, M$, gives the equilibrium point $\mathbf{b} = [b_{1,M}, b_{1,S}, \ldots, b_{N,S}]$. In order to solve the system, we have to express the service times $S(j, l)$, and the number of served packets $y_{i,t}$, as functions of $\mathbf{b}$. Such functions are, however, strictly related to the specific polling strategy adopted. In the following, for each one of the strategy proposed, we derive the algebra to solve the system and the stability conditions that guarantee the presence of a single equilibrium point.

3.1. Pure Round Robin

Pure Round Robin (PRR) is the most basic polling schemes. When polled, each station responds with a single packet, which can be either a data or a dummy packet. Then, the master turns to the next slave in the polling cycle. Although, in general, it offers poor performance (see [2] for an in-depth discussion), it is implemented in almost all the Bluetooth devices that are now available on the market. This is essentially due to the necessity of keeping the complexity of the firmware as low as possible, in order to reduce the manufacturing costs and lower the power consumption. PRR does not require complex logic to be embedded on the chip and, thus, it results the most attractive choice for low-cost and power-aware solutions.

For Pure Round Robin (or limited-1 polling, as it is sometimes referred to) we get that each queue behaves like a statistical multiplexer (SMUX) [10], with packet length $2NT$ seconds. Thus, using standard methods [10,15], we easily get the following solution for the equilibrium equation (5):

$$b_{i,t} = \frac{2\lambda_{i,t} NT(1 - \lambda_{i,t} NT)}{1 - 2\lambda_{i,t} NT}. \tag{6}$$

Indeed, since the master has no knowledge of the state of the queues at the slave's side, it has to poll all the slaves at every cycle. In other words, this is a sort of time-division multiple access (TDMA) system, with $2N$ users sharing the same channel on a per-slot basis.

The stability condition, which can be derived from basic queueing theory, requires that the average number of arrivals per cycle, at each node, is less than one. In formula, we have

$$G_{i,t} = \lambda_{i,t} \cdot T < \frac{1}{2N}, \quad i = 1, \ldots, N;\ t = M, S. \tag{7}$$

3.2. Gated Round Robin

In the case of gated service policy, only the packets buffered at the station when it gets the token are served, while the packets that arrive during the service time are set aside to be served at the next cycle. Hence, for the gated policy, we have $y_{i,t}(l) = x_{i,t}(l)$, and (5) becomes

$$b_{i,t} = \lambda_{i,t} \cdot \sum_{j=1}^{N} E\big[S(j, l) \mid \boldsymbol{\beta}(l) = \mathbf{b}\big]. \tag{8}$$

In order to derive the conditioned expectation of the service time $S(j, l)$, we assume that the end of the service phase is signalled by exchanging a couple of void packets (POLL-NULL). The service time is, thus, given by

$$S(j, l) = 2T \cdot \big[1 + \max\{x_{i,M}(l), x_{i,S}(l)\}\big].$$

Consequently, at the equilibrium point, we have

$$\begin{aligned} S(j, l)^{(eq)} &\doteq E\big[S(j, l) \mid \boldsymbol{\beta}(l) = \mathbf{b}\big] \\ &= 2T\big[1 + \max\{b_{i,M}, b_{i,S}\}\big], \end{aligned} \tag{9}$$

where $^{(eq)}$ denotes that the quantity is evaluated at equilibrium. Putting (9) in (8), we get a system of $2N$ non-linear equations of kind

$$b_{i,t} = \lambda_{i,t} \cdot 2T \sum_{j=1}^{N} \left[1 + \max\{b_{i,M}, b_{i,S}\}\right], \tag{10}$$

whose solution gives the equilibrium point for the GRR strategy.

In order to derive the stability condition for the GRR strategy, we refer to [12,16]. The stability condition, hence, may be expressed as

$$\sum_{i=1}^{N} \varrho_i < 1, \tag{11}$$

where ϱ_i is the load factor of the ith master–slave pair. This parameter is usually defined as the average number of packets generated by the link during the average customer service time. In our case, the customer service time is $2T$, i.e., the time needed to serve a single packet from both the master and slave queues. Hence, ϱ_i is given by the expectation of the maximum of two independent Poisson variables, with parameters $2T\lambda_{i,M}$ and $2T\lambda_{i,S}$, respectively. Note, that the above condition does not depend on the walk times of the system. Denoting by $\kappa_{i,t}$ a Poisson random variable with parameter $2T\lambda_{i,t}$, we have

$$\varrho_i = E\left[\max\{\kappa_{i,M}, \kappa_{i,S}\}\right]. \tag{12}$$

Setting

$$\Theta_{\lambda_{i,t}}(n) = \sum_{k=0}^{n} \frac{(2T\lambda_{i,t})^k}{k!},$$

after a few algebra, equation (12) can be written as

$$\varrho_i = \sum_{n=0}^{\infty} 1 - \mathrm{e}^{-2T\lambda_{i,M} - 2T\lambda_{i,S}} \cdot \Theta_{\lambda_{i,M}}(n) \cdot \Theta_{\lambda_{i,S}}(n). \tag{13}$$

Merging (13) with (11), we obtain the stability condition for the gated policy:

$$\sum_{i=1}^{N} \sum_{n=0}^{\infty} 1 - \mathrm{e}^{-\lambda_{i,M} - \lambda_{i,S}} \cdot \Theta_{\lambda_{i,M}}(n) \cdot \Theta_{\lambda_{i,S}}(n) < 1. \tag{14}$$

3.3. Exhaustive Round Robin

In the exhaustive service discipline, the token is held by a link until both master and slave queues are empty. Then, a POLL-NULL cycle takes place and the token is passed to the successive link in the polling cycle.

The number of packets sent at each service phase is, then, given by $y_{i,t}(l) = x_{i,t}(l) + N_{i,t}(S(i,l))$. Consequently, the equilibrium equation (5) can be written as

$$b_{i,t} = \lambda_{i,t} \cdot \sum_{j \neq i} E\left[S(j,l) \mid \boldsymbol{\beta}(l) = \mathbf{b}\right]. \tag{15}$$

The equilibrium equations are completely defined once computed the conditioned expectations of the link service time $S(j,l)$, for each link. In the exhaustive case, this time coincides with the depletion time of a master-slave pair, whose calculation is addressed in appendix A.

Furthermore, we observe that the stability condition for the ERR policy is again (14), as derived for GRR.

4. Performance metrics

In this section, we define the performance metrics considered and provide their expressions at the equilibrium point, for the three polling strategies. Note that, in general, the equilibrium point belongs to the real ($2N$-dimensional) space. Before applying EPA, thus, we need to interpolate somehow the function $E[S(j,l) \mid \boldsymbol{\beta}(l) = \mathbf{b}]$ from $\mathbb{N}^2$ to $\mathbb{R}^2$. A possible solution is presented in appendix B.

4.1. Average packet delay

A classic metric to evaluate the performance of data systems is the average packet delay, i.e., the average time elapsed between the arrival of a new data packet in the system and its departure. Denoting by $D_{i,t}$ the average delay experienced by a packet generated by node (i,t), we define the *average delay* of the system as follows

$$D = \frac{\sum_{i=1}^{N} \sum_{t=M,S} D_{i,t} G_{i,t}}{G}, \tag{16}$$

where G is the total offered traffic, given by $G = T \times \sum_{i=1}^{N} \sum_{t=M,S} \lambda_{i,t}$. The expressions of $D_{i,t}$ for each one of the three polling strategies considered are derived in the followings subsections.

4.1.1. PRR

As previously observed, the PRR case does not require any further investigation. From the literature, we have that the average delay experienced by a packet arriving at queue (i,t), given that the number of packet waiting in the queue when the node gets the token is $b_{i,t}$, can be expressed as [6]:

$$D_{i,t} = \frac{b_{i,t}}{\lambda_{i,t}} - NT + T.$$

4.1.2. GRR

In the gated case, the delay analysis is a typical random look problem. Applying EPA, however, we can simplify the analysis by approximating the expected values of the system parameters by the values they take at the equilibrium. Let τ be the arrival epoch of a generic packet at node (i,t), in a given cycle. Furthermore, let T_C be the time duration of the polling cycle. The delay experimented by the packet before being processed turns out to be

$$d_{i,t} = T_C - \tau + 2T \cdot N_{i,t}(\tau) + T.$$

Taking expectations,

$$D_{i,t} = E[T_C] - E[\tau] + 2T\lambda_{i,t} E[\tau] + T.$$

Since the arrivals are uniformly distributed in $[0, T_C)$, we have that $E[\tau] = E[T_C]/2$, and, hence, we get:

$$D_{i,t} = T + E[T_C] \cdot (0.5 + T\lambda_{i,t}). \tag{17}$$

On the basis of the previous observation, we can replace the expectation of T_C with the value $T_C^{(eq)}$ it takes at the equilibrium point. This value, in turn, is given by $T_C^{(eq)} = \sum_{i=1}^{N} S(i)^{(eq)} + rN$, where r is the walk time. Operating these substitutions, we get

$$D_{i,t} = T + \left(\sum_{i=1}^{N} S(i)^{(eq)} + rN\right) \cdot (0.5 + T\lambda_{i,t}), \tag{18}$$

where $S(i)^{(eq)}$ is given by (9).

4.1.3. ERR

In order to evaluate the mean packet delay for the exhaustive case, we use the same approach applied to the gated discipline. In this case, however, we have to distinguish two possibilities, namely packets arriving during service and during vacation periods.

Let us start considering the first situation, i.e., packets arriving during the service phase. On the basis of our definition of the exhaustive service policy, the service phase ends when *both* the master and slave queues are empty and, thus, a POLL-NULL cycle takes place. The service period for each queue can, then, be divided into two regions. The first region, whose length is denoted by S_a, extends from the beginning of the service period to the epoch when the queue is emptied for the first time. Then the queue enters the second region, of length S_b, in which it behaves like a statistical multiplexer with service time $2T$. Let us now consider a packet arriving τ_{1a} seconds after the beginning of the service phase of the node (i, t), with $\tau_{1a} < S_a$. This packet undergoes a delay $d_{i,t}$ equal to the time needed to get rid of all the packets in queue at the arrival epoch, plus the time T to serve the packet itself. Hence, we have

$$d_{i,t}^{(1a)} = 2T \cdot \big(x_{i,t}(l) + N_{i,t}(\tau_{1a})\big) - \tau_{1a} + T, \tag{19}$$

where, as usual, $x_{i,t}(l)$ is the number of packets buffered in the queue when it gets the token and $N_{i,t}(\tau_{1a})$ denotes the number of arrivals between the beginning of the service phase and the arrival we consider. Taking expectations, we get

$$D_{i,t}^{(1a)} = 2T\big(E\big[x_{i,t}(l)\big] + \lambda_{i,t} E[\tau_{1a}]\big) - E[\tau_{1a}] + T. \tag{20}$$

Assuming the arrivals are uniformly distributed, we may set

$$E[\tau_{1a}] = \frac{E[S_a(i)]}{2}.$$

Thus, we get

$$D_{i,t}^{(1a)} = 2TE\big[x_{i,t}(l)\big] + E\big[S_a(i)\big]\frac{2T\lambda_{i,t} - 1}{2} + T, \tag{21}$$

which gives the average delay of a packet arriving at node (i, t) during the first region of the service period.

Now, let us consider a packet arriving τ_{1b} seconds after the queue enters in the second region. The average delay of the packet, in this case, can be easily calculated by exploiting the well known PASTA theorem [10], which offers

$$D_{i,t}^{(1b)} = T + \frac{2T}{1 - 2T\lambda_{i,t}} = \frac{T(3 - 2T\lambda_{i,t})}{1 - 2T\lambda_{i,t}}. \tag{22}$$

Finally, let us consider a packet that arrives during the vacation period. Let τ_2 be the arrival epoch with respect to the end of the previous service phase for the corresponding node. In other words, we assume the packet arrives τ_2 seconds after the node has released the token. Denoting by S the service time of the link and by $N_{i,t}(\tau_2)$ the number of packets generated from the end of service period to the packet arrival considered, we can express the delay the packet undergoes as

$$d_{i,t}^{(2)} = T_C - S - \tau_2 + 2T \cdot N_{i,t}(\tau_2) + T. \tag{23}$$

Taking expectations, we obtain

$$D_{i,t}^{(2)} = E[T_C] - E[S] - E[\tau_2](1 - 2T\lambda_{i,t}) + T. \tag{24}$$

Assuming, once again, the uniform distribution of arrivals, we get

$$E[\tau_2] = \frac{E[T_C] - E[S]}{2}. \tag{25}$$

Substituting (25) in (24), we come to the final expression

$$D_{i,t}^{(2)} = \big(E[T_C] - E[S]\big)\frac{1 + 2T\lambda_{i,t}}{2} + T. \tag{26}$$

In short, we have partitioned the service cycle into three regions. Then, we have derived the average packet delay for packets arriving in each of such regions. The overall average delay for the ERR discipline is given by the weighted sum of these three contributions. We weight each contribution proportionally to the duration of the corresponding region with respect to the average service cycle duration. As usual, we approximate the expectations with the equilibrium values, i.e., $E[x_{i,t}(l)]$ is replaced by $b_{i,t}$ and $E[T_C]$ by $T_C^{(eq)} = \sum_{i=1}^{N} S(i)^{(eq)} + Nr$, where $S(i)^{(eq)}$ can be derived as explained in appendix A.

Furthermore, we have

$$S_a(i,t)^{(eq)} = \frac{2T \cdot b_{i,t}}{1 - 2T\lambda_{i,t}},$$
$$S_b(i,t)^{(eq)} = S(i)^{(eq)} - S_a(i,t)^{(eq)}.$$

Then, the average delay for the ERR policy can be estimated as

$$D_{i,t} = \frac{S_a(i,t)^{(eq)} D_{i,t}^{(1a)} + S_b(i,t)^{(eq)} D_{i,t}^{(1b)}}{T_C^{(eq)}} + \frac{(T_C^{(eq)} - S(i)^{(eq)}) D_{i,t}^{(2)}}{T_C^{(eq)}}, \tag{27}$$

4.2. Channel utilization

In wireless system, bandwidth is a scarce resource and, hence, it is of vital importance to fully exploit the available bandwidth. The time division duplex (TDD) architecture of Bluetooth systems makes impossible to fulfil this requirement in the case of strongly asymmetric traffic. (An example may be a multicasting in a conference room, where downlink traffic only is present). Hence, a metric of great interest is the so-called *channel utilization parameter*, defined as the average percentage of slots occupied by data packets. The channel utilization parameter, which will be denoted by η, may be obtained by the ratio of the total number of data packets sent during a service cycle and the cycle duration expressed in slots. At the equilibrium point, after some algebra, we come to the followings approximate expressions for the channel utilization parameter in the three cases considered:

$$\eta(PRR) = T \sum_{i=1}^{N} \sum_{t=M,S} \lambda_{i,t},$$

$$\eta(GRR) = \frac{\sum_{i=1}^{N} \sum_{t=M,S} b_{i,t}}{2N + 2 \cdot \sum_{i=1}^{N} \max\{b_{i,M}, b_{i,S}\}},$$

$$\eta(ERR) = \frac{\sum_{i=1}^{N} \sum_{t=M,S} b_{i,t} + \lambda_{i,t} \cdot S(i)^{(eq)}}{2N + (1/T) \sum_{i=1}^{N} S(i)^{(eq)}}.$$

Note that, in the previous expressions, the walk time has been set to $2T$ (POLL-NULL cycle).

4.3. Fairness

The fairness for wireless networks can be defined in many different ways [18]. Nevertheless, the literature still lacks to give a unique and widely accepted definition for the fairness. Basically, the fairness can be intended in two ways: a network can be fair in terms of bandwidth allocated to each user (the more the traffic, the more the bandwidth) or in terms of the delay experienced by a packet. (Note that, under the assumption of independence of the arrival processes, the delay one packet undergoes depends only on the source and not on the destination.) If we focus on the delay (which is meaningful since, in real networks, we often want to ensure a given Quality of Service), we can proceed as follows.

We assume that, in a "perfectly fair" network, each queue experiences the same average packet delay. The rational behind this assumption is the following [6]. We consider a sort of *deus ex machina*, which knows the instantaneous state of the queues at each time and schedules all the packets in a FIFO manner, independently of where they are originated. It is, then, trivial to understand that, in such a situation, each link undergoes the same average packet delay, which, in turn, gives a sort of theoretical validation of our choice.

In order to formalize this intuitive definition of fairness, we introduce the following parameter:

$$\widetilde{D} = \frac{1}{2N} \sum_{i=1}^{N} \sum_{t=M,S} D_{i,t},$$

where $D_{i,t}$ gives the average delay experimented by packets generated at node (i, t), as defined in section 4.1. Comparing $\widetilde{D}$ with the average system delay D, given by (16), we can note that the two parameters trivially coincide for balanced systems, while, in general, they are different. A measure of fairness can, hence, be given in terms of the covariance of the delay vector $\Delta = (1/T) \cdot [D_{1,M}, D_{1,S}, \ldots, D_{N,S}]$. In symbols, the delay covariance is given by

$$C = \frac{1}{2N} \sum_{i=1}^{N} \sum_{t=M,S} \left(\frac{D_{i,t} - \widetilde{D}}{T} \right)^2,$$

and, thus, the fairness index can be expressed as

$$F = \frac{1}{1 + C}. \tag{28}$$

5. Simulation results

To validate the results of our analysis, a simulator has been implemented on the OPNET software platform. We consider three scenarios that, on the one hand, are representatives of relevant situations for a WPAN and, on the other hand, offer significant insight into the disciplines behavior. In the first two scenarios, we consider a piconet consisting of three active slaves, i.e., three connections ($N = 3$), while in the third scenario we refer to a full piconet ($N = 7$).

In scenario 1, two connections are asymmetric, with traffic flowing from master to slave (down-link) only, so that $\lambda_{1,M}T = \lambda_{2,M}T = 0.01$, while $\lambda_{1,S} = \lambda_{2,S} = 0$. The third connection, instead, carries balanced traffic, $\lambda_{3,M} = \lambda_{3,S}$, and the load is increased till the system approaches saturation. Figure 1 reports the theoretical (continuous lines) and simulation (markers) results for the three polling strategies, in terms of average delay. A first evidence from the graph is the fairly good match shown by simulation and analytical curves, which proves the accuracy of the mathematical method proposed. Consequently, we evaluate the channel utilization parameter and fairness as functions of the offered traffic by means of the theoretical analysis only. The curves obtained are reported in figures 2 and 3, respectively. The second scenario is similar to the former, but for the third connection, which is assumed to be completely asymmetric, with traffic flowing in the down-link direction only. Hence, we set $\lambda_{3,S} = 0$, while $\lambda_{3,M}$ is increased till the system approaches instability. The results, in terms of average packet delay, channel utilization and fairness are reported in figures 4, 5 and 6, respectively.

We note that, as far as delay is concerned, ERR clearly outperforms GRR that, in turn, behaves better than PRR. Pure Round Robin seems to be a good choice at low traffic loads, but it quickly reaches the stability limit, making it not suitable for bandwidth-demanding services. Gated Round Robin shows good performances both in terms of fairness and channel utilization, but the average delay increases dramatically as the system approaches stability limit. Exhaustive Round Robin shows the best overall performance in terms of delay,

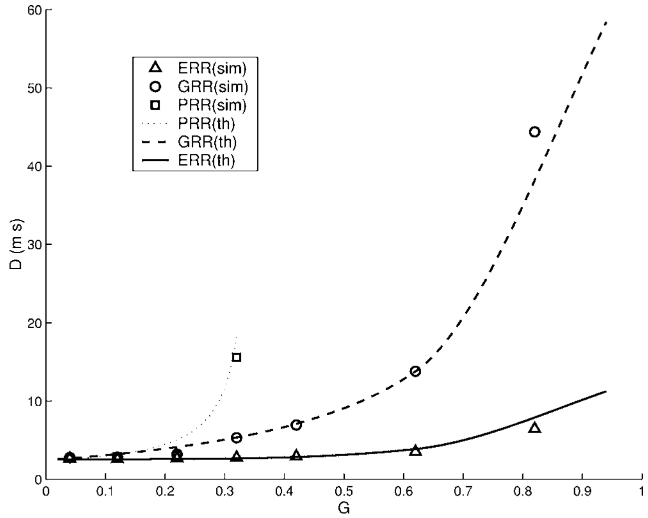

Figure 1. Average packet delay, scenario 1.

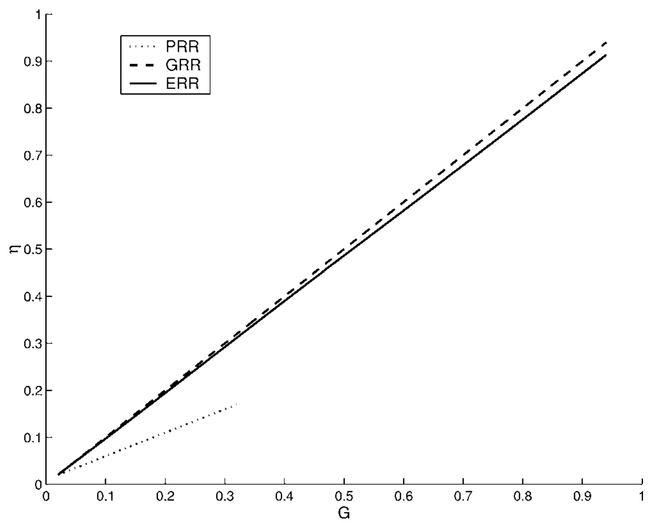

Figure 2. Channel utilization, scenario 1.

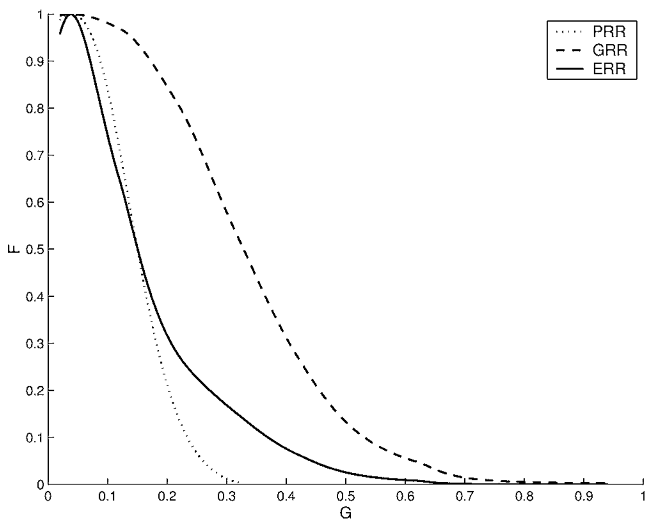

Figure 3. Measure of fairness, scenario 1.

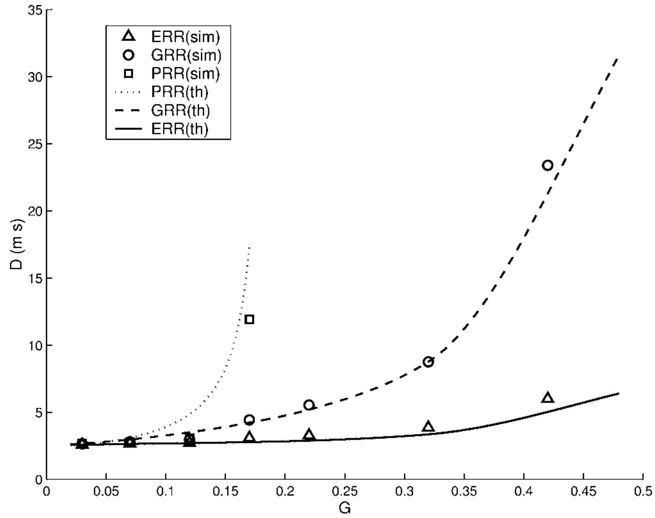

Figure 4. Average packet delay, scenario 2.

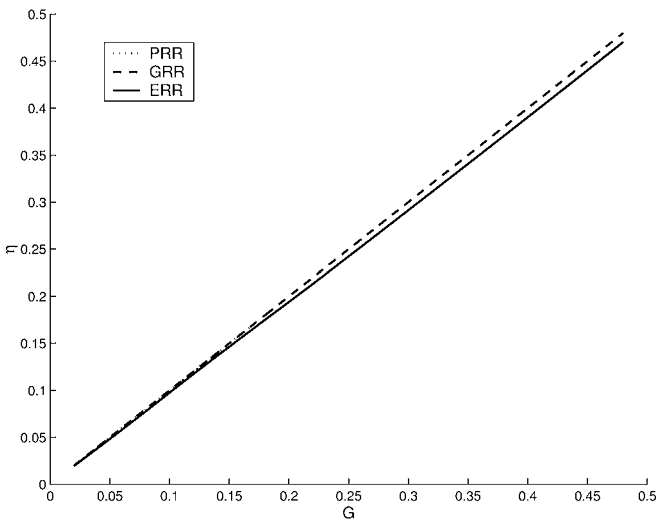

Figure 5. Channel utilization, scenario 2.

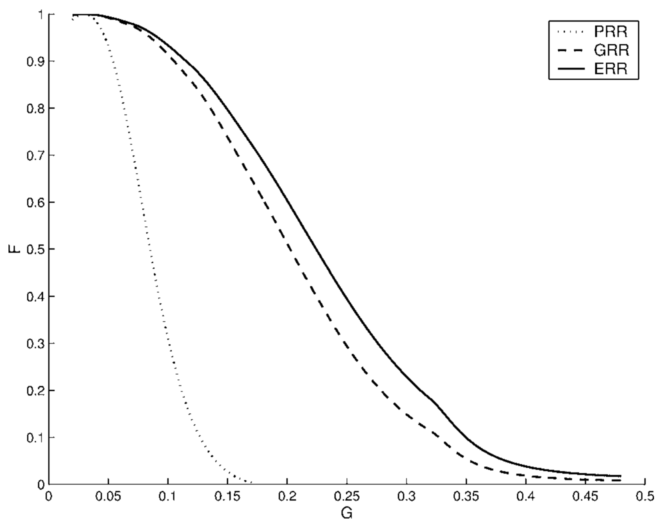

Figure 6. Measure of fairness, scenario 2.

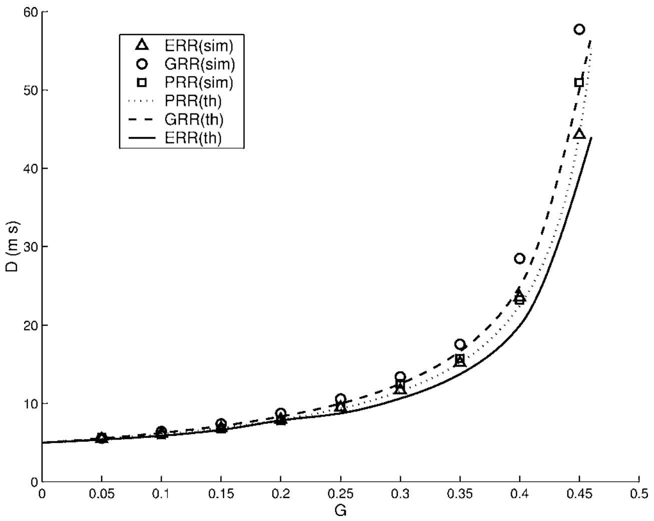

Figure 7. Average packet delay, scenario 3.

and has been proved to attain good fairness in the asymmetric case, while showing a remarkable performance worsening in the balanced case. This is essentially due to the channel capture problem, which, moreover, may cause serious problems when Quality of Service (QoS) requirements are considered. Thus, exhaustive-based schemes seem to be the best choice for increasing performance in a Bluetooth piconet, but particular attention has to be paid in the design of mechanisms able to keep fairness while retaining the good performance attained.

In the third scenario, we considered a full piconet ($N = 7$), which models a typical multicast situation. Only down-link traffic is present and the load is equally shared among the slaves ($\lambda_{i,S} = 0\ \forall i$, $\lambda_{i,M} = \lambda_{j,M}\ \forall i, j$). Figure 7 reports the delay curves for this case. We can note that there is no remarkable difference among the three disciplines considered. The same observation could be made for the channel utilization and fairness curves, which have been omitted due to their scarce significance.

From the situations analyzed, we can draw some general considerations on the behavior of the studied algorithms. In the balanced case, which is usually considered in the literature (e.g., [3]), the three polling schemes achieve, roughly, equal performance. On the contrary, when the traffic load is strongly unbalanced, which is a likely situation to take place in WPANs, GRR and ERR policies clearly outperform basic PRR, both in terms of average packet delay and channel utilization. As far as fairness is considered, we notice that the intuitive mind of PRR being the most fair protocol is far from reality in some notable cases.

6. Conclusion

In this paper we have presented an analytical approach to performance evaluation of Bluetooth polling schemes. Under the assumption of independent Poisson arrivals, we have developed a Markovian model of the system that we have studied at the equilibrium point. The results obtained by the theoretical analysis have been validated through extensive computer simulations, for three different scenarios.

Two subjects seem to be of great interest for future work. The first is the study of semi-exhaustive policies, able to keep the low average delay shown by ERR, while attaining a higher degree of fairness. The second is the design of new polling schemes, able to adapt dynamically to the variation of the traffic conditions in order to satisfied QoS requirements.

Appendix A. Depletion time of a master–slave couple

In order to solve (15) we need to calculate the expected link service time for the exhaustive discipline, given the initial state of the master and slave queues. As previously noted, it coincides with the average depletion time of a master–slave couple, given the initial queues condition. We define the random vector $\mathbf{s}(k) = (x_M(k), x_S(k))$ as the state of the master–slave couple at the kth iteration. (We assume that $k = 0$ corresponds to the moment the couple enters the service phase.) Trivially, the Pollaczeck–Khinchine equation [10] holds and the system is Markovian (as long as the couple is in service). Moreover, if the system is stable, i.e., condition (14) holds, the chain is ergodic and the complete statistics of the depletion time can be computed by means of standard methods [9].

Let $\tau_{s_i s_j}$ be the time that the system takes to enter, for the first time, the state s_j, starting from the state s_i. Once defined the states $s_1 = (i_1, i_2)$ and $s_2 = (0, 0)$, we need to find the expectation of $\tau_{s_1 s_2}$, i.e., $\overline{\tau}_{s_1 s_2} = E[\tau_{s_1 s_2}]$. In order to compute this expected value, we resort to a well-known mathematical expedient [9]. We define a bijective function $\varphi(\cdot)$ that maps the two-dimensional state space of the Markov chain into a subset of $\mathbb{N}$, such that $\varphi(s_2) = 0$. In practice, the function induces a numbering of the states of the Markov chain. Let $\mathbf{\Pi}$ be the transition probability matrix (t.p.m.) of our system, such that $\Pi(i, j)$ is the one-step transition probability from s_x to s_y, with $\varphi(s_x) = i$ and $\varphi(s_y) = j$. Thus, for each $j \in \mathbb{N}$, we have $\Pi(j, 0) = 1 - \sum_{k \neq 0} \Pi(j, k)$. Let $\boldsymbol{\pi}_0$ be the first column of the matrix $\mathbf{\Pi}$, i.e., the one corresponding to the state $s_2 = 0$. Now, let us consider the reduced t.p.m. $\mathbf{Q}$, obtained by replacing the first column and the first row of $\mathbf{\Pi}$ with zeros. Note that this corresponds to transform s_2 into an absorbing state. In a few passages we get the following relation:

$$(\mathbf{I} - \mathbf{Q})^{-1} \cdot \boldsymbol{\pi}_0 = \mathbf{w}, \tag{A.1}$$

where $\mathbf{w} = [1 \ldots 1]'$. Denoting with $\phi_i(z)$ the probability generating function of $\tau_{i,0}$, that is, the time the system takes to enter the state $s_2 = 0$ starting from $s_1 = i$, we get that

$$\overline{\tau}_{i,0} = \left.\frac{\mathrm{d}\phi_i(z)}{\mathrm{d}z}\right|_{z=1}.$$

Collecting in the vector $\mathbf{\Phi}(z) = [\phi_1(z), \ldots, \phi_{N-1}(z)]'$, we finally get [9]:

$$\overline{\boldsymbol{\tau}}_0 = \begin{bmatrix} \overline{\tau}_{1,0} \\ \overline{\tau}_{2,0} \\ \vdots \\ \overline{\tau}_{N-1,0} \end{bmatrix} = \left.\frac{\mathrm{d}\boldsymbol{\Phi}(z)}{\mathrm{d}z}\right|_{z=1} = (\mathbf{I} - \mathbf{Q})^{-1}\mathbf{w}. \quad \text{(A.2)}$$

Appendix B. Interpolation of a function from $\mathbb{N}^2$ to $\mathbb{R}^2$

In this section we propose a method for extending a function from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{R} \times \mathbb{R}$ in such a way that the resulting function is continuous. Let $p = (x, y)$ be a generic point in the positive quadrangle of the real plane. We associate to p four points in the integer lattice, defined as $p_1 = (\lfloor x \rfloor, \lfloor y \rfloor)$, $p_2 = (\lceil x \rceil, \lfloor y \rfloor)$, $p_3 = (\lceil x \rceil, \lceil y \rceil)$, $p_4 = (\lfloor x \rfloor, \lceil y \rceil)$, where the symbols $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ denote the *ceiling* and *floor* functions, respectively. The points p_1, p_2, p_3, p_4 are the vertices of a unitary square that contains p. Let us denote by $z_i = f(p_i)$ the value our function takes on point p_i, and by d_i the Euclidean distance between p and p_i. We want to find the z that minimizes the function $g(z)$, defined as

$$g(z) = \sum_{i=1}^{4} \frac{(z - z_i)^2}{d_i^2}. \quad \text{(B.1)}$$

Note that, the choice of the function $g(z)$ is somehow arbitrary. However, our choice seems reasonable and lends itself to a straightforward analysis. After a few passages, we derive the value of z which minimizes (B.1), given by

$$z = \frac{\sum_{i=1}^{4} z_i/d_i^2}{\sum_{i=1}^{4} 1/d_i^2}.$$

It is easy to verify that our function can be extended by continuity on the points z_i, leading to a continuous function defined on $\mathbb{R}_+^2$. In general, this may not be enough for our aims, since the numerical solution of the system requires a function defined on $\mathbb{R}^2$, as previously stated. Hence, we associate to each $p = (x, y) \in \mathbb{R}^2$ a point $\tilde{p} \in \mathbb{R}_+^2$, defined as $\tilde{p} = (\max\{x, 0\}, \max\{y, 0\})$. Applying the same reasoning to $\tilde{p}$, we are done.

References

[1] R. Bruno, M. Conti and E. Gregori, Wireless access to the Internet via Bluetooth: performance evaluation of the EDC scheduling algorithm, in: *Proc. of 1st Workshop on Wireless Mobile Internet*, Rome (July 2001).

[2] R. Bruno, M. Conti and E. Gregori, Bluetooth: architecture, protocols and scheduling algorithms, ACM Cluster Computing 5 (April 2002) 117–131.

[3] A. Capone, M. Gerla and R. Kapoor, Efficient polling schemes for Bluetooth picocells, in: *Proc. of IEEE ICC* (2001).

[4] A.B. Carleial and M.E. Hellman, Bistable behavior of ALOHA-type systems, IEEE Trans. Comm. 23 (April 1975) 401–410.

[5] S. Garg, M. Kalia and R. Shorey, MAC scheduling policies for power optimization in Bluetooth: a master driven TDD wireless system, in: *Proc. of IEEE VTC* (May 2000).

[6] J.L. Hammond and P.J.P. O'Reilly, *Performance Analysis of Local Computer Networks* (Addison-Wesley, Reading, MA, 1986).

[7] M. Kalia, D. Bansal and R. Shorey, MAC scheduling and SAR policies for Bluetooth: a master driven TDD pico-cellular wireless system, in: *Proc. of IEEE MoMuc* (November 1999).

[8] M. Kalia, D. Bansal and R. Shorey, Data scheduling and SAR for Bluetooth MAC, in: *Proc. of IEEE VTC*, Tokyo (May 2000).

[9] J.G. Kemeny and J.L. Snell, *Finite Markov Chains* (Van Nostrand-Reinhold, New York, 1960).

[10] L. Kleinrock, *Queueing Systems* (Wiley, New York, 1975).

[11] L. Kleinrock and S.S. Lam, Packet switching in a multiaccess broadcast channel: performance evaluation, IEEE Trans. Comm. 23 (April 1975) 410–423.

[12] H. Levy and M. Sidi, Polling systems: applications, modeling and optimization, IEEE Trans. Comm. 38 (October 1990) 1750–1760.

[13] Z. Liu, P. Nain and D. Towsley, On optimal polling policies, Queueing Systems 11 (1992) 59–83.

[14] M. Sidi, H. Levy and S.W. Fuhrmann, A queueing network with a single cyclically roving server, Queuing Systems Theory Appl. 11 (1992) 121–144.

[15] H. Takagi, *Analysis of Polling Systems* (MIT Press, Cambridge, MA, 1986).

[16] S. Tasaka, Stability and performance of the R-ALOHA packet broadcast system, IEEE Trans. Comput. 32 (August 1983) 717–726.

[17] M.E. Woodward, Equivalence of approximation techniques for solution of multidimensional Markov chains in network modelling, Electron. Lett. 27 (Juny 1991) 1019–1021.

[18] E.A. Yaiz and G. Heijenk, Polling in Bluetooth: a simplified best effort case, in: *Proc. of IEEE WPMC* (2001).

Daniele Miorandi was born in Rovereto (Trento), Italy on December 26, 1977. He received the Dr. Ing. degree (summa cum laude) in telecommunications engineering from the University of Padova, Padova (Italy) in 2001. He is currently a Ph.D. student at the Department of Information Engineering of University of Padova. His research interests lie in the area of wireless networking, with emphasis on the theoretical aspects involved. In particular, he has been working on medium access control issues, routing protocols for ad-hoc networks and performance evaluation of Bluetooth networks.
E-mail: daniele.miorandi@dei.unipd.it

Andrea Zanella received the Ph.D. degree in telecommunication engineering from the University of Padova, Italy, in 2002. Prior to that he received the Dr. Ing. degree (comparable to Master degree) in computer engineering in 1998, still from the University of Padova. He spent nine months, in 2001, as post-doc researcher at the Department of Computer Science of the University of California, Los Angeles (UCLA), where he was engaged in research on Wireless Networks and Wireless Access to Internet under the supervision of Prof. Mario Gerla. Currently, he is a research fellow in the Department of Information Engineering of the University of Padova, Italy. His research interests are mainly focused on topics related to wireless and mobile internetworking. In particular, in the last period, he has been working on the performance aspects of wireless personal area networks based on the Bluetooth standard.
E-mail: andrea.zanella@dei.unipd.it

Gianfranco Pierobon was born in Alleghe (Belluno), Italy, on September 24, 1941. He received the Dr. Ing. degree in electrical engineering from the University of Padova, Padova, Italy, in 1968. In 1969, he joined the Dipartimento di Elettrotecnica e Elettronica of the University of Padova as an Assistant Professor in the area of microwaves. Since 1975, he has been an Associate Professor teaching probability theory and statistics. Since 1986, he is Full Professor at the Department of Information Engineering of the University of Padova and teaches communication networks. His current research interests lie in the general area of communications and information theory with emphasis on wireless ad-hoc networks and space-time coding.
E-mail: pierobon@dei.unipd.it